NAIST-IS-DD1761022

Doctoral Dissertation

Hierarchical Segmentation Approach to Detecting Switching Interaction using Bayesian Non-Parametrics

Jeric C. Briones

December 15, 2020

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Jeric C. Briones

Thesis Committee:

Professor Kazushi Ikeda	(Supervisor)
Professor Shoji Kasahara	(Co-supervisor)
Research Associate Professor Takatomi Kubo	(Co-supervisor)
Assistant Professor Mark Anthony C. Tolentino	(Ateneo de Manila University)

Hierarchical Segmentation Approach to Detecting Switching Interaction using Bayesian Non-Parametrics[*]

Jeric C. Briones

Abstract

Two common topics in time series analysis are segmentation and interaction detection. On one hand, existing segmentation methods either estimate the dynamics of the time series, or extract its hierarchical structure, but usually not both. However, we are interested in a segmentation method that discovers both the dynamics and structure of the data. On the other hand, existing interaction detection methods generally assume that interaction information is time-invariant. However, we are interested in detecting interaction that changes over time, since dynamics of time series sequences may switch. While these two problems are seemingly unconnected, we want to use a segmentation approach for interaction detection.

In this thesis, the problem of segmentation and interaction detection was tackled, with the end goal of using the segments to detect interaction. For the first part, we segmented time series sequences from dynamical systems that have a hierarchical structure. To segment these time series sequences, we proposed a method combining the segmentation by beta process - autoregressive hidden Markov model and the double articulation by nested Pitman-Yor language model. For the second part, we detected switching interaction using the resulting autoregressive models by generating surrogates and performing non-parametric tests.

Based on experiments using synthetic toy datasets and real motion dataset, we demonstrated that the proposed method segments the time series in both low and

[*] Doctoral Dissertation, Graduate School of Information Science, Nara Institute of Science and Technology, December 15, 2020.

high levels, with errors smaller than those of the existing double articulation analyzer. We were also able to extract the dynamics of the time series using autoregressive coefficients. Using the discovered dynamics, the proposed method also inferred interaction information with good specificity. These results then suggest that switching interaction could be detected using the proposed segmentation approach.

Keywords:

time series analysis, non-parametric Bayesian, segmentation, interaction detection, switching interaction

Contents

List of Figures

List of Tables

List of Variables and Notations

Notation	**Description**
$\mathbf{y}_t^{(i)}$	observed variable at time t for time series i
$z_t^{(i)}$	latent variable at time t for time series i
$\mathbf{e}_t^{(i)}(k)$	Gaussian noise at time t for time series i when state k is active
$\pi_k^{(i)}$	transition probability vector for time series i when the initial state is k.
$\mathsf{UB}k$	k^{th} UB label
$\mathsf{EB}k$	k^{th} EB label
$[\mathsf{EB}k_1 \cdots \mathsf{EB}k_c \cdots \mathsf{EB}k_C]$	UB expressed in terms of its C component EBs
$\mathbf{f}_i$	binary EB vector for time series i
$\mathrm{AR}(r)$	autoregressive model with lag order r
$\mathbf{A}_k$	AR coefficient matrix of state k
Σ_k	noise covariance matrix of state k
$B \sim \mathrm{BP}\left(c, B_0\right)$	draw of beta process $\mathrm{BP}\left(c, B_0\right)$
$X \sim \mathrm{BeP}(B)$	draw of Bernoulli process $\mathrm{BeP}(B)$
$\Sigma \sim \mathrm{IW}\left(S_0, n_0\right)$	draw of inverse Wishart distribution $\mathrm{IW}\left(S_0, n_0\right)$
$\mathbf{A} \sim \mathcal{MN}\left(\mathbf{A}; M, \Sigma, L\right)$	draw of matrix normal distribution $\mathcal{MN}\left(\mathbf{A}; M, \Sigma, L\right)$
$G \sim \mathrm{PY}(G_0, \theta, d)$	draw of Pitman-Yor process $\mathrm{PY}(G_0, \theta, d)$
G_n^{W}	n-gram word model
G_n^{C}	n-gram character model
τ_{k_c}	time weight for $\mathsf{EB}k_c$
SME_k	binary significance matrix of $\mathsf{EB}k$
SMU_k	significance matrix of $\mathsf{UB}k$
$\mathbf{L}q$	toy subdataset generated from $\mathrm{AR}(q)$
$\mathbf{T}s$	toy subdataset generated from $\mathrm{AR}(1)$, with s time series sequences
$\mathbf{y}_t^{(q,i)}$	observed variable at time t for time series i from Set $\mathbf{L}q$
S-ARr	collection of segmentation results when $\mathrm{AR}(r)$ and (sub)dataset $\mathbf{S}$ were used

1. Introduction

1.1 Background

Time series modelling and analysis have been the subject of various studies in different domains, such as biology, natural language processing, speech recognition, and motion analysis. In these fields, topics such as time series segmentation and interaction detection frequently appear. In fact, several methods to segment time series sequences, as well as techniques to detect interaction between agents exist in literature. While seemingly unconnected, it is interesting to see if segmentation methods can be used to improve and/or complement interaction detection techniques.

1.1.1 Segmentation

Studies on time series segmentation usually fit a mathematical model to the data, with the obtained model used to identify and differentiate the segments [4, 35, 49]. However, there are instances when basic time series models (such as autoregressive or moving-average models) or simple linear dynamical systems may not be enough to capture the dynamics observed. Instead of fitting a single model to all the time series, dynamics could be better captured if the time series sequences are allowed to switch across models [15]. If this is the case, the question of how many segments exist in the sequence arises, as this number is not known beforehand. To solve this problem, non-parametric Bayesian methods are usually employed. Beta process - autoregressive hidden Markov model (BP-AR-HMM) [16, 18], for example, uses non-parametric Bayesian approach to identify the segments without specifying the number of segments beforehand.

Aside from the number of segments, the structure of the time series sequences is another possible issue with segmentation, as some of these sequences may have a hierarchical structure. For example, in natural language processing, sentences consist of words, where each word consists of letters. In the same vein, motion data can be viewed as a sequence of semantic actions, with each action made up of *motion primitives* [26, 56, 61]. Given this hierarchical structure, methods using hierarchical models, rather than (switching) dynamical systems, might be more appropriate for analysis since the latter methods do not consider this hierarchical structure.

Examples of these non-parametric Bayesian methods for hierarchical models include the hierarchical hidden Markov model (HHMM) [13], the nested Pitman-Yor language model (NPYLM) for sentences [38], and the double articulation analyzer (DAA) for motion data [49]. However, while hierarchical models may be able to model the data as a hierarchical sequence of symbols, they might not be capable of taking into account the dynamics directly.

Despite its limitations, using dynamical systems is still particularly interesting because of its potential as a starting point for interaction analysis. However, it is also important to incorporate the hierarchical structure in the model. That is why it is necessary to develop a method that considers both dynamics and structure, with the end goal of using the said method for interaction detection.

1.1.2 Double Articulation Analyzer

One example of a method that discovers hierarchical structure in the data is the double articulation analyzer (DAA), an unsupervised segmentation method used for motion data. The aforementioned method uses sticky hierarchical Dirichlet process HMM (sticky HDP-HMM) [14] and NPYLM to segment motion data into elemental motion and unit motion, respectively. Elemental motions are essentially low-level behaviors that can be modelled by linear dynamics. As such, using sticky HDP-HMM can be viewed as modelling a time series sequence by fitting piece-wise linear functions to the said sequence. Unit motions, on the other hand, are more complex, high-level behaviors that are more semantically meaningful. Using NPYLM chunks together the discovered elemental motions to form unit motions.

Despite discovering the structure of the time series data, DAA does not consider more complex dynamics for its low-level behaviors. Specifically, DAA models the time series sequences by fitting segment-wise linear functions to the lower level of its structure. Real world data, however, can exhibit more complex dynamics in the lower level of the structure. This is also true for motion primitives, which are usually modelled as non-linear functions [7, 57]. As such, it would be more beneficial if the segmentation method can consider more complex dynamics for the low-level behaviors, while still being able to discover the inherent hierarchical structure of the data.

1.1.3 Interaction Detection

Like segmentation, several interaction detection methods exist in literature. These include methods that use depth maps and image processing to extract features from videos and still images. Then, techniques such as deep learning, kernel models, and support vector machines are applied to these features to detect interaction [5, 40, 58]. In some of these works, the focus is on detecting causal interaction [3, 27], including causal interaction detection in neuroscience applications [12, 24]. In these studies, usual approaches combine different methods from graph theory, autoregressive models, and Granger causality. Even if there is a rich literature for this particular field, existing methods generally share the same limitations, such as they require annotations (which could be quite expensive), and they extract only a single causal relationship for a given input sequence. However, most real-world time series data are not annotated, and could exhibit more than one causal relationship in a given sequence. That is, interaction information can *switch* (that is, to change and not remain static) within a given time series sequence. Using switching interaction analysis, such as [10], would be more appropriate. But the number of interaction patterns are generally not known beforehand as well.

Furthermore, most studies on interaction detection are based on video segmentation and analysis [40, 58]. Here, videos can be considered as low-dimensional data. However, if motion capture (mocap) sequences are used to detect behaviors and interactions, then more interaction information could possibly be drawn from these high-dimensional data. These are particularly useful when applied in the field of robotics, surveillance, and human activity recognition [2, 30, 45, 58]. Furthermore, allowing interactions to *switch* would definitely help in better detecting and understanding the interaction occurring in the data.

Given these limitations, detecting these switching interactions is of particular interest to us because of its potential to better identify causal relationships. Specifically, we are interested to develop an approach that detects interaction which switches from one causal relationship to another in a given (high-dimensional) time series sequence.

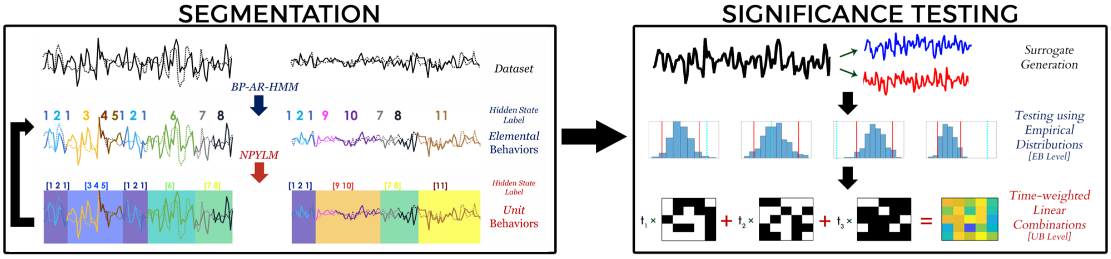


Figure 1.1: Illustration of the proposed approach, showing both the segmentation (left) and significance testing (right) steps.

1.2 Problem Setting

Given all these, we focus our attention on two things: segmentation and interaction detection. While they seem unrelated at first, we are actually interested in using the former to do the latter. That is, we are looking at using a segmentation approach to detect switching interaction from time series sequences.

In this thesis, the main problem was developing an interaction detection method using a segmentation approach. Specifically, this thesis proposed a method that enabled us to detect causal switching interactions from multiple multivariate time series sequences, each of which may exhibit various behaviors (and would thus have varying interaction information). A segmentation method was proposed to model hierarchically-structured sequences with dynamical systems. Specifically, we aimed to discover both the complex dynamics of the time series data and its inherent hierarchical structure using the proposed segmentation method.

1.3 Overview of Proposed Approach

The proposed segmentation approach to interaction detection had two steps: segmentation and significance testing. Figure 1.1 shows a schematic illustration of this proposed approach, showing the two aforementioned steps.

The first step was all about developing a segmentation method that can be used for interaction analysis. Specifically, we segmented the given time series sequences, where each resulting segment had assigned segment labels, as well as corresponding AR models. On one hand, while there are methods that uses dynamical systems for segmentation, they generally can not be used for sequences with hierarchical structure. On the other, there are methods that do consider the hierarchical structure of time series sequences, but they usually ignore the dynamics of the data. In our case, we are looking for a segmentation method for time series sequences from dynamical systems that have a hierarchical structure. To model these hierarchically-structured sequences with dynamical systems, we proposed a segmentation method that integrates the segmentation from BP-AR-HMM and the double articulation in NPYLM. The proposed method had the same structure as DAA. As such, it could also capture the hierarchical structure of the given time series sequences. However, the proposed method explicitly used dynamical systems (switching AR models to be exact) to model and capture the dynamics of the lower level of the given multivariate sequences. Lastly, ours allowed for asynchronous switching between segments across the multiple motion data sequences in consideration. This was possible because of the beta process used in BP-AR-HMM. On another note, both of the methods used in the proposed segmentation method were non-parametric Bayesian models, and had some hyperparameters to be chosen beforehand. Of particular interest for us was the AR lag order (to be used in BP-AR-HMM), since this hyperparameter dictated the complexity of the resulting models from segmentation.

The second step was about using the obtained segments and AR models to detect switching interactions. Specifically, the proposed unsupervised segmentation method was used to identify dynamic interaction from multiple multivariate time series sequences. To do this, the obtained AR models were used for hypothesis testing, which then determined which pairs of time series variables had significant interactions, and thus detected interaction between them. In this work, causal interaction was defined in terms of Granger causality [25]: interaction between two variables exist if one Granger causes the other. In other words, causal interaction was viewed as the dependence of an agent's action to the previous action of other agents, while switching interaction was defined as the series of resulting interactions caused by the changes in behavior within the same input sequence. Using a segmentation approach actually allowed us to skip using annotations completely. Moreover, detecting multiple causal relationships from a single input sequence can be possible, without having to manually partition the input data first. These were made possible by the use of AR models in the segmentation step. Of particular interest for us was the resulting AR coefficient matrices, as these are frequently used to detect interaction between variables [23, 24, 28].

Notice that in both steps, the dynamics of the time series sequences played a crucial role. We discovered the said dynamics when doing segmentation in the first part, which were then used to discover interactions in the second part. Combining the different methods mentioned beforehand, specifically using switching processes and dynamical models simultaneously, allowed us to detect causal interactions that vary (or switch) over time.

1.3.1 Scope of the Study

In theory, the proposed approach can be applied to any multivariate time series sequences that have hierarchical structure. However, this dissertation focused on applying the proposed segmentation approach to human motion data. Specifically, sequences considered were sequences of motion data involving interacting agents, whose component motion primitives can be described using AR models, and whose interaction switches from time to time [2, 45]. The proposed method was applied to two kinds of synthetic data: sets of switching AR time series sequences (with known AR lag order), and set of motion data generated using actual human motion capture data (with unknown AR lag order). These sets were used to determine the effectiveness of the proposed approach, in terms of segmentation and interaction detection.

Furthermore, of all the hyperparameters in the two non-parametric models to be used in the segmentation step, much of the focus was on exploring the effects of AR lag order. This parameter was given much importance since it determined the complexity of the model, which was eventually used in significance testing step. All other hyperparameters, on the other hand, were chosen similar to how these values were set in existing literature.

Finally, the second step of the proposed approach only involved pairwise interaction detection between two variables. That is, the proposed method only identified the existence of interaction between two variables at a time. We also did not quantify the detected interaction between these two variables, if any. Lastly, detecting spurious connections was beyond the scope of this dissertation.

