CPM: A General Feature Dependency Pattern Mining Framework for Contrast Multivariate Time Series

Qingzhe Li[a], Liang Zhao[b,*], Yi-Ching Lee[a], Avesta Sassan[a], Jessica Lin[a]

[a] *George Mason University, 4400 University Drive, Fairfax, VA, USA*
[b] *Emory University, 201 Dowman Dr, Atlanta, GA 30322*

Abstract

With recent advances in sensor technology, multivariate time series data are becoming extremely large with sophisticated but insightful inter-variable dependency patterns. Mining contrast dependency patterns in controlled experiments can help quantify the differences between control and experimental time series, however, overwhelms practitioners' capability. Existing methods suffer from determining whether the differences are caused by the intervention or by different states. We propose a novel Contrast Pattern Mining (CPM) framework to find the intervention-related differences by jointly determining and characterizing the dynamic states in both time series via multivariate Gaussian distributions. Under the CPM framework, we not only propose a new covariance-based contrast pattern model, but also integrate our previous proposed partial correlation-based model as a special case. An efficient generic algorithm is developed to optimize various CPM models by adjusting one of the sub-routines. Comprehensive experiments are conducted to analyze the effectiveness, scalability, utility, and interpretability of the proposed framework.

Keywords: contrast pattern, feature dependency, controlled experiment, driving behavior, multivariate time series

*Corresponding author
Email address: liang.zhao@emory.edu (Liang Zhao)

1. Introduction

With recent technological advances and the growing popularity of various sensors, multivariate time series datasets are becoming extremely large with complicated but very useful inter-variable dependencies that dynamically change over time. Mining contrast patterns in controlled experiments can help quantify the differences between large-scale control and experimental multivariate time series but will overwhelm domain experts' capability while manually labeling and interpreting the effects as usual [1].

Consider the controlled experiment shown in Figure 1, whose goal is to evaluate the effects of the medicine (i.e. the intervention factor) on the driver's driving behaviors. Both the control time series and the experimental time series, which are together called the Contrast Multivariate Time Series (CMTS), are recorded by in-vehicle sensors such as "brake," "accelerator," and "steering wheel." The dependency network inferred from a small time series segment can characterize the driver's current driving behavior [2]. For example, the strong dependency between the node denoting the "brake" at time t and the node of "steering wheel" at time $t+1$ precisely characterize the latent state of "turning" behavior; the weak dependency between the "brake" and "steering wheel" along with the strong dependency between the "brake" at time t and $t+1$ characterize another latent state of "deceleration" behavior. One the other hand, the intervention may or may not change the dependency patterns depends on whether the intervention affects the driver's driving behaviors. Notice that comparing two dependency patterns is meaningful only when they are characterizing the same driving behavior. To evaluate and interpret the effects of the intervention, we mine the contrast pattern in CMTS by jointly solving three subproblems: 1) determining the dynamic driving states in each time series; 2) characterizing the driving states by inferring the dependency networks; 3) contrast the dependency networks under the same driving state.

Existing work can only address one or two subproblems but not all of them. For example, Hollac et al. proposed a Toeplitz Inverse Covariance-based Clus-

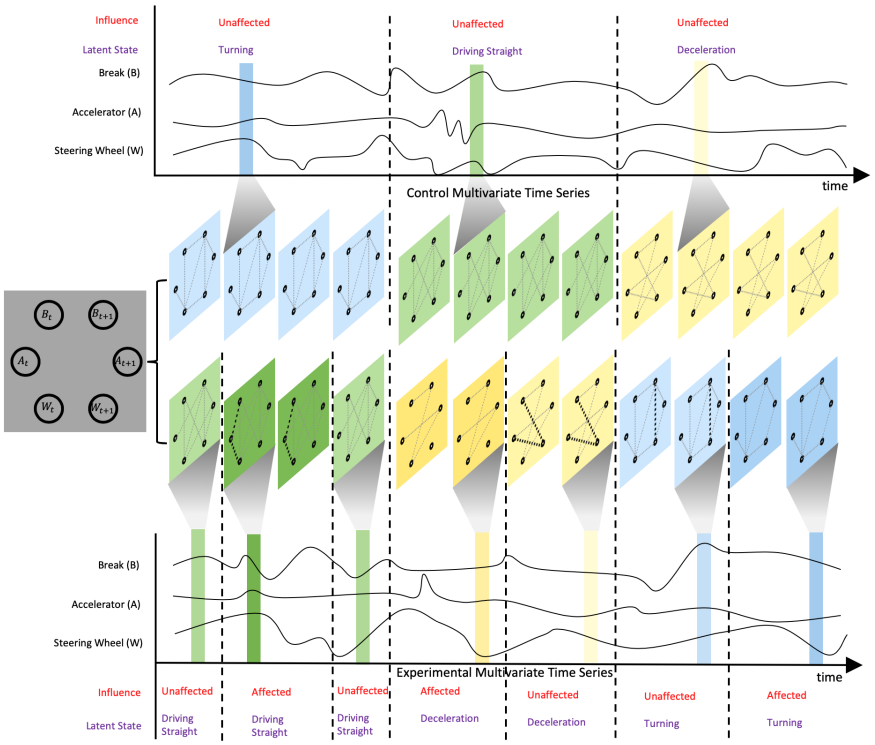


Figure 1: A controlled driving behavior experiment (better seen in colors): The control (i.e., before taking the medication) and experimental time series (i.e., after taking the medication) are plotted at the top and bottom portions. The dynamically changed dependency network for each time series are plotted at the middle portion, where the nodes denote the sensors at time t or $t+1$, the edges denote the dependency between the connected nodes. The dependency networks plotted on blue, green, or yellow plates characterize the latent driving states of 'turning", "driving straight" and "deceleration," respectively. The latent states plotted on deeper-colored plates denote the driving behaviors that are affected by the intervention. For example, the latent state plotted on deep green plate denotes the dependency network that characterizes the latent state of "driving straight" has been affected by the intervention at the corresponding time shaded in the time series. The highlighted edges demonstrate the differences between the affected and unaffected dependency networks at the same latent state.

tering (TICC) model [2] to cluster time series subsequences by characterizing a finite number of latent states via network inference without considering the contrast patterns. Liu et al. proposed the unified and contrasting graphical lasso [3] without considering various latent states in time series. Several challenges prevent the existing work from being directly utilized or trivially combined to address contrast pattern mining problem: **1. Difficulty in integrally modeling the contrast pattern mining problem for controlled experiments.** This problem requires not only identifying the latent states but also detecting

the locations of the contrast patterns, as well as characterizing the contrast patterns. These subproblems tightly couple with each other, and thus should be jointly modeled. Simply using the existing models to solve the subproblems separately will fail to obtain a joint optimal solution. **2. Difficulty in differentiating the patterns featured by the latent states and those caused by the intervention.** The dependency patterns between two latent states are different from those affected by the intervention. We notice that the dependency patterns may significantly change when switching to another latent state, while the dependency patterns may slightly change with and without the intervention under the same latent state. **3. Difficulty in characterizing the differences among the latent states under various cases.** Characterizing the "significantly" and "slightly" changes may vary in different controlled experiments. Therefore, a general framework that can characterize such difference that adapts to various cases is preferred. **4. Difficulty in jointly modeling and optimizing the discrete and continuous variables under an integrated framework.** Modeling the above-mentioned subproblems requires jointly optimizing the discrete (e.g., the latent state and contrast pattern assignments) and continuous (e.g., the dependencies) variables that are highly coupled together. Moreover, solving the optimization problem for a framework is much difficult than which for a single model. This is prohibitively difficult to solve with the existing optimization algorithms.

In our previous paper [4], we addressed the first two challenges by proposing a Contrast Pattern Mining with Partial correlation-based feature dependency regularization (CPM-P) model when the latent states can be differentiated by the partial correlations among the variables. In order to handle the cases of which the latent states can be differentiated by other meaningful metrics (e.g. covariance), we propose a contrast pattern mining framework for CMTS in various controlled experiments, along with a universe optimization framework based on Expectation-Maximization (E-M) and Alternating Direction Method of Multiplier (ADMM). Our main contributions are summarized as follows:

- **Designing a framework to learn the dynamic multivariate depen-**

dency patterns for CMTS. We propose a novel contrast pattern mining framework to jointly optimize latent state assignments, contrast pattern detection, and characterization of the contrast dependency patterns. Although this paper focuses on the application of contrast driving behavior, our framework is generic to controlled experiments in many other domains. To the best of our knowledge, this framework is the first to identify the effects of the intervention in an unsupervised and interpretable manner.

- **Proposing a contrast pattern mining model with a novel covariance based feature dependency regularization for the proposed framework.** We first model the CMTS by characterizing the generative process of the control and experimental time series. Then, we propose a novel covariance based feature dependency regularization to handle the case when the differences are distinguishable by the variance and covariance of the variables.

- **Developing efficient optimization framework for the non-convex and semi-discrete problem.** We develop a generic Expectation-Maximization (EM)-like framework for all the models under the proposed framework. The discrete variables and continuous variables are alternatively optimized by dynamic programming and developing a non-convex ADMM-based framework.

- **Conducting comprehensive evaluations and interpreting the results on a real-world controlled experiment.** Extensive experiments demonstrate the effectiveness, scalability, and robustness of the proposed method. Evaluations based on a controlled experiment on driving behaviors are provided, which demonstrates the effectiveness and interpretability of the proposed approach.

The remaining contents are organized as follows. Section 2 reviews related work. Section 3 formulates the problem of CDFD pattern mining for CMTS. Section 4 presents our CDFD pattern mining framework. Our optimization algorithms are elaborated in Section 5. Extensive experiments on synthetic datasets and real-world application are conducted in Section 6 and Section 7. The entire work concludes in Section 8.

2. Related Work

The work related to this research is summarized in the areas of 1) time series subsequence clustering, 2) time series anomaly detection, and 3) contrast pattern mining of time series.

