Contrast Feature Dependency Pattern Mining for Controlled Experiments with Application to Driving Behavior (Supplementary Material)

I. Model Optimization

In this section, we elaborate our optimization algorithm for the proposed CPM-P model.

The overall objective function defined in Equation (2) 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 partial correlation-based regularization term). Jointly optimizing these variables is prohibitively difficult to be solved by the existing algorithms. To address this challenge and optimize the proposed model, we develop an Expectation Maximization (EM)-like optimization algorithm outlined in Algorithm 1. 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.

Algorithm 1 Overall Algorithm for optimizing CPM-P model

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:       $[\theta_k, \hat{\theta}_k] \leftarrow$ ADMM_solver(λ, Ψ_k, Φ_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** $Y, \hat{Y}, Z, \theta, \hat{\theta}$

A. M-step: Optimizing the Continuous Variables

In the M-step, we fix the $(Y, \hat{Y}, Z)$ assignments and optimize K pairs of θ_k and $\hat{\theta}_k$ in parallel. Therefore, the subproblem for the k-th pair in M-step is:

$$\begin{aligned} \operatorname{argmin}_{\{\theta_k \hat{\theta}_k\} \succ 0} & -\sum_{\hat{t}}^{\hat{T}} \hat{Y}_{\hat{t},k} [Z_{\hat{t}} \ell\ell(\hat{X}_{\hat{t}}, \theta_k) + (1 - Z_{\hat{t}}) \ell\ell(\hat{X}_{\hat{t}}, \hat{\theta}_k)] \\ & -\sum_{t}^{T} Y_{t,k} \ell\ell(X_t, \theta_k) + \lambda \| Q_k - \hat{Q}_k \|_F^2 \end{aligned} \tag{1}$$

$$\text{s.t. } Q_{k,i,j} = -\theta_{k,i,j} / (\theta_{k,i,i} \theta_{k,j,j})^{\frac{1}{2}}, \hat{Q}_{k,i,j} = -\hat{\theta}_{k,i,j} / (\hat{\theta}_{k,i,i} \hat{\theta}_{k,j,j})^{\frac{1}{2}}$$

As Equation (1) 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) [1] framework.

To solve optimization problem (1) , we first rewrite the objective function in an ADMM-friendly form with some new notations. We define Φ_k as all subsequences in X and $\hat{X}$, which are assigned to the k-th latent state without contrast patterns, and we similarly define Ψ_k as the subsequences in $\hat{X}$ assigned to the k-th latent state with contrast patterns. We define $c(\Phi_k)$ and $c(\Psi_k)$ as the count of the subsequences in Φ_k and Ψ_k, and denote $S_k = \sum_{i=1}^{c(\Phi_k)} \Phi_{k,i}^{\intercal} \Phi_{k,i} / c(\Phi_k)$ and $S'_k = \sum_{i=1}^{c(\Psi_k)} \Psi_{k,i}^{\intercal} \Psi_{k,i} / c(\Psi_k)$ as the empirical covariance matrices. Let $\mathbf{Tr}(\cdot)$ be the trace of the matrix. Then the optimization problem (1) becomes:

$$\begin{aligned} & \operatorname{argmin}_{\theta_k, \hat{\theta}_k, Q_k, \hat{Q}_k} \lambda \| Q_k - \hat{Q}_k \|_F^2 \\ & + [c(\Phi_k)(\mathbf{Tr}(S_k \theta_k) - \operatorname{logdet} \theta_k) + c(\Psi_k)(\mathbf{Tr}(S'_k \hat{\theta}_k) - \operatorname{logdet} \hat{\theta}_k)]/2 \\ & \text{s.t. } Q_{k,i,j} = -\theta_{k,i,j} / (\theta_{k,i,i} \theta_{k,j,j})^{\frac{1}{2}}, \hat{Q}_{k,i,j} = -\hat{\theta}_{k,i,j} / (\hat{\theta}_{k,i,i} \hat{\theta}_{k,j,j})^{\frac{1}{2}}. \end{aligned}$$

This equation is technically difficult to solve because the elements in Q_k and $\hat{Q}_k$ are highly coupled with multiple elements in θ_k and $\hat{\theta}_k$, and the Q_k is also coupled with $\hat{Q}_k$. We thus propose an effective algorithm to decouple the variables for this non-convex subproblem. Specifically, we decouple the elements between Q_k(or $\hat{Q}_k$) and θ_k(or $\hat{\theta}_k$) by dynamically optimizing the diagonal elements and the non-diagonal elements in $Q_k(\hat{Q}_k)$. In addition, we decouple Q_k and $\hat{Q}_k$ by adding several auxiliary variables.

