

# IMPERIAL

Department of Mathematics

## Probabilistic Sequential Matrix Factorisation for 12-Lead ECG Data

Joana Levcheva

CID: 01252821

Supervised by Dr Deniz Akyildiz

August 19, 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 Levcheva

Date: August 19, 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 novel applications of the probabilistic sequential MF algorithm Probabilistic Sequential Matrix Factorisation (PSMF) ([Ömer Deniz Akyıldız et al. \(2021\)](#)) to 12-lead ECG data. We explore three 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) ([Ömer Deniz Akyıldız et al. \(2021\)](#)) and compare their performance with other probabilistic sequential MF algorithms, R-peaks detection, and forecasting an ECG component based on previous normal heart beats 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 complex ECG data, and summarise the experiments results. We find that PSMF performs well when used for imputing missing data, but when it comes to forecasting there are certain challenges which open the door for further research on extending the PSMF algorithm to better handle the complex structure of ECG data.

## Acknowledgements

TODO

# Contents

<b>1. Introduction</b>	<b>1</b>
1.1. Contributions . . . . .	3
1.2. Notation . . . . .	3
<b>2. Background</b>	<b>5</b>
2.1. Preliminaries . . . . .	5
2.1.1. Matrix Normal Distribution . . . . .	5
2.1.2. Kronecker Product . . . . .	5
2.1.3. Fourier Series/Basis . . . . .	5
2.2. PSMF . . . . .	6
2.2.1. Model . . . . .	6
2.2.2. Inference . . . . .	6
2.2.3. Parameter Estimation . . . . .	8
2.2.4. Approximating the marginal-likelihood . . . . .	9
2.2.5. PSMF Algorithm . . . . .	9
2.3. rPSMF . . . . .	9
2.3.1. Model . . . . .	10
2.3.2. Inference . . . . .	10
2.3.3. rPSMF Algorithm . . . . .	11
2.4. PSMF for Handling Missing Data . . . . .	12
2.4.1. Model . . . . .	12
2.4.2. Inference . . . . .	13
2.4.3. Algorithm . . . . .	14
2.5. Introduction to ECG . . . . .	15
<b>3. Experiments and Results</b>	<b>18</b>
3.1. Missing Data Imputation . . . . .	18
3.2. R-peaks Detection . . . . .	21
3.3. Forecasting . . . . .	21
<b>4. Conclusion</b>	<b>23</b>
4.1. Future work . . . . .	23
<b>A. Missing Data Imputation Results</b>	<b>A1</b>
A.1. PSMF . . . . .	A1
A.1.1. $r = 3$ . . . . .	A1
A.1.2. $r = 10$ . . . . .	A2

A.2.	rPSMF . . . . .	A3
A.2.1.	$r = 3$ . . . . .	A3
A.2.2.	$r = 10$ . . . . .	A4
A.3.	MLE-SMF . . . . .	A5
A.3.1.	$r = 3$ . . . . .	A5
A.3.2.	$r = 10$ . . . . .	A6
A.4.	TMF . . . . .	A7
A.4.1.	$r = 3$ . . . . .	A7
A.4.2.	$r = 10$ . . . . .	A8

# 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

$$Y \approx CX, \quad (1.1)$$

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

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

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 \(2007\)](#)) - non-sequential, Dictionary filtering ([Akyildiz and Míguez \(2019\)](#)), Probabilistic Sequential Matrix Factorization (PSMF) ([Ömer Deniz 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). Particularly for ECG data, ECG has a nonlinear SSM which doesn't suit the dictionary filtering.

Two years later, Akyildiz et.al. develop Probabilistic Sequential Matrix Factorization (PSMF) ([Ömer Deniz Akyildiz et al. \(2021\)](#)). This method is tailored to time-varying and non-stationary datasets consisting of high-dimensional time-series. Nonlinear Gaus-

sian 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:

- We show three novel applications of PSMF to complex 12-lead ECG data.
- Application 1: We apply 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 heart beats 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  $Y \in \mathbb{R}^{m \times n}$ , with  $y_k$  the  $k$ -th column of the matrix, 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  $Y$ :

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

where  $y_1, \dots, y_n$  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  $y_{1:k} = \{y_1, \dots, y_k\}$  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 coefficients matrix, and  $r$  be the approximation rank.

With  $I_d \in \mathbb{R}^{d \times d}$  we are going to denote the identity matrix, with  $\mathcal{N}(x; \mu, \Sigma)$  the multivariate normal distribution with mean  $\mu$  and  $\Sigma$  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, and with  $\mathcal{IG}(s; \alpha, \beta)$  the inverse gamma distribution with shape  $\alpha$  and scale  $\beta$ , and finally with  $\mathcal{T}(x; \mu, \Sigma, \lambda)$  the multivariate  $t$  distribution, where  $\mu$  is the mean,  $\Sigma$  the scale matrix, and  $\lambda$  is the degrees of freedom.

## 2. Background

### 2.1. Preliminaries

Intro

#### 2.1.1. Matrix Normal Distribution

**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  $\text{vec}(X) \sim \mathcal{N}(\text{vec}(M), U \otimes V)$ .

Also, if  $X \sim \mathcal{MN}(X; M, U, V)$ , then  $x \sim \mathcal{N}(x; \text{vec}(M), U \otimes V)$ , where  $x = \text{vec}(X)$  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 \(2000\)](#))

$$(2\pi)^{-\frac{1}{2}np}|U|^{-\frac{1}{2}n}|V|^{-\frac{1}{2}p} \exp\left\{-\frac{1}{2}\text{tr}(U^{-1}(X - M)V^{-1}(X - M)^T)\right\}, \quad (2.1)$$

where  $|U|$  and  $|V|$  are the determinants of  $U$  and  $V$  respectively,  $\text{tr}(\cdot)$  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

**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 \(1997\)](#)):

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix}, \quad (2.2)$$

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

Some identities?

#### 2.1.3. Fourier Series/Basis

TODO

## 2.2. PSMF

Intro

### 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 ([Ömer Deniz Akyildiz et al. \(2021\)](#)):

$$p(C) = \mathcal{MN}(C; C_0, I_d, V_0) \quad (2.3)$$

$$p(x_0) = \mathcal{N}(x_0; \mu_0, P_0) \quad (2.4)$$

$$p_\theta(x_k | x_{k-1}) = \mathcal{N}(x_k; f_\theta(x_{k-1}), Q_k) \quad (2.5)$$

$$p(y_k | x_k, C) = \mathcal{N}(y_k; Cx_k, R_k) \quad (2.6)$$

where the nonlinear mapping  $f_\theta : \mathbb{R}^r \times \Theta \rightarrow \mathbb{R}^r$  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 (2.5) and the observation model in (2.6) 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.

We are going to refer to Equation (2.3) as the dictionary prior, Equation (2.4) as the initial state of the coefficients, Equation (2.5) as the transition density, and Equation (2.6) as the observation model.

### 2.2.2. Inference

In this section, we outline the algorithm for conducting sequential inference ([Ömer Deniz Akyildiz et al. \(2021\)](#)) within the model defined by equations (2.3)–(2.6). 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.