1.3.2 Contribution of the Study

As previously mentioned, the main goal of this thesis was to develop an interaction detection method using a segmentation approach. To achieve this, a segmentation method was first proposed. This proposed segmentation method is the main contribution of this thesis.

First, existing segmentation methods either extract the dynamics of the data, or the structure of the data, but not both. The proposed segmentation method did both. That is, it discovered both the dynamics and the hierarchical structure inherent to the time series data. Furthermore, rather than fitting piece-wise linear functions to the lower level of the structure, the proposed segmentation method instead utilized switching autoregressive models. Doing so allowed us to capture more complex dynamics from the time series sequences. As extracted dynamics were more complex, the resulting segments were also more meaningful.

1.4 Organization of Dissertation

The rest of the dissertation is organized as follows: Chapters 2 and 3 show the steps in the proposed approach, with a brief introduction of the basic algorithms used. Specifically, the former will talk about the proposed segmentation method, while the latter will discuss how significance testing will be carried out. Chapters 4 and 5 outline the details of the synthetic experiments carried out and the corresponding results. Finally, Chapters 6 and 7 give some discussion of the results, as well as the conclusion and some recommendations.

2. Segmentation Method

For the segmentation step, we proposed a two-layered unsupervised segmentation method, where two non-parametric Bayesian methods are used. Specifically, the proposed segmentation method integrates BP-AR-HMM [16] and NPYLM [38] (Figure 2.1). The first layer applies BP-AR-HMM to the time series data, and discovers *elemental* behaviors (EB), which are low-level, short, simple actions (similar to motion primitives). Segmentation is indicated by the assigned EB labels at each time step. The obtained EB label sequences for each time series are then summarized, before being used as an input for the second layer. The EB labels are summarized by writing any recurring EB labels in the sequence only once. For example, the sequence $[1\ 1\ 1\ 2\ 2\ 1\ 1\ 1\ 1\ 3\ \cdots]$ is summarized as $[1\ 2\ 1\ 3\ \cdots]$. In the second layer, NPYLM is applied to the summarized sequence of EB labels, where the discovered EBs are grouped to form high-level semantic actions, called *unit* behaviors (UB). This results to a sequence of UBs, where each UB is a sequence of EBs expressed as AR models.

These two layers are then iterated a certain number of times, to improve segmentation and AR coefficient estimation accuracy. Iteration is done by using the results of the previous run as the initial segmentation of the next run. That is, the resulting UB labels from the second layer are used as initial EB labels for the first layer of the next iteration. Iteration is terminated either when the change in likelihood is below a certain threshold, or the number of maximum iterations is reached.

In the following, we briefly introduce the components of the proposed segmentation method, BP-AR-HMM and NPYLM.

2.1 Beta Process - Autoregressive Hidden Markov Model

BP-AR-HMM is an extension of hidden Markov model (HMM) where the latent variables z_t each represent an AR model with lag order r and parameter $\theta_{z_t} = \{\mathbf{A}_{z_t}, \Sigma_{z_t}\}$, and the observed variables $\mathbf{y}_t$ are described by the corresponding AR model. Furthermore, this model is also a non-parametric Bayesian model with a beta process (BP) prior (Figure 2.2). This prior is used to determine which specific EBs are active per time series sequence. Posterior inference is used to estimate the parameters of BP-AR-HMM, where Markov chain Monte Carlo (MCMC)

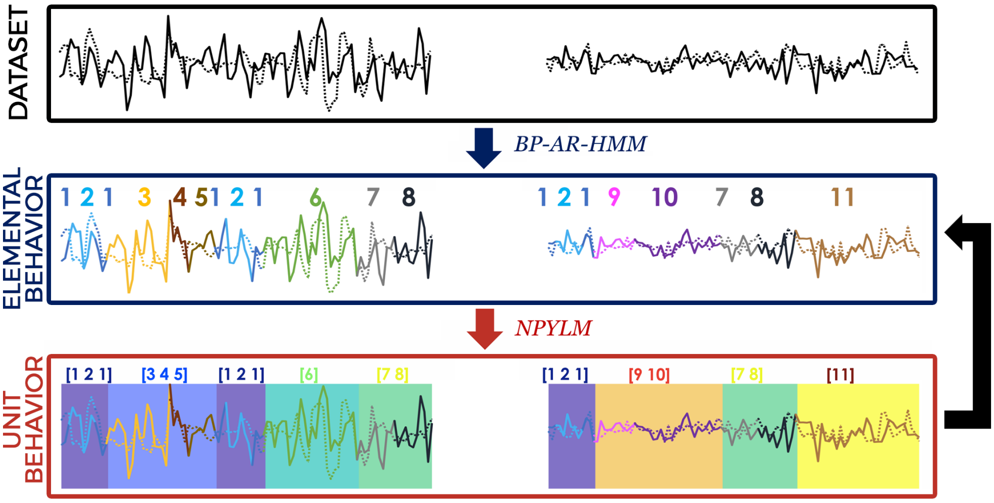


Figure 2.1: Illustration of the proposed segmentation method, showing the expected results for each step. Each time step is assigned both an EB label (line color) and UB label (background color). Summarized EB labels (colored numbers) and groupings of EB labels / UB labels (numbers in square brackets) are also indicated. Finally, included in the illustration is the iterative approach for segmentation.

algorithms are employed [16, 32]. This is because the posterior probability cannot be calculated in a closed form.

In the following, key concepts of BP-AR-HMM would be discussed to better understand the said model.

2.1.1 Beta Process

A beta process (BP) prior is placed on the vector of EB labels. This allows us to not specify the number of EBs beforehand, and thus have a library of potentially infinite EBs. BP is a completely random measure, denoted by $B \mid c, B_0 \sim \mathrm{BP}\left(c, B_0\right)$, and given by

$$B = \sum_{k=1}^{\infty} \omega_k \delta_{\theta_k}, \qquad \text{with } \alpha = B_0\left(\Theta\right), \tag{2.1}$$

where B_0 is a base measure, c the concentration parameter, and α the mass parameter. Using this prior also allows us to use an infinite-dimensional (binary) EB vector $\mathbf{f}$. A realization of EB vector for time series i, $\mathbf{f}_i \mid B \sim \mathrm{BeP}(B)$, is given by

$$\mathbf{f}_i = \sum_k f_{ik}\delta_{\theta_k}, \qquad \text{with } f_{ik} \sim \mathrm{Be}\left(\omega_k\right). \tag{2.2}$$

Here, $f_{ik} = 1$ if the k^{th} EB label (denoted by $\mathsf{EB}k$) is active for time series i.

2.1.2 AR-HMM

Let $\mathbf{y}_t^{(i)} \in \mathbb{R}^D$ be the D-dimensional observed variable at time t for time series i. As mentioned earlier, the observed variable $\mathbf{y}_t^{(i)}$ is described by an AR-HMM, with AR lag order r, latent variable (or EB label) $z_t^{(i)}$, and parameter $\theta_k = \{\mathbf{A}_k, \Sigma_k\}$. Basically, the value of the observed variables a time t is a linear combination of its past values, from time $t-r$ to time $t-1$. On the other hand, the parameters of the AR model (coefficient matrix $\mathbf{A}_k$ and noise covariance matrix Σ_k) are dependent on which latent variable $z_t^{(i)}$ is active. This latent variable switches at every time step, with the probability of it going from state k to state j being $\pi_{kj}^{(i)}$. Mathematically,

$$z_t^{(i)} \mid z_{t-1}^{(i)} \sim \pi_{z_{t-1}^{(i)}}^{(i)}, \tag{2.3}$$

$$\mathbf{y}_t^{(i)} \mid z_t^{(i)} \sim \mathcal{N}\left(\mathbf{A}_{z_t^{(i)}}\tilde{\mathbf{y}}_t^{(i)}, \Sigma_{z_t^{(i)}}\right), \quad \text{where } \mathbf{y}_t^{(i)} = \sum_{l=1}^{r} A_{l,z_t^{(i)}}\mathbf{y}_{t-l}^{(i)} + \mathbf{e}_t^{(i)}(z_t^{(i)}) \tag{2.4}$$

$$\mathbf{e}_t^{(i)}\left(z_t^{(i)}\right) \sim \mathcal{N}\left(0, \Sigma_{z_t^{(i)}}\right).$$

Here, $\pi_k^{(i)}$ refers to the transition probability for time series i when the initial EB label is k. However, not all EB labels are active for each time series. As such, *feature-constrained* transition distributions [16] are used. That is, given $\mathbf{f}_i$,

$$\pi_{kj}^{(i)} = \begin{cases} 0 & f_{ij} = 0 \\ P\left(z_t^{(i)} = j \mid z_{t-1}^{(i)} = k\right) & f_{ij} = 1 \end{cases}, \qquad \text{with } \sum_j \pi_{kj}^{(i)} = 1. \tag{2.5}$$

Furthermore, a gamma prior is placed on the transition matrix, with

$$\eta_{jk}^{(i)} \mid \gamma, \kappa \sim \mathrm{Gamma}\left(\gamma + \kappa\delta_{j,k}, 1\right) \tag{2.6}$$

$$\pi_j^{(i)} = \frac{\boldsymbol{\eta}_j^{(i)} \otimes \mathbf{f}_i}{\sum_{k|f_{ik}=1} \eta_{jk}^{(i)}}, \tag{2.7}$$

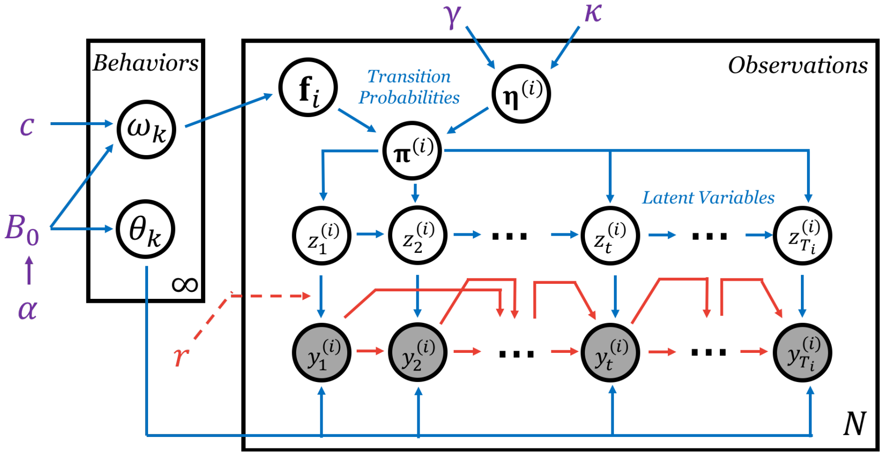


Figure 2.2: Graphical representation BP-AR-HMM, showing all the priors, processes, and hyperparameters used.

and γ, κ the transition and transition sticky parameter, respectively. Here, $\delta_{j,k}$ is the Kronecker delta function (defined as $\delta_{j,k} = 1$ for $j = k$, and $\delta_{j,k} = 0$ otherwise), and $\otimes$ is the Hadamard (or element-wise) vector product.

Finally matrix normal priors (including inverse Wishart prior) are placed on the dynamic parameters. That is,

$$\Sigma_k \mid n_0, S_0 \sim \text{IW}\left(S_0, n_0\right) \tag{2.8}$$

$$\mathbf{A}_k \mid \Sigma_k, M, L \sim \mathcal{MN}\left(\mathbf{A}_k; M, \Sigma_k, L\right) \tag{2.9}$$

where n_0 is the degrees of freedom, S_0 a scale matrix, M the mean dynamic matrix, and L, Σ_k defines covariance of $\mathbf{A}_k$.

2.1.3 Posterior Inference

Posterior inference is carried out by generating samples from the posterior distribution using Markov chain Monte Carlo (MCMC) algorithm. That is, the samples generated are (1) EB vector $\mathbf{f}$ given parameters $\boldsymbol{\theta}, \boldsymbol{\eta}$, (2) state sequences $\mathbf{z}$ given $\mathbf{f}$, $\boldsymbol{\theta}$, $\boldsymbol{\eta}$, and (3) auxiliary variables $\boldsymbol{\theta}$, $\boldsymbol{\eta}$ given $\mathbf{F}$ and $\mathbf{z}$. Moreover, the hyperparameters $\alpha, c, \kappa, \gamma$ are also sampled. Birth-death reversible jump MCMC sampling [16] and split-merge techniques [33] are used to generate unique EB vectors. In a nutshell, MCMC alternates between sampling $\mathbf{F}|\mathbf{y}, \boldsymbol{\theta}$ and $\boldsymbol{\theta}|\mathbf{y}, \mathbf{F}$ (with the hyperparameters sampled in between the cycles) to perform posterior inference.

2.1.4 Advantages

Using BP-AR-HMM as the first layer of the segmentation method provides some key advantages over the sticky HDP-HMM [14] used in DAA. First, we can segment multiple time series, and have asynchronous switching between segments at the same time. This means that the proposed segmentation method can discover common and unique EB labels across these sequences. This is not possible for sticky HDP-HMM, as it requires all the sequences to have exactly the same set of active EBs, and not just a subset from the library of EB labels. In addition, the difference between HDP and BP is better understood in the context of transition probability matrices. In the case of the former, states are assigned to each time step according to a transition matrix shared by all time series, while in the case of the latter, states are assigned according to a transition matrix specific to a particular sequence.

Also, as mentioned earlier, the use of AR models allows for the discovery of dynamic properties of the time series data. This is again not possible when using DAA. To be specific, since BP-AR-HMM uses AR models for the given time series $\mathbf{y}_t$, the interactions between pairs of variables can be expressed in their corresponding AR coefficient matrix $\mathbf{A}_k$ [23, 28]. This makes the proposed segmentation method more suitable for further interaction analysis.

2.2 Nested Pitman-Yor Language Model

NPYLM is a hierarchical language model where both "characters" and "words" are modeled using hierarchical Pitman-Yor processes [38, 42]. In each level, "characters" and "words" are modeled as n-grams (which are generated using Pitman-Yor processes), with the "character" n-gram model used in the "word" n-gram. In a broader context, the "words" are similar to high-level unit segments while "characters" are akin to low-level sub-unit segments, where these high-level unit segments are composed of low-level sub-unit segments (similar to words being formed from letters). Estimation of NPYLM parameters is done using posterior inference, where Gibbs sampling and forward-filtering-backward-sampling are mainly used [38, 42, 49].

In the following, key concepts of NPYLM would be discussed to better understand the said model.

2.2.1 Pitman-Yor Process

Pitman-Yor (PY) process is a stochastic process which generates a probability distribution G similar to a given base distribution G_0. This is denoted by

$$G \mid G_0, \theta, d \sim \mathrm{PY}(G_0, \theta, d) \tag{2.10}$$

where G_0 is a base measure, θ the concentration parameter, and d the discount parameter. This process is considered a generalization of the Dirichlet process [49].

