Diffusion-based Nonlinear Filtering for Multimodal Data Fusion

Ori Katz

Diffusion-based Nonlinear Filtering for Multimodal Data Fusion

Research Thesis

As Partial Fulfillment of the Requirements for
the Degree Master of Science in Electrical Engineering

Ori Katz

Submitted to the Senate of the Technion—Israel Institute of Technology
Iyar 5777 Haifa May 2017

Contents

Abstract

The problem of information fusion from multiple data-sets acquired by multimodal sensors has drawn significant research attention over the years. In this work we focus on a particular problem setting consisting of a physical phenomenon or a system of interest observed by multiple sensors. We assume that all sensors measure some aspects of the system of interest with additional sensor-specific and irrelevant components. Our goal is to recover the variables relevant to the observed system and to filter out the nuisance effects of the sensor-specific variables. We present an approach based on manifold learning, which is particularly suitable for problems with multiple modalities, since it aims to capture the intrinsic structure of the data and relies on minimal prior model knowledge. Specifically, we propose a nonlinear filtering scheme, which extracts the hidden sources of variability captured by two or more sensors, that are independent of the sensor-specific components. In addition to presenting a theoretical analysis, we demonstrate our technique on real measured data for the purpose of sleep stage assessment based on multiple, multimodal sensor measurements. We show that without prior knowledge on the different modalities and on the measured system, our method gives rise to a data-driven representation that is well correlated with the underlying sleep process and is robust to noise and sensor-specific effects. Another related topic that was motivated by the challenge of processing multiple modalities and is a fundamental element in any manifold learning technique is the ability to reveal the similarities between data points. In a series of recent studies, this was accomplished by the Mahalanhobis distance. Yet, the computation of the Mahalanhobis distance from data requires an estimation of the covariance matrix, which is challenging, especially when it is applied to high-dimensional data sampled from multi-scale stochastic dynamical systems. Here, we further examine the computation of the Mahalanobis distance. Specifically we address on the inherent trade off between preserving locality and minimizing the sample-variance error. We demonstrate the influence of the estimation on various aspects related to manifold learning and analyze the incurred errors. In addition, we present a new covariance matrix estimation method. Finally, we show the application of the proposed method to simulated data arising from a multiscale stochastic dynamical system and demonstrate its advantage.

Acknowledgment

The Research Thesis Was Done Under The Supervision of Professor Ronen Talmon in the Department of Electrical Engineering.

Notations

K	number of common hidden variables
$\boldsymbol{\theta}^{(k)}$	kth common hidden variable
d_k	dimension of $\boldsymbol{\theta}^{(k)}$
Θ	set of all common hidden variables
M	number of observable variables
$\boldsymbol{s}^{(m)}$	mth observable variable
D_m	dimension of $\boldsymbol{s}^{(m)}$
$\mathbf{S}$	sensitivity table
$h_m(\cdot)$	a bilipschitz mth observation function
$\boldsymbol{n}^{(m)}$	mth sensor-specific hidden (nuisance) variables
p_m	dimension of $\boldsymbol{n}^{(m)}$
$\Theta^{(m)}$	subset of Θ sensed by $\boldsymbol{s}^{(m)}$
$\mathcal{S}^{(m)}$	subset of all hidden variables measured by $\boldsymbol{s}^{(m)}$
$d^{(m)\bigcap(n)}$	alternating diffusion distances matrix on the common manifold of the mth and the nth observable variables
$\mathbf{K}^{(m)\bigcap(n)}$	alternating diffusion kernel matrix on the common manifold of the mth and the nth observable variables
$\boldsymbol{\phi}_0^{(m)\bigcap(n)}$	stationary distribution of $\mathbf{K}^{(m)\bigcap(n)}$
$d^{(\cup)}$	common diffusion distance
$K^{(\cup)}$	common diffusion kernel
$\boldsymbol{r}_i^{(j)}$	column stack of the ith frame captured by jth camera
$\mathbf{B}$	random projection matrix
$\left\{\lambda_l^{(\cdot)}\right\}_{l=0}^{N-1}$	set of eigenvalues of the row stochastic Markov matrix $K^{(\cdot)}$
$\left\{\psi_l^{(\cdot)}\right\}_{l=0}^{N-1}$	set of eigenvectors of the row stochastic Markov matrix $K^{(\cdot)}$
$n(t)$	artificial noise sensor
$\|\cdot\|_M^2$	Mahalanobis distance
$\boldsymbol{x}_t$	process's trajectory in the intrinsic space
$\boldsymbol{z}_t$	process's trajectory in the rescaled space
f	non-linear measurement function
$\boldsymbol{y}_t$	process's trajectory in the observable space
$\boldsymbol{dy_{t_i}}$	increments of the diffusion process in the observable space
$\mathbf{J}$	Jacobian matrix of f
ω_t	standard Brownian motion
$a(\cdot)$	drift function

$b(\cdot)$	diffusion coefficient
$\mathbf{C}(\boldsymbol{y_t})$	covariance matrix of the process in observable space at $\boldsymbol{y}_t$
$\widehat{\mathbf{C}}(\boldsymbol{y_t})$	estimation of the covariance matrix of the process in observable space at $\boldsymbol{y}_t$
$\mu_{\boldsymbol{dy_{t_i}}}$	empiric mean of the diffusion process's increments in the time index t_i
N	window-length for covariance matrix estimation in the time-domain
$\mathcal{B}_R(\boldsymbol{y_0})$	a set containing the data-points that resides within a d-dimensional ball centered in $\boldsymbol{y_0}$ with a radius R
R^*	the median value of the euclidean distances between the data-points in $\mathcal{B}_R(\boldsymbol{y_0})$ from $\boldsymbol{y_0}$
$\widehat{C_e}(\boldsymbol{y_0})$	estimation of the covariance matrix based on the estimation set
$\widehat{C_v}(\boldsymbol{y_0})$	estimation of the covariance matrix based on the validation set
$\mathbf{Q}$	rescale matrix based on EVD of the $\widehat{C_e}$
R^{opt}	the radius that obtains minimal distortion measure
$E_M(\cdot,\cdot)$	an error incurred by using Mahalanobis distance to approximate the true distance
$E_C(\cdot)$	covariance matrix estimation error

Abbreviations

SNR	Signal to Noise Ratio
SVM	Support Vector Machines
EVD	Eigen Value Decomposition
DM	Diffusion Map
AD	Alternating Diffusion Map
RP	Random Projection
REM	Rapid Eye Movement
NREM	Non Rapid Eye Movement
N1	First shallow sleep stage in NREM
N2	Second shallow sleep stage in NREM
N3	Deep sleep stage in NREM
CORR	Correlation
NMSE	Normalized Mean Square Error

List of Figures

List of papers

- O. Katz, R. Talmon, L. Yu-Lun and W. Hau-Tieng, "Diffusion-based nonlinear filtering for multimodal data fusion with application to sleep stage assessment", submitted to Information Fusion.

1. Introduction

1.1. Motivation and overview

Often, when measuring a phenomenon of interest that arises from a complex dynamical system, a single data acquisition method is not capable of capturing its entire complexity and characteristics, and it is usually prone to noise and interferences. Recently, due to technological advances, the use of multiple types of measurement instruments and sensors has become more and more popular; nowadays, such equipment is smaller, less expensive, and can be mounted on every-day products and devices more easily. In contrast to a single sensor, multimodal sensors may capture complementary aspects and features of the measured phenomenon, and may enable us to extract a more reliable and detailed description of the measured phenomenon.

The availability of multimodal data calls for the development of analysis and processing tools, which appropriately combine data from the different sensors and handle well the inherent challenges that arise. One particular challenge is related to the heterogeneity of the data acquired in the different modalities; datasets acquired from different sensors may comprise different sources of variability, where only few are relevant to the phenomenon of interest. This particular challenge as well as many others have been the subject of many studies. For a recent comprehensive reviews, see [1, 2, 3].

In this thesis we focus on a particular multimodal setting in which a physical phenomenon of interest is measured by multiple sensors. As a result, each sensor consists of several sources of variability; some are related to the phenomenon of interest, possibly capturing its various aspects, whereas other sources of variability are sensor-specific and irrelevant. We present an approach based on manifold learning, which is a class of nonlinear data-driven methods, e.g. [4, 5, 6, 7], and specifically, we use the framework of diffusion maps (DM) [8]. On the one hand, manifold learning is particularly suitable for problems with multiple modalities since it aims to capture the intrinsic geometric structure of the underlying data and relies on minimal prior model knowledge. This enables to handle multimodal data in a systematic manner, without the need to specially tailor a solution for each modality. On the other hand, applying manifold learning to data acquired in multiple (multimodal) sensors may capture undesired/nuisance geometric structures as well. Recently, several manifold learning techniques for multimodal data have been proposed [9, 10, 11, 12]. In [9], the authors suggest to concatenate the samples acquired by different sensors into unified vectors. However this approach is sensitive to the scaling of each dataset, which might be especially diverse among datasets acquired by different modalities. To alleviate this problem, it is proposed in [10] to use DM to obtain a "standardized" representation of each dataset separately, and then to concate-

nate these "standardized" representations ans proceed as in [9]. Although the ability to handle multimodal data is improved, this concatenation scheme does not utilize the mutual relations and co-dependencies that might exist between the datasets.

While methods such as [9, 10, 12, 13] take into account all the measured information, the methods in [14, 15] use DM to implement nonlinear filtering. Specifically, following a recent line of study in which multiple kernels are constructed and combined [16, 17, 18, 19, 20], in [14, 15], it was shown that a method based on alternating applications of diffusion operators extracts only the common source of variability among the sensors, while filtering out the sensor-specific components. The shortcoming of this method arises when having a large number of sensors; often, sensors that measure the same system capture different information and aspects of that system. As a result, the common source of variability among all the sensors captures only a partial or empty look of the system, and important relevant information may be undesirably filtered out.

Here, we address the trade off between these two approaches. That is, we aim to maintain the relevant information captured by multiple sensors, while filtering out the nuisance components. Since the relevance of the various components is unknown, our main assumption is that the sources of variability which are measured only in a single sensor, i.e., sensor-specific, are nuisance. Conversely, we assume that components measured in two or more sensors are of interest. Importantly, such an approach implements implicitly sensor selection; "bad" sensors that are, for example malfunctioned and measure only nuisance information, are automatically filtered out. These assumptions stem from the fact that the phenomenon of interest is global and not specific to one sensor. We propose a nonlinear filtering scheme, in which only the sensor-specific sources of variability are filtered out while the sources of variability captured by two or more sensors are preserved. Based on prior theoretical results [14, 15], we show that our scheme indeed accomplishes this task. We illustrate the main features of our method on a toy problem. In addition, we demonstrate its performance on real measured data in an application for sleep stage assessment based on multiple, multimodal sensor measurements. Sleep is a global phenomenon with systematic physiological dynamics that represent a recurring non-stationary state of mind and body. Sleep evolves in time and embodies interactions between different subsystems, not solely limited in the brain. Thus, in addition to the well-known patterns in electroencephalogram (EEG) signals, its complicated dynamics are manifested in other sensors such as sensors measuring breathing patterns, muscle tones and muscular activity, eyeball movements, etc. Each one of the sensors is characterized by different structures and affected by numerous nuisance processes as well. In other words, while we could extract the sleep dynamics by analyzing different sensors, each sensor captures only part of the entire sleep process, and it introduces modality artifacts, noise, and interferences. We show that our scheme allows for an accurate systematic sleep stage identification based on multiple EEG recordings as well as multimodal respiration measurements. In addition, we demonstrate its capability to perform sensor selection by artificially adding simulated malfunctions sensors.

While working on the above-mentioned problem, we have identified a gap in manifold learning. A fundamental element in any manifold learning technique is having an ability to reveal the similarity between data points. This is usually accomplished via the

construction of a distance metric, which should be robust to noise and recover the underlying structure of the data. A common choice for such a metric is the Mahalanhobis distance [21, 22, 23, 24]. However, the computation of the Mahalanhobis distance from data requires an estimation of the covariance matrix. When we have temporal sensor measurements, one way to estimate the covariance is to compute the sample covariance in a finite window length. This approach suffers from many limitations, especially when it is applied to high-dimensional data sampled from multi-scale stochastic dynamical systems [25].Here, we further examine the computation of the Mahalanobis distance. Specifically we focus on the inherent trade off between preserving locality and minimizing the sample-variance error when estimating the covariance matrix. We demonstrate the influence of the window-length on various aspects related to manifold learning and analyze the errors incurred due to the chosen window-length. We propose a new method for estimating the covariance matrix that preserves the locality while trying to minimize the sample-variance error. We apply the proposed algorithm to simulation data arises from a multiscale stochastic dynamical system, which was studied in [25], and demonstrate its advantage.

1.2. Thesis structure

The remainder of the thesis is organized as follows. In chapter 2 we provide a short theoretical background on DM and on alternating diffusion (AD), which are manifold learning techniques that will be further utilized here. In Chapter 3 we present a formulation of the common source extraction problem. We present an illustrative toy problem, and detailed description and interpretation of the proposed scheme are presented. We also demonstrate the capabilities of the proposed scheme on the presented toy problem. In chapter 4 we introduce the problem of sleep stage assessment based on multiple, multimodal sensor measurements and apply the scheme proposed in Chapter 3 to this problem. Chapter 5 deals with covariance estimation for manifold learning. We formulate the problem, demonstrate and exemplify the difficulties of covariance estimation in multi-scale stochastic dynamical systems. We propose a method for estimating the covariance matrix and exemplify its performance on the simulated problem investigated in [25]. In Chapter 6 we conclude with final remarks and discuss further research directions.

2. Related Work and Theoretical Background

2.1. Diffusion maps

DM is a non-linear data-driven dimensionality reduction method [8]. Assume we have N high-dimensional data-points $\{\boldsymbol{s}_i\}_{i=1}^N$. The DM method begins with the calculation of a pairwise affinity matrix based on a local kernel, often using some metric within a gaussian kernel, i.e.,

$$W_{i,j} = \exp\left(-\frac{\|\boldsymbol{s}_i - \boldsymbol{s}_j\|_M^2}{\varepsilon}\right), \tag{2.1}$$

where $\varepsilon > 0$ is a tuneable kernel scale and $\|\cdot\|_M$ is a metric. The choice of the metric $\|\cdot\|_M$ depends on the application; common choices are the Euclidean and the Mahalanobis distances [8, 26, 22, 23, 24]. This construction implicitly defines a weighted graph, where the data samples $\{\boldsymbol{s}_i\}_{i=1}^N$ are the nodes of the graph, and $W_{i,j}$ is the weight of the edge connecting node $\boldsymbol{s}_i$ and node $\boldsymbol{s}_j$. The next step is to normalize the affinity matrix and then to build the diffusion operator $\mathbf{K} \in \mathbb{R}^{N\times N}$, e.g., by:

$$Q_{i,i} = \left(\sum_{l=1}^{N} W_{i,l}\right)^{-1}; \mathbf{K} = \mathbf{QW}, \tag{2.2}$$

where $\mathbf{Q}$ is a diagonal matrix used for normalization, such that in this case $\mathbf{K}$ is row-stochastic. Hence, $\mathbf{K}$ can be viewed as the transition matrix of a Markov chain defined on the graph. Accordingly, for $t > 0$, $\mathbf{K}^t$ is the transition probability matrix of t consecutive steps, and $(K^t)_{i,j}$ is the probability to jump from node $\boldsymbol{s}_i$ to node $\boldsymbol{s}_j$ in t steps. Let $d_t(i,j)$ be the diffusion distance [8] between the ith and the jth data samples, i.e. d_t is defined by

$$d_t(i,j) = \sqrt{\sum_{l=1,\ldots,N} \frac{\left((K^t)_{i,l} - (K^t)_{j,l}\right)^2}{\phi_0(l)}} \tag{2.3}$$

where $\phi_0(\cdot)$ is the stationary distribution of the Markov chain. The diffusion distance has been shown to be a powerful metric for measuring geometrical similarities between data-points [8]. While the Euclidean distance compares two individual data-points and might be affected by distortions and noise, the diffusion distance demonstrates better robustness to noise since it relies on the connectivity between the two data-points using the entire data-set [8, 27].

The direct computation of the diffusion distance is cumbersome. An efficient calculation is attainable via the spectral decomposition of $\mathbf{K}$. Let $\{\lambda_l\}_{l=0}^{N-1}$ and $\{\boldsymbol{\psi}_l\}_{l=0}^{N-1}$ be the sets of eigenvalues and right eigenvectors of $\mathbf{K}$, where the eigenvalues are written in descending order. Based on the eigenvalue decomposition of $mathbfK$, we define a new representation (embedding) of the data-points:

$$\Psi_t(i) : \boldsymbol{s}_i \mapsto [\lambda_1^t\psi_1(i), \lambda_2^t\psi_2(i), \ldots, \lambda_{N-1}^t\psi_{N-1}(i)], \tag{2.4}$$

where $\psi_l(i)$ denotes the $i$the element of $\boldsymbol{\psi}_l$. The obtained embedding provides a new representation of the data, referred to as DM, in which the Euclidean distance between two embedded data-point is equal to the diffusion distance [8], i.e.:

$$d_t^2(i,j) = \|\Psi_t(i) - \Psi_t(j)\|^2 = \sum_{l\geq 1} \lambda_l^{2t}(\psi_l(i) - \psi_l(j))^2. \tag{2.5}$$

In order to achieve a compact representation in reduced dimensionality, DM is often redefined by keeping only the first L components (i.e., the L eigenvalues and eigenvectors corresponding to the largest L eigenvalues):

$$\Psi_t(i) : \boldsymbol{s}_i \mapsto [\lambda_1^t\psi_1(i), \lambda_2^t\psi_2(i), \ldots, \lambda_L^t\psi_L(i)], \tag{2.6}$$

where L is usually determined by the eigenvalues decay. For more details and full analysis of this technique see [8, 28]. The entire DM method is outlined in Algorithm 1.

The term "diffusion distance" in (2.3) suggests that $d_t(i,j)$ induces a reasonable notion of distance. For completeness, we elaborate on this point. Recall the definition of a distance.

Definition 1. *Let X be a set. A distance (or metric) on X is a function $d : X\times X \to \mathbb{R}_+$ such that for all $x, y, z \in X$:*

1. *$d(x,y) = 0$ if and only if $x = y$,*
2. *$d(x,y) = d(y,x)$,*
3. *$d(x,z) \leq d(x,y) + d(y,z)$.*

The following proposition states that the "diffusion distance" is really a metric defined on the nodes of the graph.

Proposition 2. *If $\mathbf{K}$ is full rank, then d_t is a distance function.*

Since we could not find a proof in the literature, for the sake of self-containment, we provide a proof that summarizes the discussion in [29].

Proof. We prove that (1)-(3) hold. Define $\widetilde{\mathbf{K}}^t = \boldsymbol{\Phi}^{-1}\mathbf{K}^t$ where $\boldsymbol{\Phi}$ is a diagonal matrix such that $\Phi_{k,k} = \sqrt{\phi_0(k)}$. Since $\mathbf{K}^t$ is full rank and since $\boldsymbol{\Phi}$ is non-degenerate by the construction of the weighted graph, $\widetilde{\mathbf{K}}^t$ is full rank. Denote the ith row of $\widetilde{\mathbf{K}}^t$ as $\boldsymbol{v}_i$.

Accordingly, $d_t(i,j)$ can be expressed as the Euclidean distance between the ith and the jth rows of $\widetilde{\mathbf{K}}^t$:

$$d_t(i,j) = \sqrt{\sum_{l=1,\ldots,N} \left((\widetilde{\mathbf{K}}^t)_{i,l} - (\widetilde{\mathbf{K}}^t)_{j,l} \right)^2} = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_{\mathbb{R}^N} \tag{2.7}$$

The properties of the Euclidean distance in (2.7) imply that (2) and (3) hold. If $i = j$, then $d_t(i,j) = 0$. If $d_t(i,j) = 0$, then $\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2 = 0$, implying that $\boldsymbol{v}_i = \boldsymbol{v}_j$. Since $\widetilde{\mathbf{K}}^t$ is full rank, there are no identical columns. In other words, no two different samples $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ for $i \neq j$ have identical affinities to all other samples, i.e., $\boldsymbol{v}_i \neq \boldsymbol{v}_j$. Therefore, if $\boldsymbol{v}_i = \boldsymbol{v}_j$, then $i = j$. □

2.2. Alternating diffusion

Consider a system driven by one hidden variable $\boldsymbol{\theta}$ which is measured by 2 observable variables $\boldsymbol{s}^{(1)}$ and $\boldsymbol{s}^{(2)}$. The AD algorithm, outlined in Algorithm 2, builds from the observations an AD operator that is equivalent to a simple diffusion operator (as described in Section 2.1) that would have been computed if we had a direct access to samples of the common hidden variables. This operator enables to capture only the structure of the common variables while ignoring the nuisance (sensor-specific) variables. For more details and full analysis of this algorithm see [14, 15]; here, we only bring a brief review of the method and the construction of the AD operator.

Assume we have N aligned samples (realizations) from 2 observable variables: $\left\{ (\boldsymbol{s}_i^{(1)}, \boldsymbol{s}_i^{(2)}) \right\}_{i=1}^{N}$. For each observation we build an affinity matrix $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ as follows:

$$W_{i,j}^{(1)} = \exp\left(-\frac{\|\boldsymbol{s}_i^{(1)} - \boldsymbol{s}_j^{(1)}\|_{M_1}^2}{\varepsilon^{(1)}} \right); W_{i,j}^{(2)} = \exp\left(-\frac{\|\boldsymbol{s}_i^{(2)} - \boldsymbol{s}_j^{(2)}\|_{M_2}^2}{\varepsilon^{(2)}} \right) \tag{2.8}$$

for all $i,j = 1,\ldots,N$, where $\varepsilon^{(1)}$ and $\varepsilon^{(2)}$ are the tuneable kernel scales and $\|\cdot\|_{M_1}$ and $\|\cdot\|_{M_2}$ are the chosen metrics for each set of observations. Based on the affinity matrix, we calculate the diffusion operators $\mathbf{K}^{(1)}$ and $\mathbf{K}^{(2)}$ according to:

$$\begin{aligned} Q_{i,i}^{(1)} = \left(\sum_{l=1}^{N} W_{i,l}^{(1)} \right)^{-1} \quad &; \quad Q_{i,i}^{(2)} = \left(\sum_{l=1}^{N} W_{i,l}^{(2)} \right)^{-1} \\ \mathbf{K}^{(1)} = \mathbf{Q}^{(1)}\mathbf{W}^{(1)} \quad &; \quad \mathbf{K}^{(2)} = \mathbf{Q}^{(2)}\mathbf{W}^{(2)} \end{aligned}$$

where $\mathbf{Q}^{(1)}$ and $\mathbf{Q}^{(2)}$ are diagonal matrices used for normalization. Next, we define $\mathbf{K}^{(1)\bigcap(2)} = \mathbf{K}^{(1)}\mathbf{K}^{(2)}$ as the AD operator. Note that $\mathbf{K}^{(1)\bigcap(2)}$ is row-stochastic, and hence, can be considered as a transition probability matrix of a new Markov chain that alternates between the two data sets. Namely, each step of this alternating process consists of a propagation step using $\mathbf{K}^{(1)}$ followed by a propagation step using $\mathbf{K}^{(2)}$.

