5-16-2018

Hypothesis Testing and Model Estimation with Dependent Observations in Heterogeneous Sensor Networks

SIMA SOBHIYEH
Louisiana State University and Agricultural and Mechanical College, sima.sobhie@gmail.com

HYPOTHESIS TESTING AND MODEL ESTIMATION WITH DEPENDENT OBSERVATIONS IN HETEROGENEOUS SENSOR NETWORKS

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctoral of Philosophy

in

Electrical and Computer Engineering

by
Sima Sobhiyeh
BSc, Amirkabir University of Technology, 2011
MSc, Amirkabir University of Technology, 2013
August 2018

Acknowledgments

I would like to deeply thank my adviser, Dr. Mort Naraghi-Pour, for all his support, guidance, knowledge and patience through the last four years. Without his continuous support and effort, this dissertation would not be possible. I would also like to express my appreciation to Dr. Jin, Dr. Wei, Dr. Wolenski, and Dr. Zhou for serving on my doctoral examination committee. I would like to thank my husband and my parents for being there whenever I needed it. Their support, encouragement, patience, and love has motivated me towards completing my studies. Finally, I want to thank my colleagues and my friends who supported me all along the way.

Table of Contents

Abstract

Advances in microelectronics, communication and signal processing have enabled the development of inexpensive sensors that can be networked to collect vital information from their environment to be used in decision-making and inference. The sensors transmit their data to a central processor which integrates the information from the sensors using a so-called fusion algorithm. Many applications of sensor networks (SNs) involve hypothesis testing or the detection of a phenomenon. Many approaches to data fusion for hypothesis testing assume that, given each hypothesis, the sensors' measurements are conditionally independent. However, since the sensors are densely deployed in practice, their field of views overlap and consequently their measurements are dependent. Moreover, a sensor's measurement samples may be correlated over time. Another assumption often used in data fusion algorithms is that the underlying statistical model of sensors' observations is completely known. However, in practice these statistics may not be available prior to deployment and may change over the lifetime of the network due to hardware changes, aging, and environmental conditions. In this dissertation, we consider the problem of data fusion in heterogeneous SNs (SNs in which the sensors are not identical) collecting dependent data. We develop the expectation maximization algorithm for hypothesis testing and model estimation. Copula distributions are used to model the correlation in the data. Moreover, it is assumed that the distribution of the sensors' measurements is not completely known. we consider both parametric and non-parametric model estimation. The proposed approach is developed for both batch and online processing. In batch processing, fusion can only be performed after a block of data samples is received from each sensor, while in online processing, fusion is performed upon arrival of each data sample. Online processing is of great interest since for many applications, the long delay required for the accumulation of data in batch processing is not acceptable. To evaluate the proposed algorithms, both simulation data and real-world datasets are used. Detection performances of the proposed algorithms are compared with well-known supervised and unsupervised learning methods

as well as with similar EM-based methods, which either partially or entirely ignore the dependence in the data.

Chapter 1
Introduction to Sensor Networks and Data Fusion

Advances in microelectronics communication and signal processing has enabled the development of inexpensive sensors that can be networked to collect vital information that can be used in decision making, including estimation and detection. Today, sensor networks (SNs) are widely used in many diverse applications including environmental control for buildings [1–3], health monitoring [4–7], human activity detection [8–14], human identity detection [17–21], improving the quality of medical images [22–25, 27–29], detecting and tracking speakers [30–33], disaster management [34–37], precision agriculture [15, 38–40], highway monitoring [16, 41–44], and underwater surveillance [45–49].

There are two principal paradigms of operation for SNs. In a distributed (infrastructureless) mode, autonomous sensors make local decisions based on their own measurements and the information they receive from their neighbors without a central controller. Distributed algorithms which enable the operation of such networks have been the subject of several studies in recent years [50–56]. The scope of such algorithms, however, is somewhat limited as they are not applicable for many practical scenarios. The other paradigm which is considered here is sometimes referred to as centralized. In this approach, each sensor sends its measurements to a central controller referred to as the Fusion Center (FC). The FC then combines the data received from all the sensors using a data fusion algorithm. Design of the fusion algorithm has been the subject of numerous studies in recent years [18, 33, 50, 56, 60–62].

The size of a SN in terms of the number of sensors is determined by the application and affects the complexity of the fusion rule. For example in applications such as identity detection, medical image fusion, and patient home monitoring, the number of sensors is small (at most a few tens) and it may not be possible to increase the number of sensors. Consider a SN used to identify people based on their biometric data. Such an identity detection method, uses biometric data such as facial, fingerprint or iris images. Since bio-

metrics are unique for different people, this is an inherently secure and reliable method for human identity detection, for example, for secure building access control. It is clear that the number of biometric data is limited to fewer than ten and within that limitation each sensor added to the biometric-based detection network will greatly increase the processing and hardware cost. Similarly, consider a SN used for permitting home monitoring for chronic and elderly patients. For example, small wearable sensory devices have been developed which collect heart rate, oxygen saturation, and EKG data and relay the data over a short-range (100-m) wireless network to any number of receiving devices, including PDAs, laptops, or ambulance-based terminals [5]. The number of sensors collecting data in such applications is also small.

On the other hand, in applications such as precision agriculture or disaster management, a large number of inexpensive sensors, on the order of thousands, maybe installed [15, 34–40]. As an example, Intel recently installed a large number of small sensors in a vineyard in Oregon to monitor microclimates. The sensors measured temperature, humidity, and other factors to monitor the growing cycle of the grapes.The data was used to help prevent frostbite, mold, and other agricultural problems. In a disaster management scenario, a large number of inexpensive sensors maybe dropped from a helicopter and networked to detect survivors and assist in rescue operations.

In most applications of SNs the senors employed in the network are not identical. This may be due to physical and/or environmental conditions (e.g. hardware variations, device age, noise, etc.), or the type of data that the sensors collect. In either case the measurements of each sensor may follow a distinct probabilistic model and the fusion algorithm requires a multimodal modeling and signal processing approach [60]. Such networks are referred to as heterogeneous SNs. In the latter case, since each sensor has its own measurement modality, heterogeneity in the SN can take advantage of the complementary information from the different types of sensors. For example, a SN used for energy monitoring and control in buildings can take advantage of a heterogeneous framework by employing sensors that

measure temperature, light, humidity, CO2, or sound to detect room occupancy [2], [3]. The purpose of such heterogeneous SNs can be to improve thermal comfort, air quality, health, safety, or security for the occupants of a building. Moreover, they can reduce greenhouse gas emission and energy consumption resulting from air conditioning in buildings. Also in biometric-based identity detection, there has been a lot of interest in using a heterogeneous SN sometimes referred to as Multibiometrics which fuses different biometrics for better identification [17–21].

Another example for the application of SNs where the heterogeneity of the network plays an important role is combining different medical imaging modalities. More specifically, the complementary temporal and spatial resolutions of functional MRI (fMRI) and electroencephalography (EEG) signal modalities make them attractive candidates to be fused. EEG signals achieve a temporal resolution in the millisecond range, whereas the spatial localization of EEG signals has a precision on the scale of the centimeter only. On the other hand, fMRI offers a very high spatial resolution reaching sub-millimetric scale whereas its temporal resolution is limited to the order of seconds. Thus, by fusing these two signals, functional neuroimaging data with high spatial and temporal resolution can be obtained for improved brain activity detection [22–25]. Another good example for fusing different medical imaging modalities, is the fusion of therapeutic ultrasound (US) with a navigational modality, such as Computed Tomography (CT) in order to improve guidance when activating drugs, ablating tumors or delivering drugs beyond the blood brain barrier [27–29].

Another group of applications where heterogeneous SNs are used is human activity recognition which is important for providing activity assistance and care for users. In activity recognition problems, the data collected from a heterogeneous network of motion sensors is used to detect the activity performed by a human subject. For example, in [13,14], 3D acceleration, 3D gyro, and 3D magnetometer data were collected from sensors placed on the chest, the left ankle, and the right lower arm in order to detect the user's activity among

many classes of activities including standing still, sitting, lying down, walking, climbing stairs, waist bend forward, frontal elevation of arms, knees bending, cycling, jogging, running, jumping front and back. Other studies on human activity detection include [8–11]. In collaboration with the Motion Analysis Laboratory at the Spaulding Rehabilitation Hospital, Harvard University has also developed a tiny wearable device, consisting of three-axis accelerometer, gyroscope, and electromyogram sensors, for monitoring the limb movements and muscle activity [4–7]. The receiving device collecting data from such sensory devices can be programmed to fuse the different vital sign data and/or the motion data, for example, to make a binary decision and signaling to a nearby EMT or paramedic for help when there is an adverse change in patient status.

Many approaches to data fusion assume that, given the state of nature, the sensors' local measurements are conditionally independent. However, in most practical cases this assumption fails as the data collected by the sensors can be dependent over time (e.g., when the sensors' noise is dependent [63, 64]), as well as among the sensors (e.g., when the sensors have overlapping coverage area [65]). In some applications such as human identity detection, although different biometrics (data from different sensors) of a single individual are correlated but the biometrics of one person do not effect the biometrics or identity of another person, i.e., the data samples from each sensor are independent over time. On the other hand, in other applications, such as combining EEG and fMRI data, the data samples are dependent both among the sensors and over the time samples of each sensor [22–25]. As another example, we refer to applications with spatially-temporally correlated fields involving time-varying observations such as monitoring water contamination, or the temperature of an environment with time-varying observations [66]. Recently, the impact of temporal correlation of parameters on collaborative estimation systems has been studied in [66].

One popular approach to modeling the distribution of dependent data is the use of copula distributions. The popularity of copula distributions is due to their many advantages

such as their inherent decoupling property. Copulas separate the effect of the marginal distributions and the dependence structure in the data [60]. Therefore, in a copula-based detection method, changes in both complementary and mutual information can be detected due to this inherent decoupling in the copula theory. Furthermore, this feature of the copula opens up a lot of opportunities in statistical modeling by allowing us to model nonlinear dependencies or to represent joint probability density function (PDF) models that do not necessarily have a closed form.

In probability and statistics, copula theory has been used extensively to model dependent random variables. Recently several authors have used copula theory to model dependent data in signal processing and detection applications. Employing copulas for texture classification and multi-component image segmentation problems has been investigated in [67] and [68], respectively. In [62], copulas were used to fuse acoustic and seismic measurements in a footstep detection problem. In [69], copula theory was used to detect changes between two remotely sensed images before and after the occurrence of an event.

1.1 Estimation and Detection in Sensor Networks

In many applications of SNs, the fusion algorithm intends to detect a phenomenon (referred to as the state of nature) or to estimate a set of parameters. Estimation and detection strategies can be categorized as centralized or decentralized. In centralized detection, the sensor nodes transmit their actual (raw) measurements to the FC without any pre-processing. On the other hand, in decentralized detection, each node quantizes its data before transmission to the FC. For example in a hypothesis testing problem with K different hypotheses, each sensor may make a local decision and send its decision to the FC. The FC then judiciously combines the decisions of the sensors and makes a final decision.

The log-likelihood ratio test (LLRT) and the generalized likelihood ratio test (GLRT) are two common approaches for both centralized and decentralized estimation and detection in SNs. The problem of binary hypothesis testing based on the LLRT has been considered in [18] in a biometric-based detection problem where the face matching results from two

different face matching algorithms are combined to detect the identity of an individual. The authors consider a centralized detection strategy where at each time, each face matching algorithm sends a single continuous number to the FC. The data sent from the two face matching algorithms follow different probabilistic models and are dependent. Here the author use copula distributions to model the distribution of the dependent data. Fusion of dependent decisions of sensors in a decentralized framework using LLRT and GLRT has also been recently studied in [60,61]. Here the observations of all the sensors at each time instant are assumed to be dependent while, the observation samples of each sensor are assumed to be independent and identically distributed (iid) over time and copula distributions are used to model the dependence in the data. In [60, 61] the authors also study the effect of copula mis-specification. To improve the computational complexity of fusing discrete decisions using GLRT, the authors have proposed to inject noise into the local sensor decisions which, as a result, decreases the signal-to-noise ratio of the quantized data.

1.1.1 Parametric and non-parametric estimation

In many real-world scenarios the exact underlying statistics of the sensors' measurements is not available. Moreover, these statistics may vary over time and with deployment scenarios. Therefore prior to detecting the state of nature, an estimation of the sensors' measurement statistics is necessary. There are two general approaches to system model estimation, namely parametric and non-parametric estimation. If the estimation algorithm assumes that the distributions of sensors' measurements are known except for a set of parameters which may differ from sensor to sensor and for different networks in different environmental conditions, then the algorithm is based on a parametric estimation approach. In such cases, the FC first performs parametric estimation to estimate unknown parameters of the distribution of sensors' measurements and then using the complete data model it detects the state of nature. On the other hand, some estimation algorithms assume that the underlying distribution of the sensors' measurements is completely unknown and may not even match a distribution function with a closed form. In that case we employ non-parametric

estimation where the entire distribution of the sensors' measurements is estimated before the state of nature is detected. There are two approaches for non-parametric estimation of distribution functions: the histogram-based method, and the kernel-based method. In the histogram-based method, the data sample space is partitioned into bins. Let $j = 1, \cdots, J$ represent the bin number, 2Δ the duration of each bin, N the total number of samples and N_j the number of data samples in the interval $R_j = [x_0 + 2(j-1)\Delta, x_0 + 2j\Delta)$ where x_0 is the starting point of the first bin. The probability density function of the data at each point $x \in R_j$ is then approximated by

$$f(x) = \frac{N_j}{2\Delta N} \ , \quad x \in R_j \ , \quad j = 1, \cdots, J. \tag{1.1}$$

Note that although the histogram-based estimation is easy to compute, it only approximates the density at the center of each bin R_j and uses that value for all data samples in R_j. Moreover, histogram-based estimation produces a staircase function which is not differentiable. This will produce a difficulty for detection methods based on likelihood maximization.

Kernel-based estimation is a generalization of the histogram-based estimation allowing different levels of smoothness for the estimated density function. Let $K(.)$ represent a kernel function and Δ denote the bandwidth of the kernel. Let $y_n, \ n = 1, \cdots, N$ represent the data samples. Then the density function at each point x is given by

$$f(x) = \frac{1}{\Delta N} \sum_{n=1}^{N} K(\frac{x - y_n}{\Delta}). \tag{1.2}$$

Let $1(.)$ denote the characteristic function where for a set A,

$$1_A(x) = \begin{cases} 1 & \text{for } \ x \in A \\ 0 & \text{for } \ otherwise. \end{cases} \tag{1.3}$$

Using (1.3), we can rewrite (1.1) as

$$f(x) = \frac{1}{2\Delta N} \sum_{n=1}^{N} 1_{B_n}(x), \tag{1.4}$$

where $B_n = [y_n - \Delta, y_n + \Delta]$. We can see that by choosing $K(.)$ to be the uniform density in the kernel-based estimation, i.e., $K(\frac{x-Y_n}{\Delta}) = \frac{1}{2}1_{B_n}(x)$, we obtain the histogram-based estimation. Other examples of kernel functions include the Gaussian kernel, the Epanechnikov Kernel, the Laplacian Kernel, and the Quartic kernel [70].

1.1.2 Supervised versus unsupervised learning algorithms

State detection and model estimation methods consist of machine learning algorithms such as classification and regression. Most machine learning algorithms use labeled data known as training data to determine the data model and/or estimate the unknown model parameters. A labeled data is a data sample for which we already know the state of nature or equivalently the class to which the data sample belongs. Machine learning algorithms which require labeled data are categorized as supervised learning algorithms. Some common supervised learning algorithms include K-Nearest Neighbor (KNN) and Support Vector Machines (SVM). After building a KNN or SVM classifier using training data samples, we wish to use the classifier to detect the state of nature at the arrival of a new data sample or to equivalently classify a newly received data sample. Let M denote the total number of classes, and let x denote the newly received data sample. Upon receiving x, the KNN classifier draws a sphere centered at x containing K samples from the training dataset regardless of their class. Let K_m denote the number of samples in the training dataset that are from class m and fall inside the sphere. The KNN classifier will assign class m^* to x where $m^* = \text{argmax}_m\{K_m\}$, with ties broken arbitrarily. For $K = 1$, the KNN method is called the Nearest Neighbor (NN) method since any arriving data sample will be assigned to the same class as the data sample closest to it from the training dataset [71].

On the other hand, the SVM classifier is a fundamentally binary classifier which clas-

sifies the data samples based on a hyperplane which separates the data points of the two classes from each other. The best hyperplane for an SVM classifier is the one with the largest distance between the hyperplane and the point/points that are closest to it. This distance is referred to as the margin between the classes and the hyperplane. The data samples that lie on the boundary of this margin are called the support vectors. Since the support vectors are the data samples that build an SVM classifier, these classifiers are named SVM. Note that in general, it may not always be possible to separate the data by a hyperplane. In that case, SVM finds a hyperplane that separates many, but not all data points based on a penalty parameter. This is called a soft margin. Moreover, most practical classification scenarios involve M-ary classification where $M > 2$. Thus, there have been may attempts to build M-ary SVM classifiers by combining a number of binary SVMs. One approach called the one-versus-the-rest approach constructs M binary SVM classifiers, where the binary classification decision for each classifier involves being in a class m or not [71]. A major drawback of the one-versus-the-rest approach is that the training sets are imbalanced. To over come this issue a variant of the one-versus-the-rest scheme was proposed which modifies the cost values so that each binary classifier uses the weight one for the decision of being in a class m and the weight $1/(M-1)$ for being in any class except class m [71]. Another approach is to train $M(M-1)/2$ binary SVM classifiers on all possible pairs of classes, and then to classify test points by taking a majority vote on the results of all the binary classifiers. This is called the one-versus-one approach [71]. The major drawback of this method is its high computational cost.

An alternative approach to the supervised learning approach is to devise learning algorithms that do not require labeled data. Such an approach in learning algorithms is referred to as the unsupervised learning approach. This is an important advantage of unsupervised learning methods since in many applications, providing labeled data requires a high effort or can even be impossible [1]. For example, consider a smart heating system for optimizing the energy consumption. To decide whether or not to heat a room, these systems need

to detect room occupancy. However, labeling occupancy data is not always possible [1]. Thus, developing estimation and detection algorithms that do not require labeled data is important in such applications. Some unsupervised learning algorithms used for room occupancy detection include the geometric moving average (GeoMA) and the Page-Hinkley test (PHT). In the GeoMA, after all data samples are received, the geometric average of data samples within a sliding window are calculated. For each sample greater than the geometric average calculated at that sample point, the room is said to be occupied and otherwise unoccupied. The PHT is a more sophisticated version of GeoMA. After all data samples are received, the PHT detects increasing and decreasing changes in the stream of data samples. Upon finding an increasing change, the occupancy state at that time is set to one and upon finding a decreasing change the occupancy state at that time is set to zero and in the case of no change, the occupancy state is set to that of the previous sample.

Other popular methods for unsupervised classification are the *maximum Likelihood (ML)* and the *Expectation Maximization (EM)* algorithms. In the ML algorithm, a class m^* is assigned to data sample x if it maximizes the probability $p(x|m = m^*; \theta)$ where θ represents the model parameters. If the model parameters are latent and unavailable, maximizing the probability $p(x|m = m^*; \theta)$, which is referred to as the likelihood function, can be a very complex problem. The EM algorithm is one method for jointly estimating the parameters θ and solving the ML classification problem by maximizing the expectation of the probability function $p(x, m|\theta)$ given x and the current estimation of θ instead of directly maximizing the likelihood function. Let θ^{old} denote the current estimation of θ and $Q(\theta; \theta^{old}) \triangleq E_{m|x,\theta^{old}}[\log p(x, m|\theta)]$ represent the expectation of $p(x, m|\theta)$ given x and θ^{old}. The EM algorithm is an iterative algorithm where each iteration of the algorithm consists of two steps: the expectation step and the maximization step. In the expectation step, $Q(\theta; \theta^{old})$ is evaluated for using the current estimate of the parameter set and in the maximization step, $Q(\theta; \theta^{old})$ is maximized with respect to θ to obtain a new estimation for the parameter set. The expectation and maximization steps are iteratively performed

until convergence is reached.

1.1.3 Online versus batch-mode processing in estimation and detection

Estimation and detection at the FC can be performed via either *online* or *batch-mode* processing. In *online* processing, data is processed on a sample by sample basis whereas in *batch-mode* processing, an entire batch of data samples are processed after they have been received at the FC. All the aforementioned studies which consider estimation and detection using correlated data modeled by the copula theory ([18, 51, 60, 61, 72]), consider *batch-mode* processing at the FC. However, there are major drawbacks to batch-mode processing. One important drawback is that they require a lot of memory resource and complicated computational ability at the fusion center which also means high energy-consumption. Another important drawback is the significant delay caused by the FC accumulating a large number of samples before processing can commence. For many applications such long delays are unacceptable. For instance, consider a room occupancy detection problem for energy efficiency. In such applications detection has to be done at every time instance to decide whether or not to turn on the air-conditioning system without having to wait for sensor measurements to be collected over all time instances up until the end of the day for example. Similarly, in a security application where biometric data are combined to detect the identity of each individual in a group, the decision regarding the identity of each individual has to be made upon receiving the individuals biometrics and without having to wait for the data from the rest of the group. In such applications an online detection of the state of nature at each time instant is inherently necessary and thus, batch-mode based estimation and detection algorithms cannot be applied.

Recently, online EM-like algorithms have been developed to solve online classification problems. There are two dominant approaches to online EM-like estimation. Studies in the first approach including [73–76], follow the method proposed by Titterington [73]. In this method, a stochastic approximation algorithm is employed in the M-step of the algorithm.

More specifically, after each new observation is received, the unknown parameters are updated using the gradient of the incomplete data likelihood weighted by the complete data Fisher information matrix. The Titterington algorithm maximizes complete data likelihood given old parameters and new data using Newton's method in which he replaces the Hessian matrix term by its expectation. Moreover, he shows that, in exponential family models in which the parameters are the expected value of the sufficient statistics, the recursion is exact. The second approach is more aligned with the principles of the offline EM algorithm [77, 78]. In this approach, the E-step is replaced by a stochastic approximation of the offline E-step in order to incorporate the information brought by the new observation. However, the M-step remains the same as the M-step of the offline EM algorithm. The authors show that when the data likelihood belongs to the curved exponential family, Cappe's approach to online EM converges.

1.2 Contribution of this Dissertation

In this research we have considered an hypothesis testing problem using measurements collected from a heterogeneous SN. We derive a mathematical framework based on the copula theory to model the dependence in the data. It is further assumed that the distributions of sensors' measurements are not completely known and thus model estimation is required along with hypothesis testing. We convert the detection problem to an equivalent estimation problem. Thus, we first propose an unsupervised parametric estimation algorithm based on the EM and online EM algorithms in order to estimate the model parameters and jointly detect the state of nature at each time instant. Then, we consider the case where underlying distribution of the sensors' measurements are completely unknown and we propose a kernel-based non-parametric estimation algorithm to estimate the distribution of the sensors' measurements and to detect the state of nature.

In chapter 2, the online EM-based estimation and detection algorithm is proposed where the FC can process the data on a sample-by-sample basis. In the data model, in this case, we assume that the measurements of different sensors may be correlated with

each other while the measurements of each sensor are independent over time. In practical situations, the hypothesis may not be the only factor changing the sensors' measurements. Some environmental conditions which are not necessarily of our interest to detect, may also effect the sensors' measurements. For example, when detecting room occupancy via multi-sensing, the light and temperature in a room are not only affected by the presence of people inside the room but also by the time of the day such as morning or night, or by the season of the year. Thus, it is assumed that the data from each sensor are drawn from one of several different distributions each modeling the distribution of the sensor's measurements under a different environmental condition. It is worth mentioning that since the exact number of distributions to be considered is not always known, we later show that considering a larger number of distributions will not degrade the detection performance. Whereas, failing to consider different distributions in the model will significantly deteriorate the detection performance. In chapter 3, the EM-based estimation and detection algorithm is presented for the case where we assume that the data received from the sensors are dependent both over time and among the sensors. In many applications (e.g., biomedical or object tracking) there is non-negligible dependence among the samples collected by each sensor, which if ignored, can result in significant degradation in the performance of the detection scheme. In this chapter we once again consider that the distributions of sensors' measurements are known except for a set of parameters and thus the proposed algorithm includes parameter estimation as well as hypothesis detection. We devise a model based on the copula theory and Markov chains to account for the dependency in the data collected by different sensors and over time. Finally, in chapter 4, we extend the proposed EM-based algorithm to the case where the distribution of each sensors' measurements are unknown and a non-parametric estimation of the sensors' measurements are required for hypothesis testing. In this case, we develop the EM algorithm for both batch-mode and online processing.

Chapter 2
Online Hypothesis Testing and Parameter Estimation with Observations Correlated only Among Sensors

2.1 Introduction

In this chapter, we study the problem of hypothesis testing and model parameter estimation using correlated observations from a heterogeneous network of sensors. Many approaches to estimation and detection assume that, given each hypothesis, the sensors local measurements are conditionally independent. However, due to the overlap in the sensors' field of view, in most practical scenarios the data collected by different sensors are correlated [65]. Therefore several authors have recently applied copula theory to model this correlation and to develop fusion techniques for this case [18, 51, 60, 61, 72]. These studies, however, rely on batch processing at the FC. Batch processing algorithms have large memory, computational and energy requirements. More importantly, since the FC must accumulate a large number of samples before processing can commence, batch processing entails significant delay. For many applications, however, such long delays are unacceptable. For instance, consider a room occupancy detection problem for energy efficiency. In such applications detection has to be done at every time instance without having to wait until the end of the day. Similarly, in a security application where biometric data are combined to detect the identity of each individual in a group, the decision has to be made for every individual without having to wait for the data from the rest of the group. Here, we present an online processing solution to the hypothesis testing problem where the copula theory is used to model the correlation in the data.

Moreover, most detection algorithms are developed using supervised learning algorithms which require training with labeled data. However, in many applications, providing labeled data requires a high effort or can even be impossible [1]. Here, we present an unsupervised detection algorithm based on the EM and online-EM algorithms. The novelty of the proposed algorithm lies with in the two following facts: it incorporates correlated

data modeling in an online detection and estimation algorithm, moreover, it is a learning algorithm that is both unsupervised and performs detection on a sample-by-sample basis.

The rest of this chapter is organized as follows. In Section 2.2 the problem is defined and the system model is described. In Sections 2.3.1 and 2.3.2, the batch-mode and online EM-based hypothesis testing algorithms are developed. Numerical results are presented and discussed in Section 2.4. Finally, conclusions are drawn in Section 2.5.

2.2 Problem Formulation and the System Model

We consider a network of L heterogeneous sensors employed to detect the state of nature $\mathcal{H} \in \{\mathcal{H}_0, \mathcal{H}_1, \cdots, \mathcal{H}_{K-1}\}$. At time t, sensor l transmits its measurement, denoted by $d_{l,t} \in \Re$, to the FC. After T time instances, the FC has received LT measurements which we collect into the $L \times T$ *measurement matrix* $D = [d_{l,t}]$. It is assumed that for each $l = 1, 2, \ldots, L$ and any $t_1 < t_2$, given the hypotheses at times $t = t_1, t_1 + 1, \ldots, t_2$, the sensor measurements $d_{l,t_1}, d_{l,t_1+1}, \ldots, d_{l,t_2}$ are iid. However, at each time t, the data samples $d_{l,t}$, $l = 1, 2, \cdots, L$, are correlated. Let $\mathbf{d}_t \triangleq (d_{1,t}, d_{2,t}, \cdots, d_{L,t})^{Tr}$ where the superscript Tr denotes matrix transpose. The vector $\mathbf{h}_t = (h_{0,t}, h_{1,t}, \cdots, h_{K-1,t})^{Tr}$ is used to denote the state of nature at time t. If at time t, the state of nature is $\mathcal{H}_i$, then $\mathbf{h}_t = \mathbf{e}_i$ where $\mathbf{e}_i$ is the ith standard basis vector for $\Re^K$. For the entire observation period we construct the $K \times T$ *hypothesis matrix* $H = [h_{k,t}]$.

In an offline EM algorithm, having received the measurement matrix D, the FC must detect the state of nature for $t = 1, 2, \cdots, T$. To develop this algorithm, we need to evaluate the distribution of D given the hypothesis matrix H.

Denote the conditional cumulative distribution function (CDF) and PDF of $d_{l,t}$ given $\mathcal{H}_i$ by $F_{i,l}(d_{l,t}; \tilde{\psi}_{i,l})$ and $f_{i,l}(d_{l,t}; \tilde{\psi}_{i,l})$, respectively, where $\tilde{\psi}_{i,l}$ is the set of unknown parameters of the, otherwise known, distribution. Next, we model the joint distribution of the sensors' measurements given $\mathcal{H}_i$ by the copula distribution $C_m(.; \lambda_{m,i})$ where m denotes the type of copula being considered, and $\lambda_{m,i}$ denotes the set of unknown parameters of the copula distribution $C_m(.;.)$ under hypothesis $\mathcal{H}_i$. Therefore, the conditional distribution of $\mathbf{d}_t$

given $\mathbf{h}_t$ is given by

$$F(\mathbf{d}_t|\mathbf{h}_t;\mathbf{\Psi},\mathbf{\Lambda}_m)=\prod_{i=0}^{K-1}C_m\left(F_{i,1}(d_{1,t};\tilde{\psi}_{i,1}),\cdots,F_{i,L}(d_{L,t};\tilde{\psi}_{i,L});\lambda_{m,i}\right)^{h_{i,t}} \tag{2.1}$$

where $\mathbf{\Psi}\triangleq\{\tilde{\psi}_{i,l};0\leq i\leq K-1,1\leq l\leq L\}$ is the set of distribution parameters containing KL elements and $\mathbf{\Lambda}_m\triangleq\{\lambda_{m,0},\lambda_{m,1},\cdots,\lambda_{m,K-1}\}$ is the set of parameters of the copula distribution m. From (2.1), the conditional PDF of $\mathbf{d}_t$ given $\mathbf{h}_t$ is given by

$$Pr(\mathbf{d}_t|\mathbf{h}_t;\mathbf{\Psi},\mathbf{\Lambda}_m)=\prod_{i=0}^{K-1}\Big[c_m\left(F_{i,1}(d_{1,t};\tilde{\psi}_{i,1}),\cdots,F_{i,L}(d_{L,t};\tilde{\psi}_{i,L});\lambda_{m,i}\right)\prod_{l=1}^{L}f_{i,l}(d_{l,t};\tilde{\psi}_{i,l})\Big]^{h_{i,t}} \tag{2.2}$$

where $c_m(.;.)$ denotes the copula density function of $C_m(.;.)$.

We define the auxiliary probabilities $P(h_{i,t}=1)\triangleq[\phi_{i,t}]$, which represent the probability of hypothesis $\mathcal{H}_i$ at time t. Note that these are not prior probabilities. Rather they are only used as a tool to help us transform the hypothesis detection problem into an estimation problem for $\phi_{i,t}$ which we can solve using the EM algorithm. We denote $\mathbf{\Phi}\triangleq[\phi_{i,t}]$.

Our goal is to estimate the unknown parameters $\mathbf{\Phi},\mathbf{\Psi},\mathbf{\Lambda}_m$, and calculate the hypothesis matrix H using the estimated value for $\mathbf{\Phi}$ denoted by $\hat{\mathbf{\Phi}}$. With this approach the state of nature at time t is detected as

$$\hat{h}_{i^*,t}=\begin{cases}1 & ,\ i^*=\underset{0\leq i\leq K-1}{\text{argmax}}\ \hat{\phi}_{\text{i,t}}\\ 0 & ,\ \text{else}\end{cases} \tag{2.3}$$

In this chapter, we consider three types of copulas, namely the Gaussian ($\mathcal{G}$), the Student's t ($\mathcal{T}$), and the product ($\mathcal{P}$) copulas, for their wide practical application, and define $\mathcal{M}\triangleq\{\ \mathcal{G},\mathcal{T},\mathcal{P}\}$. It should be noted however, that our approach is not limited to these cases and a similar approach can be applied in the case of other copulas. The PDF

of the Gaussian copula is given by

$$c_{\mathcal{G}}(u_1, \cdots, u_N; \mathcal{R}) = \frac{1}{|\mathcal{R}|^{\frac{1}{2}}} \exp\big[-\frac{1}{2}\mathbf{z}^{Tr}(\mathcal{R}^{-1} - I_N)\mathbf{z}\big], \tag{2.4}$$

where $\mathbf{z} = [z_1, \cdots, z_N]^{Tr}$, $z_n = G^{-1}(u_n; 0, 1)$, for $n = 1, \cdots, N$, and $G(.; \mu, \sigma)$ is the Gaussian cumulative distribution with mean μ and standard deviation σ. Moreover, I_N represents the $N \times N$ identity matrix, $\mathcal{R}$ is the $N \times N$ correlation matrix of the Gaussian copula. The Student's t copula is given by

$$c_{\mathcal{T}}(u_1, \cdots, u_N; \mathcal{R}, \eta) = |\mathcal{R}|^{\frac{-1}{2}} \frac{a(\eta, N)\left(1 + \frac{1}{\eta}\mathbf{w}^{Tr}\mathcal{R}^{-1}\mathbf{w}\right)^{-\frac{\eta+N}{2}}}{\prod_{n=1}^{N}(1 + \frac{1}{\eta}w_n^2)^{-\frac{\eta+1}{2}}}, \tag{2.5}$$

where $\mathbf{w} = [w_1, \cdots, w_N]^{Tr}$, $w_n = St_\eta^{-1}(u_n)$, for $n = 1, \cdots, N$, where St_η is the standard Student's t distribution with parameter η. Moreover, $\mathcal{R}$ is the $N \times N$ correlation matrix, η is the degree of freedom of the Student's t copula and $a(\eta, N) = \frac{\Gamma(\frac{\eta+N}{2})\Gamma(\frac{\eta}{2})^{N-1}}{\Gamma(\frac{\eta+1}{2})^N}$. We assume that η is known and thus the parameter set of the Gaussian and Student's t copulas consist only of their correlation matrices. Note that the Product copula does not have any parameters and $c_{\mathcal{P}}(u_1, \cdots, u_N) = 1$.

To model the marginal distribution of each sensors' data, we consider P different Gaussian distributions. Let $\boldsymbol{\zeta}$ be a 3-D matrix containing the variables $\zeta_{i,p,t} \in \{0, 1\}$ where $\zeta_{i,p,t} = 1$ if at time instance t, the state of nature is $\mathcal{H}_i$ and the data are drawn from the pth distribution, and 0 otherwise. Then we can write

$$F_{i,l}(d_{l,t}; \tilde{\psi}_{i,l})^{h_{i,t}} = \prod_{p=1}^{P}\big[G(d_{l,t}; \mu_{i,l,p}, \sigma_{i,l,p})\big]^{\zeta_{i,p,t}} \tag{2.6}$$

where, $\mu_{i,l,p}$ and $\sigma_{i,l,p}$ denote the mean and standard deviation of the data collected by the lth sensor, under the pth distribution and hypothesis $\mathcal{H}_i$. In the following we use the shorthand notation $G(d_{l,t}; \psi_{i,l,p})$ and $g(d_{l,t}; \psi_{i,l,p})$ for the Gaussian distribution and PDF,

respectively, where $\psi_{i,l,p} \triangleq \{\mu_{i,l,p}, \sigma_{i,l,p}\}$.