2.2.2 Hierarchical Pitman-Yor Language Model

The hierarchical Pitman-Yor language model (HPYLM) is a hierarchical structure of Pitman-Yor-distributed n-gram models with the base measure being the immediately previous $(n-1)$-gram model. To illustrate this hierarchical structure, we start with a unigram distribution G_1^{W}. A bigram distribution G_2^{W}, denoted by $G_2^{\mathrm{W}} \sim \mathrm{PY}(G_1^{\mathrm{W}}, \theta, d)$, is generated such that this distribution will be similar G_1^{W}, especially for the high-frequency units. Likewise, a trigram distribution, denoted by $G_3^{\mathrm{W}} \sim \mathrm{PY}(G_2^{\mathrm{W}}, \theta, d)$, is generated similar to the bigram distribution G_2^{W}. Following this, the n-gram model,$G_n^{\mathrm{W}} \sim \mathrm{PY}(G_{n-1}^{\mathrm{W}}, \theta, d)$, is Pitman-Yor distributed with base measure from the $(n-1)$-gram model, and the base measure of the unigram model being G_0^{W}.

The n-gram probability of a unit $w = w_t$, given a context $h = w_{t-n} \ldots w_{t-1}$, is calculated recursively as

$$p\,(w \mid h) = \frac{c\,(w \mid h) - d \cdot t_{hw}}{\theta + c(h)} + \frac{\theta + d \cdot t_h}{\theta + c(h)} p\,(w \mid h')\,, \tag{2.11}$$

where $c\,(w \mid h)$ is a n-gram count, $h' = w_{t-n-1} \ldots w_{t-1}$ is a $(n-1)$-gram context, t_{hw} is a count under the context h' (not h),

$$t_h = \sum_w t_{hw} \tag{2.12}$$

$$c\,(h) = \sum_w c\,(w \mid h)\,, \tag{2.13}$$

and d, θ are hyperparameters. In this formulation, $p\,(w \mid h')$ can be considered as a prior probability of w.

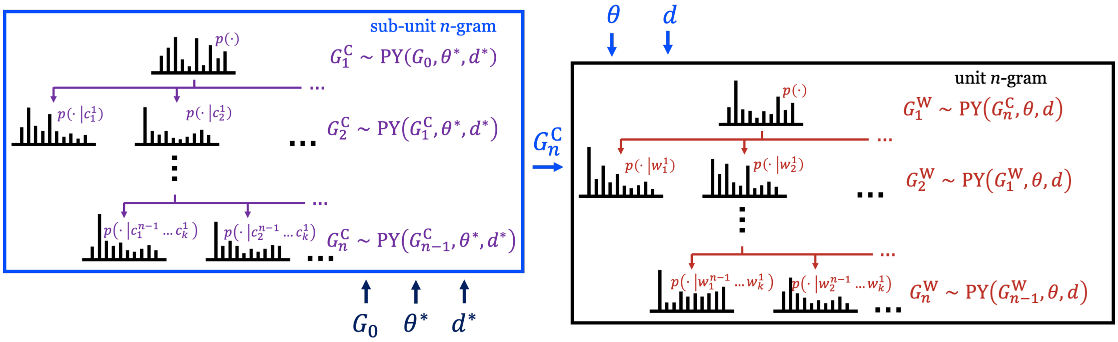

sub-unit n-gram
$p(\cdot)$
$G_1^{\mathrm{C}} \sim \mathrm{PY}(G_0, \theta^*, d^*)$
$p(\cdot \mid c_1^1)$
$p(\cdot \mid c_2^1)$
$G_2^{\mathrm{C}} \sim \mathrm{PY}(G_1^{\mathrm{C}}, \theta^*, d^*)$
$p(\cdot \mid c_1^{n-1} \dots c_k^1)$
$p(\cdot \mid c_2^{n-1} \dots c_k^1)$
$G_n^{\mathrm{C}} \sim \mathrm{PY}(G_{n-1}^{\mathrm{C}}, \theta^*, d^*)$
G_0
θ^*
d^*
G_n^{C}
θ
d
unit n-gram
$p(\cdot)$
$G_1^{\mathrm{W}} \sim \mathrm{PY}(G_n^{\mathrm{C}}, \theta, d)$
$p(\cdot \mid w_1^1)$
$p(\cdot \mid w_2^1)$
$G_2^{\mathrm{W}} \sim \mathrm{PY}(G_1^{\mathrm{W}}, \theta, d)$
$p(\cdot \mid w_1^{n-1} \dots w_k^1)$
$p(\cdot \mid w_2^{n-1} \dots w_k^1)$
$G_n^{\mathrm{W}} \sim \mathrm{PY}(G_{n-1}^{\mathrm{W}}, \theta, d)$

Figure 2.3: Visualization of NPYLM, showing how the sub-unit n-gram model is embedded in the unit n-gram model.

As the base measure G_0^{W} for the unigram model (and by extension, $p(w \mid h')$ for G_1^{W}) is not properly defined in the context of the unit n-gram model, NPYLM uses a sub-unit n-gram model G_n^{C} as the aforementioned base measure for G_1^{W}. The sub-unit n-gram model G_n^{C} uses hierarchical Pitman-Yor processes, similar to the unit n-gram model G_n^{W}. Furthermore, this sub-unit n-gram model is structured similarly to the unit n-gram model, with the probabilities for the former being calculated like the latter (that is, using Eq. 2.11 recursively). As such, an HPYLM (the sub-unit n-gram G_n^{C}) is “nested” inside another HPYLM (the unit n-gram G_n^{W}), thus explaining the “nested” part of NPYLM (Figure 2.3).

2.2.3 Posterior Inference

Posterior inference is carried out by generating samples from the posterior distribution using Gibbs sampling and forward filtering-backward sampling [38, 42, 49]. In this sampler, a unit is first removed from the current *unit* n-gram model, then a “new” unit is sampled by generating a new segmentation for the sequence of sub-units. Afterwards, the “new” unit is added to the *unit* n-gram model, thus updating the said model. This process of blocked Gibbs sampling is repeated several times, where new segmentation is generated using forward filtering-backward sampling.

2.2.4 Advantages

When using NPYLM, the input sequences are assumed to have a double articulation structure. Because it assumes a hierarchical structure for the data, NPYLM can be used to model motion sequences composed of UBs, with each UB being made up of a sequence of EBs. Using this model as the second layer of the proposed method allows us to discover high-level semantic, more meaningful actions from the low-level short, simple actions (akin to motion primitives) discovered in the first layer.

Moreover, since this is an unsupervised language model, using NPYLM in the second layer, similar to DAA, enables us to do "word" segmentation without having an existing dictionary. Lastly, blocked Gibbs sampling also significantly reduces the computational time needed to generate the samples [38, 49].

3. Testing for Significance of Interaction

In the significance testing step, the obtained AR models were used as a starting point for interaction detection. To be specific, the parameters of the AR models were used to identify which pair of time series variables had significant interactions between each other. This is similar to how several studies have used AR coefficients to define interaction [23, 28]. Aside from AR coefficients, some studies have also shown that interaction (including coupling and causality) can be detected when the time series is transformed first to the frequency domain [12, 51]. Regardless of where the interaction is detected, a common approach to test for significance in general is using the method of surrogate data [54].

3.1 Method of Surrogate Data

One important component when doing any significance testing is null-hypothesis distribution. In most cases, the test statistic in consideration has a known parametric null-hypothesis distribution. In cases where this distribution is not known, the distribution can be estimated using Monte Carlo simulation, provided that the samples from this generated distribution (or surrogates), despite being random, share most of properties from the original time series sequences [54]. The only property not shared by the original data and the surrogates is the property being tested in the null hypothesis. While originally intended for testing non-linearity in data, the method of surrogate data has been used to perform significance testing for connectivity detection [23].

The general approach when using the method of surrogate data to detect interaction is as follows. Surrogates are generated by shuffling the original time series, either in the time-domain or the frequency-domain. This is to break any existing connectivity (or interaction for this case) in the dataset. In case of multivariate time series, shuffling is done independently for each time series variable. Then, statistical tests are carried out using the distribution obtained from the surrogates. These statistical tests either use the empirical distribution of the surrogates, or some other test statistic to determine significance.

As mentioned earlier, interaction could be detected in either the time-domain or the frequency-domain. That is, surrogates can be generated by shuffling the input data in either time- or frequency-domain. Furthermore, the test statistic to be used can also be obtained from either domains. As such, we considered different combinations of where the shuffling and testing were carried out to determine the best possible set-up. For example, one case considered shuffling the time series in the time-domain, but the test statistic came from the frequency-domain.

3.1.1 Surrogate Generation

Several ways to shuffle time series data have been introduced in various research. Gilson et al. considered three ways of shuffling time series: circular permutation, random permutation, and phase randomization [23]. The first two shuffles the time series in the time-domain, while the last one shuffles the data in the frequency-domain. On the other hand, Faes et al. proposed two shuffling techniques for time series, but done in the frequency domain [12]. These shuffling techniques aim to destroy only some of the coupling, while preserving the rest. The difference between the two proposed techniques lies on whether the coupling being tested is direct causality[1] or causality[2] between a connection or a pair of variables. Regardless of the shuffling technique used, shuffling is performed independently for each time series variable.

To illustrate the shuffling techniques mentioned, consider a simple time series $\{y_t\}_{t=1:T}$. Shuffling methods presented in [23] are carried out as follows.

1. Circular permutation (CP) draws a random integer $t^* \in [1, T]$, and returns the shuffled time series $\{y_{t^*}, y_{t^*+1}, \cdots, y_T, y_1, \cdots, y_{t^*-1}\}$.

2. Random permutation (RP) considers a random permutation σ of $1, \ldots, T$, and returns the shuffled time series $\{y_{\sigma(1)}, \cdots, y_{\sigma(T)}\}$.

3. Phase randomization (PR) computes for the discrete Fourier transform $\mathfrak{F}\{y_t\}$, multiplies it by $e^{\iota\varphi_t}$ (where ι is the imaginary unit, and φ_t is randomly drawn from $[0, 2\pi)$), then takes the inverse transform to obtain the shuffled time series.

[1] *Direct causality* $y_a \rightarrow y_b$ exists if prediction of y_b based on $\{z_{ba}, y_a\}$ is better than prediction based on $\{z_{ba}\}$ only.

[2] *Causality* $y_a \Rightarrow y_b$ exists if, for at least one L, a series of L direct causality occurs, such that $y_{m_{s-1}} \rightarrow y_{m_s}$ for each $s = 1, \ldots, L$, with $m_0 = a$ and $m_L = b$.

The two shuffling method proposed by Faes, et al. involves forcing some of the AR coefficients to zero, depending on which connection is being tested. They are carried out as follows [12]:

1. If direct causality for interaction $j \rightarrow i$ is being tested, set $a_{ij} = 0$, where a_{ij} is the ij^{th} entry of the AR coefficient matrix. A reduced model is then iterated using the modified coefficient matrix, with corresponding surrogates of the time series generated using phase randomization. Here, the generated surrogates are called *Causal Fourier Transform - Direct* (CFTd).

2. If only causality for interaction $j \rightarrow i$ is of interest, then set $a_{mj} = a_{il} = 0$, where $m \neq j$, $l \neq i$. Similar to CFTd, a reduced model is also iterated using the modified coefficient matrix, with corresponding surrogates also generated using phase randomization. Here, the surrogates are then called *Causal Fourier Transform - Full* (CFTf).

3.1.2 Test Statistic

To detect interaction between time series variables, the obtained AR coefficient matrix (CM) and the computed spectral density matrix (SDM) of the Fourier-transformed time series were used as test statistic. This is because having a zero coefficient in either of these matrices indicates an absence of interaction [28, 51]. Furthermore, the empirical distribution obtained from the surrogates was used as the null-hypothesis distribution.

When testing in the time-domain, the coefficient matrix for each surrogate was estimated using Yule-Walker equation, as described in [23]. When testing in the frequency-domain, the spectral density matrix for each surrogate was computed using [6]. In both cases, the null-hypothesis distribution was made up of values corresponding to the same matrix element being tested.

3.2 Significance Testing

3.2.1 EB Level Significance

To identify switching interaction at the EB level, the resulting AR models (the AR coefficient matrix or the spectral density matrix, to be specific) were used for hypothesis testing. Time series data steps with the same EB label were concatenated first. Method of surrogate data, as described in Section 3.1, was then employed. Surrogates were generated based on the choice of shuffling technique, with the chosen test statistic estimated using available methods afterwards. Then, values of the corresponding a_{ij} (the ij^{th} entry of chosen coefficient matrix) in each surrogate made up the null-hypothesis distribution for testing the ij^{th} entry of the coefficient matrix (similar to the *local test* introduced by Gilson, et al. [23]). A coefficient entry was deemed significant if it was on the critical region of the null-hypothesis distribution, based on a specified significance level α, and a null hypothesis that no interaction exists between the time series variables.

To illustrate, let SME_k be the binary significance matrix of $\mathsf{EB}k$, with $\text{SME}_k(i,j)$ being its ij^{th} entry. Then, $\text{SME}_k(i,j) = 1$ if either $a_{ij} < \tilde{a}_{\alpha/2}$ or $a_{ij} > \tilde{a}_{1-(\alpha/2)}$. Otherwise, $\text{SME}_k(i,j) = 0$. Here, $\tilde{a}_\rho$ is the ρ^{th} percentile of the null-hypothesis distribution.

3.2.2 UB Level Significance

Significance at the UB level was determined using a time-weighted linear combination of the resulting binary significance matrix of its component EB. Combining time series models using linear combination is not new, as it has been previously used for combining forecasts [1, 62]. At the UB level, interaction exists between two time series variables if the time-weighted linear combination exceeded a specified threshold. This threshold was interpreted as the amount of time an interaction was significant. For example, a threshold of 0.90 implied that we are looking for interactions significant 90% of the time the UB under consideration was active. By detecting interactions at UB level, we avoided manually selecting which behaviors to ignore, and which behaviors to use for interaction detection.

To illustrate, consider a formed $\mathsf{UB}k$, with the corresponding component EBs $[\mathsf{EB}k_1 \cdots \mathsf{EB}k_c \cdots \mathsf{EB}k_C]$. Each $\mathsf{EB}k_c$ has a corresponding time-weight τ_{k_c}, and a binary significance matrix SME_{k_c} obtained from the EB level significance testing step. Here, the time-weight $\tau_{k_c} \in [0,1]$ represents the proportion of time steps in $\mathsf{UB}k$ which were assigned an EB label $\mathsf{EB}k_c$. The time-weighted linear combination for $\mathsf{UB}k$, SMU_k was then computed as

$$\mathrm{SMU}_k(i,j) = \sum_c \left[\tau_{k_c} \cdot \mathrm{SME}_{k_c}(i,j)\right]. \tag{3.1}$$

Finally, interaction between variables i, j was considered significant at the UB level if $\mathrm{SMU}_k(i,j) > \xi$, for some threshold $\xi \in [0,1]$.

4. Experiments

Several experiments were carried out to determine the effectiveness of the proposed approach. Two datasets were used: toy dataset made from known AR models, and motion dataset made from real human motion data. The former was primarily used to evaluate the accuracy of the proposed approach using available ground truth, while the latter was used to check the applicability of the proposed approach with real-world data.

Experiments carried out can be divided into two basic categories: segmentation experiments and interaction detection experiments. Each experiment category utilized both datasets, with the toy dataset used for accuracy evaluation while the motion dataset for real motion data applicability.

