# A Sparsification Approach for Temporal Graphical Model Decomposition

Ning Ruan[†] Ruoming Jin[†] Victor E. Lee[†]
[†] *Department of Computer Science*
*Kent State University, Kent, OH 44242*
{*nruan,jin,vlee*}*@cs.kent.edu*

Kun Huang[‡]
[‡] *Department of Biomedical Informatics*
*Ohio State University, Columbus, OH 43210*
*khuang@bmi.osu.edu*

***Abstract*—Temporal causal modeling can be used to recover the causal structure among a group of relevant time series variables. Several methods have been developed to explicitly construct temporal causal graphical models. However, how to best understand and conceptualize these complicated causal relationships is still an open problem. In this paper, we propose a decomposition approach to simplify the temporal graphical model. Our method clusters time series variables into groups such that strong interactions appear among the variables within each group and weak (or no) interactions exist for cross-group variable pairs. Specifically, we formulate the clustering problem for temporal graphical models as a regression-coefficient sparsification problem and define an interesting objective function which balances the model prediction power and its cluster structure. We introduce an iterative optimization approach utilizing the Quasi-Newton method and generalized ridge regression to minimize the objective function and to produce a clustered temporal graphical model. We also present a novel optimization procedure utilizing a graph theoretical tool based on the maximum weight independent set problem to speed up the Quasi-Newton method for a large number of variables. Finally, our detailed experimental study on both synthetic and real datasets demonstrates the effectiveness of our methods.**

***Keywords*-temporal graphical model decomposition; Quasi-Newton method; generalized ridge regression; maximum weight independent set;**

## I. Introduction

Causality modeling in time series data has drawn much research attention of late. Given a set of interacting time series, how can we determine if the history of one time series affects the development of another variable? This question about causality plays a fundamental role in economics, health and medical sciences, biology, and the decision-making process, at various domains and levels. For instance, economists are interested in if the domestic demand is a causal factor for China's economic growth [1]; financial analysts want to determine if a company's stock price is causally affected by its inventory turnover ratio [2], and neurobiologists try to understand if the time series of one brain region is a causal factor of another region [3].

Temporal causal modeling tries to recover the causal structure among a group of relevant time series variables [2]. A fundamental tool for such inference is the notion of Granger causality [4]. It is derived from the intuition that if one time series is the cause of another time series, then the former can help improve the prediction accuracy of the latter significantly. More specifically, to determine if time series $x$ is Granger-causal for $y$, we test if the auto-regressive model for $y$ using the past values of both $x$ and $y$ is statistically significantly more accurate than the model using only $y$'s own past value. The notion of Granger causality has been combined with graphical models to study the interaction between multiple time series [5]. A *temporal graphical model* is a directed graph where each vertex corresponds to a time series and each edge indicates a direct causality from the starting vertex to the end vertex. The regression coefficient from the auto-regressive model can be assigned to each corresponding edge to indicate the degree of causation or interaction.

Several methods have been developed to explicitly reconstruct temporal causal models [2], [6]. These works typically target a small number of time series variables (on the order of tens). Recently, temporal causal modeling has been extended to study relatively large complex systems, whose dynamics are captured through a set of time series. Typically, each time series measures a basic unit in the system. Basic units may interact with each other for a certain period of time, and their possible interaction relationships can be summarized through a so-called complex network topology [7]. For instance, in biology, the protein-protein interaction network specifies which two proteins may interact with each other, where the activity of each protein can be measured by the gene-expression time series profiles. Given this, time series $x$ can affect time series $y$ only if an edge $(x, y)$ links from $x$ to $y$ in the network. The sparse underlying network topology thus allows efficient computational procedures to recover the causal structure for a large number of time series.

However, a difficult problem naturally arises as we are able to construct more and more causal graphical models: how can we better understand and conceptualize these complicated causal relationships? Indeed, a causality model with only 20 variables can be overwhelming and difficult to interpret at a global level [2]. Clearly, comprehending a much larger causal model with hundreds or even thousands of variables is even more daunting and elusive. *Can we simplify the temporal causal graphical model to get a better global view of the interactions among a set of relevant time series?* This is the central problem we address in the present work.

To simplify the causal graphical model, we investigate a decomposition approach to cluster time series into groups such that strong interactions appear among the variables within each group and weak (or no) interactions exist for cross-group variable pairs. Clearly, this goal is also consistent with the model for complex systems, which tend to be composed of several smaller and relatively independent components. A key thrust here is that the decomposition model is achieved through balancing the prediction power of the causality model with the simplicity of the model. Specifically, the model simplicity is described in terms of both the sparsification of the regression coefficient matrix and an explicit cluster structure of the graphical model. From a different perspective, our approach can also be viewed as a method for clustering a set of interacting time series. What differentiates our work from the existing work on time series clustering [8], [9], [10], [11] is our clustering criteria, which is derived from temporal graphical modeling and is very challenging to optimize.

In this work, we present a novel and efficient decomposition scheme for a temporal graphical model to cluster a set of interacting time series, with the following contributions:

1. We formulate the clustering problem for temporal graphical models as a regression coefficient sparsification problem and define an interesting objective function which balances model prediction power with its cluster structure.

2. We propose an iterative optimization approach utilizing the Quasi-Newton method and generalized ridge regression to minimize the objective function and to produce a clustered temporal graphical model.

3. We develop a novel optimization procedure using a graph theoretical tool based on the *maximum weight independent set* problem to speed up the Quasi-Newton method for a large number of variables.

4. We performed a detailed experimental study on synthetic and real datasets to demonstrate the effectiveness and efficiency of our approach.

## II. Preliminaries and Problem Definition

### A. Temporal Graphical Modeling

In the following, we give an overview of temporal graphical modeling for the cause-effect relationships of multivariate time series. Let $X_i = [x_i^0, x_i^1, \cdots, x_i^L]$ be the $i$-th time series from time point 0 to the end point $L$. Let $X(j) = [x_1^j, x_2^j, \cdots, x_N^j]'$ be the snapshot vector for the value of each time series at time point $j$. Let $X = [X_1, X_2, \cdots, X_N]' = [X(0), X(1), \cdots, X(L)]$ be the matrix for all $N$ time series, where each row ($X_i$) corresponds to a time series and each column ($X(j)$) corresponds to all time series at time point $j$.