We define $\mathbf{\Theta} \triangleq [\mathbf{\Omega}, \mathbf{\Psi}, \mathbf{\Lambda}_m]$ as the set of unknown parameters of the model, where $\mathbf{\Omega} \triangleq [\omega_{i,p,t}]$ and $\omega_{i,p,t} \triangleq Pr(\zeta_{i,p,t} = 1)$. Note that $\sum_{i=0}^{K-1} \sum_{p=1}^{P} \omega_{i,p,t} = 1$ and $\phi_{i,t} = \sum_{p=1}^{P} \omega_{i,p,t}$. Now,

$$Pr(\mathbf{d}_t, \zeta_{i,p,t} = 1; \mathbf{\Theta}) = Pr(\zeta_{i,p,t} = 1; \mathbf{\Theta}) Pr(\mathbf{d}_t | \zeta_{i,p,t} = 1; \mathbf{\Theta}) =$$

$$\omega_{i,p,t}\ c_m\left(G(d_{1,t}; \psi_{i,1,p}), \cdots, G(d_{L,t}; \psi_{i,L,p}); \lambda_{m,i}\right) \prod_{l=1}^{L} g(d_{l,t}; \psi_{i,l,p}), \tag{2.7}$$

Thus, the joint probability model given the unknown parameters of the model is given by

$$Pr(D, \boldsymbol{\zeta}; \mathbf{\Theta}) = \prod_{t=1}^{T} \prod_{i=0}^{K-1} \prod_{p=1}^{P} Pr(\mathbf{d}_t, \zeta_{i,p,t} = 1; \mathbf{\Theta})^{\zeta_{i,p,t}}. \tag{2.8}$$

2.3 Proposed EM-Based Algorithm

To estimate $\mathbf{\Theta}$, we employ the EM algorithm which iterates between the *expectation step* (E-step) and the *maximization step* (M-step) until convergence is reached. The E-step computes the expectation of the log-likelihood function of complete data $(D, \boldsymbol{\zeta})$ with respect to $\boldsymbol{\zeta}$, given the current estimate of the parameters $\mathbf{\Theta}^{n-1}$, namely

$$Q(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1)}) \triangleq E_{\boldsymbol{\zeta}|D;\mathbf{\Theta}^{(n-1)}}[\ln P(D, \boldsymbol{\zeta}; \mathbf{\Theta})] \tag{2.9}$$

In the M-step, $Q(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1)})$ is maximized with respect to $\mathbf{\Theta}$ to obtain the new estimate $\mathbf{\Theta}^{(n)}$, i.e.

$$\mathbf{\Theta}^{(n)} = \text{argmax}_{\mathbf{\Theta}}\{Q(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1)})\}, \tag{2.10}$$

The idea used in [77] is to replace the expectation step with a stochastic approximation step, while keeping the M-step unchanged. Following [77], the stochastic approximation of

the E-Step at time t is given by

$$Q(\mathbf{\Theta};\mathbf{\Theta}^{(t)}) = (1-\epsilon^{(t)})Q(\mathbf{\Theta};\mathbf{\Theta}^{(t-1)}) + \epsilon^{(t)} E_{\zeta_{i,p,t}|\mathbf{d}_t;\mathbf{\Theta}^{(t-1)}}\left[\ln P(\mathbf{d}_t,\zeta_{i,p,t};\mathbf{\Theta})\right], \tag{2.11}$$

where $\{\epsilon^{(t)}\}$ is a decreasing sequence of positive step sizes. The M-step remains unchanged and is given by

$$\mathbf{\Theta}^{(t+1)} = \text{argmax}_{\mathbf{\Theta}}\{Q(\mathbf{\Theta};\mathbf{\Theta}^{(t)})\}, \tag{2.12}$$

In [77], the authors focus on the case where the complete data likelihood belongs to an exponential family satisfying

$$f(x;\Theta) = h(x)\exp\{-b(\Theta) + < s(x), r(\Theta) >\}, \tag{2.13}$$

where, $< .,. >$ denotes the scalar product between two vectors and $s(x)$ denotes the complete data sufficient statistic. In this case, the optimization problem in (2.10) reduces to

$$\overline{\theta}(s) \triangleq \text{argmax}_{\Theta}\{-b(\Theta) + < s, r(\Theta) >\}. \tag{2.14}$$

Equation (2.14) indicates that to update the parameters in each iteration of EM, the updating function, $\overline{\theta}(s)$, only requires the sufficient statistic $s(x)$. Therefore, to find the final estimate of the unknown parameters, we do not need to update $Q(\Theta;\Theta^{(t)})$ in each iteration, as in (2.11). Instead we only need to update the sufficient statistic $s(x)$. Therefore, defining

$$S^{(t^*)} \triangleq \frac{1}{t^*}\sum_{t=1}^{t^*} E_{\zeta_{i,p,t}|\mathbf{d}_t;\mathbf{\Theta}^{(t-1)}}\left[s(\mathbf{d}_t,\zeta_{i,p,t})\right], \tag{2.15}$$

where $s(\mathbf{d}_t,\zeta_{i,p,t})$ is the sufficient statistic for the complete data, the online update rule is

given by

$$S^{(t^*)} = (1 - \epsilon^{(t^*)})S^{(t^*-1)} + \epsilon^{(t^*)} E_{\zeta_{i,p,t^*}|\mathbf{d}_{t^*};\boldsymbol{\Theta}^{(t^*-1)}} \left[s(\mathbf{d}_{t^*}, \zeta_{i,p,t^*}) \right]. \tag{2.16}$$

Eq. (2.16) constitutes the E-Step of online EM and the parameter update rule, M-step, is given by

$$\Theta^{(t^*)} = \overline{\theta}(S^{(t^*)}). \tag{2.17}$$

For the convergence properties of the online EM algorithm using the stochastic approximation for the E-step we refer to [77].

In what follows we first develop the batch mode EM algorithm for which we define statistics similar to those in [77] and show that these statistics are sufficient for updating the parameters in the M-step. Later we extend the proposed method for online processing where the E-step only updates those sufficient statistics according to (2.16).

2.3.1 Proposed Batch-Mode EM-Based Algorithm

In the following the superscript (n, T) on a parameter denotes the estimated value of the parameter in the nth iteration of EM using T data samples. Moreover, the subscript m denotes the copula type where $m \in \mathcal{M}$.

- **Expectation Step (Batch Mode)**

To derive the expectation of the log-likelihood function, we start by deriving the log-likelihood function

$$L(D, \boldsymbol{\zeta}; \boldsymbol{\Theta}) \triangleq \log Pr(D, \boldsymbol{\zeta}; \boldsymbol{\Theta}) = \sum_{t=1}^{T} \sum_{i=0}^{K-1} \sum_{p=1}^{P}$$
$$\zeta_{i,p,t} \Big[\log \omega_{i,p,t} + \sum_{l=1}^{L} \log g(d_{l,t}; \psi_{i,l,p}) + \log c_m \left(G(d_{1,t}; \psi_{i,1,p}), \cdots, G(d_{L,t}; \psi_{i,L,p}); \lambda_{m,i} \right) \Big]. \tag{2.18}$$

Define $\alpha^{(n-1,T)}(i,p,t) \triangleq E[\zeta_{i,p,t}|D;\mathbf{\Theta}^{(n-1,T)}]$. Then the expectation of the log-likelihood function given the current estimate of the parameters $\mathbf{\Theta}^{(n-1,T)}$ is given by

$$Q_m(\mathbf{\Theta};\mathbf{\Theta}^{(n-1,T)}) \triangleq E_{\boldsymbol{\zeta}|D;\mathbf{\Theta}^{(n-1,T)}}[\log Pr(D,\boldsymbol{\zeta};\mathbf{\Theta})] = \sum_{t=1}^{T}\sum_{i=0}^{K-1}\sum_{p=1}^{P}$$
$$\alpha^{(n,T)}(i,p,t)\Big[\sum_{l=1}^{L}\log g(d_{l,t};\psi_{i,l,p}) + \log c_m(G(d_{1,t};\psi_{i,1,p}),\cdots,G(d_{L,t};\psi_{i,L,p});\lambda_{m,i}) + \log\omega_{i,p,t}\Big]. \tag{2.19}$$

More specifically, we can write $Q_m(\mathbf{\Theta};\mathbf{\Theta}^{(n-1,T)})$ for the three copula types under consideration as follows. For the Product copula we have

$$Q_{\mathcal{P}}(\mathbf{\Theta};\mathbf{\Theta}^{(n-1,T)}) = \sum_{t=1}^{T}\sum_{i=0}^{K-1}\sum_{p=1}^{P}$$
$$\alpha^{(n,T)}(i,p,t)\left[-\frac{L}{2}\log 2\pi + \log|\mathbf{\Sigma}_{i,p}| - \frac{1}{2}\mathbf{y}_{i,p}(t)^{Tr}\mathbf{y}_{i,p}(t) + \log\omega_{i,p,t}\right], \tag{2.20}$$

where, $\mathbf{\Sigma}_{i,p} \triangleq \text{diag}\{1/\sigma_{i,1,p},\cdots,1/\sigma_{i,L,p}\}$, $\mathbf{y}_{i,p}(t) \triangleq \mathbf{\Sigma}_{i,p}\left(\mathbf{d}_t - \boldsymbol{\mu}_{i,p}\right)$, $\boldsymbol{\mu}_{i,p} \triangleq [\mu_{i,1,p},\cdots,\mu_{i,L,p}]^{Tr}$, $|A|$ denotes the determinant of matrix A. For the Gaussian copula we have

$$Q_{\mathcal{G}}(\mathbf{\Theta};\mathbf{\Theta}^{(n-1,T)}) = \sum_{t=1}^{T}\sum_{i=0}^{K-1}\sum_{p=1}^{P}$$
$$\frac{\alpha^{(n,T)}(i,p,t)}{2}\left[-L\log 2\pi + 2\log|\mathbf{\Sigma}_{i,p}| - \log|\lambda_{\mathcal{G},i}| - \left(\mathbf{y}_{i,p}(t)\right)^{Tr}\left(\lambda_{\mathcal{G},i}^{-1}\right)\mathbf{y}_{i,p}(t) + 2\log\omega_{i,p,t}\right], \tag{2.21}$$

and finally for the Student's t copula,

$$Q_{\mathcal{T}}(\mathbf{\Theta};\mathbf{\Theta}^{(n-1,T)}) = Q_{\mathcal{P}}(\mathbf{\Theta};\mathbf{\Theta}^{(n-1,T)}) + \sum_{t=1}^{T}\sum_{i=0}^{K-1}\sum_{p=1}^{P}\alpha^{(n,T)}(i,p,t)\left[-\frac{1}{2}\log|\lambda_{\mathcal{T},i}|\right.$$
$$\left.-\frac{\eta+1}{2}\log\left|I_L + \frac{1}{\eta}\mathbf{V}_{i,p}(t)\right| - \frac{\eta+L}{2}\log\left(1+\frac{1}{\eta}\mathbf{v}_{i,p}(t)^{Tr}\lambda_{\mathcal{T},i}^{-1}\mathbf{v}_{i,p}(t)\right) + \tilde{a}(\eta,L)\right], \tag{2.22}$$

where $\mathbf{v}_{i,p}(t) \triangleq [v_{i,p,1}(t), v_{i,p,2}(t), \cdots, v_{i,p,L}(t)]^{Tr}$, and $\mathbf{V}_{i,p}(t) \triangleq \text{diag}\{v_{i,p,1}(t), \cdots, v_{i,p,L}(t)\}$, where $v_{i,p,l}(t) = St^{-1}(G(d_{l,t}; \psi_{i,l,p}))$, and $\tilde{a}(\eta, L) = \log a(\eta, L)$.

In the *Expectation step*, we need to calculate $\alpha^{(n,T)}(i,p,t)$ which is evaluated from

$$\alpha^{(n,T)}(i,p,t) = E[\zeta_{i,p,t}|D; \mathbf{\Theta}^{(n-1,T)}] = Pr(\zeta_{i,p,t} = 1|D; \mathbf{\Theta}^{(n-1,T)}) = \frac{Pr(\mathbf{d}_t, \zeta_{i,p,t} = 1; \mathbf{\Theta}^{(n-1,T)})}{\sum_{j=0}^{K-1}\sum_{q=1}^{P} Pr(\mathbf{d}_t, \zeta_{j,q,t} = 1; \mathbf{\Theta}^{(n-1,T)})}. \tag{2.23}$$

- **maximization step**

In the *Maximization step*, we maximize $Q_m(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1,T)})$ with respect to $\mathbf{\Theta}$ to obtain the new parameters $\mathbf{\Theta}^{(n,T)}$. For brevity the proofs of the lemmas stated in this section are presented in the appendix A.

To obtain the new estimate of $\mathbf{\Omega}$, we solve

$$\begin{aligned} &\underset{\omega_{i,p,t}}{\text{Maximize}} \quad Q_m(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1,T)}) \\ &\text{Subject to :} \quad \sum_{i=0}^{K-1}\sum_{p=1}^{P} \omega_{i,p,t} = 1, \end{aligned} \tag{2.24}$$

for $m \in \mathcal{M}$. Defining the function $\overline{\omega}(x) \triangleq x$, we have

Lemma 1. *By solving the optimization problem in* (2.24)*, the parameter update formula for* $\omega_{i,p,t}$ *is given by*

$$\omega_{i,p,t}^{(n,T)} = \overline{\omega}(\alpha^{(n,T)}(i,p,t)) = \alpha^{(n,T)}(i,p,t). \tag{2.25}$$

To obtain the new estimate of $\mathbf{\Lambda}_m$, $m \in \{\mathcal{G}, \mathcal{T}\}$ [1] , we solve the constrained optimization

[1] As mentioned previously, the Product copula does not have any parameters.

problem

$$\underset{\lambda_{m,i}^{-1}}{\text{Minimize}} \quad Q_m(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(n-1,T)}) \tag{2.26}$$

$$\text{Subject to} : \lambda_{m,i}^{-1} \in \Upsilon_L^+, \quad 0 \leq i \leq, K-1,$$

where Υ_L^+ is the set of $L \times L$ positive semi-definite matrices. It can be shown that $Q_m(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(n-1,T)})$ is a convex function of $\lambda_{m,i}^{-1}$ for $m \in \{\mathcal{G}, \mathcal{T}\}$.

Let us define,

$$\mathbf{y}_{i,p}^{(n-1,T)}(t) \triangleq \boldsymbol{\Sigma}_{i,p}^{(n-1,T)}(\mathbf{d}_t - \boldsymbol{\mu}_{i,p}^{(n-1,T)}), \tag{2.27}$$

$$\mathbf{v}_{i,p}^{(n-1,T)}(t) \triangleq [v_{i,p,1}^{(n-1,T)}(t), v_{i,p,2}^{(n-1,T)}(t), \cdots, v_{i,p,L}^{(n-1,T)}(t)]^{Tr}, \tag{2.28}$$

where $v_{i,p,l}^{(n-1,T)}(t) = St^{-1}(G(d_{l,t}; \psi_{i,l,p}^{(n-1,T)}))$, and the function

$$\overline{\lambda}_m \left(S_{m,1}^{(n,T)}(i), S_2^{(n,T)}(i) \right) \triangleq \frac{S_{m,1}^{(n,T)}(i)}{S_2^{(n,T)}(i)}, \tag{2.29}$$

where

$$S_{\mathcal{G},1}^{(n,T)}(i) \triangleq \frac{1}{T} \sum_{t=1}^{T} \sum_{p=1}^{P} \alpha^{(n,T)}(i,p,t) \mathbf{y}_{i,p}^{(n-1,T)}(t) (\mathbf{y}_{i,p}^{(n-1,T)}(t))^{Tr}, \tag{2.30}$$

$$S_{\mathcal{T},1}^{(n,T)}(i) \triangleq \frac{\eta + L}{T} \sum_{t=1}^{T} \sum_{p=1}^{P} \frac{\alpha^{(n,T)}(i,p,t) \mathbf{v}_{i,p}^{(n-1,T)}(t) (\mathbf{v}_{i,p}^{(n-1,T)}(t))^{Tr}}{\eta + (\mathbf{v}_{i,p}^{(n-1,T)}(t))^{Tr} (\lambda_i^{(n-1,T)})^{-1} \mathbf{v}_{i,p}^{(n-1,T)}(t)}, \tag{2.31}$$

$$S_2^{(n,T)}(i) \triangleq \frac{1}{T} \sum_{t=1}^{T} \sum_{p=1}^{P} \alpha^{(n,T)}(i,p,t), \tag{2.32}$$

Lemma 2. *The solution to the optimization problem in* (2.26) *is given by* $\lambda_{m,i} = \overline{\lambda}_m \left(S_{m,1}^{(n,T)}(i), S_2^{(n,T)}(i) \right)$.

We would like to point out that $S_{\mathcal{G},1}^{(n,T)}(i)$ is the weighted sample correlation matrix of the data and $S_2^{(n,T)}(i)$ is the mean of the weights (Both averaged over time and distribution types.). Therefore in the case of the Gaussian copulas, the solution to (2.26) is the empirical correlation matrix.

As $T \longrightarrow \infty$,[2] the matrix obtained from $\overline{\lambda}_m \left(S_{m,1}^{(n,T)}(i), S_2^{(n,T)}(i) \right)$ will be almost surely positive definite (PD). However, it does not necessarily have unit diagonal values. In order to have a valid correlation matrix, we apply the algorithm proposed by Higham [79] to obtain the closest correlation matrix to the solution of (2.26). Therefore the parameter update rule is given by

$$\lambda_{m,i}^{(n,T)} = \overline{\lambda}_m \left(S_{m,1}^{(n,T)}(i), S_2^{(n,T)}(i) \right). \tag{2.33}$$

To obtain the new estimate of $\mathbf{\Psi}$, we solve the optimization problems

$$\underset{\boldsymbol{\mu}_{i,p}}{\text{Maximize}} \quad Q_m(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1,T)}), \tag{2.34}$$

$$\begin{aligned} &\underset{\sigma_{i,l,p}}{\text{Minimize}} \quad Q_m(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1,T)}) \\ &\text{Subject to :} \quad \sigma_{i,l,p} > 0, \end{aligned} \tag{2.35}$$

Note that due to the decoupling obtained by copula based modeling, $Q(\Theta; \Theta^{(n-1,T)})$ consists of the summation of two major parts, one influenced by the marginal distribution of the sensors data and the other by the copula function. However, according to [72] and the theory of *Inference Functions for Margins* (IFM) [3] [57], the latter part of $Q(\Theta; \Theta^{(n-1,T)})$

[2] In fact when observations are independent samples of a continuous random variable, this property holds for $T \geq L$.

[3] Using IFM, extension of the proposed approach to other copulas such as the Arcamedian family and

does not play a considerable role in the optimization of $Q(\Theta;\Theta^{(n-1,T)})$ with respect to $\psi_{i,l,p}$. Since in the case of maximizing $Q(\Theta;\Theta^{(n-1,T)})$ with respect to Ψ for $m=\mathcal{T}$, a closed form solution cannot be obtained, we instead solve the more simple problem of maximizing the first part of $Q(\Theta;\Theta^{(n-1,T)})$ with respect to Ψ for which a closed form solution can be obtained.

Once again the optimization problems in hand are convex and we define the function

$$\overline{\mu}\left(S_3^{(n,T)}(i,p),S_4^{(n,T)}(i,p)\right)\triangleq\frac{S_3^{(n,T)}(i,p)}{S_4^{(n,T)}(i,p)},\tag{2.36}$$

where

$$S_3^{(n,T)}(i,p)\triangleq\frac{1}{T}\sum_{t=1}^{T}\alpha^{(n,T)}(i,p,t)\mathbf{d}_t,\tag{2.37}$$

and

$$S_4^{(n,T)}(i,p)\triangleq\frac{1}{T}\sum_{t=1}^{T}\alpha^{(n,T)}(i,p,t).\tag{2.38}$$

Note, that $S_3^{(n,T)}(i,p)$ is the weighted empirical mean of the data and $S_4^{(n,T)}(i,p)$ is the mean of the weights (Both averaged over time).

Lemma 3. *By solving the optimization problem in* (2.34) *for* $m\in\mathcal{M}$*, the parameter update formula for* $\boldsymbol{\mu}_{i,p}$ *is given by*

$$\boldsymbol{\mu}_{i,p}^{(n,T)}=\overline{\mu}\left(S_3^{(n,T)}(i,p),S_4^{(n,T)}(i,p)\right).\tag{2.39}$$

Let us define

$$\gamma_{i,l,p}^{(n,T)}\triangleq\overline{\gamma}\left(S_5^{(n,T)}(i,l,p),\lambda_i^{(n-1,T)}\right)\triangleq\left((\lambda_i^{(n-1,T)})^{-1}\right)_{l,l}S_5^{(n,T)}(i,l,p),\tag{2.40}$$

solving the corresponding optimization problems is straight forward.

where

$$S_5^{(n,T)}(i,l,p) \triangleq \frac{1}{T}\sum_{t=1}^{T}\alpha^{(n,T)}(i,p,t)\Big(d_{l,t}-\mu_{i,l,p}^{(n-1,T)}\Big)^2 \tag{2.41}$$

is the weighted empirical variance of the data (averaged over time), and $(A)_{k,l}$ denotes the element from the k th row and l th column of the matrix A. Moreover,

$$\beta_{i,l,p}^{(n,T)} \triangleq \overline{\beta}\left(\boldsymbol{\mathcal{S}}_{\backslash l}^{(n,T)}(i,l,p),(\lambda_i^{(n-1,T)})_{\backslash l}\right) \triangleq \frac{1}{2}\sum_{\substack{k=1\\k\neq l}}^{L}\left((\lambda_i^{(n-1,T)})^{-1}\right)_{k,l}\mathcal{S}_k^{(n,T)}(i,l,p), \tag{2.42}$$

where the vector $\boldsymbol{\mathcal{S}}^{(n,T)}(i,l,p)=\left[\mathcal{S}_k^{(n,T)}(i,l,p)\right]_L$ is defined as

$$\boldsymbol{\mathcal{S}}^{(n,T)}(i,l,p) \triangleq \frac{1}{T}\sum_{t=1}^{T}\alpha^{(n,T)}(i,p,t)\left(\mathbf{d}_t-\boldsymbol{\mu}_{i,p}^{(n-1,T)}\right)\odot\mathbf{y}_{i,p}^{(n,T)}(t), \tag{2.43}$$

and $\odot$ denotes element-wise product. Moreover, in (2.42), the notation $A_{\backslash l}$ denotes all the elements of the lth column of the matrix A expect for the lth element and for a vector $\mathbf{a}$, the notation $\mathbf{a}_{\backslash l}$ denotes all the elements expect for the lth element of the vector $\mathbf{a}$. We define the functions

$$\overline{\sigma}_{\mathcal{G}}\left(S_4^{(n,T)}(i,p),\beta_{i,l,p}^{(n,T)},\gamma_{i,l,p}^{(n,T)}\right) \triangleq \frac{\gamma_{i,l,p}^{(n,T)}}{-\beta_{i,l,p}^{(n,T)}+\sqrt{(\beta_{i,l,p}^{(n,T)})^2+\gamma_{i,l,p}^{(n,T)}S_4^{(n,T)}}}, \tag{2.44}$$

and for $m\in\{\mathcal{T},\mathcal{P}\}$,

$$\overline{\sigma}_m\left(S_4^{(n,T)}(i,p),S_5^{(n,T)}(i,p)\right) \triangleq \frac{S_5^{(n,T)}(i,p)}{S_4^{(n,T)}(i,p)}. \tag{2.45}$$

Lemma 4. *By solving the optimization problem in* (2.35) *for* $m\in\mathcal{M}$*, the parameter update*

formula for $\sigma_{i,l,p}$ is given by

$$\sigma_{i,l,p}^{(n,T)} = \begin{cases} \overline{\sigma}_m \left(S_4^{(n,T)}, \beta_{i,l,p}^{(n,T)}, \gamma_{i,l,p}^{(n,T)} \right), \ \textit{for } m = \mathcal{G}, \\ \overline{\sigma}_m \left(S_4^{(n,T)}, S_5^{(n,T)} \right), \ \textit{for } m \in \{\mathcal{T}, \mathcal{P}\}. \end{cases} \tag{2.46}$$

2.3.2 Proposed Online EM-Based Algorithm

Similar to the approach in [80], our proposed online algorithm consists of two stages. In the first stage which is called the initialization stage, an initial estimate of the parameters are calculated. To this end, the batch-mode EM algorithm, described in Section 2.3.1, is performed using a small number of data samples, say $T_0 \ll T$.

In the second stage, upon receiving a measurement sample from the sensors at time $t^* > T_0$, the FC forms the vector $\mathbf{d}_{t^*} = [d_{1,t^*}, \ldots, d_{L,t^*}]^{Tr}$ and performs the two steps of the online algorithm for a predetermined small number of N "mini-iterations". To initialize the parameters, at any time $t \geq T_0$, we set $\mathbf{\Theta}^{(0,t+1)} = \mathbf{\Theta}^{(N,t)}$. In other words the last estimated parameter from the N mini-iterations from the t-th data sample will be used as the initial parameter for the mini-iterations of $t+1$ sample.

To develop the online EM algorithm for the problem at hand, we need to derive the stochastic approximation of batch-mode E-step. The online M-step will be the same as the M-step of the batch-mode EM.

The update formulas for the M-step of the batch-mode EM in (2.25), (2.29), (2.39), and (2.42), (2.40), (2.46) indicate that the updated quantities are functions of the statistics $S_{m,1}$, $m \in \{\mathcal{G}, \mathcal{T}\}$ S_j, $j = 2, \ldots, 5$, and $\boldsymbol{\mathcal{S}}$ defined in (2.30), (2.31), and (2.32), (2.37), (2.38), (2.41), and (2.43), respectively. Therefore, as in the online version of EM, we only need to update the statistics sufficient for updating the parameters in the M-step. Thus we define

$$S_{m,1}^{(n,t^*)} = \frac{1}{t^*} \sum_{t=1}^{t^*} E_{\zeta_{i,p,t}|\mathbf{d}_t;\Theta^{(n-1,t)}} \left[s_{m,1}^{(n)}(\zeta_{i,p,t}^{(n)}) \right], \tag{2.47}$$

for $m \in \{\mathcal{G}, \mathcal{T}\}$, and

$$S_j^{(n,t^*)} = \frac{1}{t^*} \sum_{t=1}^{t^*} E_{\zeta_{i,p,t}|\mathbf{d}_t;\Theta^{(n-1,t)}} \left[s_j^{(n)}(\zeta_{i,p,t}^{(n)}) \right], \tag{2.48}$$

for $j = 2, \cdots, 5$, and

$$\boldsymbol{\mathcal{S}}^{(n,t^*)} = \frac{1}{t^*} \sum_{t=1}^{t^*} E_{\zeta(i,p,t)|\mathbf{d}_t;\Theta^{(n-1,t)}} \left[\boldsymbol{\mathcal{f}}^{(n)}(\zeta_{i,p,t}^{(n)}) \right]. \tag{2.49}$$

Note that in the online case, the superscript (n,t) denotes estimated parameter in the nth iteration $(1 \leq n \leq N)$ at time instant t and

$$s_{\mathcal{G},1}^{(n)}(\zeta_{i,p,t}^{(n)}) = \sum_{p=1}^{P} \zeta_{i,p,t}^{(n)} \mathbf{y}_{i,p}^{(n-1)}(t) (\mathbf{y}_{i,p}^{(n-1)}(t))^{Tr}, \tag{2.50}$$

$$s_{\mathcal{T},1}^{(n)}(\zeta_{i,p,t}^{(n)}) = (\eta + L) \sum_{p=1}^{P} \frac{\zeta_{i,p,t}^{(n)} \mathbf{v}_{i,p}^{(n-1)}(t) (\mathbf{v}_{i,p}^{(n-1)}(t))^{Tr}}{\eta + (\mathbf{v}_{i,p}^{(n-1)}(t))^{Tr} (\lambda_{\mathcal{T},i}^{(n-1,t)})^{-1} \mathbf{v}_{i,p}^{(n-1)}(t)}, \tag{2.51}$$

$$s_2^{(n)}(\zeta_{i,p,t}^{(n)}) = \sum_{p=1}^{P} \zeta_{i,p,t}^{(n)}, \tag{2.52}$$

$$s_3^{(n)}(\zeta_{i,p,t}^{(n)}) = \zeta_{i,p,t}^{(n)} d_{l,t}, \tag{2.53}$$

$$s_4^{(n)}(\zeta_{i,p,t}^{(n)}) = \zeta_{i,p,t}^{(n)}, \tag{2.54}$$

$$s_5^{(n)}(\zeta_{i,p,t}^{(n)}) = \zeta_{i,p,t}^{(n)} (d_{l,t} - \mu_{i,l,p}^{(n-1,t)})^2, \tag{2.55}$$

$$\boldsymbol{\mathcal{f}}^{(n)}(\zeta_{i,p,t}^{(n)}) = \zeta_{i,p,t}^{(n)} (\mathbf{d}_t - \boldsymbol{\mu}_{i,p}^{(n-1,t)}) \odot \mathbf{y}_{i,p}^{(n)}(t). \tag{2.56}$$

Moreover,

$$\alpha^{(n)}(i,p,t) \triangleq E_{\zeta(i,p,t)|\mathbf{d}_t;\Theta^{(n-1,t)}} \left[\zeta_{i,p,t}^{(n)} \right] = \frac{P(\mathbf{d}_t, \zeta_{i,p,t}^{(n)} = 1 | \Theta^{(n-1,t)})}{\sum_{j=0}^{M-1} \sum_{q=1}^{P} P(\mathbf{d}_t, \zeta_{j,q,t}^{(n)} = 1 | \Theta^{(n-1,t)})} \tag{2.57}$$

and

$$P(\mathbf{d}_t, \zeta_{i,p,t}^{(n)} = 1|\Theta^{(n-1,t)}) = \omega_{i,p,t}^{(n-1)} \prod_{l=1}^{L} g(d_{l,t}; \psi_{i,l,p}^{(n-1,t)})$$
$$\mathcal{G}\left(G(d_{1,t}; \psi_{i,1,p}^{(n-1,t)}), \cdots, G(d_{L,t}; \ {}_{i,L,p}^{(n-1,t)}); \lambda_i^{(n-1,t)}\right). \tag{2.58}$$

Remark 5. *A comparison of* (2.30), (2.31), (2.32), (2.37), (2.38), (2.41), (2.43), *with* (2.50), (2.51), (2.52), (2.53), (2.54), (2.55), (2.56), *respectively, reveals the motivation for the definition of the sufficient statistics in* (2.50)-(2.56). *As can be seen the sufficient statistics in* (2.50)-(2.56) *lack the averaging over time. This averaging, however, is performed in* (2.47), (2.48) *and* (2.49).

Let $\epsilon^{(t^*)}$ be a decreasing sequence. Then, using (2.16), the E-step of our proposed online algorithm is given by

$$\begin{aligned}
S_{m,1}^{(n,t^*)} &= (1-\epsilon^{(t^*)}) S_{m,1}^{(N,t^*-1)} + \epsilon^{(t^*)} E_{\zeta_{i,p,t^*}^{(n)}|\mathbf{d}_{t^*};\Theta^{(n-1,t^*)}}[s_{m,1}^{(n)}(\boldsymbol{\zeta}_{t^*}^{(n)})],\ m \in \{\mathcal{G}, \mathcal{T}\}\ , \\
S_j^{(n,t^*)} &= (1-\epsilon^{(t^*)}) S_j^{(N,t^*-1)} + \epsilon^{(t^*)} E_{\zeta_{i,p,t^*}^{(n)}|\mathbf{d}_{t^*};\Theta^{(n-1,t^*)}}[s_j^{(n)}(\boldsymbol{\zeta}_{t^*}^{(n)})],\ j = 2, \cdots, 5, \\
\boldsymbol{\mathcal{S}}^{(n,t^*)} &= (1-\epsilon^{(t^*)}) \boldsymbol{\mathcal{S}}^{(N,t^*-1)} + \epsilon^{(t^*)} E_{\zeta_{i,p,t^*}^{(n)}|\mathbf{d}_{t^*};\Theta^{(n-1,t^*)}}[\boldsymbol{\mathcal{J}}^{(n)}(\zeta_{i,p,t^*}^{(n)})].
\end{aligned} \tag{2.59}$$

The M-step of the proposed online algorithm does not change and consists of the update functions

$$\omega_{i,p,t^*}^{(n)} = \overline{\omega}\left(\alpha^{(n)}(i,p,t^*)\right), \tag{2.60}$$

$$\boldsymbol{\mu}_{i,p}^{(n,t^*)} = \overline{\mu}(S_3^{(n,t^*)}, S_4^{(n,t^*)}), \tag{2.61}$$

$$\sigma_{i,l,p}^{(n,t^*)} = \begin{cases} \overline{\sigma}_{\mathcal{G}}(S_4^{(n,t^*)}, \beta_{i,l,p}^{n,t^*}, \gamma_{i,l,p}^{n,t^*}), \ m = \mathcal{G}, \\ \overline{\sigma}_m(S_4^{(n,t^*)}, S_5^{(n,t^*)}), \ m \in \{\mathcal{T}, \mathcal{P}\} \ , \end{cases} \tag{2.62}$$

when $m \in \mathcal{M}$, and

$$\lambda_{m,i}^{(n,t^*)} = \overline{\lambda}_m \left(S_{m,1}^{(n,t^*)}, S_2^{(n,t^*)} \right), \tag{2.63}$$

for $m \in \{\mathcal{G}, \mathcal{T}\}$. In (2.62),

$$\beta_{i,l,p}^{(n,t^*)} = \overline{\beta} \left(\boldsymbol{\mathcal{S}}_{\setminus l}^{(n,t^*)}, (\lambda_i^{(n-1,t^*)})_{\setminus l} \right), \tag{2.64}$$

$$\gamma_{i,l,p}^{(n,t^*)} = \overline{\gamma} \left(S_4^{(n,t^*)}, (\lambda_i^{(n-1,t^*)})_{l,l} \right). \tag{2.65}$$

For each time instant t, we need to initialize the algorithm for the first iteration ($n = 1$) of the N mini-iterations. Therefore, for $i = 0, \cdots, K-1$, $p = 1, \cdots, P$, and $l = 1, \cdots, L$, we let $\omega_{i,p,t^*}^{(n-1)} = \omega_{i,p,t^*}^{(0)} = \frac{1}{KP}$, $\underset{i,l,p}{(n-1,t^*)} = \underset{i,l,p}{(0,t^*)} = \underset{i,l,p}{(N,t^*-1)}$, and $\lambda_i^{(n-1,t^*)} = \lambda_i^{(0,t^*)} = \lambda_i^{(N,t^*-1)}$.