4.1 Datasets

4.1.1 Toy Dataset

Time series sequences with known AR lag order were generated to evaluate the segmentation and interaction detection accuracy. Specifically, three subdatasets, $\mathbf{L}q$, $q = 1, 2, 3$, were generated from switching q^{th} lag order AR models with hierarchical structure. Each subdataset $\mathbf{L}q$ had four time series sequences of four dimensions each. The time series $\mathbf{y}_t^{(q,i)} \in \mathbb{R}^4$, $i = 1 : 4$, were formed by concatenating UBs randomly chosen from a library of four UBs (based on predefined transition probability matrices), with the collection of UBs different for each subdataset. Regardless of the subdataset, each $\mathsf{UB}k$ used in Set $\mathbf{L}q$ was composed of a maximum of three EBs, where each EB was an AR(q), an AR model of AR lag order q, with sparse AR(q) coefficient matrices. EBs from the same UB shared similar sparsity patterns for their respective AR coefficient matrices. For example, $\mathsf{UB1}$ could be composed of $\mathsf{EB2}$ and $\mathsf{EB4}$, with both having block diagonal matrices for their respective AR coefficient matrices.

On average, 7.50, 5.63, and 4.00 of the 16 coefficient entries in each lag were *significant* (or non-zero entries) for each subdataset, respectively. Furthermore, elements of the AR coefficient matrices were set within the range $(-1, 1)$, where the significant entries had an average absolute value of 0.3738, 0.2989, and 0.2373 for each subdataset, respectively.

Large-scale Toy Data Aside from the three aforementioned subdatasets, another set of three additional subdatasets, $\mathbf{T}s, s = 10, 20, 100$, were also generated exclusively for segmentation experiments only. These sequences were used to explore how the proposed method would perform on a larger-scale simulation setting. These additional subdatasets were generated from switching AR(1) models, using the same parameter settings as described above but with s time series sequences, ($s = 10, 20, 100$), instead of the previous four sequences only.

4.1.2 Motion Dataset

A second dataset was generated to determine the applicability of the proposed segmentation method with real motion data. The motion dataset had four time series sequences, each having 16 dimensions. The time series $\mathbf{y}_t^{(i)} \in \mathbb{R}^{16}$, $i = 1 : 4$, were generated by concatenating UBs randomly chosen from a library of six UBs. The six UBs were as follows:

UB1	:	walk towards each other then shake hands,
UB2	:	linked arms while walking,
UB3	:	synchronized walking,
UB4	:	alternating squats,
UB5	:	alternating jumping jacks,
UB6	:	synchronized jumping jacks.

These corresponded to the motion capture sequences of the actions of Subjects 18 to 23 in CMU Graphics Lab - Motion Capture Library [8]. Each of the 16 dimensions represents a joint angle measurement, where the first eight measurements were for the first person, while the next eight were for the second person. The eight selected joint angle measurements for each person were the same, and were similar to the measurements chosen in [16]. They were the following:

right and left (R/L) shoulders, R/L elbows,
R/L knee, R/L ankles.

These angles were also normalized, and block-averaged with window size 12 time steps.

Some things to note for this dataset. As the mocap sequences represented UBs, true EBs for this dataset were not known. Thus, the true AR models for this dataset were also unknown. Consequently, true AR lag orders and true AR coefficient matrices were also unavailable. Lastly, true significant interactions were also not known for this dataset. This is generally the case for any real world data, as they are not annotated for interaction, and ground truth interactions are often subjective [10]. Despite this difficulty, it was still interesting what interactions the proposed method could detect.

4.2 Segmentation

Several experimental set-ups for segmentation were carried out for both datasets. As the true AR lag order was generally unknown beforehand, we performed the proposed segmentation method to each (sub)dataset using various AR lag order. Moreover, results from both sets were compared to those from DAA.

4.2.1 Toy Data Evaluation

The proposed segmentation method with AR orders $r = 1 : 3$ was applied to each subdataset in the toy dataset. The result of the proposed method with AR(r) applied to the dataset $\mathbf{L}q$ was denoted by $\mathbf{L}q$-$\mathbf{AR}r$. The proposed segmentation method was carried out thirty times, where each run had ten iterations[3] each. In each iteration, the UB labels obtained from the prior iteration were used as the initial EB labels of the next one.

The following values were used for the parameters of BP-AR-HMM: the concentration parameter $c = 3$, the mass parameter $\alpha = 2$, both with Gamma$(1, 1)$ prior, for the beta process; the transition parameter $\gamma = 1$, the transition sticky parameter $\kappa = 25$, with Gamma$(1, 1)$ and Gamma$(100, 1)$ prior, respectively, for the transition matrix. The first 5,000 samples of the MCMC algorithm were discarded and considered as burn-in, while the next 5,000 samples were used for the analysis. The resulting state sequences for the first layer were summarized separately for each run, where states with associated time shorter than 1% of the total time were discarded. These summarized sequences were then used as input for NPYLM. The parameters for NPYLM were set as follows: the discount parameter $d = 0.5$ with Beta$(1.5, 1)$ prior; the concentration parameter $\theta = 0.1$

[3] Instead of having a terminating condition for the iteration step, the number of iterations was fixed because of computational constraints.

with Gamma$(10, 0.1)$ prior. Similar to BP-AR-HMM, the first 5,000 samples of the blocked Gibbs sampling were discarded as burn-in, while the next 5,000 samples were used for analysis. The hyperparameters for both layers were chosen similar to that of [18] (for BP-AR-HMM) and [42] (for NPYLM). Posterior inference for BP-AR-HMM was carried out using the codes in [32], while the codes in [41] were used for the posterior inference of NPYLM.

Large-scale Toy Data Unlike the the first three subdatasets, the experiments were limited to only AR(1) for the additional subdatasets. That is, the proposed method with AR(1) was applied to subdataset **T**s ($s = 10, 20, 100$), using the same settings as described above, but with ten, ten, and three runs for **T10**, **T20**, and **T100**, respectively. Each run still had ten different chains of sampling. Similar to the results of the other toy subdatasets, results were denoted by **T**s**-AR1**.

4.2.2 Real Motion Data Applicability

The applicability of the proposed segmentation method to real motion data was evaluated by applying the proposed method with AR orders $r = 1 : 3$ to the motion dataset. The result of the proposed method with AR(r) applied to this dataset was denoted by **CMU-AR**r. The parameters in the proposed method were set in the same way as the previous subsection, with the exception of the transition sticky parameter, which was set to $\kappa = 200$.

4.2.3 Comparison with DAA

The proposed segmentation method was compared with DAA using both datasets. The state sequences were again summarized separately for each run in the two datasets. The parameters of DAA were chosen such that they were comparable to the parameter values of the proposed method. Since sticky HDP-HMM is usually used when there is only one time series sequences, the sequences in each (sub)dataset were concatenated first into one long time series, before being used as input for the first step of DAA. The resulting EB labels were then split back accordingly, before being summarized. For the toy dataset, states with associated time shorter than 1% of the total time were discarded, while no states were discarded for the CMU dataset, as the states switched frequently in the first step. The results were denoted by **Lq-DAA** and **CMU-DAA** for the toy and CMU datasets, respectively. DAA was performed using the codes recommended in [47].

4.3 Interaction Detection

Interaction detection experiments were also carried out for both datasets using several experimental set-ups. However, only results from **L1-AR1** and **CMU-AR1** were used in all the experimental set-ups of the second step. Results from experimental set-ups with higher AR lag order were not utilized.

Similar to the segmentation step, the toy dataset was used to evaluate the detection accuracy, while the motion dataset was used to check for real motion data applicability. Specifically, EB level significance testing was performed for both datasets, while UB level significance testing was only carried out for the motion dataset. UB level significance testing was not carried out for the toy dataset since the ground truth interaction was not available for this level.

4.3.1 EB Level Significance

To identify significant interactions, eight different test for significance methods were considered. For each of the test methods considered, 500 surrogates were generated, and a significance level of $\alpha = 0.01$ was used. That is, the ij^{th} matrix entry was considered significant if it was less than the 0.5^{th} percentile or greater than the 99.5^{th} percentile of the null-hypothesis distribution. For evaluation purposes, note that interaction between a pair (r, s) in the original coefficient matrix was considered significant if it had a non-zero value. Table 4.1 lists down these eight test methods, including in which domain shuffling and testing occurred.

Results for each test method were then compared using the following evaluation measures. In the following, TP, FP, TN, and FN refers to the number of true positives, false positives, true negatives, and false negatives, respectively. Each of the eight test methods was performed 50 times to have better comparison results.

1. False Positive Ratio (FPR) $= \dfrac{\text{FP}}{\text{FP} + \text{TN}}$

   This measures the accuracy of undetected insignificant entries with respect to the actual insignificant entries, with a lower value being desired.

2. Sensitivity $= \dfrac{\text{TP}}{\text{TP} + \text{FN}}$

   Also called *recall*, this measures the accuracy of detected significant entries with respect to the actual significant entries, with a higher value being desired.

3. Specificity $= \dfrac{\text{TP}}{\text{FP} + \text{TN}}$

   This measures the accuracy of detected insignificant entries with respect to the actual insignificant entries, with a higher value being desired.

4. Positive Predictive Value (PPV) $= \dfrac{\text{TP}}{\text{TP} + \text{FP}}$

   Also called *precision*, this measures the accuracy of detected significant entries with respect to the test results, with a higher value being desired.

5. F_1 score $= \dfrac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$

   This is basically the harmonic mean of precision (PPV) and recall (sensitivity), with a higher value being desired.

6. Accuracy $= \dfrac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$

7. Run time (in seconds)

Based on the metric comparisons, a subset of these eight test methods was selected, and applied to the motion dataset to identify significant interactions between the joint angle measurements. Test for significance of interaction for the results of the motion dataset were carried out similar to the toy dataset. For each of the test method chosen, 500 surrogates were generated, and a significance level of $\alpha = 0.01$ was used. This was repeated 50 times, and was performed for ten different segmentation results.

4.3.2 UB Level Significance

To determine significance of interaction at the UB level, time-weighted linear combinations SMU_k for each formed UB in each run were computed. For each component EB in a given formed UB, time weights τ_{k_c} were first determined by computing the fraction of total time each component EB appeared in the given UB. Then, the binary significance matrix SME_{k_c} for each component EB were multiplied to their corresponding time weights, before being added together. Interaction between variables i, j was significant at the UB level if the average of these time-weighted combinations exceeded 0.90. This process was repeated for all formed UBs in a given run.

Similar to the EB level significance testing, UB level significance testing was carried out independently for the ten different segmentation runs of EB level significance testing performed for the motion dataset.

Table 4.1: Different combinations of shuffling and testing methods that were used and compared to. Note that all were applied to the toy dataset, but only a subset was applied to the motion dataset.

Test Method	Shuffling		Testing	
	Domain	Method	Domain	Statistic
CP-CM	Time	Circular Permutation	Time	Coefficient Matrix
RP-CM	Time	Random Permutation	Time	Coefficient Matrix
CP-SDM	Time	Circular Permutation	Frequency	Spectral Density Matrix
RP-SDM	Time	Random Permutation	Frequency	Spectral Density Matrix
PR-CM	Frequency	Phase Randomization	Time	Coefficient Matrix
PR-SDM	Frequency	Phase Randomization	Frequency	Spectral Density Matrix
CFTf-SDM	Frequency	Causal FT (Full)	Frequency	Spectral Density Matrix
PR-SDM	Frequency	Causal FT (Direct)	Frequency	Spectral Density Matrix

5. Results

The proposed segmentation approach to switching interaction detection yielded positive results. First, the proposed segmentation method was successful in extracting both EBs and UBs (Figure 5.1). Despite model mismatch, segmentation results showed decent estimation accuracy. We have also demonstrated its effectiveness with real motion data, as well as its superiority to DAA. Second, interaction detection was also relatively successful, as shown by the computed specificity on the experiment for the toy dataset.

5.1 Toy Data Evaluation

5.1.1 Segmentation

Figure 5.2 shows the confusion matrices for the EB labels (left) and UB labels (right) of an example segmentation result for the toy dataset. Confusion matrices are usually used to show the correspondence between true and estimated labels. The values shown in this figure indicate the correspondence between true labels and estimated labels, with the values normalized per column to allow for one-to-many correspondence between true and estimated labels. Specifically, values close to 1 indicate that the correspondence between true and estimated labels have high specificity, while rows with several entries close to 1 indicate that more than one estimated label correspond to one true label.

Aside from the correspondence between labels, evaluation metrics were also computed. Estimation accuracy was primarily measured in terms of Hamming distance (HDist), which is the number of time steps where the estimated label was different from the true label, summed for all time series sequences. Values were normalized by dividing the computed HDist by the total number of time steps of the given sequence. For this metric, the lower the HDist, the better. For the toy data, the smallest average normalized EB HDist were obtained when the true AR lag order was used. That is, **L1-AR1** had smaller EB HDist than those of **L1-AR2** and **L1-AR3**, as seen in Figure 5.3. Similarly, EB HDist of **L2-AR2** and **L3-AR3** were also the smallest in their respective subdatasets. It was also observed that the second layer of the segmentation step reduced the error observed in the first layer (red lines in Figure 5.3). Except for the results of **L1-AR1**, UB

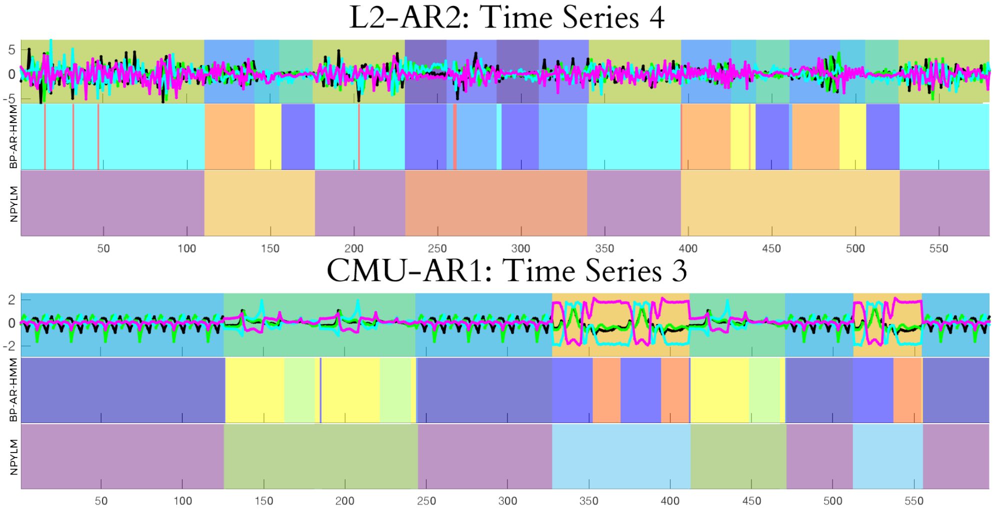


Figure 5.1: Example of segmentation results, with true EB (top layer), estimated EB (middle layer), and estimated UB (bottom layer) for each time series. Plots of the first four dimensions of the toy (top figure) and motion datasets (bottom figure) are also shown in the top layer.