Broadly, in each propagation step, the Markov chain jumps with high probability to neighboring samples that are similar in terms of the kernel. Combining alternating steps

Algorithm 1 Diffusion Maps

Input: High-dimensional samples from an observable variables: $\{\boldsymbol{s}_i\}_{i=1}^N$.
Output: L dimensional representation of the data-set $\{\Psi_t(i)\}_{i=1}^N$ where $\Psi_t(i) \in \mathbb{R}^L$.

1. Calculate the affinity matrix $\mathbf{W}$:

$$W_{i,j} = \exp\left(-\frac{\|\boldsymbol{s}_i - \boldsymbol{s}_j\|_M^2}{\varepsilon}\right) \tag{2.10}$$

2. Compute the diffusion operator (transition matrix) $\mathbf{K}$:

$$Q_{i,i} = \left(\sum_{l=1}^{N} W_{i,l}\right)^{-1}; \mathbf{K} = \mathbf{QW}, \tag{2.11}$$

3. Calculate the spectral decomposition of $\mathbf{K}$ and obtain its eigenvalues $\{\lambda_l\}_{l=0}^{N-1}$ and eigenvectors $\{\psi_l\}_{l=0}^{N-1}$.

4. Define a new embedding for the data-points:

$$\Psi_t(i) : \boldsymbol{s}_i \mapsto [\lambda_1^t\psi_1(i), \lambda_2^t\psi_2(i), \ldots, \lambda_L^t\psi_L(i)] \tag{2.12}$$

where $t > 0$ is a selected number of steps and $\psi_l(i)$ denotes the ith element of ψ_l.

results in consecutive jumps according to similarities in the first set and then according to similarities in the second set. Overall, only similarities in terms of the common components among the two views are maintained.

Formally, we define the diffusion distance between the ith and the jth sample based on the AD operator as the following Euclidean distance

$$d_t^{(1)\bigcap(2)}(i,j) = \sqrt{\sum_{l=1,\ldots,N} \frac{\left(\left((K^{(1)\bigcap(2)})^t\right)_{i,l} - \left((K^{(1)\bigcap(2)})^t\right)_{j,l}\right)^2}{\phi_0^{(1)\bigcap(2)}(l)}} \tag{2.9}$$

where $\boldsymbol{\phi}_0^{(1)\bigcap(2)}$ is the stationary distribution of $\mathbf{K}^{(1)\bigcap(2)}$ and $t > 0$ is the number of alternating steps. The following corollary is an immediate results of Proposition 2.

Corollary 3. *If* $\mathbf{K}^{(1)\bigcap(2)}$ *is full rank, then* $d_t^{(1)\bigcap(2)}$ *is a distance function.*

It can be shown that this distance is equivalent to the diffusion distance that would have been computed if we had a direct access to observable variables that see only the common variable [14].

Algorithm 2 Alternating Diffusion

Input: Aligned samples from 2 observable variables: $\left\{(\boldsymbol{s}_i^{(1)}, \boldsymbol{s}_i^{(2)})\right\}_{i=1}^{N}$.

Output: Diffusion distances $d_t^{(1)\bigcap(2)}$.

1. Calculate two pairwise affinity matrices $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ based on a gaussian kernel as follows:

$$W_{i,j}^{(1)} = \exp\left(-\frac{\|\boldsymbol{s}_i^{(1)} - \boldsymbol{s}_j^{(1)}\|_M^2}{\varepsilon^{(1)}}\right); W_{i,j}^{(2)} = \exp\left(-\frac{\|\boldsymbol{s}_i^{(2)} - \boldsymbol{s}_j^{(2)}\|_M^2}{\varepsilon^{(2)}}\right) \tag{2.13}$$

   for all $i, j = 1, \ldots, N$, where $\varepsilon^{(1)}$ and $\varepsilon^{(2)}$ are the kernel scales and $\|\cdot\|_M^2$ is the chosen metric.

2. Create two diffusion operators $\mathbf{K}^{(1)}$ and $\mathbf{K}^{(2)}$:

$$\begin{array}{ccc} Q_{i,i}^{(1)} = \left(\sum_{l=1}^{N} W_{i,l}^{(1)}\right)^{-1} & ; & Q_{i,i}^{(2)} = \left(\sum_{l=1}^{N} W_{i,l}^{(2)}\right)^{-1} \\ \mathbf{K}^{(1)} = \mathbf{Q}^{(1)}\mathbf{W}^{(1)} & ; & \mathbf{K}^{(2)} = \mathbf{Q}^{(2)}\mathbf{W}^{(2)} \end{array}$$

3. Build the alternating-diffusion kernel:

$$\mathbf{K}^{(1)\bigcap(2)} = \mathbf{K}^{(1)}\mathbf{K}^{(2)} \tag{2.14}$$

4. Compute the altenating-diffusion distance between each two points (i, j)

$$d_t^{(1)\bigcap(2)}(i,j) = \sqrt{\sum_{l=1,\ldots,N} \frac{\left(\left((K^{(1)\bigcap(2)})^t\right)_{i,l} - \left((K^{(1)\bigcap(2)})^t\right)_{j,l}\right)^2}{\phi_0^{(1)\bigcap(2)}(l)}} \tag{2.15}$$

   where $\boldsymbol{\phi}_0^{(1)\bigcap(2)}$ is the stationary distribution of $\mathbf{K}^{(1)\bigcap(2)}$ and $t > 0$ is a tuneable parameter.

3. Diffusion-based Nonlinear Filtering for Multimodal Data Fusion

The problem of information fusion from multiple data-sets acquired by multimodal sensors has drawn significant research attention over the years. In this chapter, we focus on a particular problem setting consisting of a physical phenomenon or a system of interest observed by multiple sensors. We assume that all sensors measure some aspects of the system of interest with additional sensor-specific and irrelevant components. Our goal is to recover the variables relevant to the observed system and to filter out the nuisance effects of the sensor-specific variables. We propose an approach based on manifold learning, which is particularly suitable for problems with multiple modalities, since it aims to capture the intrinsic structure of the data and relies on minimal prior model knowledge. Specifically, we propose a nonlinear filtering scheme, which extracts the hidden sources of variability captured by two or more sensors, that are independent of the sensor-specific components. In addition to presenting a theoretical analysis in Chapter 2 we will demonstrate our technique on real measured data for the purpose of sleep stage assessment based on multiple, multimodal sensor measurements. We will show that without prior knowledge on the different modalities and on the measured system, our method gives rise to a data-driven representation that is well correlated with the underlying sleep process and is robust to noise and sensor-specific effects.

The remainder of the chapter is organized as follows. In Section 3.1 we present a formulation for the common source extraction problem and present an illustrative toy problem. In Section 3.3 detailed description and interpretation of the proposed scheme are presented. In Section 3.4, we demonstrate the capabilities of the proposed scheme on the toy problem introduced in Section 3.1. In Chapter 4 we demonstrate the performances of the proposed scheme for application of sleep stage identification based on multimodal measured data recorded in a special sleep clinic.

3.1. Problem setting

Consider a system driven by a set of K hidden random (high-dimensional) variables $\Theta = \{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(K)}\}$, where $\boldsymbol{\theta}^{(k)} \in \mathbb{R}^{d_k}$. The system is measured by M observable variables $\boldsymbol{s}^{(m)},\ m = 1, \ldots, M$, where each sensor has access to only a partial view of the entire system and its driving variables Θ. To formulate it, we define a "sensitivity table" given by the binary matrix $\mathbf{S} \in \mathbb{Z}_2^{K \times M}$, indicating the variables sensed by each observable variable. Specifically, the (k, m)th element in $\mathbf{S}$ indicates whether the hidden variable $\boldsymbol{\theta}^{(k)}$ is measured by the observable variable $\boldsymbol{s}^{(m)} \in \mathbb{R}^{D_m}$. The observable variables are

Table 3.1.: List of important notation.

Nomenclature	
K	number of common hidden variables
$\boldsymbol{\theta}^{(k)}$	kth common hidden variable
d_k	dimension of $\boldsymbol{\theta}^{(k)}$
Θ	set of all common hidden variables
M	number of observable variables
$\boldsymbol{s}^{(m)}$	mth observable variable
D_m	dimension of $\boldsymbol{s}^{(m)}$
$\mathbf{S}$	sensitivity table
$h_m(\cdot)$	a bilipschitz mth observation function
$\boldsymbol{n}^{(m)}$	mth sensor-specific hidden (nuisance) variables
p_m	dimension of $\boldsymbol{n}^{(m)}$
$\Theta^{(m)}$	subset of Θ sensed by $\boldsymbol{s}^{(m)}$
$\mathcal{S}^{(m)}$	subset of all hidden variables measured by $\boldsymbol{s}^{(m)}$

therefore given by

$$\boldsymbol{s}^{(m)} = h_m(\Theta^{(m)}, \boldsymbol{n}^{(m)}) \tag{3.1}$$

where $h_m(\cdot)$ is a bilipschitz observation function, $\boldsymbol{n}^{(m)} \in \mathbb{R}^{p_m}$ are hidden random variables captured only by the mth observable variable, and $\Theta^{(m)}$ is the subset of driving hidden variables of interest sensed by $\boldsymbol{s}^{(m)}$, given by

$$\Theta^{(m)} = \left\{ \boldsymbol{\theta}^{(k)} | \forall k, S_{k,m} = 1 \right\} \subseteq \Theta, m = 1, \ldots, M \tag{3.2}$$

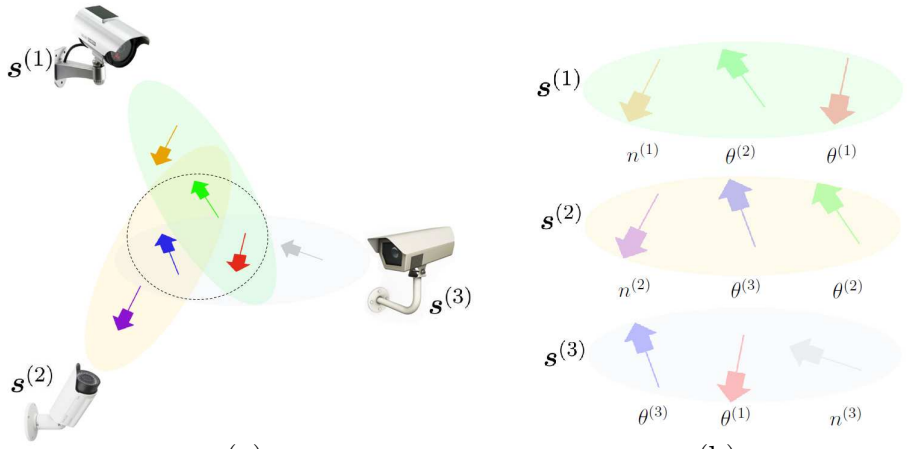
The random hidden variables $\boldsymbol{n}^{(m)}$ are *sensor-specific* (associated only with the mth observer). They are conditionally independent given the hidden variables of interest and will be assumed as noise/nuisance variables. We further assume that each random hidden variable in Θ is measured by at least two observable variables, such that $\sum_{m=1}^{M} S_{k,m} \geq 2$ for each $k = 1, \ldots, K$. As a result, we refer to the hidden variables $\boldsymbol{\theta}^{(k)}$ in Θ as *common variables*.

In order to simplify the notation, we denote the subset of all hidden variables (both common and sensor-specific) measured by the mth observable by $\mathcal{S}^{(m)} = \{\Theta^{(m)}, \boldsymbol{n}^{(m)}\}$. Furthermore, we assume that the dimensions of the observations and the hidden variables satisfy

$$D_m \geq \sum_{\boldsymbol{\theta}^{(k)} \in \Theta^{(m)}} (d_k + p_k), \ k = 1, 2, ..., M \tag{3.3}$$

i.e., the observations are in higher dimension than the hidden common and nuisance variables.

An observation of the system denoted as $(\boldsymbol{s}_i^{(1)}, \boldsymbol{s}_i^{(2)}, \ldots, \boldsymbol{s}_i^{(M)})$ is a realization of all the observables corresponding to all the sensors simultaneously. It is associated with a realization of the hidden variables $\Theta_i = (\boldsymbol{\theta}_i^{(1)}, \ldots, \boldsymbol{\theta}_i^{(K)})$ and realizations of the M hidden nuisance variables $(\boldsymbol{n}_i^{(1)}, \ldots, \boldsymbol{n}_i^{(M)})$. Given N observation samples $\left\{ (\boldsymbol{s}_i^{(1)}, \boldsymbol{s}_i^{(2)}, \ldots, \boldsymbol{s}_i^{(M)}) \right\}_{i=1}^{N}$,



(a) (b)

Figure 3.1.: Toy problem setup. (a) The coverage area of each camera, the system's range of interest is marked by the dashed circle. (b) Sample snapshot taken by each camera.

our goal is to obtain a parametrization for the underlying realizations of the common hidden random variables $\left\{(\boldsymbol{\theta}_i^{(1)}, \ldots, \boldsymbol{\theta}_i^{(K)})\right\}_{i=1}^{N}$ while filtering out the nuisance variables $\left\{(\boldsymbol{n}_i^{(1)}, \ldots, \boldsymbol{n}_i^{(M)})\right\}_{i=1}^{N}$. We note that the observations index i may represent the time index in case of time series.

3.2. Illustrative toy problem

We illustrate the problem setting using the following toy example. Consider six rotating arrows captured in simultaneous snapshots by three different cameras. We assume that each arrow rotates at different speed, and that each camera can capture only a partial image of the entire system. The partial view of each camera is depicted in Figure 3.1. Thus, overall, each camera captures a sequence of snapshots (a movie) of three rotating colored arrows. Further illustration of the entire system and of the captured images by each camera can be seen in the following link https://youtu.be/a-yb7ScdnnA.

In this problem setting, the hidden variables are the six rotation angles of the arrows: the common variables $\Theta = \{\theta^{(1)}, \theta^{(2)}, \theta^{(3)}\}$ are the three rotation angles of the centred arrows, which are marked by the dashed circle in Figure 3.1, and the nuisance variables $\{n^{(1)}, n^{(2)}, n^{(3)}\}$ are the three rotation angles of the peripheral arrows, since each is captured only by a single camera. It should be noted that none of the arrows is common to all of the cameras, meaning that the set of common components within the entire set of observables is empty.

In order to identify the hidden variables, we use different colors for the arrows. The arrows rotating according to the common variables $\Theta = \{\theta^{(1)}, \theta^{(2)}, \theta^{(3)}\}$ are colored in red, green and blue, respectively, and the arrows rotating according to the nuisance

variables $\{n^{(1)}, n^{(2)}, n^{(3)}\}$ are colored in orange, purple and gray, respectively. The hidden variables measured by each camera are $\mathcal{S}^{(1)} = \{\theta^{(1)}, \theta^{(2)}, n^{(1)}\}$, $\mathcal{S}^{(2)} = \{\theta^{(2)}, \theta^{(3)}, n^{(2)}\}$ and $\mathcal{S}^{(3)} = \{\theta^{(3)}, \theta^{(1)}, n^{(3)}\}$. Our goal is to obtain a parametrization of the rotation angles of the three common arrows $\Theta = \{\theta^{(1)}, \theta^{(2)}, \theta^{(3)}\}$ given the three movies from the cameras, without any prior knowledge on the system and the problem structure. In the sequel, we will use this toy problem for demonstrating important aspects and how our method accomplishes this task.

3.3. Nonlinear filtering scheme

AD provides us with an access to the common variables between a pair of observable variables. By using AD as a building block, we propose a generalization for a set of multiple (i.e., more than two) observable variables. Consider the system described in Section 3.1 with aligned samples from M observable variables: $\left\{(\boldsymbol{s}_i^{(1)}, \boldsymbol{s}_i^{(2)}, \ldots, \boldsymbol{s}_i^{(M)})\right\}_{i=1}^{N}$. The observable variables are driven by a set of K hidden random variables $\Theta = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(K)})$ and contaminated by a set of M nuisance sensor-specific variables $(\boldsymbol{n}^{(1)}, \ldots, \boldsymbol{n}^{(M)})$. Our goal is to obtain a parametrization of the common hidden random variables Θ from the observations.

More specifically, in the context of our problem, consider a pair of observable variables $\boldsymbol{s}^{(m)}$ and $\boldsymbol{s}^{(n)}$. Applying AD to $\boldsymbol{s}^{(m)}$ and $\boldsymbol{s}^{(n)}$ yields the common hidden variables measured by the two. Therefore, its operation can be written as

$$\mathcal{S}^{(m)} \bigcap \mathcal{S}^{(n)} = \Theta^{(m)} \bigcap \Theta^{(n)} \tag{3.4}$$

In other words, AD captures only a subset of the common hidden variables $\Theta^{(m)} \bigcap \Theta^{(n)}$, and in addition, filters out the nuisance variables $\boldsymbol{n}^{(m)}$ and $\boldsymbol{n}^{(n)}$, which are specific to each observation.

The main idea in our method is based on the fact that the desired set of variables Θ can be derived from the union of the pairwise intersections between all pairs, meaning that:

$$\Theta = \bigcup_{m \neq n} \left(\Theta^{(m)} \bigcap \Theta^{(n)}\right). \tag{3.5}$$

A direct implementation of the scheme in (3.5) is not feasible, since the pairwise intersections of $\Theta^{(m)}$ and $\Theta^{(n)}$ are not accessible to us. However, note that by substituting (3.4) in (3.5), we get

$$\Theta = \bigcup_{m \neq n} \left(\mathcal{S}^{(m)} \bigcap \mathcal{S}^{(n)}\right) \tag{3.6}$$

meaning that Θ can be expressed using the accessible observations sets $\mathcal{S}^{(m)}$ through the union of the intersections of all possible pairs. Thus, this scheme for recovering Θ can be implemented by multiple applications of AD to all possible pairs of observable variables.

The union is implemented through the formulation of a new kernel in which the affinity between each pair of samples is given by the sum of the diffusion distances over all pairs of observations. Therefore for each kernel resulting from an application of AD to a single pair of observations, we compute the following diffusion distance $d_t^{(m)\bigcap(n)}$, similarly to (2.9)

$$d_t^{(m)\bigcap(n)}(i,j) = \sqrt{\sum_{l=1}^{N} \frac{\left(\left((K^{(m)\bigcap(n)})^t\right)_{i,l} - \left((K^{(m)\bigcap(n)})^t\right)_{j,l}\right)^2}{\phi_0^{(m)\bigcap(n)}(l)}}, \tag{3.7}$$

where $\boldsymbol{\phi}_0^{(m)\bigcap(n)}$ is the stationary distribution of $\mathbf{K}^{(m)\bigcap(n)}$ and $t > 0$ is a tuneable parameter indicating the number of AD steps. We then define the *common diffusion distance* $d_t^{(\cup)}$ as a summation over the alternating diffusion distances (3.7) resulting from applications to all possible pairs of observations, according to

$$d_t^{(\cup)}(i,j) = \sum_{1\le m,n\le M, m\neq n} d_t^{(m)\bigcap(n)}(i,j) \tag{3.8}$$

where $i,j = 1,\ldots,N$. We now show that $d_t^{(\cup)}$ is a metric.

Proposition 4. *Let X be a set and consider two distance functions $d_1, d_2 : X\times X \to \mathbb{R}_+$. Define $d(x,y) = d_1(x,y) + d_2(x,y)$ for all $x,y \in X$. Then d is a distance function as well. In particular, if $\mathbf{K}^{(m)\bigcap(n)}$ are full rank for all $m,n = 1,\ldots,M, m\neq n$, then $d_t^{(\cup)}$ is a distance function.*

Proof. By definition d is $d : X\times X \to \mathbb{R}_+$. We prove that properties (1)–(3) in Definition 1 hold. Using the symmetry property of d_1 and d_2 we have that $d(x,y) = d(y,x)$. Consider $x,z \in X$, using property (3) of d_1 and d_2, for any $y \in X$ $d(x,z) = d_1(x,z) + d_2(x,z) \le d_1(x,y) + d_1(y,z) + d_2(x,y) + d_2(y,z) = d(x,y) + d(y,z)$. If $x = y$ then $d(x,y) = 0$. If $d(x,y) = 0$, using the non-negativity property (1) of d_1 and d_2 we have that $d_1(x,y) = 0$ and $d_2(x,y) = 0$. From property (1) we obtain that $x = y$.

Now, by Corollary 3, if $\mathbf{K}^{(m)\bigcap(n)}$ is full rank, then, $d_t^{(m)\bigcap(n)}$ is a distance function, and therefore, by a straight-forward generalization, it follows that $d_t^{(\cup)}$ is a distance function. □

Based on the common diffusion distance $d_t^{(\cup)}$, then we calculate an affinity matrix

$$W_{i,j}^{(\cup)} = \exp\left(-\frac{d_t^{(\cup)}(i,j)}{\varepsilon^{(\cup)}}\right), \tag{3.9}$$

where $\varepsilon^{(\cup)} > 0$ is the chosen kernel scale. Next we normalize the affinity matrix and build the common diffusion operator $\mathbf{K}^{(\cup)} \in \mathbb{R}^{N\times N}$

$$Q_{i,i}^{(\cup)} = \left(\sum_{l=1}^{N} W_{i,l}^{(\cup)}\right)^{-1}; \mathbf{K}^{(\cup)} = \mathbf{Q}^{(\cup)}\mathbf{W}^{(\cup)} \tag{3.10}$$

In conclusion, the new graph with kernel $\mathbf{K}^{(\cup)}$ consists of two main components. First, the intersections between any pair of observations $\mathcal{S}^{(m)} \bigcap \mathcal{S}^{(n)}$ are implemented using AD that provides the extraction of the common hidden variables $\Theta^{(m)} \bigcap \Theta^{(n)}$. Second, the union $\bigcup_{m,n} \left(\Theta^{(m)} \bigcap \Theta^{(n)}\right)$ is implemented via the summation of the resulting diffusion distances from the AD applications. By construction, in the kernel $\mathbf{K}^{(\cup)}$, the connectivity between the ith and the jth data samples is proportional to the intrinsic distance $\|\Theta_i - \Theta_j\|$. This means that the common global diffusion kernel $\mathbf{K}^{(\cup)}$ can be used for obtaining a low-dimensional representation of Θ. The proposed scheme described in this section is summarized in Algorithm 4 and is referred to as Common Graph.