At each time instant t^*, at the end of the N mini-iterations, we compute $\phi_{i,t^*} = \sum_{p=1}^{P} \omega_{i,p,t^*}^{(N)}$. The detection rule then decides $\mathcal{H}_{i^*}$ as the state of nature at t^* where $i^* = \underset{0 \le i \le K-1}{\text{argmax}} \ \phi_{\text{i,t}^*}$. The entire procedure for the estimation of the parameter set and the detection of the hypotheses is summarized in Algorithm 1.

2.4 Numerical Results

In this section we present numerical results from both simulation data and two real-world datasets to verify the efficacy of the proposed method.

In *Step 1* of the algorithm we set the initial values of the probabilities $\omega_{i,p,t} = \frac{1}{KP} = \frac{1}{4}$. The initial values of the copula parameters $\lambda_{m,i}$ are chosen to be the $L \times L$ identity matrix for $m \in \{\mathcal{G}, \mathcal{T}\}$. The initial values of $\mu_{i,l,p}$ and $\sigma_{i,l,p}$ are obtained from the unsupervised method of K-means. Also the number of mini-iterations is set to $N = 4$. Finally in

Data: $\mathbf{d}_t$; sensor measurements' at time instance $t > T_0$.
Result: online updated of Θ and detection of $\mathbf{h}_t$.

begin

Step1: initialization:

Assume an initial value for Θ as follows:

Set $\tilde{\omega}_{i,p,t}^{(0,1)} = \frac{1}{KP}$,

Set $\lambda_{m,i}^{(0,1)} = I_L, m \in \{\mathcal{G}, \mathcal{T}\}$,

Apply K-means to $\mathbf{d}_{1:T_0}$ and set $\overset{(0,1)}{i,p,l}$ as cluster means of the K-means;

Apply batch EM on $\mathbf{d}_{1:T_0}$ to compute $\Theta^{(N,T_0)}$;

Step2: online updates:

while $\boldsymbol{d}_t$ *is received,* $t > T_0$ **do**

initialize parameters using $\Theta^{(0,t)} = \Theta^{(N,t-1)}$;

for $1 \leq n \leq N$ **do**

online E Step:

Find $\alpha^{(n)}(i,p,t)$ using (2.57);

Update S_j, $S_{m,1}$, $\boldsymbol{\mathcal{S}}$ for $m \in \{\mathcal{G}, \mathcal{T}\}$ and $2 \leq j \leq 5$ using (2.47)-(2.49);

online M Step:

Update $\omega_{i,p,t}^{(n)}$ using (2.60),

Update $\lambda_{m,i}^{(n,t)}$, $m \in \{\mathcal{G}, \mathcal{T}\}$ using (2.63),

Calculate $\beta_{i,l,p}^{(n,t)}$, $\gamma_{i,l,p}^{(n,t)}$ using (2.64), (2.65),

Update $\boldsymbol{\mu}_{i,p}^{(n,t)}$ and $\sigma_{i,l,p}^{(n,t)}$ using (2.61) and (2.62),

end

Compute $\phi_{i,t} = \sum_{p=1}^{P} \omega_{i,p,t}^{(N)}$;

Calculate $i^* = \underset{0 \leq i \leq K-1}{\text{argmax}}\ \phi_{\text{i,t}}$;

Set $\mathbf{h}_t = \mathbf{e}_{i^*}$.

end

end

Algorithm 1: Online parameter estimation and hypothesis detection.

the stochastic approximation of the E-step, we use $\epsilon^{(t^*)} = \frac{1}{t^*}$. This sequence satisfies the sufficient condition for the convergence of the online algorithm [77]. We should point out that in the online processing of all our simulations, by $t = 2000$, all the estimated parameters have converged to a relative distance of less than 0.1 from their actual values.

To evaluate the detection performance of the algorithms, we define the metric *hypothesis*

discriminability $\Delta_H(t)$ given by

$$\Delta_H(t) \triangleq \frac{1}{Kt} \sum_{i=0}^{K-1} \sum_{\tau=1}^{t} \left| h_{i,\tau} - \hat{h}_{i,\tau} \right|. \tag{2.66}$$

We present *hypothesis discriminability* for each of the models based on the Gaussian, Student's t, and Product copulas, denoted by MBGC, MBSC, and MBPC, respectively. We compare their performances with the case where the model parameters are completely known. Since the latter provides the best possible detection performance, it is referred to as the lower bound (LB).

2.4.1 Simulations

In the simulations, an online binary hypothesis testing problem is considered, i.e., $K = 2$. We assume that the measurement data are collected under two different environmental conditions, i.e., $P = 2$. To model the correlation among the sensors' data, we consider two cases of sensors placement, a 1D array and a 2D grid. The first case, referred to as case (a) herein, is used in applications such as traffic monitoring where the sensors may be positioned along a road. The second case, referred to as case (b), is used in applications such as disaster management and precision agriculture. In these cases the data collected by neighboring sensors are highly correlated, but as the euclidean distance between two sensors increases the correlation in their measurements decreases. Let $\rho(l_2, l_1)$ denote the distance between two sensors l_1 and l_2. The correlation between the two sensors' data denoted by $(\lambda_{m,i})_{l_1,l_2}$ is assumed to be given by [58, 59],

$$(\lambda_{m,i})_{l_1,l_2} = \exp\{-\rho(l_2, l_1)\}. \tag{2.67}$$

Once the simulated data is generated, we run the proposed online method as described in Algorithm 1. In all our simulations, for *Step 1* of the proposed algorithm, where the batch-mode EM is executed to initialize the online EM, the number of time samples is set

to $T_0 = 10L$.

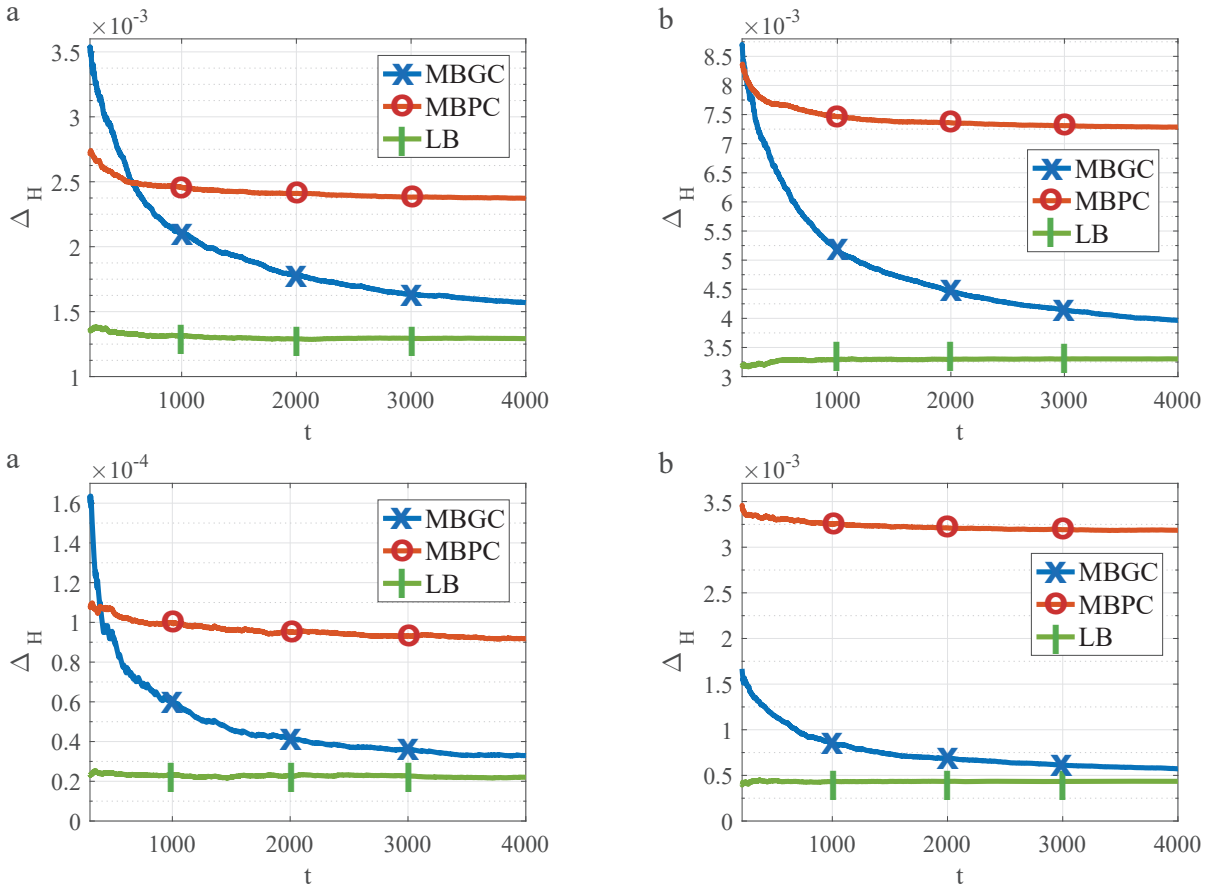

Figure 2.1: Hypothesis discriminability versus the number of samples t for MBGC, MBPC and LB. Top: $L = 10$ sensors. Bottom: $L = 20$ sensors.

Hypothesis discriminability Δ_H as a function of the number of samples t is shown in Fig. 2.1 for the three algorithms MBGC, MBPC, and LB, for both cases (a) and (b) for $L = 10$ and $L = 20$. It can be seen that as the new data samples arrive, the performance of the proposed online algorithm improves significantly. Moreover, MBPC which ignores the correlation in the data has a significant performance loss compared with the proposed correlation based method. Moreover, as t increases, the advantage of MBGC over MBPC improves. In addition, the performance of MBGC converges to that of the lower bound LB which has perfect knowledge of the underlying model parameters. For example, in case (a), for $t = 4000$, $L = 20$, LB is only 17% better than MBGC. We would like to reiterate that the difference between MBGC and LB is that, while LB only attempts to detect the

state of nature, MBGC must estimate the model parameters as well as detect the state of nature.

Clearly the computational complexity of MBGC and MBSC are higher than that of MBPC and LB. In particular, the computational complexity of MBPC is $O(NPML)$. On the other hand the complexity of MBGC and MBSC which needs to estimate the correlation matrices $\lambda_{\mathcal{G},i}$ and $\lambda_{\mathcal{T},i}$ are both $O(NPML^4)$. In practice, the number of mini-iterations N, the number of distributions P and the number of hypotheses M are not very large. Therefore the computational complexity of MBGC and MBSC are mostly determined by the number of sensors L. Thus for networks consisting of a large numbers of sensors, MBGC and MBSC become computationally expensive.

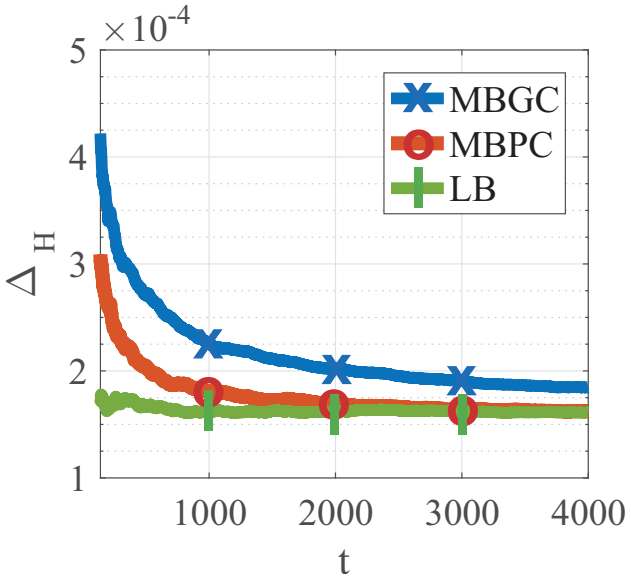

Figure 2.2: Hypothesis discriminability of MBGC, MBPC and LB versus t, for $L = 5$, and independent data.

When the sensors' measurements are independent, MBPC is the appropriate model for the data and other copula-based methods do not match the nature of the sensors' data. In order to determine the performance loss due to this mismatch, in Fig. 2.2 we present the hypothesis discriminability for MBGC, MBPC, and LB when the data from $L = 5$ sensors are independent. It can be seen that the performance of MBPC converges to that of LB and that given enough data samples, the performance of MBGC is only slightly worse than that

of LB. This indicates that given enough data samples, MBGC can correctly estimate the correlation matrix as the identity matrix and achieve similar performance as MBPC. Also note that since the sensors' measurements are independent, they carry more information than when they are correlated. As a result hypothesis discriminability is lower for $L = 5$ sensors than the case of $L = 10$ sensors in Fig. 2.1.

Hitherto we have assumed that if the sensor measurements are samples from P different marginal PDFs, the algorithm also assumes $P^* = P$ different marginals. If the algorithm assumes $P^* > P$ marginals, there will not be any performance loss. The algorithm will resolve this over-fitting and will evaluate only P different parameters for the marginal PDFs. On the other hand, when $P^* < P$, there will be a performance loss due to the under-fitting of the data. In Table 2.1 we present the minimum achievable value of Δ_H for this case when the sensor measurements are samples from $P = 2$ different marginal PDFs, while the algorithm assumes $P^* \in \{1,\ 2,\ 3,\ 4,\ 5\}$ marginal PDFs. Comparing the results for $P^* = 1$ with the case that $P^* = 2$ shows that, due to this mismatch, Δ_H increases by an order of magnitude while Δ_H for $P^* > 2$ is similar to when $P^* = 2$. This shows that when the actual value of P is unknown, one should over estimate it and select a larger value for the algorithm. This increases the complexity of the algorithm, somewhat, but would result in better performance.

Table 2.1: Minimum hypothesis discriminability for $P = 2$ and $L = 20$

P^*	1	2	3	4	5
Δ_H for case (a)	$2e-4$	$3e-5$	$4e-5$	$5e-5$	$5e-5$
Δ_H for case (b)	$3e-3$	$6e-4$	$7e-4$	$8e-4$	$9e-4$

In Fig. 2.3, Δ_H for case (a) is plotted versus the number of sensors for MBGC, MBPC and LB. The number of samples is $t = 4000$. Note that for this number of samples, Δ_H for all three algorithms has reached its floor value. Fig. 2.3 shows that MBGC's performance is close to that of LB and shows a clear advantage over MBPC which ignores the correlation in the data.

The figure also shows that as the number of sensors increases, the advantage of MBGC

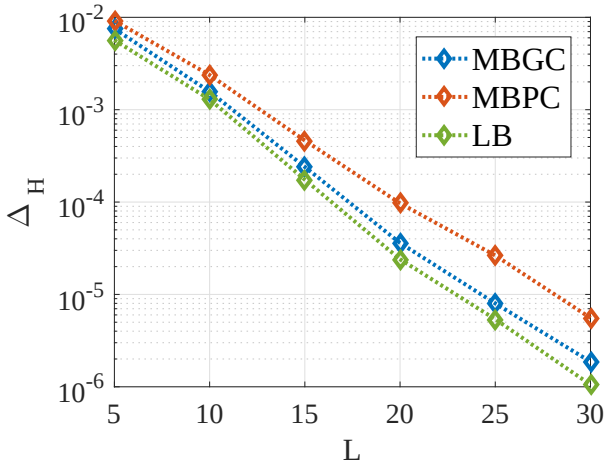

Figure 2.3: Hypothesis discriminability versus L, for $t = 4000$ samples for MBGC, MBPC and LB and case (a).

over MBPC increases. The figure also shows a trade-off in the number of sensors vs. the complexity of the fusion algorithm. For example Δ_H for MBGC with $L = 20$ sensors is the same as Δ_H for MBPC with $L = 25$ sensors. It should be noted however, that in real world scenarios, it may not always be possible to increase the number of sensors. For example in multimodal sensing where sensors measure a limited number of parameters. We also observe that as L increases, the performance of MBGC is degraded slightly compared to that of LB. The reason is that for each time instant, MBGC must estimate $4+4L+4L+2L^2$ parameters [4] , as a result, a good estimation of these parameters becomes more difficult and requires more data samples.

In Fig. 2.4, we present Δ_H for MBSC, MBPC and LB for $L = 6$ sensors for both cases (a) and (b). Similar conclusions as in the case of MBGC can be drawn for the MBSC results.

To demonstrate the accuracy of parameter estimations using the IFM method, we evaluate the estimation errors of the mean and standard deviation for the Student's t and

[4]This corresponds, respectively, to $\{\omega_{i,p,t}\}, \{\boldsymbol{\mu}_{i,p}\}, \{\sigma_{i,l,p}\}, \{\lambda_{\mathcal{G},i}\}$.

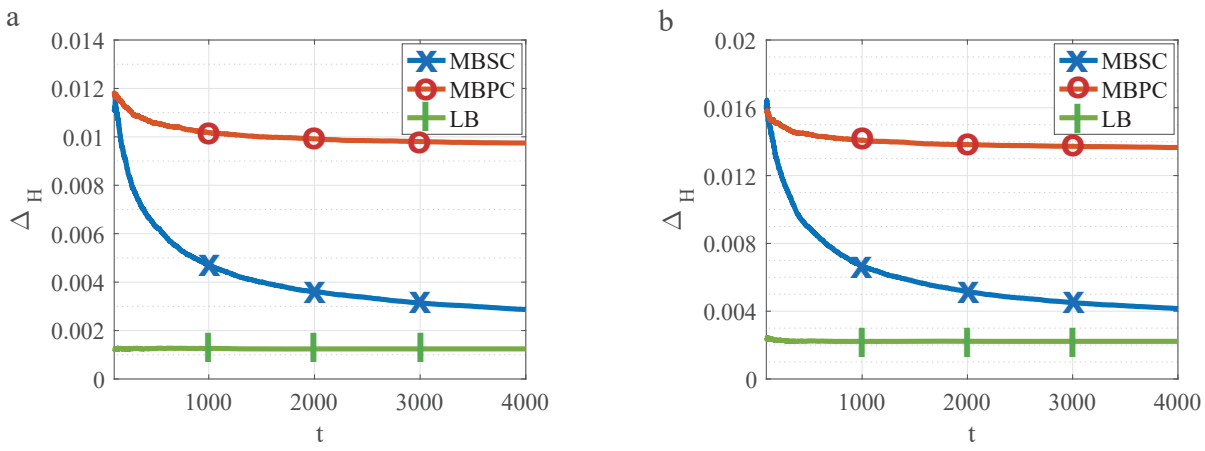

Figure 2.4: Hypothesis discriminability versus the number of samples t for MBSC, MBPC and LB, $L = 6$ sensors.

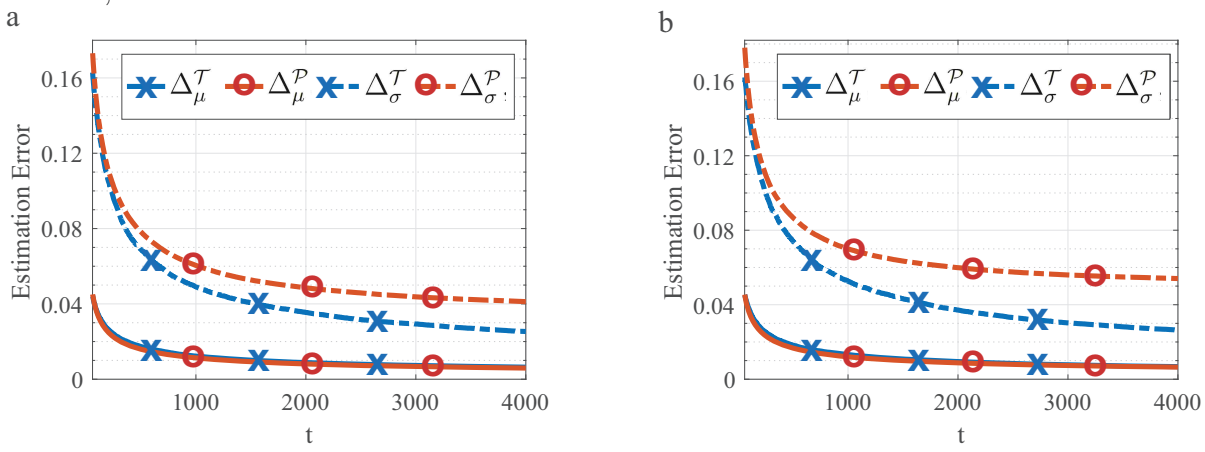

Figure 2.5: Estimation errors, $\Delta_\mu^{\mathcal{T}}$, $\Delta_\mu^{\mathcal{P}}$, and $\Delta_\sigma^{\mathcal{T}}$, $\Delta_\sigma^{\mathcal{P}}$, versus the number of samples t for MBSC, MBPC and LB, $L = 6$ sensors.

product copulas ($m \in \{\mathcal{P}, \mathcal{T}\}$) using

$$\Delta_\mu^m(t) = \frac{1}{KPL} \sum_{i=0}^{K-1} \sum_{p=1}^{P} \sum_{l=1}^{L} \left| \frac{\mu_{i,l,p}^{(N,t)} - \mu_{i,l,p}^{\text{Actual}}}{\mu_{i,l,p}^{\text{Actual}}} \right|, \tag{2.68}$$

$$\Delta_\sigma^m(t) = \frac{1}{KPL} \sum_{i=0}^{K-1} \sum_{p=1}^{P} \sum_{l=1}^{L} \left| \frac{\sigma_{i,l,p}^{(N,t)} - \sigma_{i,l,p}^{\text{Actual}}}{\sigma_{i,l,p}^{\text{Actual}}} \right|. \tag{2.69}$$

The results presented in Fig. 2.5, for $L = 6$ indicate the accuracy of parameter estimation and that parameter estimation errors are smaller for MBSC. Moreover, as t increases the

improvement of MBSC over MBPC increases.

2.4.2 Numerical Results for Real Data

We also evaluate the performance of the proposed algorithm using two real-world datasets, namely, the *Room Occupancy Detection* (ROD) [2] and *Activity Recognition based on Multisensor data fusion* (AReM) [8] datasets, both available at *https://archive.ics.uci.edu*. The ROD dataset consist of temperature, humidity, light, and CO2 sensory data used for binary hypothesis testing where $\mathcal{H}_0$ represents an unoccupied room and $\mathcal{H}_1$ represents an occupied room. In this dataset, ground-truth occupancy was obtained from time stamped pictures that were taken every minute. The AReM dataset contains data collected from a wireless SN worn by an actor with the purpose of detecting the actor's daily activities. Here, we consider three activities, bending, cycling, and lying down, which correspond to $\mathcal{H}_i$ for $i = 0, 1, 2$, respectively. In this dataset, $L = 6$, i.e., there are 6 streams of data over time which can be fused to detect $\mathcal{H}_i$.

We define $1 - \Delta_H$ as the *Detection Accuracy* (DA) and compare the performance of the proposed method with other well-known supervised and unsupervised methods in terms of DA. We consider both MBPC and MBGC for the proposed method. However, note that neither of these two copulas, perfectly match the correlation structure in the data. For the AReM dataset, we consider $T_0 = 1500$ and for the ROD dataset, we consider $T_0 = 2000$. The supervised methods include Support Vector Machines (SVM) and K-Nearest Neighbor (KNN). As for the unsupervised methods with which we compare the proposed method, for the AReM dataset we consider the Kmeans method, and for the ROD dataset, we consider the Page-Hinkley Test (PHT) and the Geometric Moving Average (GeoMA) which are two unsupervised methods devised specifically for room occupancy detection problems [1]. We should point out that these unsupervised methods all use batch-mode processing. To train the supervised learning algorithms, a training dataset of 8144 samples is used for the ROD dataset. As for the AReM dataset, 70% of the data samples are used to train the supervised algorithms and the remaining 30% are used for testing.

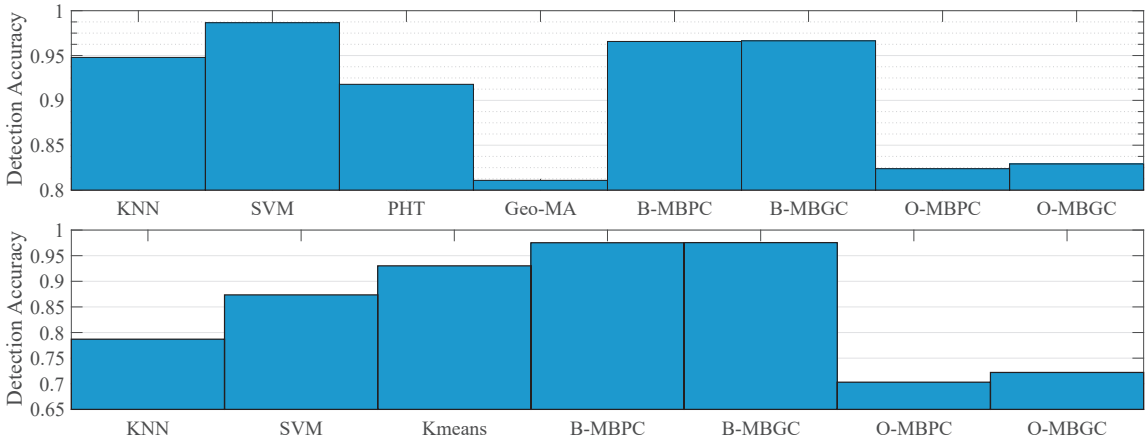

Figure 2.6: DA for different methods using real-world datasets. Top: ROD dataset (M=2, L=3). Bottom: AReM dataset (M=3, L=6).

In Fig. 2.6, DA is presented for 9000 and 8460 testing samples of the ROD and the AReM datasets, on the top and bottom rows respectively. In this figure, B-MBGC and O-MBGC denote the batch and online modes of MBGC, respectively. Similarly, B-MBPC and O-MBPC denote the batch and online modes for MBPC, respectively. Figure 2.6 shows that the proposed batch-mode algorithm has higher DA than other unsupervised and even some supervised methods. The DA of the proposed algorithm for online processing is worse than its DA in batch-mode. However, we know that as T_0 increases, DA of the proposed online algorithm increases converging to DA of the proposed batch mode algorithm. For example for the ROD dataset when T_0 increases from 2000 samples to 4000 samples, DA values of O-MBPC and O-MBGC increase from 0.824 and 0.829 (reported in Fig. 2.6) to 0.894 and 0.905, respectively. Moreover, as discussed before, online processing is inevitable in many practical scenarios.

In Fig. 2.7, the actual state of nature at each time instance of the ROD dataset is plotted as well as the detection results achieved at each time instance with different supervised and unsupervised methods (SVM, PHT, Geo-MA, B-MBGC, and O-MBGC) using the ROD dataset.

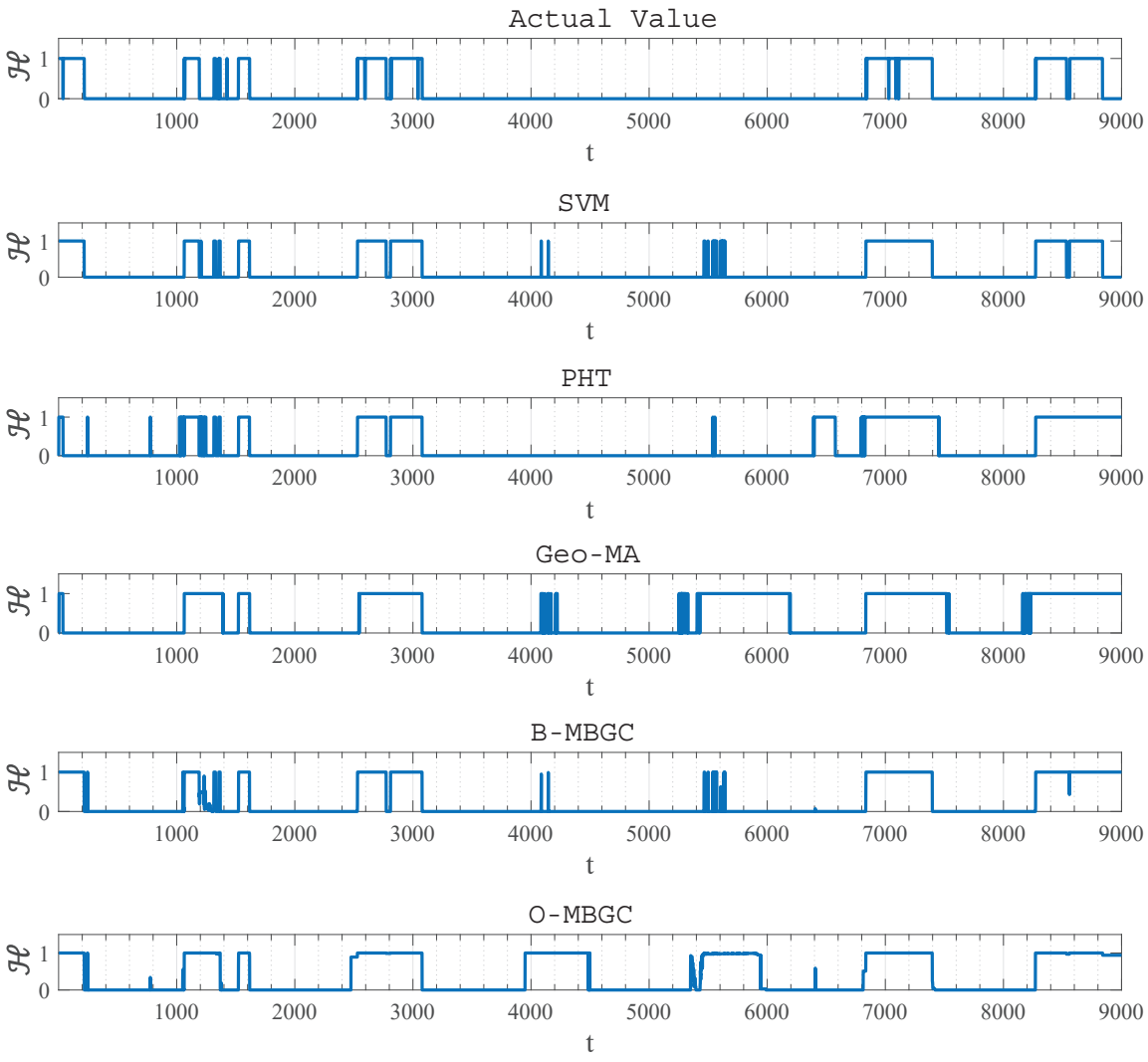

Figure 2.7: The actual (top row) and estimated values of the state of nature at each time instance using different methods (SVM, PHT, Geo-MA, B-MBGC, and O-MBGC) for the ROD dataset.

2.5 Conclusion

An online expectation maximization (EM) based algorithm is presented for data fusion involving model parameter estimation and hypothesis testing based on observations from a network of heterogeneous sensors. The sensor measurements are assumed to be correlated and copula theory is used to model this correlation. Moreover, it is assumed that the statistical model for the sensor data is not completely known.

The batch-mode EM is first developed for case studies of this problem including the Gaussian, Student's t, and product copulas where model parameters are estimated and the

state of nature is detected at all time instances. This algorithm is then extended to an online EM based approach. In the online method, upon receiving sensors' measurements at each time instance, the model parameters are updated and the state nature at the current time is detected.

Results obtained from both simulation and real-world data show significant improvements in hypothesis testing compared to other unsupervised and even some supervised learning methods. Moreover, in the case where data are correlated, the proposed method including copula modeling outperforms the method ignoring the correlation in sensors' measurements while in the case where the data are independent, given enough data samples, the performance of the proposed method converges to that of the method which correctly assumes an independent data model.

Chapter 3
Hypothesis Testing and Parameter Estimation with Observations Correlated Both Among Sensors and Over Time

3.1 Introduction

In this chapter we consider centralized detection in a sensor network consisting of heterogeneous sensors whose received data,under each hypothesis, are drawn from different marginal PDFs. It is assumed that the data received from the sensors are dependent over time and among the sensors. Moreover, we consider the scenario in which the joint distribution (and by extension, the marginal distributions) of the sensor observations are not completely known.

In many applications (e.g., biomedical or object tracking) there is non-negligible dependence among the samples collected by each sensor, which if ignored, can result in significant degradation in the performance of the detection scheme. Our goal is to model the dependent observations of the sensors and devise a data fusion algorithm to detect the state of nature. To model the dependence in the data collected by different sensors and over time, the copula theory and Markov chains are employed. There are many advantages in employing the copula distribution for modeling the distribution of dependent data. One of the most important is that copulas separate the effect of the marginal distributions and the dependence structure in the data [60]. Therefore, in a copula-based detection method, changes in both complementary and mutual information can be detected due to this inherent decoupling in the copula theory. Furthermore, this feature of the copula opens up a lot of opportunities in statistical modeling by allowing us to model nonlinear dependencies or to represent joint PDF models that do not necessarily have a closed form.

The problem of binary hypothesis testing and dependent data fusion based on the LLRT has been considered in [18] where the face matching results from two different face matching algorithms are combined to detect the identity of an individual. Using the LLRT and the GLRT, fusion of dependent decisions of sensors has been recently studied in [60, 61] where

it is further assumed that the copula function may be mis-specified.To improve the computational complexity of fusing discrete decisions using GLRT, the authors have proposed to inject noise into the local sensor decisions which, as a result, decreases the signal-to-noise ratio of the quantized data. In [18, 60, 61], at each time instant the observations of all the sensors are assumed to be dependent. However, dependence of data samples over time is not investigated since the observation samples of each sensor are assumed to be independent and identically distributed (iid) over time.

In this chapter, we develop a method based on the EM algorithm to estimate the unknown parameters of the underlying joint and marginal PDFs, and to detect the hypotheses at each time. While parameter estimation in the presence of latent variables has been studied extensively before [71], parameter estimation in this dissetation is in the context of hypothesis testing which has not been previously investigated in the case of our general data models.

The rest of this chapter is organized as follows. In section 3.2, the problem is formulated and a probabilistic model is derived for the system under consideration. In section 3.3, the proposed EM-based algorithm is described to solve the estimation and detection problem. In section 3.4, we investigate a case study including the Gaussian and Student's t copulas. In section 3.5, our simulation method is presented, and in section 3.6, simulation results are presented and discussed. Finally, the chapter ends with a conclusion in section 3.7.

3.2 Problem Formulation and System Model

We consider a sensor network consisting of L heterogeneous sensors employed to detect the state of nature $\mathcal{H} \in \{\mathcal{H}_0, \mathcal{H}_1\}$. By heterogeneous sensors we mean that each sensor's measurement follows a different parametric distribution. At time t, sensor l transmits its measurement, denoted by $d_{l,t} \in \Re$ to the FC. After T time instances, the FC has received LT measurements from all the sensors which we denote by the $L \times T$ matrix $D = [d_{l,t}]$. Two variables $h_{0,t}$ and $h_{1,t}$ are used to denote the state of nature at time t where $h_{i,t} \in \{0, 1\}$ for $i = 0, 1$. Here $h_{0,t} = 1 - h_{1,t}$ and $h_{i,t} = 1$ indicates that the state of nature at time t is

$\mathcal{H}_i$. For the entire observation period we construct a $2 \times T$ matrix $H = [h_{i,t}]$ which we call the hypothesis matrix.