#### Inference

For the optimal inference recursions we have  $\theta$  fixed, and we assume we know the 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:

$$p(x_k | y_{1:k-1}) = \int p(x_{k-1} | y_{1:k-1})p(x_k | x_{k-1})dx_{k-1}. \quad (2.7)$$

Given that past marginal is known, (2.7) is independent of the dictionary.

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

$$p(y_k | y_{1:k-1}) = \int \int p(y_k | c, x_k) p(x_k | y_{1:k-1}) p(c | y_{1:k-1}) dx_k dc. \quad (2.8)$$

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

$$p(c | y_{1:k}) = p(c | y_{1:k-1}) \frac{p(y_k | c, y_{1:k-1})}{p(y_k | y_{1:k-1})}, \quad (2.9)$$

where

$$p(y_k | c, y_{1:k-1}) = \int p(y_k | c, x_k) p(x_k | y_{1:k-1}) dx_k. \quad (2.10)$$

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

$$p(x_k | y_{1:k}) = p(x_k | y_{1:k-1}) \frac{p(y_k | x_k, y_{1:k-1})}{p(y_k | y_{1:k-1})}, \quad (2.11)$$

where

$$p(y_k | x_k, y_{1:k-1}) = \int p(y_k | x_k, c) p(c | y_{1:k-1}) dc. \quad (2.12)$$

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

### Approximate Sequential Inference

In this section we make the recursions tractable by through using approximations, and outline the approximate sequential inference. Equation (2.3) can be rewritten as  $p(c) = \mathcal{N}(c; c_0, V_0 \otimes I_d)$ . Let the given filters at time  $k-1$  be  $p(x_{k-1} | y_{1:k-1}) = \mathcal{N}(x_{k-1}; \mu_{k-1}, P_{k-1})$  and  $p(c | y_{1:k-1}) = \mathcal{N}(c; c_{k-1}, V_{k-1} \otimes I_d)$ . 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  $\tilde{p}(\cdot)$  for the approximate densities when the distribution is not exact.

For the prediction we have to compute (2.7) 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 (Ömer Deniz Akyıldız et al. (2021))

$$\tilde{p}(x_k | y_{1:k-1}) = \mathcal{N}(x_k; \bar{\mu}_k, \bar{P}_k), \quad (2.13)$$

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  $f_\theta$  calculated at  $\bar{\mu}_{k-1}$ .

For the update step we are once again interested in the dictionary update and the coefficient update. Full derivation of the updates can be found in the PSMF paper ([Ömer Deniz Akyildiz et al. \(2021\)](#)). Here, we are just going to state the update rules.

### Dictionary Update:

$$\tilde{p}(y_k | c, y_{1:k-1}) = \mathcal{N}(y_k; C\bar{\mu}_k, \eta_k \otimes I_d), \quad (2.14)$$

where  $\eta_k = \text{tr}(R_k + C_{k-1}\bar{P}_k C_{k-1}^T)/d$ .

**Text Maybe add the missed proposition for etc here text.**

### Coefficient Update:

$$\tilde{p}(x_k | y_{1:k}) = \mathcal{N}(x_k; \mu_k, P_k), \quad (2.15)$$

with

$$\mu_k = \bar{\mu}_k + \bar{P}_k C_{k-1}^T (C_{k-1}\bar{P}_k C_{k-1}^T + \bar{R}_k)^{-1} (y_k - C_{k-1}\bar{\mu}_k), \quad (2.16)$$

$$P_k = \bar{P}_k - \bar{P}_k C_{k-1}^T (C_{k-1}\bar{P}_k C_{k-1}^T + \bar{R}_k)^{-1} C_{k-1}\bar{P}_k, \quad (2.17)$$

where  $\bar{R}_k = R_k + \bar{\mu}_k^T V_{k-1} \bar{\mu}_k \otimes I_d$ .

### 2.2.3. Parameter Estimation

We need to estimate the parameters of  $f_\theta$  in Equation [2.5](#). In order to do that we need to solve the following optimisation problem

$$\theta^* \in \operatorname{argmax} \log p_\theta(y_{1:n}). \quad (2.18)$$

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**. There is also an online (recursive) version (refer to [Ömer Deniz Akyildiz et al. \(2021\)](#)) that can be used on streaming data but we won't outline here.

### Iterative Estimation:

Performing multiple passes over data with

$$\theta_i = \theta_{i-1} + \gamma \nabla \log \tilde{p}_\theta(y_{1:n})|_{\theta=\theta_{i-1}} \quad (2.19)$$

at the  $i$ -th iteration. Again, because  $\nabla \log p_\theta(y_{1:n})$  is intractable we are going to use the approximation  $\nabla \log \tilde{p}_\theta(y_{1:n}) = \sum_k^n \tilde{p}_\theta(y_k | y_{1:k-1})$ . This is done during forward filtering.

### 2.2.4. Approximating the marginal-likelihood

Given  $\tilde{p}_\theta(y_k|y_{1:k-1}, c) = \mathcal{N}(y_k; Cf_\theta(\mu_{k-1}), \eta_k \otimes I_d)$ , and  $\tilde{p}(c|y_{1:k-1}) = \mathcal{N}(c; c_{k-1}, V_{k-1} \otimes I_d)$ , for the negative log-likelihood we have ([Ömer Deniz Akyildiz et al. \(2021\)](#))

$$-\log \tilde{p}_\theta(y_k|y_{1:k-1}) \stackrel{c}{=} \frac{d}{2} \log \left( \|f_\theta(\mu_{k-1})\|_{V_{k-1}}^2 + \eta_k \right) + \frac{1}{2} \frac{\|y_k - C_{k-1}f_\theta(\mu_{k-1})\|^2}{\eta_k + \|f_\theta(\mu_{k-1})\|_{V_{k-1}}^2}, \quad (2.20)$$

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

### 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 the original PSMF paper ([Ömer Deniz Akyildiz et al. \(2021\)](#)).

---

#### Algorithm 1 Iterative PSMF

---

- 1: Initialize  $\gamma, \theta_0, C_0, V_0, \mu_0, P_0, (Q)_k \geq 1, (R)_k \geq 1$ .
  - 2: **for**  $i \geq 1$  **do**
  - 3:    $C_0 = C_n, \mu_0 = \mu_n, P_0 = P_n, V_0 = V_n$
  - 4:   **for**  $1 \leq k \leq n$  **do**
  - 5:     Compute predictive mean of  $x_k$ :
  - 6:      $\bar{\mu}_k = f_{\theta_{i-1}}(\mu_{k-1})$
  - 7:     Compute predictive covariance of  $x_k$ :
  - 8:      $\bar{P}_k = F_k P_{k-1} F_k^\top + Q_k$ , with  $F_k = \frac{\partial f(x)}{\partial x} \Big|_{x=\bar{\mu}_{k-1}}$
  - 9:     Update dictionary mean  $C_k$ :
  - 10:     $C_k = C_{k-1} + \frac{(y_k - C_{k-1}\bar{\mu}_k)\bar{\mu}_k^T V_{k-1}^T}{\bar{\mu}_k^T V_{k-1} \bar{\mu}_k + \eta_k}$
  - 11:     Update dictionary covariance  $V_k$ :
  - 12:     $V_k = V_{k-1} - \frac{V_{k-1} \bar{\mu}_k \bar{\mu}_k^T V_{k-1}^T}{\bar{\mu}_k^T V_{k-1} \bar{\mu}_k + \eta_k}$
  - 13:     Update coefficient mean  $\mu_k$ :
  - 14:     $\mu_k = \bar{\mu}_k + \bar{P}_k C_{k-1}^T (C_{k-1} \bar{P}_k C_{k-1}^T + \bar{R}_k)^{-1} (y_k - C_{k-1} \bar{\mu}_k)$
  - 15:     Update coefficient covariance  $P_k$ :
  - 16:     $P_k = \bar{P}_k - \bar{P}_k C_{k-1}^T (C_{k-1} \bar{P}_k C_{k-1}^T + \bar{R}_k)^{-1} C_{k-1} \bar{P}_k$
  - 17:     Update parameters:
  - 18:     $\theta_i = \theta_{i-1} + \gamma \nabla \log \tilde{p}_\theta(y_{1:n})|_{\theta=\theta_{i-1}}$
- 

## 2.3. rPSMF

There are cases when the ... cannot be set/specify...