Three final remarks follow. First, the proposed implementation of the union via diffusion distance summation enhances the common variables that appear multiple times in the various intersections. By doing so, we slightly abuse the definition of the union, where duplicates are all "put together". In other words, in the strict definition of a union, in contrast to our implementation, common hidden variables related to two or more intersection results should be taken into account only once. Depending on the application at hand, this may be a desired property, and the derivation of a scheme in which each common components has a uniform gain is postponed to future work.

Second, the proposed algorithm can be viewed from a nonlinear filtering standpoint. By applying the proposed algorithm, we maintain or even enhance the common hidden variables, while filtering out the nuisance variables that are sensor/observation-specific.

The third remark is specifically related to the application to sleep stage identification described in Section 3.4. In this application, we have empirically found that a modified computation of $d_t^{(\cup)}$ gives rise to improve performance. This modification results in a "smoother" embedding, better representing the sleep stage. In the alternative implementation, rather than calculating $d_t^{(\cup)}$ as in (3.14), we calculate it in the following way. First, for each pair of sensors we apply the standard DM based on the pairwise kernels $\mathbf{K}^{(m)\bigcap(n)}$, $1 \leq m, n \leq M, m \neq n$ computed in (2.14). For each pair we obtain a $L_{(m)\bigcap(n)}$-dimensional representation, where $L_{(m)\bigcap(n)}$ is a chosen parameter for the pair (m, n), estimated using the "spectral gap" of the decay of the eigenvalues of $\mathbf{K}^{(m)\bigcap(n)}$. Second, we concatenate the low-dimensional representations obtained from the previous step into a single vector. In other words, we now have N concatenated L-dimensional vectors, where $L = \sum_{1\leq m,n\leq M, m\neq n} L_{(m)\bigcap(n)}$, representing the N observations taken simultaneously from all M sensors. Third, we calculate the pairwise distance $d_t^{(\cup)}$ between the N new concatenated vectors. Broadly, this technique is similar to [10], only here we combine the already "filtered" components (the results of AD rather than DM). Since these vectors consist of components from different sensors, we chose to use a modified version of the Mahalanhobis distance. This heuristic method for calculating the common diffusion $d_t^{(\cup)}$ is summarized in Algorithm 3. This modified Mahalanobis distance was first introduced in [26], and since then, was used to exhibit remarkable capability to standardize measurements from different sources, e.g. in [21, 22, 23, 24]. In [22, 23], it was shown to build intrinsic representations by revealing a hidden process driving the measurements. Recently, this technique was applied to multimodal data in [12]. Importantly, compared with AD and our proposed method, these methods [22, 23, 12]

combine the information embodied in all the measurements and do not attempt to suppress nuisance variables or to extract only the common components.

The numerical implementation of the Mahalanobis distance deserves a remark. The computation of the Mahalanobis distance requires estimation of the local covariance matrices of the vectors, each of size $L \times L$. This computation might be computationally cumbersome when L is large, as often in our case. In order to relax the required computational load, prior to the computation of the Mahalanobis distance, one can project the concatenated samples onto a lower dimensional vector space, for example, using RPs [30], and then, compute the Mahalanobis distance for the projected samples with reduced dimensionality.

3.4. Experimental results on the toy problem

Consider the toy problem described in Section 3.2. We simulate 6 hidden scalar variables: 3 common variables $\left(\theta^{(1)}, \theta^{(2)}, \theta^{(3)}\right)$ and 3 nuisance variables $\left(n^{(1)}, n^{(2)}, n^{(3)}\right)$. The variables are statistically independent and uniformly distributed in $[0, 2\pi]$. We then build 3 sets of N RGB images: $\{\boldsymbol{r}_i^{(1)}\}, \{\boldsymbol{r}_i^{(2)}\}, \{\boldsymbol{r}_i^{(3)}\},\ i = 1, \ldots, N$. The sensitivity table of this example is given by

$$\mathbf{S}^T = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}. \tag{3.17}$$

Each image contains 3 arrows, where each arrow is rotated according to a randomly generated angle: the angles of the arrows in $\boldsymbol{r}_i^{(1)}$ are $\left(\theta_i^{(1)}, \theta_i^{(2)}, n_i^{(1)}\right)$, the angles in $\boldsymbol{r}_i^{(2)}$ are $\left(\theta_i^{(2)}, \theta_i^{(3)}, n_i^{(2)}\right)$, and the angles in $\boldsymbol{r}_i^{(3)}$ are $\left(\theta_i^{(3)}, \theta_i^{(1)}, n_i^{(3)}\right)$. The dimensionality of each RGB image is $36 \times 96 \times 3$. We column-stack the RGB images, i.e., $\boldsymbol{r}_i^{(1)}, \boldsymbol{r}_i^{(2)}, \boldsymbol{r}_i^{(3)}$ are vectors of length $J = 10368$. The proposed algorithm is data-driven, and therefore, it does not assume any prior knowledge on the nature of observations. In order to highlight this important property, we use RPs. First, RPs with sufficiently large dimension maintain the underlying geometry, yet the image appearances are lost, which shows that our algorithm does not apply any image processing. Second, in the original images, the different hidden variables are manifested in separate coordinates/pixels; RPs mix the hidden variables, enabling a more challenging extraction task. We generate $D = 1600$ orthonormal vectors $\{\boldsymbol{b}_i\}_{i=1}^D$ of length J and denote by $\mathbf{B} \in \mathbb{R}^{J \times D}$ the matrix whose columns are these random vectors. We build the data of the sensors (cameras) by RPs $\boldsymbol{s}_i^{(m)} = \mathbf{B}^T \boldsymbol{r}_i^{(m)}$, where m is the camera index. In the case of data acquired by cameras, $\mathbf{B}$ can be viewed as the coding system in the cameras. An illustration of the images and their RPs is depicted in Figure 3.2. Illustration of the "movies" of the RPs captured by each camera can be seen in the following link https://youtu.be/91N6mhlYQYY.

We first apply DM separately to each set of observations. Figure 3.3 presents 2-dimensional views of the obtained 3-dimensional embeddings. Each subfigure presents a scatter plot of embedded data-points. Each data-point is an image (a frame in the

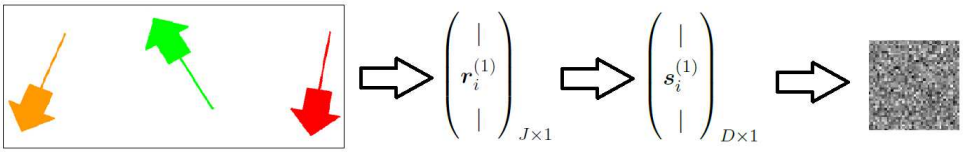


Figure 3.2.: Random projection diagram of the ith image. Each RGB image was column stacked into a vector of length $J = 10368$. Then it was projected on a subspace of $\mathbb{R}^D$ using an orthonormal set $\{v_i\}_{i=1}^D$. The projection is illustrated by a gray-scale 40×40 image. As can be seen the image's property are lost through this projection.

movie) captured by a certain camera after a random-projection $\boldsymbol{s}_i^{(m)}$, where i is the frame index and m is the camera index. The axes of the scatter plot are the first 3 components of the obtained embedding derived from the corresponding camera. The embedded data-points are colored according to the rotating angles $\left(\theta^{(1)}, \theta^{(2)}, \theta^{(3)}\right)$ and 3 noise variables $\left(n^{(1)}, n^{(2)}, n^{(3)}\right)$. It should be noted that this information (the color) was added after calculating the embedding and was not taken into account in the computation of embedding. The subfigure in the lth column and in the mth row contains the embedded data-points derived from the mth camera $\{\boldsymbol{s}_i^{(m)}\}_{i=1}^N$, and its data-points are colored according to the rotating angle of the lth arrow. In the 3 left columns the color coding is according to $\{\theta_i^{(1)}\}_{i=1}^N$, $\{\theta_i^{(2)}\}_{i=1}^N$, $\left\{\theta_i^{(3)}\right\}_{i=1}^N$, and in the 3 right columns the color coding is according to $\{n_i^{(1)}\}_{i=1}^N$, $\{n_i^{(2)}\}_{i=1}^N$, $\{n_i^{(3)}\}_{i=1}^N$. In other words, in each row the same scatter plot is shown, but with different color coding. The 3-dimensional scatter plots are rotated so that the obtained color gradient is best visualized from our 2-dimensional view point. For example, the subfigures in the second row are derived from the observations from the second camera $\{\boldsymbol{s}_i^{(2)}\}_{i=1}^N$. The data-points in the first column are colored according to the rotation angles $\{\theta_i^{(1)}\}_{i=1}^N$, in the second column according to $\{\theta_i^{(2)}\}_{i=1}^N$, etc.

As can be seen, in each row, 3 scatter plots exhibit a smooth color gradient, 2 from the left 3 columns and 1 from the right 3 columns, corresponding to the variables sensed by the respective camera. In the 3 left columns, we see that the color gradients indicates accurate detection of the common variables according to the sensing matrix $\mathbf{S}$. On the 3 right columns, only in the diagonal subfigures exhibit a smooth color gradient, indicating that each captures only its own nuisance variable, as expected. In conclusion, Figure 3.3 implies that the obtained embeddings by DM provide accurate parametrizations of the hidden variables measured by each observation (camera), both the common and the nuisance variables.

The proposed algorithm is applied to the three sets of observations. The obtained embedding is depicted in Figure 3.4. The same 3 dimensional scatter plot of the ob-

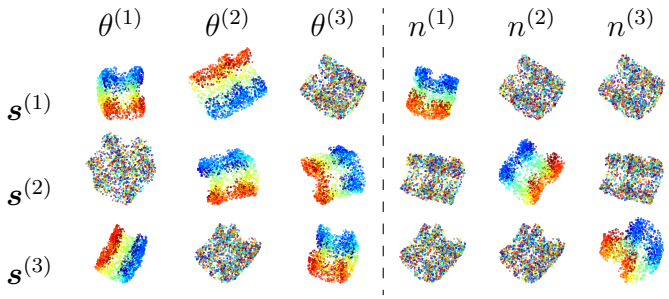


Figure 3.3.: 3D embedding obtained by applying diffusion map on a single observer. The subfigures are arranged such that subfigures in each row are obtained from the same observer. The data-points in each column are colored according to different arrow's rotation angles.

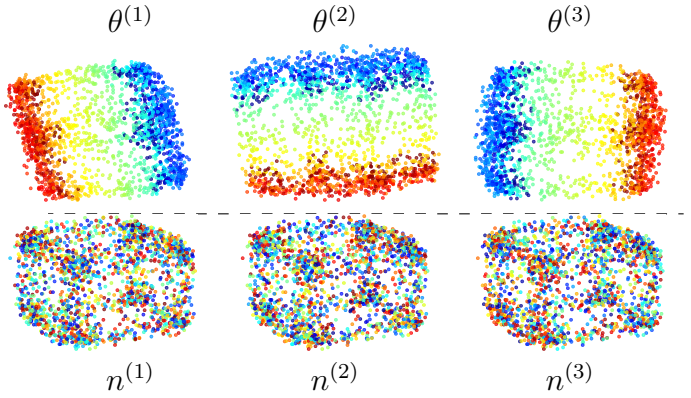


Figure 3.4.: 3D embedding obtained by applying the proposed algorithm on the observers set. The subplots in the first rows are colored according to the common variables, the subplots in the second row are colored according to the noise variables. As can be seen, the obtained parametrization corresponds to the common variables.

tained embedding is shown with different color coding. The subfigures in the top row are colored (from left to right) according to the common variables $\left\{\theta_i^{(1)}\right\}_{i=1}^{N}$, $\left\{\theta_i^{(2)}\right\}_{i=1}^{N}$, $\left\{\theta_i^{(3)}\right\}_{i=1}^{N}$, while the subfigures in the bottom row are colored (from left to right) according to the nuisance variables $\left\{n_i^{(1)}\right\}_{i=1}^{N}$, $\left\{n_i^{(2)}\right\}_{i=1}^{N}$, $\left\{n_i^{(3)}\right\}_{i=1}^{N}$. As in Figure 3.3, the 3 dimensional embedding is rotated, such that the corresponding color gradient is emphasized from the depicted 2 dimensional point of view. We can see from the obtained color gradients that the embedding provides a parametrization of only the common variables, meaning that the proposed algorithm manages to extract all 3 of the common variables $\left(\theta^{(1)}, \theta^{(2)}, \theta^{(3)}\right)$ (despite having none in common to all *three* observations), while suppressing all 3 nuisance observation-specific variables $\left(n^{(1)}, n^{(2)}, n^{(3)}\right)$. Upon publication, the Matlab code and data of this toy problem will be made available online.

Algorithm 3 Mahalanobis-based Union Scheme

Input: $M(M-1)$ alternating-diffusion operators $\mathbf{K}^{(m)\bigcap(n)}$, $1 \leq m, n \leq M, m \neq n$
Output: Alternating-diffusion distance $d_t^{(\cup)}$

1. For $1 \leq m, n \leq M, m \neq n$, calculate the spectral decomposition of each kernel $\mathbf{K}^{(m)\bigcap(n)}$, and obtain its eigenvalues $\left\{\lambda_l^{(m)\bigcap(n)}\right\}_{l=0}^{N-1}$ and eigenvectors $\left\{\boldsymbol{\psi}_l^{(m)\bigcap(n)}\right\}_{l=0}^{N-1}$.

2. For $1 \leq m, n \leq M, m \neq n$, build an $L_{(m)\bigcap(n)}$-dimensional representation using standard DM (2.6) for each time sample $i = 1 \dots N$.

$$\Psi_t^{(m)\bigcap(n)}(i) = [\lambda_1^t \psi_1^{(m)\bigcap(n)}(i), \lambda_2^t \psi_2^{(m)\bigcap(n)}(i), \dots, \lambda_{L_{(m)\bigcap(n)}}^t \psi_{L_{(m)\bigcap(n)}}^{(m)\bigcap(n)}(i)] \tag{3.11}$$

   where $t > 0$ is a tuneable parameter.

3. For each time sample $i = 1 \dots N$, concatenate the low-dimensional representations into a single vector

$$\begin{aligned}\Psi_t^{(\cup)}(i) = \Big(& \Psi_t^{(1)\bigcap(2)}(i), \Psi_t^{(1)\bigcap(3)}(i), \dots, \Psi_t^{(1)\bigcap(M)}(i), \\ & \Psi_t^{(2)\bigcap(1)}(i), \Psi_t^{(2)\bigcap(3)}(i), \dots, \Psi_t^{(2)\bigcap(M)}(i), \\ & \dots \\ & \Psi_t^{(M)\bigcap(1)}(i), \Psi_t^{(M)\bigcap(2)}(i), \dots, \Psi_t^{(M)\bigcap(M-1)}(i) \Big)\end{aligned} \tag{3.12}$$

4. Calculate $d_t^{(\cup)}$ using the Mahalanobis distance:

$$d_t^{(\cup)}(i,j) = \|\Psi_t^{(\cup)}(i) - \Psi_t^{(\cup)}(j)\|_{\text{Mahalanobis}} \tag{3.13}$$

   for $i, j = 1, \dots, N$.

Algorithm 4 Common Graph

Input: Aligned samples from M sets of observations: $\left\{(\boldsymbol{s}_i^{(1)}, \boldsymbol{s}_i^{(2)}, \ldots, \boldsymbol{s}_i^{(M)})\right\}_{i=1}^{N}$.
Output: Low-dimensional representation of the common hidden random variables Θ.

1. For each pair of observation sets $1 \leq m, n \leq M, m \neq n$, apply alternating diffusion (Algorithm 2), and obtain the diffusion distance $d_t^{(m)\bigcap(n)}$.

2. Compute the distance $d_t^{(\cup)}$

$$d_t^{(\cup)}(i,j) = \sum_{1 \leq m,n \leq M, m \neq n} d_t^{(m)\bigcap(n)}(i,j) \tag{3.14}$$

   for $i, j = 1, \ldots, N$.

3. Based on the common diffusion distance $d_t^{(\cup)}$ calculate an affinity matrix

$$W_{i,j}^{(\cup)} = \exp\left(-\frac{(d_t^{(\cup)}(i,j))^2}{\varepsilon^{(\cup)}}\right) \tag{3.15}$$

4. Construct the diffusion operator $\mathbf{K}^{(\cup)}$:

$$Q_{i,i}^{(\cup)} = \left(\sum_{l=1}^{N} W_{i,l}^{(\cup)}\right)^{-1}; \mathbf{K}^{(\cup)} = \mathbf{Q}^{(\cup)}\mathbf{W}^{(\cup)} \tag{3.16}$$

5. Apply standard diffusion maps (steps 3 and 4 in Algorithm 1) using $\mathbf{K}^{(\cup)}$, and obtain an L-dimensional representation of Θ.

4. Application to Sleep Stage Assessment

As mentioned in Chapter 3, the problem of extracting the common hidden variables from multiple data sets acquired by different observables can be perceived as a problem of nonlinear filtering. To demonstrate the potential of the particular nonlinear filtering scheme presented in Chapter 3 in processing real data, we apply the proposed algorithm to sleep data, where the ultimate goal is to devise an automatic system for sleep stage assessment.

4.1. Sleep: introduction and background

Sleep is a global and recurrent physiological process, which is in charge of the memory consolidation, the learning redistribution, tissue regeneration, immune system enhancement, etc [31]. The sleep dynamics are characterized by particular temporal physiological features, which are intimately related to the quality of sleep. The clinically acceptable sleep stage is mainly determined by reading recorded electroencephalogram (EEG) signals based on the Rechtschaffen and Kales (R&K) criteria [32, 33]. In the R&K criteria, the sleep dynamics are divided into two broad stages: rapid eye movement (REM), and non-rapid eye movement (NREM) [31]. The NREM stage is further divided into two shallow sleep stages, which are denoted N1 and N2, and a deep sleep stage, which is denoted N3. In addition to the interest stemming from physiological aspects, sleep stage assessment has important clinical applications. For example, REM is associated with perceptual skill improvement [34], NREM sleep is associated with Alzheimer's disease [35], poor sleep quality is associated with weaning failure [36], etc. Besides personal health purposes, the sleep quality is also responsible for several public catastrophes [37]. These facts indicate the importance of an accurate automatic annotation system for sleep stage assessment and its broad applications.

In the past decades, various automatic annotation methods have been proposed. Those methods mainly extract various features from the EEG recordings for the purpose of studying sleep dynamics [38], such as time domain summary statistics, spectral or coherence features, time-frequency features, and information entropy, just to name a few [39, 40, 41]. Recently, a theoretically solid approach suitable for analyzing and estimating the dynamics of the brain activity from recorded EEG signals has been proposed in [22, 23]. A particular aspect of sleep dynamics, which has not gained much research attention in the line of research mentioned above, is that sleep is not localized solely in the brain and is reflected in other physiological systems as well. For example, the regulation

of mechanoreceptor and the chemoreceptor leads to breathing pattern variability in the respiratory signal. We have a remarkably regular breathing during N3 stage and irregular breathing with fast varying instantaneous frequency and amplitude during REM stage. Those physiological phenomena motivated various studies to explore the relation between the sleep stage and the patterns in the respiratory signals, e.g. [42, 43, 44]. Physiologically, these variations are not originated from the same controller, and phenomenologically do not have the same patterns in the recorded time series. Thus, while we could observe the sleep dynamics via observing the characteristics of different sensors, each of them reflects only part of the sleep dynamic, and is complicated by the nature of the sensor.

4.2. Previous work and relation to the common graph problem

Based on the above description, an automatic approach for assessing the sleep stage was presented in [21]. It relies on the assumption that there exist hidden low-dimensional physiological processes driving the sleep dynamics, and hence the accessible measured signals. However, these hidden processes may be deformed by the observation procedures; each observation (e.g., an EEG channel measuring brain activity or a chest belt measuring respiration) can be influenced by nuisance factors, which are sensor- or channel-specific (e.g., the specific type of sensors and their exact positions), yet our interest is in the intrinsic variables related to the sleep stages. In [21], empirical intrinsic geometry (EIG) method [22, 23], which is based on nonlinear independent component analysis [26] and was proven to be invariant to the measurement modality, was applied for building an intrinsic representation of the measured data. In [45], this method was extended to a pair of sensors. It was shown that by analyzing the measurements taken simultaneously from two sensors, a more reliable intrinsic representation of the sleep dynamics can be obtained, compared with the analysis based only on a single signal.

Here we extend the algorithm shown in [45], and process jointly multiple channels. We show that extracting the underlying common variables from multiple data sets acquired in different channels recovers systematically a representation, which is well correlated with the sleep stage. The analogy to the setting described in chapter 3 is as follows. We assume that the sleep dynamics are intimately related to hidden controllers that affect the respiratory as well as the brain neural system. These controllers are not accessible to us; yet, they can be recovered by analyzing observations from multiple channels/sensors, each captures different, partial yet complementary aspects of it. Under this assumption, our interest is in obtaining the intrinsic variables underlying the measurements related to these controllers. On the one hand, by analyzing multiple observation channels we can gather more information on the hidden controllers. On the other hand, observations from each channel might be deformed by the different acquisition and measurement modalities and may be affected by noise and interferences, specific to the particular (type of) sensor. In the context of this work, this trade off is addressed by defining the

intrinsic variables (related to the hidden controllers of interest) as those which are not sensor-specific, and hence, the variables of interest are those that are common among at least two observables.

4.3. Experimental setup and implementation details

Data collected from twenty subjects without sleep apnea were examined. The demographic characteristics of these individuals fall within the normal ranges. We used recordings of 6 hours per subject, which were performed in the sleep center at Chang Gung Memorial Hospital (CGMH), Linkou, Taoyuan, Taiwan. The institutional review board of the CGMH approved the study protocol (No. 101-4968A3) and the enrolled subjects provided written informed consent. See [21] for more details.