In time series analysis, inference about cause-effect relationships is commonly based on the concept of Granger causality [4], which is defined in terms of predictability and exploits the direction of the flow of time to achieve a causal ordering of dependent variables. Simply speaking, given two time series $X_i$ and $X_j$, Granger causality tests if time series $X_i$ at time point $t+1$, $X_i^{t+1}$, can be better predicted if we consider both time series $X_i$ and $X_j$ from time $t-u$ to $t$ than if we only consider the time series $X_i$ itself. When the Granger test is restricted to revealing linear relationships among different variables, it is closely related to the linear vector autoregressive (VAR) model. Let $X_V$ be the submatrix of $X$ which contains only the time series of information set $V = \{v_1, \ldots, v_{|V|}\}$. We formally represent time series $X_j$ in a VAR model as follows:

$$X_j(t) = \sum_{u=1}^{T} \sum_{v \in V} \phi_{jv}(t-u) x_v^{(t-u)} + \varepsilon(t),$$

where each $\phi_{jv}(t-u)$ is the coefficient indicating the causal influence from $X_v$ to $X_j$, and $\{\varepsilon(t), t \in \mathbb{Z}\}$ is a white noise process with non-singular covariance matrix $\Sigma$. In this sense, we say $X_i$ is Granger-non-causal for $X_j$ with respect to $X_V$ if and only if $\phi_{ji}(t-u) = 0$ for all $1 \le u \le T$.

The *path diagram* proposed by Eichler [5] combines the notion of (multivariate) Granger causality with a graphical model, thus forming the basis of temporal graphical modeling. A path diagram is a directed graph $G = (V, E)$, such that each vertex represents a time series and each edge $(v, v')$ exists if and only if $v$ is a causal factor to $v'$. Path diagrams aid in visualizing the causal relationships among different variables. To construct a path diagram, for each vertex (or variable in time series data), we identify which neighbors are Granger causal for it. Efficient algorithms introduced in [2], [7] are able to construct the path diagram and recover the causal relationships. Next, using the path diagram concept, we formally define our problem of studying the global view of interactions among multivariate time series.

### B. Problem Definition

Our goal is to decompose a temporal graphical model into clusters of interacting time series. Our input includes a set of time series and the path diagram, $G = (V, E)$, which indicates the potential causal relationship between any two time series. By selectively removing links of low importance, we seek to break the path diagram into disconnected components. Then, each time series variable is affected by (or interacts with) time series inside its own component but not between components. Dropping the cross-group causal factor should result in a minimal loss of prediction accuracy.

To formalize this requirement, we utilize the vector autoregressive model. The clustering structure of the temporal graphical model is represented as the regression coefficient matrix $\Phi(u)$ having a *block diagonal* structure:

$$\Phi(u) = \begin{pmatrix} \Phi_1(u) & 0 \cdots & \cdots & 0 \\ 0 & \Phi_2(u) & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \Phi_K(u) \end{pmatrix} \quad (1)$$

For any vertex pair $i$,$j$ belonging to different clusters, $\Phi(u)_{ij} = 0$, for any $u$. Each $\Phi_k(u)$ is a square matrix. Since

the regression coefficient matrices are used to simplify the path diagram, for each $\Phi_k(u)_{ij} \neq 0$, we need $(i,j) \in E$. We do not introduce any new causal relationship in the VAR model besides those in the path diagram $G$. Formally, we define the *clustered regression coefficient matrices* as follows.

*Definition 1:* **(Clustered Regression Coefficient Matrices)** Let $X(t)$, $1 \leq t \leq L$, be the $N$ time series and $G = (V, E)$ be its path diagram for the causal modeling. Let $f$ be the clustering assignment function, i.e., for each vertex $i$, $f(i)$ is its cluster ID. A clustering coefficient matrix $\Phi(u)$ is referred to as a *clustered regression coefficient matrix* if it satisfies the following two properties: 1) for any vertices $i$ and $j$, $f(i) \neq f(j) \Rightarrow \Phi(u)_{ij} = 0$; and 2) for any vertices $i$ and $j$, $\Phi(u)_{ij} \neq 0 \Rightarrow f(i) = f(j)$ and $(i,j) \in E$, i.e., vertex $i$ potentially is a causal factor of $j$ in the path diagram.

Basically we require the clustered regression coefficient matrices to comply with the causal prediction described by the path diagram. Note that in general, we can require them to comply with other known knowledge of such causal prediction. For instance, in analysis of complex systems, we may replace the path diagram with the underlying interaction relationships (the so-called complex network).

To maximize the predictive accuracy while minimizing its representation cost, we define a *cost* function for $\Phi(u)$ which is the sum of all residuals (regression errors) plus a regularization penalty:

$$cost = \sum_{t=1}^{L} ||X(t) - \sum_{u=1}^{T} \Phi(u)X(t-u)||^2 + \alpha(\sum ||\Phi(u)||^2)$$

where $\sum ||\Phi(u)||^2$ is the $L_2$ penalty for the regression coefficient, and $\alpha$ is the complexity parameter that controls the amount of shrinkage.

Clearly, we would like to minimize the prediction error (*cost*). In other words, we seek the coefficient matrices for the desired clustering $f$ which minimize the clustering cost. However, one problem with this criteria is that if we do not constrain the clustering assignment function, the minimum cost configuration tends to group all the vertices in one cluster and leave other clusters empty. To tackle this problem, we apply a constraint to balance the size of clusters.

To achieve this, we introduce a cluster membership matrix $C$ with $N$ rows and $K$ columns, where each row corresponds to a vertex and each column corresponds to a cluster. Each entry $C_{ik}$ acts as an indicator variable: $C_{ik} = 1$ means vertex $i$ belongs to cluster $C_k$, and $C_{ik} = 0$ means vertex $i$ does not belong to cluster $C_k$. In addition, we have $\sum_{k=1}^{K} C_{ik} = 1$.

Utilizing the cluster membership matrix, we can rewrite our optimization problem. For simplicity, we only consider the time window $T = 1$ here ($\Phi = \Phi(1)$). Our framework and algorithm can easily be generalized to $T > 1$.