Regarding the dependency of observations in time, we assume that if the state of nature at time t is different from that at time $t-1$, i.e., $h_{i,t} \neq h_{i,t-1}$, then the sensors' observations at time t are independent from the observations at time $t-1$. However, if the state of nature does not change from $t-1$ to t, i.e., $h_{i,t} = h_{i,t-1}$, then the observations are dependent in time and follow a Markovian model. More specifically, we assume that given $d_{l,t-1}$, $d_{l,t}$ is conditionally independent of all the past measurements of all the sensors. Therefore, given that $h_{i,t} = 1 \neq h_{i,t-1}$[1], we denote the conditional distribution function of $d_{l,t}$ (given H) by $F_{i,l}(d_{l,t}|H;\psi_{i,l})$, where $\psi_{i,l}$ is the set of unknown parameters of the, otherwise known, distribution function $F_{i,l}(d_{l,t}|H;\psi_{i,l})$. On the other hand, if $h_{i,t} = 1 = h_{i,t-1}$[2], we denote the conditional distribution function of $d_{l,t}$ given $d_{l,t-1}$ (and H) by $F_{i,l}(d_{l,t}|d_{l,t-1},H;\tilde{\psi}_{i,l})$, where $\tilde{\psi}_{i,l}$ denotes the set of unknown parameters of the, otherwise known, distribution function $F_{i,l}(d_{l,t}|d_{l,t-1},H;\tilde{\psi}_{i,l})$.

Regarding the dependency of observations among the sensors, two cases are considered. First, when the state of nature changes at time t, the dependency among sensors' measurements is modeled by the copula distribution given by

$$F(\mathbf{d}_t|H;\Psi,\lambda_{1,i}) = C_1\left(F_{i,1}(d_{1,t}|H;\psi_{i,1}),\cdots,F_{i,L}(d_{L,t}|H;\psi_{i,L});\lambda_{1,i}\right) \tag{3.1}$$

where it is assumed that the hypothesis at t is $\mathcal{H}_i$, $\mathbf{d}_t = [d_{1,t}, d_{2,t}, \cdots, d_{L,t}]^{Tr}$, where superscript Tr denotes transpose, and $\Psi = \{\psi_{i,l}\}$ is the collection of unknown distribution parameters. Moreover, $\lambda_{1,i}$, $i = 0, 1$ denotes the set of unknown parameters of the copula distribution $C_1(.)$ under the hypothesis $\mathcal{H}_i$. We denote $\Lambda_1 \triangleq \{\lambda_{1,0}, \lambda_{1,1}\}$. Next, when the state of nature does not change at time t, the dependency of sensors' measurements is

[1]This implies that the state of nature at time t is $\mathcal{H}_i$ and is different from the state at time $t-1$.
[2]This implies that the state of nature at time t is $\mathcal{H}_i$ and is the same as the state at time $t-1$.

modeled by a different copula distribution given by

$$F(\mathbf{d}_t|\mathbf{d}_{t-1}, H; \tilde{\Psi}, \lambda_{2,i}) =$$
$$C_2\left(F_{i,1}(d_{1,t}|d_{1,t-1}, H; \tilde{\psi}_{i,1}), \cdots, F_{i,L}(d_{L,t}|d_{L,t-1}, H; \tilde{\psi}_{i,L}); \lambda_{2,i}\right) \tag{3.2}$$

where again, it is assumed that the hypothesis at time t is $\mathcal{H}_i$, and $\tilde{\Psi} = \{\tilde{\psi}_{i,l}\}$ denotes the collection of unknown distribution parameters. Moreover, $\lambda_{2,i}$, $i = 0, 1$ denotes the set of unknown parameters of the copula distribution $C_2(.)$ under the hypothesis $\mathcal{H}_i$. We denote $\Lambda_2 \triangleq \{\lambda_{2,0}, \lambda_{2,1}\}$. For the two copula distributions $C_1(.)$ and $C_2(.)$ it is assumed that their distribution types are known a priori. However, their parameters Λ_1, Λ_2, are unknown and need to be estimated. Thus, the set of all unknown parameters can be denoted by $\widetilde{\Theta} = \{\Psi, \widetilde{\Psi}, \Lambda_1, \Lambda_2\}$.

Taking the derivative of the distribution functions in (3.1) and (3.2), we obtain the corresponding PDFs. In order to unify our notations and to show the dependence of these quantities on the set of parameters, we denote these PDFs by $P(\mathbf{d}_t|H; \tilde{\Theta})$ and $P(\mathbf{d}_t|, \mathbf{d}_{t-1}, H; \tilde{\Theta})$, respectively, as

$$P(\mathbf{d}_t|H; \tilde{\Theta}) = \left(\prod_{l=1}^{L} f_{i,l}(d_{l,t}; \psi_{i,l})\right)$$
$$c_1\left(F_{i,1}(d_{1,t}; \psi_{i,1}), F_i^{(2)}(d_{2,t}; \psi_i^{(2)}), \cdots, F_{i,L}(d_{L,t}; \psi_{i,L})\right), \tag{3.3}$$

and

$$P(\mathbf{d}_t|\mathbf{d}_{t-1}, H; \tilde{\Theta}) = \tag{3.4}$$
$$\left(\prod_{l=1}^{L} f_{i,l}(d_{l,t}|d_{l,t-1}; \tilde{\psi}_{i,l})\right) c_2\left(F_{i,1}(d_{1,t}|d_{1,t-1}; \tilde{\psi}_{i,1}), \cdots, F_{i,L}(d_{L,t}|d_{L,t-1}; \tilde{\psi}_{i,L})\right),$$

where $c_j(.)$ represented the derivative of $C_j(.)$ for $j = 1, 2$. The joint PDF of $(\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_T)$

can now be written as

$$P(\mathbf{d}_1, \cdots, \mathbf{d}_T|H;\tilde{\Theta}) = \prod_{t=1}^{T}\prod_{i=0}^{1} P(\mathbf{d}_t|H;\tilde{\Theta})^{h_{i,t}h_{1-i,t-1}} P(\mathbf{d}_t|, \mathbf{d}_{t-1}, H;\tilde{\Theta})^{h_{i,t}h_{i,t-1}} \tag{3.5}$$

It should be noted that in (3.5), at each time t only one of the two cases of time-dependent or time-independent holds. This is ensured by the exponents $h_{i,t}h_{i,t-1}$ and $h_{i,t}h_{1-i,t-1}$. Using (3.3) and (3.4) in (3.5) and replacing the measurement matrix, D, for $(\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_T)$ we get

$$\begin{aligned} P(D|H;\tilde{\Theta}) = &\prod_{t=1}^{T}\prod_{i=0}^{1} \\ &\left[\left(\prod_{l=1}^{L} f_{i,l}(d_{l,t};\psi_{i,l})\right) c_1\left(F_{i,1}(d_{1,t};\psi_{i,1}), \cdots, F_{i,L}(d_{L,t};\psi_{i,L});\lambda_{1,i}\right)\right]^{h_{i,t}h_{1-i,t-1}} \times \\ &\left[\left(\prod_{l=1}^{L} f_{i,l}(d_{l,t}|d_{l,t-1};\tilde{\psi}_{i,l})\right)\right. \\ &\left. c_2\left(F_{i,1}(d_{1,t}|d_{1,t-1};\tilde{\psi}_{i,1}), \cdots, F_{i,L}(d_{L,t}|d_{L,t-1};\tilde{\psi}_{i,L});\lambda_{2,i}\right)\right]^{h_{i,t}h_{i,t-1}} \end{aligned} \tag{3.6}$$

Having received the measurement matrix D, the FC desires to detect the state of nature during each time instant in the observation period T. In order to effectively accomplish this task, the FC must also estimate the unknown parameters $\tilde{\Theta}$. To this end we assign probabilities $\phi_{0,t}$ and $\phi_{1,t} = 1 - \phi_{0,t}$ to the states $\mathcal{H}_0$ and $\mathcal{H}_1$ at time t, respectively, and for $i, j = 0, 1$ we define $\tilde{\phi}_{i,j,t} = P(h_{i,t} = 1, h_{j,t-1} = 1)$. Note that these are not prior probabilities and are only used as a tool to help us decide on the state of $\mathcal{H}$ at time t. In the following sections, we estimate $\tilde{\phi}_{i,i,t}$ and $\tilde{\phi}_{i,1-i,t}$, and then using the fact that $\phi_{0,t} = \tilde{\phi}_{0,0,t} + \tilde{\phi}_{0,1,t}$ and $\phi_{1,t} = \tilde{\phi}_{1,0,t} + \tilde{\phi}_{1,1,t}$, the values of $\phi_{0,t}$, and $\phi_{1,t}$ are calculated. The state of nature is then estimated to be $\mathcal{H}_0$ if $\phi_{0,t} > \phi_{1,t}$, and $\mathcal{H}_1$, otherwise[3]. It can be observed that since $\phi_{0,t} + \phi_{1,t} = 1$, then $\sum_{i=1}^{0}\sum_{j=1}^{0} \tilde{\phi}_{i,j,t} = 1$ should hold.

[3]In this way the detection of the hypotheses is transformed into an estimation problem for $\Phi \triangleq [\phi_{i,t}]$ which we can be solved using the EM algorithm.

Let

$$\tilde{\Phi} \triangleq \begin{bmatrix} \tilde{\phi}_{0,0,1} & \tilde{\phi}_{0,0,2} & \cdots & \tilde{\phi}_{0,0,T} \\ \tilde{\phi}_{0,1,1} & \tilde{\phi}_{0,1,2} & \cdots & \tilde{\phi}_{0,1,T} \\ \tilde{\phi}_{1,0,1} & \tilde{\phi}_{1,0,2} & \cdots & \tilde{\phi}_{1,0,T} \\ \tilde{\phi}_{1,1,1} & \tilde{\phi}_{1,1,2} & \cdots & \tilde{\phi}_{1,1,T} \end{bmatrix} \tag{3.7}$$

denote the $4 \times T$ joint hypothesis probability matrix. The complete unknown parameter set is now defined by $\Theta \triangleq \left\{\widetilde{\Phi}, \Psi, \widetilde{\Psi}, \Lambda_1, \Lambda_2, \right\}$.

The maximum likelihood estimation of Θ from D is given by $\hat{\Theta} = \arg\max_{\Theta} P(D|\Theta)$. However, the distribution $P(D|\Theta)$ is not directly available and can only be obtained from $P(D|\Theta) = \sum_H P(D, H|\Theta)$. Using (3.6), the probability, $P(D, H|\Theta)$, is given by

$$\begin{aligned} P(D, H|\Theta) = P(D|H;\Theta)P(H|\Theta) = \prod_{t=1}^{T}\prod_{i=0}^{1} \\ \left[\frac{\tilde{\phi}_{i,1-i,t}}{\phi_{1-i,t-1}} c_1\left(F_{i,1}(d_{1,t}), \cdots, F_{i,L}(d_{L,t})\right) \prod_{l=1}^{L} f_{i,l}(d_{l,t})\right]^{h_{i,t}h_{1-i,t-1}} \times \\ \left[\frac{\tilde{\phi}_{i,i,t}}{\phi_{i,t-1}} c_2\left(F_{i,1}(d_{1,t}|d_{1,t-1}), \cdots, F_{i,L}(d_{L,t}|d_{L,t-1})\right) \prod_{l=1}^{L} f_{i,l}(d_{l,t}|d_{l,t-1})\right]^{h_{i,t}h_{i,t-1}}. \end{aligned} \tag{3.8}$$

Hereafter, for the sake of brevity we drop the parameters , $\lambda_{1,i}$, $\lambda_{2,i}$, $\psi_{i,l}$, and $\tilde{\psi}_{i,l}$ from the notations of the PDFs $c_1(F_{i,1}(d_{l,t}), \cdots, F_{i,L}(d_{L,t}); \lambda_{1,i})$ and $c_2(F_{i,1}(d_{l,t}|d_{l,t-1}), \cdots, F_{i,L}(d_{L,t}|d_{L,t-1}); \lambda_{2,i})$, and from the notations of the distribution functions $F_{i,1}(d_{1,t}; \psi_{i,l})$ and $F_{i,1}(d_{1,t}|d_{1,t-1}; \tilde{\psi}_{i,l})$, and their respective PDFs.

In (3.8), we assume that at the starting time, i.e., $t = 1$, the hypothesis is independent of its state in the previous time, namely at $t = 0$. Thus, for $h_{i,1} = 1$, we set $h_{1-i,0} = 1$ and only the first term in the equation holds where $\phi_{1-i,t-1} = 1$ for $t = 1$.

To estimate the parameter set, Θ, we employ the EM algorithm in which the state of nature during the observation period, H, constitutes the latent variable.

3.3 The Proposed EM-Based Algorithm

In order to estimate the parameter set Θ with the EM algorithm, we use the joint conditional PDF of the measurement matrix D and the hypotheses matrix H given the parameter set Θ where we choose H to be the latent variable.

3.3.1 The Expectation Step

In the *expectation step* of EM, the expectation of the log-likelihood function, $L(\Theta; D, H) \triangleq \log P(D, H|\Theta)$, denoted by $Q(\Theta; \Theta^{(n-1)}$, is calculated with respect to H, given the measurement matrix, D, and the recent estimate of the parameter set $\Theta^{(n-1)}$. This is derived in (3.9) and (3.10), where for $i = 0, 1$, $\alpha_1^{(n)}(i,t) \triangleq E[h_{i,t}h_{1-i,t-1}|D; \Theta^{(n-1)}]$ and $\alpha_2^{(n)}(i,t) \triangleq E[h_{i,t}h_{i,t-1}|D; \Theta^{(n-1)}]$. Evaluation of $\alpha_1^{(n)}(i,t)$ and $\alpha_2^{(n)}(i,t)$ is discussed in Appendix B.

$$
\begin{aligned}
&L(\Theta; D, H) = \sum_{t=1}^{T}\sum_{i=0}^{1}\sum_{l=1}^{L} \\
&h_{i,t}h_{1-i,t-1}\left[\frac{1}{L}\log(\frac{\tilde{\phi}_{i,1-i,t}}{\phi_{1-i,t-1}}) + \frac{1}{L}\log c_1\left(F_{i,1}(d_{1,t}), \cdots, F_{i,L}(d_{L,t})\right) + \log f_{i,l}(d_{l,t})\right] \\
&+ h_{i,t}h_{i,t-1}\left[\frac{1}{L}\log(\frac{\tilde{\phi}_{i,i,t}}{\phi_{i,t-1}}) + \frac{1}{L}\log c_2\left(F_{i,1}(d_{1,t}|d_{1,t-1}), \cdots, F_{i,L}(d_{L,t}|d_{L,t-1})\right)\right. \\
&\left.+ \log f_{i,l}(d_{l,t}|d_{l,t-1})\right], \qquad (3.9)
\end{aligned}
$$

$$
\begin{aligned}
&Q(\Theta; \Theta^{(n-1)}) \triangleq E_{H|D;\Theta^{(n-1)}}[L(\Theta; D, H)] = \sum_{t=1}^{T}\sum_{i=0}^{1}\sum_{l=1}^{L} \\
&\alpha_1^{(n)}(i,t)\left[\frac{1}{L}\log\frac{\tilde{\phi}_{i,1-i,t}}{\tilde{\phi}_{1-i,i,t-1} + \tilde{\phi}_{1-i,1-i,t-1}} + \frac{1}{L}\log c_1\left(F_{i,1}(d_{1,t}), \cdots, F_{i,L}(d_{L,t})\right) + \log f_{i,l}(d_{l,t})\right] \\
&+ \alpha_2^{(n)}(i,t)\left[\frac{1}{L}\log\frac{\tilde{\phi}_{i,i,t}}{\tilde{\phi}_{i,i,t-1} + \tilde{\phi}_{i,1-i,t-1}} + \frac{1}{L}\log c_2\left(F_{i,1}(d_{1,t}|d_{1,t-1}), \cdots, F_{i,L}(d_{L,t}|d_{L,t-1})\right)\right. \\
&\left.+ \log f_{i,l}(d_{l,t}|d_{l,t-1})\right] \qquad (3.10)
\end{aligned}
$$

3.3.2 The Maximization Step

In the *maximization step*, $Q(\Theta;\Theta^{(n-1)})$ is maximized with respect to the parameter set Θ to obtain the next parameter set.

To maximize $Q(\Theta;\Theta^{(n-1)})$ with respect to $\widetilde{\phi}_{i,j,t}$, $i,j=0,1$ we need to consider the constraint $\sum_{i=0}^{1}(\widetilde{\phi}_{i,i,t}+\widetilde{\phi}_{i,1-i,t})=1$. Therefore, we use the Lagrangian $\mathcal{L}_\phi$, given by

$$\mathcal{L}_\phi \triangleq Q(\Theta;\Theta^{(n-1)})+\epsilon_\phi\left[\sum_{i=0}^{1}(\widetilde{\phi}_{i,i,t}+\widetilde{\phi}_{i,1-i,t})-1\right] \tag{3.11}$$

whose derivatives with respect to $\widetilde{\phi}_{i,i,t}$ and $\widetilde{\phi}_{i,1-i,t}$ are

$$\frac{\partial\mathcal{L}_\phi}{\partial\widetilde{\phi}_{i,1-i,t}}=\frac{\alpha_1^{(n)}(i,t)}{\widetilde{\phi}_{i,1-i,t}}+\epsilon_\phi=0 \tag{3.12}$$

$$\frac{\partial\mathcal{L}_\phi}{\partial\widetilde{\phi}_{i,i,t}}=\frac{\alpha_2^{(n)}(i,t)}{\widetilde{\phi}_{i,i,t}}+\epsilon_\phi=0 \tag{3.13}$$

Multiplying the two sides of (3.12) and (3.13) by $\widetilde{\phi}_{i,1-i,t}$ and $\widetilde{\phi}_{i,i,t}$, respectively, and summing the results together and over i gives $\epsilon_\phi=-\sum_{i=0}^{1}[\alpha_1^{(n)}(i,t)+\alpha_2^{(n)}(i,t)]=-1$. From this we get that $\widetilde{\phi}_{i,1-i,t}^{(n)}=\alpha_1^{(n)}(i,t)$ and $\widetilde{\phi}_{i,i,t}^{(n)}=\alpha_2^{(n)}(i,t)$.

To maximize $Q(\Theta;\Theta^{(n-1)})$ with respect to $\psi_{i,l}$ and $\tilde{\psi}_{i,l}$ for $i=0,1$ and $l=1,\cdots,L$, the constraints imposed by the selected distributions of sensors' measurements must be taken into account. In general, we solve

$$\begin{aligned}&\underset{\psi_{i,l}}{\text{Maximize}} \quad Q(\Theta;\Theta^{(n-1)}) \\ &\text{Subject to}: \int_{-\infty}^{\infty} f_{i,l}(x;\psi_{i,l})\,dx=1,\ 1\le l\le L, i=0,1\end{aligned} \tag{3.14}$$

to obtain Ψ and similarly for $\tilde{\Psi}$. For the parameters of the copula distributions, we solve

$$\begin{aligned} &\underset{\lambda_{j,i}}{\text{Maximize}} \quad Q(\Theta;\Theta^{(n-1)}) \\ &\text{Subject to}: \int_0^1 \cdots \int_0^1 c_j(\mathbf{x};\lambda_{j,i})\, d\mathbf{x} = 1,\ j = 1,2, i = 0,1 \end{aligned} \tag{3.15}$$

to obtain Λ_1 and Λ_2.

In the next section, we consider a case study including two important classes of copulas, namely the Gaussian and the Student's t copulas and, we solve the two optimization problems presented above for these two cases. It should be pointed out, however, that the proposed system model and the EM-based algorithm is not limited to this case. In particular, it can be used for any marginal PDFs and copula density functions for which the optimization problems in the maximization step of EM can be solved.

3.4 A Case Study: Gaussian and Student's *t* Copulas

In this section we derive the update equations of the unknown parameters for the case where the PDFs of the measurement data from each sensor under the hypothesis $\mathcal{H}_i$ can be modeled by a Gaussian PDF. More specifically, we assume that $f_{i,l}(d_{l,t}) \sim \mathcal{N}(\psi_{i,l}, (\sigma_{i,l})^2)$, where the variance $(\sigma_{i,l})^2$ is known, and the mean $\psi_{i,l}$ is unknown. In addition, we require a model to represent the first order dependence in the data collected over time for when the state of nature does not change at time t. The Autoregressive Model (AR), has been widely used as a first-order Markov process for parametric analysis and modeling of signals in a variety of contexts including speech and seismic signal processing, spectral estimation, process control and others [81]. Thus, we assume that when the state of nature does not change at time t, the samples collected from each sensor follow a first order AR model with parameter ξ, namely

$$d_{l,t} = \xi d_{l,t-1} + \nu_{i,l,t}, \tag{3.16}$$

where for $i = 0, 1$, $\{\nu_{i,l,t}\}$ is the iid Gaussian innovation process with unknown mean $\psi_{i,l}$ and known standard deviation $\sigma_{i,l}$. Therefore, $f_{i,l}(d_{l,t}|d_{l,t-1}; \tilde{\psi}_{i,l}) \sim \mathcal{N}(\tilde{\psi}_{i,l}, (\sigma_{i,l})^2)$, where $\tilde{\psi}_{i,l} = \eta d_{l,t-1} + \psi_{i,l}$. We first study the case where the two copulas $c_1(.)$ and $c_2(.)$ are modeled by the Gaussian copula and next we consider the case where they are modeled by the Student's t copula.

3.4.1 The Gaussian Copula

From (2.4), it is apparent that the Gaussian copula is parametrized solely by its correlation matrix $\mathcal{R}$ [4]. Therefore, in our model of the sensors' measurements, we assume that, for $j = 1, 2$ and $i = 0, 1$, $\lambda_{j,i}$ constitutes the unknown correlation matrix of copula $c_j(.)$ under hypothesis $\mathcal{H}_i$.

$$
\begin{aligned}
Q_{\mathcal{G}}(\Theta; \Theta^{(n-1)}) &= \sum_{t=1}^{T} \sum_{i=0}^{1} \sum_{l=1}^{L} \frac{\alpha_1^{(n)}(i,t)}{L} \log \frac{\tilde{\phi}_{i,1-i,t}}{\tilde{\phi}_{1-i,i,t-1} + \tilde{\phi}_{1-i,1-i,t-1}} \\
&+ \frac{\alpha_2^{(n)}(i,t)}{L} \log \frac{\tilde{\phi}_{i,i,t}}{\tilde{\phi}_{i,i,t-1} + \widetilde{\phi}_{i,1-i,t-1}} - \sum_{j=1}^{2} \frac{\alpha_j^{(n)}(i,t)}{2L} \left[\log |\lambda_{j,i}| + \mathbf{y}_{j,i}(t)^{Tr} (\lambda_{j,i}^{-1} - I_L) \mathbf{y}_{j,i}(t)\right] \\
&- \sum_{j=1}^{2} \frac{\alpha_j(i,t)}{2} \left[\log(2\pi(\sigma_{i,l})^2) + \mathbf{y}_{j,i}(t)^{Tr} \mathbf{y}_{j,i}(t)\right] \qquad (3.17)
\end{aligned}
$$

Using the Gaussian marginals and the Gaussian copulas in (3.10) we obtain (3.17), where $\mathbf{y}_{j,i}(t) = [y_{j,i,1}(t), \cdots, y_{j,i,L}(t)]^{Tr}$, for $j = 1$, $k = 1, 2, \cdots, L$,

$$
y_{j,i,k}(t) = G^{-1}(F_{i,k}(d_{k,t}); 0, 1) = (d_{k,t} - \psi_{i,k})/\sigma_{i,k}, \qquad (3.18)
$$

[4] In the case where the marginal distributions are also Gaussian, as is in this case study, the corresponding multivariate distribution is the multivariate Gaussian distribution. In this case, let the variances of the N marginal distributions be denoted by σ_i^2 for $i = 1, \cdots, N$. Then, the correlation matrix of the copula, $\mathcal{R}$, is related to the covariance matrix of the multivariate distribution, which we denote by Σ_{xx}, in the following manner:

$$
\mathcal{R} = \begin{bmatrix} 1 & \cdots & \rho_{1,N} \\ \rho_{1,2} & \cdots & \rho_{2,N} \\ \vdots & & \vdots \\ \rho_{1,N} & \cdots & 1 \end{bmatrix}, \quad \Sigma_{xx} = \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_1\sigma_N\rho_{1,N} \\ \sigma_1\sigma_2\rho_{1,2} & \cdots & \sigma_2\sigma_N\rho_{2,N} \\ \vdots & & \vdots \\ \sigma_1\sigma_N\rho_{1,N} & \cdots & \sigma_N^2 \end{bmatrix}.
$$

and for $j = 2$, $k = 1, 2, \cdots, L$,

$$y_{j,i,k}(t) = G^{-1}(F_{i,k}(d_{k,t}|d_{k,t-1}); 0, 1) = (d_{k,t} - \tilde{\psi}_{i,k})/\sigma_{i,k}, \tag{3.19}$$

where $G^{-1}(.; 0, 1)$ is the inverse of the Gaussian CDF with mean zero and variance one.

First, we consider maximization of $Q_{\mathcal{G}}(\Theta; \Theta^{(n-1)})$ with respect to the correlation matrices $\lambda_{1,i}$ and $\lambda_{2,i}$. Clearly, the correlation matrix $\lambda_{j,i}$ is a positive definite (PD) matrix with unit diagonal elements. However, at this point we relax these constraints and only require that $\lambda_{j,i}$ be a positive semi-definite (PSD) matrix. We will later discuss how a solution is obtained which satisfies the required constraints. Let Υ_L^+ denote the convex set of $L \times L$ PSD matrices. Then, it can be seen that maximization of $Q_{\mathcal{G}}(\Theta; \Theta^{(n-1)})$ with respect to $\lambda_{j,i}$ is equivalent to the following optimization problem.

$$\begin{aligned} &\underset{\lambda_{j,i}}{\text{Minimize}}\ r(\lambda_{j,i}) = \sum_{t=1}^{T} \alpha_j^{(n)}(i,t) \left[\log|\lambda_{j,i}| + \mathbf{y}_{j,i}^{(n-1)}(t)^{Tr} \lambda_{j,i}^{-1} \mathbf{y}_{j,i}^{(n-1)}(t)\right] \\ &\text{Subject to :} \qquad \lambda_{j,i} \in \Upsilon_L^+ \end{aligned} \tag{3.20}$$

It is well known that for any matrix $A \in \Upsilon_L^+$, the function $\log|A|$ is concave while the function A^{-1} is convex. This implies that $r(\lambda_{j,i})$ is not a convex function. To get around this problem, let $E_{j,i} \triangleq \lambda_{j,i}^{-1}$. Clearly, $E_{j,i} \in \Upsilon_L^+$. Then, the optimization problem in (3.20) can be written as

$$\begin{aligned} &\underset{E_{j,i}}{\text{Minimize}}\ \tilde{r}(E_{j,i}) = \sum_{t=1}^{T} \alpha_j^{(n)}(i,t) \left[-\log|E_{j,i}| + \mathbf{y}_{j,i}^{(n-1)}(t)^{Tr} E_{j,i} \mathbf{y}_{j,i}^{(n-1)}(t)\right] \\ &\text{Subject to :} \qquad E_{j,i} \in \Upsilon_L^+ \end{aligned} \tag{3.21}$$

Now, in the objective function in (3.21), the first term is convex and the second term is linear. Therefore the objective function is convex. In addition, the set Υ_L^+ is closed and convex. Therefore, (3.21) has a unique solution. In particular, the gradient of the objective

function is given by

$$\nabla_{E_{j,i}}\left(\tilde{r}(E_{j,i})\right) = -\left\{\sum_{t=1}^{T}\alpha_j^{(n)}(i,t)\right\}E_{j,i}^{-1} + \sum_{t=1}^{T}\alpha_j^{(n)}(i,t)\mathbf{y}_{j,i}^{(n-1)}(t)\mathbf{y}_{j,i}^{(n-1)}(t)^{Tr} \tag{3.22}$$

Therefore, setting the gradient to zero we get

$$\lambda_{j,i}^{(n)} = (E_{j,i}^{(n)})^{-1} = \frac{\sum_{t=1}^{T}\alpha_j^{(n)}(i,t)\mathbf{y}_{j,i}^{(n-1)}(t)\mathbf{y}_{j,i}^{(n-1)}(t)^{Tr}}{\sum_{t=1}^{T}\alpha_j^{(n)}(i,t)} \tag{3.23}$$

Remark 6. *Examination of* (3.23) *shows that these are weighted empirical correlation matrices calculated from the data and since the values of* $\alpha_1(i,t)$ *and* $\alpha_2(i,t)$ *are either very close to zero or very close to one, the weighting acts as selecting the appropriate data to calculate the best estimation for the correlation matrix. For example, for calculating* $\lambda_{1,i}$*, only the data collected at the time instances when the hypotheses change from* $\mathcal{H}_{1-i}$ *to* $\mathcal{H}_i$ *are involved while for calculating* $\lambda_{2,i}$*, only the data collected at the time instances when the hypotheses do not change from* $\mathcal{H}_i$ *are involved.*

It is straightforward to verify that these matrices are PSD, i.e., $\lambda_{1,i}, \lambda_{2,i} \in \Upsilon_L^+$. However, to be legitimate correlation matrices for the copula densities, they must also have unit diagonal elements and be non-singular. In what follows we discuss these issues.

As discussed previously, $\lambda_{j,i}$ is an $L \times L$ matrix where L is the number of sensors. From (3.23), it is evident that up to T rank-one matrices are added to obtain the new value for $\lambda_{j,i}$. It turns out that if there are L linearly independent vectors in the set $\mathcal{Y}_j = \{\alpha_j(i,t)\mathbf{y}_{j,i}(t), t = 1, 2, \cdots, T\}$, then $\lambda_{j,i}$ will be full rank and, therefore, positive definite. Now, since for $t = 1, 2, \cdots, T$, $\mathbf{y}_{1,i}(t)$ are independent samples (see (3.18)), as $T \longrightarrow \infty$, almost surely the set $\mathcal{Y}_1$ will have L linearly independent vectors. In the case of $\lambda_{2,i}$, the samples in the set $\mathcal{Y}_2$ are dependent. However, they contain an innovation component which is an iid sequence (see (3.16) and (3.18)). Therefore, as $T \longrightarrow \infty$, again the set $\mathcal{Y}_2$ will (almost surely) have L linearly independent vectors. In fact, it is easy to see that for the case of continuous random vectors (e.g., Gaussian random vectors under consideration),

for any $T \geq L$, with probability one the set of vectors $\{\mathbf{y}_{1,i}(t), t = 1, 2, \cdots, T\}$ are linearly independent, and similarly for the set of vectors $\{\mathbf{y}_{2,i}(t),\ t = 1, 2, \cdots, T\}$, In practice we choose $T >> L$ in order to obtain a good estimate of the correlation matrices and this ensures that the estimated matrices are non-singular.

The second constraint for $\lambda_{j,i}$ is that it must have unit diagonal elements. This is not guaranteed. To resolve this issue, we propose to use the method suggested by Higham [79], where it is shown that given a symmetric matrix, there exists a unique correlation matrix (i.e., a matrix which is PSD and has unit diagonal elements), which is closest to the symmetric matrix in the sense of weighted Frobenius norm. This is actually a projection of the symmetric matrix on the set of PSD matrices with unit diagonal. We use the iterative algorithm proposed in [79] to compute the nearest correlation matrix to each of the symmetric matrices, $\lambda_{1,i}$ and $\lambda_{2,i}$.

Next, we maximize $Q_{\mathcal{G}}(\Theta; \Theta^{(n-1)})$ with respect to $\psi_{i,l}$ which is equivalent to maximizing

$$\mathcal{A}(\boldsymbol{\psi}_i) = -\sum_{t=1}^{T}\sum_{j=1}^{2} \frac{\alpha_j^{(n)}(i,t)}{2} \mathbf{y}_{j,i}(t)^{Tr} (\lambda_{j,i}^{(n-1)})^{-1} \mathbf{y}_{j,i}(t) \tag{3.24}$$

with respect to $\boldsymbol{\psi}_i = [\psi_{i,1}, \psi_i^{(2)}, \cdots, \psi_{i,L}]^{Tr}$. Thus, we calculate the gradient of $\mathcal{A}(\boldsymbol{\psi}_i)$ with respect to $\boldsymbol{\psi}_i$ which is given by

$$\nabla_{\boldsymbol{\psi}_i}\left(\mathcal{A}(\boldsymbol{\psi}_i)\right) = \sum_{t=1}^{T}\sum_{j=1}^{2} \alpha_j^{(n)}(i,t) (\Sigma_i^{(n-1)} \lambda_{j,i}^{(n-1)})^{-1} \mathbf{y}_{j,i}(t) \tag{3.25}$$

where $\Sigma_i^{(n-1)} = \text{diag}(\sigma_{i,1}^{(n-1)}, \sigma_{i,2}^{(n)}, \cdots, \sigma_{i,L}^{(n-1)})$. Setting (3.25) to zero we obtain the new

value of $\boldsymbol{\psi}_i$ as

$$\boldsymbol{\psi}_i^{(n)} = \left(\sum_{t=1}^{T}\sum_{j=1}^{2}\alpha_j^{(n)}(i,t)(\lambda_{j,i}^{(n-1)})^{-1}\right)^{-1}\sum_{t=1}^{T}\alpha_1^{(n)}(i,t)(\lambda_{1,i}^{(n-1)})^{-1}\mathbf{d}_t + \alpha_2^{(n)}(i,t)(\lambda_{2,i}^{(n-1)})^{-1}(\mathbf{d}_t - \xi\mathbf{d}_{t-1}) \tag{3.26}$$

Equation (3.26) implies that the estimated mean is similar to the empirical mean of the data weighted by the inverse of the correlation matrices. The new value of $\tilde{\psi}_{i,l}$ is then obtained from $\tilde{\psi}_{i,l}^{(n)} = \xi d_{l,t-1} + \ {}^{(n)}_{i,l}$.

3.4.2 The Student's t Copula

Considering the degree of freedom to be known, the Student's t copula is also parametrized by its correlation matrix, $\mathcal{R}$.