We apply the common graph according to Algorithm 4 for extracting the common hidden variables separately to two sets of sensors. The first set includes 3 signals: abdominal and chest motions, which are recorded by piezo-electric bands, and airflow, which is measured using thermistors and nasal pressure, all 3 at sampling rate of 100 Hz. The second set comprises recordings from 4 EEG channels: C3A2, C4A1, O1A2 and O2A1 at sampling rate of 200 Hz. The recorded respiratory signals are denoted by $R_m, m = 1, \ldots, 3$ and the EEG signals are denoted by $E_m, m = 1, \ldots, 4$.

Prior to the application of our method, each of the single-channel recordings was preprocessed by applying the scattering transform as in [21], which was shown to improve the regularity and stability of signals with respect to various deformations [46]. We then apply Algorithm 4 separately twice: once to the respiratory set, and once to the EEG set.

In order to demonstrate the inherent "sensor selection" capability of the proposed method, for each set of measurements we added an artificial "pure noise" sensor to simulate possible sensor failure. To further demonstrate the robustness of this sensor selection, the noise sensor consists of a highly non-stationary sequence generated by modulating a sine-wave according to

$$\begin{aligned} \phi(\tau) &= \frac{1}{2} + \frac{1}{4}\sin\left(\frac{2\pi\tau}{512 \cdot 10}\right) \\ n(t) &= \sin\left(2\pi \int_0^t \phi(\tau) d\tau\right) \end{aligned} \tag{4.1}$$

where $n(t)$ is the continuous time signal and $\phi(\tau)$ can be viewed as the instantaneous frequency of $n(t)$. We sample the obtained modulated sine-wave $n(t)$ at a sampling rates of 200 Hz and 100 Hz for the EEG set and for the respiratory set, respectively. It should be noted that this particular non-stationary "noise" implementation was chosen just for the sake of demonstration, and any other (non-stationary) sequence could be chosen instead.

We compare the results of the common graph algorithm, analyzing multiple sensors, with the results attained by the standard DM applied separately to each individual sensor. In addition, we compare the results to two competing schemes analyzing multiple

sensors. In the first scheme, we concatenate the scattering transform components from each sensor, and then, apply the standard DM. We note that conceptually this scheme takes into account the information captured by all the sensors without any filtering. We refer to the first scheme as the *concatenation scheme*. In the second scheme, we apply AD to the entire set of sensors. Namely, we calculate the diffusion kernel $\mathbf{K}^{(m)}$ for the mth sensor, where $m = 1 \dots M$ and build an AD kernel based on the product of all the kernels, that is, $\mathbf{K} = \mathbf{K}^{(1)}\mathbf{K}^{(2)} \cdots \mathbf{K}^{(M)}$. Then, we apply DM with this AD kernel. Theoretically, this scheme takes into account only the information that is captured simultaneously by all of the sensors, namely $\bigcap_{m \neq n} \left(\mathcal{S}^{(m)} \bigcap \mathcal{S}^{(n)}\right)$, thereby performing excessive filtering. We refer to the second scheme as the *multiplication scheme*.

Two additional remarks regarding the implementation. First, the calculation of the affinity matrices, which is a core element in the tested methods, is carried out using the Mahalanobis distance variant presented in [26], which was discussed in Section 3.3. Second, to be able to depict information embodied in more than three eigenvectors, we randomly project the embeddings attained by the competing algorithms to 3 dimensions. This allows us to visually inspect the portion of the relevant information and the portion of the nuisance information manifested in the representations obtained by the different algorithms. We use the same projection in all tested methods.

4.4. Results

The RPs of the embeddings are depicted in Figure 4.1. The RPs based on the single channel DM applied to the O2A1 EEG channel and to the airflow channel are depicted in the top row. The RPs based on the concatenation scheme, multiplication scheme and the proposed algorithm are depicted in the second, third and bottom rows, respectively. The embeddings depicted in the left column are based on the EEG set, and the embeddings depicted in the right column are based on the respiratory set. Each embedded point is colored according to its respective sleep stage, as identified by a human expert. Importantly, the information on the sleep stage (e.g., the color) was not taken into account in the algorithms forming the embeddings.

Figure 4.1 provides a visual illustration of the obtained parametrization with respect to the sleep stage. By comparing the Figure 4.1a and Figure 4.1b with Figure 4.1g and Figure 4.1h we can observe the improvement achieved by the additional information obtained from combining information from multiple sensors. In addition, by comparing Figures 4.1c-4.1f with Figure 4.1g and Figure 4.1h we observe the improvement achieved by filtering out of the sensor-specific nuisance variables. In these comparisons, it can be seen that the embeddings obtained by using the proposed algorithm results in a better parametrization of the sleep stage evaluation; different sleep states appear to be more separated, especially in the case of the respiratory signals.

To objectively assess the quality of the obtained embeddings, we use multi-class support vector machine (SVM). To ensure convergence and to prevent overfitting, we process only the 15 most dominant eigenvectors from each embedding. It should be noted that due to the obtained fast decay of the eigenvalues, taking only the 15 most dominant

eigenvectors preserves the geometrical structure of the data. We randomly partition the data into 2 sets – a training set (consisting of 75% of the samples) and a validation set (consisting of 25% of the samples). The validation set contains $1,250$ time segments, which consists (on average) of 13.2%(165) segments labeled as awake stage, 10%(125) segments labeled as REM stage, 11.2%(140) segments labeled as N1 stage, 49.6%(620) segments labeled as N2 stage and 16%(200) segments labeled as N3 stage. The trained classifier is used to classify the sleep stage in the validation set. We repeat this classification 10 times, for different randomly chosen partitions of training and validation sets. The average classification results for each scheme are depicted in Table 4.1. The obtained classification results achieved by the proposed algorithm are superior compared to the obtained results from other schemes, both in the case of the EEG set and in the case of the respiratory set. In these results, the advantages of proper filtering are evident, as it can be seen that in contrast to the proposed algorithm, the concatenation scheme and the multiplication scheme attain inferior classification results, and in some cases, their results are comparable to the results achieved by processing data from only a single sensor. In the case of the multiplication scheme this may be explained by too excessively filtering. In the case of the concatenation scheme, where no filtering is applied, this may be explained by the existence of interferences and noise.

The results in Table 4.1 may provide additional insights related to the sleep dynamics that extend the scope of the evaluation of the algorithms. The classification results achieved by the multiplication scheme are inferior comparing to the results achieved by single-sensor schemes in the case of the respiratory set, whereas in the case of the EEG set the achieved results are similar to the single-sensor schemes. This supports the hypothesis that different EEG recordings exhibit more homogeneous geometrical structures, with possibly less noise and fewer distortions, compared to the data acquired via the different respiratory recordings.

The homogeneity of the EEG set might explain another interesting observation stemming from these classification results. In the case of the EEG set, we can see that combining the information acquired from multiple sensors using the proposed algorithm results with superior results, even compared to the results that would have been achieved using the best single-sensor scheme. This implies that the proposed algorithm manages to simultaneously cancel the effect of the additional noise-sensor as well as to properly integrate the information embodied in the multiple sensors. Conversely, in the case of the respiratory set, we can see that even though the proposed algorithm manages to improve the results achieved by the CFlow channel, it did not manage to improve the results achieved by the single sensor schemes based on the ABD or the THO channel. Yet, it did manage to cancel the effect of the noise-sensor, but not as successfully as in the case of the EEG set. In this regard, it is worth emphasizing that the evaluation of the results from each sensor and from each scheme are based on unknown sleep stage labelling. Thus, the "quality" of the different sensors are not known in advance, and obtaining a result based on our sensor fusion scheme that is comparable to the results attained by the best single sensor is still of value.

Figure 4.2 further illustrates the poor embeddings and classification results achieved by the concatenation scheme. The same embeddings, which are depicted in Figure

Table 4.1.: Classification results using SVM. The prediction errors (standard deviations) based on the different embeddings are presented. The total error (standard deviation) is calculated by a weighted mean of the prediction errors (standard deviations) in each sleep stage. (a) The classification based on the respiratory set. (b) The classification based on the EEG set.

Prediction errors based on the respiratory set						
Sensor/Scheme	Awake	REM	N1	N2	N3	Total
CFlow	0.383	0.258	0.545	0.179	0.244	0.264 (0.073)
ABD	0.299	0.154	0.426	0.162	0.185	0.209 (0.051)
THO	0.262	0.15	0.412	0.153	0.164	0.196 (0.051)
Noise	0.559	0.513	0.583	0.582	0.529	0.562 (0.039)
Concatenation scheme	0.476	0.474	0.552	0.516	0.5	0.507 (0.039)
Multiplication scheme	0.343	0.262	0.555	0.265	0.265	0.307 (0.069)
Common Graph	0.252	0.183	0.443	0.178	0.201	0.22 (0.048)

(a)

Prediction errors based on the EEG set						
Sensor/Scheme	Awake	REM	N1	N2	N3	Total
O1A2	0.267	0.281	0.624	0.132	0.273	0.25 (0.056)
O2A1	0.283	0.25	0.603	0.142	0.24	0.244 (0.069)
C4A1	0.305	0.273	0.623	0.139	0.291	0.258 (0.068)
C3A2	0.298	0.276	0.619	0.132	0.289	0.254 (0.06)
Noise	0.551	0.499	0.579	0.58	0.53	0.557 (0.042)
Concatenation scheme	0.342	0.325	0.499	0.307	0.325	0.34 (0.039)
Multiplication scheme	0.219	0.191	0.47	0.238	0.2	0.252 (0.045)
Common Graph	0.227	0.153	0.425	0.133	0.179	0.188 (0.036)

(b)

4.1, are presented here, but this time with a different color – now according to the instantaneous frequency of the noise sensor (4.1). As can be observed, in contrast to the embeddings achieved by the proposed algorithm or by the multiplication scheme, the embeddings achieved by the the concatenation scheme are well correlated with the instantaneous frequency in the noise sensor, indicating that the undelying structure is wrongly captured. This further illustrates the difference between the filtering effects of our algorithm and other methods, which are based to the fusion of data from all the sensors [9, 10, 12].

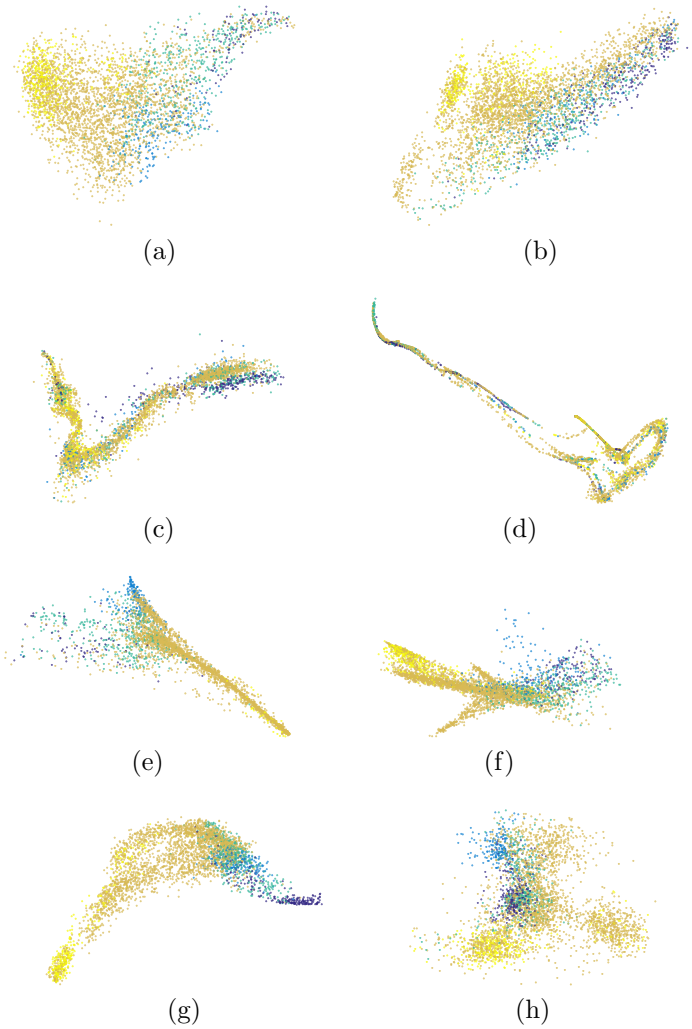
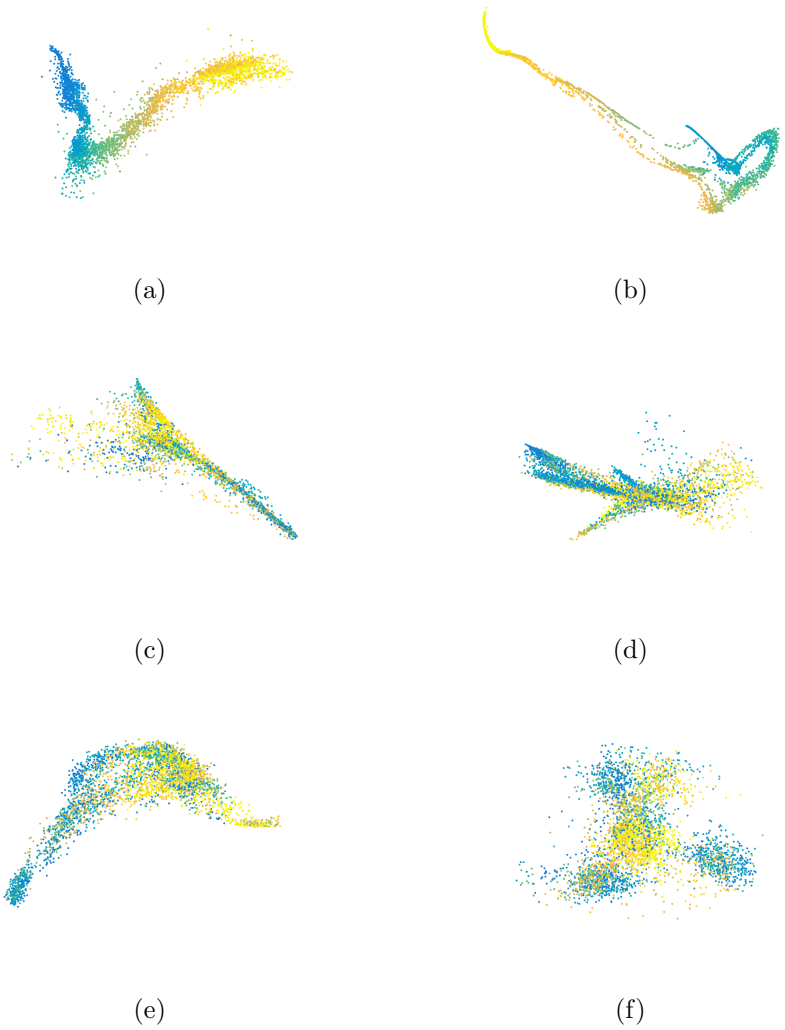


Figure 4.1.: The 3D RPs of the embeddings obtained by single-sensor DM (top row), the concatenation scheme (second row), the multiplication scheme (third row), and Algorithm 4 (bottom row). The points are colored according to the sleep stage. The embeddings are based on the O2A1 channel in (a) and on the airflow measurements in (b). From the second row to the bottom row, the embeddings on the left column are based on the EEG set, and on the right column are based on the respiratory set.



(a) (b)

(c) (d)

(e) (f)

Figure 4.2.: The same embeddings as in Figure 4.1, colored according to the instantaneous frequency of the noise sensor.

5. Mahalanobis Distance Estimation for Manifold Learning

A core element in any manifold learning technique is having the ability to reveal the similarity between data points. This is usually done using a distance metric, which should be robust to noise and recover the underlying structure of the data. The Mahalanobis distance is such a metric:

$$\|\boldsymbol{y_2} - \boldsymbol{y_1}\|_M^2 = \frac{1}{2}\big(\boldsymbol{y_2} - \boldsymbol{y_1}\big)^T \mathbf{C^{-1}} \big(\boldsymbol{y_2} - \boldsymbol{y_1}\big). \tag{5.1}$$

An essential element for calculating the Mahalanobis distance is the estimation of the covariance matrix $\mathbf{C}$. In this work we will focus on data sets acquired with temporal order. In classical manifold learning techniques time series are processed as data sets of samples, ignoring their temporal dynamics. Here we will incorporate the time dependency of consecutive data points in the time series and utilize it for the purpose of the covariance matrix estimation. For example, the covariance estimation proposed in [26] is based on a finite window length. The task of estimating the covariance matrix from a noisy dataset is challenging, especially when it is applied to data sampled from multi-scale stochastic dynamical systems. In [25] a rigorous analysis of the covariance estimation error is provided. The errors analyzed in [25] originate from two sources: the dynamics of the system and the curvature of the measurement function. Another source of error which is not taken into account in [25] is the finite sampling of the data. Demonstrations illustrating the influence of the finite sampling are given in Appendix A, where we discuss the influence of the estimation errors on various aspects of manifold learning as well as the considerations that influence the choice of the window length.

The remainder of this chapter is structured as follows. In Section 5.1 we formulate the problem. In Section 5.2 we propose a new method for estimating the covarince matrix that deals with the inherent trade off between preserving locality and minimizing the sample-variance error. In Section 5.3 we apply the proposed algorithm to a simulation of a multiscale stochastic dynamical system [25] and demonstrate its performance. In Section 5.4 we analyze the error incurred due to the chosen window length and compare it to the errors analysis provided in [25]. We note that the formulation and the proposed solution in this chapter are presented for a general multi-dimensional process, yet, the analysis and demonstrations are given for a one dimensional process. Extending the analysis for higher dimensions is postponed for future work.

5.1. Formulation and definitions

Consider a system driven by a hidden-variable $\boldsymbol{x_t} \in \mathbb{R}^n$. We denote the trajectory of $\boldsymbol{x_t}$ as the process in the intrinsic space, which is inaccessible. Instead, we observe its mapping through a non-linear function $f : \mathbb{R}^n \to \mathbb{R}^d, n \leq d$:

$$y_t^j = f(\boldsymbol{x_t}), \quad 1 \leq j \leq d \tag{5.2}$$

We refer to $\boldsymbol{y_t}$ as the process in the observable space. Given a sample of N datapoints from a trajectory $\boldsymbol{y_t}$ in the observable space: $\{\boldsymbol{y_{t_i}}\}_{i=1}^N$, the problem here is to recover a parametrization that is one-to-one with the diffusion process in the intrinsic space $\{\boldsymbol{x_{t_i}}\}_{i=1}^N$.

In the context of manifold learning, a primary task is to recover the pairwise distances in the intrinsic space. While, the inverse of the mapping function f is unknown, by assuming that f is locally bilipschitz, we can use its linear approximation at every point. In [26] the authors use linear approximation in order to express the Euclidean distance between the samples in the intrinsic space, $\boldsymbol{x_t}$, using the accessible samples in the observable space $\boldsymbol{y_t}$:

$$\|\boldsymbol{x_{t_2}} - \boldsymbol{x_{t_1}}\|^2 = \frac{1}{2}\big(\boldsymbol{y_{t_2}} - \boldsymbol{y_{t_1}}\big)^T \Big((\mathbf{J}\mathbf{J}^T)^{-1}(\boldsymbol{y_{t_2}}) + (\mathbf{J}\mathbf{J}^T)^{-1}(\boldsymbol{y_{t_1}}) \Big) \big(\boldsymbol{y_{t_2}} - \boldsymbol{y_{t_1}}\big) + \mathcal{O}(\|\boldsymbol{y_{t_2}} - \boldsymbol{y_{t_1}}\|_2^4) \tag{5.3}$$

where $\mathbf{J}$ is the $d \times n$ Jacobian matrix of f, whose elements are

$$J_{ij} = \frac{\partial f^i}{\partial x^j}, 1 \leq i \leq d, 1 \leq j \leq n \tag{5.4}$$

The approximation in (5.3) cannot be evaluated directly since that the measurement function f and its Jacobian $\mathbf{J}$ are unknown. Therefore they should be estimated from the data. In order to be able to estimate f (or $\mathbf{J}$) from the sampled datapoints, an assumption about the data should be made. A common assumption in the framework of manifold learning is that the hidden variable is driven by a Brownian motion [47, 48], and defined by the following stochastic differential equation (SDE):

$$dx_t^i = a^i(x_t^i)dt + b^i(x_t^i)d\omega_t^i, \quad 1 \leq i \leq n \tag{5.5}$$

where ω_t^i are independent standard Brownian motions, $a^i(\cdot)$ are drift functions and $b^i(\cdot)$ are the diffusion coefficients. In [26] the authors use this assumption in order to estimate (5.3). According to Itô's lemma the coordinates of the process $\boldsymbol{y_t}$ satisfy the following stochastic differential equation:

$$dy_t^j = \sum_{i=1}^{d} \big(\frac{1}{2}(b^i)^2 f_{ii}^j + a^i f_i^j\big)dt + \sum_{i=1}^{d} b^i f_i^j d\omega_t^i \tag{5.6}$$

So that the covariance of y_t is given by

$$C_{jk} \triangleq Cov(dy^j, dy^k) = \sum_{i=1}^{d} (b^i)^2 f_i^j f_i^k \tag{5.7}$$

And in matrix form can be written as:

$$\mathbf{C} = \mathbf{J}\mathbf{B}^2\mathbf{J}^T \tag{5.8}$$

where $\mathbf{B}$ is a diagonal matrix with $B_{ii} = b^i(x^i)$. Assuming that $\mathbf{B}$ is the identity matrix, we can express (5.3) as:

$$\|\boldsymbol{x_{t_2}} - \boldsymbol{x_{t_1}}\|_2^2 = \|\boldsymbol{y_{t_2}} - \boldsymbol{y_{t_1}}\|_M^2 + \mathcal{O}(\|\boldsymbol{y_{t_2}} - \boldsymbol{y_{t_1}}\|_2^4) \tag{5.9}$$

Where $\|\cdot\|_M$ is a modification of the Mahalanobis distance, which is defined by:

$$\|\boldsymbol{y_{t_2}} - \boldsymbol{y_{t_1}}\|_M^2 = \frac{1}{2}\big(\boldsymbol{y_{t_2}} - \boldsymbol{y_{t_1}}\big)^T\Bigg(C^{-1}(\boldsymbol{y_{t_2}}) + C^{-1}(\boldsymbol{y_{t_1}})\Bigg)\big(\boldsymbol{y_{t_2}} - \boldsymbol{y_{t_1}}\big) \tag{5.10}$$

where $C(\boldsymbol{y_t})$ is the covariance of the observed diffusion process at $\boldsymbol{y_t}$. The calculation of the Mahalanobis distance requires knowledge of the covariance matrix, which is not accessible to us. However, we can estimate the covariance matrices empirically. There are various of methods for estimating the covariance matrix; in this work we assume that the samples are given in temporal order (a sequence in time), and consider a method that estimates the covariance from the increments of the diffusion process in the observable space:

$$\boldsymbol{dy_{t_i}} \triangleq \boldsymbol{y_{t_i}} - \boldsymbol{y_{t_{i-1}}} \tag{5.11}$$

Such that the estimator for the covariance matrix at a certain datapoint $\boldsymbol{y_{t_i}}$ is:

$$\widehat{\mathbf{C}}(\boldsymbol{y_{t_i}}) = \frac{1}{N_w}\sum_{j=1}^{N}\big(\boldsymbol{dy_{t_{i+j}}}\big)\big(\boldsymbol{dy_{t_{i+j}}}\big)^T - \big(\mu_{\boldsymbol{dy_{t_i}}}\big)\big(\mu_{\boldsymbol{dy_{t_i}}}\big)^T \tag{5.12}$$

Where N is a chosen parameter specifying the window-length used for the covariance estimation and $\mu_{\boldsymbol{y_{t_i}}}$ is the empirical mean at time t_i, i.e.:

$$\mu_{\boldsymbol{dy_{t_i}}} = \sum_{j=1}^{N}\boldsymbol{dy_{t_{i+j}}} \tag{5.13}$$

As mentioned above, this estimation suffers from many errors when it is applied to data sampled from multi-scale stochastic dynamical systems. In Section 5.4 we will discuss these errors in more details.