*Definition 2:* **(Optimal Decomposition Problem)** The optimal decomposition is to find a cluster membership matrix $C$ and its corresponding regression coefficient matrix $\Phi$, such that

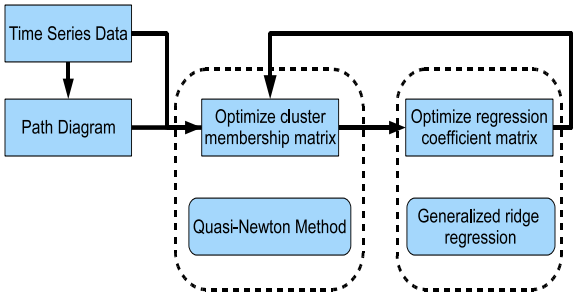



Figure 1. Overview of Algorithm

$$cost = \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{t=1}^{L}(C_{ik}x_i(t) - \sum_{j=1}^{N}\phi_{ij}x_j(t-1)C_{ik}C_{jk})^2 + \alpha(\sum_{i=1}^{N}\sum_{j=1}^{N}\phi_{ij}^2) + \beta\sum_{k=1}^{K}(\sum_{i=1}^{N}C_{ik})^2 \quad (2)$$

is minimized where $C_{ik} \in \{0,1\} \wedge \sum_{k=1}^{K} C_{ik} = 1$.

Note that the last term $\beta\sum_{k=1}^{K}(\sum_{i=1}^{N} C_{ik})^2$ is our size constraint for balancing the size of clusters. It is not hard to see that $\sum_{k=1}^{K}(\sum_{i=1}^{N} C_{ik})^2$ is minimized if and only if $\sum_{i=1}^{N} C_{ik}$ for every k are equal. That is, $\sum_{k=1}^{K}(\sum_{i=1}^{N} C_{ik})^2$ serves as a normalized clusters' size factor for the cost function.

By using the cluster membership matrix in the cost formula, we cause the regression coefficient matrix $\Phi$ to be sparse. This is because the clustering coefficient $\phi_{ij}$ is only useful when $C_{ik} = C_{jk} = 1$, i.e., both time series $i$ and $j$ belong to the same cluster. We can see that the decomposition problem is a combined integer (binary membership matrix) and numerical (regression coefficients) optimization problem. This problem is quite challenging as it contains a large number of $(N^2 + NK)$ unknown variables, where the cluster membership matrix contains $NK$ unknown variables and the regression coefficient matrix has $N^2$ unknown variables.

## III. An Iterative Optimization Procedure

Our solution to the optimal decomposition problem employs the relaxation strategy, which generalizes the binary membership matrix $C$ to be a probabilistic membership matrix. For each time series $i$, we relax the membership entry $C_{ik}$ to be the probability of time series $i$ in cluster $k$, i.e., $C_{ik} = p(k|i)$, ($0 \leq p(k|i) \leq 1$ and $\sum_k p(k|i) = 1$). This relaxation allows us to treat both clustering and regression numerically.

Specifically, our optimization procedure will optimize the clustering membership matrix and regression coefficient matrix in an alternating and iterative fashion (as illustrated in Figure 1). To begin with, we apply an efficient algorithm developed in [7] to extract the path-diagram from the provided time series data. Given this, two optimization steps

are iteratively employed to improve our objective function until *cost* reaches a local minimum. In the first step, we seek the optimal probabilistic membership matrix $[p(k|i)]$ where the regression coefficient matrix $\Phi = [\phi_{ij}]$ is fixed. The traditional Quasi-Newton method can be used to handle it. In the second step, we optimize the regression coefficient matrix assuming that $[p(k|i)]$ is given. We formulate this problem as a generalized ridge regression problem and solve it using existing approaches. Next, we describe these two steps in detail.

**Step 1: Optimizing Probabilistic Membership Matrix**. In this step, we assume the regression coefficient matrix is given and try to optimize the probabilistic membership matrix to minimize the *cost*.

First, we incorporate constraints into the cost formula using the Lagrange multiplier method:

$$F = cost + \sum_{i=1}^{N} \lambda_i (\sum_{k=1}^{K} p(k|i) - 1)$$

where $\lambda_i$ is Lagrange multiplier for membership constraint $\sum_{k=1}^{K} p(k|i) = 1$. Then, we compute its derivatives with respect to each entry $p(r|s)$ of the membership matrix as follows:

$$\frac{\partial F}{\partial p(r|s)} = \sum_{t=1}^{L} (x_s(t) - \sum_{j=1}^{N} \phi_{sj} x_j(t-1) p(r|j))^2$$
$$-2 \sum_{i=1}^{N} p(r|i) \phi_{is} \sum_{t=1}^{L} x_s(t-1)(x_i(t) - \sum_{j=1}^{N} \phi_{ij} x_j(t-1) p(r|j))$$
$$+2\beta \sum_{i=1}^{N} p(r|i) + \lambda_s$$

where $p(r|s)$ is the probability of vertex $s$ being in cluster $r$.

It is hard to get a closed form for each optimal $p(r|s)$ as there is no easy way to solve a set of quadratic equations ($\frac{\partial F}{\partial p(r|s)} = 0$). The classical Newton method can handle this type of optimization problem. Let $X$ be the vector of variables (i.e. vector $\{p(k|i), \lambda_i\}$ with $NK + N$ elements in our problem). The typical iterative update scheme is expressed via gradient $\nabla f(X^{(n)})$ as follows:

$$X^{(n+1)} = X^{(n)} - [H(X^{(n)})]^{-1} \nabla f(X^{(n)}) \qquad (3)$$

where $X^{(n)}$ is the estimated value of $X$ in the $n$-th iteration, and $H(X)$ is the *Hessian matrix*. Specially, for our optimization problem, $H(X)$ is a $(NK+N) \times (NK+N)$ square matrix.