Subsequence clustering of time series: Subsequence clustering is an important technique to identify the latent states in a long time series. However, clustering all overlapped subsequences using (dis)similarity-based approaches has been proven to produce meaningless results [5] due to the reuse of the data points in neighboring subsequences. Since then, some meaningful similarity-based approaches have been proposed to avoid this pitfall. For example, Rakthanmanon et al. proposed a parameter-free framework using minimum description length framework to cluster time series subsequences by ignoring some data [6]. Recently, Fotsol et al. proposed a scalable U-shapelet discovery algorithm for time series clustering along with a new dissimilarity score based on local correlation [7]. These (dis)similarity-based approaches cluster time series by their "shapes" [8], as opposed to our dependency-based patterns among the variables in multivariate time series. Further, model-based time series clustering approaches such as those based on an autoregressive moving average model [9], Gaussian mixture model [10], or hidden Markov model [11], typically consider the whole sequence rather than subsequence except for Toeplitz Inverse Covariance-based Clustering (TICC) [2], proposed by Hallac et al. The TICC approach clusters the subsequences in a single multivariate time series according to connectivity patterns estimated by a graphical lasso. Most recently, Li et al. further extended TICC by introducing adaptive cluster switching penalty according to the distance between neighboring subsequences [12]. TICC only focuses on single time series, which neither considers the relationship between the control and experimental time series nor mines their contrast patterns.

Anomaly detection in time series: Anomaly detection techniques can be categorized into threshold based and non-threshold based approaches. The threshold based approaches include one-class SVM [13], frequency of mismatches

[14], soft anomaly scores[15], elliptic envelope [16], isolation forest [17], and local outlier factor [18]. There are multiple deep learning-based time series anomaly detection techniques. Malhotra et al. [19] proposed an LSTM-based Encoder-Decoder architecture to detection anomalies, which first builds a model that minimizes the reconstruction error by using the "normal" time series data, then detects the anomalies according to the reconstruction errors in another time series with anomalies. Following the same Encoder-Decoder architecture, Zhang et al. [20] recently propose a more sophisticated anomaly detection model for multivariate time series. The threshold-based approaches cannot directly solve our contrast pattern mining problem because of the lack of prior knowledge to determine the threshold even for the domain experts. For example, the one-class SVM based approach [21] needs to set the radius to define the boundary; the frequency or anomaly score-based approaches need to set a threshold as the boundary frequency or score; the Encoder-Decoder based model needs to detect the anomaly by setting up a threshold on the reconstruction error. The non-threshold based approaches include clustering-based approaches [22], parametric approaches [23], subspace anomaly detection methods for multivariate time series [24], and discord-based approaches [25]. However, these approaches mostly assume the anomalies rarely occur in the dataset, which does not hold for our contrast pattern mining problem.

Contrast pattern mining for time series: Research on contrast pattern mining between two multivariate time series has recently emerged. Researchers have explored multivariate time series generated in functional MRI to mine the contrast patterns by proposing various network inference models [26, 3]. For instance, Lee et al. proposed a CNN based deep neural network [26] to identify contrasting dependency networks inferred from the entire time series. Similarly, Liu et al. proposed a contrast graphical lasso model [3] for whole time series. The model derives a single contrast dependency network that corresponds to two multivariate time series. However, neither of these methods considers the fact that the contrast patterns are only meaningful while they are compared under the same latent state in the subsequence level.

Table 1: Notations

Notation	Description
X,$\hat{X}$	contrast multivariate time series
T, $\hat{T}$	the lengths (i.e. number of rows) of X,$\hat{X}$
Y, $\hat{Y}$	latent state assignments, where $\|Y\| = T$ and $\|\hat{Y}\| = \hat{T}$
Z	contrast pattern indicator for $\hat{X}$, and $\|Z\| = \hat{T}$
$\theta_k, \hat{\theta}_k$	contrast inverse covariance matrices of the k-th latent state
K	the count of the latent states
w	sliding window size
β, γ, λ	regularization parameters

3. Problem Setup

We first define the relevant terminologies, then present the new research problem of contrast dynamic feature dependency pattern mining in controlled experiments.

A multivariate time series $x = [x_1, \cdots, x_m]$ is a time-ordered sequence of m vectors where $x_t \in \mathbb{R}^{n \times 1}$ is a multivariate observation that contains n variables at time t. Instead of following the independent and identically distributed (i.i.d.) assumption, the observation of x_t is also dependent on its context. To capture both dependencies among different sensors and different nearby time indices, we first concatenate the observations x_t and its $w-1$ successors extracted by a sliding window of size $w \ll m$, which formulates an nw-dimensional row vector $X_t = [x_t^\intercal, \cdots, x_{t+w-1}^\intercal]$ to denote a multivariate time series *subsequence*. Then we stack all these subsequences, from X_1 to X_T, into a matrix $X \in \mathbb{R}^{T \times nw}$ where $T = m - w + 1$. By doing this, the dependencies in original time series x can be represented by the dependencies among the $n \cdot w$ **features**/columns in X. Due to the one-to-one relationship between x and X for a given w, we still call X a multivariate time series. The multivariate time series data X usually exhibits different *latent states* that may dynamically switch over time. These latent states are reflected by both the values of different features and their dependency patterns. For instance, the multivariate time series that record a driving session, can involve three latent states: "Acceleration," "Deceleration," and "Turning." For the "Acceleration" state, its dependency network should contain a strong dependency between the "Accelerator" sensor at time t and time $t+1$, and should not contain any strong dependency between other features. The other

two latent states should be characterized by completely different dependency patterns. We use $Y \in \{0,1\}^{T \times K}$ to denote the assignments of the latent state for all subsequences where the hyperparameter K is the number of latent states. Specifically, $Y_{t,k} = 1$ if X_t belongs to the k-th latent state; otherwise, $Y_{t,k} = 0$. As X only contains continuous values, each latent state can be naturally characterized by a multivariate Gaussian distribution parameterized by an inverse covariance matrix $\theta_k \in \mathbb{R}^{nw \times nw}$. The inverse covariance matrix θ_k may encode the dependency network $G_k = (X_t, \theta_k)$ as a correlation network or partial correlation network [27] whose nodes denote the features and whose weighted edges denote the correlation or partial correlation between the connected features.

In controlled experiments, the two multivariate time series are generated from the control and the experimental sessions. They are contrasted to explore the possible differences caused by the intervention. We call the two multivariate time series in the controlled experiments *contrast multivariate time series (CMTS)*. As other factors are strictly controlled to diminish their effects on the subject, the two multivariate time series usually share the same set of latent states. Formally, the CMTS is defined as follows:

Definition 1. *[**Contrast Multivariate Time Series**] The contrast multivariate time series (CMTS) contains two multivariate time series such that 1) the control multivariate time series $X \in \mathbb{R}^{T \times nw}$ are generated without the intervention and the experimental multivariate time series $\hat{X} \in \mathbb{R}^{\hat{T} \times nw}$ are generated with the intervention, 2) X and $\hat{X}$ share the same set of the latent states.*

In the controlled experiments on driving behavior, the driver was asked to drive without the intervention (e.g., without taking the medicine), and then the same driver was asked to drive with the intervention such as taking medicine, drinking alcohol, and etc. The control factors can be the same driver, same routines, and so on. To identify the effects of the intervention, it is natural to contrast their patterns under the same latent state. Concretely, the contrast pattern in controlled experiments is formally defined as follows:

Definition 2. *[**Contrast Dynamic Feature Dependency**] For the subsequence $\hat{X}_{\hat{t}}$ belonging to the k-th latent state, if the intervention changes the*

original feature dependencies θ_k *into a new one* $\hat{\theta}_k$*, there exists the* ***contrast dynamic feature dependency (CDFD)*** *pattern at time* $\hat{t}$*. Hence, we use a* contrast indicator $Z \in \{0,1\}^{\hat{T}\times 1}$*, to signify the existence of CDFD in* $\hat{X}$ *caused by the intervention.* $Z_t = 0$ *if there exists CDFD in* $\hat{X}_t$*; otherwise,* $Z_t = 1$*.*

A driver may accelerate more quickly after she/he takes medicine that stimulates adrenaline in some road segments, which demonstrates the contrast pattern in the controlled experiments. In this paper, our goal is to identify and characterize the CDFD patterns for CMTS in controlled experiments. The problem is formally defined as follows:

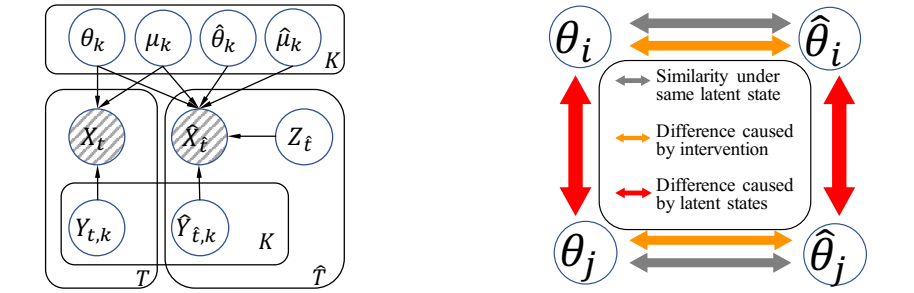
Problem Formulation: Given the CMTS X and $\hat{X}$ our goal is to simultaneously discover the interpretable CDFD patterns, including 1) to determine the latent state assignments Y and $\hat{Y}$ for X and $\hat{X}$, respectively, 2) to characterize the K latent states by learning their CDFD patterns $\theta = \{\theta_k\}_k^K$ and $\hat{\theta} = \{\hat{\theta}_k\}_k^K$, and 3) to decide the Z assignments by detecting the CDFD.

For example, for the problem of mining the CDFD patterns in the controlled experiment on driving behavior, in order to test the effectiveness of taking some medicine, the research goals are : 1) to determine the driving state assignments Y and $\hat{Y}$, 2) to characterize the K latent driving states encoded by θ and $\hat{\theta}$, and 3) to decide the Z assignments based on whether driving behaviors have been changed after medication.

The above problem poses the following main technical challenges: 1) *Difficulty in modeling the CMTS for controlled experiments.* In CMTS, different features are not independent. This problem requires new methods that can simultaneously characterize their underlying distributions and how they are changed by the intervention dynamically over time, which cannot be addressed by existing methods. 2) *Difficulty in differentiating the patterns derived from latent states and those from interventions.* For the contrast pattern under the same latent state, the CDFD patterns encoded by θ_k and $\hat{\theta}_k$ are similar in nature but could also be differentiated because of the intervention. However, quantitatively and rigorously distinguishing such similarity and difference is important yet prohibitive for the existing methods. 3) *Difficulty in jointly inferring all the model*

parameters. None of the existing algorithms can directly solve the optimization problem for two reasons: First, it is challenging to handle the coupling between θ and $\hat{\theta}$, which is necessary to ensure that θ_k and $\hat{\theta}_k$ are characterizing the same latent state for any $k = 1, \cdots, K$; second, it is also challenging to jointly optimize the discrete variables (i.e., $Y, \hat{Y}$, and Z) and continuous variables (i.e. θ and $\hat{\theta}$) in a unified optimization framework effectively.