HDist was generally smaller than EB HDist. A possible explanation for this is that the errors caused by EB labels contained in the same UB label were no longer found after performing the second layer. However, in cases where the formed UBs were undersegmented (right figures in Figure 5.4), the resulting UB HDist were generally overstated. This undersegmentation was the primary reason why UB HDist results of **L1-AR1** were higher than their corresponding EB HDist. The same observations were also seen in their respective adjusted Rand index (ARI) (right figures in Figure 5.3). For this metric, the higher ARI, the better.

The joint log probabilities (LogPr), $P\left(\mathbf{y}, \mathbf{F}, \mathbf{z}\right)$, at the EB level were also examined. Similar to EB HDist, the joint LogPr values were optimal when the true AR lag order was used (Figure 5.5). Based from these observations, it seemed like true AR lag order can be determined by finding the AR lag order r^* such that

$$r^* = \arg\min_r \left(\text{EB HDist}\right) = \arg\max_r \left(\text{EB LogPr}\right), \tag{5.1}$$

where $\arg\min_r(f)$ returns the AR lag order where f is minimum (or maximum for $\arg\max_r(\cdot)$). Since the joint LogPr values are available even in the absence of ground truth, these values can then be a potential criterion for selecting the model

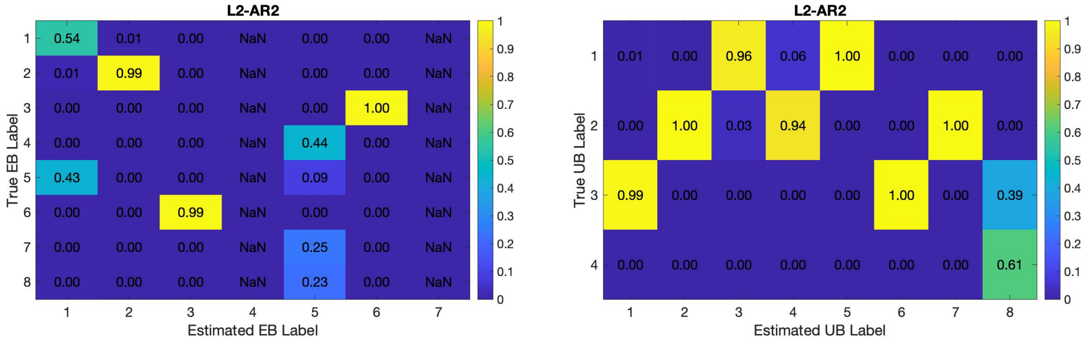


Figure 5.2: Confusion matrices for an example of segmentation results of the toy dataset, showing both EB (left) and UB (right) label correspondence. Numbers normalized per column. `NaN` columns indicate that the estimated label was no longer present in the summarized sequence.

with cross-validation. However, in cases where a single r^* does not exist, a new criterion would be needed to determine the true AR lag order.

On another note, it was also observed that the number of EBs discovered, as well as the segmentation itself, for each of the thirty run varied (Figure 5.4). Despite this, the resulting UB segmentation were still more or less similar (Figure 5.6). The proposed proposed segmentation method was able to identify the same UBs despite discovering different EBs. We could then say that while the the EBs discovered depended on the run, the formed UBs were not.

When compared to the results of DAA, the proposed segmentation method generally had better normalized HDist than those from DAA. The computed HDist for **L**q-DAA were always somewhere in the middle: never the largest, but never also the smallest (Figure 5.3). For example, **L3-DAA** had the worst EB HDist and UB HDist, while EB HDist and UB HDist for **L2-DAA** were better than those for **L2-AR3** but worse than those for **L2-AR2** and **L2-AR1**. Similar to earlier observations, the obtained segments from the second layer of DAA were also relatively similar across the different runs. This was despite the frequent switching at the first layer. For **L1-DAA** and **L3-DAA**, several formed UBs were undersegmented, while **L2-DAA** had cases of oversegmented UBs. In fact, six and ten of the thirty runs for **L1** and **L3**, respectively, only had one EB and one UB, thus failing to segment the time series sequences. This failure in segmentation was generally not seen in any of the other runs in **L**q**-AR**r, with **L1-AR1** the only

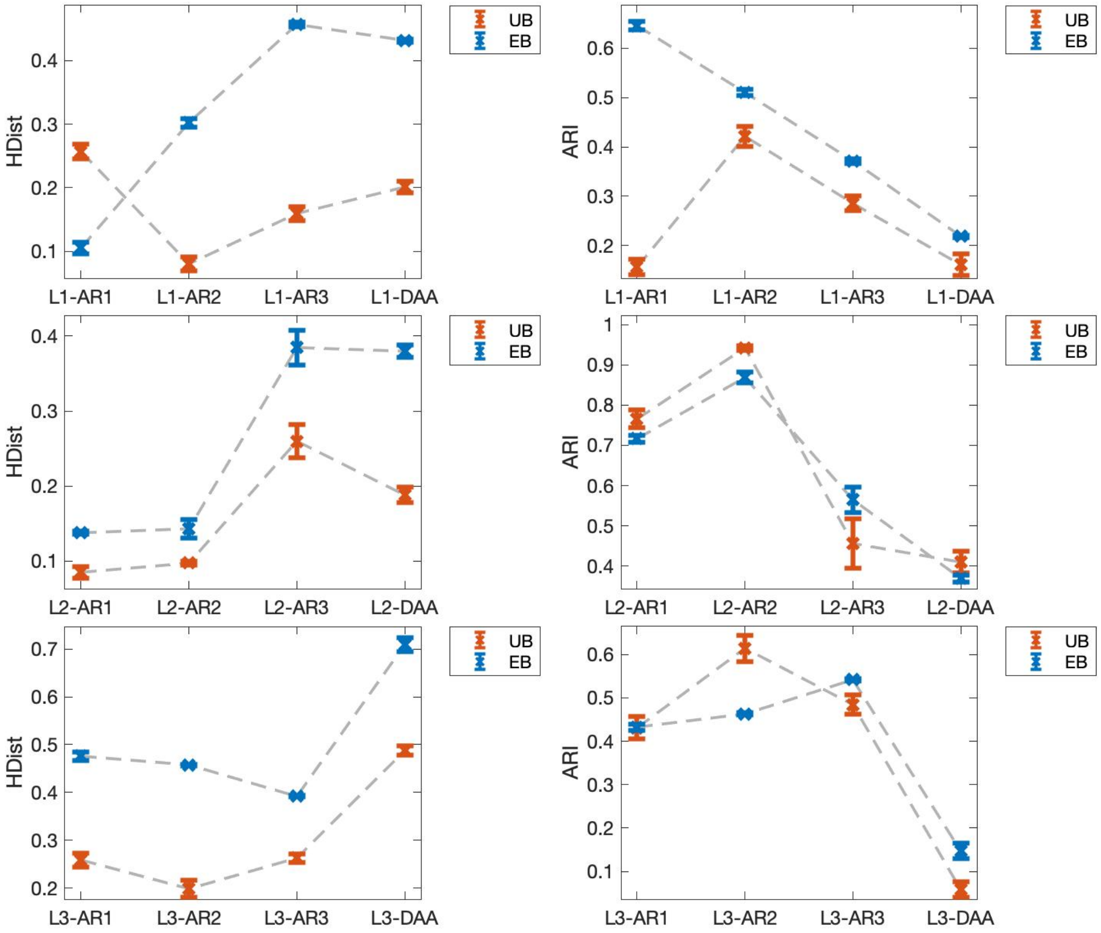


Figure 5.3: The average normalized HDist (left) and adjusted Rand index (right) over all runs for each experiment in the toy dataset. Bars indicate one standard error. Lower HDist and higher ARI are desired.

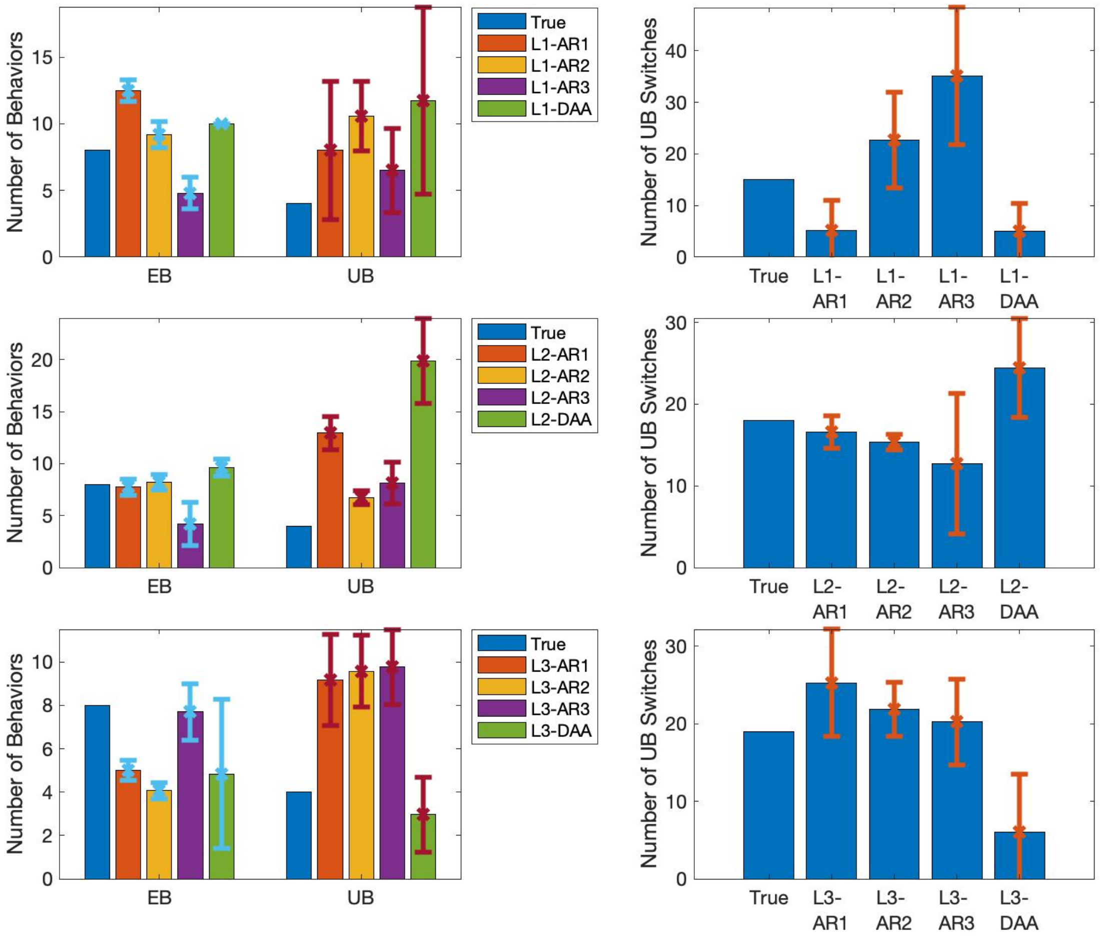


Figure 5.4: Average number of discovered EBs and UBs (left) and average number of UB switches (right) for each experiment in the toy dataset. As the number of switches are less than the true value for experiments using the toy set, resulting UBs are generally undersegmented. Bars indicate one standard deviation.

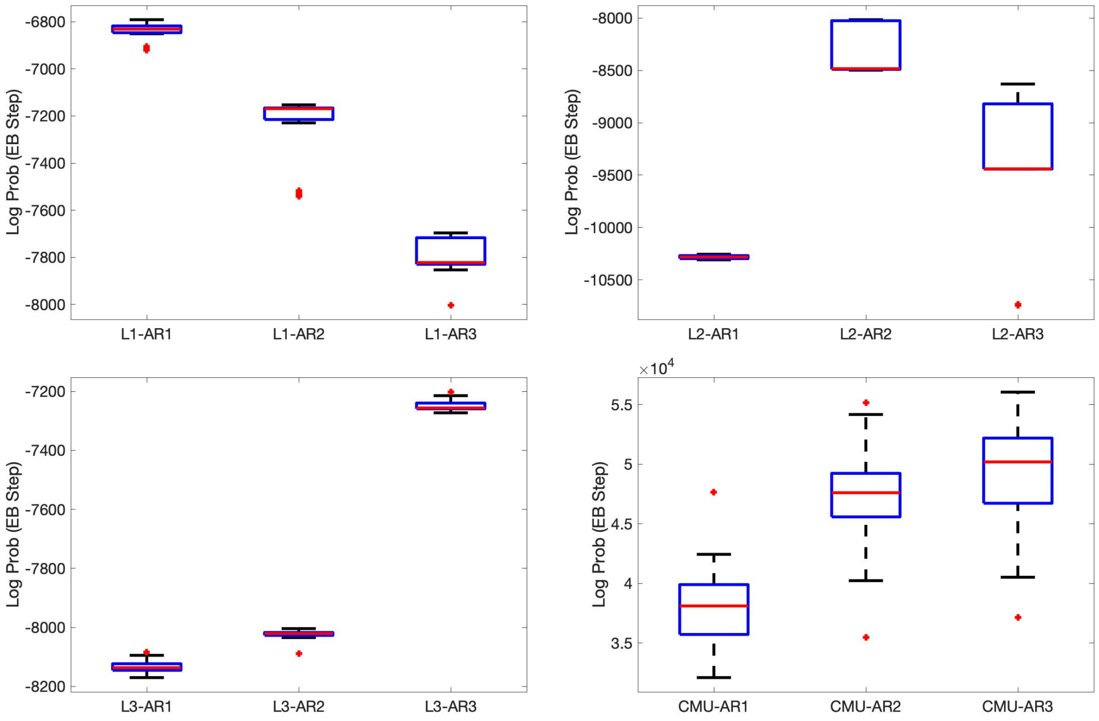


Figure 5.5: Boxplots of the joint log probabilities $P(\mathbf{y}, \mathbf{F}, \mathbf{z})$ of the EB step for the toy and motion datasets. Red lines indicate median, edges of blue box are the first and third quartiles, while the whiskers are most extreme data points except outliers.

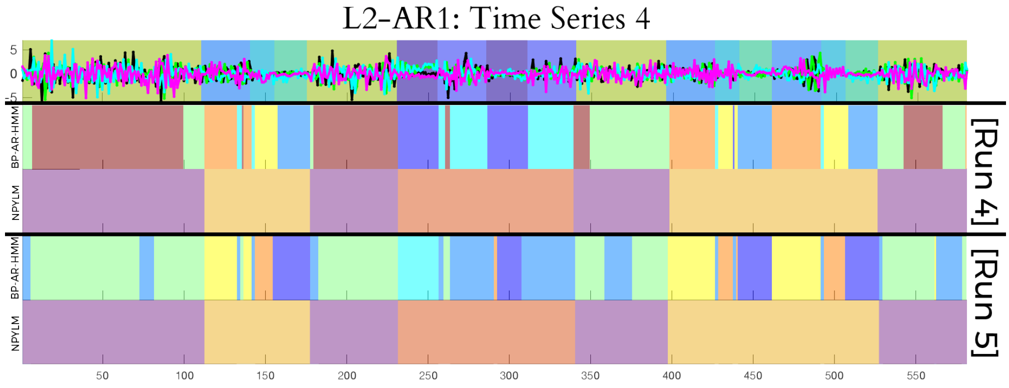


Figure 5.6: Example of segmentation result for Time Series 4 using the proposed segmentation method, obtained from two different runs of **L2-AR1**.