In this case, we need to maximize $Q_{\mathcal{T}}(\Theta;\Theta^{(n-1)})$ with respect to $\lambda_{j,i}$ for $j = 1,2$ and $i = 0,1$. For the case of Student's t copulas, we only need to consider the term in $Q_{\mathcal{T}}(\Theta;\Theta^{(n-1)})$ which contains $\lambda_{j,i}$, namely

$$\mathcal{B}_{j,i}(\lambda_{j,i}) \triangleq \sum_{t=1}^{T}\alpha_1^{(n)}(i,t)\log c_j(F_{i,1}(d_{1,t}),\cdots F_{i,L}(d_{L,t});\lambda_{j,i}). \tag{3.27}$$

In this case, c_j is the Student's t copula, and $\lambda_{j,i}$'s are the unknown correlation matrices of the copulas (denoted by $\mathcal{R}$). The degrees of freedom is assumed to be known for both copulas and denoted by $\eta_{j,i}$ for copula c_j, $j = 1,2$ under hypothesis $\mathcal{H}_i$, $i = 0,1$. Replacing (2.5) into (3.27), we get

$$\mathcal{B}_{j,i}(\lambda_{j,i}) = \sum_{t=1}^{T}\alpha_j^{(n)}(i,t)\left[\gamma_{j,i}^{(n-1)} - \frac{1}{2}\log|\lambda_{j,i}| - \frac{\eta_{j,i}+L}{2}\log(1+\frac{1}{\eta_{j,i}}\mathbf{v}_{j,i}^{(n-1)}(t)^{Tr}\lambda_{j,i}^{-1}\mathbf{v}_{j,i}^{(n-1)}(t))\right] \tag{3.28}$$

In (3.28), $\mathbf{v}_{j,i}^{(n-1)}(t) = [v_{j,i,1}^{(n-1)}(t), v_{j,i,2}^{(n-1)}(t),\cdots, v_{j,i,L}^{(n-1)}(t)]^{Tr}$ where, for $l = 1,\cdots,L$ and $i =$

$0, 1$, $v_{1,i,l}^{(n-1)}(t) = St^{-1}[F_{i,l}(d_{l,t}; \psi_{i,l}^{(n-1)})]$, $v_{2,i,l}^{(n-1)}(t) = St^{-1}[F_{i,l}(d_{l,t}|d_{l,t-1}; \tilde{\psi}_{i,l}^{(n-1)})]$, $St^{-1}[.]$ is the inverse of the standard Student's t distribution, and,

$$\gamma_{j,i}^{(n-1)} = \log \frac{\Gamma(\frac{\eta_{j,i}+L}{2})\Gamma(\frac{\eta_{j,i}}{2})^{L-1}}{\Gamma(\frac{\eta_{j,i}+1}{2})^{L} \prod_{l=1}^{L}(1 + \frac{1}{\eta_{j,i}} v_{j,i,l}^{(n-1)}(t)^2)^{-\frac{\eta_{j,i}+1}{2}}} \tag{3.29}$$

In (3.28), the convexity of the term $\log(1 + \frac{1}{\eta_{j,i}} \mathbf{v}_{j,i}^{(n-1)}(t)^{Tr} \lambda_{j,i}^{-1} \mathbf{v}_{j,i}^{(n-1)}(t))$ can not be established since it is a composition of a convex function (matrix inversion) with a concave function (log function). Therefore, once again we apply the change of variable $\mathcal{E}_{j,i} = \lambda_{j,i}^{-1}$ to get

$$\tilde{\mathcal{B}}_{j,i}(\mathcal{E}_{j,i}) = \sum_{t=1}^{T} \alpha_j^{(n)}(i,t) \left[\gamma_{j,i}^{(n-1)} + \frac{1}{2} \log |\mathcal{E}_{j,i}| - \frac{\eta_{j,i}+L}{2} \log(1 + \frac{1}{\eta_{j,i}} \mathbf{v}_{j,i}^{(n-1)}(t)^{Tr} \mathcal{E}_{j,i} \mathbf{v}_{j,i}^{(n-1)}(t)) \right]. \tag{3.30}$$

Note that the optimization of the function in (3.30) is equivalent to the optimization of $Q(\Theta; \Theta^{(n-1)})$ with respect to $\lambda_{j,i}$. The function in (3.30) is the difference of two concave functions. Optimization of such functions has been fully investigated in the literature [82], [83]. In particular, in the case of optimizing the difference of two convex functions over a bounded polyhedral set, convergence to the global solution is achieved in finite time [84]. We would like to note that this is the case for the problem at hand.

Now, setting the derivative of $\tilde{\mathcal{B}}_{j,i}$ with respect to $\mathcal{E}_{j,i}$ to zero we get

$$(\mathcal{E}_{j,i}^{(n)})^{-1} = \frac{\eta_{j,i}+L}{\sum_{t=1}^{T} \alpha_j^{(n)}(i,t)} \sum_{t=1}^{T} \frac{\alpha_j^{(n)}(i,t) \mathbf{v}_{j,i}^{(n-1)}(t) \mathbf{v}_{j,i}^{(n-1)}(t)^{Tr}}{\eta_{j,i} + \mathbf{v}_{j,i}^{(n-1)}(t)^{Tr} \mathcal{E}_{j,i} \mathbf{v}_{j,i}^{(n-1)}(t)} \tag{3.31}$$

We use the method suggested in [85] to solve (3.31) iteratively with the starting point being the estimate of the correlation matrix for the Gaussian copula. In other words the initial point of the iterations is calculated using (3.23). When the iterations converge, we directly obtain $(\mathcal{E}_{j,i}^{(n)})^{-1} = \lambda_{j,i}^{(n)}$. Once again, the matrices calculated by (3.31) are PD when T is large enough but the matrices are not guaranteed to have unit diagonal values. Thus,

we apply the algorithm in [79], to obtain the nearest correlation matrices to the matrices obtained from (3.31).

Next, we need to optimize $Q(\Theta;\Theta^{(n-1)})$ with respect to $\psi_{i,l}$ for $i=0,1$ and $l=1,\cdots,L$. Unfortunately, it is not possible to derive a closed form solution for this optimization problem. However, (3.26) shows that the copula function only slightly modifies the empirical estimation of $\psi_{i,l}$ by weighting the empirical mean using the inverse of the correlation matrices. Therefore, in each iteration of the EM algorithm we only use the empirical mean of the data, i.e.,

$$\overset{(n)}{\psi}_{i,l} = \frac{\sum_{t=1}^{T} \alpha_1^{(n)}(i,t) d_{l,t} + \alpha_2^{(n)}(i,t)(d_{l,t} - \xi d_{l,t-1})}{\sum_{t=1}^{T} \alpha_1^{(n)}(i,t) + \alpha_2^{(n)}(i,t)}, \tag{3.32}$$

as the updating formula for$\psi_{\ i,l}$ in the case of the Student's t copula. The numerical results in Section 3.6 show that the error in our estimates of the parameters$\psi_{\ i,l}$ is very small.

The entire procedure for the estimation of the parameter set and the detection of the hypotheses is summarized in Algorithm 2.

3.5 Simulation

In this section we describe the simulation set up used to obtain the numerical results. Experiments are performed for various types of sensors with $L=4$ and $L=8$. The observation data from the sensors are produced by simulation. In order to generate dependent data over time we use the auto-regressive model in (3.16), where for $i=0,1$, $\{\nu_{i,l,t}\}$ is an iid Gaussian process with mean $\psi_{i,l}$ and standard deviation $\sigma_{i,l}$. The values used for $\psi_{i,l}$ and $\sigma_{i,l}$ in this example are presented in Table 3.1. In other words, as mentioned previously, for the case that the hypothesis changes, the data is generated according to the PDF $f_{i,l}(d_{l,t}) \sim \mathcal{N}(\psi_{i,l},(\sigma_{i,l})^2)$, whereas for the case that the hypothesis does not change, the data is generated according to the PDF $f_{i,l}(d_{l,t}|d_{l,t-1}) \sim \mathcal{N}(\xi d_{l,t-1}+\psi_{i,l},(\sigma_{i,l})^2)$. Note that the parameters $\sigma_{i,l}$, and ξ are assumed to be known a priori but the mean values $\psi_{i,l}$ are unknown and will be estimated using the proposed method.

Data: Measurement matrix, D
Result: Estimation of the parameter set, Θ, detection of the hypotheses, H.

begin

Estimating parameters set, Θ, using the EM Algorithm:

Assume an initial value for Θ as follows:

$\tilde{\phi}_{i,j,t}^{(0)} = .25$ for $i, j = 0, 1$, $t = 1, \cdots, T$,

$\lambda_{j,i}^{(0)} = I_L$ for $j = 1, 2$, $i = 0, 1$,

Apply K-means to D then, initialize $\overset{(0)}{}_{i,l}$ as the cluster means;

while $e > 10^{-2}$ **do**

E Step:

Find $\alpha_j^{(n)}(i,t)$, using (B.2);

M Step:

Update $\tilde{\Phi}$ with $\widetilde{\phi}_{i,1-i,t}^{(n)} = \alpha_1^{(n)}(i,t)$ and $\widetilde{\phi}_{i,i,t}^{(n)} = \alpha_2^{(n)}(i,t)$,

Update Ψ using (3.26)/(3.32) for Gaussian/Student's t copulas,

Update Λ using (3.23)/(3.31) for Gaussian/Student's t copulas;

Calculate convergence criterion;

$e_\psi = \frac{1}{2L} \sum_{i=0}^{1} \sum_{l=1}^{L} |\frac{\psi_{i,l}^{(n)} - \psi_{i,l}^{(n-1)}}{\overset{(n-1)}{}_{i,l}}|$,

$e_\lambda = \frac{1}{4L^2} \sum_{j=1}^{2} \sum_{i=0}^{1} \frac{\|\lambda_{j,i}^{(n)} - \lambda_{j,i}^{(n-1)}\|_1}{\|\lambda_{j,i}^{(n-1)}\|_1}$,

$e = (e_\psi + e_\lambda)/2$;

end

Calculate $\hat{\phi}_{i,t} = \hat{\widetilde{\phi}}_{i,i,t} + \hat{\widetilde{\phi}}_{i,1-i,t}$ for $i = 0, 1$, $t = 1, \cdots, T$;

Calculate $i^* = \arg\max_i \hat{\phi}_{i,t}$;

Set the state of nature at time t as $\mathcal{H}_i^*$;

end

Algorithm 2: Estimating the parameter set and detecting the hypotheses.

To establish the dependence among the data collected by different sensors, we generate the data according to the Gaussian and Student's t copulas as discussed in Section 3.4.

Once the simulated data is generated, we run the proposed method as described in Section 3.3. We set the initial values of the probabilities $\tilde{\phi}_{i,j,t} = .25$. The initial values of the copula parameters $\lambda_{1,i}$ and $\lambda_{2,i}$ are chosen to be the $L \times L$ identity matrix. Finally, the initial values of $\psi_{i,l}$ are obtained from the unsupervised method of K-means [71].

The proposed EM-based algorithm converges empirically and in all our experiments the convergence is reached in fewer than 10 iterations. Once the final estimate $\hat{\Theta} = \{ \hat{\Psi}, \hat{\tilde{\Psi}}, \hat{\Lambda_1}, \hat{\Lambda_2}, \hat{\tilde{\Phi}} \}$ is obtained, we use $\hat{\tilde{\Phi}}$ to detect the hypothesis $\hat{H}$ as described in Section

Table 3.1: values of the means, $\psi_{i,l}$, and the standard deviations, $\sigma_{i,l}$, used to produce the simulated measurement data received from each sensor l under the hypothesis $\mathcal{H}_i$.

l	$\psi_{0,l}$	$\sigma_{0,l}$	$\psi_{1,l}$	$\sigma_{1,l}$
1	-1	4	15	3.2
2	-7	4	13	3.2
3	-13	3	-7	2.4
4	-11	2.5	1	2
5	-.2	2.5	-.1	2
6	7	1.5	13	1.2
7	6	1.5	12	1.2
8	1	1	11	.8

3.2.

As mentioned previously, hypothesis testing where dependence is accounted for both among the data collected by different sensors and among the samples collected over time by a single sensor has not been previously considered. Therefore, in order to demonstrate the effect of considering dependence both over time and among the sensors, we compare the proposed method (subsequently referred to as *Case 1*) with the EM methods which consider only some of the dependence in the data as discussed below.

1. *Case 2*: Dependence in the data collected by different sensors is included in the model but dependence in the samples collected by each sensor over time is ignored. In this case, the updating formulas for the expectation step of the EM algorithm will be

$$\alpha^{(n)}(i,t) = \frac{\phi_{i,t}^{(n-1)} c\left(F_{i,1}(d_{1,t}), \cdots, F_{i,L}(d_{L,t})\right) \prod_{l=1}^{L} f_{i,l}(d_{l,t})}{\sum_{j=0}^{1} \phi_{j,t}^{(n-1)} c\left(F_{j,1}(d_{1,t}), \cdots, F_{j,L}(d_{L,t})\right) \prod_{l=1}^{L} f_{j,l}(d_{l,t})},$$

and the updating formulas for the maximization step of the EM algorithm will be

$\phi_{i,t}^{(n)} = \alpha^{(n)}(i,t)$ and,

$$\lambda_i^{(n)} = \frac{\sum_{t=1}^{T} \alpha^{(n)}(i,t)\mathbf{y}_i^{(n-1)}(t)\mathbf{y}_i^{(n-1)}(t)^{Tr}}{\sum_{t=1}^{T} \alpha^{(n)}(i,t)},,$$
$$\boldsymbol{\psi}_i^{(n)} = \left(\sum_{t=1}^{T} \alpha^{(n)}(i,t)(\lambda_i^{(n-1)})^{-1}\right)^{-1} \left(\sum_{t=1}^{T} \alpha^{(n)}(i,t)(\lambda_i^{(n-1)})^{-1}\mathbf{d}_t\right),$$

for the case of the Gaussian copula, and

$$\lambda_i^{(n)} = \frac{\mu_i + L}{\sum_{t=1}^{T} \alpha^{(n)}(i,t)} \sum_{t=1}^{T} \frac{\alpha^{(n)}(i,t)\mathbf{y}_i^{(n-1)}(t)\mathbf{y}_i^{(n-1)}(t)^{Tr}}{\mu_i + \mathbf{y}_i^{(n-1)}(t)^{Tr}(\lambda_i^{(n)})^{-1}\mathbf{y}_i^{(n-1)}(t)}, \quad {}_{i,l}^{(n)} = \frac{\sum_{t=1}^{T} \alpha^{(n)}(i,t)d_{l,t}}{\sum_{t=1}^{T} \alpha^{(n)}(i,t)},$$

for the case of the Student's t copula. Note that, in this case, there is only one copula. Therefore, the subscript j in $\alpha_j(i,t)$, c_j, $\lambda_{j,i}$, $\eta_{j,i}$ and $\mathbf{y}_{j,i}(t)$ is dropped.

2. *Case 3*: Dependence in the data samples collected by each sensor over time is included in the model but dependence among the data collected by different sensors is ignored. In this case, there are no copulas in the model and the updating formulas for the expectation step of the EM algorithm is obtained by (B.2) where, in this case,

$$Pr(\mathbf{d}_t, \mathbf{d}_{t-1}|h_{i,t} = 1, h_{i,t-1} = j-1; \Theta^{(n-1)}) =$$
$$\begin{cases} \prod_{l=1}^{L} f_{i,l}(d_{l,t}) f_{1-i,l}(d_{l,t-1}) \ ; & \text{for } j = 1 \\ \prod_{l=1}^{L} f_{i,l}(d_{l,t}|d_{l,t-1}) f_{i,l}(d_{l,t-1}) \ ; & \text{for } j = 2 \end{cases}$$

and the updating formulas for the maximization step of the EM algorithm will be $\phi_{i,t}^{(n)} = \alpha_1^{(n)}(i,t) + \alpha_2^{(n)}(i,t)$, and

$${}_{i,l}^{(n)} = \frac{\sum_{t=1}^{T} \alpha_1^{(n)}(i,t)d_{l,t} + \alpha_2^{(n)}(i,t)(d_{l,t} - \xi d_{l,t-1})}{\sum_{t=1}^{T} \alpha_1^{(n)}(i,t) + \alpha_2^{(n)}(i,t)}.$$

3. *Case 4*: Dependence among the data collected by different sensors and dependence among the data samples collected over time by each sensor are both ignored in the

model. In this case, there are no copulas in the model and the updating formulas for the expectation step of the EM algorithm will be

$$\alpha^{(n)}(i,t) = \frac{\phi_{i,t}^{(n-1)} \prod_{l=1}^{L} f_{i,l}(d_{l,t})}{\sum_{j=0}^{1} \phi_{j,t}^{(n-1)} \prod_{l=1}^{L} f_{j,l}(d_{l,t})},$$

and the updating formulas for the maximization step of the EM algorithm will be $\phi_{i,t}^{(n)} = \alpha^{(n)}(i,t)$ and

$$\overset{(n)}{}_{i,l} = \frac{\sum_{t=1}^{T} \alpha^{(n)}(i,t) d_{l,t}}{\sum_{t=1}^{T} \alpha^{(n)}(i,t)}.$$

3.6 Numerical Results and Discussion

To evaluate the detection performance of our algorithm we define the metric *hypothesis discriminability* Δ_H given by

$$\Delta_H \triangleq \frac{1}{2T} \sum_{i=0}^{1} \sum_{t=1}^{T} \left| h_{i,t} - \hat{h}_{i,t} \right| \tag{3.33}$$

and to evaluate the accuracy of our estimations we define

$$\Delta_\Psi \triangleq \frac{1}{2L} \sum_{i=0}^{1} \sum_{l=1}^{L} \left| \frac{\hat{\psi}_{i,l} - \psi_{i,l}}{\psi_{i,l}} \right| \tag{3.34}$$

and

$$\Delta_\Lambda \triangleq \frac{1}{4L^2} \sum_{j=1}^{2} \sum_{i=0}^{1} \frac{\|\hat{\lambda}_{j,i} - \lambda_{j,i}\|_1}{\|\lambda_{j,i}\|_1} \tag{3.35}$$

In (3.35) we have used the 1-norm in order to measure the error between the estimated and actual correlation coefficients of pairs of sensors.

In simulations for Figs. 3.1-3.3, the data received by the FC is dependent among the

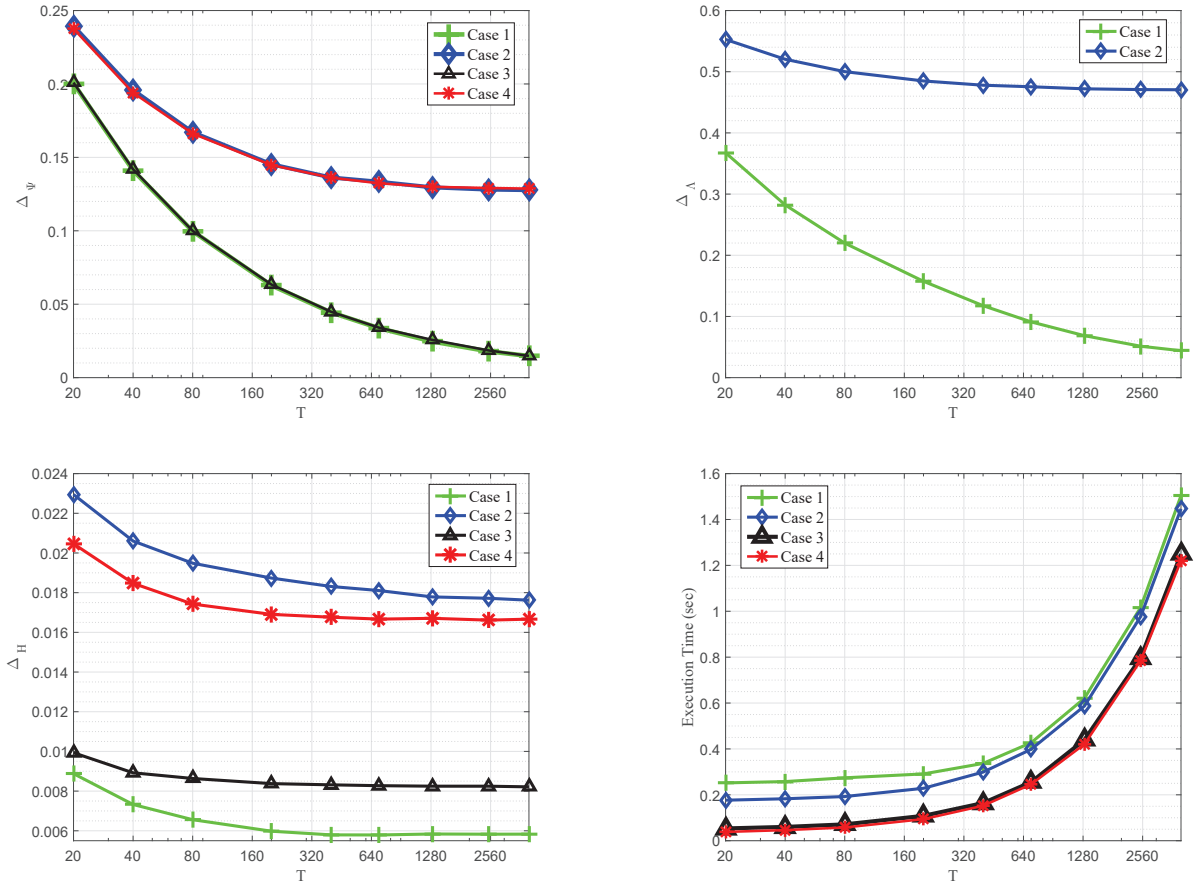

Figure 3.1: Simulation results for Cases 1-4 with the model based on the Gaussian copula and $L = 4$.

sensors and over time. However, as discussed in Section 3.5, only our proposed algorithm (Case 1) exploits both dependencies while the other three cases ignore part or all of the dependence. Our goal is to demonstrate the improvement that can be achieved when all the dependence in the data is utilized.

The results of hypothesis discriminability, the error in estimation of the marginal distribution parameters and the copula parameters for Cases 1-4 are presented in Fig.'s 3.1, 3.2, and 3.3 as a function of the number of time samples T. In Figs. 3.1 and 3.3, the model is based on the Gaussian copula with $L = 4$ and $L = 8$, respectively, whereas in Fig. 3.2, $L = 4$ and the model based on the Student's t copula is used. The computation time for all four cases are also presented in Fig.'s 3.1-3.3. The execution times are measured on a computer with 8 GB RAM and an Intel(R) Xeon(R) @ 2.00 GHz CPU with 2 processors,

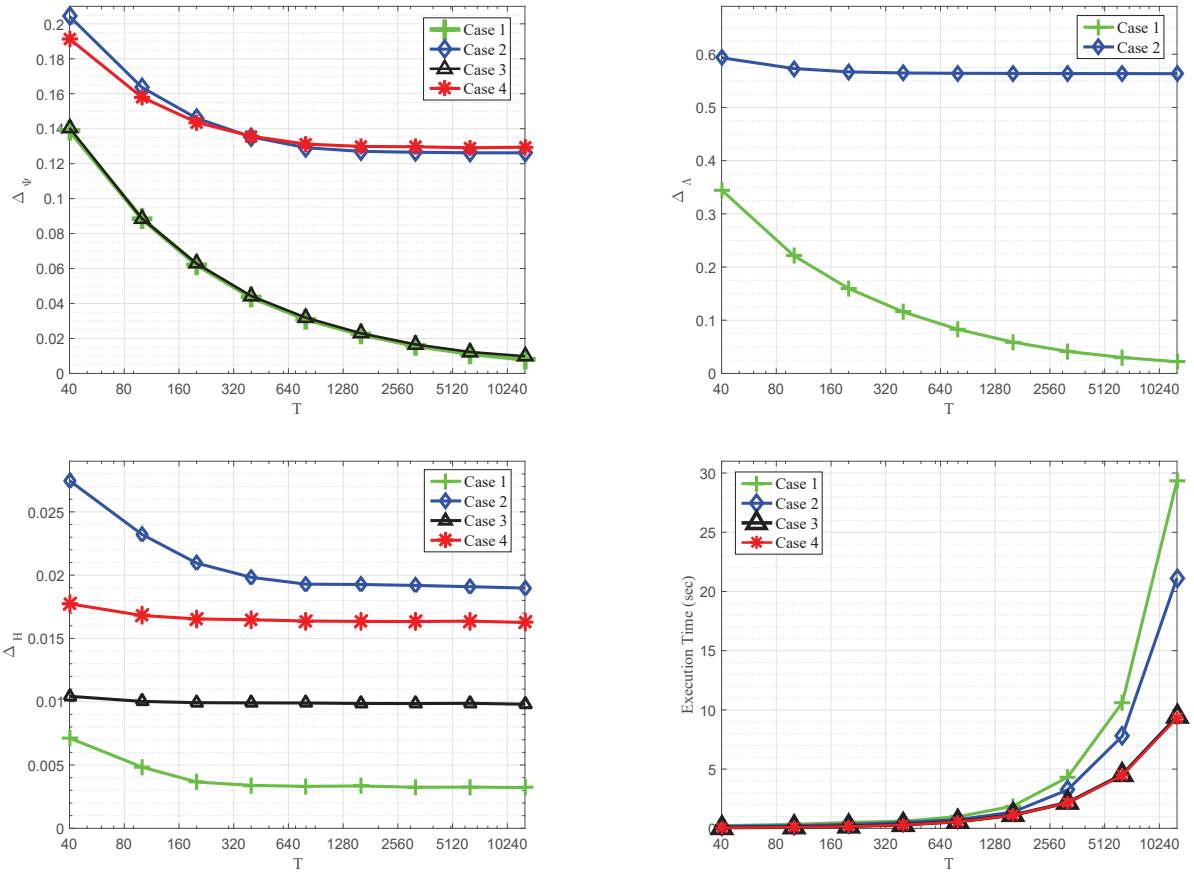

Figure 3.2: Simulation results for Cases 1-4 with the model based on the Student's t copula and $L = 4$.

and the program is executed with MATLAB R2013a.

It can be seen that the performances of hypothesis testing as well as parameter estimations improve significantly in Case 1, where the algorithm is capable of exploiting the dependence among the sensors along with the dependence among the data samples from each sensor. For example, in the model based on the Gaussian copula, for $L = 4$, when the number of time samples is larger than 100, the hypothesis discriminability of the proposed method improves by about .002 (or 25%) compared with Case 2, and by about .012 (or 65%) compared with Case 3. Similarly, in the model based on the Student's t copula, for $L = 4$, when the number of time samples is larger than 300, the hypothesis discriminability of the proposed method (Case 1) improves by about .006 (or 60%) compared with Case 2 and by about .016 (or 80%) compared with Case 3. In addition, the improvement gained

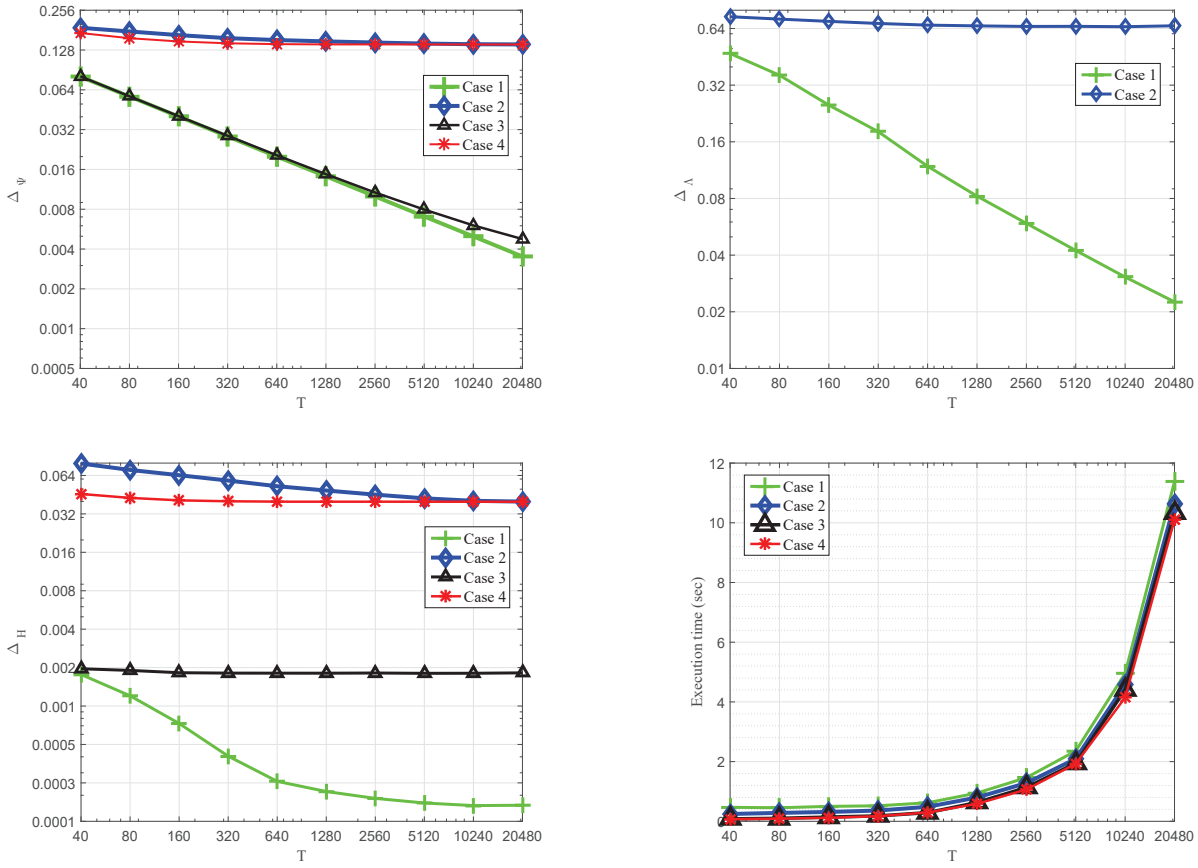

Figure 3.3: Simulation results for Cases 1-4 with the model based on the Gaussian copula and $L = 8$.

by using the proposed model is even higher when the number of sensors increases from $L = 4$ to $L = 8$. For example, in the model based on the Gaussian copula, for $L = 8$, when the number of time samples are larger than 640, the hypothesis discriminability for Case 1 is about .0017 (or 85%) less than Case 2, and about .045 (or 99%) less than Case 3. We should point out that when the number of samples T is very small (less than 20 for $L = 4$ and less than 40 for $L = 8$), the correlation matrices estimated in the EM algorithm are ill-conditioned. As a result, the copula densities cannot be defined and the proposed algorithm fails.

By comparing Figs. 3.1 and 3.3, we observe that, as expected, when the number of time samples, T, increases, the reliability of estimations improve. However, since the number of hypotheses to be detected is equal to T, as T increases, the hypothesis discriminability

reaches a floor[5]. Another observation is that for a larger L, a larger number of time samples are required to achieve the best possible detection performance that the proposed algorithm can offer. This is due to the fact that for a larger L, more parameters must be estimated. Moreover, Figs. 3.1-3.3 show that Cases 1 and 2 which exploit the dependence among sensors' data reach their best possible performance with a larger number of data samples. This can be better observed in Fig. 3.3 where L is larger. This is due to the fact that, the latter two algorithms need to estimate the $L \times L$ correlation matrices. A good estimate of these matrices requires a larger number of samples T and this value also increases with L.

Figs. 3.1-3.3 also show that the performance of the case that ignores the dependence of data over time (Case 2) is worse than the performance of the algorithm that ignores dependence over time and among the sensors (Case 4). This indicates that a bad dependence model has a more destructive effect than assuming independence. Therefore, in cases where the data is dependent over time, ignoring this dependence and only modeling the dependence among the sensors not only does not improve but also degrades the results. Modeling the dependence among the data from different sensors and assuming that the data samples from each sensor are iid is only effective if the data samples are actually iid [60, 61]. However, if the data are dependent over time (as is the case in many practical applications), then ignoring the dependence over time and only modeling the dependence among the sensors results in a worse performance than ignoring the dependencies all together.

Finally, the results show that ignoring the dependence of data over time has a more destructive effect on the performance of the algorithm in comparison to ignoring the dependence among the data collected by different sensors. This is expected since the dependence over time directly effects the mean value of the marginal PDFs, i.e., in the cases where dependence over time is ignored, $\psi_{i,l}$ is used as the mean instead of $\tilde{\psi}_{i,l} = \xi d_{l,t} + \psi_{i,l}$. Since for binary hypothesis testing, the mean value of the PDFs under each hypothesis directly effects the decision threshold, incorrect estimates of these mean values results in

[5]The initial improvements in hypothesis discriminability are due to the improved estimation of the parameter set Θ as T increases.

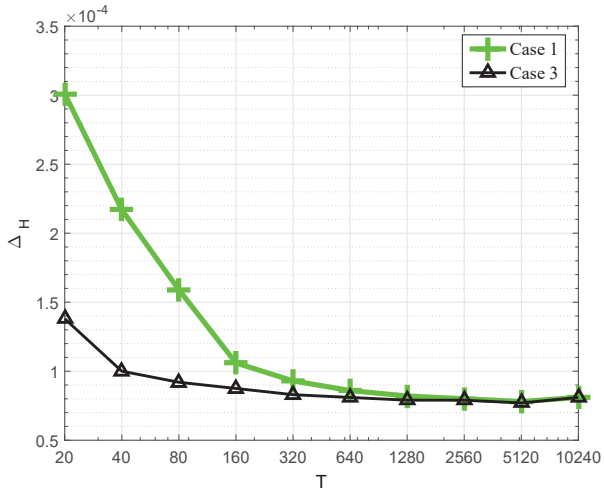

Figure 3.4: The hypothesis discriminability versus T for both Case 1 (green curve) and Case 3 (black curve) when $L = 4$. The data collected by different sensors are actually independent from each other.

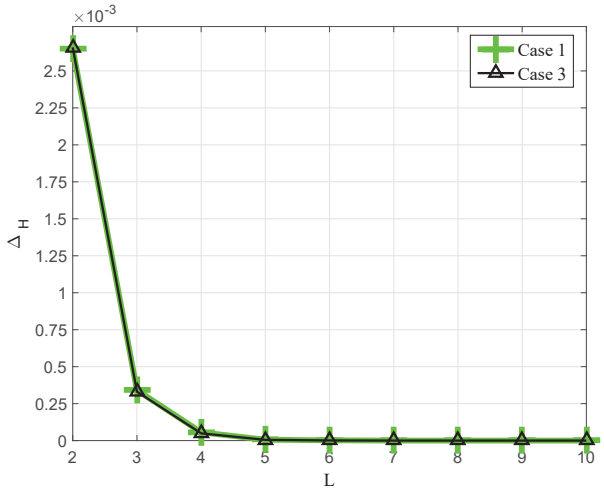

Figure 3.5: The hypothesis discriminability versus L for both Case 1 (green curve) and Case 3 (black curve) when $T = 500L^2$. The data collected by different sensors are actually independent from each other.

poor detection performance.

In some practical applications, we may not know whether the data collected from different sensors are dependent or not. Therefore, an important question is how would the proposed algorithm perform if the sensor data is in fact independent. Fig.'s 3.4 and 3.5 show the performance of Case 1 and Case 3 when the data collected by different sensors are independent. For Case 1 we assume a Gaussian copula. In Fig. 3.4 the hypothesis

discriminability is plotted vs. T for $L = 4$. It can be seen that as T increases the performance of Case 1 approaches that of Case 3. Therefore, given enough data samples, the model which assumes dependent data among the sensors will perform as well as the model that is "matched" to the data and assumes independent data. This is due the fact that given enough data samples the algorithm in Case 1 will be able to compute the correlation matrices accurately. In Fig. 3.5 hypothesis discriminability is plotted vs the number of sensors L for the number of samples $T = 500L^2$. It can be seen that in this case the two algorithms (Case 1 and Case 3) have similar performances.