### 2.3.1. Model

The robust variant of PSMF (rPSMF) incorporates robustness to outliers and model misspecifications by integrating a multivariate t-distribution into the framework. The inference and parameter estimation follows the one for PSMF. The rPSMF model is defined as

$$p(s) = \mathcal{IG}(s; \lambda_0/2, \lambda_0/2) \quad (2.21)$$

$$p(C | s) = \mathcal{MN}(C; C_0, I_d, sV_0) \quad (2.22)$$

$$p(x_0 | s) = \mathcal{N}(x_0; \mu_0, sP_0) \quad (2.23)$$

$$p_\theta(x_k | x_{k-1}, s) = \mathcal{N}(x_k; f_\theta(x_{k-1}), sQ_0) \quad (2.24)$$

$$p(y_k | x_k, C, s) = \mathcal{N}(y_k; Cx_k, sR_0), \quad (2.25)$$

where ...

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

$$\mathcal{IG}(s; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \left( \frac{1}{s} \right)^{\alpha+1} \exp \left( -\frac{\beta}{s} \right), \quad (2.26)$$

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

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

$$\mathcal{T}(y; \mu, \Sigma, \lambda) = \frac{1}{(\pi\lambda)^{d/2} |\Sigma|^{1/2}} \frac{\Gamma((\lambda+d)/2)}{\Gamma(\lambda/2)} \left( 1 + \frac{\Delta^2}{\lambda} \right)^{-(\lambda+d)/2}, \quad (2.27)$$

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

### 2.3.2. Inference

Inference in rPSMF incorporates the handling of the robust parameters and scale mixing variable  $s$  introduced to manage outliers and model misspecifications. The introduction of multivariate t distribution leads to an increase of the degrees of freedom in the update equations by  $d$  at every iteration:

$$\lambda_k = \lambda_{k-1} + d. \quad (2.28)$$

If we let

$$\Delta_{1,k}^2 = (y_k - C_{k-1}\bar{\mu}_k)^T (C_{k-1}\bar{P}_k C_{k-1}^T + \bar{R}_k)^{-1} (y_k - C_{k-1}\bar{\mu}_k) \quad (2.29)$$

and

$$\omega_k = (\lambda_{k-1} + \Delta_{1,k}^2) / (\lambda_{k-1} + d), \quad (2.30)$$

then (Ömer Deniz Akyildiz et al. (2021))  $s_0 = s$ ,  $s_k = \omega_k^{-1} s_{k-1}$ , and for the noise covariances update we have  $Q_k = \omega_k Q_{k-1}$ , and  $R_k = \omega_k R_{k-1}$ . 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 2.17 by  $\omega_k$ :

$$P_k = \omega_k \left( \bar{P}_k - \bar{P}_k C_{k-1}^T \left( C_{k-1} \bar{P}_k C_{k-1}^T + \bar{R}_k \right)^{-1} C_{k-1} \bar{P}_k \right). \quad (2.31)$$

If we let

$$\bar{\rho}_k = \bar{\mu}_k^T V_{k-1} \bar{\mu}_k + \eta_k \quad (2.32)$$

and

$$\Delta_{2,k}^2 = \|y_k - C_{k-1} \bar{\mu}_k\|^2 / \bar{\rho}_k, \quad (2.33)$$

then the update of  $V_k$  becomes (**revisit this in PSMF - it was missed, the explicit equation**)

$$V_k = \varphi_k \left( V_{k-1} - \frac{V_{k-1} \bar{\mu}_k \bar{\mu}_k^T V_{k-1}}{\bar{\mu}_k^T V_{k-1} \bar{\mu}_k + \eta_k} \right), \quad (2.34)$$

where  $\varphi_k = (\lambda_{k-1} + \Delta_{2,k}^2) / (\lambda_{k-1} + d)$ .

### 2.3.3. 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 the original PSMF paper (Ömer Deniz Akyildiz et al. (2021)).

**Algorithm 2** Iterative rPSMF

---

1: Initialize  $\gamma, \theta_0, C_0, V_0, \mu_0, P_0, Q_0, R_0$ .

2: **for**  $i \geq 1$  **do**

3:    $C_0 = C_T, \mu_0 = \mu_T, P_0 = P_T, V_0 = V_T$

4:   **for**  $1 \leq k \leq T$  **do**

5:     Predictive mean of  $x_k$ :

6:        $\bar{\mu}_k = f_{\theta_{i-1}}(\mu_{k-1})$

7:     Predictive covariance of  $x_k$ :

8:        $\bar{P}_k = F_k P_{k-1} F_k^\top + Q_k, \quad \text{where } F_k = \frac{\partial f(x)}{\partial x} \Big|_{x=\bar{\mu}_{k-1}}$

9:     Compute scaling factor for the dictionary update

10:       $\varphi_k = \frac{\lambda_{k-1}}{\lambda_{k-1} + d} + \frac{(y_k - C_{k-1}\bar{\mu}_k)^\top (y_k - C_{k-1}\bar{\mu}_k)}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k}$

11:      where  $\eta_k = \text{Tr}(C_{k-1}\bar{P}_k C_{k-1}^\top + R_{k-1})/d$ .

12:     Mean and covariance updates of the dictionary

13:       $C_k = C_{k-1} + \frac{(y_k - C_{k-1}\bar{\mu}_k)\bar{\mu}_k^\top V_{k-1}}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k} \text{ and } V_k = \varphi_k \left( V_{k-1} - \frac{V_{k-1}\bar{\mu}_k\bar{\mu}_k^\top V_{k-1}}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k} \right)$

14:     Compute scaling factor for the coefficient update

15:       $\omega_k = \lambda_{k-1} + (y_k - C_{k-1}\bar{\mu}_k)^\top S_k^{-1} (y_k - C_{k-1}\bar{\mu}_k)$

16:      where  $S_k = C_{k-1}\bar{P}_k C_{k-1}^\top + \bar{R}_k$  and  $\bar{R}_k = R_{k-1} + \bar{\mu}_k^\top V_{k-1}\bar{\mu}_k \otimes I_d$ .