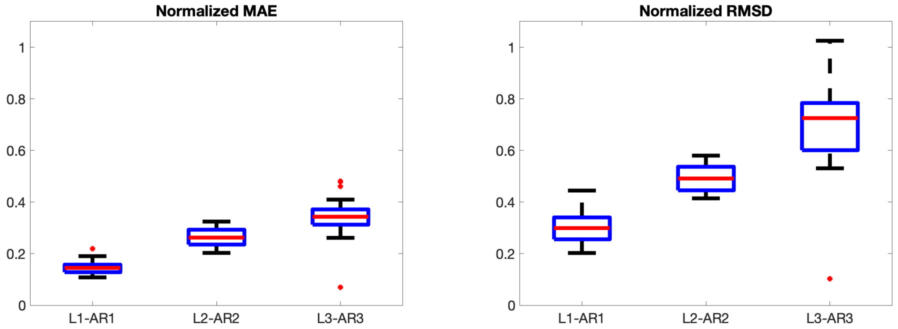


Figure 5.7: The average normalized AR coefficient estimation errors, measured using mean absolute error and root mean squared error, both normalized using the range of coefficient values. Estimation errors were computed only for **L**q**-AR**r where $q = r$.

exemption.

Lastly, the proposed segmentation method had decent AR coefficient estimation accuracy (Figure 5.7), with **L1-AR1** having the smallest normalized errors. For the said set, the average normalized mean absolute error (NMAE) was around 15% of the range of coefficient values, while the average normalized root mean squared error (NRMSE) was around 30% of the range of values. As having higher true AR lag orders naturally yielded more complex models and thus have higher estimation errors, the earlier observation was not that surprising.

Large-scale Toy Dataset Based on the results of **T**s**-AR1**, the proposed segmentation method seem to have more discovered EBs and formed UB labels when used on large datasets. There were 13.90, 17.70, and 21.33 discovered EB labels and 21.00, 19.70, and 39.33 formed UBs for **T10-AR1**, **T20-AR1** and **T100-AR1**, respectively. The increased number of discovered labels led to more discovered labels corresponding to the same true label. Taking these multiple correspondences into account when computing for HDist, the resulting (adjusted) EB HDist are 0.6112, 0.6759 and 0.7086 for **T10-AR1**, **T20-AR1** and **T100-AR1**, respectively, while the corresponding UB HDist are 0.1096, 0.1633 and 0.1864, respectively. Note, however, that the resulting UB HDist were still smaller than the EB HDist, which is consistent with previous observations. That is, correct segmentation in UB level can be reproduced, as long as the observed errors in the EB level were within the same UB label. These results then suggest that the proposed method can discover correct high-level segmentation, despite errors in the low-level segmentation and regardless of the number of time series sequences considered.

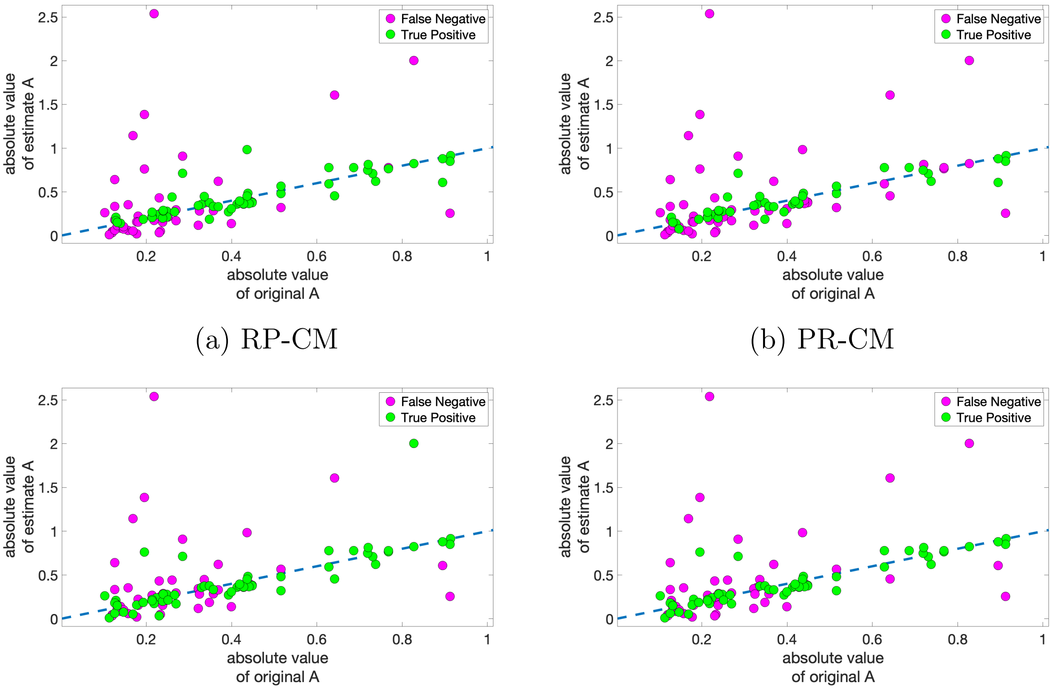
5.1.2 Significance Testing

The resulting AR models from the run with the lowest EB HDist from **L1-AR1** were used in the second step. Table 5.1 summarizes the evaluation measures computed from each test method. The values were averaged first over the fifty trials, before being averaged over all AR models discovered. Based on these values, using time-domain statistic (i.e, the coefficient matrix) led to better FPR, specificity, and accuracy values. Moreover, run times were faster for this configuration. Using frequency-domain statistic (i.e., spectral density matrix) resulted to better sensitivity values. On the other hand, shuffling in the frequency domain had better PPV values while shuffling in the time domain had better F_1 score. Considering all these, doing interaction detection using the RP-CM seemed to be the best configuration. While not all of its metric values were the best, all of them were in the top three. Thus, it was the best compromise between accuracy and run time.

Aside from the eight evaluation metrics, we were also interested to see how interaction detection fared vis-a-vis the coefficient matrix. First, it was observed that test methods with time-domain statistic had an issue with identifying small values of the coefficient matrix as significant. This can be seen in Figure 5.8, where small values of the coefficient matrix were mostly found to be not significant. Furthermore, the histograms of the resulting significant and non-significant entries indicated that they have zero and non-zero means, respectively (Figure 5.9). As such, while using time-domain statistic (such as RP-CM) yielded generally better results, there is a possibility that low-valued interactions would not be detected should this test method be used. This observation was true for the other trials of the said test method. On the other hand, unlike RP-CM and PR-CM, RP-SDM and CFTd-SDM seemed to be capable of finding small values to be significant. That is, using frequency-domain statistic could better detect small-valued interactions. This observation was also true for other trials of the same configuration. Figure 5.9 indicates that the histograms for both the significant and non-significant entries seemed to be similar, as the two histograms coincided with each other.

Table 5.1: Evaluation measures computed from the eight test methods used. Top three values per metric are in boldface, while best values are in blue.

Test Method	FPR	Sensitivity	Specificity	PPV	F_1 Score	Accuracy	Run Time
CP-CM	**0.0957**	0.4973	**0.9043**	0.7483	0.5970	**0.6928**	**1.42**
RP-CM	**0.0570**	**0.5361**	**0.9430**	**0.8428**	**0.6706**	**0.7415**	**1.54**
CP-SDM	0.1820	0.4978	0.8180	0.6942	0.6060	0.6403	298.59
RP-SDM	0.1897	**0.5931**	0.8103	0.7545	**0.6819**	0.6886	298.37
PR-CM	0.1550	0.5094	0.8450	**0.8487**	**0.6717**	0.6741	3242.29
PR-SDM	**0.0574**	0.4138	**0.9426**	0.8368	0.5550	**0.6976**	**2.11**
CFTf-SDM	0.2004	0.4238	0.7996	0.7434	0.4926	0.6322	$3,282.82$
CFTd-SDM	0.1841	**0.5790**	0.8159	**0.8417**	0.6411	0.6876	$4,231.10$



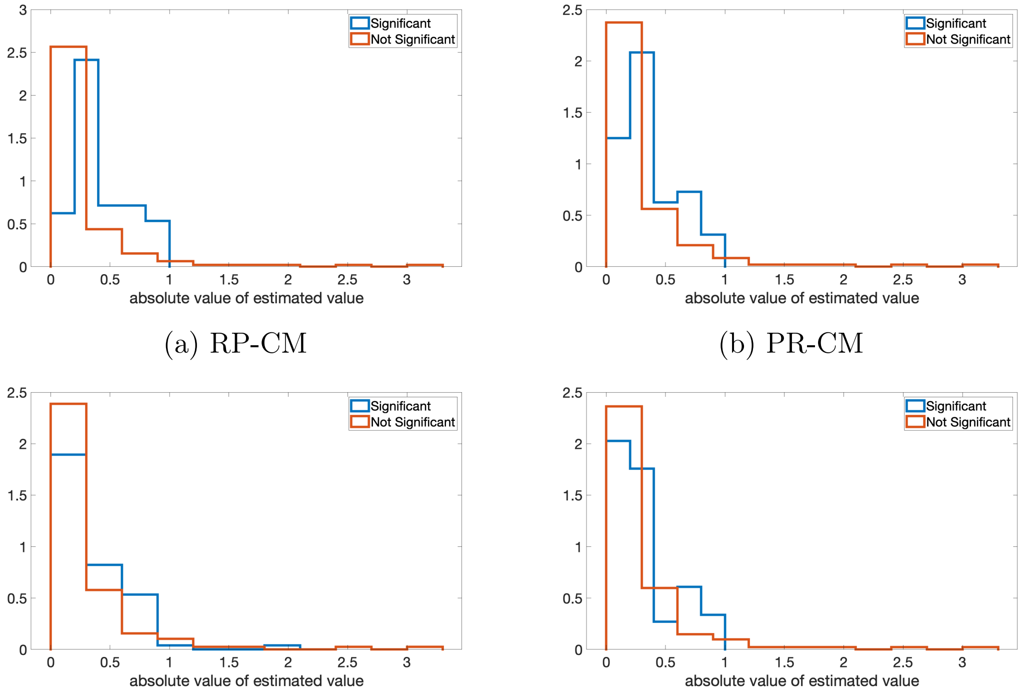
(a) RP-CM

(b) PR-CM

(c) RP-SDM

(d) CFTd-SDM

Figure 5.8: The four figures compare the actual (absolute) value of the coefficient with the estimated (absolute) values of all the ‘true’ significant interactions. True positives and false negatives are labeled accordingly. Results displayed are for one trial in the test method used.



(a) RP-CM

(b) PR-CM

(c) RP-SDM

(d) CFTd-SDM

Figure 5.9: The four figures display the histogram of the significant and non-significant entries based on the results of the test method used. Results displayed are for one trial in the test method used.

5.2 Real Motion Data Applicability

5.2.1 Segmentation

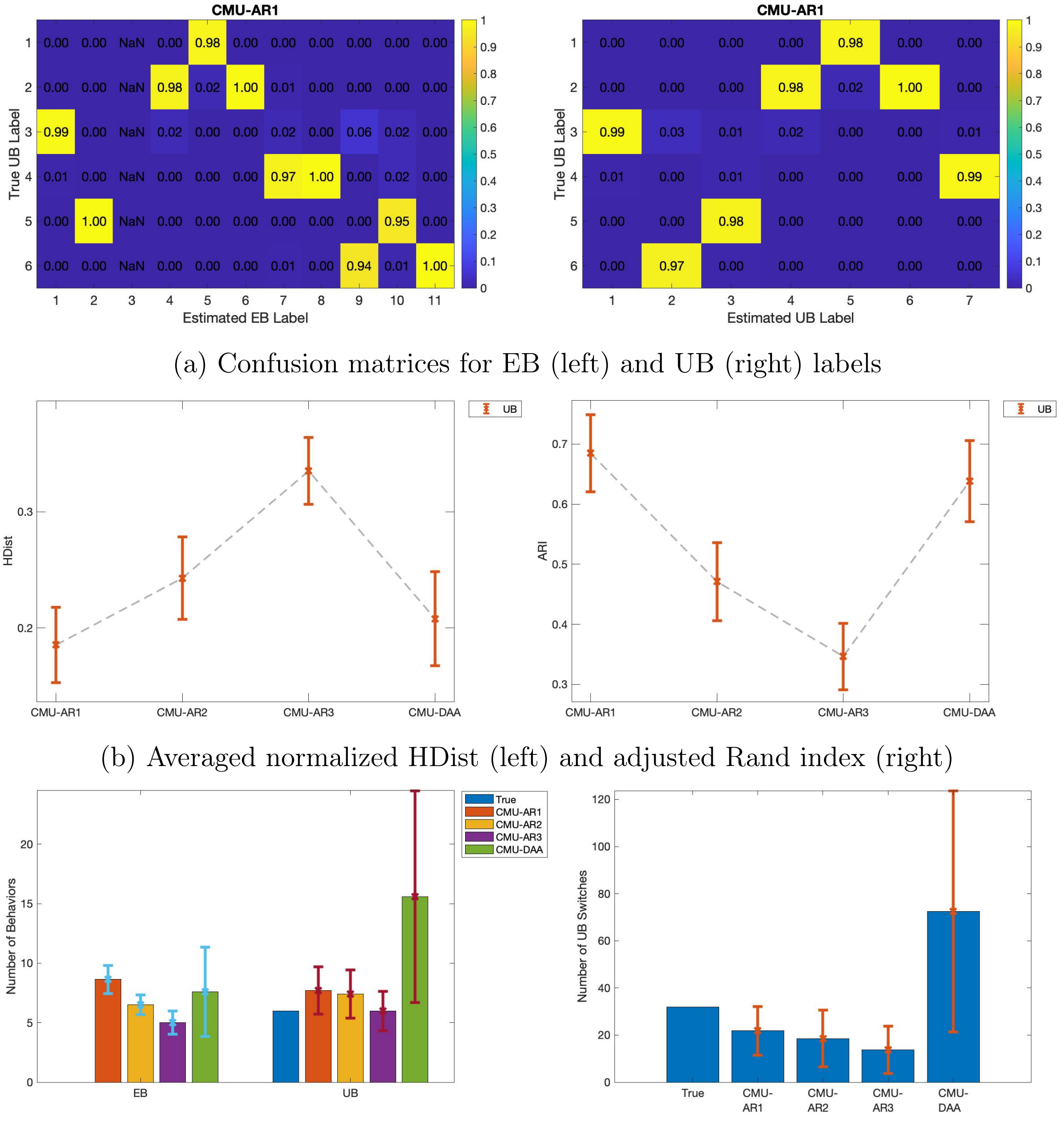
For the motion dataset, **CMU-AR1** had the smallest average normalized UB HDist when compared with **CMU-AR2** and **CMU-AR3** (Figure 5.10b). However, **CMU-AR2** and **CMU-AR3** had larger LogPr values than **CMU-AR1** (Figure 5.5). Because r^* (from Eq. 5.1) does not exist, the optimal AR lag order could not be determined. Despite this, the obtained segmentation were still fairly accurate and acceptable, regardless of AR order used.