Clearly, it is too costly to evaluate this Hessian matrix, even if $N$ and $K$ are not very large. To deal with this problem, we employ the Quasi-Newton method [12], which seeks to approximate the Hessian matrix, by avoiding the direct inversion of the Hessian matrix. In this method, we focus on solving the following linear system:

$$H^{(n)}(X^{(n+1)} - X^{(n)}) = \nabla f(X^{(n)})$$

If we substitute $X$ with appropriate variables, we can express the linear system of our problem as:

$$H^{(n)} \begin{pmatrix} C^{(n+1)} - C^{(n)} \\ \lambda^{(n+1)} - \lambda^{(n)} \end{pmatrix} = \nabla F(C^{(n)}, \lambda^{(n)})$$

Given this, we can apply another formula, such as the Davidon-Fletcher-Powell (DFP) formula [12], to iteratively update and approximate the Hessian matrix. Thus, the Quasi-Newton method can help construct the probabilistic membership matrix which results in a local minimum of *cost*.

**Step 2: Optimizing Regression Coefficient Matrix**. In the second step, we assume the probabilistic membership matrix is fixed and try to optimize regression coefficient matrix $\Phi$ in order to minimize the overall cost. As we will see, this subproblem corresponds to a generalized *ridge regression*, so we can obtain the closed form solution to optimize $\Phi$ efficiently.

To simplify this optimization problem, we first observe that each row of the regression coefficient matrix $\Phi_i^T = (\phi_{i1}, \cdots, \phi_{iN})$ can be optimized independently. This is because we can decompose the objective function (*cost*) into several sub-objective functions $F_i$ such that $cost = \sum_{i=1}^{N} F_i$, where

$$F_i = \sum_{k=1}^{K} \sum_{t=1}^{L} (p(k|i) x_i(t) - p(k|i) \sum_{j=1}^{N} \phi_{ij} x_j(t-1) p(k|j))^2$$
$$+ \alpha \sum_{j=1}^{N} \phi_{ij}^2 + \beta \left( \sum_{k=1}^{K} p(k|i) (2 \sum_{j=1}^{N} p(k|j) - p(k|i)) \right) \qquad (4)$$

Each $F_i$ is uniquely determined by the corresponding row $\Phi_i^T$, and can be solved independently. Moreover, the global minimum of *cost* is achieved by each $F_i$ obtaining its own minimum.

Given this, we now focus on how to optimize $F_i$ directly. To better understand this problem, we rewrite it in a matrix form. Let $y_k$ be the vector $(p(k|i)x_i(1), p(k|i)x_i(2), \cdots, p(k|i)x_i(L))^T$. Let $X_k$ be the the matrix with $L$ rows and $N$ columns where its entry at $t$-th row and $j$-th column is $p(k|i)p(k|j)x_j(t-1)$. Basically, each row of $X_k$ corresponds to a different time point, and each column of $X_k$ records a different time series. Using these two vectors, we can rewrite $F_i$ as follows:

$$F_i = \sum_{k=1}^{K} (y_k - X_k \Phi_i^T)^T (y_k - X_k \Phi_i^T) + \alpha \Phi_i^T \Phi_i + M_i \qquad (5)$$

where $M_i$ is the last term in Eq. 4, which is constant with regard to $\Phi$. Note that if $K = 1$ (with only one cluster), then we have the traditional ridge regression problem [13].

*Lemma 1:* The optimal $\Phi_i$ that minimizes $F_i$ is

$$\Phi_i = (\sum_{k=1}^{K} X_k^T X_k + \alpha I)^{-1} (\sum_{k=1}^{K} X_k^T y_k)$$

where $I$ is the identity matrix.

**Proof Sketch:** Simply by noting:

$$\frac{\partial F_i}{\partial \Phi_i} = -2\sum_{k=1}^{K} X_k^T(y_k - X_k\Phi_i) + \alpha\Phi_i$$

$$\frac{\partial^2 F_i}{\partial \Phi_i \Phi_i^T} = -2\sum_{k=1}^{K} X_k^T X_k + \alpha$$

Since $\frac{\partial^2 F_i}{\partial \Phi_i \Phi_i^T} > 0$ (shrinkage coefficient $\alpha > 0$), we set the first derivative to zero,

$$\sum_{k=1}^{K} X_k^T(y_k - X_k\Phi_i) + \alpha\Phi_i = 0$$

and obtain our results. □

Finally, we note that the matrix $X_k$ of each $F_i$ does not need to contain $N$ columns since the causality path-diagram $G = (V, E)$ is typically sparse. For each time series $i$, there is only a small number of other variables which will be its causal factors, and our regression coefficient matrix will only consider those variables. Thus, we only need to find $\phi_{ij}$ for those causal variables. Given this, we can see that each $X_k$ typically has only a small number of columns, making our method very efficient for computing the regression coefficient matrix.

**Overall Algorithm:** The overall procedure to decompose the path-diagram involving these two steps is sketched in Algorithm 1. We start by initializing the membership matrix $P$ (Line 1). The initial membership assignment can be purely random or utilize the knowledge of path-diagram structure (for instance, by a spectral clustering on the path-diagram). Then, we iteratively invoke the aforementioned two steps for optimizing membership matrix $P$ and coefficient matrix $\Phi$ (Lines 3 and 4). We repeat them until some stop criteria is satisfied, e.g., the improvement of the overall cost is very small or a certain number of iterations is reached (Line 5). Following that, hard cluster assignments can be made utilizing the optimal probabilistic membership (Line 6). A basic method is to simply assign each time series $i$ to its most probable cluster $k$, i.e., $k = argmax_r p(r|i)$. The key building block of our procedure is the employment of steps 1 and 2 to optimize *cost* to a local minimum (as formally stated in Theorem 1).

*Theorem 1:* The *cost* of path-diagram decomposition converges to a local minimum as we iteratively invoke steps 1 and 2.

**Complexity Analysis:** The complexity of our optimization procedure is as follows. In the first step, optimizing the probabilistic membership matrix, computing Eq. 3 and Hessian matrix approximation each take $O(N_1^2)$ time, where $N_1 = NK + N$. If the number of iterations is $k$, optimizing probabilistic membership can be completed in $O(kN_1^2)$ time. In the second step, optimizing the regression coefficient matrix, the highest computational cost is for matrix inversion. Inversion of a matrix can be computed in $O(R_i^3)$, where $R_i$ is the number of causal factors for the $i$-th time series variable. The overall time complexity of the second step is thus $O(\sum_{i=1}^{N} R_i^3)$. Since the number of causal factors for each variable is small, we can treat it as a constant for $N$ is large. Thus, the second step is much more efficient than the first step. In next section, we develop novel methods to speed up the computation process for the first step.