$$\begin{aligned} & \operatorname{argmin} \; [c(\Phi_k)(\mathbf{Tr}(S_k \theta_k) - \operatorname{logdet} \theta_k) + c(\Psi_k)(\mathbf{Tr}(S'_k \hat{\theta}_k) - \operatorname{logdet} \hat{\theta}_k)]/2 \\ & + \lambda \| V_k \|_F^2 \qquad \text{s.t. } \theta_k = P_k, \hat{\theta}_k = \hat{P}_k, V_k = Q_k - \hat{Q}_k, \\ & Q_{k,i,j} = -P_{k,i,j} / (P_{k,i,i} P_{k,j,j})^{\frac{1}{2}}, \text{ and } \hat{Q}_{k,i,j} = -\hat{P}_{k,i,j} / (\hat{P}_{k,i,i} \hat{P}_{k,j,j})^{\frac{1}{2}} \end{aligned}$$

The augmented Lagrangian in scaled form is:

$$\begin{aligned} L_\alpha = & [c(\Phi_k)(\mathbf{Tr}(S_k \theta_k) - \operatorname{logdet} \theta_k) + c(\Psi_k)(\mathbf{Tr}(S'_k \hat{\theta}_k) - \operatorname{logdet} \hat{\theta}_k)]/2 \\ & + \lambda \| V_k \|_F^2 + \frac{\alpha}{2} \Big[\| \theta_k - P_k + U_k^{(1)} \|_F^2 + \| \hat{\theta}_k - \hat{P}_k + U_k^{(2)} \|_F^2 + \\ & \sum_{i,j}^{nw,nw} (Q_{k,i,j} + \frac{P_{k,i,j}}{(P_{k,i,i} P_{k,j,j})^{\frac{1}{2}}} + U_{k,i,j}^{(3)})^2 + (\hat{Q}_{k,i,j} + \frac{\hat{P}_{k,i,j}}{(\hat{P}_{k,i,i} \hat{P}_{k,j,j})^{\frac{1}{2}}} + U_{k,,i,j}^{(4)})^2 + \\ & \| V_k - Q_k + \hat{Q}_k + U_k^{(5)} \|_F^2 - \| U_k^{(1)} \|_F^2 - \| U_k^{(2)} \|_F^2 - \| U_k^{(3)} \|_F^2 - \| U_k^{(4)} \|_F^2 - \| U_k^{(5)} \|_F^2 \Big], \end{aligned}$$

where $\alpha > 0$ is the penalty parameter of ADMM typically initialized to 1; $P_k, \hat{P}_k, Q_k, \hat{Q}_k$ are auxiliary variables; and $U_k^{(1)}, \cdots, U_k^{(5)}$ are the scaled dual variables. The optimization algorithm based on ADMM is outlined in Algorithm 2, where each variable is optimized iteratively, by fixing other variables, until the stopping criteria is satisfied as specified in [1] (§3.3).

Update θ_k: The subproblem of θ_k-update is as follows:

$$\operatorname{argmin}_{\theta_k \succ 0} \; c(\Phi_k)[\mathbf{Tr}(S_k \theta_k) - \log \det \theta_k]/2 + \frac{\alpha}{2} \| \theta_k - P_k + U_k^{(1)} \|_F^2.$$

The solution is $\theta_k = D_k \Delta D_k^{\intercal}$, where Δ_k is a diagonal matrix

Algorithm 2 M-step: Optimize continuous variables by ADMM

Require: Φ_k, Ψ_k, λ
Ensure: solution $\theta_k, \hat{\theta}_k$

```
{θ_k, θ̂_k, P_k, P̂_k, Q_k, Q̂_k, V_k} ← 0^{nw×nw}
repeat
    Update θ_k ← D_k Δ D_k^T
    Update θ̂_k ← D̂_k Δ̂_k D̂_k^T
    Update P_k by Equation (3)
    Update P̂_k by Equation (4)
    Update Q_k by Equation (5)
    Update Q̂_k by Equation (6)
    Update V_k by Equation (7)
    Update the dual variables by Equation (8)
until the stopping criteria is satisfied [1] (Chapter 3.3)
return θ_k, θ̂_k
```