17:     Mean and covariance updates of coefficients

18:       $\mu_k = \bar{\mu}_k + \bar{P}_k C_{k-1}^\top S_k^{-1} (y_k - C_{k-1}\bar{\mu}_k) \text{ and } P_k = \omega_k (\bar{P}_k - \bar{P}_k C_{k-1}^\top S_k^{-1} C_{k-1}\bar{P}_k)$

19:     Update noise covariances:

20:       $Q_k = \omega_k Q_{k-1} \text{ and } R_k = \omega_k R_{k-1}$

21:     Update degrees of freedom:

22:       $\lambda_k = \lambda_{k-1} + d$

23:     Parameter update:

24:       $\theta_i = \theta_{i-1} + \gamma \sum_{k=1}^T \nabla_{\theta} \log p_{\theta}(y_k | y_{1:k-1})|_{\theta=\theta_{i-1}}$

---

## 2.4. PSMF for Handling Missing Data

Intro

### 2.4.1. Model

$$p(C) = \mathcal{MN}(C; C_0, I_d, V_0) \quad (2.35)$$

$$p(x_0) = \mathcal{N}(x_0; \mu_0, P_0) \quad (2.36)$$

$$p_{\theta}(x_k | x_{k-1}) = \mathcal{N}(x_k; f_{\theta}(x_{k-1}), Q_k) \quad (2.37)$$

$$p(z_k | x_k, C) = \mathcal{N}(z_k; M_k C x_k, M_k R_k M_k^T), \quad (2.38)$$

where  $z_k = m_k \odot y_k$ ,  $m_k \in \{0, 1\}^d$  is a mask vector. Zero corresponds to missing entries, and ones otherwise. Also,  $z_k = M_k y_k$ ,  $M_k = \text{diag}(m_k)$ . Reverse this... then state the model definition

### 2.4.2. Inference

Let  $\tilde{p}(c|z_{1:k-1}) = \mathcal{N}(c; c_{k-1}, V_{k-1} \otimes I_d)$  and

$$\tilde{p}(z_k|c, z_{1:k-1}) = \mathcal{N}(z_k; M_k C \bar{\mu}_k, \eta_k \otimes I_d), \quad (2.39)$$

where

$$\eta_k = \frac{\text{tr}(M_k R_k M_k^T + M_k C_{k-1} \bar{P}_k C_{k-1}^T M_k^T)}{m}. \quad (2.40)$$

...

$$\tilde{p}(z_k|c, z_{1:k-1}) = \mathcal{N}(z_k; H_k c, \eta_k \otimes I_d), \quad (2.41)$$

where  $c = \text{vec}(C)$ , and  $H_k = \bar{\mu}_K^T \otimes M_k$ . Using the approximation (+ sth else)  $\bar{\mu}_k^T V_{k-1} \bar{\mu}_k \otimes M_k \approx \bar{\mu}_k^T V_{k-1} I_d$  for the covariance update it follows

$$P_k = V_{k-1} \otimes I_d - \frac{V_{k-1} \bar{\mu}_k \bar{\mu}_k^T V_{k-1}}{\bar{\mu}_k^T V_{k-1} \bar{\mu}_k + \eta_k} \otimes M_k, \quad (2.42)$$

but ... so ...

$$P_k \approx V_k \otimes I_d, \quad (2.43)$$

where  $V_k$  is in the same form of missing-data free updates.

For the mean update we get

$$C_k = C_{k-1} + \frac{(z_k - M_k C_{k-1} \bar{\mu}_k) \bar{\mu}_k^T V_{k-1}}{\bar{\mu}_k^T V_{k-1} \bar{\mu}_k + \eta_k}, \quad k \geq 1. \quad (2.44)$$

To update  $x_k$ , once we fix  $C_{k-1}$ , everything straightforwardly follows by replacing  $C_{k-1}$  by  $M_k C_{k-1}$  in the update rules for  $(x_k)_{k \geq 1}$ . Finally, the negative log-likelihood  $-\log \tilde{p}_\theta(z_k|z_{1:k-1})$  can be derived similarly to the non-missing case in Sec. §3.2.5, and equals

$$-\log \tilde{p}_\theta(z_k|z_{1:k-1}) \stackrel{c}{=} \frac{1}{2} \sum_{j=1}^d \log u_{jk} + \frac{1}{2} (z_k - M_k C_{k-1} f_\theta(\mu_{k-1}))^\top U_k^{-1} (z_k - M_k C_{k-1} f_\theta(\mu_{k-1})), \quad (2.45)$$

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

TODO: updates related to  $x_k$ ?

### 2.4.3. Algorithm

WIP and TODO: add  $\eta_k$ , correct iterative update parameter

---

**Algorithm 3** PSMF with Missing Data Imputation

---

- 1: Initialize  $\gamma, \theta_0, C_0, V_0, \mu_0, P_0, (Q)_{k \geq 1}, (R)_{k \geq 1}$ , and missing data mask  $M$ .
  - 2: **for**  $i \geq 1$  **do**
  - 3:    $C_0 = C_n, \mu_0 = \mu_n, P_0 = P_n, V_0 = V_n$
  - 4:   **for**  $1 \leq k \leq n$  **do**
  - 5:     Compute predictive mean of  $x_k$ :
  - 6:      $\bar{\mu}_k = f_{\theta_{i-1}}(\mu_{k-1})$
  - 7:     Compute predictive covariance of  $x_k$ :
  - 8:      $\bar{P}_k = F_k P_{k-1} F_k^\top + Q_k$ , with  $F_k = \frac{\partial f(x)}{\partial x} \Big|_{x=\bar{\mu}_{k-1}}$
  - 9:     Update dictionary mean  $C_k$ :
  - 10:     $C_k = C_{k-1} + \frac{(z_k - M_k C_{k-1} \bar{\mu}_k) \bar{\mu}_k^\top V_{k-1}^\top}{\bar{\mu}_k^\top V_{k-1} \bar{\mu}_k + \eta_k}$
  - 11:     Update dictionary covariance  $V_k$ :
  - 12:      $V_k \approx V_k \otimes I_d$ , where
  - 13:      $V_k = V_{k-1} - \frac{V_{k-1} \bar{\mu}_k \bar{\mu}_k^\top V_{k-1}}{\bar{\mu}_k^\top V_{k-1} \bar{\mu}_k + \eta_k}$
  - 14:     Update coefficient mean  $\mu_k$ :
  - 15:      $\mu_k = \bar{\mu}_k + \bar{P}_k C_{k-1}^\top M_k S_k^{-1} (z_k - M_k C_{k-1} \bar{\mu}_k)$
  - 16:     where  $S_k = M_k C_{k-1} \bar{P}_k C_{k-1}^\top M_k^\top + M_k R_k M_k^\top$
  - 17:     Update coefficient covariance  $P_k$ :
  - 18:      $P_k = \bar{P}_k - \bar{P}_k C_{k-1}^\top M_k^\top S_k^{-1} M_k C_{k-1} \bar{P}_k$
  - 19:     Update parameters:
  - 20:     Iterative:  $\theta_i = \theta_{i-1} + \gamma \sum_{k=1}^n \nabla \log \tilde{p}_\theta(z_k | z_{1:k-1})|_{\theta=\theta_{i-1}}$
  - 21:     Recursive:  $\theta_k = \theta_{k-1} + \gamma \nabla \log \tilde{p}_\theta(z_k | z_{1:k-1})|_{\theta=\theta_{k-1}}$
-

## 2.5. Introduction to 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. Standard ECG recordings are made on paper at a rate of 25 mm per second, with a vertical calibration of 1 millivolt per centimeter. ([Klabunde \(2012 - 2012\)](#))

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.