**Algorithm 1** PathDiagramDecomposition($X$,$G$,$K$)

**Parameter:** $X$ is the time series matrix
**Parameter:** $G$ is the path diagram
**Parameter:** $K$ is the number of clusters
1: initialize the membership matrix $P$;
2: **repeat**
3: step 1: optimize probabilistic membership matrix $P$;
4: step 2: optimize regression coefficient matrix $\Phi$;
5: **until** stop criteria is satisfied;
6: assign each time series to appropriate cluster using probabilistic membership matrix $P$;

## IV. A Scalable Approach for Membership Matrix Optimization

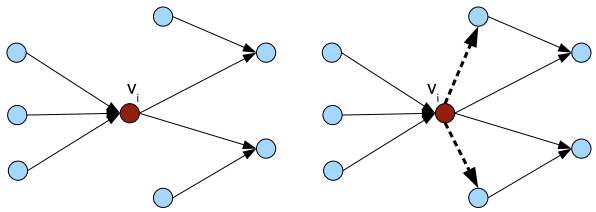
According to the complexity analysis in the previous section, optimizing the probabilistic membership matrix is the computational bottleneck of our iterative optimization procedure. The main issue is that the Quasi-Newton method is very costly for a large number of variables. In this section, we introduce a strategy which takes the variable dependence relationship into consideration and optimizes each variable (or a small number of variables) independently, assuming the relationships are fixed. Identifying the subdivision of variables that minimizes the final cost can be formulated as a *maximum weight independent set* problem.

### A. Covering Structure

Recall that the path diagram $G$ indicates the causal relationship between any two vertices. If $(v_i, v_j) \in E$, then $v_i$ is potentially a causal factor of $v_j$. This also corresponds to $\phi_{ij}$ in the regression coefficient matrix $\Phi$ being nonzero. In addition, from our objective function, we observe that membership $p(k|i)$ relates to only its predecessors, successors and the predecessors of its successors in path diagram $G$. The predecessors of $v_i$ correspond to those vertices in $G$ having an edge pointing to $v_i$, and the successors of $v_i$ correspond to those vertices that $v_i$ points to. In other words, the predecessors of $v_i$ contribute to the prediction of the $i$-th time series, and $v_j$ contributes to the prediction of its successors. In order to describe the set of time series variables (vertices) which $v_i$ directly relates to, we introduce the *covering structure* of $v_i$.

*Definition 3:* The **covering structure** of vertex $v_i$ is the set of vertices in a path diagram consisting of $v_i$'s predecessors, its successors, and the predecessors of its successors (see Figure 2(a)).

We say $v_j$ is *independent* of $v_i$ if $v_j$ is not in the *covering structure* of $v_i$. This independent relationship is symmetric, that is, $v_j$ is independent of $v_i$ means $v_i$ is independent of $v_j$ as well. In the following, we will see that if we assume




(a) (b)

Figure 2. **Covering Structure of** $v_i$

the probabilistic membership for each vertex in the covering structure of $v_i$ is fixed, then, we can find the optimal probabilistic membership for $v_i$, $(p(1|i), \cdots, p(K|i))$, by solving a set of simple linear equations.

**Optimizing individual membership $p(k|i)$ with respect to the covering structure of $v_i$:** To see the relationship between the membership function for a vertex $v_i$ and its covering structure, we first decompose the *cost* function into three parts (we omit the shrinkage term as it is a constant during the membership optimization):

$$cost = F_i + \sum_{j \in suc(v_i)} F_j + \sum_{j \in V \setminus (\{v_i\} \cup suc(v_i))} F_j \qquad (6)$$

where $F_i$ and $F_j$ correspond to the prediction errors for time series variables (vertices) $v_i$ and $v_j$, respectively, and $suc(v_i)$ is the immediate successors of vertex $v_i$ in the path diagram. To find the optimal $p(k|i)$, we perform the first order derivative of the *cost* function.

Let $y_i$ be the vector $(x_i(1), x_i(2), \cdots, x_i(L))^T$. Let $x_i$ be the vector $(x_i(0), x_i(1), \cdots, x_i(L-1))^T$. Let $Z_k$ be the matrix of size $L \times N$ where its entry at row $t$ and column $i$ is $x_i(t-1)p(k|i)$. Given this, we can write the derivative as follows (the derivative of the third term in Eq. 6 is zero):

$$\frac{\partial cost}{\partial p(k|i)} = \frac{\partial F_i}{\partial p(k|i)} + \frac{\partial (\sum_{s \in suc(v_i)} F_s)}{\partial p(k|i)}, \text{ where}$$

$$\frac{\partial F_i}{\partial p(k|i)} = 2p(k|i)(y_i^T - Z_k \Phi_i^T)^T (y_i^T - Z_k \Phi_i^T) + 2\beta \sum_{j=1}^{N} p(k|j),$$

$$\frac{\partial (\sum_{s \in suc(v_i)} F_s)}{\partial p(k|i)} = -2 \sum_{s \in suc(v_i)} p^2(k|s) \phi_{si} (y_s^T - Z_k \Phi_s^T)^T x_i$$

Its second order derivative is clearly greater than zero.

$$\frac{\partial^2 cost}{\partial p^2(k|i)} = 2(y_i^T - Z_k \Phi_i^T)^T (y_i^T - Z_k \Phi_i^T) + 2\beta + 2 \sum_{s \in suc(v_i)} p^3(k|s) \phi_{si}^2 x_i^T x_i > 0$$

Note that $\beta > 0$ because we want to minimize $\beta \sum_{k=1}^{K} (\sum_{i=1}^{N} C_{ik})^2$ when $\sum_{i=1}^{N} C_{ik}$ for every k is equal.