Similar to earlier observations, the number of discovered EBs and the EB labels still varied for each run for this dataset. Yet, the resulting segmentation at the UB step were still quite similar. For example, consider the resulting segmentation in Figure 5.11. Here, the UB [1 8 1] was UB5 : alternating jumping jacks motion. In another segmentation, the same UB label corresponded to [D E], with the component EBs referring to completely different things. Despite the difference in component EBs, both [1 8 1] and [D E] referred to the same true UB (in this case, UB5). Thus, the proposed segmentation method still identified the same semantic behaviors despite discovering different motion primitives in each run.

Comparing the proposed method with that of DAA, the proposed method again had better results than DAA. **CMU-AR1** had slightly smaller UB HDist (0.1815) compared to that of **CMU-DAA** (0.2080), with the resulting segmentation from the second step still comparably similar across the different runs. Moreover, **CMU-AR1** also had higher UB ARI (0.6847) compared to **CMU-DAA** (0.6384). Unlike the the obtained results from the proposed method, the formed UBs in the CMU dataset using DAA were mostly oversegmented. As such, adjustments were made to match the resulting labels with the ground truth (Figure 5.10). Similar to earlier observations when DAA is applied to the toy set, several runs (seven of thirty to be exact) in **CMU-DAA** failed to segment the time series sequences. In comparison, segmentation failure was observed in only two, two, and four of the thirty runs in **CMU-AR1**, **CMU-AR2**, and **CMU-AR3**, respectively.

5.2.2 Interaction Detection

The obtained AR models from the previous step were used in the significance testing step, using RP-CM. Figures 5.12 and 5.13 show an example interaction



(a) Confusion matrices for EB (left) and UB (right) labels

(b) Averaged normalized HDist (left) and adjusted Rand index (right)

(c) Average number of discovered behaviors (left) and UB switches (right)

Figure 5.10: Evaluation metrics used in the toy dataset were also computed similarly for the motion dataset. Bars indicate one standard error.

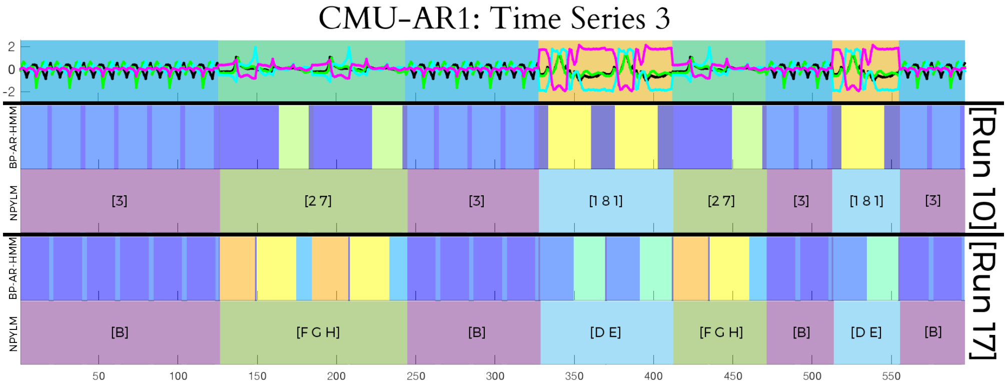
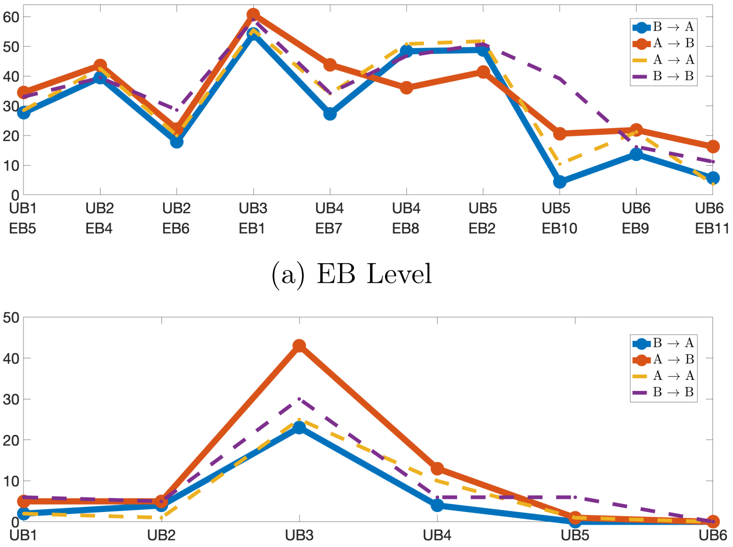


Figure 5.11: Example of segmentation result for Time Series 3 using the proposed segmentation method, obtained from two different runs of **CMU-AR1**.

detection result from one of the segmentation results used. In Figure 5.12, the number of significant entries for each elemental behavior discovered were displayed, with the numbers grouped depending on whose joint angle measurements were interacting. Using the significance matrix as reference, the block matrices on the diagonals indicate the interaction between the joint angles of the same agent, while the off-diagonal block matrices represent the interaction between the joint angle of one agent and the joint angle of the other.

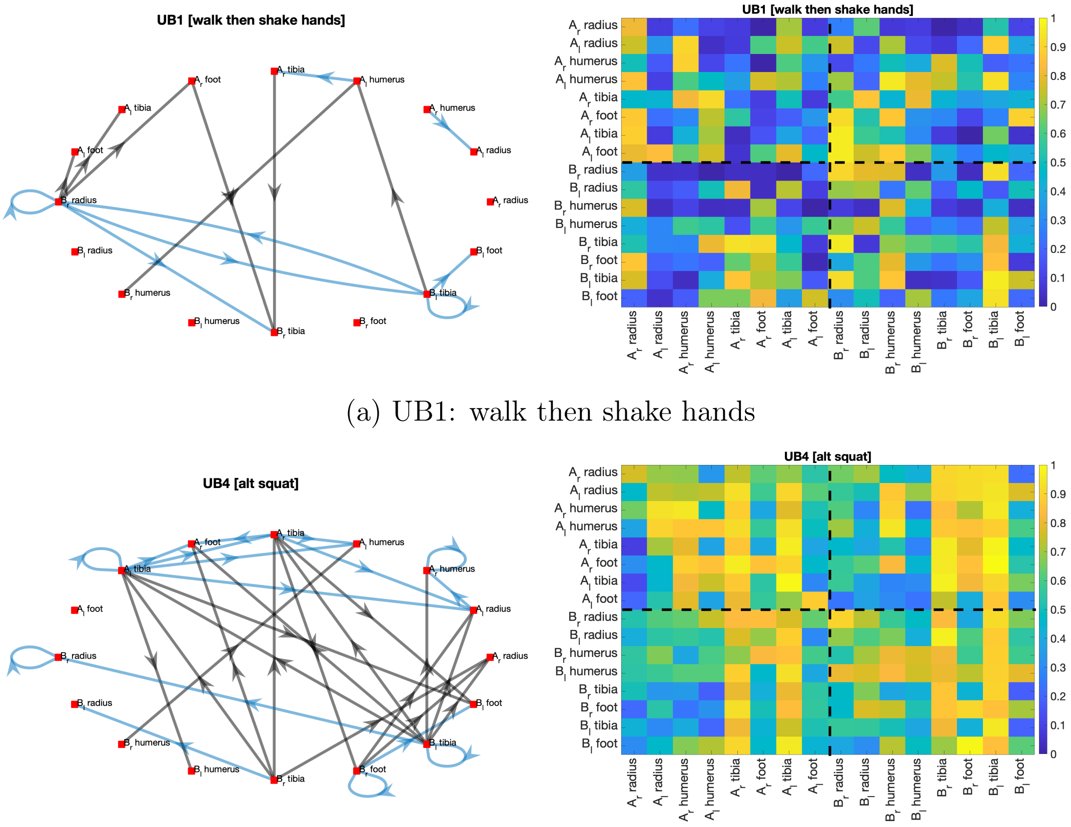
As the number of discovered EB labels varied per run, so did the results from the interaction detection at the EB level. What was common to all of them, though, was the high number of coefficient entries deemed significant. Furthermore, it was also observed that the component EBs of UB3 always had a higher number of significant entries when compared to the other EBs. For reference, this UB referred to the synchronized walking movement. At the UB level, however, fewer interaction pairs were deemed significant at a threshold of 0.90. Similar to the results at the EB level, results at the UB level varied as well. What was common, though, was the results for UB4 : alternating squats. For this UB, most of the joint angles of both agents depended on knee angle of both agents (Figure 5.13). This then suggests that performing alternating squats depended on how the knee angle moves.



(a) EB Level

(b) UB Level

Figure 5.12: The number of significant entries discovered for one of the segmentation results used. Solid lines represent interactions of joint angle measurements from different agents, while dashed lines represent interactions of joint angle measurements from the same agent.



(a) UB1: walk then shake hands

(b) UB4: alternating squats

Figure 5.13: Example of resulting connectivity plots for the significance matrix of UB1 and UB4. A directed edge $A \to B$ means that B depends on the lag values of A. Blue lines in the connectivity plots represent interaction between joint angles of the same agent, while black lines represent interaction between the joint angles of different agents.

6. Discussion

6.1 Segmentation

Based on the results using toy and motion datasets, the proposed segmentation method could segment multiple related time series, even if the true AR lag order was not used. However, accuracy would be compromised if there was a mismatch between AR lag order used and the true AR lag order. Nonetheless, not using the true AR order still gave an acceptable segmentation. To be specific, using AR(1), regardless of the true AR order, was worth considering. This gave a simpler model, but one where some accuracy in the first layer was given up. Fortunately, some of these errors were corrected in the second layer. Furthermore, it was also shown that correct segmentation could be obtained, regardless of the number of time series sequences in the dataset.

On another note, despite having different results at the EB step in each run, the proposed segmentation method had similar results at the UB step. The proposed method properly and consistently identified the high-level segments, despite discovering different low-level segments. As such, we could say that the high-level segments were actually invariant to the low-level segments.

6.1.1 Comparison with DAA

While similar in structure, the proposed segmentation method has key differences over DAA, with these differences most evident in the first layer of the proposed method. As the first layer of the proposed segmentation method uses BP-AR-HMM, we gain the advantages of the said model over sticky HDP-HMM. First, sequences can have shared and unique EB labels, thanks to the BP prior. That is why some of the EB labels are active in all of the sequences, while some are only active in some of the sequences. Second, using BP (in BP-AR-HMM), instead of HDP (in sticky HDP-HMM), allowed for sequence-specific transition matrices, instead of transition matrices used by all sequences. That is why not only can the sequences have different sets of active EB labels, they can also have different switching patterns for EB labels. Third, as there is an AR in BP-AR-HMM, the first layer of the proposed method can explicitly capture the dynamics of the data. These resulting AR models are useful in carrying out the second step of the proposed approach.

Based on resulting experiments, we confirmed that the proposed method indeed had some advantages over DAA. For one, the proposed method allowed for asynchronous switching of segments, unlike DAA. For DAA to allow such asynchronous switching, we had to first concatenate the time series sequences, and thus implicitly assume that they form a single time series sequence only. This was clearly not the case, as the start and end of each sequences were not connected. Second, the proposed method had better accuracy than DAA. One major cause of the high HDist for **Lq-DAA** and **CMU-DAA** was its tendency to discover only one EB, and consequently only one UB. This means that the method failed to segment the dataset at all, thus increasing the average HDist significantly.

6.2 Significance Testing

Results from the significance testing step indicated that switching interaction could be identified using the proposed segmentation approach. Specifically the obtained AR models from the segmentation step could be used to perform interaction detection. Furthermore, the choice of shuffling method and test statistic greatly affected the computed evaluation metric, where each configuration had its own set of advantages and disadvantages. Accuracy and sensitivity were moderately good while specificity was good, regardless of the test method used. Of particular interest was the trade-off between the ability to detect small-valued interactions and the run time, as seen from using SDM as test statistic. To be specific, methods that used frequency-domain statistic seemed to be capable of detecting interactions with small coefficient values. However, their run times were way longer than their time-domain counterpart. Given all these, RP-CM was deemed the best option to use for interaction detection at the EB level. It offered the best compromise between accuracy and complexity among the eight test methods considered.

Another problem that arose in the significance testing step at the EB level was the high number of significant entries discovered, as seen from the results of the motion dataset. Interpreting the results obtained at the EB level proved to be challenging, as there were cases where almost all of the entries were deemed significant. Having all entries significant would then make detecting significant interaction moot and useless. Fortunately, this was remedied when interaction detection was done at the UB level. Using the time-weighted combinations actually allowed for easier interpretation, since the temporal aspect of the interaction was now considered. Despite allowing for easier interpretation, the question of what should be the threshold value arose at this step. The threshold value would greatly affect how many variable pairs would become significant, and thus has to be carefully considered.

Moreover, it was observed that in some runs, formed UBs only had one component EB. In such cases, the significance testing at the UB level had the same result as the testing at the EB level, rending the second significance testing moot. Interpreting the results were then still difficult, thus removing the advantage of adding a second significance testing layer.

7. Conclusions

7.1 Summary

In this thesis, we put forward the problem of using a segmentation approach to switching interaction detection. Specifically, this thesis was divided into two major parts: (1) proposing a segmentation method for interaction data, and (2) developing an interaction detection method for switching interaction data using the segmentation method from (1).

To segment time series sequences from hierarchically-structured dynamical systems, we proposed a method integrating the segmentation by BP-AR-HMM and the double articulation by NPYLM. That is, we proposed an iterative two-layered unsupervised segmentation method where the first layer used BP-AR-HMM to discover low-level segments, and the second layer used NPYLM to form high-level segments. Results indicated that the proposed segmentation method can discover both low-level and high-level segments with good accuracy. Furthermore, the proposed method had high-level segments invariant to the discovered low-level segments.

To detect switching interaction, we proposed to use the obtained AR models from the first step. That is, we used the method of surrogate data to identify which entries in either the AR coefficient matrix or the spectral density matrix were significant. Surrogates were generated using different shuffling techniques, with shuffling carried out in either the time-domain or frequency-domain. Results indicated that each test method configuration had its own set of advantages and disadvantages. RP-CM (random permutation shuffling, with the coefficient matrix as test statistic) emerged as the best compromise. Results from this test method yielded moderate accuracy and good specificity. Significance at the UB level was then identified using time-weighted linear combinations. This particularly helped in reducing the number of significant entries discovered, as there were quite a handful discovered at the EB level.

7.2 Conclusion

Given all these, we then conclude that we can detect switching interaction using the proposed segmentation approach. Specifically, the proposed segmentation method can segment multiple time series sequences into low-level (EB) and high-level (UB) segments, and can be used to identify variable pairs with significant switching interaction. This is because the proposed method not only extracted EBs and UBs, but also estimated the dynamics using AR coefficients. Since using AR(1) is a feasible and viable option, the resulting coefficient matrices are simple and can be used for interaction analysis in the second step. Using method of surrogate data allows us to identify which variable pairs have significant interactions. Moreover, switching interactions can be detected when the aforementioned method is coupled with the obtained AR models from the discovered segments in the first step.