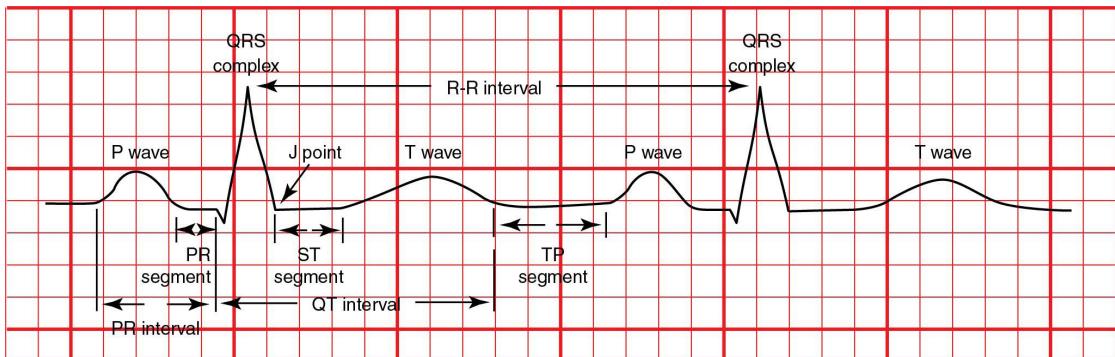


Figure 2.1.: Components of the ECG trace. ([Wesley and Huszar \(2021\)](#))

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:

- **Bipolar Limb Leads:** (Figure 3.1)
  - **Lead I:** Measures the voltage difference between the right arm and left arm.
  - **Lead II:** Measures the voltage difference between the right arm and left leg.
  - **Lead III:** Measures the voltage difference between the left arm and left leg.
- **Augmented Unipolar Limb Leads:** (Figure 3.1)
  - **aVR:** Measures the electrical difference from the right arm to a central point between the left arm and left leg.
  - **aVL:** Measures from the left arm to a central point.
  - **aVF:** Measures from the left leg to a central point.

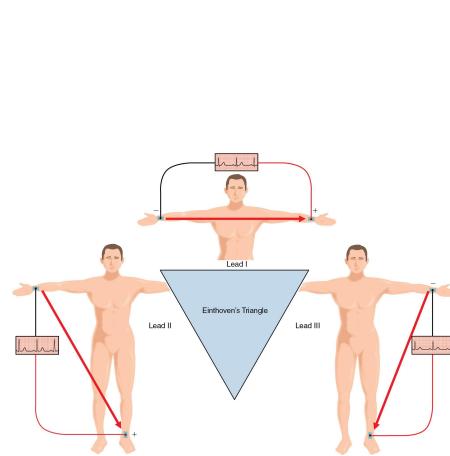


Figure 2.2.: Bipolar limb leads.  
(Wesley and Huszar (2021))

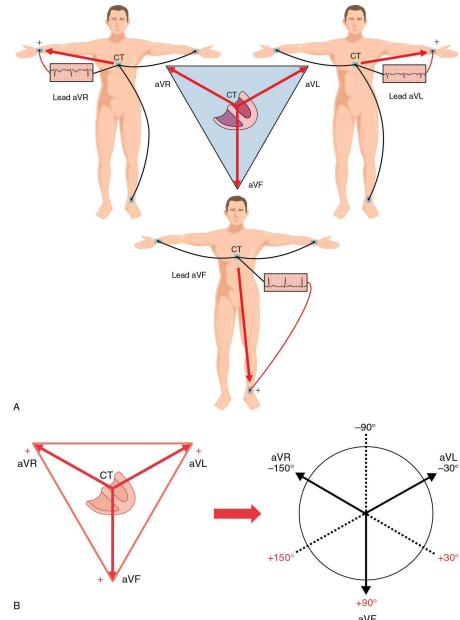


Figure 2.3.: Augmented unipolar limb leads. (Wesley and Huszar (2021))

### Precordial Unipolar Leads: (Figure 2.4)

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

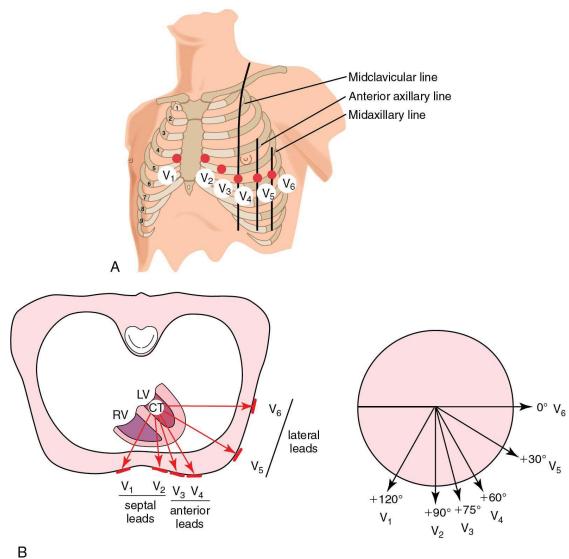


Figure 2.4.: Precordial unipolar leads. (Wesley and Huszar (2021))

The so called **Einthoven's Triangle** (can be seen in Figure 3.1) 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:

$$\text{Lead I} + \text{Lead III} = \text{Lead II}. \quad (2.46)$$

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.

## 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.

### 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. 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 heart beat (PQRST complex) is of approximately 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 the PSMF paper (Ömer Deniz 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).

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 Ömer Deniz Akyildiz et al. (2021), for TMF we set the weight matrix to identity, for PSMF and MLE-SMF we set  $R_k := R = \rho \otimes I_r$ , 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 initializations and randomly generated patterns of missing data.

In Table 3.1 we can see the results for  $r = 3$ . We see that PSMF and rPSMF have lower RMSEs compared to MLE-SMF and TMF.

$r = 3$						
	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	<b>98.51</b> (26.56)	<b>109.04</b> (29.08)	<b>124.91</b> (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	<b>122.79</b> (38.89)	<b>154.79</b> (45.69)	<b>186.31</b> (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 also measure the proportion of missing values that lie within a  $2\sigma$  coverage interval of the approximate posterior distribution. Table 3 shows how our method improves over the uncertainty quantification of MLE-SMF (the other methods do not provide a posterior distribution). This illustrates the added value of the matrix-variate prior on  $C$ , as well as our inference scheme. Note that rPSMF obtains a higher coverage percentage than PSMF on three of the datasets, which is due to the sequential updating of the noise covariances.