Now we enhance the objective function to take the probability constraint $\sum_k p(k|i) = 1$ into consideration:

$$newcost = F_i + \sum_{j \in suc(v_i)} F_j + \sum_{j \notin \{v_i\} \cup suc(v_i)} F_j + \sum_i \lambda_i (\sum_k p(k|i) - 1) \qquad (7)$$

The derivative of the new objective function with respect to each probabilistic membership $p(k|i)$, $(1 \leq k \leq K)$ is as follows:

$$\frac{\partial (newcost)}{\partial p(k|i)} = \frac{\partial F_i}{\partial p(k|i)} + \frac{\partial (\sum_{s \in suc(v_i)} F_s)}{\partial p(k|i)} + \lambda_i$$

Note that each of those $K$ equations is linear. If we include the constraint equation $\sum_k p(k|i) = 1$, we get a linear system with $K+1$ equations of $K+1$ variables: $K$ variables of $p(k|i)$ and one Lagrange multiplier $\lambda_i$. Since $K$ is typically very small and each variable is related to only a small number of causal factors (specified in the path diagram), we can solve each such linear system in almost constant time. For all the time series variables, we have $O(NK^3)$ time complexity.

However, we cannot apply these simultaneously as they all assume the probabilistic memberships of vertices in their covering structure are fixed. Our strategy is to find a set of vertices which can maximally optimize the overall *cost*, and then adjust their probabilistic memberships together. We can repeat this procedure until no improvement can be made. Given this, our problem is to find a set of vertices which maximally optimizes the overall *cost*, and are independent of each other with respect to the *covering structure*, such that no vertex appears in the covering structure of others.

### B. Maximum Weight Independent Set Approach

In the following, we transform the problem of choosing a set of vertices which can maximally optimize the overall *cost* into a maximum weight independent set problem. The intuition is that if a set of vertices are pairwise *independent*, then their cost improvements can be simply added together as their memberships do not rely on each other.

Given this, we introduce the *cover graph* by aggregating all the covering structures together. Specifically, *the cover graph $G_c = (V, E_c)$ is an undirected graph, where $V$ is the vertex set in the path diagram, and an edge $(v_i, v_j)$ exists if and only if $v_j$ is in the covering structure of $v_i$ or vice versa*. In other words, $v_i$ and $v_j$ are not independent. Then, we assign a weight to each vertex $v_i$ in cover graph $G$ as:

$$\Delta cost(v_i) = cost - cost(v_i)$$

where $cost(v_i)$ is the minimized cost after we optimize the membership of vertex $v_i$ independently and $cost$ is the original one. Thus, we can see the problem of choosing a set of vertices which can maximally optimize the overall *cost* is an instance of the maximum weight independence set problem. The *maximum weight independent set* (MWIS) problem is one of the well-known and well-studied problems in combinatorial optimization. While it has been proven to be NP-hard, efficient heuristic algorithms exist [14], [15]. We can apply any of them here.

Putting the probability constraint together with $newcost$ (Eq. 7), we reformulate it as a linear combination of inde-

pendent vertices:

$$newcost' = \sum_{i \in V_s} (F_i + \sum_{j \in suc(v_i)} F_j) + \sum_{j \notin V_s \cup suc(V_s)} F_j + \sum_{i \in V_s} \lambda_i (\sum_k p(k|i) - 1)$$

where $V_s$ is a set of independent vertices. Note that because they are independent, their first order derivatives still form a linear system. It consists of $|V_s| \times (K+1)$ equations for $|V_s| \times (K+1)$ variables, i.e., $|V_s| \times K$ variables of $p(k|i)$ and $|V_s|$ Lagrange multipliers $\lambda_i$ for probability constraints. Thus, we can apply efficient linear solvers, such as the Cholesky Factorization-based Minimum Degree method [16], to find the optimal membership assignment for all the vertices in $V_s$.

The sketch of our MWIS-based membership optimization scheme is outlined in Algorithm 2. For each vertex in the cover graph, we calculate its cost improvement and take it as vertex weight (Line 2 to 5). Then, a set of independent vertices in terms of covering structure is updated in order to maximally improve the *cost* objective function (Line 6 to 7). Finally, we repeat this process until the overall cost converges or certain stop criteria are satisfied (Line 8). Clearly, the $cost$ function is monotonically reduced by successively invoking membership optimization of the independent set; therefore, it converges to a local minimum.

**Algorithm 2** MembershipOptimization($G_c$,$P$)

**Parameter:** $G_c$ is the Cover Graph; $P$ is the Membership Matrix

1: **repeat**
2: **for all** Time series variable $v \in V(G_c)$ **do**
3: optimize membership of $v$ (i.e. $P_v$) by fixing the membership of its covering structure;
4: assign $v$ with weight based on improvement of cost;
5: **end for**
6: find a maximum weight independent set $IS$;
7: update probabilistic membership of vertices in $IS$;
8: **until** stop criteria is satisfied
9: **return** optimized membership matrix $P$ and improved $cost$

## V. Experimental Evaluation

In this section, we validate the accuracy and usefulness of our proposed approaches for temporal graphical modeling decomposition. First, we perform this validation on synthetic data with a known ground truth. Then we apply our approaches to analysis of real-world GDP data.

We apply our two methods for experimental evaluation: 1) iterative optimization based on the Quasi-Newton method (*newton*); 2) iterative optimization based on the MWIS method (*mwis*) where each vertex is updated. For purposes of comparison, two benchmarks are also used. The first benchmark (denoted as *Cor_Ncut*), uses the Pearson Correlation test to generate interaction relationships among different variables, then Ncut [17] is employed for clustering; The second benchmark (denoted as *Dcut*), applies directed spectral clustering [18] for path diagram decomposition. We implemented all algorithms using Matlab. All experiments were performed on an AMD 2.0 GHz dual-core Opteron with 4GB RAM.

### A. Synthetic Data

**Synthetic data generator:** In order to evaluate the decomposition (clustering) accuracy of our approaches, we utilize the following synthetic data generator which can specify the ground truth. Basically, the time series data are generated from a approximate block diagonal regression coefficient matrix, in two steps. In the first step, a community-based graph is constructed based on the method described in [19]. This graph is used as the underlying path diagram for time series data generation in the second step. Here, we can specify separate average vertex degrees for intra-community connections (denoted as $Z_{in}$) and inter-community connections (denoted as $Z_{out}$). In the second step, we apply the method introduced in [2] to obtain the time series data. Initially, each edge of the underlying path diagram is assigned a randomly generated weight as its regression coefficient. In addition, we skew the random values so that the regression coefficients for the intra-community pairs are generally larger than those for the inter-community pairs. Next, we repeatedly apply the path diagram's edge weights to generate time series data. In terms of matrix operations, the next time step's data is obtained by multiplying the regression coefficient matrix with the current time step's data vector and then adding a Gaussian noise vector with mean of zero. The process is essentially the same as the vector autoregression process described in Sec II-A, if the history length $T = 1$.