whose i-th element $\Delta_{k,i,i} = (\Lambda_{k,i,i} + \sqrt{\Lambda^2_{k,i,i} + 8\alpha c(\Phi_k)})/4\alpha$, and $D_k \Lambda_k D_k^\intercal$ is the eigendecomposition of $2\alpha(P_k - U_k^{(1)}) - c(\Phi_k) S_k$.
Update $\hat{\theta}_k$: The subproblem of $\hat{\theta}_k$-update can be solved similarly to θ_k-update. The solution is $\hat{\theta}_k = \hat{D}_k \hat{\Delta}_k \hat{D}_k^\intercal$, where $\hat{\Delta}_k$ is a diagonal matrix whose i-th element $\hat{\Delta}_{k,i,i} = (\hat{\Lambda}_{k,i,i} + \sqrt{\hat{\Lambda}^2_{k,i,i} + 8\alpha c(\Psi_k)})/4\alpha$, and $\hat{D}_k \hat{\Lambda} D_k^\intercal$ is the eigendecomposition of $2\alpha(\hat{P}_k - U_k^{(2)}) - c(\Psi_k) S'_k$.
Update P_k and $\hat{P}_k$: Updating P_k and $\hat{P}_k$ follow the same routine, so we only discuss the solution to P_k. The subproblem of P_k-update is:

$$\begin{aligned} \text{argmin}_{P_k} \ & \frac{\alpha}{2}\|\theta_k - P_k + U_k^{(1)}\|_F^2 + \frac{\alpha}{2}\|\hat{\theta}_k - \hat{Q} + U_k^{(2)}\|_F^2 \\ & + \sum\nolimits_{i,j}^{nw,nw} \frac{\alpha}{2}[Q_{k,i,j} + P_{k,i,j}/(P_{k,i,i}P_{k,j,j})^{1/2} + U_{k,i,j}^{(3)}]^2. \end{aligned} \tag{2}$$

This problem is extremely difficult to solve since the elements in P_k are mutually and nonlinearly coupled with each other in the last term. To address this, we innovatively propose a new method based on "divide-and-conquer." Specifically, we notice that all the non-diagonal elements' values depend on the diagonal ones (i.e. $P_{k,i,i}$), which hence can be first calculated with a closed form solution. Then we consider the diagonal elements as constants, and the non-diagonal elements (i.e., $P_{k,i,j}, i \neq j$) are updated by:

$$P_{k,i,i} \leftarrow \theta_{k,i,i} + U_{k,i,i}^{(1)};\ P_{k,i,j} \leftarrow \theta_{k,i,j} + U_{k,i,i}^{(1)} - d_k(Q_{k,i,j} + U_{k,i,j}^{(3)})/(1 + d_k^2), \tag{3}$$

where $d_k = (P_{k,i,i}P_{k,j,j})^{-1/2}$. Following the same routine for $\hat{P}_k$-update, the solutions are as follows:

$$\hat{P}_{k,i,i} \leftarrow \hat{\theta}_{k,i,i} + U_{k,i,i}^{(2)};\ \hat{P}_{k,i,j} \leftarrow \hat{\theta}_{k,i,j} + U_{k,i,i}^{(2)} - \hat{d}_k(\hat{Q}_{k,i,j} + U_{k,i,j}^{(4)})/(1 + \hat{d}_k^2), \tag{4}$$

where $i \neq j$ and $\hat{d}_k = (\hat{P}_{k,i,i}\hat{P}_{k,j,j})^{-1/2}$.
Update Q_k and $\hat{Q}_k$: Updating Q_k and $\hat{Q}_k$ follow the same routine, so we only elaborate that for Q_k here:

$$\text{argmin}_{Q_k} \ \frac{\alpha}{2}\|Q_k - \Pi_k + U_k^{(3)}\|_F^2 + \frac{\alpha}{2}\|V_k - Q_k + \hat{Q}_k + U_k^{(5)}\|_F^2,$$

where $\Pi_{k,i,j} = -P_{k,i,j}/(P_{k,i,i}P_{k,j,j})^{1/2}$. The solution is:

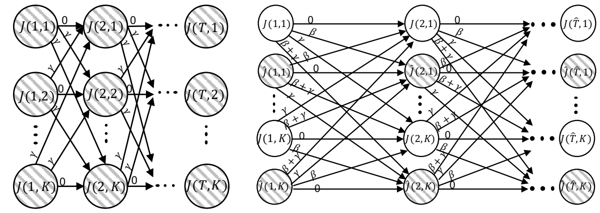
$$Q_k \leftarrow (V_k + \hat{Q}_k + \Pi_k - U_k^{(3)} + U_k^{(5)})/2. \tag{5}$$