5.2. Proposed solution

In this section we propose a heuristic algorithm for adaptive estimation of the covariance. We consider the multi-dimensional Itô's drift-diffusion process in (5.5) measured by a possibly non-linear bilipschitz function, as described in (5.2). Given a set of measured datapoints $\{\boldsymbol{y_{t_i}}\}_{i=1}^N$, our goal is to estimate the covariance matrix at a certain datapoint.

In contrast to the estimator described in (5.12), where the estimation is carried out within an arbitrary window in the time domain, we propose to estimate the covariance matrix within a ball in the observable space of $\boldsymbol{y_t}$. The proposed estimator is obtained by the following expression:

$$\widehat{C}(\boldsymbol{y_0})\Big|_{\mathcal{B}_R(\boldsymbol{y_0})} = \frac{1}{N_{\mathcal{B}_R(\boldsymbol{y_0})}} \sum_{\boldsymbol{y}\in\mathcal{B}_R(\boldsymbol{y_0})} (\boldsymbol{dy})(\boldsymbol{dy})^T - \left(\mu_{\boldsymbol{dy}}|_{\mathcal{B}_R(\boldsymbol{y_0})} \right)\left(\mu_{\boldsymbol{dy}}|_{\mathcal{B}_R(\boldsymbol{y_0})} \right)^T \tag{5.14}$$

where $\mathcal{B}_R(\boldsymbol{y_0})$ is a set containing the datapoints that resides within a d-dimensional ball centered at $\boldsymbol{y_0}$ with a radius R. The motivation for this choice is that the covariance matrix at a certain point depends directly on its coordinates in the observable space, and indirectly on its time-index. By estimating the covariance matrix locally within a ball around a certain datapoint in the observable space we can gather several realizations of the diffusion process taken from different passages, possibly at different times, of the diffusion process around that point. By doing so we manage to suppress the error incurs by the sample variance and maintaining the locality of our estimation. In Figure 5.1 we demonstrate such a situation, in which two trajectories intersect at the same point of interest at different times. By definition, in our context, the two trajectories

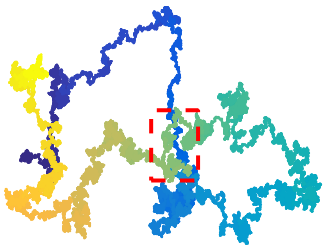


Figure 5.1.: Illustration of a 2-dimensional diffusion process, the datapoints are colored according to the time index. The marked region contains passages of two trajectories, from different phases of the process.

originating from the point of interest are equivalent for the purpose of estimating the local covariance, and hence, both are used. It should be noted that we still utilize the fact that the datapoints are time arranged by estimating the covariance from the increments of the diffusion process in the observable space (as in (5.12)).

Here, the problem of finding the optimal window length N is analogous to the problem of finding the optimal radius R. We propose an adaptive method for choosing the proper radius for the covariance matrix estimation. The method is based on a heuristic criterion for evaluating the distortion that incurs using a certain radius. For a certain datapoint, $\boldsymbol{y_0}$, given a certain radius, R, we pick $\mathcal{B}_R(\boldsymbol{y_0})$. For each datapoint in $\mathcal{B}_R(\boldsymbol{y_0})$ we calculate the distance from $\boldsymbol{y_0}$ and denote the median of the obtained distances by R^*. We take $\mathcal{B}_R(\boldsymbol{y_0})$ and divide it into two equally-sized subsets: estimation subset and validation subset. The estimation subset contains the datapoints in $\mathcal{B}_R(\boldsymbol{y_0})$ whose distances from $\boldsymbol{y_0}$ are smaller than R^*, i.e. $\mathcal{B}_{R^*}(\boldsymbol{y_0})$. The validation subset consists of the remaining points in $\mathcal{B}_R(\boldsymbol{y_0})$, i.e. $\mathcal{B}_R(\boldsymbol{y_0})\backslash\mathcal{B}_{R^*}(\boldsymbol{y_0})$. We calculate the estimation of the covariance matrix

using the increments of the datapoints in the estimation set, and denote it by $\widehat{C_e}(\boldsymbol{y_0}) = \widehat{C}(\boldsymbol{y_0})\Big|_{\mathcal{B}_{R^*}(\boldsymbol{y_0})}$. We then calculate its eigen-value decomposition: $\widehat{C_e}(\boldsymbol{y_0}) = \mathbf{V}\mathbf{E}\mathbf{V}^T$, and formulate a rescale matrix, $\mathbf{Q}$, which is defined by $\mathbf{Q} \triangleq \mathbf{V}\mathbf{E}^{-1/2}\mathbf{V}^T$. We multiply each of the increments of the datapoints in the validation set by $\mathbf{Q}$ and calculate the sample covariance matrix, i.e. $\widehat{Q_v}(\boldsymbol{y_0}) = \mathbf{Q}^T\widehat{C_v}(\boldsymbol{y_0})\mathbf{Q}$ where $\widehat{C_v}(\boldsymbol{y_0}) = \widehat{C}(\boldsymbol{y_0})\Big|_{\mathcal{B}_R(\boldsymbol{y_0})\backslash\mathcal{B}_{R^*}(\boldsymbol{y_0})}$.

If we managed to accurately capture the local behavior of the measurement function around $\boldsymbol{y_0}$ via $\widehat{C_e}(\boldsymbol{y_0})$ then $\widehat{Q_v}(\boldsymbol{y_0})$ is the identity matrix. When the tested ball is to small and the sample variance is dominant, or when the tested ball is too large, and the curvature of function distorts the estimation, $\widehat{Q_v}(\boldsymbol{y_0})$ is distorted. We measure these distortions by the Frobenius norm of $\widehat{Q_v}(\boldsymbol{y_0}) - \mathbf{I}$. We repeat the previous procedure for a range of radii R and choose the radius that provides the minimal distortion. The entire algorithm is outlined in Algorithm 5. The range of radii is a chosen parameter of this algorithm; empirically, we have found that including the median of the pairwise distances in the observable space yields accurate estimations.

Performing a grid search might be computationally cumbersome. Moreover, the achieved performance strictly depends on the grid resolution. Formulating this problem as a minimization problem of the proposed criterion might enable more efficient methods for finding the optimal radius. This exceeds the scope of this work and will be examined in future work.

Algorithm 5 Adaptive covariance estimation

Input: Time-arranged measurements of a diffusion process $\{\boldsymbol{y_{t_i}}\}_{i=1}^N$ and range of radii, R's: $[R_1, R_2]$.
Output: The estimated covariance matrix around a specific point $\boldsymbol{y_{t_0}}$.

1. For each $R \in [R_1, R_2]$:
   a) Pick the set of datapoints that resides inside $\mathcal{B}_R(\boldsymbol{y_0})$ and calculate their distances from $\boldsymbol{y_0}$.
   b) Split $\mathcal{B}_R(\boldsymbol{y_0})$ to estimation set, $\mathcal{B}_{R^*}(\boldsymbol{y_0})$, and validation set, $\mathcal{B}_R(\boldsymbol{y_0})\backslash\mathcal{B}_{R^*}(\boldsymbol{y_0})$.
   c) Calculate $\widehat{C_e}(\boldsymbol{y_0})$ by applying (5.14) on the estimation set.
   d) Based on the eigen-value decomposition of $\widehat{C_e}(\boldsymbol{y_0})$, calculate the rescale matrix: $\mathbf{Q}$.
   e) Calculate $\widehat{C_v}(\boldsymbol{y_0})$ by applying (5.14) on the estimation set.
   f) Calculate $\widehat{Q_v}(\boldsymbol{y_0})$ and the obtained distortion for the current R using the frobenius norm of $\widehat{Q_v}(\boldsymbol{y_0}) - \mathbf{I}$.
2. Find the radius that achieved minimum distortion, we will denote it by R^{opt}.
3. The covariance estimation at $\boldsymbol{y_0}$ is obtained by applying (5.14) on $\mathcal{B}_{R^{opt}}(\boldsymbol{y_0})$.

5.3. Application to multi-scale SDE reduction

In this section we demonstrate the performance of Algorithm 5 by applying it to the simulated problem presented in [25]. We consider the following two dimensional Itô's drift-diffusion process, defined by the following SDE:

$$\begin{aligned} dx_t^1 &= adt + d\omega_t^1 \\ dx_t^2 &= \tfrac{adt}{\varepsilon} + \tfrac{1}{\varepsilon} d\omega_t^2 \end{aligned} \tag{5.15}$$

Where $\varepsilon \ll 1$ is a small scaling parameter. In this example x^1 is a slow variable and x^2 is a fast (nuisance) variable. This process is measured by a measurement function, denoted by f. In order to recover the slow variable from the data we will use the Mahalanobis distance, which measures distances normalized by the respective variance in each direction. We will demonstrate this claim by applying the Mahalanobis distance defined in (5.1) on the variables defined in (5.15). Specifically, in (5.15) the inverse of the covariance matrix does not depend on the coordinates of x, and is given by:

$$\mathbf{C}_{\mathbf{x}}^{-\mathbf{1}} = \begin{bmatrix} 1 & 0 \\ 0 & \varepsilon \end{bmatrix}, \tag{5.16}$$

and the Mahalanobis distance between two samples is:

$$\|\boldsymbol{x_{t_2}} - \boldsymbol{x_{t_1}}\|_M^2 = (x_{t_2}^1 - x_{t_1}^1)^2 + \varepsilon \cdot (x_{t_2}^2 - x_{t_1}^2)^2 = \|\boldsymbol{z_{t_2}} - \boldsymbol{z_{t_1}}\|_2^2 \tag{5.17}$$

Where $\boldsymbol{z}_t$ is a stochastic process of the same dimension of $\boldsymbol{x}_t$, in which the slow variables are maintained and the fast variables are attenuated by $\sqrt{\varepsilon}$:

$$z_t^1 = x_t^1 \tag{5.18}$$

$$z_t^2 = \sqrt{\varepsilon} \cdot x_t^2 \tag{5.19}$$

This rescaling allows us to use the Euclidean distance between samples of the rescaled variables $\boldsymbol{z}_t$ and provides a distance measure in which the slow variables are also pronounced. In our case, the system's variables $\boldsymbol{x}_t$ are inaccessible, instead we observe its mapping through f: $\boldsymbol{y_t} = f(\boldsymbol{x_t})$. As we have already shown in 5.1, applying the modified definition of Mahalanobis in (5.10) to the observables $\boldsymbol{y_t}$, provides the same distance measure, in which the fast variable collapses. We calculate the Mahalanobis distance using the proposed method and obtain a parametrization for the slow variable. In order to evaluate the performance of the proposed method we compare the calculated embedding with the hidden process x^1. The comparison is done via two criteria: correlation and normalized MSE (NMSE), where the NMSE between $\boldsymbol{x_t}$ and and $\hat{\boldsymbol{x}}_{\boldsymbol{t}}$ is defined as follows:

$$\text{NMSE} = \frac{\int (x_t^1 - \hat{x_t^1})^2 dt}{\int (x_t^1)^2 dt \int (\hat{x_t^1})^2 dt} \tag{5.20}$$

We compare these scores to the scores that would have been achieved using arbitrary time-window estimation, for different window durations. In order to obtain an upper

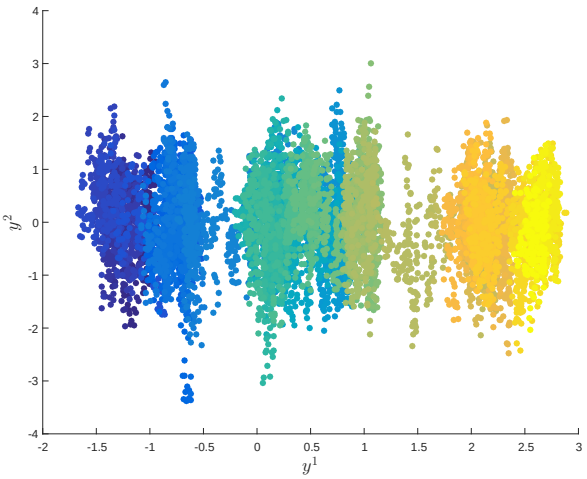


Figure 5.2.: Simulated datapoints from (5.15) with $a = 1$, $\varepsilon = 10^{-3}$ sampled by sampling frequency of $Fs = 10KHz$.

bound for the best performance within the framework of diffusion-maps we consider the case in which the covariance matrix is known.

In the first example the measurement function f is the identity function. The measured datapoints are depicted in Figure 5.2. We compare the achieved scores for various values of window length, Figure 5.3 illustrates the time span of the windows used to estimate the local covariance matrix (indicated by the red marks). The achieved scores

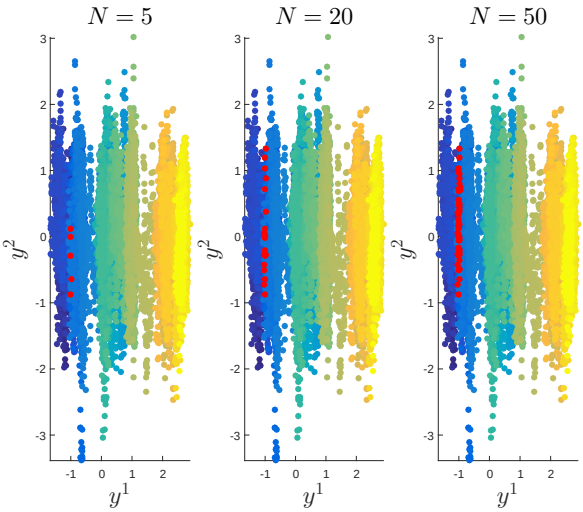


Figure 5.3.: Illustration for some of the time-windows used for the covariance estimation. The datapoints used for estimating the covariance in each window are colored in red. The used time-window lengths in (a),(b) and (c) are 5,20 and 50 samples respectively.

are depicted in Figure 5.4. It can be seen that using window-lengths which are greater than $N \simeq 10$ the estimation within an arbitrary time-window achieves comparable accu-

racy to the case where the covariance is known. We can also observe that the proposed method manages to obtain these maximal scores as well. In the second example in [25]

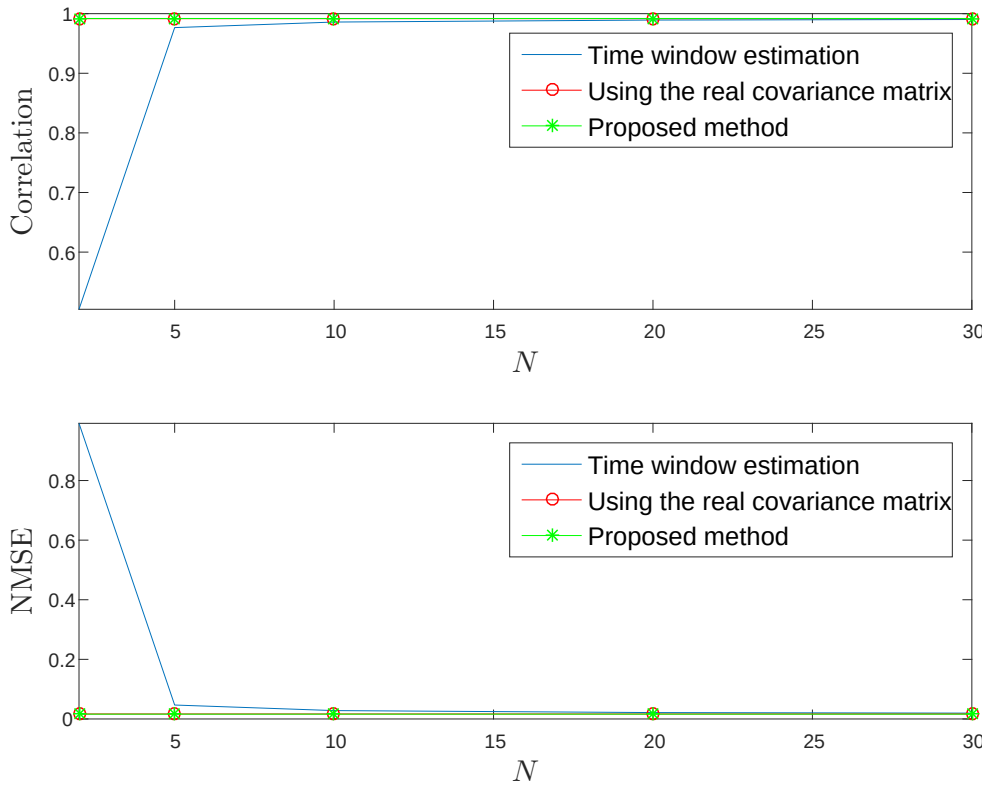


Figure 5.4.: The achieved correlations and normalized MSEs for different methods for calculating the Mahalanobis distance. The achieved scores for different time window-lengths are marked by the blue line. The obtained scores when the covariance matrix is known is marked by the vertical red line. The achieved scored by the proposed method are marked by the vertical green line.

the measurement function f is the following:

$$\begin{bmatrix} y_t^1 \\ y_t^2 \end{bmatrix} = f(\boldsymbol{x_t}) = \begin{bmatrix} x_t^1 + (x_t^2)^2 \\ x_t^2 \end{bmatrix}$$

The measured datapoints are depicted in Figure 5.5. The time spans of the windows used to estimate the local covariance matrix are depicted in Figure 5.6. The achieved scores are depicted in Figure 5.7. In contrast to the previous case, here we can see that increasing the window-length results with large errors (induced by the curvature of f) . We observe that the proposed algorithm obtains the same scores as the optimal window-length.

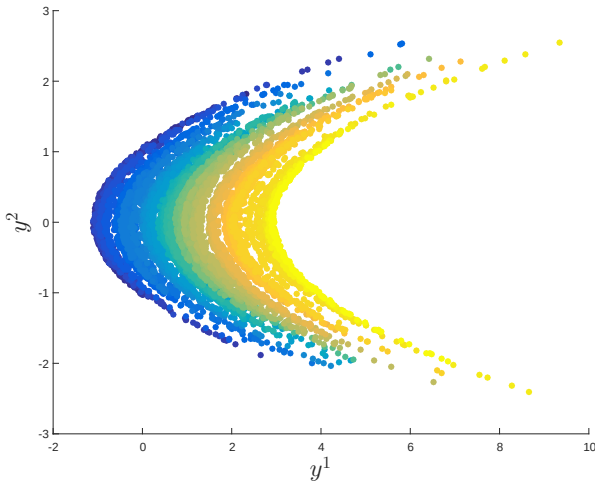


Figure 5.5.: Simulated datapoints from (5.15) with $a = 1$, $\varepsilon = 10^{-3}$ sampled by sampling frequency of $Fs = 10Khz$.

5.4. Error analysis of the covariance estimation within a time-window

For simplicity, the analysis in this section is focused a one dimensional process. Consider the one-dimensional diffusion process defined in (B.1). We denote the increments of the diffusion process in the observable space by:

$$dy_{t_i} = y_{(i+1)\Delta} - y_{i\Delta}$$

where $\frac{1}{\Delta}$ is the sampling frequency. The covariance estimator at x_0 is :

$$\widehat{f'}(x_0)^2 = \mu_{dy^2_{t_0}} - \left(\mu_{dy_{t_0}}\right)^2$$

where $\mu_{(.)}$ is the empirical mean defined in (5.13).

In [25] a detailed analysis of the estimation errors related to the Mahalanobis distance is presented. Two errors are considered, the calculation of the errors is based on the assumption that infinite number of realizations ($N \to \infty$) are available. The first error is defined as the error incurred by using Mahalanobis distance to approximate the true distance:

$$E_M(y_{t_1}, y_{t_2}) \triangleq \|x_{t_2} - x_{t_1}\|_2^2 - \|y_{t_2} - y_{t_1}\|_M^2 \tag{5.21}$$

Where x_{t_i} are the datapoints in the intrinsic space, y_{t_i} are the datapoints in the embedded space and $\|y_{t_1} - y_{t_2}\|_M$ is the modified Mahalanobis distance defined in (5.10). This error incurs because the Mahalanobis distance assumes linearity, an assumption that can be made only locally. The second error described in [25] is caused due to the dynamics. This error results from the drift of the stochastic process; this drift colors the gaussian distribution of the stochastic process steps and distorts the covariance matrix estimation. It is defined as:

$$E_C(y_t) = \widehat{C}(y_t, \delta t) - C(y_t) \tag{5.22}$$

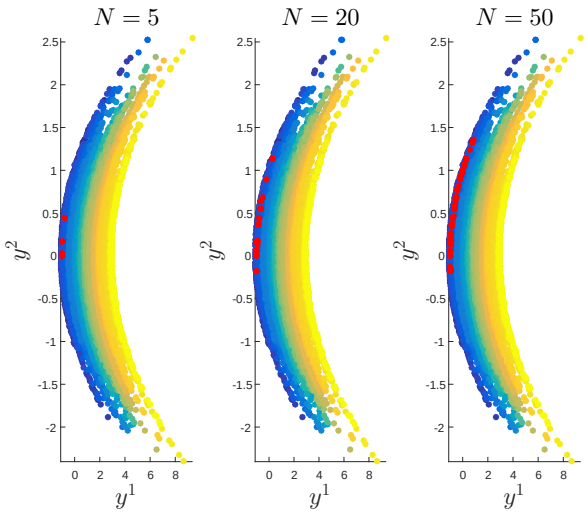


Figure 5.6.: Illustration for some of the time-windows used for the covariance estimation. The datapoints used for estimating the covariance in each window are colored in red. The used time-window lengths in (a),(b) and (c) are 5,20 and 50 samples respectively

Where:

$$\widehat{C}(y_t, \delta t) = \frac{1}{\delta t}\left[E(y^2_{t+\delta t}|y_t) - E(y_{t+\delta t}|y_t)^2\right] \tag{5.23}$$

These two errors are calculated analytically in [25].