**Decomposition Accuracy:** An accurate decomposition (clustering) is one where the clusters generated by the algorithm closely correspond to the known true clusters. To measure and compare accuracy, we apply a technique developed for cluster ensembles [20]. Let $B = (U, V)$ be the complete bipartite graph where each vertex in $U$ corresponds to each cluster generated by a clustering algorithm, and each vertex in $V$ corresponds to a true cluster. Moreover, for each edge $(u_i, v_j)$, we assign a weight, equal to the size of the intersection set for the two clusters corresponding to $u_i$ and $v_j$. Thus, the clustering accuracy computation is transformed to finding a maximum bipartite matching for $B$. We accumulate the sum of weights of all edges in this matching. The ratio of this sum over the total number of variables in the data is the clustering accuracy.

We evaluate the decomposition accuracy of our approaches using two groups of time series datasets. The first group is on a small number of time series variables (on the order of tens). The second group is on a relatively large number of time series variables (on the order of hundreds).

**Results for a small number of time series variables:** The experimental results are shown in Table 1. Each experiment is parameterized by the number of variables for the time series data ($\#Vars$) and the number of communities ($K$).

| #Vars | $Z_{out} = 0.1 \times \#Vars/K$ | | | |
|---|---|---|---|---|
| | Cor_Ncut | Dcut | newton | mwis |
| 10 | 0.7 | 0.6 | 1 | 1 |
| 20 | 0.8 | 1 | 1 | 1 |
| 30 | 0.63 | 0.7 | 1 | 0.87 |
| 40 | 0.725 | 0.8 | 1 | 0.975 |
| 50 | 0.66 | 0.72 | 1 | 0.98 |
| | $Z_{out} = 0.2 \times \#Vars/K$ | | | |
| 10 | 0.7 | 0.6 | 0.8 | 0.8 |
| 20 | 0.8 | 0.8 | 1 | 0.95 |
| 30 | 0.63 | 0.57 | 0.93 | 0.87 |
| 40 | 0.58 | 0.48 | 1 | 0.975 |
| 50 | 0.5 | 0.54 | 0.94 | 0.8 |
| | $Z_{out} = 0.3 \times \#Vars/K$ | | | |
| 10 | 0.7 | 0.8 | 1 | 1 |
| 20 | 0.75 | 0.55 | 0.9 | 0.85 |
| 30 | 0.6 | 0.6 | 0.9 | 0.83 |
| 40 | 0.58 | 0.4 | 0.68 | 0.65 |
| 50 | 0.6 | 0.34 | 0.68 | 0.66 |

Table I
CLUSTERING ACCURACY ON SMALL DATASETS

For these tests, we varied the average number of inter-community connections ($Z_{out} = 0.1 \times \#Vars/K$, $Z_{out} = 0.2 \times \#Vars/K$ and $Z_{out} = 0.3 \times \#Vars/K$), while fixing the average number of intra-community connections to be $Z_{in} = 0.5 \times \#Vars/K$. Note that $\#Vars/K$ represents the number of vertices in each community. For each set of $Z$ values, we made five datasets, varying $\#Vars$ from 10 to 50. The vertices in each dataset were decomposed into different numbers of communities, ranging from 2 to 5 communities.

As we can see, *newton* consistently obtains the best clustering accuracy among all four algorithms. Overall, the clustering accuracy of *newton* is better than benchmarks *Cor_Ncut* and *Dcut* by an average of $27.2\%$ and $32\%$, respectively. In addition, *mwis* is better than *Cor_Ncut* and *Dcut* by an average of $23.9\%$ and $28.8\%$, respectively. These results show that traditional spectral clustering *Dcut* cannot accurately decompose even relatively small set of time series. In contrast, both of our methods perform well, with 100% accuracy in several cases.

In addition, *Cor_Ncut* and *Dcut*, which also employ spectral clustering, are faster than others in terms of clustering time, especially in Matlab which has been highly optimized for matrix computation. As for our two algorithms, *mwis* takes from 2 seconds to 10 minutes for each dataset, while *newton* needs only 1 to 69 seconds to finish. As expected, *newton* is the best on datasets with a small number of variables.

**Results for a large number of time series variables:** In the second experiment, we generated the times series data with the number of vertices ranging from 100 to 800, while fixing the average number per vertex of intra-community connections $Z_{in} = 30$, and the average for inter-community connection $Z_{out} = 20$. The community-based path diagrams in these datasets contain from 2 to 8 communities. As we expected, *newton* was computationally inefficient, even crashing Matlab in some instances.