3.7 Conclusion

We consider the problem of binary hypothesis testing in a wireless sensor network consisting of heterogeneous sensors. The sensors' measurements are assumed to be dependent both among the samples collected by each sensor and among the data collected by different sensors. The dependence in the data is modeled using the copula theory. It is assumed that the probability distribution of the sensors' data involves unknown parameters. We proposed a method based on the expectation maximization (EM) algorithm to estimate the unknown parameters and to detect the state of nature given the measurements of all sensors. We formulate our problem for the cases of two copulas, namely the Gaussian and Student's t copulas. Results are presented for four different cases where the model: 1) assumes dependence over time and among the sensors, 2) ignores the dependence over time only, 3) ignores the dependence among the sensors only, 4) ignores all the dependence in the data. These results quantify the performance of the algorithms in terms of detecting the sate of nature and in estimating the unknown parameters. It is shown that ignoring the dependence over time is more detrimental than ignoring the dependence among the sensors. However, including both dependencies results in the best performance as expected. It is also shown that when the data is independent among the sensors, given enough data samples, the proposed algorithm which assumes dependence among the sensors is able to estimate the actual correlation matrices and accurately detect the hypotheses.

Chapter 4
Online Hypothesis Testing and Non-Parametric Model Estimation Based on Correlated Observations

4.1 Introduction

In this chapter, we study the problem of hypothesis testing and non-parametric model estimation using correlated observations from a heterogeneous network of sensors. Once again we use the copula theory to model the dependence in the data and an EM based algorithm is used to perform estimation and detection via an unsupervised learning process. Consequently, the proposed algorithm does not require any labeled data. Moreover, we once again present an online as well as a batch-mode processing approach to the estimation and detection problem. In online processing, data samples are processed, system model is updated and a decision regarding the state of nature is made, all upon the arrival of each data sample, i.e., on an sample-by-sample basis. This is in contrast to batch processing which operates on a long data block to perform the above operations.

In previous chapters we have considered that the underlying marginal PDF of the sensors' data is known except for some parameters which need to be estimated.There we used parametric estimation methods to solve the problem. However, in some practical applications a Gaussian PDF or any other well-known distribution in closed form may not closely match the distribution of the sensors' data. To illustrate this point, we consider three real-world datasets: the *Room Occupancy Detection* (ROD) dataset available in [2], the face matching (NIST-face) dataset available at [86], and the *Activity Recognition based on Multisensor data fusion* (AReM) dataset available in [8].

The ROD dataset is used for binary hypothesis testing where $\mathcal{H}_0$ represents an unoccupied room and $\mathcal{H}_1$ represents an occupied room. In this chapter we use Light (in Lux) and CO2 (in ppm) sensory data from the ROD dataset to detect the occupancy status of a room. The ground-truth occupancy for this room was obtained from time stamped pictures taken every minute. In Fig. 4.1, the histogram of the data of each sensor is plotted under

$\mathcal{H}_0$ (in orange color) and $\mathcal{H}_1$ (in blue color), respectively.

The NIST-face dataset contains face matching scores from applying two different face matching algorithms to pairs of facial images. Here we combine the two face matching scores for binary hypothesis testing where $\mathcal{H}_1$ indicates that the pair of facial images match and $\mathcal{H}_0$ indicates that the pair of facial images do not match. In Fig. 4.2, the histogram of each face matching score is plotted under $\mathcal{H}_0$ (in orange color) and $\mathcal{H}_1$ (in blue color), respectively.

The AReM dataset contains data collected from a wireless SN worn by an actor performing activities including bending, cycling, and lying down. Infrared Intelligent Spectroradiometer (IRIS) sensors were placed on the actors chest, right ankle and left ankle. When a sensor is transmitting, all other sensors receive the data and calculate the Received Signal Strength (RSS). The data being fused at the fusion center are the set of RSS values between the transmitting sensors and the other sensors. Thus, in this dataset, six types of data, namely the average RSS for Chest-Right Ankle, Chest-Left Ankle, Right Ankle-Left Ankle, and the variance RSS for Chest-Right Ankle, Chest-Left Ankle, Right Ankle-Left Ankle, are combined at the FC. The goal is to detect the activity performed by the actor at each time using these 6 streams of data. In Fig. 4.3, the histograms of the average RSS for Chest-Right Ankle, Chest-Left Ankle, and Right Ankle-Left Ankle are plotted given the bending (blue colored), cycling (purple colored) and lying down (orange colored) activities, respectively.

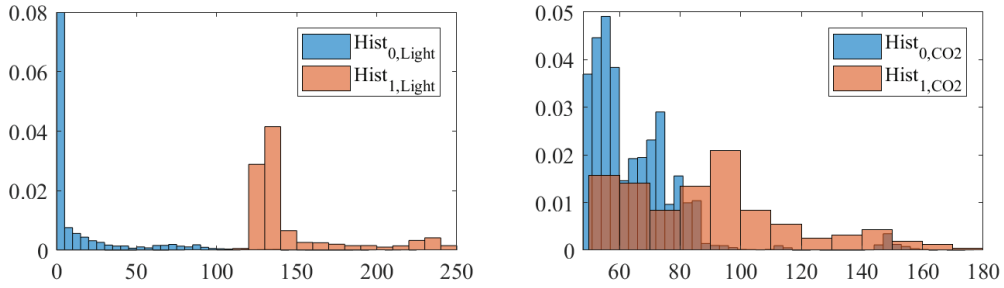

Figure 4.1: Histogram of Light and CO2 sensory data plotted under $\mathcal{H}_0$ (red color) and $\mathcal{H}_1$ (blue color).

From Fig.'s 4.1-4.3 it is clear that one or P Gaussian PDFs or other well-known distri-

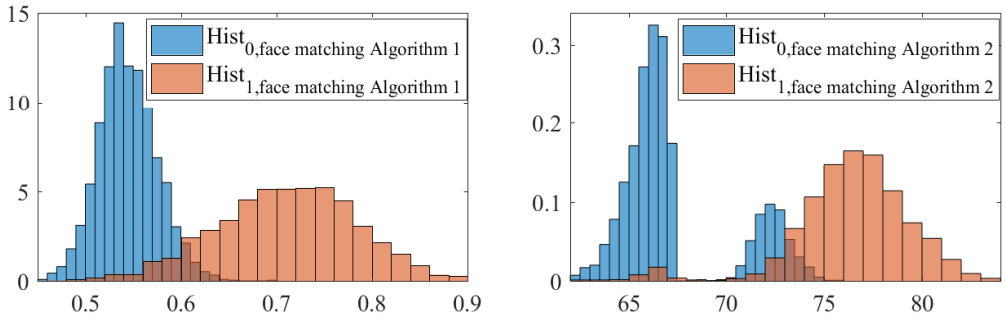

Figure 4.2: Histogram of face matching scores obtained from face matching Algorithm 1 and 2, plotted under $\mathcal{H}_0$ (red color) and $\mathcal{H}_1$ (blue color).

butions that have a closed form cannot match the marginal distribution of the collected data very well. To better model the distribution of data, in such cases, a non-parametric estimation approach is preferred to estimate the marginal PDFs of the sensors' data. Therefore, in this chapter we devise an online EM based algorithm for nonparametric estimation of the underlying PDF of each sensor's measurements under each hypothesis while detecting the state of nature at each time instant.

The novelty of the proposed algorithm is that: it develops an online detection and non-parametric model estimation algorithm for correlated observations, moreover, as a learning algorithm it is an unsupervised method.

The rest of this chapter is organized as follows. In Section 4.2 the problem is defined and the system model is described. In Sections 4.3.1 and 4.3.2, the batch-mode and online EM-based hypothesis testing algorithms are developed. Numerical results are presented and discussed in Section 4.4. Finally, conclusions are drawn in Section 4.5.

4.2 Problem Formulation and the System Model

We consider a network of L heterogeneous sensors employed to detect the state of nature $\mathcal{H} \in \{\mathcal{H}_0, \mathcal{H}_1, \cdots, \mathcal{H}_{K-1}\}$. At time t, sensor l transmits its measurement, denoted by $d_{l,t} \in \Re$, to the FC. After T time instances, the FC has received LT measurements which we collect into the $L \times T$ *measurement matrix* $D = [d_{l,t}]$. It is assumed that for each $l = 1, 2, \ldots, L$ and any $t_1 < t_2$, given the hypotheses at times $t = t_1, t_1 + 1, \ldots, t_2$, the sensor measurements $d_{l,t_1}, d_{l,t_1+1}, \ldots, d_{l,t_2}$ are iid. However, at each time t, the data samples

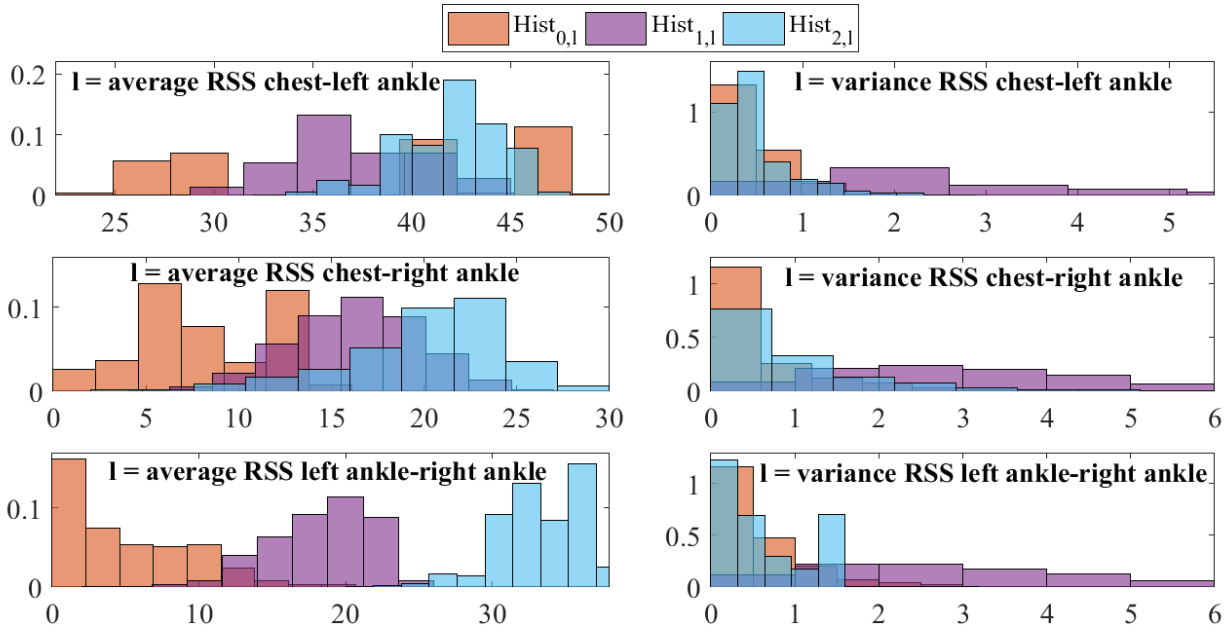

Figure 4.3: Histogram of average and variance RSS for Chest-Right Ankle, Chest-Left Ankle, and Right Ankle-Left Ankle plotted given the bending (blue colored), cycling (purple colored) and lying down (red colored) activities, respectively.

$d_{l,t}$, $l = 1, 2, \cdots, L$, are correlated. Let $\mathbf{d}_t \triangleq (d_{1,t}, d_{2,t}, \cdots, d_{L,t})^{Tr}$ where the superscript Tr denotes matrix transpose. The vector $\mathbf{h}_t = (h_{0,t}, h_{1,t}, \cdots, h_{K-1,t})^{Tr}$ is used to denote the state of nature at time t. If at time t, the state of nature is $\mathcal{H}_i$, then $\mathbf{h}_t = \mathbf{e}_i$ where $\mathbf{e}_i$ is the ith standard basis vector for $\Re^K$. For the entire observation period we construct the $K \times T$ *hypothesis matrix* $H = [h_{k,t}]$.

In an offline EM algorithm, having received the measurement matrix D, the FC must detect the state of nature for $t = 1, 2, \cdots, T$. To develop this algorithm, we need to evaluate the distribution of D given the hypothesis matrix H.

Let $F_{i,l}(d_{l,t})$ and $f_{i,l}(d_{l,t})$ denote the cumulative distribution function (CDF) and PDF of sensor l under hypothesis $\mathcal{H}_i$ evaluated at $d_{l,t}$. Note that $F_{i,l}(d_{l,t})$ and $f_{i,l}(d_{l,t})$ are assumed to be unknown and should be estimated. Let $\mathcal{F} \triangleq \{f_{i,l}(.)\}$ be the set of unknown density functions with KL elements. We model the joint distribution of the sensors' measurements under $\mathcal{H}_i$ by the copula distribution $C_m(F_{i,1}(d_{1,t}), \cdots, F_{i,L}(d_{L,t}; \lambda_{m,i})$ where m denotes the type of copula being considered, and $\lambda_{m,i}$ denotes the set of unknown parameters of the

copula distribution $C_m(.;.)$ under hypothesis $\mathcal{H}_i$. Therefore, the conditional distribution of $\mathbf{d}_t$ given $\mathbf{h}_t$ is given by

$$F(\mathbf{d}_t|\mathbf{h}_t;\mathcal{F},\mathbf{\Lambda}_m) = \prod_{i=0}^{K-1} C_m\left(F_{i,1}(d_{1,t}),\cdots,F_{i,L}(d_{L,t});\lambda_{m,i}\right)^{h_{i,t}} \tag{4.1}$$

where $\mathbf{\Lambda}_m \triangleq \{\lambda_{m,0},\lambda_{m,1},\cdots,\lambda_{m,K-1}\}$ is the set of parameters of the copula distribution m. From (4.1), the conditional PDF of $\mathbf{d}_t$ given $\mathbf{h}_t$ is given by

$$Pr(\mathbf{d}_t|\mathbf{h}_t;\mathcal{F},\mathbf{\Lambda}_m) = \prod_{i=0}^{K-1}\left[c_m\left(F_{i,1}(d_{1,t}),\cdots,F_{i,L}(d_{L,t});\lambda_{m,i}\right)\prod_{l=1}^{L} f_{i,l}(d_{l,t})\right]^{h_{i,t}} \tag{4.2}$$

where $c_m(.;.)$ denotes the copula density function of $C_m(.;.)$.

We define the auxiliary probabilities $P(h_{i,t}=1) \triangleq \phi_{i,t}$, which represent the probability of hypothesis $\mathcal{H}_i$ at time t. Note that these are not prior probabilities. Rather they are only used as a tool to help us transform the hypothesis detection problem into an estimation problem for $\phi_{i,t}$ which we can solve using the EM algorithm. We denote $\mathbf{\Phi} \triangleq [\phi_{i,t}]$ and define $\mathbf{\Theta} \triangleq [\mathbf{\Phi},\boldsymbol{\mathcal{F}},\mathbf{\Lambda}_m]$ as the set of unknown parameters and functions of the model.

Our goal is to estimate the unknown parameters and functions $\mathbf{\Theta}$, and calculate the hypothesis matrix H using the estimated value for $\mathbf{\Phi}$ denoted by $\hat{\mathbf{\Phi}}$. With this approach the state of nature at time t is detected as

$$\hat{h}_{i^*,t} = \begin{cases} 1 & ,\ i^* = \underset{0\le i\le K-1}{\operatorname{argmax}}\ \hat{\phi}_{\mathrm{i,t}} \\ 0 & ,\ \text{else} \end{cases} \tag{4.3}$$

Thus, the joint probability model given the unknown parameters of the model is given by

$$Pr(D,H;\mathbf{\Theta}) = \prod_{t=1}^{T}\prod_{i=0}^{K-1} Pr(\mathbf{d}_t,h_{i,t}=1;\mathbf{\Theta})^{h_{i,t}}, \tag{4.4}$$

where

$$Pr(\mathbf{d}_t, h_{i,t} = 1; \mathbf{\Theta}) = Pr(h_{i,t} = 1; \mathbf{\Theta}) Pr(\mathbf{d}_t | h_{i,t} = 1; \mathbf{\Theta})$$
$$= \phi_{i,t}\ c_m\left(F_{i,1}(d_{1,t}), \cdots, F_{i,L}(d_{L,t}); \lambda_{m,i}\right) \prod_{l=1}^{L} f_{i,l}(d_{l,t}), \tag{4.5}$$

4.3 Proposed EM-Based Algorithm

To estimate $\mathbf{\Theta}$, we employ the EM algorithm which iterates between the *expectation step* (E-step) and the *maximization step* (M-step) until convergence is reached. The E-step computes the expectation of the log-likelihood function of complete data (D, H) with respect to H, given the current estimate of the parameters $\mathbf{\Theta}^{n-1}$, namely

$$Q(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1)}) \triangleq E_{H|D;\mathbf{\Theta}^{(n-1)}}[\ln P(D, H; \mathbf{\Theta})] \tag{4.6}$$

In the M-step, $Q(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1)})$ is maximized with respect to $\mathbf{\Theta}$ to obtain the new estimate $\mathbf{\Theta}^{(n)}$, i.e.,

$$\mathbf{\Theta}^{(n)} = \text{argmax}_{\mathbf{\Theta}}\{Q(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1)})\}, \tag{4.7}$$

As mentioned in Chapter 2, the idea used in [77] is to replace the expectation step with a stochastic approximation step, while keeping the M-step unchanged. However, in cases where the updating functions in the maximization step are only functions of the complete data sufficient statistics, $s(\mathbf{d}_t, h_{i,t})$, we do not need to update $Q(\Theta; \Theta^{(t)})$ in the expectation step. Alternatively, we update the expectation of the sufficient statistic $s(\mathbf{d}_t, h_{i,t})$ as in

$$S^{(t^*)} = (1 - \epsilon^{(t^*)}) S^{(t^*-1)} + \epsilon^{(t^*)} E_{h_{i,t^*}|\mathbf{d}_{t^*};\mathbf{\Theta}^{(t^*-1)}}\left[s(\mathbf{d}_{t^*}, h_{i,t^*})\right], \tag{4.8}$$

where $\{\epsilon^{(t)}\}$ is a decreasing sequence of positive step sizes. The unchanged M-step, is then

given by

$$\Theta^{(t^*)} = \overline{\theta}(S^{(t^*)}), \tag{4.9}$$

where the function $\overline{\theta}(.)$ is obtained from the batch EM by $\overline{\theta}(S^{(n)}) \triangleq \text{argmax}_{\mathbf{\Theta}}\{Q(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1)})\}$ where $S^{(n)} = \frac{1}{T}\sum_{t=1}^{T} E_{h_{i,t}|\mathbf{d}_t;\mathbf{\Theta}^{(n-1)}}[s(\mathbf{d}_t, h_{i,t})]$.

In what follows we first develop the batch mode EM algorithm for which we define statistics similar to those in [77] and show that these statistics are sufficient for updating the parameters in the M-step. Later we extend the proposed method for online processing where the E-step only updates those sufficient statistics according to (4.8).

4.3.1 Proposed Batch-Mode EM-Based Algorithm

In the following the superscript (n, T) on a parameter denotes the estimated value of the parameter in the nth iteration of EM using T data samples. Moreover, the subscript m denotes the copula type where $m \in \mathcal{M}$.

To derive the expectation of the log-likelihood function, we start by deriving the log-likelihood function

$$L(D, H; \mathbf{\Theta}) \triangleq \log Pr(D, H; \mathbf{\Theta}) = \sum_{t=1}^{T}\sum_{i=0}^{K-1}$$
$$h_{i,t}\Big[\log \phi_{i,t} + \sum_{l=1}^{L} \log f_{i,l}(d_{l,t}) + \log c_m\left(F_{i,1}(d_{1,t}), \cdots, F_{i,L}(d_{L,t}); \lambda_{m,i}\right)\Big]. \tag{4.10}$$

Define $\alpha^{(n-1,T)}(i,t) \triangleq E[h_{i,t}|D; \mathbf{\Theta}^{(n-1,T)}]$. Then the expectation of the log-likelihood function given the current estimate of the parameters $\mathbf{\Theta}^{(n-1,T)}$ is given by

$$Q_m(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1,T)}) \triangleq E_{H|D;\mathbf{\Theta}^{(n-1,T)}}[\log Pr(D, H; \mathbf{\Theta})] = \sum_{t=1}^{T}\sum_{i=0}^{K-1}$$
$$\alpha^{(n,T)}(i,t)\Big[\sum_{l=1}^{L} \log f_{i,l}(d_{l,t}) + \log c_m(F_{i,1}(d_{1,t}), \cdots, F_{i,L}(d_{L,t}); \lambda_{m,i}) + \log \phi_{i,t}\Big]. \tag{4.11}$$

In the *Expectation step*, we need to calculate $\alpha^{(n,T)}(i,t)$ which is evaluated from

$$\alpha^{(n,T)}(i,t) = E[h_{i,t}|D;\mathbf{\Theta}^{(n-1,T)}] = Pr(h_{i,t}=1|D;\mathbf{\Theta}^{(n-1,T)})$$
$$= \frac{Pr(\mathbf{d}_t, h_{i,t}=1;\mathbf{\Theta}^{(n-1,T)})}{\sum_{j=0}^{K-1} Pr(\mathbf{d}_t, h_{j,t}=1;\mathbf{\Theta}^{(n-1,T)})}. \tag{4.12}$$

In the *Maximization step*, we maximize $Q_m(\mathbf{\Theta};\mathbf{\Theta}^{(n-1,T)})$ with respect to $\mathbf{\Theta}$ to obtain the new parameters $\mathbf{\Theta}^{(n,T)}$. To obtain the new estimate of $\mathbf{\Phi}$, we solve

$$\begin{aligned} &\underset{\phi_{i,t}}{\text{Maximize}} \quad Q_m(\mathbf{\Theta};\mathbf{\Theta}^{(n-1,T)}) \\ &\text{Subject to :} \quad \sum_{i=0}^{K-1} \phi_{i,t} = 1, \end{aligned} \tag{4.13}$$

for $m \in \mathcal{M}$. Defining the function $\overline{\phi}(x) \triangleq x$, we have

Lemma 7. *By solving the optimization problem in* (4.13)*, the parameter update formula for* $\phi_{i,t}$ *is given by* [1]

$$\phi_{i,t}^{(n,T)} = \overline{\phi}(\alpha^{(n,T)}(i,t)) = \alpha^{(n,T)}(i,t). \tag{4.14}$$

To obtain the new estimate of $\mathcal{F}$, we use the kernel-based non-parametric estimation method given by

$$f_{i,l}^{(n,T)}(x) = \frac{1}{\sigma_{i,l}\sum_{t=1}^{T}\alpha^{(n,T)}(i,t)} \sum_{t=1}^{T} \alpha^{(n,T)}(i,t) g(x; d_{l,t}, \sigma_{i,l}), \tag{4.15}$$

where $g(.; d_{l,t}, \sigma_{i,l})$ is the Gaussian kernel with mean $d_{l,t}$ and standard deviation $\sigma_{i,l}$. Equation (4.15) indicates that a kernel is placed at every data point and the weights $\alpha^{(n,T)}(i,t)$ ensure that the contribution of each data point $d_{l,t}$ to the distribution of data under each hypothesis depends on the probability of that hypothesis being true at time t. Moreover,

[1] The proof of lemma 7, follows from the proof of lemma 1.

to update the functions $\mathcal{F}$, only the kernel weights $\alpha^{(n,T)}(i,t)$ are updated at each iteration and the parameter $\sigma_{i,l}$ is not updated at each iteration. On the contrary, at the start of the EM algorithm an initial estimate of the parameter $\sigma_{i,l}$ is calculated with an ad hoc approach and then held fixed throughout the iterations. To calculate the initial estimate of $\sigma_{i,l}$, we calculate two quantities which have direct relationship with the standard deviation of the l th sensors' data under $\mathcal{H}_i$. Then, $\sigma_{i,l}$ will be calculated in proportion to the multiplication of these two factors. To obtain the first factor we compute the histogram of all measurements of the lth sensor under $\mathcal{H}_i$ at the tallest bin, i.e., $\max\{\text{Hist}_{i,l}\}$. The first factor is the logarithm of the ratio of $\max\{\text{Hist}_{i,l}\}$ over the total number of data samples collected by the lth sensor under $\mathcal{H}_i$. Let $[\mathbf{d}]_{l,i}$ denote the data samples collected by the lth sensor under $\mathcal{H}_i$, and $n(a)$ represent the number of elements in the vector a, then the first factor is chosen as $\log \frac{\max\{\text{Hist}_{i,l}\}}{n([\mathbf{d}]_{l,i})}$. For the second factor we compute the difference between the first and third quartiles of $\mathbf{d}_{l,i}$ which we denote by $q_{i,l,1}$ and $q_{i,l,3}$. Now, the second factor is the logarithm of the ratio of $q_{i,l,3} - q_{i,l,1}$ over the range of $\mathbf{d}_{l,i}$, i.e., $\log \frac{q_{i,l,3}-q_{i,l,1}}{\max\{[\mathbf{d}]_{l,i}\}-\min\{[\mathbf{d}]_{l,i}\}}$. We then set

$$\sigma_{i,l} = 2 \log \frac{\max\{\text{Hist}_{i,l}\}}{n([\mathbf{d}]_{l,i})} \log \frac{q_{i,l,3} - q_{i,l,1}}{\max\{[\mathbf{d}]_{l,i}\} - \min\{[\mathbf{d}]_{l,i}\}}. \tag{4.16}$$

Note that the two factors effecting the calculation of $\sigma_{i,l}$, represent a rough estimation for how narrow the largest peak of the density function $f_{i,l}(.)$ is. For example, the second factor represents a rough estimation of whether most of the data in $[\mathbf{d}]_{l,i}$ are within a small portion of the entire range of the data in $[\mathbf{d}]_{l,i}$. This is an indication of a narrow peak of significant height in the density function $f_{i,l}(.)$. For density functions with narrow peaks of significant height, a kernel with smaller bandwidth is required to ensure that the significant peak in the function can be well-represented.

To obtain the new estimate of $\mathbf{\Lambda}_m$, we solve the constrained optimization problem

$$\begin{aligned} &\underset{\lambda_{m,i}^{-1}}{\text{Minimize}} \quad Q_m(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1,T)}) \\ &\text{Subject to :} \int_0^1 \cdots \int_0^1 c_m \left(F_{i,1}^{(n-1,T)}(x_1) \cdots F_{i,L}^{(n-1,T)}(x_L); \lambda_{m,i} \right) dx_1 \cdots dx_L = 1. \end{aligned} \tag{4.17}$$

The solution to the optimization problem in (4.17) depends on the copula type and thus to present a more detailed mathematical solution to the problem, in what follows we consider a case study including the Gaussian and Product copulas for their wide scope of applications and we present closed form solutions to the optimization problem in (4.17) for these copulas. However, note that the proposed method for model estimation and hypothesis detection is not limited to these two copulas and can be applied to any copula for which the optimization problem in (4.17) has a solution. The solution to the optimization problem in (4.17) depends on the copula type and thus to present a more detailed mathematical solution to the problem, in what follows we consider a case study including the Gaussian and Product copulas for their wide scope of applications and we present closed form solutions to the optimization problem in (4.17) for these copulas. However, note that the proposed method for model estimation and hypothesis detection is not limited to these two copulas and can be applied to any copula for which the optimization problem in (4.17) has a solution.

- **Case Study: Gaussian and Product Copulas**

  For this case study, (4.11) can be written more accurately as

$$Q_{\mathcal{P}}(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1,T)}) = \sum_{t=1}^{T} \sum_{i=0}^{K-1} \alpha^{(n,T)}(i,t) \Big[\sum_{l=1}^{L} \log f_{i,l}(d_{l,t}) + \log \phi_{i,t} \Big], \tag{4.18}$$

for the Product copula, and as

$$\begin{aligned} Q_{\mathcal{G}}(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1,T)}) = &\sum_{t=1}^{T} \sum_{i=0}^{K-1} \alpha^{(n,T)}(i,t) \Big[\sum_{l=1}^{L} \log f_{i,l}(d_{l,t}) \\ &- \frac{1}{2} \log |\lambda_{\mathcal{G},i}| - \frac{1}{2} \left(\mathbf{y}_i(t) \right)^{Tr} \left(\lambda_{\mathcal{G},i}^{-1} - I_L \right) \mathbf{y}_i(t) + \log \phi_{i,t} \Big], \end{aligned} \tag{4.19}$$

for the Gaussian copula, where, $|A|$ denotes the determinant of matrix A, $\mathbf{y}_i(t) \triangleq [G^{-1}(F_{i,1}(d_{1,t}); 0, 1), \cdots, G^{-1}(F_{i,L}(d_{L,t}); 0, 1)]^{Tr}$ and $G^{-1}(.; 0, 1)$ is the inverse of the Gaussian distribution with mean zero and variance one.

Note that in the case of the Product copula, measurements of different sensors are assumed to be independent and there are no copula parameters $\mathbf{\Lambda}_m$. On the other hand, in the case of the Gaussian copula, the unknown parameter of the copula consists of its correlation matrix, in other words, $\lambda_{\mathcal{G},i}$'s are $L \times L$ positive definite matrices with unit diagonal values. Thus to obtain the solution to the optimization problem in (4.17), we first solve the optimization problem

$$\begin{aligned} &\underset{\lambda_{\mathcal{G},i}^{-1}}{\text{Minimize}} \quad Q_{\mathcal{G}}(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1,T)}) \\ &\text{Subject to : } \lambda_{\mathcal{G},i}^{-1} \in \Upsilon_L^+, \quad 0 \leq i \leq, K-1, \end{aligned} \tag{4.20}$$

where Υ_L^+ is the set of $L \times L$ positive semi-definite matrices. It can be shown that $Q_{\mathcal{G}}(\mathbf{\Theta}; \mathbf{\Theta}^{(n-1,T)})$ is a convex function of $\lambda_{\mathcal{G},i}^{-1}$. Let us define, the function

$$\overline{\lambda}_{\mathcal{G}}\left(S_1^{(n,T)}(i), S_2^{(n,T)}(i)\right) \triangleq \frac{S_1^{(n,T)}(i)}{S_2^{(n,T)}(i)}, \tag{4.21}$$

where

$$S_1^{(n,T)}(i) \triangleq \frac{1}{T}\sum_{t=1}^{T} \alpha^{(n,T)}(i,t)\mathbf{y}_i^{(n-1,T)}(t)(\mathbf{y}_i^{(n-1,T)}(t))^{Tr}, \tag{4.22}$$

$$S_2^{(n,T)}(i) \triangleq \frac{1}{T}\sum_{t=1}^{T} \alpha^{(n,T)}(i,t). \tag{4.23}$$

Lemma 8. *The solution to the optimization problem in* (4.20) *is given by* $\lambda_{\mathcal{G},i} = \overline{\lambda}_{\mathcal{G}}\left(S_1^{(n,T)}(i), S_2^{(n,T)}(i)\right)$.

The proof of lemma 8, follows from the proof of lemma 2.

We would like to point out that $S_1^{(n,T)}(i)$ is the weighted sample correlation matrix of the data and $S_2^{(n,T)}(i)$ is the mean of the weights (averaged over time.). Therefore in the case of the Gaussian copulas, the solution to (4.20) is the empirical correlation matrix.

As $T \longrightarrow \infty$,[2] the matrix obtained from (4.21) will be almost surely positive definite (PD). However, it does not necessarily have unit diagonal values. In order to have a valid correlation matrix, we apply the algorithm proposed by Higham [79] to obtain the closest correlation matrix to the solution of (4.20). Let, $\mathcal{NC}\{A\}$ represent the operation of obtaining the nearest correlation matrix to the matrix A, then the parameter update rule is given by

$$\lambda_{\mathcal{G},i}^{(n,T)} = \mathcal{NC}\left\{\overline{\lambda}_{\mathcal{G}}\left(S_1^{(n,T)}(i), S_2^{(n,T)}(i)\right)\right\}. \tag{4.24}$$

4.3.2 Proposed Online EM-Based Algorithm

Our proposed online algorithm consists of two stages. In the first stage which is called the initialization stage, an initial estimate of the parameters and the unknown marginal density functions are calculated. To this end, the batch-mode EM algorithm, described in Section 4.3.1, is performed using a small number of data samples, say $T_0 < T$.

In the second stage, upon receiving a measurement sample from the sensors at time $t^* > T_0$, the FC forms the vector $\mathbf{d}_{t^*} = [d_{1,t^*}, \ldots, d_{L,t^*}]^{Tr}$ and performs the two steps of the online algorithm. To initialize the unknown parameters and functions, at any time $t > T_0$, we use their estimate at the previous time instant, i.e., $\mathbf{\Theta}^{(t-1)}$.

To develop the online EM algorithm for the problem at hand, we need to derive the stochastic approximation of batch-mode E-step. The online M-step will be the same as the M-step of the batch-mode EM.

The update formulas for the M-step of the batch-mode EM in (4.13), (4.15), and (4.24),

[2]In fact when observations are independent samples of a continuous random variable, this property holds for $T \geq L$.

indicate that the updated quantities are functions of the statistics S_1 and S_2. Therefore, as in the online version of EM, we only need to calculate the new value of $\alpha^{(t)}(i,t)$ and update the sufficient statistics for updating the parameters in the M-step. Thus we define

$$S_j^{(t^*)} = \frac{1}{t^*} \sum_{t=1}^{t^*} E_{h_{i,t}|\mathbf{d}_t;\Theta^{(t-1)}} \left[s_j^{(t)}(h_{i,t}) \right], \tag{4.25}$$

for $j = 1, 2$. Note that in the online case, the superscript (t) denotes estimated parameter at the tth time and

$$s_1^{(t)}(h_{i,t}) = h_{i,t} \mathbf{y}_i^{(t-1)}(t) (\mathbf{y}_i^{(t-1)}(t))^{Tr}, \tag{4.26}$$

$$s_2^{(t)}(h_{i,t}) = h_{i,t}, \tag{4.27}$$

where, $\mathbf{y}_i^{(t-1)}(t) = [y_{i,1}^{(t-1)}(t), \cdots, y_{i,L}^{(t-1)}(t)]$ and

$$y_{i,l}^{(t-1)}(t) = G^{-1}(F_{i,l}^{(t-1)}(d_{l,t}); 0, 1). \tag{4.28}$$

Moreover,

$$\alpha^{(t)}(i,t) \triangleq E_{h_{i,t}|\mathbf{d}_t;\Theta^{(t-1)}} [h_{i,t}] = \frac{P(\mathbf{d}_t, h_{i,t} = 1|\Theta^{(t-1)})}{\sum_{j=0}^{M-1} P(\mathbf{d}_t, h_{j,t} = 1|\Theta^{(t-1)})} \tag{4.29}$$

and

$$P(\mathbf{d}_t, h_{i,t} = 1|\Theta^{(t-1)}) =$$
$$\phi_{i,t}^{(t-1)} \prod_{l=1}^{L} f_{i,l}^{(t-1)}(d_{l,t}) c_m \left(F_{i,1}^{(t-1)}(d_{l,t}), \cdots, F_{i,L}^{(t-1)}(d_{l,t}); \lambda_{m,i}^{(t-1)} \right). \tag{4.30}$$

Remark 9. *A comparison of* (4.22), (4.23) *with* (4.26), (4.27) *, respectively, reveals the motivation for the definition of the sufficient statistics in* (4.26), (4.27). *As can be seen the sufficient statistics in* (4.26), (4.27) *lack the averaging over time. This averaging, however,*

is performed in (4.25).