4. The Methodologies



(a) Plate notations of the generative model

(b) Relationships among the CDFD networks

Figure 2: The Contrast Pattern Mining Framework

This section proposes the Contrast Pattern Mining (CPM) framework. We first establish the generative model for CMTS and then present our new covariance based contrast feature dependency regularization. Finally, the temporal regularization on the latent state assignments are presented.

4.1. Generative Model of CMTS

The time series subsequences X_t and $\hat{X}_{\hat{t}}$ in CMTS are continuous variables in controlled driving behavior experiments, so they are modeled to be sampled from a set of multivariate Gaussian distributions. The generative process is depicted in Figure 2a. By Definition 1 for any $(t=1, \cdots, T)$, X_t belongs to one of the K latent states, and hence for each subsequence X_t at time index t, we draw $X_t \sim \mathcal{N}(X_t|\theta_k, \mu_k)$ in Figure 2a, where θ_k and μ_k are respectively the

inverse covariance matrix and the mean vector to be estimated by the X_t data assigned to the k-th latent state. The joint conditional distribution of X is:

$$p(X|Y,\theta) = \prod_{k,t}^{K,T} \mathcal{N}(X_t|\theta_k,\mu_k)^{Y_{t,k}},$$

where $\theta = \{\theta_k\}_k^K$. Similarly, for any $(\hat{t} = 1, \cdots, \hat{T})$, $\hat{X}_{\hat{t}}$ also belongs to one of the K latent states. However, the vectors $\hat{X}_{\hat{t}}$ that belong to the k-th latent state are possibly generated either from a new distribution $\mathcal{N}(\hat{X}_{\hat{t}}|\hat{\theta}_k,\hat{\mu}_k)$ or from $\mathcal{N}(\hat{X}_{\hat{t}}|\theta_k,\hat{\mu}_k)$. Here, $\hat{\theta}_k$ and $\hat{\mu}_k$ are respectively the inverse covariance matrix and the mean vector to be estimated by the $\hat{X}_{\hat{t}}$ data assigned to the k-th latent state. Specifically, when $Z_{\hat{t}} = 0$ (i.e., the CDFD pattern exists), we draw $\hat{X}_{\hat{t}} \sim \mathcal{N}(\hat{X}_{\hat{t}}|\hat{\theta}_k,\hat{\mu}_k)$ in Figure 2a. Otherwise, when $Z_{\hat{t}} = 1$ (i.e., the CDFD pattern does not exist), we draw $\hat{X}_{\hat{t}} \sim \mathcal{N}(\hat{X}_{\hat{t}}|\theta_k,\hat{\mu}_k)$ in Figure 2a. Therefore the conditional joint distribution of $\hat{X}_{\hat{t}}$ is:

$$p(\hat{X}|\hat{Y},Z,\theta,\hat{\theta}) = \prod_{k,\hat{t}}^{K,\hat{T}} [\mathcal{N}(\hat{X}_{\hat{t}}|\theta_k,\hat{\mu}_k)^{Y_{t,k}}]^{Z_{\hat{t}}} [\mathcal{N}(\hat{X}_{\hat{t}}|\hat{\theta}_k,\hat{\mu}_k)^{\hat{Y}_{t,k}}]^{(1-Z_{\hat{t}})}.$$

Based on the equations above, the joint likelihood of $(X, \hat{X})$ conditioned on the parameters $Y, \hat{Y}, Z, \theta$, and $\hat{\theta}$ is:

$$p(X,\hat{X}|Y,Z,\theta,\hat{\theta}) = p(X|Y,\theta) \cdot p(\hat{X}|\hat{Y},Z,\theta,\hat{\theta}). \quad (1)$$

Therefore, given the CMTS $(X, \hat{X})$ data, maximizing the likelihood of Equation (1) is equivalent to minimizing the negative log likelihood, which leads to our loss function:

$$\mathcal{L}(Y,\hat{Y},Z,\theta,\hat{\theta}) = -\sum_{t,k}^{T,K} Y_{t,k}\ell\ell(X_t,\theta_k) - \sum_{\hat{t},k}^{\hat{T},K} \hat{Y}_{t,k}[Z_{\hat{t}}\ell\ell(\hat{X}_{\hat{t}},\theta_k) + (1-Z_{\hat{t}})\ell\ell(\hat{X}_{\hat{t}},\hat{\theta}_k)], \quad (2)$$

where $\ell\ell(a,\Gamma) = -\frac{1}{2}(a^\intercal - \mu)^\intercal \Gamma (a^\intercal - \mu) + \frac{1}{2}\log\det\Gamma - \frac{n}{2}\log(2\pi))$ denotes the log likelihood that vector a comes from the Gaussian distribution with the inverse covariance matrix Γ.

4.2. Feature Dependency Regularization

As discussed in Section 3, it is only meaningful to contrast two feature dependency patterns for the same latent state. However, for existing models, it is very difficult to characterize the complicated relationships among feature dependency networks belonging to different latent states with or without the contrast patterns. For example, consider the four feature dependency patterns encoded by $\theta_i, \hat{\theta}_i, \theta_j, \hat{\theta}_j$, as shown in Figure 2b. θ_i and $\hat{\theta}_i$ should be similar (i.e., the grey

arrows) such that both of them characterize the i-th latent state, but on the other hand, they should also be different (i.e., the orange arrows) to characterize the effect caused by the intervention. In addition, θ_i and θ_j (or $\hat{\theta}_i$ and $\hat{\theta}_j$) should characterize the differences between the i-th and j-th latent states in a different way (i.e., the red arrows). To ensure θ_k and $\hat{\theta}_k$ are characterizing the same latent state, the difference between θ_k and $\hat{\theta}_k$ should be regularized under appropriate metrics. Traditionally, the inverse covariance matrices are mostly regularized by penalizing the element-wise differences with an L1-norm or L2-norm (e.g., [28]). However, these regularizations are flawed because a single element in the inverse covariance matrix does not have any mathematical or statistical meaning. Moreover, the values of the non-diagonal elements depend on the diagonal elements. Both flaws prohibit directly regularizing the inverse covariance matrices. To address these flaws, we propose a new feature dependency regularization framework for the CDFD pattern mining problem:

$$\mathcal{R}_{\mathcal{C}}(\theta, \hat{\theta}) = \lambda \cdot \sum\nolimits_{k}^{K} \boldsymbol{\delta}(\boldsymbol{f}(\theta_k), \boldsymbol{f}(\hat{\theta}_k)),$$

where $\lambda \in \mathbb{R}$ is the contrast regularization parameter that penalizes the differences between two dependency networks. $\boldsymbol{\delta}(\cdot, \cdot)$ can be any metrics that are meaningful for the transformation function $\boldsymbol{f}(\cdot)$. Our regularization framework is generic and can adopt different metrics that fit to contrast patterns. In the rest of this paper, we first propose a novel covariance-based metric under the CPM framework, then we plug our previously proposed partial correlation based regularization to the same framework.

Covariance Based Transformation Function: Although a single element of the inverse covariance matrix θ_k and $\hat{\theta}_k$ is mathematically meaningless, every element of the covariance matrices, which are defined as: $\boldsymbol{f}(\theta_k) = \theta_k^{-1}$ and $\boldsymbol{f}(\hat{\theta}_k) = \hat{\theta}_k^{-1}$, is interpretable. For example, the i-th diagonal element $\theta_{k,i,i}^{-1}$ and $\hat{\theta}_{k,i,i}^{-1}$ represents the variance of the i-th feature, the non-diagonal elements $\theta_{k,i,j}^{-1}$ and $\hat{\theta}_{k,i,j}^{-1}$ $(i \neq j)$ represent the covariance between the i-th and j-th features. Thus, our covariance-based regularization is defined as:

$$\boldsymbol{\delta}(\boldsymbol{f}(\theta_k), \boldsymbol{f}(\hat{\theta}_k)) := \|\theta_k^{-1} - \hat{\theta}_k^{-1}\|_F^2 \tag{3}$$

We use the squared L2-norm for $\boldsymbol{\delta}(\cdot)$ when no assumption is made. It can also be replaced by other metrics under a specific assumption.

Partial Correlation-Based Transformation Function: Our previously proposed partial correlation based regularization [4] can seamlessly adapt to this framework. To do so, we define our partial correlation-based transformation functions as $\boldsymbol{f}(\theta_k) = \rho_k$ and $\boldsymbol{f}(\hat{\theta}_k) = \hat{\rho}_k$, where the non-diagonal elements of the partial correlation matrices can be computed by $\rho_{k,i,j} = -\theta_{k,i,j}/(\theta_{k,i,i}\theta_{k,j,j})^{\frac{1}{2}}$ and $\hat{\rho}_{k,i,j} = -\hat{\theta}_{k,i,j}/(\hat{\theta}_{k,i,i}\hat{\theta}_{k,j,j})^{\frac{1}{2}}$. Finally, we define the partial correlation-based regularization adapting to the proposed framework as:

$$\boldsymbol{\delta}(\boldsymbol{f}(\theta_k), \boldsymbol{f}(\hat{\theta}_k)) := \|\rho_k - \hat{\rho}_k\|_F^2. \tag{4}$$

4.3. Temporal Regularization

Due to the nature of temporal continuity in time series, neighboring points tend to have consistent latent state assignments and contrast indicator values. We thus penalize the divergence of the assignments between the neighboring time indices by proposing the following smoothing term:

$$\mathcal{R}_{\mathcal{T}}(Y, \hat{Y}, Z) = \sum\nolimits_{t=2}^{T} \gamma \mathbb{1}(Y_t \neq Y_{t-1}) + \sum\nolimits_{\hat{t}=2}^{\hat{T}} [\beta \mathbb{1}(Z_{\hat{t}} \neq Z_{\hat{t}-1}) + \gamma \mathbb{1}(\hat{Y}_{\hat{t}} \neq \hat{Y}_{\hat{t}-1})],$$

where $\mathbb{1}(\cdot)$ is an indicator function that maps "True" values to 1 and "False" values to 0, β is the penalty if $Z_t \neq Z_{t-1}$, and γ is the penalty of switching among the K latent states.