$r = 3$				$r = 10$			
Missing %:	20%	30%	40%	Missing %:	20%	30%	40%
PSMF	0.14	0.12	0.10	PSMF	0.24	0.18	0.14
rPSMF	<b>0.80</b>	<b>0.75</b>	<b>0.66</b>	rPSMF	<b>0.60</b>	<b>0.50</b>	<b>0.37</b>
MLE-SMF	0.07	0.06	0.06	MLE-SMF	0.07	0.06	0.05

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

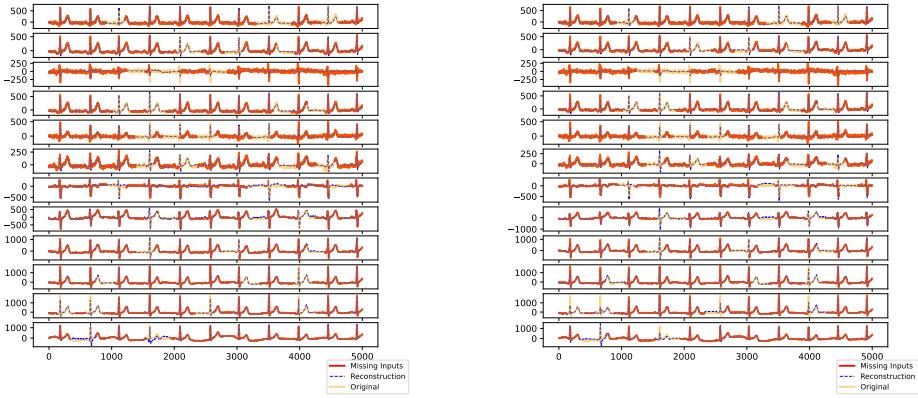


Figure 3.1.: Reconstruction with rank 3 of 20% missing data. Left: PSMF; Right: rPSMF.

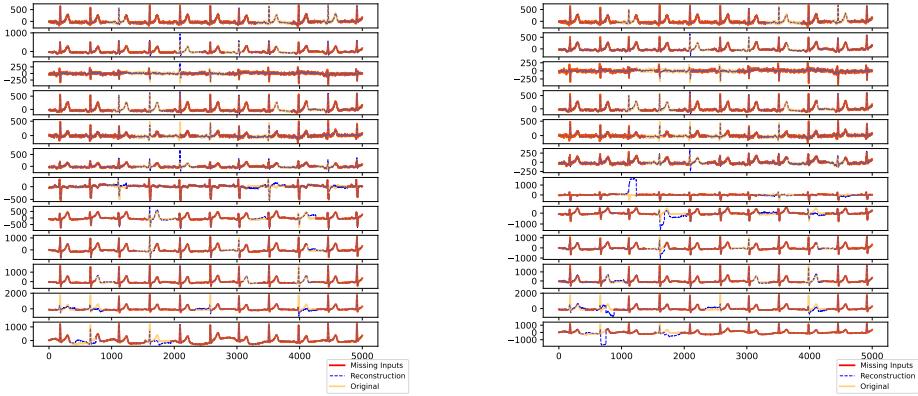


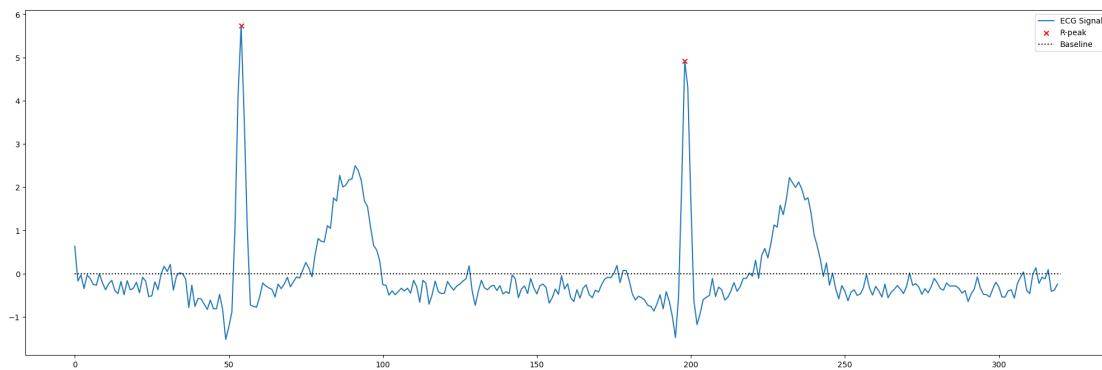
Figure 3.2.: Reconstruction with rank 10, 20% coverage. Left: PSMF; Right: rPSMF.

TODO: add more results. here or in the appendix/supplement?

### 3.2. R-peaks Detection

TODO:

- intro: why it is an important task, mention current methods?
- introduce the idea: smoother peaks, remove reconstruction from the original data
- threshold
- (potential) issues, improvements, results
- figure caption, more result figures



### 3.3. Forecasting

TODO:

- original data (5000) vs subsample (1500)
- smoothing vs no smoothing
- Fourier basis and number of terms, higher rank
- encountered issues, conclusions
- add figures for the basis, loss curves, forecast (show different experiments results)

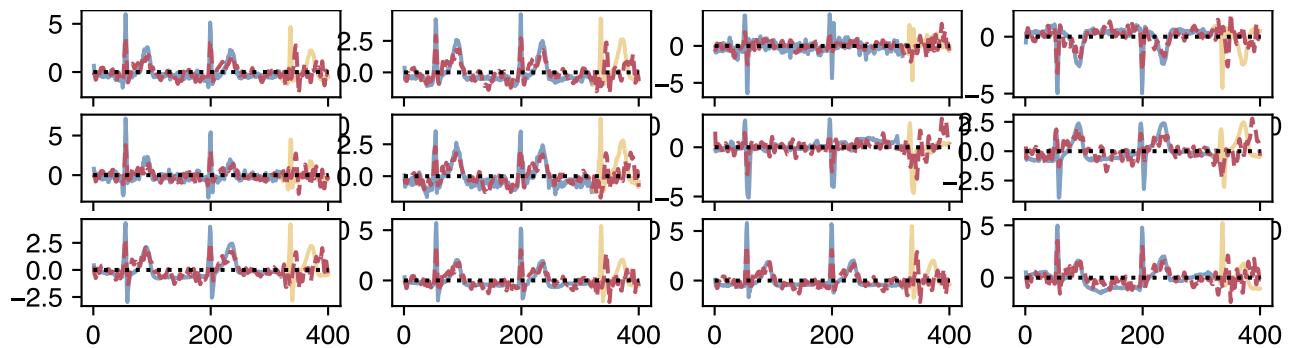


Figure 3.3.: My caption here

## 4. Conclusion

TODO: summarise results

### 4.1. Future work

TODO:

- More complex model to better suit the complexity of the ECG data
- Better computational efficiency

Why modelling ecg is useful? Specifically normal sinus rhythm.

Why detecting ECG is useful?

# 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. $r = 3$

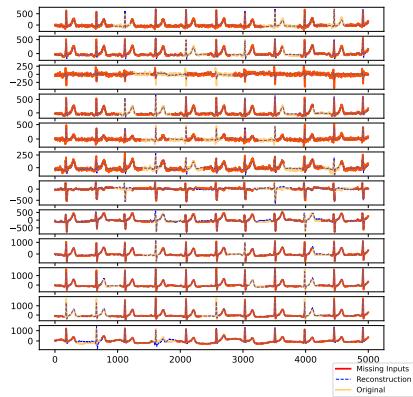


Figure A.1.: 20% missing data.