Similarly, the solution for $\hat{Q}_k$ is as follows:

$$\hat{Q}_k \leftarrow (-V_k + Q_k + \hat{\Pi}_k - U_k^{(4)} - U_k^{(5)})/2, \tag{6}$$

where $\hat{\Pi}_{k,i,j} = -\hat{P}_{k,i,j}/(\hat{P}_{k,i,i}\hat{P}_{k,j,j})^{1/2}$.
Update V_k: The subproblem of V_k-update is as follows:

$$\text{argmin}_{V_k} \ \lambda\|V_k\|_F^2 + \frac{\alpha}{2}\|V_k - Q_k + \hat{Q}_k + U_k^{(5)}\|_F^2. \tag{7}$$



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

Fig. 1: E-step: Optimizing the discrete variables

It has a closed form solution: $V_k \leftarrow \alpha(Q_k - \hat{Q}_k - U_k^{(5)})/(2\lambda + \alpha)$.
Update the dual variables $U_k^{(1)}, U_k^{(2)}, U_k^{(3)}, U_k^{(4)}, U_k^{(5)}$: The dual variables are updated as follows:

$$\begin{aligned} & U_k^{(1)} \leftarrow U_k^{(1)} + \theta_k - P_k;\ U_k^{(2)} \leftarrow U_k^{(2)} + \hat{\theta}_k - \hat{P}_k;\ U_k^{(3)} \leftarrow U_k^{(3)} + \Pi_k - Q_k; \\ & U_k^{(4)} \leftarrow U_k^{(4)} + \hat{\Pi}_k - \hat{Q}_k;\ U_k^{(5)} \leftarrow U_k^{(5)} + V_k - Q_k + \hat{Q}_k. \end{aligned} \tag{8}$$

B. 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:

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

Optimize Y: We optimize Y assignment by:

$$\text{argmin}_Y \sum\nolimits_{t=2}^{\intercal} (\gamma \mathbb{1}(Y_t \neq Y_{t-1})) - \sum\nolimits_{t,k}^{T,K} Y_{t,k} \ell\ell(X_t, \theta_k). \tag{9}$$

The assignment optimization problem the in above equation can be formulated and solved as a classic problem of finding the minimum cost Viterbi path [2] in a fully connected network, as shown in Figure 1a. 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 $J_{\min}(T)$.
Optimize $\hat{Y}, Z$: We optimize $\hat{Y}$ and Z assignments by:

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

The optimization problem in the above equation can also be formulated and solved as finding the minimum cost Viterbi path [2] in a larger fully connected network of size $2K \times \hat{T}$. As shown in Figure 1b, 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.

The synthetic datasets are generated in 3 steps:

II. Generating the Synthetic Datasets:

The synthetic datasets are generated in 3 steps:

Step 1: Generating the ground truth latent state and CDFD pattern assignments. Given the latent state count $K \in \{2, 3, 4, 5\}$, we first generate $2K$ segments by repeating each element in the list of $[1, 1, 2, 2, \cdots, K, K]$ 500 times. Then the ground truth latent state assignments are generated by randomly concatenating these $2K$ variables. We repeat the above process twice to generate the ground truth latent state assignments for the control time series (i.e.,Y) and the experimental time series (i.e.,$\hat{Y}$). To generate the ground truth CDFD pattern assignments, we randomly selected 250 consecutive points in each segment as the position with the CDFD pattern (i.e.,Z).

Step 2: Generating the inverse covariance matrices. In order to maintain the fairness of the performance comparison, we follow the process described in TICC [3], which enforces the positive definite and block Toeplitz constraints on the inverse covariance matrices. To generate $\{\hat{\theta}_k\}_k^K$, we multiply a random value for each element in θ_k and then adjust the values to maintain the positive definite and block Toeplitz constraints.

Step 3: Generating the CMTS data. We incrementally generate the multivarite observations to build the time series. The observation assigned to the k-th latent state in control time series is sampled from the conditional multivariate Gaussian distribution [4] parameterized by θ_k, where the condition is the previous $w-1$ observations. Similarly, each observation assigned to the k-th latent state in the experimental time series is sampled from the conditional Gaussian distribution, which is parameterized by θ_k if this observation is labeled *without* a CDFD pattern in the ground truth assignments, or parameterized by $\hat{\theta}_k$ if this observation is labeled *with* a CDFD pattern.

References

[1] N. P. Stephen Boyd and et.al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, pp. 1–122, 2011.

[2] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *Transactions on Information Theory*, pp. 260–269, 1967.

[3] D. Hallac, S. Vare, S. Boyd, and J. Leskovec, "Toeplitz inverse covariance-based clustering of multivariate time series data," ser. KDD, 2017, pp. 215–223.

[4] K. I. Park and Park, *Fundamentals of Probability and Stochastic Processes with Applications to Communications*. Springer, 2018.