The two errors described in [25] assume infinite amount of datapoints within time interval of δt, whereas in practice, only finite amount of samples are given. Due to the finite window length an additional estimation error incurs, which consists of two independent sources. The first source of error is the sample variance. Assuming that the samples within the window are N realizations of trajectories starting from x_0, the expression for this error is:

$$\sigma_{s.v} = f'(x_0)^2\sqrt{\frac{2}{N-1}}.$$

It practice we do not have N realizations, we have only one realization at x_0. The other $N-1$ samples are realizations taken from the neighborhood of x_0, assuming that the changes of the first derivative of f are relatively small. Unfortunately, this assumption does not always hold; this leads us to the second source of error - the variation of the covariance within the window due to f. The variability of the covariance within the window depends on the higher derivatives of the measurement function, and on the diffusion process parameters. The total error is:

$$\sigma^2 = E\big(\widehat{f}'(x_0)^2 - f'(x_0)^2\big)^2 = E\Big(\mu_{dy^2_{t_0}} - \big(\mu_{dy_{t_0}}\big)^2 - f'(x_0)^2\Big)^2 \tag{5.24}$$

The calculation of a closed form expression of the latter error is difficult. Nevertheless, using Itô's lemma and linear approximations for f, calculating the constant error

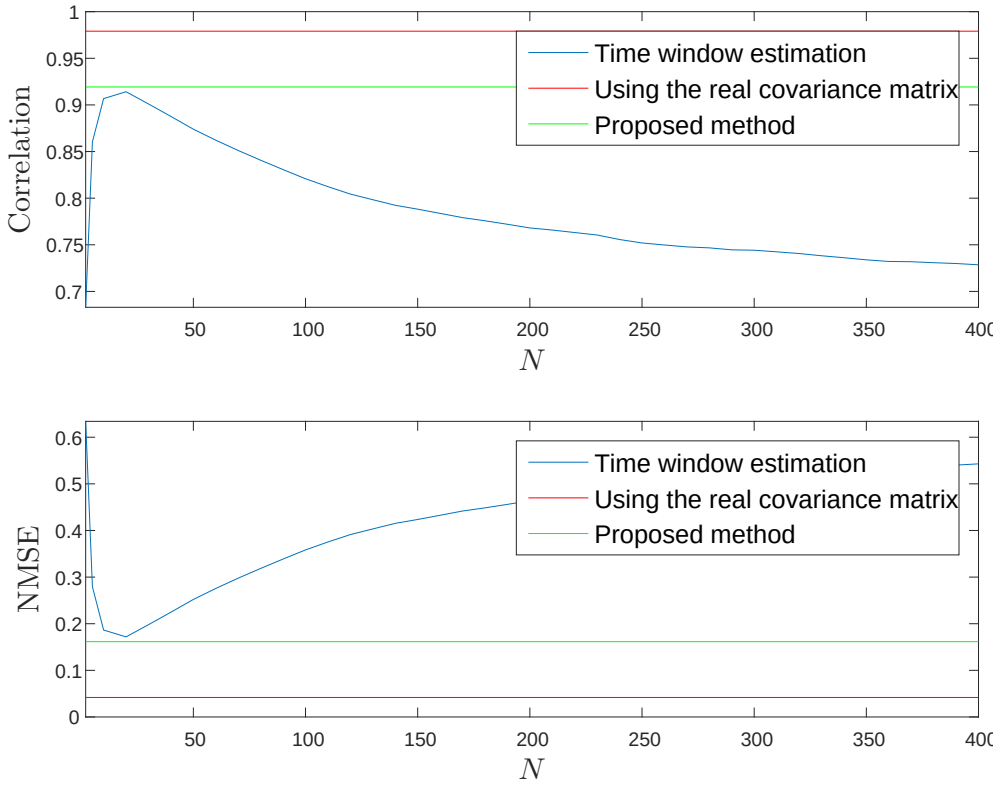


Figure 5.7.: The achieved correlations and normalized MSEs for different methods for calculating the Mahalanobis distance. The achieved scores for different time window-lengths are marked by the blue line. The obtained scores when the covariance matrix is known is marked by the vertical red line. The achieved scored by the proposed method are marked by the vertical green line.

resulting from this estimator is feasible:

$$\begin{aligned} E\big(\widehat{f'(x_0)^2}\big) =& \Big(f'(x_0)^2 + \big(f'(x_0)f''(x_0)a + \frac{1}{2}f''(x_0)^2 b\big)\Delta + \frac{2}{3}f''(x_0)^2(a\Delta)^2 - \frac{1}{12}f''(x_0)^2(a\Delta)^4\Big) \\ &+ T\Big(\big(\frac{1}{2}f'(x_0)f''(x_0)a + \frac{1}{4}f''(x_0)^2 b\big) + \frac{1}{2}f''(x_0)^2 a^2\Delta\Big) \\ &+ T^2\Big(\frac{1}{2}f''(x_0)^2 a^2 + \frac{1}{12}f''(x_0)^2 a^4\Delta^2\Big) \end{aligned} \tag{5.25}$$

where T is the window's duration, defined by $T = N\Delta$. The derivation for the expression in (5.25) is brought in Appendix A. As can be expected in large windows, a constant error incurs when the measurement function is curved or when the drift is relatively large. The constant error in (5.25) incurs due to the curvature of the measurement function. This assumption can be done locally for close enough datapoints (where the drift a or the window's time-interval T are relatively small compared to the curvature of f) but results with distortions as the distance increases.

Although this error and the error introduced in [25] are both related to the curvature of the measurement function, it should be noted that they are different. The error in (5.21) is the error caused by approximating the distance using Mahalanobis with linear model. In contrast to the error in (5.24), this error assumes that the covariance matrix is known (i.e. it is estimated using infinite realizations of the process), and does not relate to the estimation error of the covariance matrix. The error introduced in (5.24) results from the estimation of the covariance matrix itself, which might be deformed since we have only one realization at every point, and the covariance at each point is slightly different due to the curvature of the measurement function f. By taking $N \to \infty$ and the time interval as $\Delta = \frac{\delta t}{N}$ and substituting in the expression in (5.25), we can derive the same expression calculated in [25].

6. Conclusions and Future Work

In this research we have addressed two problems in the field of manifold learning. The first problem was introduced in Chapter 3. We presented the problem of information fusion from multiple, multimodal sensors. The primary focus is on a setting in which all sensors observe the same system, but each introduces different variables – some are related to various aspects of the system of interest, whereas others are sensor-specific and irrelevant. We presented a nonlinear data fusion scheme for suppressing the sensor-specific variables while preserving the system variables measured by two or more sensors. Experimental results demonstrate the applicability of our method to artificial toy problem and to recorded multimodal data for the purpose of sleep stage assessment.

A core element in the proposed scheme in Chapter 3, and in other manifold learning algorithms designed for multimodal data, is the calculation of the Mahalanobis distance, which requires estimation of local covariance matrices. In Chapter 5 we presented the problem of estimating the covariance matrix based on data sampled from multi-scale stochastic dynamical systems. We reviewed the errors analyzed in [25] and addressed another cause for error due to the finite sampling of the data. We demonstrated and exemplified the inherent trade off between preserving locality and minimizing the sample-variance error and proposed a heuristic method for dealing with this trade off. We demonstrated the performance of the proposed method on simulated data from [25].

The work presented in this thesis lay the foundation to a number of research directions that can be further investigated. For example, the core of the presented technique in Chapter 3 is the implementation of an abstract notion of the intersection and union of multimodal data sets. While the intersection between two sets is well defined and theoretically explained [15], the union of two (or more) sets still calls for rigorous analysis. Future work will include such analysis as well as development of a union scheme that respects uniqueness. Another research direction that stems from the work in Chapter 3 is related to the use of the intersection and union notions. In Chapter 3 we utilized the proposed intersection and union schemes in order to construct a method that extracts the common sources of variabilities from multiple sets of observations. Another use that can be accomplished by utilizing these schemes is getting an assessment for the relevance and quality of a single observation, for the purpose of measuring a certain phenomenon. This can be expressed in a more generalized way; using the proposed schemes we can assess the correspondence between two sets of observations based on the "strength" of the common component between them. Deriving a method for correspondence measure between two sets of observation can be done in a future work. The last research direction is broader and related to the main task addressed in Chapter 3. The proposed non-linear filtering scheme fuses information from various observations, and extracts the sources of variability which are related to a certain phenomenon of interest. This phenomenon of

interest is defined as the collection of all the sources of variability which are captured by two or more sensors. This definition is empirically supported and motivated by the fact that noises are more likely to be related to a single observation, and would not be captured by multiple observations. However, if a certain noise is observed by two or more observation it would not be filtered out. Moreover, if a single aspect of our system of interest is measured only by a single observations, it would be filter out. Noting that our proposed scheme is basically a graph, whose nodes are the observations, the edges are the intersections and the last layer is the union scheme, we can generalize it into more complex topological structure that will enable us to modify and clarify the desirable characteristic of the phenomenon of interest. For example, utilizing extended schemes that extracts intersections of three or more sensors. Moreover, by implementing those graphs as a weighted graphs we can turn this idea into a supervised scheme, aiming to measure a certain phenomenon.

Appendices

Appendices

A. Derivation of (5.25)

Consider the diffusion process in (B.1) :

$$dx_t = adt + \sqrt{b}d\omega_t$$
$$\rightarrow x_t - x_0 = at + \sqrt{b}\int_0^t d\omega_\tau$$

Measured by a bilipschitz function $y_t = f(x_t)$. According to Itô's lemma:

$$dy_t = \big(f'(x_t) + \tfrac{1}{2}bf''(x_t)\big) + f'(x_t)\sqrt{b}d\omega_t$$
$$\rightarrow y_t - y_0 = \int_0^t \big(f'(x_\tau) + \tfrac{1}{2}bf''(x_\tau)\big)d\tau + \int_0^t f'(x_\tau)\sqrt{b}d\omega_\tau$$

where Δ is the sampling time interval. Using the linear approximation of $f(x)$:

$$\begin{aligned} f'(x_t) &\backsimeq f'(x_0) + (x_t - x_0)f''(x_0) \\ &= f'(x_0) + \big(at + \sqrt{b}\int_0^t d\omega_\tau\big)f''(x_0) \\ &= f'(x_0) + af''(x_0)t + \sqrt{b}f''(x_0)\int_0^t d\omega_s \end{aligned}$$

$$\begin{aligned} f''(x_t) &\backsimeq f''(x_0) + (x_t - x_0)f^{(3)}(x_0) \\ &= f''(x_0) + \big(at + \sqrt{b}\int_0^t d\omega_\tau\big)f^{(3)}(x_0) \\ &= f''(x_0) + af^{(3)}(x_0)t + \sqrt{b}f^{(3)}(x_0)\int_0^t d\omega_\tau \end{aligned}$$

Substituting these in the expression for dy_{t_i}:

$$\begin{aligned} dy_{t_i} = &\int_{i\Delta}^{(i+1)\Delta} \big(af'(x_0) + a^2f''(x_0)t + af''(x_0)\sqrt{b}\int_0^t d\omega_\tau\big)dt \\ &+ \int_{i\Delta}^{(i+1)\Delta} \frac{1}{2}\big(bf''(x_0) + abf^{(3)}(x_0)t + b\sqrt{b}f^{(3)}(x_0)\int_0^t d\omega_\tau\big)dt \\ &+ \int_{i\Delta}^{(i+1)\Delta} \sqrt{b}\big(f'(x_0) + af''(x_0)\big)d\omega_t + \int_{i\Delta}^{(i+1)\Delta} \big(bf''(x_0)\int_0^t d\omega_\tau\big)d\omega_t \end{aligned}$$

After some simplifications:

$$\begin{aligned} dy_{t_i} &= \Delta\Big(af'(x_0) + \frac{1}{2}f''(x_0) + \Delta\big(i+\frac{1}{2}\big)\big(a^2f''(x_0) + \frac{1}{2}abf^{(3)}(x_0)\big)\Big) \\ &+\sqrt{b}\big(f''(x_0) + \frac{b}{2}f^{(3)}(x_0)\big)\int_{i\Delta}^{(i+1)\Delta}\int_{0}^{t} d\omega_\tau d\omega_t \\ &+ \int_{i\Delta}^{(i+1)\Delta} \sqrt{b}\big(f'(x_0) + af''(x_0)\big)d\omega_t \end{aligned}$$

The expression for the covariance estimator at x_0 is:

$$\widehat{f'}(x_{t_i})^2 = \overline{dy_{t_i}{}^2} - \big(\overline{dy_{t_i}}\big)^2 \tag{A.1}$$

Substituting and calculating the mean, provides the following expression:

$$\begin{aligned} E\big(\widehat{f'}(x_{t_i})^2\big) =& \Big(f'(x_{t_i})^2 + \big(f'(x_{t_i})f''(x_{t_i})a + \frac{1}{2}f''(x_{t_i})^2b\big)\Delta + \frac{2}{3}f''(x_{t_i})^2(a\Delta)^2 - \frac{1}{12}f''(x_{t_i})^2(a\Delta)^4\Big) \\ &+ N\Big(\big(\frac{1}{2}f'(x_{t_i})f''(x_{t_i})a + \frac{1}{4}f''(x_{t_i})^2b\big)\Delta + \frac{1}{2}f''(x_{t_i})^2(a\Delta)^2\Big) \\ &+ N^2\Big(\frac{1}{2}f''(x_{t_i})^2(a\Delta)^2 + \frac{1}{12}f''(x_{t_i})^2(a\Delta)^4\Big) \end{aligned} \tag{A.2}$$

B. On the importance of a proper choice of the window-length

The influence of the chosen window-length on different aspects of a manifold learning technique is not direct and depends on other parameters in the algorithm as well. The choose of the window-length affects the estimation of the covariance matrix, the estimation of the covariance matrix affects the estimation of the pairwise affinity matrix, which finally affects the calculation of the manifold's kernel. In the first step we will demonstrate the influence of the window-length on the pairwise affinities estimation, as shown in B. In the next step we evaluate the window-length influence on the manifold's kernel estimation, as shown in B. Finally, in 3.4 will discuss what considerations should be taken when choosing the window-length. For simplicity, we will consider one-dimensional diffusion process defined by the following SDE:

$$dx_t = adt + \sqrt{b}d\omega_t \tag{B.1}$$

Measured by a non-linear bilipchitz function $y_t = f(x_t)$. As mentioned in 5, a proper choose of the window-length parameter is acute in two scenarios. When measuring a drift-based process even with a slightly curved function, or when measuring a brownian based process with a curved function. In order to witness the influence of the curvature of the functions we will demonstrate two measurement functions with different curvatures.

Influence on the pairwise affinities estimation

In this subsection we wish to demonstrate the influence of the covariance matrix deformation on the estimation of the pairwise affinities when using the Mahalanobis distance. Remind the expression for Mahalanobis distance between y_{t_1} and y_{t_2}:

$$\|y_{t_1} - y_{t_2}\|_M^2 = \frac{1}{2}\big(y_{t_1} - y_{t_2}\big)^T \left(C^T\big(y_{t_1}\big) + C^T\big(y_{t_2}\big) \right)^{-1} \big(y_{t_1} - y_{t_2}\big) \tag{B.2}$$

We would like to examine the deformation between $\|y_{t_1} - y_{t_2}\|_M$ and the distance in the intrinsic space $\|x_{t_1} - x_{t_2}\|$. Since that manifold learning algorithms use the pairwise distances estimations locally (within a small neighborhood) the performance evaluation of this estimation should be done by a criteria that respects the local neighborhood of a certain data-point. We will consider this local neighborhood as a ball with radius ε, and the proposed criteria are calculated within this ball. The first criterion will be denoted by MD(ε). It indicates how many points will be mistakenly neglect by taking only the points within the radius of ε. Meaning how many points, x_t, resides inside a ball of radius ε: $\|x_t - x_{t_0}\| < \varepsilon$ but estimated mistakenly to be outside $\|y_t - y_{t_0}\|_M > \varepsilon$. The second criterion will be denoted by MSE(ε). It indicates the mean square error between the estimated distances using Mahalanobis distance $\|y_t - y_{t_0}\|_M$ and the intrinsic distance $\|x_t - x_{t_0}\|$, calculated only for points that resides within a ball of radius ε from x_{t_0}. The last criterion is CORR(ε). It indicates the correlation between the calculated Mahalanobis distances $\|y_t - y_{t_0}\|_M$ and the real distances $\|x_t - x_{t_0}\|$, calculated only for points that resides within a ball of radius ε from x_{t_0}. In order to evaluate those criteria we will perform a simulation. Using this simulations we would be able to demonstrate the importance of choosing a proper window-length. In order to evaluate the quality of the obtained results we would like to bound it by the best and worst achievable performances within the framework of the Mahalanobis distance. The higher bound for the performances is the case where the the first derivative is known. The lower bound for the performances is not well defined, we will consider it to the case where almost no information about the first derivative is known and it is "estimated" by randomly choosing a value between $[0, 2f'(x_t)]$. We simulate a diffusion process x_t starting from $x_0 = 0$ with a constant drift step of $a = 1$ and constant energy coefficient $b = 1$, sampled by sampling frequency of $Fs = 10KHz$. We use two measurement functions, a "slightly" curved one: $f_1(x) = 1 + 5x + x^2$, and a curved one: $f_2(x) = 1 + 5x + 10x^2$. The optimal window-lengths for f_1 and f_2 are 200 and 35, respectively. The simulation steps are as described in 6. The obtained results where averaged $10,000$ times.

Algorithm 6 Simulation for evluating pairwise affinities criteria

Input: Diffusion process parameters: drift (a), brownian energy coefficient (b), measurement function $\big(f(\cdot)\big)$ and sampling interval (dt).
Output: Pairwise affinities criteria $\big($MD(ε),MSE(ε) and CORR(ε)$\big)$.

1. Generate 100 points according to the given parameters $\big(a, b, dt, f(\cdot)\big)$.
2. Find the optimal window length (empirically, as done in B).
3. For each point calculate the first derivative of $f'(x_t)$ in the following ways:
   a) Using variance based estimation, with the optimal window length for each point.
   b) Using variance based estimation, with arbitrary length (e.g. $N = 10$).
   c) Using the real expression for the first derivation $f'(x_t)$. This expression will serve us for calculating the upper bound for the performance that can be achieved using the linear approximation of the Mahalanobis distance.
   d) A random value taken uniformly from the interval $[0, 2f'(x(t))]$. This expression will serve us for calculating the lower bound.
4. For each method ((a)-(d) in the previous step) calculate the Mahalanobis distance from each point to x_{t_0} using (B.2) and then calculate MD(ε),MSE(ε) and CORR(ε).

Figures B.1 and B.2 compares the calculated Mahalanobis distances with the real distances, for each of the chosen measurement functions f_1 and f_2. The comparison when using f_1 as the measurement function are depicted in Figure B.1. It can be seen that the use of the real derivative obtains a very good correlation between the Mahalanobis distance and the intrinsic distance, this result is not surprising since that the chosen measurement function is almost linear. Another interesting result that stems from the figure is the importance of a proper window-length chose, as can be seen, arbitrary window-length is almost equivalent to randomly drawing an estimation.

The comparison when using f_2 as the measurement function are depicted in Figure B.2. As can be seen, unlike in the slightly curved function, for this function even knowing the precise value of the derivative does not ensure good correlations, this fact results from the function's curvature. Another interesting, yet depressing, result that stems from Figure B.2 is that even when choosing the optimal window-length, the obtained correlations are close to the correlations that would have been achieved using arbitray window-length. But, it should be noted that for small enough distances the correlation between the estimated pairwise distances and the real distances improves. In the next subsection we will further discuss this issue.

The measured MD(ε), MSE(ε) and CORR(ε) appears in Figures B.3 and B.4 for f_1 and f_2 respectively. It can be seen that when measuring with f_2, which is particularly curved function, using non-optimal window-length (such as $N = 150$) is almost as worst

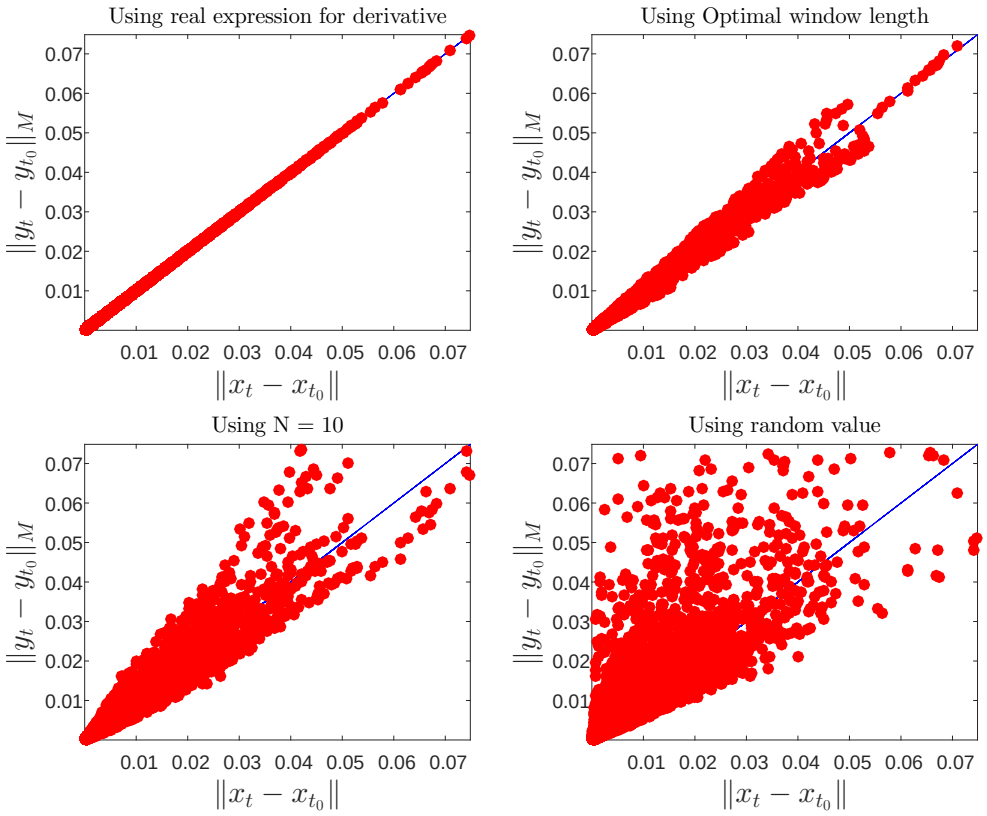


Figure B.1.: The calculated Mahalanobis distance verses the intrinsic distance from x_{t_0} using: (a) the real expression for the derivatives (b) estimated values obtained by using the optimal window-length (c) estimated values obtained by using the arbitrary window-length (d) random values drawn uniformly from $[0, 2f(x_t)]$.

as using random drawing as the estimation of the first derivative.