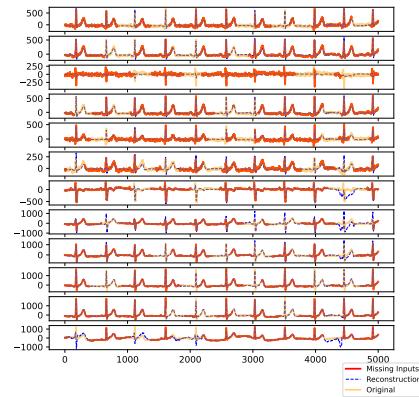


Figure A.2.: 30% missing data.

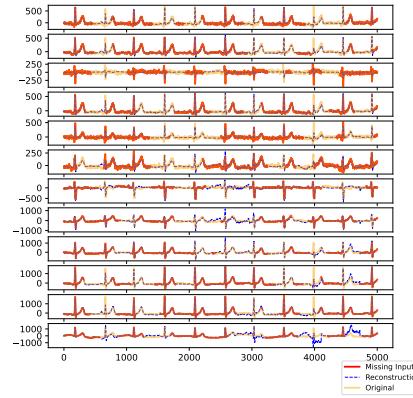


Figure A.3.: 40% missing data.

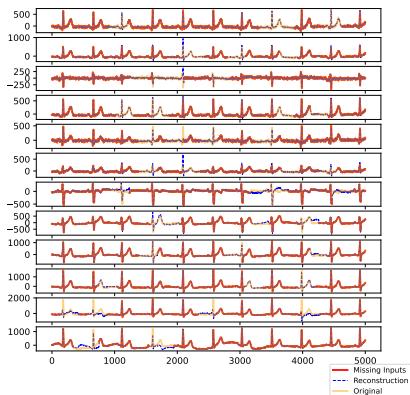
**A.1.2.**  $r = 10$ 

Figure A.4.: 20% missing data.

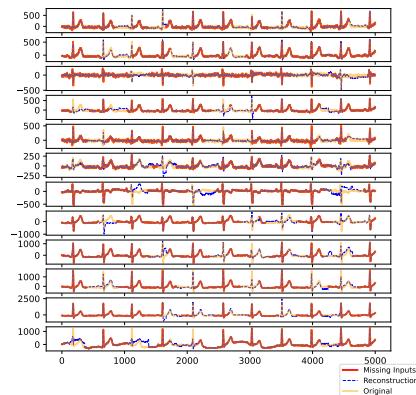


Figure A.5.: 30% missing data.

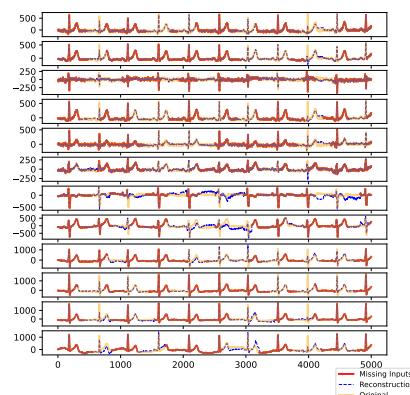


Figure A.6.: 40% missing data.

## A.2. rPSMF

### A.2.1. $r = 3$

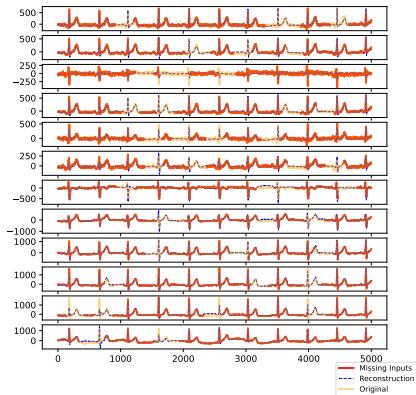


Figure A.7.: 20% missing data.

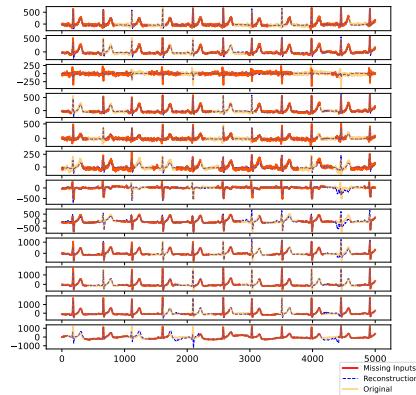


Figure A.8.: 30% missing data.

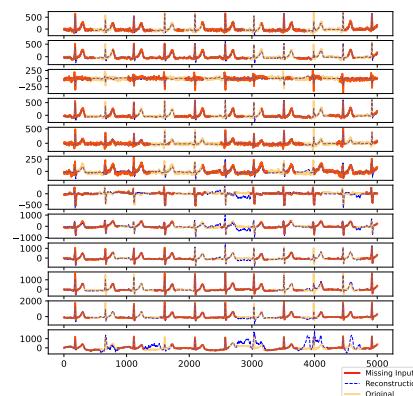


Figure A.9.: 40% missing data.

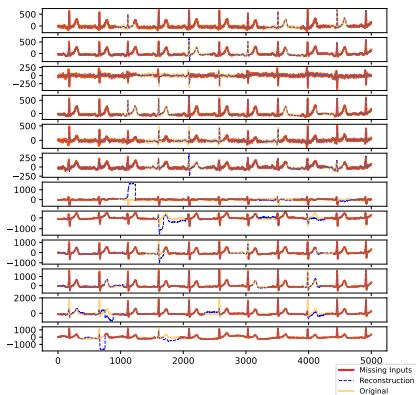
**A.2.2.  $r = 10$** 

Figure A.10.: 20% missing data.

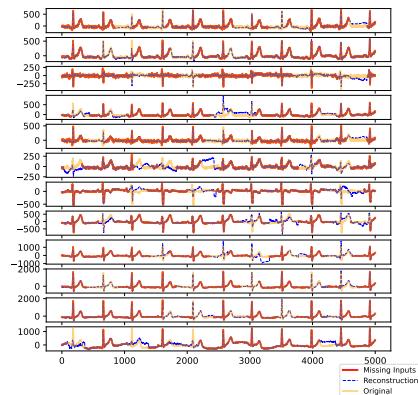


Figure A.11.: 30% missing data.

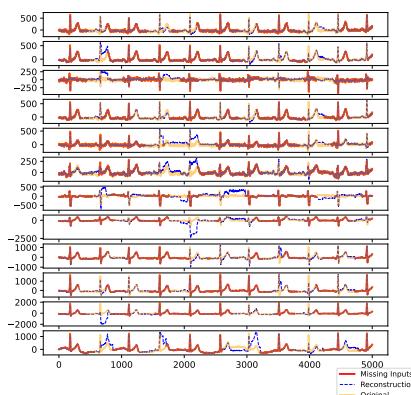


Figure A.12.: 40% missing data.

### A.3. MLE-SMF

#### A.3.1. $r = 3$

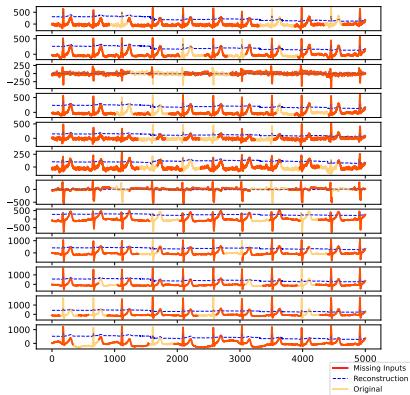


Figure A.13.: 20% missing data.