4.4. The Overall Objective Function:

The objective of the CPM framework for CMTS is as follows:

$$\arg\min_{\theta, \hat{\theta}, Y, \hat{Y}, Z} \quad \mathcal{L}(Y, \hat{Y}, Z, \theta, \hat{\theta}) + \mathcal{R}_{\mathcal{C}}(\theta, \hat{\theta}) + \mathcal{R}_{\mathcal{T}}(Y, \hat{Y}, Z), \tag{5}$$

where $\{\theta_k, \hat{\theta}_k\} \succ 0$ are positive definite matrices such that logdet$(\cdot)$ is defined in a valid domain. The hyper-parameters K and w, can be chosen based on prior knowledge, through cross-validation, or by a principled method such as the Bayesian information criterion [29]. If the number of subsequences assigned to any latent state is too small (e.g. < 30) to learn a good θ_k and $\hat{\theta}_k$, this indicates that the value of K should be decreased. Since the short term temporal dependency is much stronger than the long term one in real-world applications, the window size w should be small (e.g. $w < 10$).

Algorithm 1 Overall Optimization Framework

Require: $X, \hat{X}, K, w, n, \lambda, \beta, \gamma$
Ensure: solution $Y, \hat{Y}, Z, \theta, \hat{\theta}$
1: $\{Y, \hat{Y}\} \leftarrow$ random initialization
2: $Z \leftarrow \mathbf{0}$
3: **repeat**
4: **for** $k = 1, \cdots, K$ **do**
5: $\Phi_k \leftarrow \{X_t | Y_{t,k} = 1\} \bigcup \{\hat{X}_{\hat{t}} | \hat{Y}_{\hat{t},k} = 1 \text{ AND } Z_{\hat{t}} = 1\}$
6: $\Psi_k \leftarrow \{\hat{X}_{\hat{t}} | \hat{Y}_{\hat{t},k} = 1 \text{ AND } Z_{\hat{t}} = 0\}$
7: $[\theta_k, \hat{\theta}_k] \leftarrow \text{ADMM\_solver}(\lambda, \Psi_k, \Phi_k)$
8: **end for**
9: $Y \leftarrow$ Updating Y by fixing θ
10: $(\hat{Y}, Z) \leftarrow$ Updating $\hat{Y}$ and Z by fixing θ and $\hat{\theta}$
11: **until** $Y, \hat{Y}$ and Z assignments are stationary
12: **return** $Y, \hat{Y}, Z, \theta, \hat{\theta}$

5. Optimization Algorithms for CPM Framework

In this section, the parameter optimization framework for the proposed CPM framework are elaborated and analyzed.

The optimization general objective of CPM framework in Equation (5) is a mixture of combinational optimization of discrete variables (i.e., $Y, \hat{Y}, Z$) and continuous variables (i.e., $\theta, \hat{\theta}$) with the non-convex term (i.e. the feature dependency regularization term). Jointly optimizing these variables under different regularization terms is prohibitively difficult to be solved by the existing algorithm or framework. To address this challenge and optimize the proposed model, we propose an Expectation Maximization (EM)-like optimization framework, outlined in Algorithm 1, which can adapt to solve the general objective framework defined in Equation (5) for all types of feature dependency regularization terms. After a random initialization of the discrete variables' assignments, Lines 3-12 alternatively optimize the continuous variables and discrete variables until the discrete assignments are stationary. Specifically, the maximization step (M-step) optimizes θ and $\hat{\theta}$ in Lines 4-8, and then the expectation step (E-step) optimizes the $Y, \hat{Y}$ and Z assignments in Lines 9-10. The details of solving each step are elaborated below.

Algorithm 2 M-step: Optimize continuous variables by ADMM

Require: Φ_k, Ψ_k, λ
Ensure: solution $\theta_k, \hat{\theta}_k$
$\{\theta_k, \hat{\theta}_k, P_k, \hat{P}_k, Q_k, \hat{Q}_k, V_k\} \leftarrow 0^{nw \times nw}$
repeat
  Update $\theta_k \leftarrow D_k \Delta D_k^\intercal$
  Update $\hat{\theta}_k \leftarrow \hat{D}_k \hat{\Delta}_k \hat{D}_k^\intercal$
  Update P_k and $\hat{P}_k$
  Update Q_k and $\hat{Q}_k$
  Update V_k
  Update the dual variables
until the stopping criteria is satisfied [30] (Chapter 3.3)
return $\theta_k, \hat{\theta}_k$

5.1. M-step: Optimizing the Continuous Variables

In M-step, we fix the assignments of the discrete variables and optimize K pairs of θ_k and $\hat{\theta}_k$ in parallel, so the subproblem for each k-th pair in M-step is:

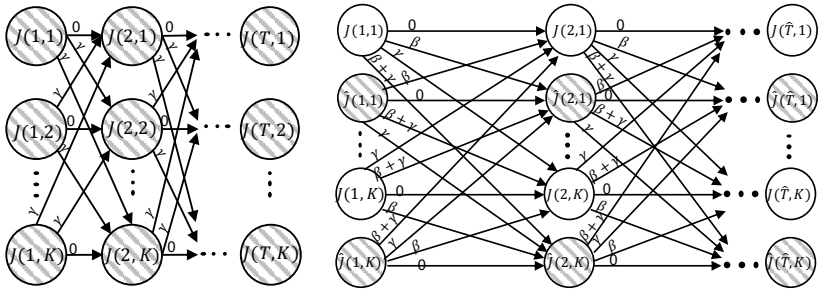
$$\underset{\{\theta_k, \hat{\theta}_k\} \succ 0}{\arg\min}\ \lambda \boldsymbol{\delta}(\theta_k, \hat{\theta}_k) - \sum\nolimits_t^T Y_{t,k} \ell\ell(X_t, \theta_k) - \sum\nolimits_{\hat{t}}^{\hat{T}} \hat{Y}_{t,k} [Z_{\hat{t}} \ell\ell(\hat{X}_{\hat{t}}, \theta_k) + (1 - Z_{\hat{t}}) \ell\ell(\hat{X}_{\hat{t}}, \hat{\theta}_k)] \quad (6)$$

Generic Framework for solving Equation (6) As Equation (6) contains non-smooth and non-convex terms, which make the equation difficult to solve directly, we propose to optimize it by following the alternating direction method of multiplier (ADMM) [30] framework. To do so, we first re-write Equation (6) into an ADMM-friendly format:

$$\begin{aligned} &\arg\min\nolimits_{\{\theta_k, \hat{\theta}_k\} \succ 0} - \sum\nolimits_{\hat{t}}^{\hat{T}} \hat{Y}_{t,k} [Z_{\hat{t}} \ell\ell(\hat{X}_{\hat{t}}, \theta_k) + (1 - Z_{\hat{t}}) \ell\ell(\hat{X}_{\hat{t}}, \hat{\theta}_k)] \\ &- \sum\nolimits_t^T Y_{t,k} \ell\ell(X_t, \theta_k) + \lambda \boldsymbol{\delta}(Q_k, \hat{Q}_k) \text{ s.t. } Q_k = \boldsymbol{f}(\theta_k), \hat{Q}_k = \boldsymbol{f}(\hat{\theta}_k). \end{aligned} \quad (7)$$

This ADMM framework can adapt to various definitions of $\boldsymbol{\delta}(\cdot)$ and $\boldsymbol{f}(\cdot)$ by adjusting the subproblems of optimizing Q_k and $\hat{Q}_k$ accordingly. We elaborated the process of optimizing Equation (7) with covariance based regularization term in [1]. The optimization of Equation (7) with partial correlation based regularization is elaborated in our previous paper [4].

[1] https://github.com/qingzheli/partial-correlation-based-contrast-pattern-mining/blob/master/PR19_1008supplemental.pdf



(a) Reassign Y (b) Reassign $\hat{Y}$ and Z

Figure 3: E-step: Optimizing the discrete variables

Algorithm 3 E-step: Reassign Latent States for Control Sequence

Require: K, X, θ, γ
Ensure: solution Y
$\{preCst, curCst\} \leftarrow 0^{1\times K}$
$\{prePath, curPath\} \leftarrow$ list of K empty lists
for $t = 1, \cdots, T$ **do**
  for $k = 1, \cdots, K$ **do**
    $minIdx \leftarrow$ index of minimum value of $PrevCost$
    if $preCst[minIdx]+\beta > preCst[k]$ **then**
      $curCst[k] \leftarrow preCst[k]-\ell\ell(X_t, \theta_k)$; $curPath[k] \leftarrow prePath[k]$.append$(k)$
    else
      $curCst[k] \leftarrow preCst[k]+\beta-\ell\ell(X_t, \theta_k)$; $curPath[k] \leftarrow PrevPath[minIdx]$.append$(k)$
    end if
  end for
  $preCst \leftarrow curCst$;
  $prePath \leftarrow curPath$
end for
$lastMinIdx \leftarrow$ index of minimum value of $curCst$; $path \leftarrow currPath[lastMinIdx]$; $Y \leftarrow 0^{T\times K}$
for $t = 1, \cdots, T$ **do**
  $Y_{t,Path[t]} \leftarrow 1$
end for
return Y

5.2. E-step: Optimizing the Discrete Variables

In the E-step, we fix θ_k and $\hat{\theta}_k$ for all $k = 1, \ldots, K$, and optimize the $Y, \hat{Y}$ and Z assignments below:

$$\arg\min_{Y,\hat{Y},Z} \sum\nolimits_{t=2}^{T} \gamma \mathbb{1}(Y_t \neq Y_{t-1}) + \sum\nolimits_{\hat{t}=2}^{\hat{T}} \beta \mathbb{1}(Z_{\hat{t}} \neq Z_{\hat{t}-1}) + \gamma \mathbb{1}(\hat{Y}_{\hat{t}} \neq \hat{Y}_{\hat{t}-1})$$
$$- \sum\nolimits_{t,k}^{T,K} Y_{t,k} \ell\ell(X_t, \theta_k) - \sum\nolimits_{\hat{t},k}^{\hat{T},K} \hat{Y}_{t,k} [Z_{\hat{t}} \ell\ell(\hat{X}_{\hat{t}}, \theta_k) + (1 - Z_{\hat{t}}) \ell\ell(\hat{X}_{\hat{t}}, \hat{\theta}_k)].$$