Influence on the manifold's kernel estimation

In the previous subsection we have demonstrated the influence of the window-length parameter on the ability to estimate the pairwise distances. In the framework of manifold learning this pairwise distances are then used for estimating the manifold's kernel. In this subsection we wish to demonstrate the influence of the window-length parameter on the estimated manifold's kernel. Given a certain spectrum matrix, estimated by calculating the pairwise distances using Mahalanobis distance with a certain window-length, we will examine the obtained estimated kernel. We will denote the spectrum matrix by $\mathbf{D}$, where the (i, j)th entry is the estimated pairwise distance, i.e. $D_{i,j} = \|y_{t_i} - y_{t_j}\|_M$. We will consider the case of graph-laplacian, i.e. the kernel is calculated via an affinity matrix, denoted by $\mathbf{W}$, which is build as in (2.1). Based on the calculated affinity matrix we then compute the diffusion operator, denoted by $\mathbf{K}$, as in (2.2). We repeat the simulation

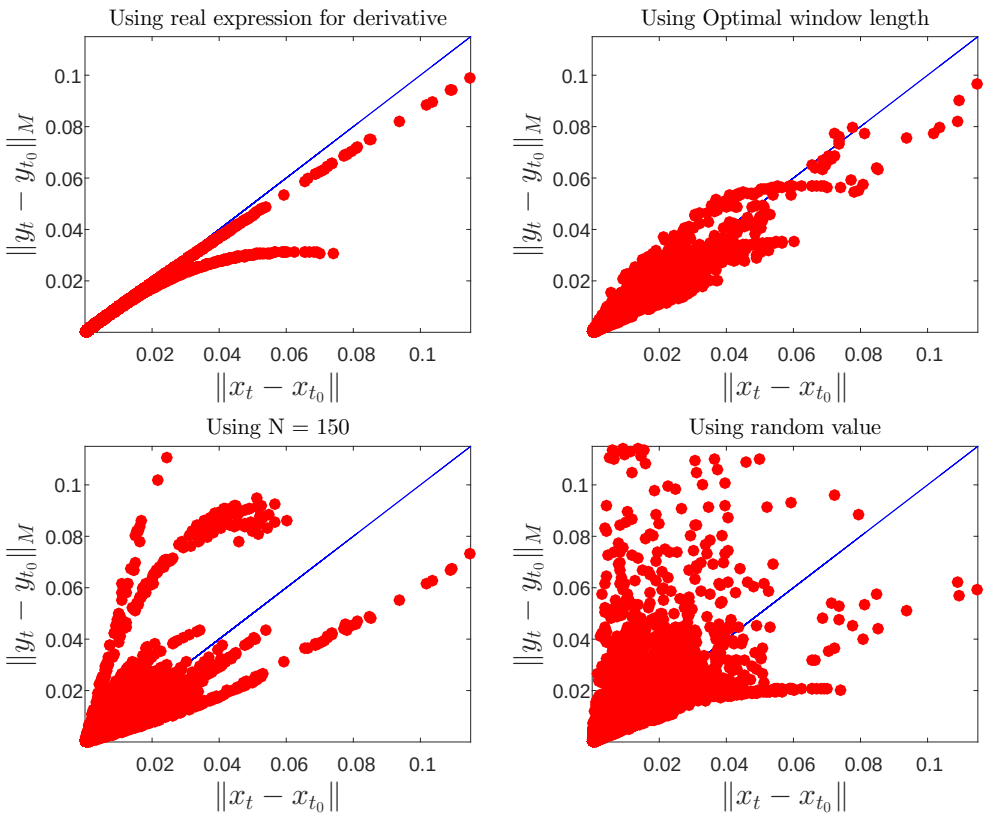


Figure B.2.: The calculated Mahalanobis distance verses the intrinsic distance from x_{t_0} using: (a) the real expression for the derivatives (b) estimated values obtained by using the optimal window-length (c) estimated values obtained by using the arbitrary window-length (d) random values drawn uniformly from $[0, 2f(x_t)]$.

described in the previous subsection, but this time, for each generated diffusion process we build the estimated manifold kernel. We compare the estimated kernel with the real kernel, i.e. the kernel that would have been calculated if we had access to the process's samples in the intrinsic space, we will denote this kernel by $\mathbf{K}^*$. Comparing two kernels is not well defined, we suggest two types of criteria for the kernels comparison. The first type is related to the fact that the kernel is a stochastic transitions matrix. The latter is related to the spectral decomposition of the matrix, which is relevant for application of dimensionality reduction. It should be noted that those criteria are directly depend on the chose of the kernel scale ε. Therefore we measure it for range of ε. Remind that the kernel is a stochastic transitions matrix, we can use statistical distances for measuring the distortion in $\mathbf{K}$ that occurred in a single data-point by taking the Bhattacharya distance between the corresponding row in $\mathbf{K}$ and the corresponding row in $\mathbf{K}^*$. The second criterion is related to the spectral decomposition of $\mathbf{K}$. We calculate the most significant eigen-vector $\mathbf{K}$ and $\mathbf{K}^*$ and compare them by using mean-square error and correlation. We perform a simulation with the same parameters taken in subsection B

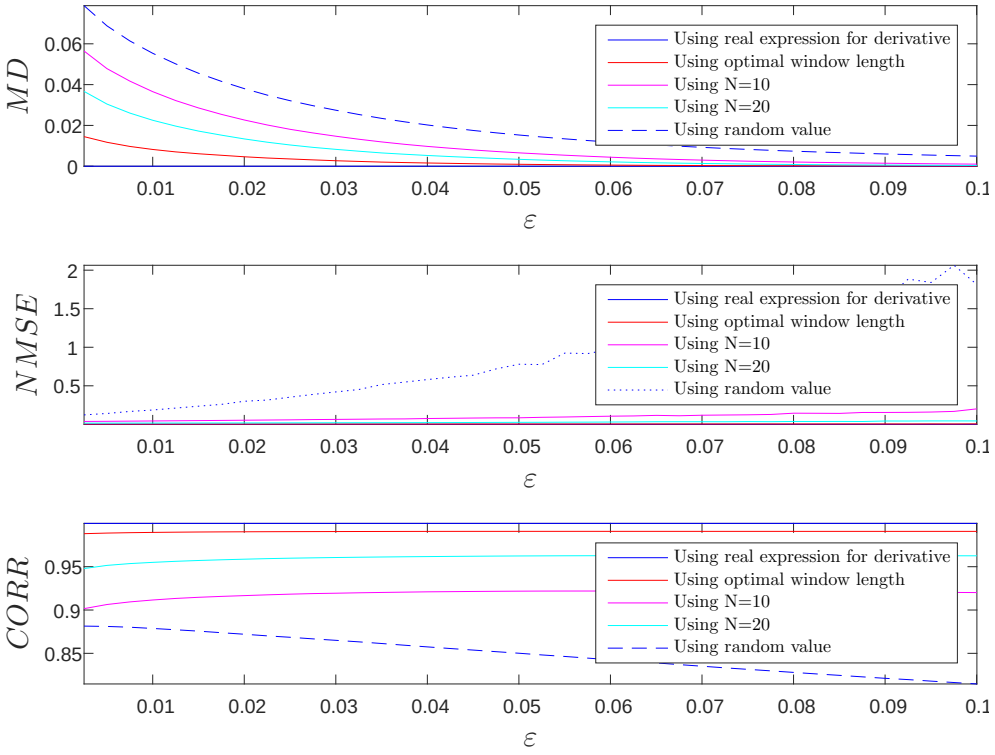


Figure B.3.: Measured MD(ε), MSE(ε) using: (a) the real expression for the derivatives (b) estimated values obtained by using the optimal window-length (c) estimated values obtained by using the arbitrary window-length (d) random values drawn uniformly from $[0, f(x_t)]$.

and take f_1 as the measurement function. In addition to the simulation steps described in 6, for each ε we calculate $D_b(\varepsilon)$,$D_{mse}(\varepsilon)$,$V_{mse}(\varepsilon)$ and $V_{corr}(\varepsilon)$. $D_b(\varepsilon)$ and $D_{mse}(\varepsilon)$ are obtained by averaging the distance (Bhattacharyya or $||\cdot||_2$, respectively) between rows of $\mathbf{K}^*$ and $\mathbf{K}$. $V_{mse}(\varepsilon)$ and $V_{corr}(\varepsilon)$ are obtained by averaging (the mean-square error or the correlation) between the most significant eigen-vector of $\mathbf{K}$ and the most significant eigen-vector of $\mathbf{K}^*$. We repeat the simulation 10 times and average the results. The results are depicted in Figures B.5 and B.6.

A common choice for ε is the median of the pairwise distances. This choice is made in order to ensure the connectivity of the graph. This value is marked by a horizontal red line in Figures B.5 and B.6. It should be noted that this value is inaccessible to us, and every application requires different tuning of this parameter. In Figure B.5 we can see the achieved $D_b(\varepsilon)$ and $D_{mse}(\varepsilon)$ for a range of ε:

As can be expected, the best kernel estimation is achieved when using the real value for the derivatives. When using the optimal window-length, or close enough window-length, we approach those performances. We should pay attention to an odd phenomenon that arises from Figure B.5 and might be misleading. In figure B.5 we can see that for large choose of ε the obtained Bhattacharyya distances are getting smaller. Allegedly, it seems that choosing large ε yields with good estimation for $\mathbf{K}^*$, but in practice this

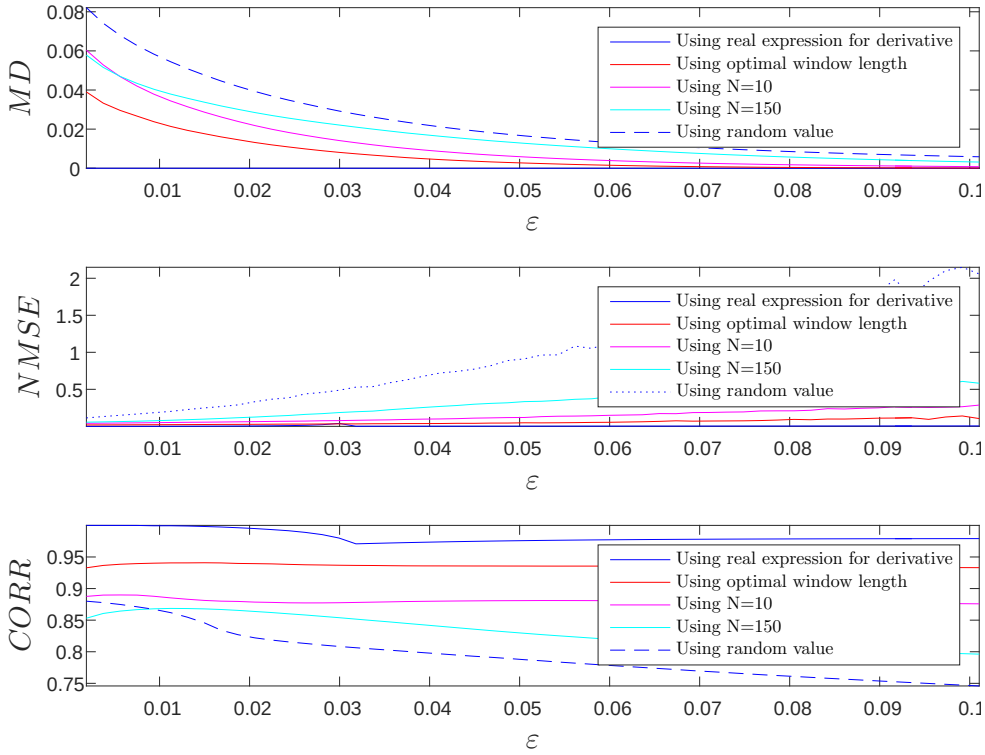


Figure B.4.: measured MD(ε), NMSE(ε) using: (a) the real expression for the derivatives (b) estimated values obtained by using the optimal window-length (c) estimated values obtained by using the arbitrary window-length (d) random values drawn uniformly from $[0, f(x_t)]$.

phenomenon results from the fact that for large ε the kernel matrix $\mathbf{K}^*$ converges to a uniform distribution kernel. Meaning that we easily managed to estimate $\mathbf{K}^*$, but $\mathbf{K}^*$ does not characterizes $\boldsymbol{x_t}$ any more. This argument also arises from figure B.6. In figure B.6 we can see the $V_{mse}(\varepsilon)$ and $V_{corr}(\varepsilon)$ for a range of ε:

We can see that even when using the real value for the first derivative an error is occurred due to the curvature of the function. As in figure B.5, also here small errors are obtained when using large ε. Another interesting insight that we can learn from figure B.6 is about the range of ε in which proper chose of the window-length is necessary. It can be seen that for large ε the performances are highly depend on the chose of the window-length, while that for small epsilons we get the same performance no matter what is the chosen window-length. The reason is that for small ε both $\mathbf{K}^*$ and $\mathbf{K}$ converges to the identity matrix.

Considerations for choosing the proper window-length

In the previous subsections we showed the influence of the window-length parameter on different aspects relevant to manifold learning techniques, and showed the motivation for choosing a proper window-length for the covariance estimation. In this subsection we

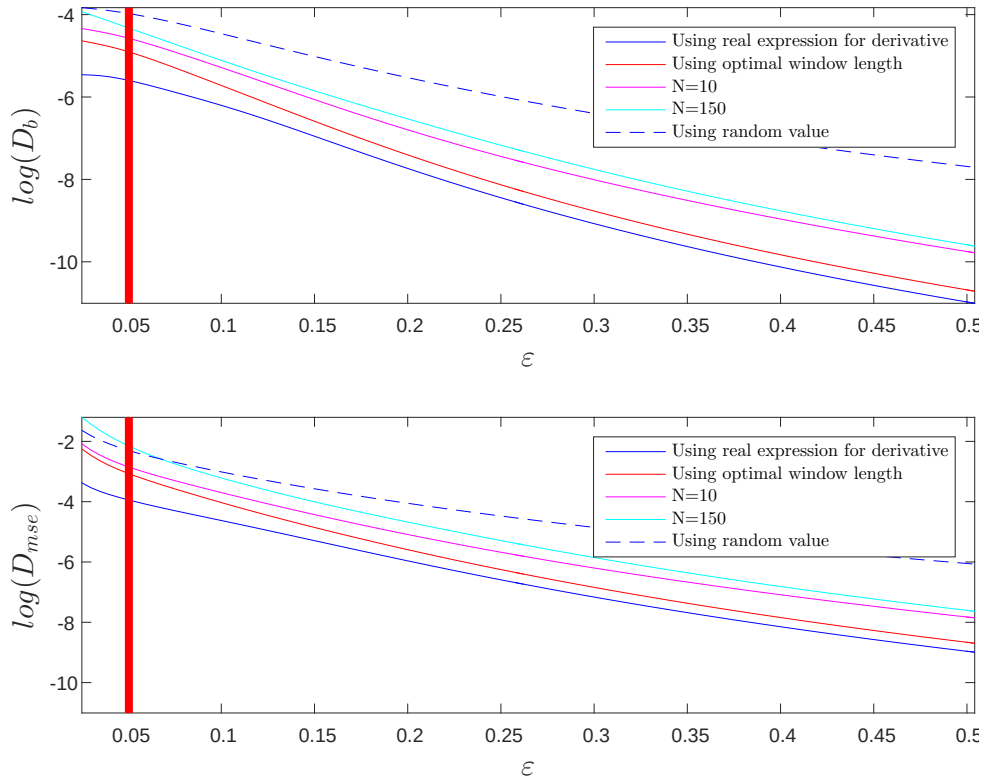


Figure B.5.: $D_{mse}(\varepsilon)$ and $D_b(\varepsilon)$. The vertical red line is the median value of the pairwise distances.

will discuss what considerations should be taken when choosing the window-length, we will demonstrate our claims by simulations. We consider the one-dimensional diffusion process defined (B.1). It is clear that when addressing the problem of finding the optimal window length we should pay attention to the following trade-off. On the one hand estimation of the covariance matrix should be done using a long window in order to reduce the sample variance error. On the other hand estimation of the covariance matrix using a too long window will increase the error results from the curvature of the measurement function. This trade-off is dominated by two factors: the "density" of the process and the curvature of the measurement function. A "dense" process, which we will denote by the name - brownian based process, provides relatively large amount of data-points, in that case, choosing large window-length enables us to overcome the sample-variance without suffering from the curvature of the measurement function. On the opposite side, a deterministic process driven mainly by the drift, which we will denote by the name - drift based process, does not encounter the errors caused by the sample variance, and performs well even when taking small window-lengths. In this case increasing the window length might expose the estimation to errors caused by the curvature of the measurement function. Meaning that the optimal window-length depends on the "density" of the process and the curvature of the function. However, in some cases those two factors, the "density" of the diffusion process and the curvature of the function,

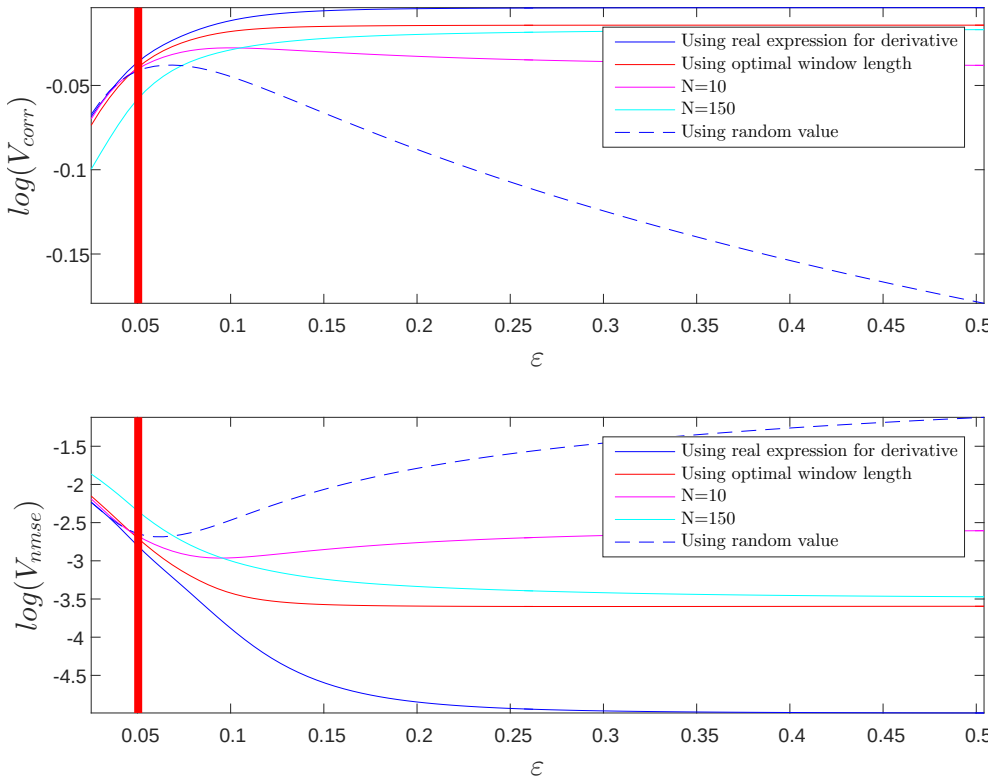


Figure B.6.: $V_{mse}(\varepsilon)$ and $V_{corr}(\varepsilon)$. The vertical red line is the median value of the pairwise distances.

might contradict. This trade-off will be particularly noticed in two cases. The first case is a brownian based process which is measured by a considerably curved function. The second case is a drift based processes measured by a slightly curved function. In order to distinguish between sparse and dense processes we will define α_d as the ratio between the average step size of the Brownian motion and the step size caused by the constant drift:

$$\alpha_d = \frac{\sqrt{bdt}}{adt} \tag{B.3}$$

A process for whom $\alpha_d \to 0$ will be denoted as a drift based process, a process for whom $\alpha_d \to \infty$ will be denoted as a brownian based process. The second factor will be simply defined by the ratio between the first and the second derivatives of the measurement function. We denote the first and the second derivatives at a certain point, x_0, with c_1 and c_2, respectively, such that:

$$f(x) = f(x_0) + c_1(x - x_0) + \frac{1}{2}c_2(x - x_0)^2$$

The curvature factor is defined by the ratio between c_1 and c_2:

$$\alpha_f = \frac{c_1}{c_2} \tag{B.4}$$

We demonstrate the above-mentioned trade off using simulated data. The simulations are carried out in 4 scenarios, obtained by combinations of various diffusion processes with different measurement functions. In the first example we consider a brownian based process ($\alpha_d = 100$) measured by a curved function ($\alpha_f = 1$). In the second example we simulate a the same settings as in the first example, but with higher drift, while maintaining the value α_d by increasing the energy of the Brownian motion. In the third example we simulate a drift based process ($\alpha_d = 0.1$) measured by a function with "linear" characteristics ($\alpha_f = 0.2$). In the last example we simulate a brownian based process ($\alpha_d = 50$) measured by a curved function ($\alpha_f = 1$). We simulate those scenarios and calculate the obtained errors for different sizes of window lengths. The results were averaged over 500 simulations. The simulation's parameters for each one of the scenarios are summarized in the following table:

	Example 1	Example 2	Example 3	Example 4
a	1	2	1	2
b	1	4	10^{-6}	1
c_1	5	5	5	5
c_2	5	5	1	5
α_d	100	100	0.1	50
α_f	1	1	0.2	1