Thus, the proposed segmentation approach (and consequently, the proposed segmentation method) can extract interactions from time-switching variables. This was made possible thanks to the dynamics discovered, expressed as switching AR models. Furthermore, the proposed method has potential applications to many fields, such as causality analysis, computational neuroscience, and cognitive interaction design studies [53].

7.3 Recommendations

Despite having promising results in this thesis, several recommendations are in order to further improve this work. First of all, concerns raised in Section 1.3.1 could be addressed. For one, the proposed segmentation method should be applied to actual sequences from motion capture data involving two interacting humans to further verify its applicability to real motion data. While we did apply the proposed method to actual motion data, the switches in the sequences used were synthetic in nature. As such, it would be interesting to see if the proposed method would work for sequences with natural switching behaviors. Moreover, the significance testing step could be improve further to take into account spurious connections. These connections are caused by indirect dependencies between variable pairs, such as when the two variables have a common driver or "cause" [44]. Runge, et al. suggested using transfer entropy (specifically conditional mutual information (CMI)) and causal discovery algorithms, instead of Granger causality, as the basis for interaction detection [44].

There are also recommendations on the model side of the proposed approach. These include working on a (theoretically) unified model for the segmentation step, similar to *non-parametric Bayesian DAA* [50], extending the significance testing step for higher AR lag order, so as to relax the AR(1) assumption, and modifying the second step to improve the interpretability of the results. Improvements to surrogate generation should also be considered. While doing random permutation was deemed the best way to generate surrogates in [23], Runge argued that it had the tendency to inflate false positives, especially for multivariate setting. One downside of using random permutation is that it also destroys any connections between variables that are not being tested. As such, he recommends to use a local permutation scheme, where connections not being tested are preserved, similar to how [12] used CFTd surrogates.

Acknowledgements

The journey towards this printed thesis has been a long and winding road. But I made it because of them. I may have written this document, but this document was built on their shoulders.

To MEXT, to the members of NAIST community,
to the members of Ikeda Lab, former and current alike, students and staff alike
to Kubo-sensei, to Ikeda-sensei,
to Noypi@NAIST, to HOHOL is life
to the friends I lost on the way, to the friends who left me,
to my friends I gained on the way, to my friends who stayed with me,
to Terentedos, to my friends waiting in the Philippines,
to my family,
to my God,

全部誠にありがとうございます、
isang taos pusong pasasalamat sa inyong lahat.
Pilipinas, pabalik na ako!

References

[1] R. Adhikari and R. K. Agrawal. Combining multiple time series models through a robust weighted mechanism. In *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, pages 455–460, 2012.

[2] R. Alazrai, Y. Mowafi, and C. S. G. Lee. Anatomical-plane-based representation for human-human interactions analysis. *Pattern Recognition*, 48(8):2346–2363, 2015.

[3] M. Ayazoglu, B. Yilmaz, M. Sznaier, and O. Camps. Finding causal interactions in video sequences. In *2013 IEEE International Conference on Computer Vision*, pages 3575–3582, 2013.

[4] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. Hodgins, and N. Pollard. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface*, pages 185–194, 2004.

[5] S. J. Berlin and M. John. Human interaction recognition through deep learning network. In *2016 IEEE International Carnahan Conference on Security Technology (ICCST)*, pages 1–4, 2016.

[6] Biophysics and U. o. T. Biosignals Laboratory, Department of Physics. eMVAR – Extended Multivariate Autoregressive Modelling Toolbox. Available at `http://www.science.unitn.it/~nollo/research/sigpro/eMVAR.html`, 2011.

[7] B. Bruno, F. Mastrogiovanni, A. Sgorbissa, T. Vernazza, and R. Zaccaria. Human motion modelling and recognition: A computational approach. In *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 156–161, 2012.

[8] CMU. Carnegie Mellon University Graphics Lab - Motion Capture Library. Available at `http://mocap.cs.cmu.edu/`, 2009.

[9] L. Damiano, B. Peterson, and M. Weylandt. Bayesian hierarchical hidden Markov models applied to financial time series. Available at `https://github.com/luisdamiano/gsoc17-hhmm`, 2017.

[10] Z. Dzunic and J. Fisher III. Bayesian switching interaction analysis under uncertainty. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 220–228, 2014.

[11] L. Faes, G. Nollo, S. Erla, C. Papadelis, C. Braun, and A. Porta. Detecting nonlinear causal interactions between dynamical systems by non-uniform embedding of multiple time series. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 102–105, 2010.

[12] L. Faes, A. Porta, and G. Nollo. Testing frequency-domain causality in multivariate time series. *IEEE Transactions on Biomedical Engineering*, 57(8):1897–1906, 2010.

[13] S. Fine, Y. Singer, and N. Tishby. The Hierarchical Hidden Markov Model: Analysis and Applications. *Machine Learning*, 32(1):41–62, 1998.

[14] E. Fox, E. Sudderth, M. Jordan, and A. Willsky. An HDP-HMM for systems with state persistence. In *Proceedings of the 25th International Conference on Machine Learning*, pages 312–319, 2008.

[15] E. Fox, E. Sudderth, M. Jordan, and A. Willsky. Nonparametric bayesian learning of switching linear dynamical systems. In *Advances in Neural Information Processing Systems 21*, pages 457–464, 2008.

[16] E. Fox, E. Sudderth, M. Jordan, and A. Willsky. Sharing features among dynamical systems with beta processes. In *Advances in Neural Information Processing Systems 22*, pages 549–557, 2009.

[17] E. Fox, E. Sudderth, M. Jordan, and A. Willsky. A sticky HDP-HMM with application to speaker diarization. *Annals of Applied Statistics*, 5(2 A):1020–1056, 2011.

[18] E. Fox, E. Sudderth, M. Jordan, and A. Willsky. Joint modeling of multiple time series via the beta process with application to motion capture segmentation. *Annals of Applied Statistics*, 8:1281–1313, 2014.

[19] A. Gensler, T. Gruber, and B. Sick. Blazing fast time series segmentation based on update techniques for polynomial approximations. In *Proceedings*

- *IEEE 13th International Conference on Data Mining Workshops, ICDMW 2013*, pages 1002–1011, 2013.

[20] Z. Ghahramani and T. Griffiths. Infinite latent feature models and the indian buffet process. In *Advances in Neural Information Processing Systems 18*, pages 475–482, 2006.

[21] Z. Ghahramani, T. Griffiths, and P. Sollich. Bayesian nonparametric latent feature models. *Bayesian Statistics*, 8:1–25, 2007.

[22] Z. Ghahramani and G. Hinton. Parameter estimation for linear dynamical systems. Technical report, University of Toronto, 1996.

[23] M. Gilson, A. Tauste Campo, X. Chen, A. Thiele, and G. Deco. Nonparametric test for connectivity detection in multivariate autoregressive networks and application to multiunit activity data. *Network Neuroscience*, 1(4):357–380, 2017.

[24] R. Goebel, A. Roebroeck, D.-S. Kim, and E. Formisano. Investigating directed cortical interactions in time-resolved fMRI data using vector autoregressive modeling and Granger causality mapping. *Magnetic Resonance Imaging*, 21(10):1251–1261, 2003.

[25] C. Granger. Testing for causality: A personal viewpoint. *Journal of Economic Dynamics and Control*, 2:329–352, 1980.

[26] E. C. Grigore and B. Scassellati. Discovering action primitive granularity from human motion for human-robot collaboration. In *Proceedings of Robotics: Science and Systems*, 2017.

[27] A. Gupta, P. Srinivasan, J. Shi, and L. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2012–2019, 2009.

[28] L. Harrison, W. Penny, and K. Friston. Multivariate autoregressive modeling of fMRI time series. *NeuroImage*, 19(4):1477–1491, 2003.

[29] K. Heller, Y. W. Teh, and D. Gorur. Infinite hierarchical hidden Markov models. In *Artificial Intelligence and Statistics*, pages 224–231, 2009.

[30] S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. In *Computer Vision and Image Understanding*, volume 96, pages 129–162, 2004.

[31] M. Hu, C. Ingram, M. Sirski, C. Pal, S. Swamy, and C. Patten. A hierarchical HMM implementation for vertebrate gene splice site prediction. Technical report, University of Waterloo, 2000.

[32] M. Hughes. NPBayesHMM : Nonparametric Bayesian HMM toolbox, for Matlab. Available at `https://github.com/michaelchughes/NPBayesHMM`, 2016.

[33] M. Hughes, E. Fox, and E. Sudderth. Effective split-merge Monte Carlo methods for nonparametric models of sequential data. In *Advances in Neural Information Processing Systems 25*, pages 1–9, 2012.

[34] X. Ji, C. Wang, and Z. Ju. A new framework of human interaction recognition based on multiple stage probability fusion. *Applied Sciences*, 7(6), 2017.

[35] E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: a survey and novel approach. In *Data Mining in Time Series Databases*, pages 1–21, 2004.

[36] B. Lake, R. Salakhutdinov, and J. Tenenbaum. One-shot learning by inverting a compositional causal process. In *Advances in Neural Information Processing Systems 26*, pages 2526–2534, 2013.

[37] C. McComb. Adjusted Rand Index. Available at `https://github.com/cmccomb/rand_index`, 2018.

[38] D. Mochihashi, T. Yamada, and N. Ueda. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - ACL-IJCNLP '09*, volume 1, pages 100–108, 2009.

[39] M. Mohan, R. Mendonca, and M. Johnson. Towards quantifying dynamic human-human physical interactions for robot assisted stroke therapy. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 913–918, 2017.

[40] S. Motiian, F. Siyahjani, R. Almohsen, and G. Doretto. Online human interaction detection and recognition with multiple cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):649–663, 2017.

[41] G. Neubig. latticelm. Available at `https://github.com/neubig/latticelm`, 2016.

[42] G. Neubig, M. Mimura, S. Mori, and T. Kawahara. Learning a language model from continuous speech. In *11th Annual Conference of the International Speech Communication Association (InterSpeech 2010)*, pages 1053–1056, 2010.

[43] N. Nguyen and A. Yoshitaka. Human interaction recognition using hierarchical invariant features. *International Journal of Semantic Computing*, 9(02):169–191, 2015.

[44] J. Runge. Causal network reconstruction from time series: From theoretical assumptions to practical estimation. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):075310, 2018.

[45] M. Ryoo and J. Aggarwal. Semantic representation and recognition of continued and recursive human activities. *International Journal of Computer Vision*, 82(1):1–24, 2009.

[46] A. Saeedi, M. Hoffman, M. Johnson, and R. Adams. The segmented iHMM: a simple, efficient hierarchical infinite HMM. In *International Conference on Machine Learning*, pages 2682–2691, 2016.

[47] T. Taniguchi. Double Articulation Analyzer. Available at `http://daa.tanichu.com/code`, 2017.

[48] T. Taniguchi, K. Hamahata, and N. Iwahashi. Unsupervised segmentation of human motion data using a sticky hierarchical dirichlet process-hidden markov model and minimal description length-based chunking method for imitation learning. *Advanced Robotics*, 25(17):2143–2172, 2011.

[49] T. Taniguchi and S. Nagasaka. Double articulation analyzer for unsegmented human motion using Pitman-Yor language model and infinite hidden Markov model. In *2011 IEEE/SICE International Symposium on System Integration, SII 2011*, pages 250–255, 2011.

[50] T. Taniguchi, S. Nagasaka, and R. Nakashima. Nonparametric Bayesian double articulation analyzer for direct language acquisition from continuous speech signals. *IEEE Transactions on Cognitive and Developmental Systems*, 8(3):171–185, 2016.

[51] A. Tank, N. Foti, and E. Fox. Bayesian structure learning for stationary time series. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, pages 872–881, 2015.

[52] Y. W. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[53] The Administrative Group of "Cognitive Interaction Design". Cognitive Interaction Design. Available at `https://www.cognitive-interaction-design.org/english-digest-1`, 2014.

[54] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. Doyne Farmer. Testing for nonlinearity in time series: the method of surrogate data. *Physica D: Nonlinear Phenomena*, 58(1):77–94, 1992.

[55] S. Thurner, M. Szell, and R. Sinatra. Emergence of good conduct, scaling and Zipf laws in human behavioral sequences in an online world. *PLOS ONE*, 7(1):1–7, 2012.

[56] P. Viviani and M. Cenzato. Segmentation and coupling in complex movements. *Journal of Experimental Psychology: Human Perception and Performance*, 11(6):828, 1985.

[57] B. Williams, M. Toussaint, and A. Storkey. Modelling motion primitives and their timing in biologically executed movements. In *Advances in Neural Information Processing Systems 20*, pages 1609–1616, 2007.

[58] K. Yun, J. Honorio, D. Chattopadhyay, T. Berg, and D. Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 28–35, 2012.

[59] S. Zeger, R. Irizarry, and R. Peng. On time series analysis of public health and biomedical data. *Annual Review of Public Health*, 27:57–79, 2006.

[60] H.-L. Zeng, R. Lemoy, and M. Alava. Financial interaction networks inferred from traded volumes. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(7):P07008, 2014.

[61] F. Zhou, F. De La Torre, and J. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):582–596, 2013.

[62] H. Z. Zou and Y. Yang. Combining time series models for forecasting. *International Journal of Forecasting*, 20(1):69 – 84, 2004.

Publication List

Journal Papers

[1] J. Briones, T. Kubo, K. Ikeda. Extraction of Hierarchical Behavior Patterns Using a Non-Parametric Bayesian Approach. *Frontiers in Computer Science* 2:546917. 2020. doi: 10.3389/fcomp.2020.546917

Peer-Reviewed Conferences and Workshops

[1] J. Briones, T. Kubo, K. Ikeda. Detecting Switching Causal Interactions Using Hierarchical Segmentation Approach. *NeurIPS 2018 Workshop: NeurIPS 2018 Workshop on Causal Learning.* Quebec, Canada. Dec 2018.

[2] J. Briones, T. Kubo, K. Ikeda. Detecting Switching Interaction Using Non-Parametric Bayesian Segmentation Approach. *NeurIPS 2018 Workshop: BNP @ NeurIPS 2018: All of Baysian Nonparametrics (Especially the Useful Bits).* Quebec, Canada. Dec 2018.

[3] J. Briones, T. Kubo, K. Ikeda. Double Articulation Approach for Segmentation of Human Interaction using BP-AR-HMM and NPYLM. *NeurIPS 2017 Workshop: Advances in Modeling and Learning Interactions from Complex Data.* California, USA. Dec 2017.

Non-Peer-Reviewed Conferences and Workshops

[1] J. Briones, T. Kubo, K. Ikeda. A Segmentation-Based Approach for Detecting Switching Interaction. *10th IBRO World Congress of Neuroscience.* Daegu, South Korea. Sep 2019.

[2] J. Briones, T. Kubo, K. Ikeda. A Segmentation Approach for Interaction Analysis Using Non-Parametric Bayesian Methods. *62nd Annual Conference of the Institute of Systems, Control and Information Engineers.* Kyoto, Japan. May 2018.