Let $\epsilon^{(t^*)}$ be a decreasing sequence and $\boldsymbol{h}_{t^*} = [h_{0,t^*}, \cdots, h_{M-1,t^*}]$. Then, using (4.8), the E-step of our proposed online algorithm is given by

$$S_j^{(t^*)} = (1-\epsilon^{(t^*)}) S_j^{(t^*-1)} + \epsilon^{(t^*)} E_{h_{i,t^*}|\mathbf{d}_{t^*};\Theta^{(t^*-1)}}[s_j^{(t^*)}(\boldsymbol{h}_{t^*})],\ j=1,2. \tag{4.31}$$

The M-step of the proposed online algorithm does not change and includes the parameter update equations

$$\phi_{i,t^*}^{(t^*)} = \overline{\phi}\left(\alpha^{(t^*)}(i,t^*)\right), \tag{4.32}$$

$$\lambda_{\mathcal{G},i}^{(t^*)} = \overline{\lambda}_{\mathcal{G}}\left(S_1^{(t^*)}, S_2^{(t^*)}\right). \tag{4.33}$$

As for updating the marginal density functions in the online case we need to take into consideration that the memory is limited and thus as new data are received it is not possible to assign and store kernels for each of them in order to later reconstruct the marginal PDFs. As a result we suggest uniformly sampling the PDFs estimated at N points and only storing the N samples from the PDFs. Let $x_{i,l,n}^{(t)}, n = 1, \cdots, N$ denote the N uniform samples of the measurement space of sensor l under $\mathcal{H}_i$ at time t. In this case, the online update of the marginal PDF consists of updating the evaluation of the function at these N points. Note that in cases where the newly arrived data point does not fall within the range of the previously received data points, a re-sampling is performed and thus $x_{i,l,n}^{(t)}$ will not be the same as $x_{i,l,n}^{(t-1)}$ for all $n = 1, \cdots, N$. To perform a resampling in online processing, we define the statistics

$$S_3^{(t)} = [\min((S_3^{(t-1)})_{1,i}, \delta_{mn,1,i,t}), \cdots, \min((S_3^{(t-1)})_{L,i}, h_{i,t}\delta_{mn,L,i,t})], \tag{4.34}$$

$$S_4^{(t)} = [\min((S_4^{(t-1)})_{1,i}, \delta_{mx,1,i,t}), \cdots, \min((S_4^{(t-1)})_{L,i}, \delta_{mx,L,i,t})], \tag{4.35}$$

for $t \leq T_0$, where, $\delta_{mn,l,i,t} = \begin{cases} d_{l,t},\ h_{i,t} = 1 \\ \infty,\ h_{i,t} = 0 \end{cases}$, $\delta_{mx,l,i,t} = \begin{cases} d_{l,t},\ h_{i,t} = 1 \\ -\infty,\ h_{i,t} = 0 \end{cases}$. Moreover,

$$(S_3^{(T_0)})_{l,i} = \min(\delta_{mn,l,i,1}, \cdots, \delta_{mn,l,i,T_0}),\ (S_4^{(T_0)})_{l,i} = \max(\delta_{mx,l,i,1}, \cdots, \delta_{mx,l,i,T_0}). \tag{4.36}$$

Then, the range of the measurements of each sensor l under $\mathcal{H}_i$ will be updated at each time t according to $(S_4^{(t)})_{l,i} - (S_3^{(t)})_{l,i}$ which we divide by the number of samples N to get the uniform sampling $x_{i,l,n}^{(t)}$.

Now, upon arrival of each data $\mathbf{d}_t$, first the density functions is updated at the previous sampling points according to

$$f_{i,l}^{(t)}(x_{i,l,n}^{(t-1)}) = \frac{1}{S_2^{(t)}}\Big(S_2^{(t-1)} f_{i,l}^{(t-1)}(x_{i,l,n}^{(t-1)}) + \frac{\alpha^{(t)}(i,t)}{\sigma_{i,l}} g(x_{i,l,n}^{(t-1)}; d_{l,t}, \sigma_{i,l})\Big). \tag{4.37}$$

Then, using linear interpolation, the density function is evaluated at the new sampling points $x_{i,l,n}^{(t)}$.

At each time instant t^*, after executing the expectation and maximization steps, the detection rule decides $\mathcal{H}_{i^*}$ to be the state of nature at t^* where $i^* = \underset{0 \leq i \leq K-1}{\operatorname{argmax}}\ \phi_{\mathrm{i,t^*}}$. The proposed algorithm is summarized in Algorithm 3.

Data: $\mathbf{d}_t$; sensor measurements' at time instance $t > T_0$.

Result: online updated value of Θ and detection of $\mathbf{h}_t$.

begin

Step1: initialization:

Assume an initial value for Θ as follows:

Set $\tilde{\phi}_{i,t}^{(0,T_0)} = \frac{1}{K}$, $\lambda_{\mathcal{G},i}^{(0,T_0)} = I_L$;

Apply K-means to $\mathbf{d}_{1:T_0}$;

Calculate $\sigma_{i,l}$ using (4.16) and the K-means clustering results;

Calculate $f_{i,l}^{(0,T_0)}(d_{l,t})$ for $t = 1, \cdots, T_0$, using (4.15) and the computed $\sigma_{i,l}$;

Apply batch EM on $\mathbf{d}_{1:T_0}$ to compute $\Theta^{(N,T_0)}$;

Evaluate $f_{i,l}^{(N,T_0)}(x_{i,l,n})$ at uniform sample points $x_{i,l,n}$ using (4.15);

Set $\Theta^{(T_0)} = \Theta^{(N,T_0)}$;

Step2: online updates:

while $\boldsymbol{d}_t$ *is received,* $t > T_0$ **do**

initialize parameters and functions using $\Theta^{(t)} = \Theta^{(t-1)}$;

online E Step:

Find $\alpha^{(t)}(i,t)$ with (4.29), update S_j, $j = 1, \cdots, 4$ with (4.31),(4.34),(4.35);

online M Step:

Update $\phi_{i,t}^{(t)}$, $\lambda_{\mathcal{G},i}^{(t)}$, $\boldsymbol{\mathcal{F}}^{(t)}$ using (4.32), (4.33), (4.37), respectively;

online Detection:

Calculate $i^* = \underset{0 \le i \le K-1}{\operatorname{argmax}} \phi_{\mathrm{i,t}}$ and set $\mathbf{h}_t = \mathbf{e}_{i^*}$.

end

end

Algorithm 3: Online parameter estimation and hypothesis detection.

4.4 Numerical Results and Discussion

We evaluate the performance of the proposed algorithm using real-world datasets, namely, the *Room Occupancy Detection* (ROD) [2], the *NIST-face* [86], and the *Activity Recognition based on Multisensor data fusion* (AReM) [8] datasets, introduced in Section 4.1. Table 4.1, summarizes the parameters used for applying the proposed algorithm to each dataset.

Table 4.1: Parameters used for each dataset in the proposed algorithm

Dataset	# of hypotheses (K)	# of sensors (L)	# of samples (N)	T_0	T
ROD	2	2	100	2000	8000
NIST-face	2	2	100	500	5000
AReM	3	6	100	1500	4500

.

To model the correlation in the data, we consider both Product and Gaussian copulas in the proposed method. Once again, to evaluate the detection performance of the proposed algorithm with both batch and online processing, we use *hypothesis discriminability* (Δ_H) and *Detection Accuracy* (DA) as defined in previous chapters. The notations BEM_m and $OEM_m^{T_0}$ denote the batch and online modes of the proposed EM algorithm using T_0 initialization samples and the copula type m, where $m = P, G$ denote the Product and Gaussian copulas, respectively. Moreover, Δ_H^m represents the hypothesis discriminability of the proposed algorithm using the copula type m. Finally, the notation $\text{Hist}_{i,l}$ represents the histogram of the data collected by sensor l under hypothesis $\mathcal{H}_i$.

Figures 4.4-4.6, present the histogram of the data collected by each sensor under each hypothesis along with the corresponding estimated marginal PDF using the proposed algorithm in batch mode. Since the PDF estimation results look similar for both copulas, we have only presented one of them (the Gaussian copula) as an example. In all cases, $\sigma_{i,l}$ is calculated according to (4.16). Moreover, in Fig.'s 4.4-4.6, hypothesis discriminability is presented for the batch EM algorithm using both copulas. According to these results, for the NIST-face and AReM datasets, the Gaussian copula slightly outperforms the Product

copula in terms of hypothesis discriminability, whereas for the ROD dataset, $\Delta_H^G = \Delta_H^P$. In the ROD dataset, the correlation matrix of the data is close to the identity matrix and thus the Product copula is a better match rather than the Gaussian copula. However, using $T = 8000$ data samples in the batch mode, the Gaussian copula performed as well as the Product copula in terms of hypothesis discriminability since the method using the Gaussian copula had enough data samples to estimate the correlation matrix λ_i to be close to the identity matrix.

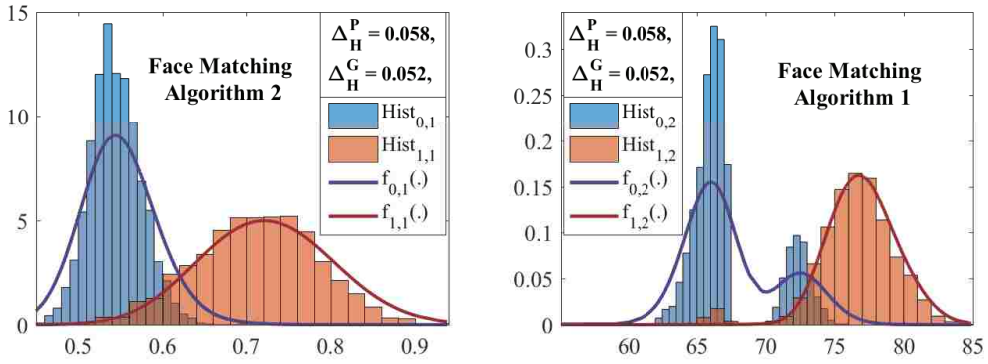

Figure 4.4: Estimation and detection results in batch mode for the NIST-face dataset.

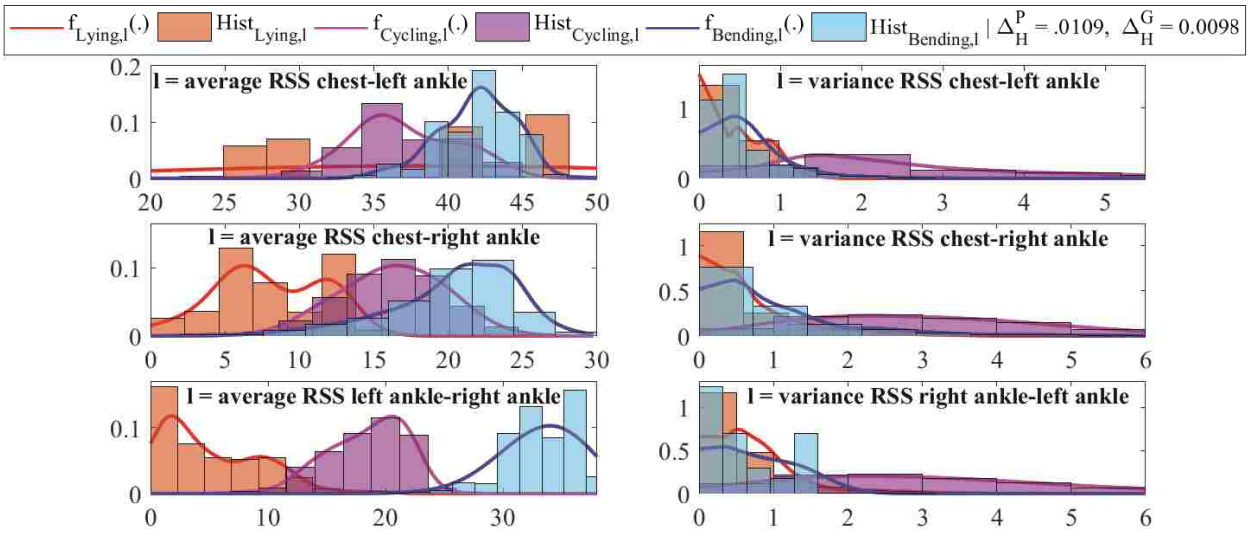

Figure 4.5: Estimation and detection results in batch mode for the AReM dataset.

Figures 4.7-4.9, present the histogram of the data collected by each sensor under each hypothesis along with the corresponding estimated marginal PDF using the proposed algorithm in online mode. In the online mode uniform sampling of the estimated PDFs are

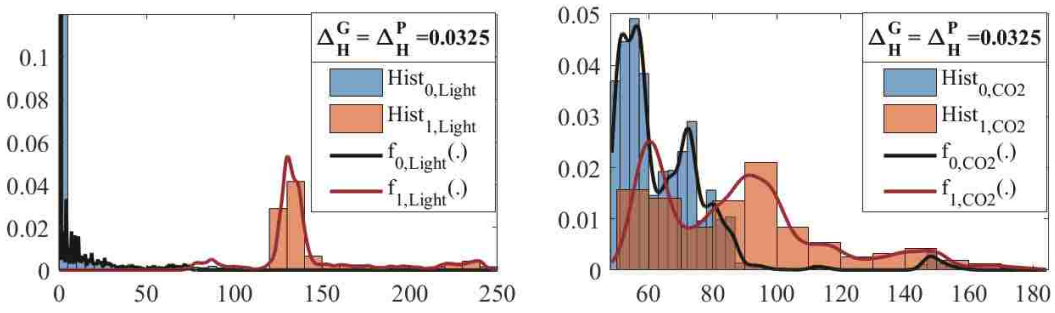

Figure 4.6: Estimation and detection results in batch mode for the ROD dataset.

used in order to keep required memory fixed. In Fig.'s 4.7 and 4.9, the effect of using different number of samples (N) for storing the marginal pdfs is shown for the NIST-face and ROD datasets, respectively. In Fig. 4.8, the estimated marginal PDF using the proposed algorithm in online mode with $N = 100$, is presented for the AReM dataset. Moreover, in Fig.'s 4.7-4.9, hypothesis discriminability is presented for the online EM algorithm for each value of N.

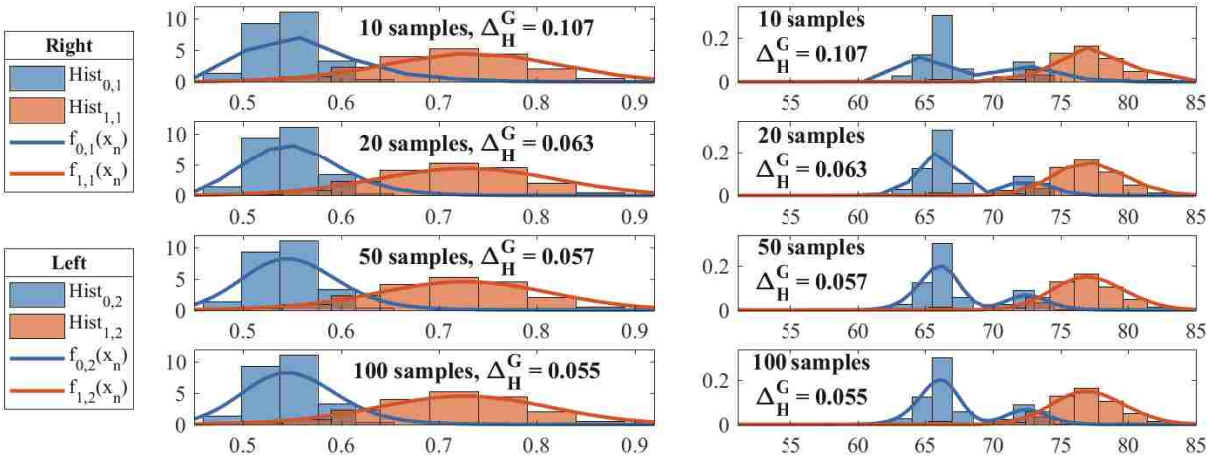

Figure 4.7: Estimation and detection results in online mode for the NIST-face dataset.

In Fig.'s 4.10-4.12, DA of the proposed algorithm in both batch and online modes are presented and compared with other well-known supervised and unsupervised methods for the NIST-face, AReM, and ROD datasets, respectively. In Fig.'s 4.10 and 4.11, DA is presented for both Gaussian and Product copulas and the results show a degradation in DA when ignoring the correlation in the data (i.e., using the Product copula). However, the effect of modeling the correlation in the data is more prominent for the NIST-face

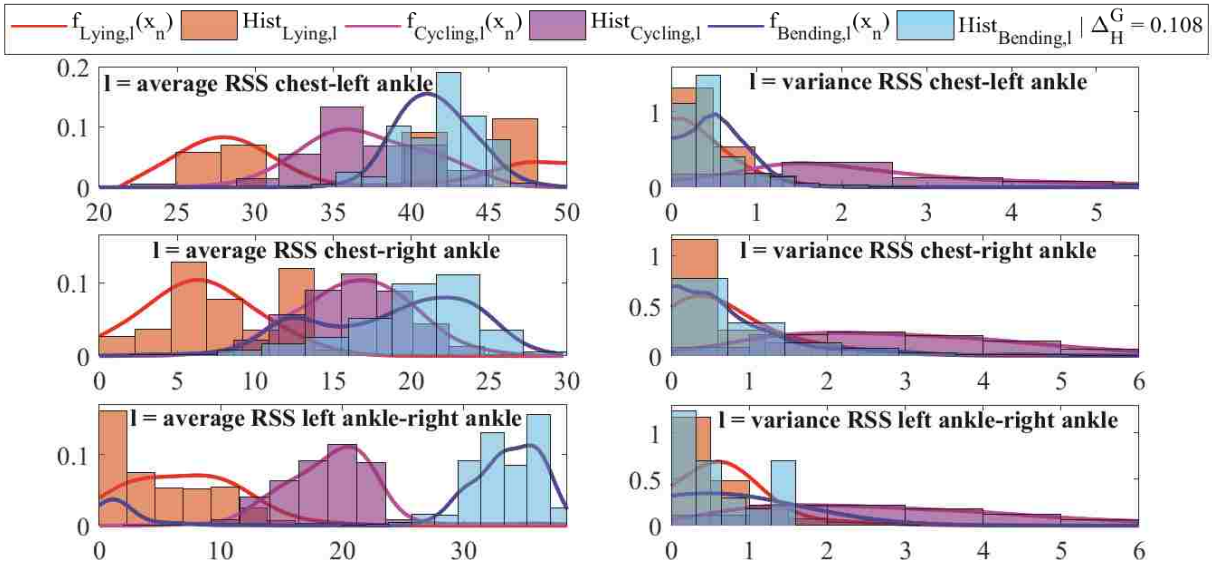

Figure 4.8: Estimation and detection results in online mode for the AReM dataset.

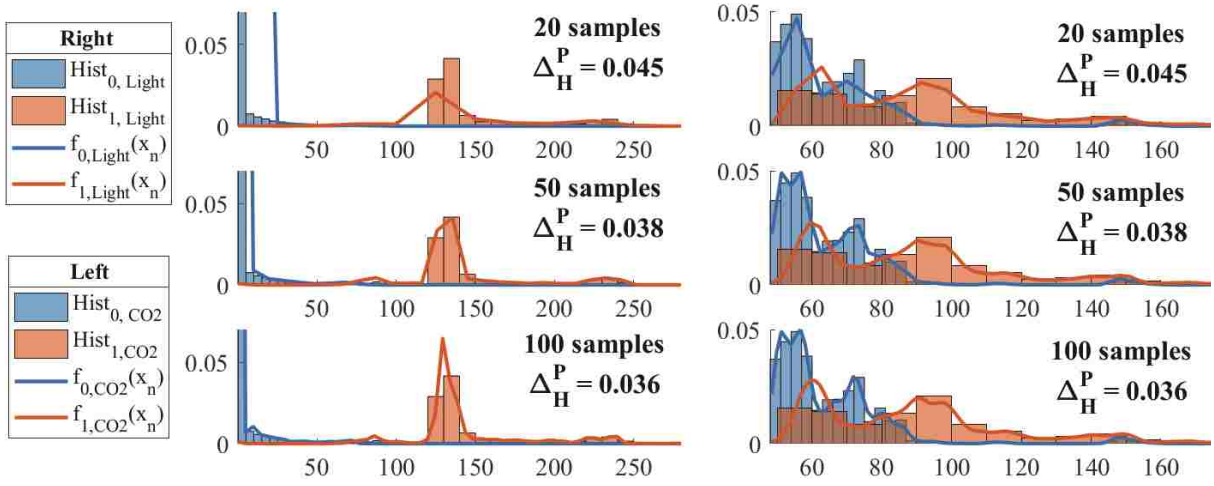

Figure 4.9: Estimation and detection results in online mode for the ROD dataset.

dataset. Thus, for the NIST-face datast, the effect of using these two different copula types are compared in Fig. 4.10 for both batch and online processing. For the AReM dataset, the effect of using different number of samples T_0 in *Step 1* of algorithm 3 is also presented in Fig. 4.11. This figure shows that the proposed algorithm is sensitive to the initialization (*Step 1* of algorithm 3), as using $T_0 = 3000$ brings DA very close to the DA achieved in the batch mode whereas the DA using $T_0 = 1500$ is drastically less than the DA using $T_0 = 3000$. For the ROD dataset, the Product copula is a better match and thus DA is the same using the Gaussian and Product copulas in batch mode, however, in the online mode, DA is

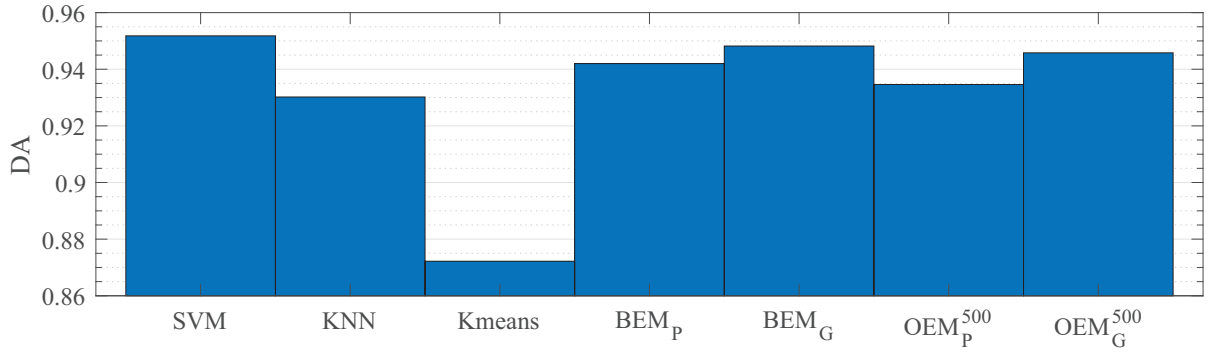

Figure 4.10: Detection Accuracy of different methods for the NIST-face dataset.

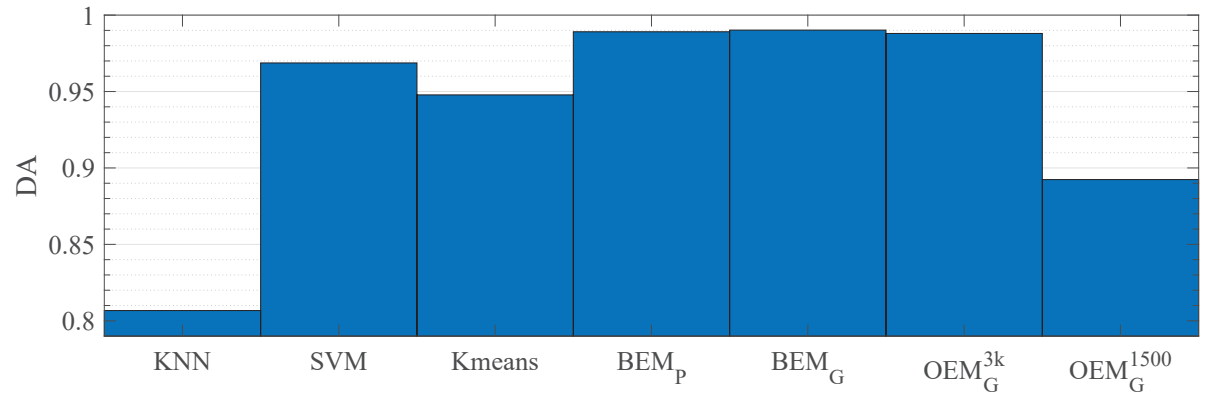

Figure 4.11: Detection Accuracy of different methods for the AReM dataset.

slightly worse when using the Gaussian copula rather than the Product copula. As a result in Figure 4.12, DA is presented for the Product copula in both batch and online modes and for different values of T_0. This figure shows that as T_0 increases, DA increases. To compare with other methods, we use the Support Vector Machines (SVM) and K-Nearest Neighbor (KNN) methods as supervised learning methods. As for the unsupervised methods with which we compare our proposed method, for the NIST-face and AReM datasets we consider the Kmeans clustering method, and for the ROD dataset, we consider the Page-Hinkley Test (PHT) and the Geometric Moving Average (GeoMA) which are two unsupervised methods devised specifically for room occupancy detection problems [1]. We should point out that these unsupervised methods all use batch-mode processing. For the supervised learning algorithms, a different training dataset is also required. For the NIST-face, ROD, and AReM datasets, training datasets containing 5000, 8144, and 3000 labeled data samples

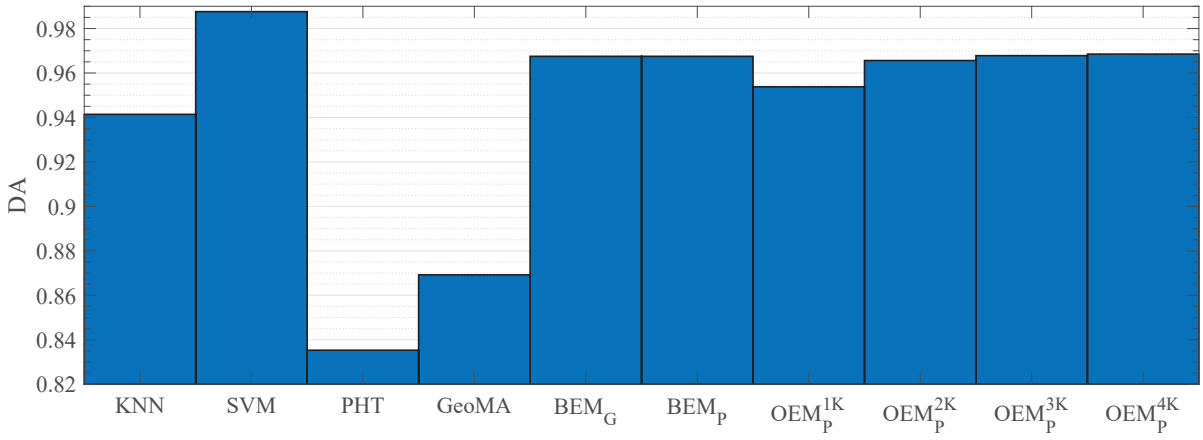

Figure 4.12: Detection Accuracy of different methods for the ROD dataset.

were used, respectively. Fig.'s 4.10-4.12 show that the proposed algorithm in both batch and online modes have higher DA than other unsupervised and even some supervised methods.

In Fig.'s 4.13-4.15, the actual state of nature at each time instance (H^{Actual}) is plotted along with the detected hypothesis at each time instance using different supervised and unsupervised methods ($H_{method\ name}$) for the NIST-face, AReM, and ROD datasets, respectively.

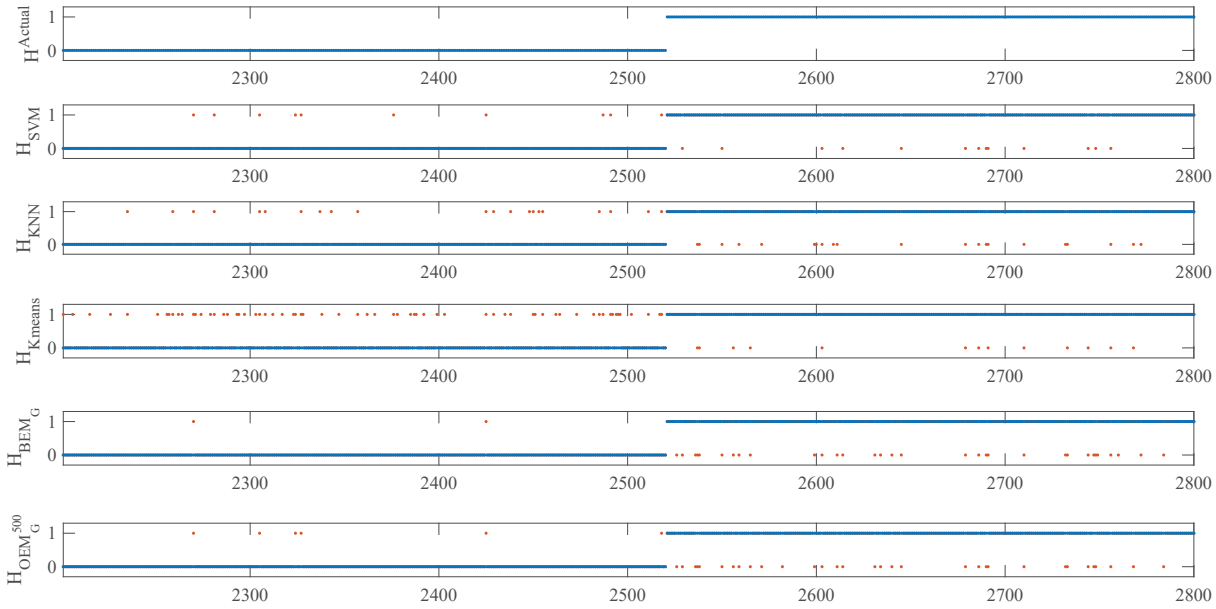

Figure 4.13: The actual (top row) and estimated values of the state of nature at each time instance using different methods (SVM, KNN, Kmeans, BEM_G, and OEM_G^{500}) for the NIST-face dataset.

Finally, we compare the non-parametric based estimation method presented in this sec-

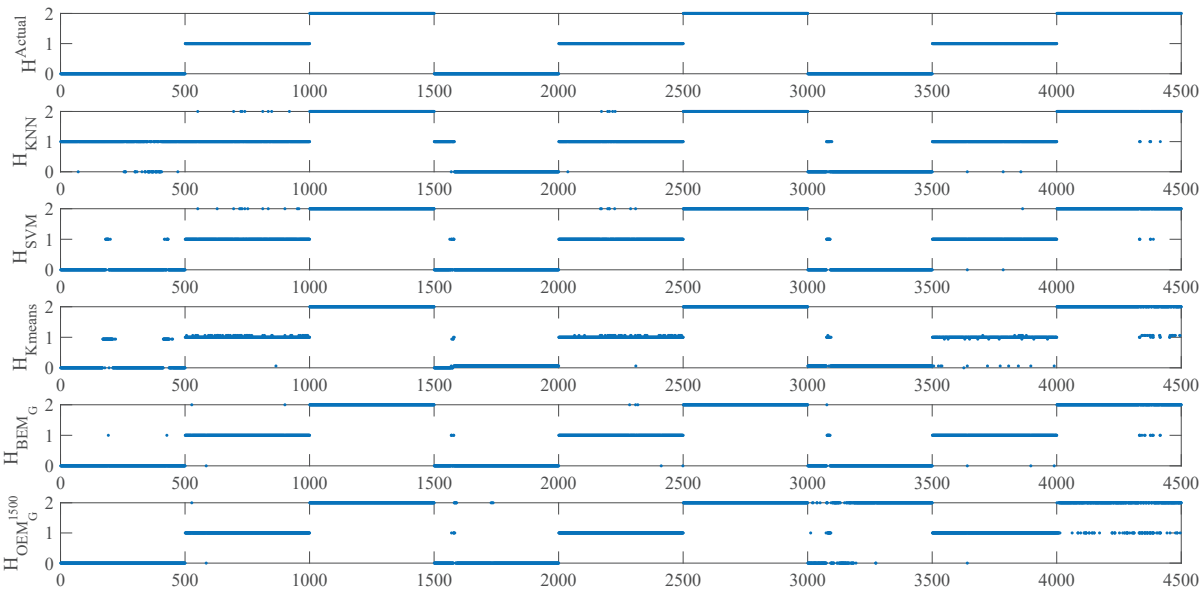

Figure 4.14: The actual (top row) and estimated values of the state of nature at each time instance using different methods (SVM, KNN, Kmeans, BEM_G, and OEM_G^{1500}) for the AReM dataset.

tion with the parametric based estimation method presented in Chapter 2. Once again we use the ROD, NIST-face and AReM datasets. In table 4.2, hypothesis discriminability is presented for the three datasets using the non-parametric and parametric based estimation methods in batch mode. Table 4.2 shows that for the NIST-face and AReM datasets, non-parametric estimation outperforms parametric estimation whereas for the ROD dataset, parametric estimation slightly outperforms non-parametric based estimation. This indicates that for the ROD dataset, the model using one of P possible Gaussian distributions is a perfect match to this problem. We can conclude that in cases (such as the ROD dataset) where the assumptions of the parametric model accurately match the physics of the problem, parametric estimation is the preferred choice. However, in such cases, non-parametric based estimation performs nearly as good as parametric based estimation in terms of hypothesis discriminability. On the contrary, in cases where the parametric model does not accurately match the distribution of the data (such as the NIST-face and AReM datasets), using non-parametric estimation results in significantly better hypothesis discriminability.