Optimizing Y: We optimize Y assignment by:

$$\arg\min_Y \sum\nolimits_{t=2}^{\intercal} (\gamma \mathbb{1}(Y_t \neq Y_{t-1})) - \sum\nolimits_{t,k}^{T,K} Y_{t,k} \ell\ell(X_t, \theta_k). \tag{8}$$

The assignment optimization problem the in above equation can be formulated and solved as a classic problem of finding the minimum cost Viterbi path [31] in a fully connected network, as shown in Figure 3a. The t-th layer represents the index t, and the k-th row represents the k-th latent state. The node $J(t, k)$

denotes the cost of assigning $Y_{t,k} = 1$, where $J(t,k) = -\ell\ell(X_t, \theta_k)$. The weights on the edges are the penalties of switching between the latent states. The optimization problem in the E-step is equivalent to finding an optimal path from the first layer to the T-th layer such that the total cost at the edges and the nodes is minimal, which can be solved by dynamic programming in $O(T)$ time (K is a constant parameter). Specifically, for $t = 2, \cdots, T$, the cost of each node in the $(t-1)$-th layer is updated by $J(t,k) \leftarrow \min(J_{\min}(t-1) + \gamma, J(t-1,k)) + J(t,k)$ where $J_{\min}(t)$ is the minimal costs of all nodes in the t-th layer. Finally, the latent assignments Y can be determined by backtracking the path to reach $J_{\min}(T)$. Algorithm 3 elaborates the above process, where the overall time complexity of updating Y is $O(T)$.

Optimizing $\hat{Y}, Z$: We optimize $\hat{Y}$ and Z assignments by:

$$\arg\min_{\hat{Y},Z} \sum_{\hat{t}=2}^{\hat{T}} \gamma \mathbb{1}(\hat{Y}_{\hat{t}} \neq \hat{Y}_{\hat{t}-1}) - \sum_{\hat{t},k}^{\hat{T},K} \hat{Y}_{t,k}[Z_{\hat{t}} \ell\ell(\hat{X}_{\hat{t}}, \theta_k) + (1 - Z_{\hat{t}}) \ell\ell(\hat{X}_{\hat{t}}, \hat{\theta}_k)].$$

The optimization problem in the above equation can also be formulated and solved as finding the minimum cost Viterbi path [31] in a larger fully connected network of size $2K \times \hat{T}$. As shown in Figure 3b, the node $J(\hat{t}, k)$ denotes the cost of assigning $\hat{Y}_{\hat{t},k} = 1$ and $Z_{\hat{t}} = 1$, and node $\hat{J}(\hat{t}, k)$ denotes the cost of assigning $\hat{Y}_{\hat{t},k} = 1$ and $Z_{\hat{t}} = 0$, where $\hat{J}(t,k) = -\ell\ell(\hat{X}_t, \hat{\theta}_k)$ and $J(t,k) = -\ell\ell(\hat{X}_t, \theta_k)$. Specifically, in order to find an optimal path from the first layer to the last layer such that the total costs at edges and nodes is minimal, for each $\hat{t} = 2$ to $\hat{T}$, the cost of each node in the $\hat{t}$-th layer is updated by:

$$\hat{J}(\hat{t}, k) \leftarrow \min(\hat{J}_{\min}(\hat{t}-1) + \gamma, J_{\min}(\hat{t}-1) + \beta + \gamma, \hat{J}(\hat{t}-1, k), J(\hat{t}-1, k) + \beta) + \hat{J}(\hat{t}, k)$$

$$J(\hat{t}, k) \leftarrow \min(J_{\min}(\hat{t}-1) + \gamma, \hat{J}_{\min}(\hat{t}-1) + \beta + \gamma, J(\hat{t}-1, k), \hat{J}(\hat{t}-1, k) + \beta) + J(\hat{t}, k)$$

where $J_{\min}(\hat{t})$ and $\hat{J}_{\min}(\hat{t})$ are the minimal costs to $\hat{t}$-th layer of all J-nodes and all $\hat{J}$-nodes, respectively. Finally, the $\hat{Y}$ and Z assignments can be decided by backtracking the path to reach $J_{\min}(\hat{T})$ and $J_{\min}(\hat{T})$, which ever is smaller. Algorithm 4 elaborates the above process, where the overall time complexity of updating $\hat{Y}$ and Z is $O(\hat{T})$.

Algorithm 4 E-step: Reassign Latent States for Experimental Sequence

Require: $K, \hat{X}, \theta, \hat{\theta}, \beta, \gamma$
Ensure: solution $\hat{Y}, Z$
1: $\{preCst0, preCst1, curCst0, curCst1\} \leftarrow 0^{1\times K}$
2: $\{prePath, curPath\} \leftarrow$ list of $2K$ empty lists
3: **for** $\hat{t} = 1, \cdots, \hat{T}$ **do**
4: **for** $k = 1, \cdots, K$ **do**
5: $minIdx0 \leftarrow$ index of minimum value of $PrevCost0$
6: $minIdx1 \leftarrow$ index of minimum value of $PrevCost1$
7: $minCst0 \leftarrow \min(preCst0[k], preCst0[minIdx0]+\gamma, preCst1[k]+\beta, preCst1[minIdx1]+\beta+\gamma)$
8: **if** $minCst0 \leftarrow preCst0[k]$ **then**
9: $curCst0[k] \leftarrow minCst0 - \ell\ell(\hat{X}_{\hat{t}}, \hat{\theta}_k)$;$curPath[2*k] \leftarrow prePath[2*k]$.append$(2*k)$
10: **else if** $minCst0 = preCst0[minIdx0]+\gamma$ **then**
11: $curCst0[k] \leftarrow minCst0 + \gamma - \ell\ell(\hat{X}_{\hat{t}}, \hat{\theta}_k)$; $curPath[2*k] \leftarrow prePath[2*minIdx0]$.append$(2*k)$
12: **else if** $minCst0 = preCst1[k]+\beta$ **then**
13: $curCst0[k] \leftarrow minCst0+\beta-\ell\ell(\hat{X}_{\hat{t}}, \hat{\theta}_k)$; $curPath[2*k] \leftarrow prePath[2*k+1]$.append$(2*k)$
14: **else**
15: $curCst0[k] \leftarrow minCst0+\beta+\gamma-\ell\ell(\hat{X}_{\hat{t}}, \hat{\theta}_k)$; $curPath[2*k] \leftarrow prePath[2*minIdx1+1]$.append$(2*k)$
16: **end if**
17: $minCst1 \leftarrow \min(preCst1[k], preCst1[minIdx1]+\gamma, preCst0[k]+\beta, preCst0[minIdx0]+\beta+\gamma)$
18: **if** $minCst1 = preCst1[k]$ **then**
19: $curCst1[k] \leftarrow minCst1-\ell\ell(\hat{X}_{\hat{t}}, \theta_k)$; $curPath[2*k+1] \leftarrow prePath[2*k+1]$.append$(2*k+1)$
20: **else if** $minCst1 = preCst1[minIdx1]+\gamma$ **then**
21: $curCst1[k] \leftarrow minCst1+\gamma-\ell\ell(\hat{X}_{\hat{t}}, \theta_k)$; $curPath[2*k+1] \leftarrow prePath[2*minIdx1+1]$.append$(2*k+1)$
22: **else if** $minCst1 = preCst0[k]+\beta$ **then**
23: $curCst1[k] \leftarrow minCst1+\beta-\ell\ell(\hat{X}_{\hat{t}}, \theta_k)$; $curPath[2*k+1] \leftarrow prePath[2*k]$.append$(2*k+1)$
24: **else**
25: $curCst1[k] \leftarrow minCst1+\beta+\gamma-\ell\ell(\hat{X}_{\hat{t}}, \theta_k)$; $curPath[2*k+1] \leftarrow prePath[2*minIdx1]$.append$(2*k+1)$
26: **end if**
27: **end for**
28: $preCst0 \leftarrow curCst0$; $preCst1 \leftarrow curCst1$; $prePath \leftarrow curPath$
29: **end for**
30: $lastMinIdx0 \leftarrow$ index of minimum value of $curCst0$
31: $lastMinIdx1 \leftarrow$ index of minimum value of $curCst1$
32: **if** $curCst0[lastMinIdx0] < curCst1[lastMinIdx1]$ **then**
33: $path \leftarrow currPath[2*lastMinIdx0]$
34: **else**
35: $path \leftarrow currPath[2*lastMinIdx1+1]$
36: **end if**
37: **for** $\hat{t} = 1, \cdots, \hat{T}$ **do**
38: $\hat{Y}_{\hat{t}, Path[\hat{t}]/2} \leftarrow 1$; $Z_{\hat{t}} \leftarrow Path[\hat{t}]\%2$
39: **end for**
40: **return** $\hat{Y}, Z$

5.3. Complexity Analysis

The overall complexity of Algorithm 1 can be considered as $O((T+\hat{T}) + n^2)$, which is analyzed below. After initializing the variables in Lines 1 and 2, the algorithm alternatively performs the M-step in Lines 4-8 and performs the E-step in Lines 9-10. One iteration of our EM-like algorithm consisted of optimizing $Y, \hat{Y}$ and Z assignments in the E-step whose complexity is $O((T+\hat{T}))$, and optimizing θ and $\hat{\theta}$ in the M-step of whose complexity is $O(T+\hat{T})$ for

computing the empirical covariance matrices plus $O((nw)^3)$ for each updating iteration in our ADMM algorithm, which can be reduced to $O((nw)^2)$ when using the gradient decent to avoid computing the matrix inversion. Typically, our ADMM algorithm will satisfy the stopping criteria with a good solution [30] after a few tens of iterations, so the number of iterations in our ADMM algorithm is considered as a constant number. The total number of iterations of our E-M algorithm depends on the data, but typically converges in dozens of iterations and thus can also be considered as a constant number. In most cases, w should be small (e.g.$w < 10$) as explained in section 4.4, which can be considered as the constant. Therefore, so the overall complexity of our algorithm is $O((T + \hat{T}) + n^2)$ in practice.

6. Experiments on Synthetic Datasets

The performance of the proposed Contrast Pattern Mining (CPM) framework is evaluated on 12 datasets with different settings. All the experiments are conducted on a computer with one Intel(R) Xeon(R) CPU and 32 GB memory.