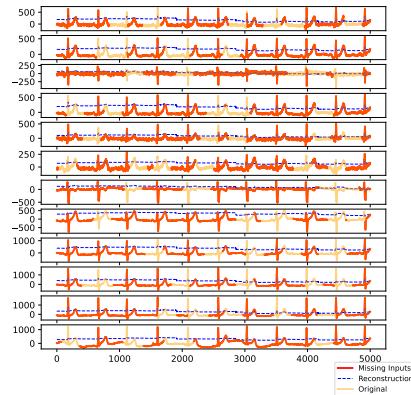


Figure A.14.: 30% missing data.

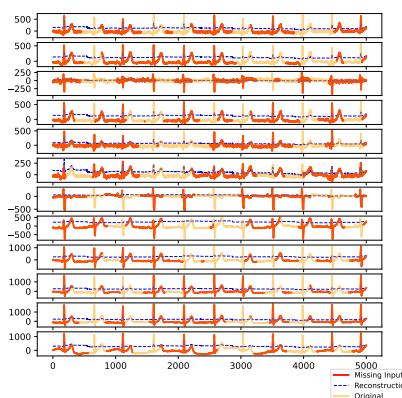


Figure A.15.: 40% missing data.

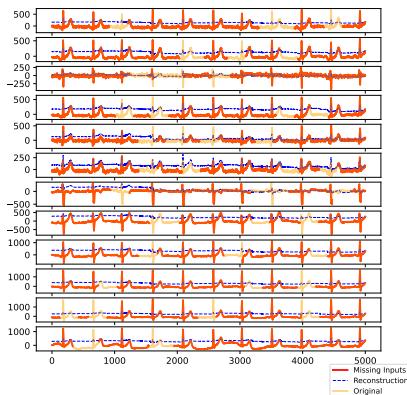
**A.3.2.**  $r = 10$ 

Figure A.16.: 20% missing data.

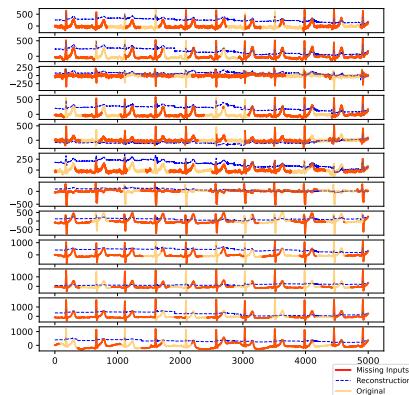


Figure A.17.: 30% missing data.

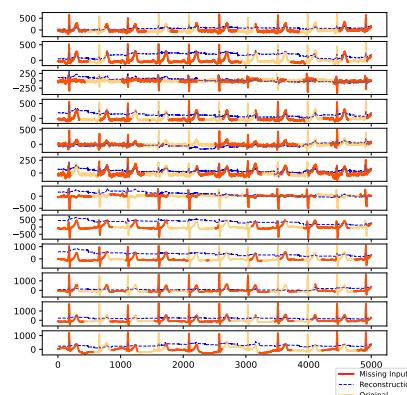


Figure A.18.: 40% missing data.

## A.4. TMF

### A.4.1. $r = 3$

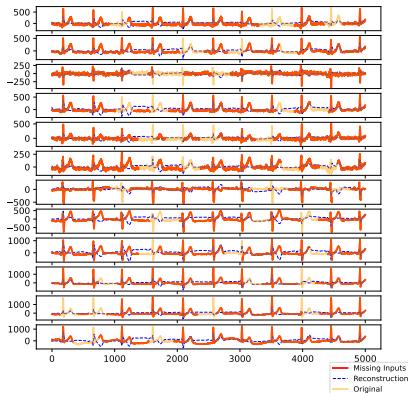


Figure A.19.: 20% missing data.

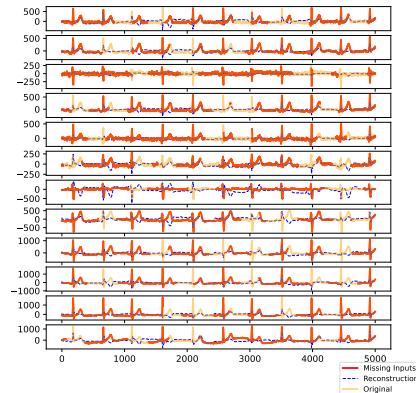


Figure A.20.: 30% missing data.

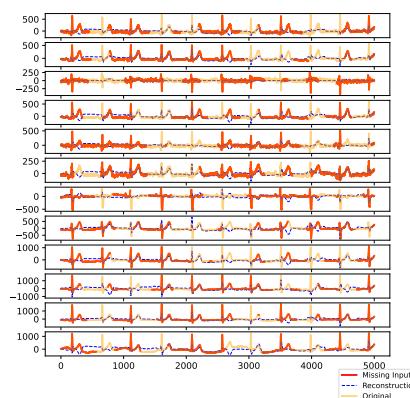


Figure A.21.: 40% missing data.

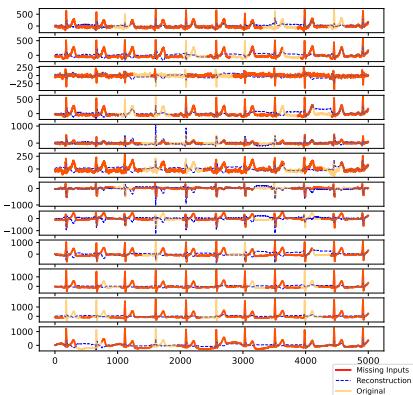
**A.4.2.**  $r = 10$ 

Figure A.22.: 20% missing data.

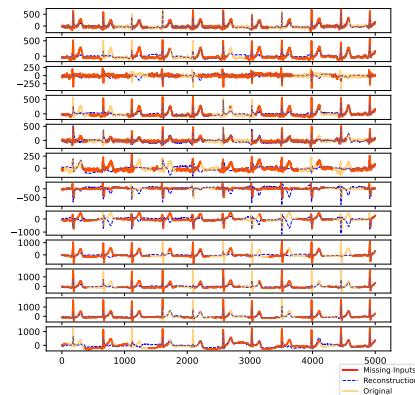


Figure A.23.: 30% missing data.

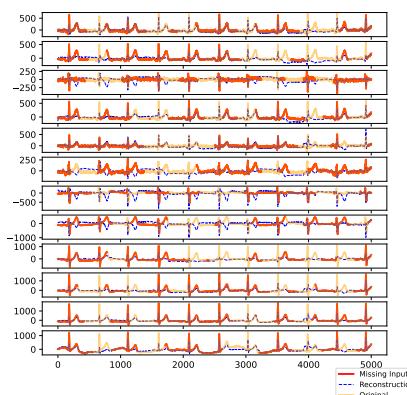


Figure A.24.: 40% missing data.

# 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.
- 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=1oOoAQAAQAAJ>.
- 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](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.
- Ömer Deniz Akyildiz, Gerrit J. J. van den Burg, Theodoros Damoulas, and Mark F. J. Steel. Probabilistic sequential matrix factorization, 2021.