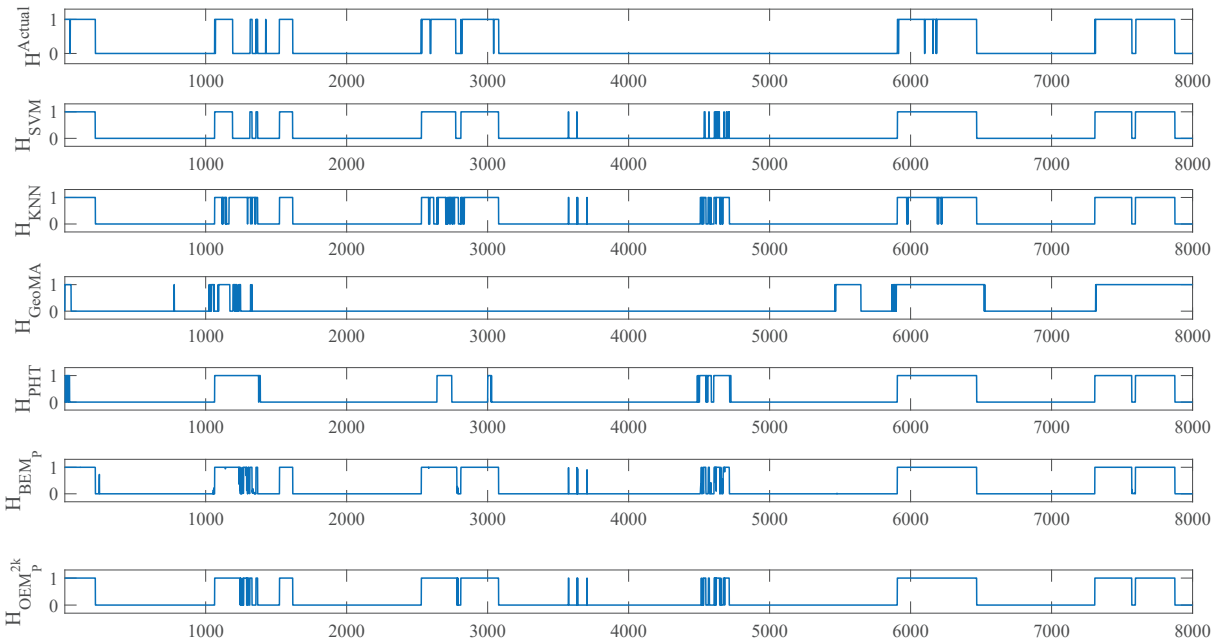

Figure 4.15: The actual (top row) and estimated values of the state of nature at each time instance using different methods (SVM, KNN, PHT, GeoMA, BEM_P, and OEM_P^{2k}) for the ROD dataset.

Table 4.2: Comparing hypothesis discriminability using parametric and non-parametric based estimation.

Dataset	ROD	NIST-face	AReM
Parametric (P-Gaussians)	.031	.070	.017
Non-Parametric (kernel-based)	.032	.052	.0098

4.5 Conclusion

An online expectation maximization (EM) based algorithm is presented for data fusion involving non-parametric model estimation and hypothesis testing based on observations from a network of heterogeneous sensors. The sensor measurements are assumed to be correlated and copula theory is used to model this correlation. Moreover, it is assumed that the statistical model for the sensor data is not completely known. The batch-mode EM is first developed for case studies of this problem including the Gaussian and product copulas where marginal density functions of the measurements of all sensors are estimated along with other model parameters and the state of nature is detected at all time instances. This algorithm is then extended to an online EM based approach. In the online method, upon

receiving sensors measurements at each time instance, the density functions and model parameters are updated and the state nature at the current time is detected. Results obtained from real-world data show significant improvements in hypothesis testing compared to other unsupervised and even some supervised learning methods. Moreover, in the case where data are correlated, the proposed method including copula modeling outperforms the method ignoring the correlation in sensors measurements while in the case where the data are independent, given enough data samples, the performance of the proposed method converges to that of the method which correctly assumes an independent data model.

References

[1] V. Becker and W. Kleiminger, "Exploring zero-training algorithms for occupancy detection based on smart meter measurements," Computer Science - Research and Development, 2017.

[2] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models," Energy and Buildings, vol. 112, pp. 28-39, 2016.

[3] E. Arens, M. Shi, T. Webster, and D. Wang, "How the number and placement of sensors controlling room air distribution systems affect energy use and comfort," in ICEBO-International Conference for Enhanced Building Operations. Energy Systems Laboratory (http://esl.tamu.edu); Texas A& M University (http://www.tamu.edu). Available electronically from http : / /hdl .handle .net /1969 .1 /5187, 2002.

[4] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton, "Codeblue: An ad hoc sensor network infrastructure for emergency medical care," in International workshop on wearable and implantable body sensor networks, vol. 5, 2004.

[5] T. R. Fulford-Jones, G.-Y. Wei, and M. Welsh, "A portable, low-power, wireless two-lead ekg system," in Engineering in Medicine and Biology Society, 2004. IEMBS 04. 26th Annual International Conference of the IEEE, vol. 1. IEEE, 2004, pp. 2141-2144.

[6] K. Lorincz, D. J. Malan, T. R. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton, "Sensor networks for emergency response: challenges and opportunities," IEEE pervasive Computing, vol. 3, no. 4, pp. 16-23, 2004.

[7] M. Welsh, D. Myung, M. Gaynor, and S. Moulton, "Resuscitation monitoring with a wireless sensor network." in Circulation, vol. 108, no. 17.

[8] F. Palumbo, C. Gallicchio, R. Pucci, and A. Micheli, "Human activity recognition using multisensor data fusion based on reservoir computing," Journal of Ambient Intelligence and Smart Environments, vol. 8, no. 2, pp. 87-107, 2016.

[9] F. Palumbo, P. Barsocchi, C. Gallicchio, S. Chessa, and A. Micheli, "Multisensor data fusion for activity recognition based on reservoir computing," in International Competition on Evaluating AAL Systems through Competitive Benchmarking. Springer, 2013, pp. 24-35.

[10] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in Wearable Computers (ISWC), 2012 16th Inter- national Symposium on. IEEE, 2012, pp. 108-109.

[11] A. Reiss and D. Stricker, "Creating and benchmarking a new dataset for physical activ- ity monitoring," in Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments. ACM, 2012, p. 40.

[12] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen, "Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition," in Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems. ACM, 2015, pp. 127-140.

[13] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, Rojas, A. Saez, and C. Villalonga, "mhealthdroid: a novel framework for agile development of mobile health applications," in International Workshop on Ambient Assisted Living. Springer, 2014, pp. 91-98.

[14] O. Banos, C. Villalonga, R. Garcia, A. Saez, M. Damas, J. A. Holgado-Terriza, S. Lee, H. Pomares, and I. Rojas, "Design, implementation and validation of a novel open framework for agile development of mobile health applications," Biomedical engineering online, vol. 14, no. 2, p. S6, 2015.

[15] A. R. de la Concepcin, R. Stefanelli, D. Trinchero, "Ad-hoc multilevel wireless sensor networks for distributed microclimatic diffused monitoring in precision agriculture," 2015 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet), pp. 14-16, 2015.

[16] W. Balid, H. Tafish, H. H. Refai, "Versatile real-time traffic monitoring system using wireless smart sensors networks," 2016 IEEE Wireless Communications and Networking Conference, pp. 1-6, 2016.

[17] K. Veeramachaneni, L. A. Osadciw, and P. K. Varshney, "An adaptive multimodal biometric management algorithm," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 35, no. 3, pp. 344-356, 2005.

[18] S. Iyengar, P. Varshney, and T. Damarla, "A parametric copula-based framework for hypothesis testing using heterogeneous data," IEEE Transactions on Signal Processing, vol. 59, no. 5, pp. 2308-2319, May 2011.

[19] L. Hong and A. Jain, "Integrating faces and fingerprints for personal identification," IEEE transactions on pattern analysis and machine intelligence, vol. 20, no. 12, pp. 1295-1307, 1998.

[20] S. Prabhakar and A. K. Jain, "Decision-level fusion in fingerprint verification," Pattern Recognition, vol. 35, no. 4, pp. 861-874, 2002.

[21] M. Haghighat, M. Abdel-Mottaleb, and W. Alhalabi, "Discriminant correlation analysis: Real-time feature level fusion for multimodal biometric recognition," IEEE Transactions on Information Forensics and Security, vol. 11, no. 9, pp. 1984-1996, 2016.

[22] B. He and Z. Liu, "Multimodal functional neuroimaging: Integrating functional MRI and EEG/MEG," Biomedical Engineering, IEEE Reviews in, vol. 1, pp. 23-40, 2008.

[23] T. Deneux and O. Faugeras, "EEG-FMRI fusion of paradigm-free activity using kalman filtering," Neural Computation, vol. 22, no. 4, pp. 906-948, April 2010.

[24] J. Daunizeau, C. Grova, G. Marrelec, J. Mattout, S. Jbabdi, M. Pélégrini-Issac, J.-M. Lina, and H. Benali, "Symmetrical event-related EEG/FMRI information fusion in a variational Bayesian frame-work," Neuroimage, vol. 36, no. 1, pp. 69-87, 2007.

[25] M. Luessi, S. D. Babacan, R. Molina, J. R. Booth, and A. K. Kat- saggelos, "Bayesian symmetrical EEG/FMRI fusion with spatially adaptive priors," Neuroimage, vol. 55, no. 1, pp. 113-132, 2011.

[26] C. F. Caskey, M. Hlawitschka, S. Qin, L. M. Mahakian, R. D. Cardiff, J. M. Boone, and K. W. Ferrara, "An open environment CT-US fusion for tissue segmentation during interventional guidance," PloS one, vol. 6, no. 11, p. e27372, 2011.

[27] C. F. Caskey, M. Hlawitschka, Sh. Qin, L. M. L. M. Mahakian, R. D. Cardiff, J. M. Boone, K. W. Ferrara, "An open environment CT-US fusion for tissue segmentation during interventional guidance," PloS one, vol. 6, no. 11, pp. e27372, 2011.

[28] O. Ukimura, M. Mitterberger, K. Okihara, T. Miki, G. M. Pinggera, R. Neururer, R. Peschel, F. Aigner, J. Gradl, G. Bartsch et al., "Real- time virtual ultrasonographic radiofrequency ablation of renal cell car- cinoma," BJU international, vol. 101, no. 6, pp. 707-711, 2008.

[29] T. Kitada, T. Murakami, N. Kuzushita, K. Minamitani, K. Nakajo, K. Osuga, E. Miyoshi, H. Nakamura, B. Kishino, S. Tamura et al., "Effectiveness of real-time virtual sonography-guided radiofrequency ablation treatment for patients with hepatocellular carcinomas," Hepatology Research, vol. 38, no. 6, pp. 565-571, 2008.

[30] R. Cutler and L. Davis, "Look who's talking: speaker detection using video and audio correlation," in 2000 IEEE International Conference on Multimedia and Expo (ICME 2000), vol. 3, 2000, pp. 1589-1592 vol.3.

[31] M. J. Beal, N. Jojic, and H. Attias, "A graphical model for audiovisual object tracking," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 25, no. 7, pp. 828-836, July 2003.

[32] Y. Rui and Y. Chen, "Better proposal distributions: object tracking using unscented particle filter," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), vol. 2, 2001, pp. II-786-II-793 vol.2.

[33] D. Gatica-Perez, G. Lathoud, J. Odobez, and I. McCowan, "Audiovisual probabilistic tracking of multiple speakers in meetings," IEEE Transactions on Audio, Speech, and Language Processing, vol. 15, no. 2, pp. 601-616, Feb 2007.

[34] K. Sohraby, D. Minoli, and T. Znati, Wireless sensor networks: technology, protocols, and applications. John Wiley & Sons, 2007.

[35] I. Benkhelifa, N. Nouali-Taboudjemat, and S. Moussaoui, "Disaster management projects using wireless sensor networks: An overview," in 2014 28th International

Conference on Advanced Information Networking and Applications Workshops, May 2014, pp. 605-610.

[36] M. Bahrepour, N. Meratnia, M. Poel, Z. Taghikhaki, and P. J. M. Havinga, "Distributed event detection in wireless sensor networks for disaster management," in 2010 International Conference on Intelligent Networking and Collaborative Systems, Nov 2010, pp. 507-512.

[37] R. I. da Silva, V. D. D. Almeida, A. M. Poersch, and J. M. S. Nogueira, "Wireless sensor network for disaster management," in 2010 IEEE Net- work Operations and Management Symposium - NOMS 2010, April 2010, pp. 870-873.

[38] J. Burrell, T. Brooke, and R. Beckwith, "Vineyard computing: Sensor networks in agricultural production," IEEE Pervasive computing, vol. 3, no. 1, pp. 38-45, 2004.

[39] T. Wark, P. Corke, P. Sikka, L. Klingbeil, Y. Guo, C. Crossman, P. Valencia, D. Swain, and G. Bishop-Hurley, "Transforming agriculture through pervasive wireless sensor networks," IEEE Pervasive Comput- ing, vol. 6, no. 2, pp. 50-57, April 2007.

[40] C. T. Kone, A. Ha?d, and M. Boushaba, "Performance management of IEEE 802.15.4 wireless sensor network for precision agriculture," IEEE Sensors Journal, vol. 15, no. 10, pp. 5734-5747, Oct 2015.

[41] M. Friesen, R. Jacob, P. Grestoni, T. Mailey, M. R. Friesen, and R. D. McLeod, "Vehicular traffic monitoring using bluetooth scanning over a wireless sensor network," Canadian Journal of Electrical and Computer Engineering, vol. 37, no. 3, pp. 135-144, Summer 2014.

[42] H. M. Sherif, M. A. Shedid, and S. A. Senbel, "Real time traffic accident detection system using wireless sensor network," in 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR), Aug 2014, pp. 59-64.

[43] M. Mousa, M. Abdulaal, S. Boyles, and C. Claudel, "Wireless sensor network-based urban traffic monitoring using inertial reference data," in 2015 International Conference on Distributed Computing in Sensor Systems, June 2015, pp. 206-207.

[44] S. Faye and C. Chaudet, "Characterizing the topology of an urban wire- less sensor network for road traffic management," IEEE Transactions on Vehicular Technology, vol. 65, no. 7, pp. 5720-5725, July 2016.

[45] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," Ad Hoc Networks, vol. 6, no. 4, pp. 621-655, 2008.

[46] D. N. Sandeep and V. Kumar, "Review on clustering, coverage and connectivity in underwater wireless sensor networks: A communication techniques perspective," IEEE Access, vol. 5, pp. 11 176-11 199, 2017.

[47] S. M. Dehnavi, M. Ayati, and M. R. Zakerzadeh, "Three dimensional target tracking via underwater acoustic wireless sensor network," in 2017 Artificial Intelligence and Robotics (IRANOPEN), April 2017, pp. 153-157.

[48] P. Gjanci, C. Petrioli, S. Basagni, C. Phillips, L. Boloni, and D. Turgut, "Path finding for maximum value of information in multi-modal underwater wireless sensor networks," IEEE Transactions on Mobile Computing, vol. PP, no. 99, pp. 1, 2017.

[49] M. Rao, N. K. Kamila, and K. V. Kumar, "Underwater wireless sensor network for tracking ships approaching harbor," in 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Oct 2016, pp. 1872-1876.

[50] E. Soltanmohammadi and M. Naraghi-Pour, "Nonparametric density estimation, hypotheses testing, and sensor classiffication in centralized detection," Information Forensics and Security, IEEE Transactions on, vol. 9, no. 3, pp. 426-435, March 2014.

[51] A. Sundaresan and P. K. Varshney, "Location estimation of a random signal source based on correlated sensor observations," IEEE Transactions on Signal Processing, vol. 59, no. 2, pp. 787-799, Feb 2011.

[52] X. Luo, M. Dong, and Y. Huang, "On distributed fault-tolerant detection in wireless sensor networks," IEEE Transactions on Computers, , vol. 55, no. 1, pp. 58-70, jan. 2006.

[53] P. Varshney, Distributed Detection and Data Fusion, 1st ed. New York: Springer-Verlag, 1997.

[54] S. Marano, V. Matta, and L. Tong, "Distributed detection in the presence of byzantine attacks," Signal Processing, IEEE Transactions on, vol. 57, no. 1, pp. 16-29, jan. 2009.

[55] R. Blum, S. Kassam, and H. Poor, "Distributed detection with multiple sensors I. advanced topics," Proceedings of the IEEE, vol. 85, no. 1, pp. 64-79, jan 1997.

[56] Q. Tian and E. Coyle, "Optimal distributed detection in clustered wireless sensor networks," IEEE Transactions on Signal Processing, vol. 55, no. 7, pp. 3892-3904, July 2007.

[57] H. Joe, J. J. Xu, "The Estimation Method of Inference Functions for Margins for Multivariate Models," Faculty Research and Publications, Oct 1996.

[58] J. O. Berger, V. De Oliveira, B. Sanso, "Objective Bayesian Analysis of Spatially Correlated Data," Journal of the American Statistical Association, vol. 96, no. 456, pp. 1361-1374, 2001.

[59] N. Li, Y. Liu, F. Wu, B. Tang, "OD Model: A Data Correlation Model Based on Spatial Location in Wireless Sensor Networks," 2010 WASE International Conference on Information Engineering, vol. 1, pp. 268-271, Aug. 2010.

[60] S. Iyengar, R. Niu, and P. Varshney, "Fusing dependent decisions for hypothesis testing with heterogeneous sensors," Signal Processing, IEEE Transactions on, vol. 60, no. 9, pp. 4888-4897, Sept 2012.

[61] A. Sundaresan, P. Varshney, and N. Rao, "Copula-based fusion of correlated decisions," IEEE Transactions on Aerospace and Electronic Systems, vol. 47, no. 1, pp. 454-471, January 2011.

[62] S. Iyengar, P. Varshney, and T. Damarla, "On the detection of footsteps based on acoustic and seismic sensing," in Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers, 2007 (ACSSC 2007). , Nov 2007, pp. 2248-2252.

[63] V. Aalo, "On distributed detection with correlated sensors: two examples," IEEE Transactions on Aerospace and Electronic Systems, vol. 25, no. 3, pp. 414-421, May 1989.

[64] P. Willett, P. Swaszek, and R. Blum, "The good, bad and ugly: distributed detection of a known signal in dependent Gaussian noise," IEEE Transactions on Signal Processing,, vol. 48, no. 12, pp. 3266-3279, Dec 2000.

[65] E. Drakopoulos and C. Lee, "Optimum multisensor fusion of correlated local decisions," IEEE Transactions on Aerospace and Electronic Systems, vol. 27, no. 4, pp. 593-606, Jul 1991.

[66] A. Jindal and K. Psounis, "Modeling spatially correlated data in sen- sor networks," ACM Transactions on Sensor Networks (TOSN), vol. 2, no. 4, pp. 466-499, 2006.

[67] Y. Stitou, N. Lasmar, and Y. Berthoumieu, "Copulas based multivariate gamma modeling for texture classification," in IEEE International Conference on Acoustics, Speech and Signal Processing, 2009 (ICASSP 2009), April 2009, pp. 1045-1048.

[68] N. Brunel, W. Pieczynski, and S. Derrode, "Copulas in vectorial hidden markov chains for multicomponent image segmentation." in IEEE International Conference on Acoustics, Speech and Signal Processing 2005 (ICASSP 2005), vol. 2, pp. 717-720.

[69] G. Mercier, G. Moser, and S. Serpico, "Conditional copulas for change detection in heterogeneous remote sensing images," Geoscience and Re- mote Sensing, IEEE Transactions on, vol. 46, no. 5, pp. 1428-1441, May 2008.

[70] B. W. Silverman, Density Estimation for Statistics and Data Analysis. Chapman & Hall, 1986.

[71] C. M. Bishop, Pattern Recognition and Machine Learning, 1st ed. Springer, 2006.

[72] S. Sobhiyeh and M. Naraghi-Pour, "Hypothesis testing with dependent observations," IEEE Transactions on Signal Processing, vol. 65, no. 5, pp. 1183-1195, March 2017.

[73] D. M. Titterington, "Recursive parameter estimation using incomplete data," Journal of the Royal Statistical Society. Series B (Methodological), pp. 257-267, 1984.

[74] P.-J. Chung and J. F. Böhme, "Recursive EM and sage-inspired algorithms with application to DOA estimation," IEEE Transactions on Signal Processing, vol. 53, no. 8, pp. 2664-2677, 2005.

[75] Z. Liu, J. Almhana, V. Choulakian, and R. McGorman, "Online EM algorithm for mixture with application to internet traffic modeling," Computational statistics & data analysis, vol. 50, no. 4, pp. 1052-1071, 2006.

[76] S. Wang and Y. Zhao, "Almost sure convergence of Titterington's recursive estimator for mixture models," Statistics & probability letters, vol. 76, no. 18, pp. 2001-2006, 2006.

[77] O. Cappé and E. Moulines, "Online EM algorithm for latent data models," Journal of the Royal Statistical Society: Series B, Royal Statistical Society, vol. 71, no. 3, pp. 593-613, 2009.

[78] O. Cappé, "Online expectation-maximisation," Mixtures: Estimation and Applications, pp. 31-53, 2011.

[79] N. J. Higham, "Computing the nearest correlation matrixa problem from finance," IMA Journal of Numerical Analysis, vol. 22, no. 3, pp. 329-343, 2002. [Online]. Available: http://imajna.oxfordjournals.org/

[80] B. Chai, C. Jia, and J. Yu, "An online expectation maximization algorithm for exploring general structure in massive networks," Physica A: Statistical Mechanics and its Applications, vol. 438, pp. 454-468, 2015.

[81] M. G. Hall, A. V. Oppenheim, and A. S. Willsky, "Time-varying parametric modeling of speech," Signal Processing, vol. 5, no. 3, pp. 267-285, 1983.

[82] R. Horst and N. V. Thoai, "Dc programming: overview," Journal of Optimization Theory and Applications, vol. 103, no. 1, pp. 1-43, 1999.

[83] P. D. Tao and L. T. H. An, "Convex analysis approach to dc programming: Theory, algorithms and applications," Acta Mathematica Vietnamica, vol. 22, no. 1, pp. 289-355, 1997.

[84] H. Tuy, "Global minimization of a difference of two convex functions," Mathematical Programming Studies, Nonlinear Analysis and Optimization, vol. 30, pp. 150-182, Feb 2009.

[85] E. Bouy'e, V. Durrleman, A. Nikeghbali, G. Riboulet, and T. Roncalli, Copulas for Finance-a reading guide and some applications, ser. Technical Report. Credit Lyonnais, 2000.

[86] NIST Biometric Scores Set 2004 [Online]. Available: http://www.itl.nist.gov/iad/894.03/biometricscores/

Appendix A
Proof of Lemmas

A.1 Proof for Lemma 1

To solve (2.24), we define the Lagrangian function

$$\mathcal{L}_{m,\omega} = \tilde{Q}_{m,\omega}(\omega_{i,p,t}) + \epsilon_\omega \left[1 - \sum_{i=0}^{K-1}\sum_{p=1}^{P} \omega_{i,p,t}\right]. \tag{A.1}$$

Taking the derivative of $\mathcal{L}_{m,\omega}$ with respect to $\omega_{i,p,t}$ and setting it to zero we get

$$\frac{\partial \mathcal{L}_{m,\omega}}{\partial \omega_{i,p,t}} = \frac{\alpha^{(n,T)}(i,p,t)}{\omega_{i,p,t}} + \epsilon_\omega = 0. \tag{A.2}$$

Multiplying both sides of (A.2) by $\omega_{i,p,t}$ and summing the results over i and p gives $\epsilon_\omega = -\sum_{i=0}^{K-1}\sum_{p=1}^{P} \alpha^{(n,T)}(i,p,t) = -1$. From this we get that

$$\omega_{i,p,t} = \alpha^{(n,T)}(i,p,t). \tag{A.3}$$

A.2 Proof for Lemma 2

To solve (2.26), we take the derivative of $\tilde{Q}_{m,\lambda}\left(\lambda_{m,i}^{-1}\right)$ with respect to $\lambda_{m,i}^{-1}$ where $m \in \{\mathcal{G}, \mathcal{T}\}$. We have

$$\frac{\partial \tilde{Q}_{\mathcal{G},\lambda}\left(\lambda_{\mathcal{G},i}^{-1}\right)}{\partial \lambda_{\mathcal{G},i}^{-1}} = \sum_{t=1}^{T}\sum_{p=1}^{P} \frac{\alpha^{(n,T)}(i,p,t)}{2}\left[\mathbf{y}_{i,p}^{(n-1,T)}(t)\left(\mathbf{y}_{i,p}^{(n-1,T)}(t)\right)^{Tr} - \lambda_{\mathcal{G},i}\right], \tag{A.4}$$

$$\frac{\partial \hat{Q}_{\mathcal{T},\lambda}\left(\lambda_{\mathcal{T},i}^{-1}\right)}{\partial \lambda_{\mathcal{T},i}^{-1}} = \sum_{t=1}^{T}\sum_{p=1}^{P} \frac{\alpha^{(n,T)}(i,p,t)}{2}\left[-\lambda_{\mathcal{T},i} + \frac{(\eta+L)\mathbf{v}_{i,p}^{(n-1,T)}(t)(\mathbf{v}_{i,p}^{(n-1,T)}(t))^{Tr}}{\eta + (\mathbf{v}_{i,p}^{(n-1,T)}(t))^{Tr}\lambda_{\mathcal{T},i}^{-1}\mathbf{v}_{i,p}^{(n-1,T)}(t)}\right]. \tag{A.5}$$

Setting (A.4) and (A.5) to zero, we get

$$\lambda_{\mathcal{G},i} = \frac{\sum_{t=1}^{T}\sum_{p=1}^{P} \alpha^{(n,T)}(i,p,t)\mathbf{y}_{i,p}^{(n-1,T)}(t)(\mathbf{y}_{i,p}^{(n-1,T)}(t))^{Tr}}{TS_1^{(n,T)}(i)}, \tag{A.6}$$

and

$$\lambda_{\mathcal{T},i} = \frac{(\eta+L)}{TS_1^{(n,T)}(i)}\sum_{t=1}^{T}\sum_{p=1}^{P} \frac{\alpha^{(n,T)}(i,p,t)\mathbf{v}_{i,p}^{(n-1,T)}(t)(\mathbf{v}_{i,p}^{(n-1,T)}(t))^{Tr}}{\eta + (\mathbf{v}_{i,p}^{(n-1,T)}(t)^{Tr})\lambda_{\mathcal{T},i}^{-1}\mathbf{v}_{i,p}^{(n-1,T)}(t)}, \tag{A.7}$$

respectively.

A.3 Proof for Lemma 3

To solve (2.34), we take the derivative of $\tilde{Q}_{m,\mu}(\boldsymbol{\mu}_{i,p}^{-1})$ with respect to $\boldsymbol{\mu}_{i,p}$. For $m \in \{\mathcal{G}, \mathcal{T}, \mathcal{P}\}$, we have

$$\frac{\partial \tilde{Q}_{m,\mu}(\boldsymbol{\mu}_{i,p})}{\partial \boldsymbol{\mu}_{i,p}} = \sum_{t=1}^{T} \alpha^{(n,T)}(i,p,t)\boldsymbol{\Sigma}_{i,p}^{(n-1,T)}(\tilde{\lambda}_{m,i}^{(n-1,T)})^{-1}\boldsymbol{\Sigma}_{i,p}^{(n-1,T)}\left(\boldsymbol{\mu}_{i,p} - \mathbf{d}_t\right). \tag{A.8}$$

Setting (A.8) to zero, we get

$$\begin{aligned}&\boldsymbol{\Sigma}_{i,p}^{(n-1,T)}(\tilde{\lambda}_{m,i}^{(n-1,T)})^{-1}\boldsymbol{\Sigma}_{i,p}^{(n-1,T)}\sum_{t=1}^{T}\alpha^{(n,T)}(i,p,t)\mathbf{d}_t = \\ &\boldsymbol{\Sigma}_{i,p}^{(n-1,T)}(\tilde{\lambda}_{m,i}^{(n-1,T)})^{-1}\boldsymbol{\Sigma}_{i,p}^{(n-1,T)}\boldsymbol{\mu}_{i,p}\sum_{t=1}^{T}\alpha^{(n,T)}(i,p,t).\end{aligned} \tag{A.9}$$

However, since the matrices $\boldsymbol{\Sigma}_{i,p}^{(n-1,T)}$ and $(\tilde{\lambda}_{m,i}^{(n-1,T)})^{-1}$ are both invertible, (A.9) reduces to

$$\boldsymbol{\mu}_{i,p} = \frac{\sum_{t=1}^{T}\alpha^{(n,T)}(i,p,t)\mathbf{d}_t}{TS_4^{(n,T)}(i)}. \tag{A.10}$$

A.4 Proof for Lemma 4

To solve (2.35), we take the derivative of $\tilde{Q}_{m,\sigma}(\sigma_{i,p,l})$ with respect to $\boldsymbol{\Sigma}_{i,p}$. For $m \in \mathcal{M}$, we have

$$\frac{\partial \tilde{Q}_{m,\sigma}(\sigma_{i,l,p}^{-1})}{\partial \boldsymbol{\Sigma}_{i,p}} = \sum_{t=1}^{T}\alpha^{(n,T)}(i,p,t)\left[-\boldsymbol{\Sigma}_{i,p}^{-1} + \tilde{\lambda}_{m,i}^{(n-1,T)}\boldsymbol{\Sigma}_{i,p}\left(\boldsymbol{\mu}_{i,p}^{(n-1,T)} - \mathbf{d}_t\right)\left(\boldsymbol{\mu}_{i,p}^{(n-1,T)} - \mathbf{d}_t\right)^{Tr}\right]. \tag{A.11}$$

Setting (A.11) to zero, we get

$$\boldsymbol{\Sigma}_{i,p}^{-1}(\tilde{\lambda}_{m,i}^{(n-1,T)})^{-1}\boldsymbol{\Sigma}_{i,p}^{-1} = \frac{\sum_{t=1}^{T}\alpha^{(n,T)}(i,p,t)\left(\boldsymbol{\mu}_{i,p}^{(n-1,T)} - \mathbf{d}_t\right)\left(\boldsymbol{\mu}_{i,p}^{(n-1,T)} - \mathbf{d}_t\right)^{Tr}}{TS_4^{(n,T)}(i)}. \tag{A.12}$$

Consequently, for $m = \mathcal{G}$, we have

$$\sigma_{i,l,p} = \frac{\gamma_{i,l,p}^{(n,T)}}{-\beta_{i,l,p}^{(n,T)} + \sqrt{(\beta_{i,l,p}^{(n,T)})^2 + \gamma_{i,l,p}^{(n,T)}S_4^{(n,T)}}}, \tag{A.13}$$

and for $m \in \{\mathcal{T}, \mathcal{P}\}$, we have

$$\sigma_{i,l,p} = \frac{\sum_{t=1}^{T}\alpha^{(n,T)}(i,p,t)(d_{l,t} - \mu_{i,l,p}^{(n-1,T)})^2}{TS_4^{(n,T)}(i,p)}, \tag{A.14}$$

Appendix B
Expectation Step of Algorithm Proposed in Chapter 3

Following the discussion in section 3.3.1, $\alpha_1^{(n)}(i,t) \triangleq E[h_{i,t}h_{1-i,t-1}|D;\Theta^{(n-1)}]$ and $\alpha_2^{(n)}(i,t) \triangleq E[h_{i,t}h_{i,t-1}|D;\Theta^{(n-1)}]$ are calculated in the *Expectation step* of the proposed EM-based algorithm. In this section, we discuss the evaluation of $\alpha_j^{(n)}(i,t),\ j=1,2$. According to the definition we have

$$\alpha_j^{(n)}(i,t) = Pr(h_{i,t}=1, h_{i,t-1}=j-1|D;\Theta^{(n-1)}) = \frac{\sum_{\tilde{H}} P(D, h_{i,t}=1, h_{i,t-1}=j-1, \tilde{H}|\Theta^{(n-1)})}{\sum_{H} P(D,H|\Theta^{(n-1)})}, \tag{B.1}$$

where, $\tilde{H}$ is a $2\times(T-2)$ matrix containing all columns of H except for the two columns corresponding to times $t-1$ and t. The number of terms in the summation in (B.1) increases exponentially with T. Therefore, we propose using a reasonable approximation to calculate $\alpha_j(i,t)$ as presented in (B.2), where only the data from two time instances, namely the current and the previous time, are directly involved in the decision made about the state of nature at the current and the previous time. Thus, we have

$$\alpha_j^{(n)}(i,t) \approx Pr(h_{i,t}=1, h_{i,t-1}=j-1|\mathbf{d}_t,\mathbf{d}_{t-1};\Theta^{(n-1)}) = \frac{Pr(\mathbf{d}_t,\mathbf{d}_{t-1}|h_{i,t}=1,h_{i,t-1}=j-1;\Theta^{(n-1)})\tilde{\phi}^{(n-1)}_{i,n_{j,i},t}}{\sum_{k=0}^{1}\sum_{l=1}^{2} Pr(\mathbf{d}_t,\mathbf{d}_{t-1}|h_{k,t}=1,h_{k,t-1}=l-1;\Theta^{(n-1)})\tilde{\phi}^{(n-1)}_{k,n_{l,k},t}} \tag{B.2}$$

where, $n_{l,k}=|l-2+k|$,

$$\begin{aligned} &Pr(\mathbf{d}_t,\mathbf{d}_{t-1}|h_{i,t}=1,h_{i,t-1}=0;\Theta^{(n-1)}) = \\ &Pr(\mathbf{d}_t|\mathbf{d}_{t-1},h_{i,t}=1,h_{i,t-1}=0;\Theta^{(n-1)})Pr(\mathbf{d}_{t-1}|h_{i,t}=1,h_{i,t-1}=0;\Theta^{(n-1)}) = \\ &Pr(\mathbf{d}_t|h_{i,t}=1,h_{i,t-1}=0;\Theta^{(n-1)})Pr(\mathbf{d}_{t-1}|h_{i,t}=1,h_{i,t-1}=0;\Theta^{(n-1)}) = \\ &\left[c_1\left(F_{i,1}(d_{1,t}),\cdots,F_{i,L}(d_{L,t})\right)\prod_{l=1}^{L} f_{i,l}(d_{l,t})\right]\left[c_1\left(F_{i,1}(d_{1,t-1}),\cdots,F_{i,L}(d_{L,t-1})\right)\prod_{l=1}^{L} f_{i,l}(d_{l,t-1})\right], \end{aligned} \tag{B.3}$$

and,

$$\begin{aligned} &Pr(\mathbf{d}_t,\mathbf{d}_{t-1}|h_{i,t}=1,h_{i,t-1}=1;\Theta^{(n-1)}) = \\ &Pr(\mathbf{d}_t|\mathbf{d}_{t-1},h_{i,t}=1,h_{i,t-1}=1;\Theta^{(n-1)})Pr(\mathbf{d}_{t-1}|h_{i,t}=1,h_{i,t-1}=1;\Theta^{(n-1)}) \\ &= \left(\prod_{l=1}^{L} f_{i,l}(d_{l,t}|d_{l,t-1})\right) c_2\left(F_{i,1}(d_{1,t}|d_{l,t-1}),\cdots,F_{i,L}(d_{L,t}|d_{l,t-1})\right) \\ &\times \left(\prod_{l=1}^{L} f_{i,l}(d_{l,t-1})\right) c_1\left(F_{i,1}(d_{1,t-1}),\cdots,F_{i,L}(d_{L,t-1})\right). \end{aligned} \tag{B.4}$$

Vita

Sima Sobhiyeh was born in Tehran, Iran in 1989. She received her B.Sc. degree in Electrical Engineering from the Amirkabir University of Technology (AUT) in 2011. She received the M.Sc. degree from the same institution in 2013. She joined the School of Electrical Engineering and Computer Science, Louisiana State University (LSU), Baton Rouge, Louisiana, in August 2014. Since then she has been working towards a Ph.D. degree in systems (communication & signal processing) under the supervision of Dr. Mort Naraghi-Pour. During her Ph.D. studies, she has been working at LSU as a Graduate Research/Teaching Assistant. Her research interests include signal/image processing. She plans to complete her doctoral studies in summer 2018.