6.1. Experimental Setup

Because of the unsupervised nature and the lack of public available real-world datasets with labels for the contrast pattern mining problem, we evaluate the proposed model on 11 synthetic datasets. Datasets 1-3 are generated with random latent state assignments to evaluate the effectiveness and the sensitivities to the hyper-parameters, and Datasets 4-12 are generated with various sizes to test scalability. The generation process, comparison methods,and evaluation metrics are described in turn.

Evaluation metrics: To compare the effectiveness of the proposed method and other methods, the predicted latent state and CDFD pattern assignments are evaluated with the ground truth labels described above. To ensure a fair comparison of effectiveness among all methods, the number of latent states K in all the methods is fixed to the corresponding K used to generate the datasets.

All methods are evaluated as a clustering problem with K clusters for the latent state assignments by using macro F_1 score defined as the mean F_1 scores of all clusters. The predicted CDFD pattern assignments are evaluated as an anomaly detection problem by using F_1 score defined as the harmonic mean of the precision and recall. The closer the F_1 score to 1, the better the result.

Comparison methods and our methods: To the best of our knowledge, there is no integrated method capable of mining CDFD pattern for CMTS generated from controlled experiments. Therefore, all the comparison methods need to predict the latent state and the CDFD pattern assignments in two steps. In Step 1, the subproblem of determining latent state assignments are equivalent to the time series subsequence clustering problem. We compared with two traditional distance-based clustering methods, including the classic K-means and the state-of-the-art K-shape [8] methods, and two model-based methods, including the classic Gaussian Mixture Models (GMM) [10] and the state-of-the-art TICC [2] methods. For Step 2, the contrast patterns detection problem can be evaluated as an anomaly detection problem. The control time series, which only contain the "normal" data, are used to train the anomaly detection model, and the experimental time series, which contain both "normal" and "abnormal" data, are used to detect the anomalies. Therefore, we consider four anomaly detection methods for the contrast pattern detection problem: one-class support vector machine (1SVM) [32] (with linear kernel), Elliptic Envelope (EE) [16], Isolation Forest (IF) [17], and Local Outlier Factor (LOF) [18]. For all these anomaly detection methods, the default values are used for all parameters except for the "outlier ratio," which is set to 50%. That is the same as the "true" outlier ratio in the synthetic datasets to get relatively good results for the comparison methods. Notice that our method does not need prior knowledge of the outlier ratio. To validate the proposed regularization terms, the baseline method, which only contains our loss function and the temporal regularization term by setting $\lambda = 0$ in Equation (5), is also considered. For our methods, we implemented and solved the proposed Contrast Patter Framework with Covariance feature dependency regularization (i.e., CPM-C method) defined Equation

Table 2: The (macro) F_1 scores of predicted $Y, \hat{Y}$, and Z assignments

Method	Dataset 1			Dataset 2			Dataset 3		
	Y	$\hat{Y}$	Z	Y	$\hat{Y}$	Z	Y	$\hat{Y}$	Z
K-means+1SVM	0.50	0.51	0.58	0.33	0.34	0.61	0.28	0.27	0.60
K-means+EE	0.50	0.51	0.23	0.33	0.34	0.25	0.28	0.27	0.25
K-means+IF	0.50	0.51	0.23	0.33	0.34	0.26	0.28	0.27	0.26
K-means+LOF	0.50	0.51	0.15	0.33	0.34	0.18	0.28	0.27	0.21
K-shape+1SVM	0.51	0.51	0.54	0.34	0.33	0.56	0.26	0.24	0.55
K-shape+EE	0.51	0.51	0.23	0.34	0.33	0.25	0.26	0.24	0.25
K-shape+IF	0.51	0.51	0.24	0.34	0.33	0.26	0.26	0.24	0.25
K-shape+LOF	0.51	0.51	0.14	0.34	0.33	0.19	0.26	0.24	0.21
TICC+1SVM	**0.99**	0.72	0.47	0.29	0.24	0.48	0.25	0.23	0.51
TICC+EE	**0.99**	0.72	0.35	0.29	0.24	0.25	0.25	0.23	0.25
TICC+IF	**0.99**	0.72	0.29	0.29	0.24	0.27	0.25	0.23	0.25
TICC+LOF	**0.99**	0.72	0.30	0.29	0.24	0.20	0.25	0.23	0.25
GMM+1SVM	0.95	0.87	0.49	0.85	**0.80**	0.50	0.83	0.78	0.52
GMM+EE	0.95	0.87	0.22	0.85	**0.80**	0.22	0.83	0.78	0.24
GMM+IF	0.95	0.87	0.23	0.85	**0.80**	0.24	0.83	0.78	0.25
GMM+LOF	0.95	0.87	0.16	0.85	**0.80**	0.18	0.83	0.78	0.21
Baseline ($\lambda=0$)	0.94	0.92	0.89	0.86	0.63	0.88	0.83	0.59	0.76
CPM-C (ours)	**0.99**	**0.99**	**0.98**	**0.99**	0.66	**0.98**	**0.99**	**0.82**	**0.95**
CPM-P (ours)	**0.99**	**0.99**	**0.98**	**0.99**	0.75	**0.89**	**0.99**	**0.99**	**0.98**

(3) and the Partial correlation based transformation function defined in Equation (4) (i.e., the CPM-P method). We set the hyper-parameters in our methods by $\lambda = 1000, \beta = 2, \gamma = 10$. We also provide extensive sensitivity analysis of the hyper-parameters below.

6.2. Performance

In this section, the performance in terms of effectiveness, hyper-parameter sensitivity, and scalability are analyzed in turn.

Effectiveness: In this section, the effectiveness of the comparison methods and the proposed models are evaluated on Dataset 1, Dataset 2, and Dataset 3, which respectively include two, three, and four latent states. The (macro) F_1 scores of the predicted latent state (i.e.,$Y, \hat{Y}$) and CDFD pattern (i.e.,Z) assignments are shown in Table 2. The method named with the "+" sign in its name is a two-step method. For each column, the top-2 best performers are written in bold and the top-1 best performer is also underlined. As the results show, our methods are the top-2 performers in 8 out of 9 (macro) F_1 scores. Our CPM-C and CPM-P methods are respectively 15% and 22% better on average than most the best comparison methods except for the baseline method, which is a special case of our model without the regularization terms. The distance-

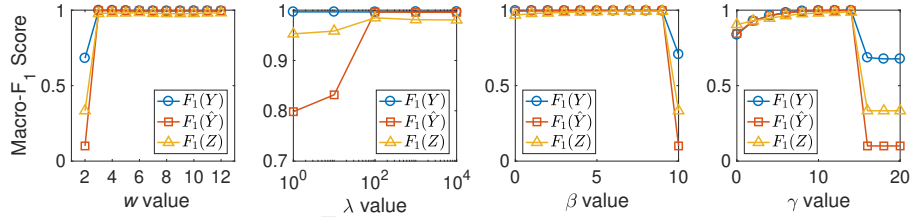
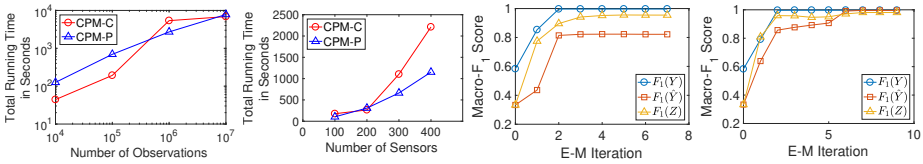


Figure 4: Sensitivity Tests



(a) Running Time vs Observation Count

(b) Running Time vs Sensor Count

(c) Convergence process of CPM-C

(d) Convergence process of CPM-P

Figure 5: Efficiency evaluation

based methods perform worse than the model-based methods, as opposed to the dependency-based patterns in these datasets. After intensively tuning the hyper-parameters, TICC achieves the macro F_1 score of 0.99 on Y assignments in Dataset 1, but still fails in other datasets because Dataset 1 only contains two latent states, which is a relatively simple dataset. For the datasets with more than two latent states, the TICC method with the $L1$ regularization term still suffers from differentiating the latent states and the contrast patterns caused by the intervention. GMM generally performs better than other comparison methods on Y and $\hat{Y}$ assignments. For Z assignments, all comparison methods except for our methods and the baseline method do not perform well with the highest score at 0.60, which is still close to random guessing. Because the contrast patterns highly depend on the latent state assignments, the imperfect results on the latent state assignments lead to worse results for the CDFD pattern assignments. As none of the comparison methods can solve the CDFD pattern mining problem, we will focus on analyzing the results of our method in the rest of this section. The performance of the baseline is better than other comparison methods, which validates the effectiveness of our generative model proposed in Section 4.1, but still worse than our methods, which validates the usefulness of both proposed feature dependency regularization terms.

Parameter sensitivity tests: The sensitivities of the hyper-parameters in CPM-P method are tested on Dataset 3, and the parameter sensitivities in CPM-C method on other datasets follow similar trends. A basic setting of $K = 4, \lambda = 1000, \beta = 2, \gamma = 10, w = 5$ is adjusted by varying a single parameter each time. The results of the sensitivity test are plotted in Figure 4. As shown in Figure 4, our method is insensitive to all the parameters within the range shown, which provides a good hint to set the parameters when applying our method to other datasets. Concretely, the model performs well when the window size w is between 3 and 12. Recall the "true" window size of the datasets is $w = 5$, so when $w = 2$ the macro F-1 scores are relatively low since most of long-term temporal dependencies are absent in X and $\hat{X}$. On the other hand, w should not be too large because the model needs high computational cost to estimate the long-term dependencies that are rare in the CMTS. The value of the regularization parameter λ can be chosen from a very large range of 100 to 10,000, as seen in Figure 4. When the value of λ is too small (i.e.,$\lambda < 100$), the constraint between the contrast dependency networks of the same latent state will be loose. Consequently, the model may lose the ability to ensure these contrast dependency networks are characterizing the same latent states, which also validate the importance of our contrast pattern regularization term. For zero-normalized data, the values of β and γ can be chosen between 1 and 5, and between 5 and 12, respectively. Values greater than the recommended range will cause the model to prefer less changes when assigning the latent states.