In figure B.7 we can see realizations of the above-mentioned scenarios. It can be noticed that in the observed space, unlike in the intrinsic one, the properties of dy_t change during the evolution of the process. In the case of a brownian based process, the volatility of the variance is very noticeable, while that in the case of a drift-based process the changes are manifested by a constant slope. We define the normalized MSE (NMSE) of $\widehat{f'}(x_0)^2$

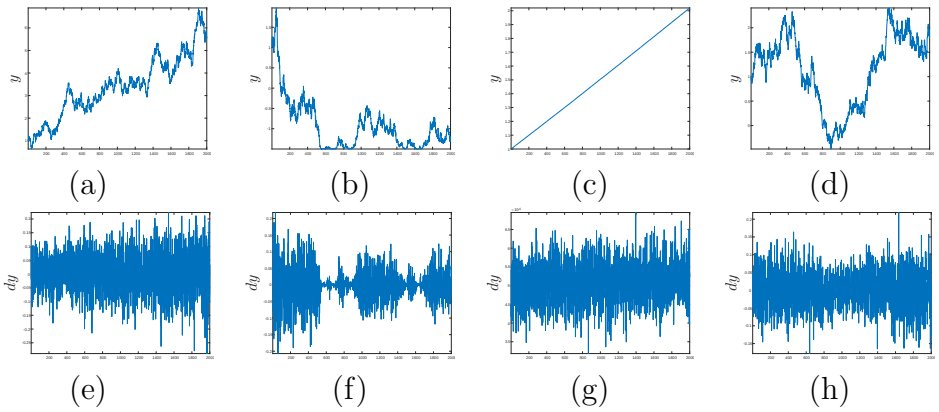


Figure B.7.: Process realizations. Plots (a)-(d) shows the diffusion processes in the embedded space (y) for examples 1-4 respectively. Plots (e)-(f) shows th discrete derivatives of the diffusion processes in the embedded space (dy) for examples 1-4 respectively.

in the following way:

$$\text{NMSE} = \frac{E\left(\widehat{f'}(x_0)^2 - f'(x_0)^2\right)^2}{f'(x_0)^4} \tag{B.5}$$

Since we have not managed to derive a close expression for the normalized MSE, this value was computed empirically by simulations. We evaluate the precision of two approximations for the normalized MSE. The first approximation is based on the sample variance MSE, defined in (5.24). The latter approximation, denoted as the drift based approximation, is based on the calculation of $E\left(\widehat{f'}(x_0)^2\right)$ in (5.25). We approximate the normalized MSE by replacing the term $E\left(\widehat{f'}(x_0)^2 - f'(x_0)^2\right)^2$ with $\left(E(\widehat{f'}(x_0)^2) - f'(x_0)^2\right)^2$. In order to evaluate the proposed approximation we plot the empirically calculated normalized MSEs along with its approximations, for various window lengths. The results are depicted in Figure B.8. As can be seen, the sample variance approximation holds for small window lengths. For large window-length neither the sample variance approximation nor the drift based approximation holds. The empirically calculated normalized

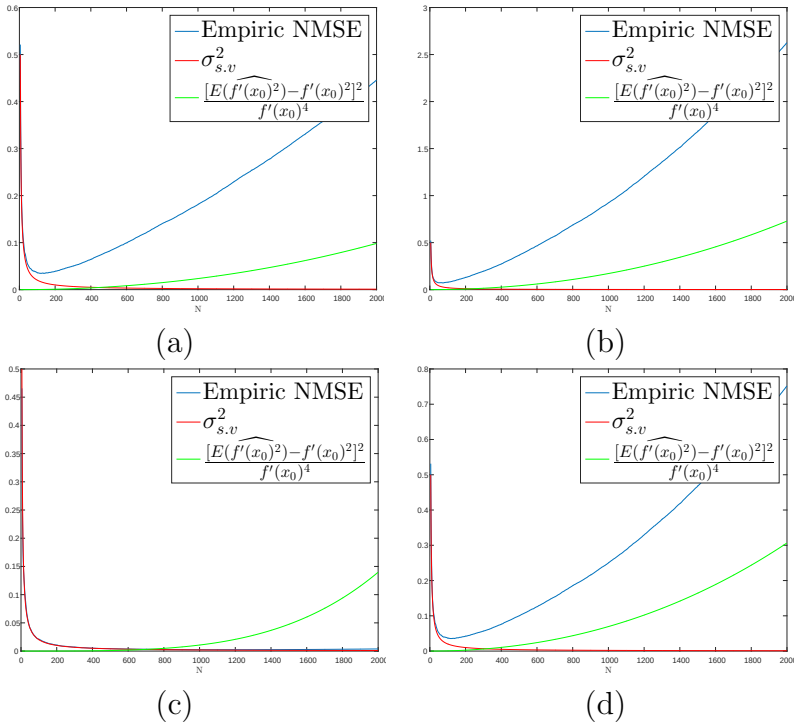


Figure B.8.: Normalized value of $f'(x_0)^2$ during the processes evolution. Plots (a)-(d) correspond to examples 1-4 respectively. The blue line is the empirically computed normalized MSE. The red line is the sample variance approximation, the green line is the drift based approximation.

MSEs along with its approximations, for various window lengths are depicted in Figure B.8. By comparing the plots in figure B.8 we can see that the drift based approximation holds for the drift-based process, while that for the brownian based processes it suffers from large errors. Another point that states from by comparing plots (a) and (b) is that

increasing the step size by 2 results with an optimal window length which is shorter by half. Some other measure that might be useful for future work is the estimator's variance. The estimator's variances for the above-mentioned scenarios are depicted in figure B.9. As can be seen in figure B.9 when using large windows, except of the constant error,

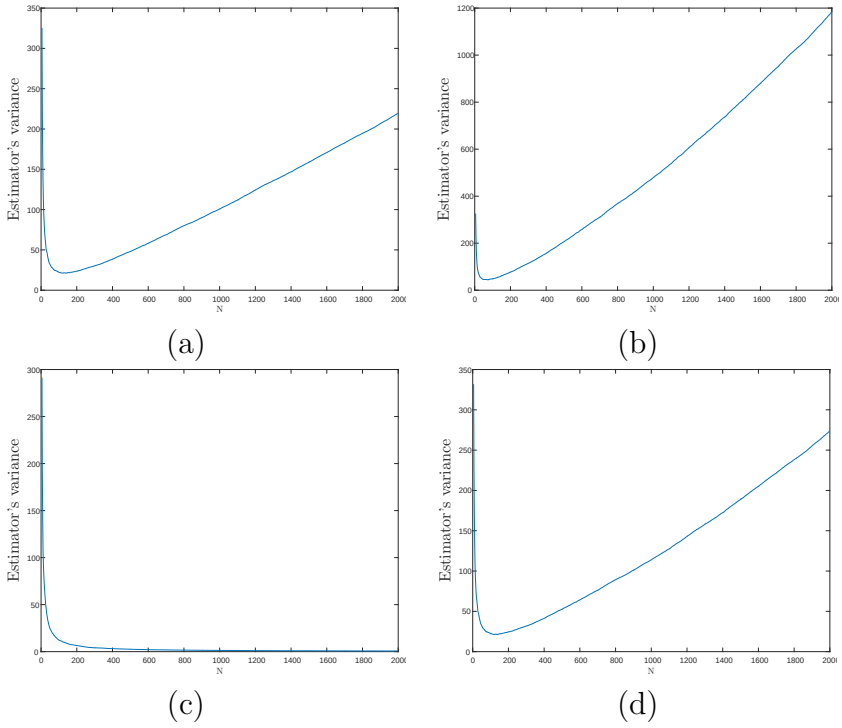


Figure B.9.: The estimator's variance during the process evolution. Plots (a)-(d) correspond to examples 1-4 respectively.

the estimator suffers from a large variance. This property characterizes all the examples which are based on the diffussion-based processes. In the case of the drift-based process the estimator's variance does not increases as the window-length gets larger, this can be expected due to the "deterministic" characteristics of the process.

Bibliography

[1] Bahador Khaleghi, Alaa Khamis, Fakhreddine O Karray, and Saiedeh N Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[2] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: An overview of methods, challenges, and prospects," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, 2015.

[3] Raffaele Gravina, Parastoo Alinia, Hassan Ghasemzadeh, and Giancarlo Fortino, "Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges," *Information Fusion*, vol. 35, pp. 68–80, 2017.

[4] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 260, pp. 2319–2323, 2000.

[5] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 260, pp. 2323–2326, 2000.

[6] D. L. Donoho and C. Grimes, "Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data," *Proc. Nat. Acad. Sci.*, vol. 100, pp. 5591–5596, 2003.

[7] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural. Comput.*, vol. 15, no. 6, pp. 1373–1396, June 2003.

[8] R. R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 5–30, 2006.

[9] M. Davenport, C. Hegde, M. F. Duarte, and R. G. Baraniuk, "Joint manifolds for data fusion," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2580–2594, 2010.

[10] Yosi Keller, Ronald Coifman, Stphane Lafon, and Steven W. Zucker, "Audio-visual group recognition using diffusion maps," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 403–413, Jan. 2010.

[11] Or Yair and Ronen Talmon, "Local canonical correlation analysis for nonlinear common variables discovery," *IEEE Transactions on Signal Processing*, vol. 65, no. 5, pp. 1101–1115, 2016.

[12] Moshe Salhov, Ofir Lindenbaum, Avi Silberschatz, Yoel Shkolnisky, and Amir Averbuch, "Multi-view kernel consensus for data analysis and signal processing," *arXiv preprint arXiv:1606.08819*, 2016.

[13] Ofir Lindenbaum, Arie Yeredor, Moshe Salhov, and Amir Averbuch, "Multiview diffusion maps," *arXiv preprint arXiv:1508.05550*, 2015.

[14] R. R. Lederman and R. Talmon, "Learning the geometry of common latent variables using alternating-diffusion," *Appl. Comp. Harmon. Anal.*, 2015.

[15] Ronen Talmon and Hau-tieng Wu, "Latent common manifold learning with alternating diffusion: analysis and applications," *arXiv preprint arXiv:1602.00078*, 2016.

[16] Virginia R de Sa, "Spectral clustering with two views," in *ICML workshop on learning with multiple views*, 2005.

[17] Virginia R. de Sa, Patrick W. Gallagher, Joshua M. Lewis, and Vicente L. Malave, "Multi-view kernel construction," *Machine Learning*, vol. 79, no. 1-2, pp. 47–71, May 2010.

[18] Byron Boots and Geoffrey J. Gordon, "Two-manifold problems with applications to nonlinear system identification," in *Proc. 29th Intl. Conf. on Machine Learning (ICML)*, 2012.

[19] Tomer Michaeli, Weiran Wang, and Karen Livescu, "Nonparametric canonical correlation analysis," *arXiv preprint arXiv:1511.04839*, 2015.

[20] Avishag Shemesh, Ronen Talmon, Ofer Karp, Idan Amir, Moshe Bar, and Yasha Jacob Grobman, "Affective response to architecture–investigating human reaction to spaces with different geometry," *Architectural Science Review*, pp. 1–10, 2016.

[21] Hau-tieng Wu, Ronen Talmon, and Yu-Lun Lo, "Assess sleep stage by modern signal processing techniques," *Biomedical Engineering, IEEE Transactions on*, vol. 62, no. 4, pp. 1159–1168, 2015.

[22] R. Talmon and R.R. Coifman, "Empirical intrinsic geometry for nonlinear modeling and time series filtering," *Proc. Nat. Acad. Sci.*, vol. 110, no. 31, pp. 12535–12540, 2013.

[23] R. Talmon and R. R. Coifman, "Intrinsic modeling of stochastic dynamical systems using empirical geometry," *Appl. Comput. Harmon. Anal.*, 2014, Tech. Report YALEU/DCS/TR1467.

[24] Ronen Talmon, Stéphane Mallat, Hitten Zaveri, and Ronald R Coifman, "Manifold learning for latent variable inference in dynamical systems," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 3843–3856, 2015.

[25] Carmeline J Dsilva, Ronen Talmon, C William Gear, Ronald R Coifman, and Ioannis G Kevrekidis, "Data-driven reduction for multiscale stochastic dynamical systems," *arXiv preprint arXiv:1501.05195*, 2015.

[26] A. Singer and R. R. Coifman, "Non-linear independent component analysis with diffusion maps," *Appl. Comput. Harmon. Anal.*, vol. 25, no. 2, pp. 226 – 239, 2008.

[27] N. El Karoui and H.-T. Wu, "Graph connection laplacian methods can be made robust to noise," *Annals of Statistics*, vol. 44, no. 1, pp. 346–372, 2016.

[28] A. Singer and H.-T. Wu, "Spectral convergence of the connection laplacian from random samples," *Information and Inference*, 2016.

[29] Amit Singer, "Lecture notes in "massive data analysis"," University Lecture, 2015.

[30] Emmanuel J Candes and Terence Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies," *IEEE transactions on information theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[31] T. Lee-Chiong, *Sleep Medicine: Essentials and Review*, Oxford, 2008.

[32] A. Rechtschaffen and A. Kales, *A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects*, Washington: Public Health Service, US Government Printing Office, 1968.

[33] C. Iber, S. Ancoli-Isreal, A.L. Chesson Jr., and S.F. Quan, *The AASM Manual for Scoring of Sleep and Associated Events-Rules: Terminology and Technical Specification*, American Academy of Sleep Medicine, 2007.

[34] A Karni, D Tanne, B S Rubenstein, J J Askenasy, and D Sagi, "Dependence on REM sleep of overnight improvement of a perceptual skill," *Science*, vol. 265, no. 5172, pp. 679–682, 1994.

[35] Jae-eun Kang, Miranda M. Lim, Randall J. Bateman, James J. Lee, Liam P. Smyth, John R. Cirrito, Nobuhiro Fujiki, Seiji Nishino, and David M. Holtzman, "Amyloid-b Dynamics are regulated by Orexin and the sleep-wake cycle," *Science*, vol. 326, no. Nov 13, pp. 1005–1007, 2009.

[36] Ferran Roche Campo, Xavier Drouot, Arnaud W Thille, Fabrice Galia, Belen Cabello, Marie-Pia D'Ortho, and Laurent Brochard, "Poor sleep quality is associated with late noninvasive ventilation failure in patients with acute hypercapnic respiratory failure.," *Critical care medicine*, vol. 38, no. 2, pp. 477–85, 2010.

[37] H. R. Colten and B. M. Altevogt, "Functional and economic impact of sleep loss and sleep-related disorders," in *Sleep Disorders and Sleep Deprivation: An Unmet Public Health Problem*, H. R. Colten and B. M. Altevogt, Eds. The National Academies Press, 2006.

[38] V. Bajaj and R. B. Pachori, "Automatic classification of sleep stages based on the time-frequency image of EEG signals," *Computer Methods and Programs in Biomedicine*, vol. 112, no. 3, pp. 320 – 328, 2013.

[39] N. Kannathal, M. Choo, U. Acharya, and P. Sadasivan, "Entropies for detection of epilepsy in EEG," *Computer Methods and Programs in Biomedicine*, vol. 80, pp. 187–194, 2005.

[40] S. Blanco, R. Quiroga, O. Rosso, and S. Kochen, "Time-frequency analysis of electroencephalogram series," *Physical Review E*, vol. 51, no. 3, pp. 2624–2631, 1995.

[41] S. Geng, W. Zhou, Q. Yuan, D. Cai, and Y. Zeng, "EEG non-linear feature extraction using correlation dimension and hurst exponent," *Neurological Research*, vol. 33, no. 9, pp. 908–912, 2011.

[42] G. S. Chung, B. H. Choi, K. K. Kim, Y. G. Lim, J. W. Choi, D.-U. Jeong, and K.-S. Park, "REM sleep classification with respiration rates," in *Information Technology Applications in Biomedicine, 2007. ITAB 2007. 6th International Special Topic Conference on*, 2007, pp. 194–197.

[43] G. Guerrero-Mora, P. Elvia, A.M. Bianchi, J. Kortelainen, M. Tenhunen, S.L. Himanen, M.O. Mendez, E. Arce-Santana, and O. Gutierrez-Navarro, "Sleep-wake detection based on respiratory signal acquired through a pressure bed sensor," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, 2012, pp. 3452–3455.

[44] J. Sloboda and M. Das, "A simple sleep stage identification technique for incorporation in inexpensive electronic sleep screening devices," in *Aerospace and Electronics Conference (NAECON), Proceedings of the 2011 IEEE National*, 2011, pp. 21–24.

[45] R. R. Lederman, R. Talmon, H.-t. Wu, Y.-L. Lo, and R. R. Coifman, "Alternating diffusion for common manifold learning with application to sleep stage assessment," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5758–5762.

[46] S. Mallat, "Group invariant scattering," *Pure and Applied Mathematics*, vol. 10, no. 65, pp. 1331–1398, 2012.

[47] Jonathan C Mattingly, Andrew M Stuart, and Michael V Tretyakov, "Convergence of numerical time-averaging and stationary measures via poisson equations," *SIAM Journal on Numerical Analysis*, vol. 48, no. 2, pp. 552–577, 2010.

[48] Sean P Meyn and Richard L Tweedie, *Markov chains and stochastic stability*, Springer Science & Business Media, 2012.

סינון לא לינארי מבוסס דיפוזיה להיתוך מידע מולטימודאלי

אורי כץ

סינון לא לינארי מבוסס דיפוזיה להיתוך מידע מולטימודאלי

חיבור על מחקר
לשם מילוי חלקי של הדרישות לקבלת תואר מגיסטר למדעים
בהנדסת חשמל

אורי כץ

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
אייר התשע"ז חיפה מאי 2017

תודות

המחקר נעשה בהנחיית פרופ' רונן טלמון מהפקולטה להנדסת חשמל בטכניון.

תקציר

בשנים האחרונות, בעקבות התפתחויות טכנולוגיות בתחומי החישה והאחסון, השימוש בחיישנים מסוגים שונים נהיה יותר ויותר נפוץ. מגמות של מזעור והקטנת העלות של אמצעי חישה מאפשרים לשלבם בקלות במגוון רחב של מוצרים. בניגוד למדידה באמצעות חיישן בודד, מדידה באמצעות מספר חיישנים מסוגים שונים מאפשרת למדוד היבטים משלימים של התופעה אותה אנו מודדים וכתוצאה מכך לקבל תמונה מקיפה ואמינה יותר של התופעה הנמדדת. התפתחויות אלו מעוררות עניין וצורך בפיתוח של כלים לעיבוד וניתוח אותות שמבוססים על היתוך המידע שהורכש ממספר חיישנים. על כן, האתגר של היתוך מידע הנרכש מחיישנים מולטימודאלים צובר תאוצה ועניין רב. בעבודה זו נתמקד בתרחיש הכולל תופעה או מערכת פיזיקלית, אותה אנו מעוניינים לחקור, הנמדדת על ידי מספר חיישנים, לאו דווקא בעלי אותה מודאליות. כל חיישן מודד היבט מסוים של התופעה, אבל חשוף להפרעות נוספות שאינן רלבנטיות לתופעה אותה אנו חוקרים. המטרה שלנו היא למצוא ייצוג של התופעה בה אנו מעוניינים, ולהנחית את ההפרעות שאינן קשורות אליה. אנו מציגים גישה המבוססת על שיטות של שערוך יריעה. שיטות אלו מתאימות במיוחד לעיבוד של מידע מולטימודאלי מאחר שהן חושפות את המבנה הגיאומטרי של המידע ולא מניחות ידע מוקדם או מודל מסוים, שלעיתים רבות אינו ידוע. בפרט, אנו מציעים פתרון המבוסס על סכמה לסינון לא לינארי שמחלצת רק את התופעות הנמדדות על ידי יותר מחיישן אחד, ומנחיתה תופעות אשר ייחודיות לחיישן בודד. המוטיבציה לבחירה זו היא שהפרעות או רעשים הם לרוב ייחודיים לחיישן מסוים, בעוד שהתופעה אותה אנו מודדים היא גלובלית ועל כן תהיה משותפת למספר חיישנים. בנוסף להצגת הניתוח התיאורטי, אנו מדגימים את השיטה שלנו על בעיית צעצוע ועל מדידות הנרכשות ממספר חיישנים בעלי מודאליות שונה למטרת שערוך של מצבי שינה מנבדקים במעבדות שינה. אנו מפעילים את הסכמה המוצעת לסינון הלא לינארי על המדידות מהחיישנים השונים ומקבלים ייצוג של תהליך השינה, הנלמד מתוך המדידות. מתוך הייצוג המתקבל אנו מסווגים את מצב השינה ומראים, על ידי מדדים אובייקטיבים, שגם ללא ידע מוקדם על המודאליות של החיישנים השיטה שלנו מניבה ייצוג, בעל התאמה טובה לתהליך השינה וחסין לרעש ולתופעות המיוחסות לחיישן בודד.

נושא נוסף שנבע מהאתגר בו נתקלנו בעת עיבוד של מידע הנרכש מחיישנים מולטימודאלים ומהווה אבן בניין בסיסית בכל אלגוריתם ללמידת יריעה הוא היכולת לזהות דימיון בין נקודות מידע. מספר מחקרים מהשנים האחרונות מעידים שמרחק מהלנוביס הוא כלי טוב למטרה זו מאחר והוא עמיד לרעש ומאפשר לחשוף את המבנה החבוי של סט המידע. אולם, החישוב של מרחק מהלנוביס מצריך שערוך מקומי של מטריצת הקווריאנס מתוך סט המידע הנתון לנו. שערוך מהימן של מטריצת הקווריאנס הינה משימה מאתגרת. כאשר המידע מסודר בזמן השיטה הנפוצה לשערוך של מטריצת הקוואריאנס היא על סמך חישוב של קוואריאנס המדגם בחלון זמן באורך סופי. גישה זו סובלת ממגבלות רבות, במיוחד כאשר היא מיושמת על מידע שנדגם ממערכת סטוכאסטית, דינאמית ומרובת קצבים. בעבודתנו אנחנו סוקרים את התהליך של חישוב מרחק מהלנוביס, ובפרט, אנו מתמקדים בשקלול התמורות הטמון בין שימור מקומיות השערוך לבין מזעור שגיאת המדגם. אנו מדגימים את ההשפעה של שגיאות השערוך של מטריצת הקווריאנס על היבטים שונים בלמידת היריעה, מנתחים אותם ומדגימים את ההשפעה המכרעת של קביעת אורך החלון על אותם היבטים. בנוסף, אנו מציגים גישה חדשה לשערוך של מטריצת הקווריאנס ומדגימים את יתרונותיה עבור ניתוח מערכת סטוכאסטית, דינאמית ומרובת קצבים.