Scalability and learning curves: To validate the scalability of the proposed algorithm with respect to the number of observations, four datasets, namely Datasets 4-7, are generated by setting the variable count, sliding window size, and number of the latent states all to 5. The number of observations for Datasets 4-7 ranges from 10^4 to 10^7. The total running time is plotted using a log-log scale in Figure 5a. Our algorithm grows almost linearly over the number of observations, and can optimize CMTS with 10 million data points in less than three hours to be stationary on the assignments. To validate the scalability of the proposed algorithm with respect to the number of sensors, Datasets 8-

11 respectively include 100, 200, 300, and 400 sensors. Each dataset includes 4,000 observations that are generated by setting both the sliding window size and the latent state count to 2. The running time is plotted in Figure 5b. We can see that the proposed algorithm can easily scale to hundreds of sensors. For example, the running time on the dataset with 400 sensors is less than 40 minutes. As we analyzed in Section 5.3, the running time should grow linearly with respect to the number of observations and grow quadratically with respect to the number of sensors. In all, the experimental results illustrated in Figure 5a and 5b are aligned with our time complexity analysis. To valid our algorithm converges very quickly to the optimal solution, we plot the (macro) F_1 scores at each E-M iteration before converging in Figure 5c and 5d, which shows that our algorithm can generate near-optimal results after two E-M iterations.

7. Experiments on Real-world Datasets

We apply our contrast pattern mining framework to a controlled driving behavior experiment, which evaluates the influence of an attention deficit hyperactivity disorder (ADHD) medicine by contrasting driving behaviors before and after taking the medicine on the drivers diagnosed with ADHD [33][1]. In this section, we provide comprehensive case studies for the proposed methods as well as the baseline methods on real-world CMTS.

7.1. Experimental setup

Real-world datasets: Ten drivers diagnosed with ADHD participated this controlled experiment. Each driver was asked to drive twice on the same high-fidelity driving simulator, once before and once after taking the ADHD medicine.The control multivariate time series records a 10-minute driving session at 60 Hz before taking the ADHD medicine, and the experimental multivariate time series records another 10-minute driving session at 60 Hz after taking the medicine. Therefore, the total length of the control and experimental time series is 72,000. The detailed descriptions of this controlled experiment

can be found in [1]. The control factors include the same driver, same driving simulator, and similar driving scenario. The intervention is taking the ADHD medicine. The driving simulator can record the multivariate observation in six dimensions, which are Brake (B), steering Wheel (W), Accelerator (A), Velocity (V), latitude, and longitude. We use the first three dimensions' data (i.e., 3-D time series) to predict the latent states and the contrast patterns, then use the velocity time series and the latitude-longitude time series (i.e., trajectory) to validate the predictions by our method.

Parameter settings: We choose the number of latent states $K = 4$ for these datasets since for any value of $K \geq 4$, the model still assigns most of the points to 4 latent states. The other hyper-parameters are set as follows: $w = 5, \lambda = 1000, \beta = 5, \gamma = 10$. For the baseline methods, the default parameters are used except for TICC which are intensively tuned because TICC always assign the whole data to one and only one latent state with its default parameters.

Evaluation methods: For each proposed model under our contrast pattern mining framework, we first examine the predicted latent state assignments, then examine the identified contrast patterns. Because of the lack of labeled driving behavior data, we are unable to use metric-based evaluation methods. To evaluate the predicted latent states (i.e., the Y and $\hat{Y}$ assignments), we first plot the predicted latent state assignments on previously unseen data such as the velocity time series and the trajectories, then examine whether the predicted latent states match the true driving states on the velocity and trajectory data. In other words, if each one of the predicted latent states can be interpreted as one of the true driving states on the velocity and trajectory data, the effectiveness can be validated. To examine the identified contrast patterns in an interpretable way, we first compute the partial correlation networks from the learned inverse covariance matrices θ and $\hat{\theta}$, then analyze the patterns in these networks via their closeness centrality scores [34]. Specifically, in these partial correlation networks, each node denotes a feature F_i, where $F \in \{W, A, V, B\}$ denotes the variables, and i is the relative index within the subsequences. For example, the feature B_0 denotes the feature corresponding to the brake variable

at the first observation of the subsequence. The edge weight is the absolute value of the partial correlation coefficient of the node connected. Recall that the closeness centrality score reflects the "significance" of the node in the network. The higher the closeness centrality score, the more significant the node is. Because the nodes have different relative significance under different driving states. For example, the brake-related nodes play more significant roles than other nodes under the driving state of "deceleration," and the steering wheel-related nodes play more significant roles under the driving state of "turning." Therefore, two partial correlation networks of the same latent state are expected to have similar closeness scores for the same node, and the intervention is unlikely to significantly change such similarities. On the other hand, two partial correlation networks of the different latent states are expected to have different closeness scores. Finally, we evaluate the effectiveness of the proposed model based on the aforementioned expectations.

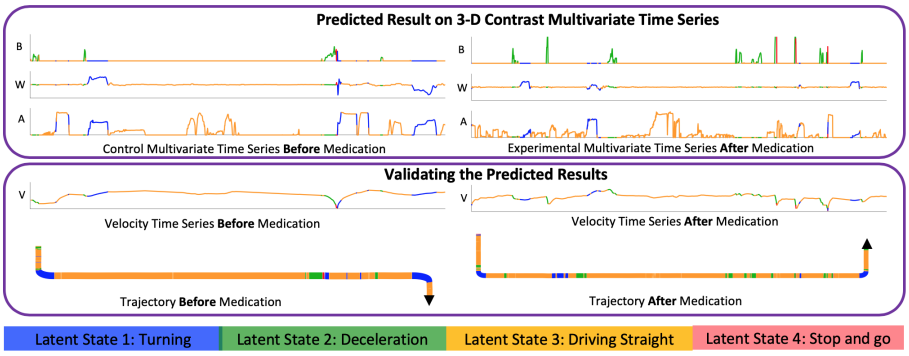
7.2. Performance

In this section, the results predicted by our CPM-C model under the proposed framework are visualized and interpreted. The performance of CPM-P model was elaborated in our previous paper [4]. In addition, the results predicted baselines methods (e.g. TICC, GMM, K-means, and K-shape) are also visualized as the ablation studies.

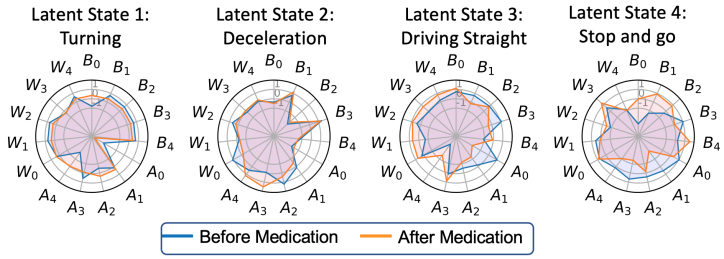
The closeness centrality score of each feature reflects how much the feature affects the other features.

7.2.1. Performance of the CPM-C model

Evaluate the latent state assignments predicted by CPM-C model: The predicted latent state assignments are plotted in Figure 6a. Each of the predicted latent states plotted in the top portion can be validated by the velocity time series and the trajectory plotted in the bottom portion. For example, first of all, all the blue latent states are located at the corners in both trajectories, which belong to the "turning" state. Secondly, the green latent state highly



(a) Validating latent state assignments: The predicted latent states in the multivariate time series are plotted in the top portion. Each predicted latent state plotted can be interpreted as a driving state, which can be validated in the lower portion.



(b) Validating contrast patterns: The contrast patterns are analyzed by the (zero-normalized) closeness centrality scores of all features (e.g., B_0). Under the same latent state, the areas of before and after taking the medication mostly overlap with each other. The non-overlapped areas indicate the driving behaviors have been changed by the medicine.

Figure 6: Applying CPM-C model on ADHD medicine controlled experiment

matches the "deceleration" state as the green segments in the velocity time series mostly decrease. Moreover, the orange segments in the velocity time series mostly increase or remain stable, which can be interpreted as driving straight. Finally, the red latent state corresponds to the state of "stopping and go", which can also be verified from the red segments of the velocity time series, the velocity first decreases to 0 and then increases,

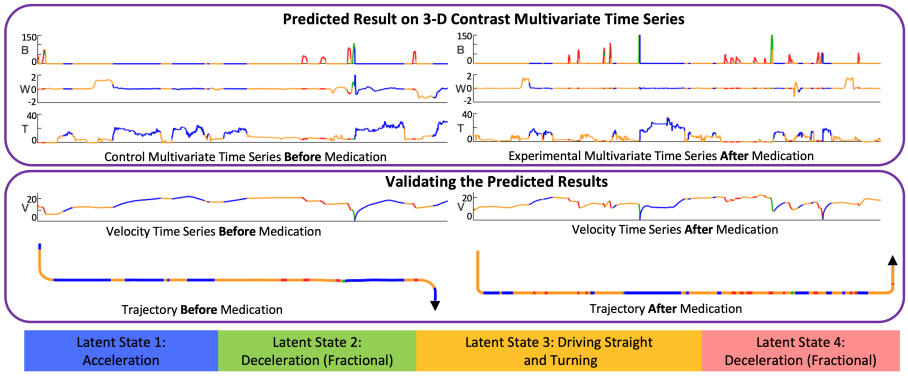
Case studies on the contrast patterns mined by CPM-C model: The zero-normalized closeness centrality scores for each latent state are plotted in Figure 6b. We have three observations: 1) each latent state has a unique "shape" in terms of the relative importance of the features; 2) the areas under the same latent state mostly overlap, which validates the usefulness of the proposed covariance-based regularization that can pair the latent states; and 3) the mined

contrast patterns are highly interpretable through the minor differences of the centrality scores under the same latent state. For example, the plots of the "Turning" latent state suggest that the "brake"-related nodes are less important after the medication, which can be interpreted as this driver being less likely to use the brake pedal while turning the vehicles, which indicates driver turn the vehicles with a stabler velocity comparing to the same driver's behavior before the medication. For another example, the plots of the "Driving Straight" latent state in Figure 6b show that, after taking the medicine, the centrality scores of the steering wheel-related nodes are higher than before taking the medicine. This result suggests this driver was actively turning the steering wheel while driving straight after taking the medicine, which is unsafe driving behavior. This interesting and counter-intuitive contrast pattern has not been mined by other methods before. We find that the unsafe driving behavior may be indirectly caused by the side-effects of the ADHD medicine, which is known that the ADHD medicine could cause the tics on some patients such that they cannot hold the steering wheel stable while driving straight [35].

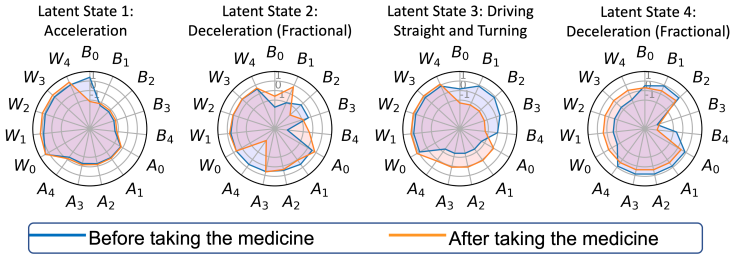
7.2.2. Performance of the comparison methods

In this subsection, we conduct the case studies on the comparison methods. Recall that all the comparison methods require two steps to predict the latent state and contrast pattern assignments, respectively. For latent state assignment in Step 1, we consider all comparison methods, introduced in our synthetic experiments. For contrast pattern assignment in Step 2, the other comparison methods perform much worse than 1SVM in our synthetic experiments, so we only consider 1SVM to avoid redundancy. To further validate the contrast dynamic feature dependency pattern, we obtain the contrast dependency networks through maximum likelihood estimation given the latent state assignments and contrast assignments generated from the previous steps. We use the same visualization way to evaluate the performance of the comparison methods.

The results generated by TICC-based method are plotted in Figure 8. The



(a) Incorrect latent state assignments: The predicted latent states are plotted in different colors. Some predicted latent state contains two or more true driving states (e.g. the orange state). Some latent states overlapping with each other (e.g., the green and red state).



(b) Unreliable contrast patterns: The contrast patterns cannot denote the differences caused by the medicine because the dependency networks under the same latent state do not match the same true driving state for fair comparison.

Figure 7: Applying TICC method on the ADHD medicine controlled experiment

predicted latent states are plotted in different colors in Figure 8(a). Some predicted latent state mistakenly contains two or more true driving states. For example, the orange latent state contains data for both driving straight and turning. From the steering wheel (W) dimension, we can clearly see that some orange part contains both zero and non-zero radian. TICC may mistakenly consider the data of driving straight and the data of turning should generally belong to the same driving state and contain some differences caused by the intervention. On the other hand, some latent states denote the same true driving state because of the intervention. For example, the green and red latent states both denote the deceleration state. The differences between the green and red states are caused by the intervention but TICC failed to differentiate

them. The examples above clearly demonstrate that the TICC approach is not suitable to address the challenge of differentiating the patterns caused by various driving states and those caused by the medicine. In other words, the latent states predicted by TICC cannot map to exact one true driving state, which prevents further fairly contrast the contrast pattern under the same latent state. For example, in Latent State 3 in Figure 8(b), the dependency networks before and after taking the medicine have very different characteristics because the differences are likely to be caused by the different driving state rather than the medicine. Also, one cannot conclude the differences in Latent State 2 is caused by the medicine because Latent State 2 cannot completely denote the driving state of deceleration. Only Latent State 1 seems to be properly predicted. In conclusion, the contrast pattern cannot be properly extracted by TICC based approach because the latent states identified by TICC cannot exclusively denote the true driving states.

The results of GMM, K-means, and K-shape methods are plotted in our supplementary materials in [2]. These methods also suffer from the same challenge as in the case study on TICC but generally perform worse than TICC. For example, none of the latent states predicted by these methods can denote a complete true driving state.

8. Conclusion

In this paper, we propose a novel framework to formulate the contrast pattern mining problem as an optimization problem, which integrates latent state identification, dynamic feature dependency inference, and contrast pattern detection. Extensive experimental evaluations validates its effectiveness, scalability, and robustness. The proposed method is applied to a controlled experiment on the effects of ADHD medicine on driving behaviors, which demonstrates the use case and interpretability of the proposed methods. We should point out

[2] https://github.com/qingzheli/partial-correlation-based-contrast-pattern-mining/blob/master/PR19_1008supplemental.pdf

that the proposed approach focuses on linear dependency patterns, which cannot handle the multivariate time series (MTS) data with complex non-linear dependencies (e.g. MTS generated from functional MRI scan). Therefore, one of the future research direction would be modeling the dynamic non-linear dependency by exploring the deep learning based models such as auto-encoder and recurrent neural networks, which are good at modeling non-linear and dynamic relationships. Also, we only applied the contrast pattern mining framework to the controlled experiments on driving behavior, but the proposed framework is generic to any controlled experiments that not only care about the overall contrast results but also care about the dynamic contrast pattern that occurred during the experiment. To apply the proposed framework to another controlled experiment, the feature dependency regularization term might need to be redesigned, but the general objective and optimization techniques should be similar to the proposed approach.

References

[1] Y.-C. Lee, C. W. McIntosh, et al., Design of an experimental protocol to examine medication non-adherence among young drivers diagnosed with adhd: A driving simulator study, Contemporary clinical trials communications 11 (2018) 149–155.

[2] D. Hallac, S. Vare, S. Boyd, J. Leskovec, Toeplitz inverse covariance-based clustering of multivariate time series data, in: KDD'17, 2017, pp. 215–223.

[3] X. Liu, X. Kong, A. B. Ragin, Unified and contrasting graphical lasso for brain network discovery, in: SIAM'17, 2017, pp. 180–188.

[4] Q. Li, L. Zhao, Y.-C. Lee, Y. Ye, J. Lin, L. Wu, Contrast feature dependency pattern mining for controlled experiments with application to driving behavior, in: ICDM'19, 2019, pp. 1192–1197.

[5] E.Keogh, J.Lin, W.Truppel, Clustering of time series subsequences is meaningless: implications for previous and future research, in: ICDM'03, 2003, pp. 115–122.

[6] T. Rakthanmanon, E. J. Keogh, etal., Mdl-based time series clustering, Knowledge and information systems 33 (2) (2012) 371–399.

[7] V. S. S. Fotso, E. M. Nguifo, P. Vaslin, Frobenius correlation based u-shapelets discovery for time series clustering, Pattern Recognition (2020) 107301.

[8] J. Paparrizos, L. Gravano, k-shape: Efficient and accurate clustering of time series, in: SIGMOD'15, 2015, pp. 1855–1870.

[9] Y. Xiong, D.-Y. Yeung, Time series clustering with arma mixtures, Pattern Recognition 37 (8) (2004) 1675–1689.

[10] J. D.Banfield, A. E.Raftery, Model-based gaussian and non-gaussian clustering, Biometrics 49 (3) (1993) 803–821.

[11] P. Smyth, Clustering sequences with hidden markov models, in: M.C.Mozer, M.I.Jordan, T.Petsche (Eds.), NIPS'97, 1997, pp. 648–654.

[12] L. Li, W. Li, J. Liao, X. Hu, Adaptive state continuity-based sparse inverse covariance clustering for multivariate time series, in: 2019 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), IEEE, 2019, pp. 68–74.

[13] J.Ma, S.Perkins, Time-series novelty detection using one-class support vector machines, in: IJCNN, Vol. 3, 2003, pp. 1741–1745 vol.3.

[14] J. B. Cabrera, L. Lewis, R. K. Mehra, Detection and classification of intrusions and faults using sequences of system calls, SIGMOD (2001) 25–34.

[15] T. Lane, C. E. Brodley, Temporal sequence learning and data reduction for anomaly detection, (TISSEC'99) 2 (3) 295–331.

[16] P. J. Rousseeuw, K. V. Driessen, A fast algorithm for the minimum covariance determinant estimator, Technometrics 41 (3) (1999) 212–223.

[17] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, TKDD 6 (1) (2012) 3.

[18] H.-P. K. Markus M Breunig, etal., Lof: identifying density-based local outliers, in: ACM sigmod record, Vol. 29, 2000, pp. 93–104.

[19] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, Lstm-based encoder-decoder for multi-sensor anomaly detection, arXiv

preprint arXiv:1607.00148 (2016).

[20] C. Zhang, D. Song, et al., A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data, in: AAAI'19, Vol. 33, 2019, pp. 1409–1416.

[21] S. Mauceri, J. Sweeney, J. McDermott, Dissimilarity-based representations for one-class classification on time series, Pattern Recognition 100 (2020) 107122.

[22] U. Rebbapragada, P. Protopapas, etal., Finding anomalous periodic time series, Machine learning 74 (3) (2009) 281–313.

[23] P. Sun, S. Chawla, B. Arunasalam, Mining for outliers in sequential databases, in: SIAM, 2006, pp. 94–105.

[24] X. Wang, J. Lin, N. Patel, M. Braun, Exact variable-length anomaly detection algorithm for univariate and multivariate time series, DMKD 32 (6) (2018) 1806–1844.

[25] E. Keogh, J. Lin, S.-H. Lee, H. Van Herle, Finding the most unusual time series subsequence: algorithms and applications, Knowledge and Information Systems 11 (1) (2007) 1–27.

[26] J. B. Lee, X. Kong, Y. Bao, C. Moore, Identifying deep contrasting networks from time series data: Application to brain network analysis, in: SDM'17, 2017, pp. 543–551.

[27] S. L. Lauritzen, Graphical models, Vol. 17, Clarendon Press, 1996.

[28] D. Hallac, Y. Park, S. Boyd, J. Leskovec, Network inference via the time-varying graphical lasso, in: KDD'17, 2017, pp. 205–213.

[29] G. Schwarz, et al., Estimating the dimension of a model, The annals of statistics 6 (2) (1978) 461–464.

[30] N. P. Stephen Boyd, etal., Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (1) (2011) 1–122.

[31] A. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, Transactions on Information Theory 13 (2) (1967) 260–269.

[32] B. Schölkopf, J. C. Platt, etal., Estimating the support of a high-dimensional distribution, Neural computation 13 (7) (2001) 1443–1471.

[33] D. Barragan, Y.-C. Lee, Pre-crash driving behaviour of individuals with and without adhd, in: 6th International Conference on Driver Distraction and Inattention, 2018.

[34] L. C. Freeman, Centrality in social networks conceptual clarification, Social networks 1 (3) (1978) 215–239.

[35] R. Boorady, Side effects of adhd medication, Child Mind Institute (2014).