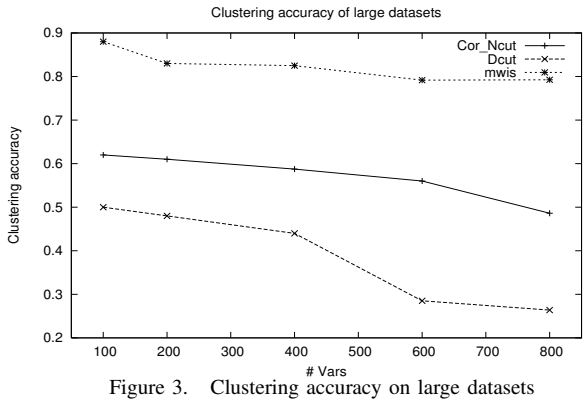


Figure 3. Clustering accuracy on large datasets

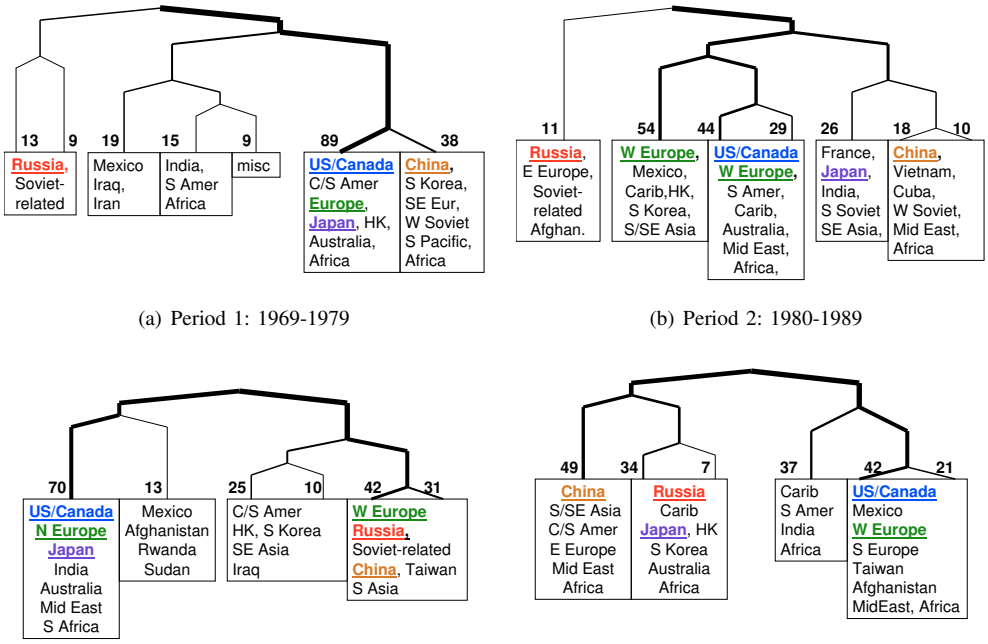
Figure 3 shows the clustering accuracy of large-scale datasets. Here, the clustering accuracy of *mwis* is better than that of *Cor_Ncut* by an average of $30.1\%$. From the figure, we can see our *mwis* is significantly better than the benchmark *Dcut*, outperforming it by approximately $52.5\%$ clustering accuracy. It is interesting to observe that the clustering accuracy of both *Cor_Ncut* and *Dcut* tend to decrease as the number of variables increases. However, our algorithm *mwis* maintains good performance on all large-scale datasets. Moreover, *mwis* was able to achieve their accuracy even with a limited number of iterations.

### B. Real Data

To validate our approaches in a real-world application, we use global economic data to seek temporal country-country dependencies. Our dataset consists of GDP (gross domestic product) for 192 countries, as collected by the USDA (http://www.ers.usda.gov/Data/Macroeconomics/). The time series data for each country is its annual GDP growth rate over the period from 1969 to 2007. We subdivide the time range into four time periods of approximately 10 years each: 1969-1979, 1980-1989, 1990-1999, and 1998-2007. We apply the $MWIS$-based decomposition approach to top-down hierarchical bipartitioning down to 6 or 7 partitions, to group countries into interdependent groups. Our results exhibit meaningful, sometimes fascinating clusterings.

Most notable is how the grouping of the Soviet Republics changes across the four time periods. In period 1 (1970s), the top partition separates out Russia and 21 other nations, indicating that the most significant division at the global level is to separate out these economies from the rest of the world. Most of the 22 are either Soviet Republics (8), other communist states (5), or received strong Soviet support (3 - Angola, Uganda, Ethiopia). The remaining countries, in the Middle East or Africa, found the 1970s to be an unsettling time. None of these 22 nations were Western capitalist nations.

In the 1980s, the top partition separates out Russia again, but with only 11 peers, including Soviet-occupied Afghanistan. The smaller size may indicate that some communist state economies were beginning to interact more with the rest of the world and less within the communist bloc.




(a) Period 1: 1969-1979

(b) Period 2: 1980-1989

(c) Period 3: 1990-1999

(d) Period 4: 1998-2007

Figure 4. GDP Growth Rate Time Series

After Period 2, this high-level communist bloc is gone. In Period 3, Russia is in a $4^{th}$ tier group of 31 mixed nations. In Period 4, it is again in a $4^{th}$ tier group, this time with capitalist countries Japan and Australia.

Note that these clusterings are different from what would have been obained if one ignored the temporal dependence and merely tried to match similar patterns of GDP growth. Using ordinary trajectory matching, the top-level Soviet partitions in periods 1 and 2 would not have formed, because these states did not all share the same growth pattern.

Another interesting observation relates to shifting patterns of dependence and independence among four key entities: The United States, Russia, China, and Western Europe. The U.S. always shows close temporal ties with Canada, and at least part of Europe. It is also always maximally separated from Russia. However, China and Japan change their affiliation with each time period. A final observation is the changing balance of cluster sizes, as indicated by the link thicknesses. This can provide some new insight into shifting balances of power.

## VI. Related Work

Causal modeling or identification of causal relationships have been an area of active scientific research [21], [22]. Traditionally, inference about cause-effect relationships is commonly based on the concept of Granger causality, first proposed by Clive Granger [4] in 1969. Recently, several researchers have combined the notion of Granger causality with graphical models [23], [24] to visualize the cause-effect interactions for multivariate time series data[25], [5]. However, to the best of our knowledge, no effort has been made to try to simplify and to derive a global view of a temporal causal model. As we argued, this is clearly very important for understanding the interactions among the time series variables.

Our work is also related to time series clustering, which has been extensively studied in the data mining and machine learning communities [8], [9], [10], [11]. What differentiates our work from the existing work is that we focus on the interaction of time series variables. Existing time series clustering methods do not assume that time series interaction is relevant. Instead, they focus on deriving distance measures or probabilistic models to capture the similarity between time series. Basically, their goal is to group similar time series into a cluster. However, the goal here is to cluster time series through their causal relationships. As a simple example, two identical time series would not be Granger-causal of one another (adding one time series will not improve the prediction of the time series for itself). Thus, we are not compelling to put them together into the same component.

## VII. Conclusion

In summary, we have formulated a novel objective function for the decomposition problem in temporal graphical models. We then introduced an iterative optimization approach utilizing the Quasi-Newton method and generalized ridge regression to minimize the objective function. To improve the efficiency of the Quasi-Newton method on datasets with a large number of variables, we employ a *maximum weight independent set*-based approach. Our experiments on synthetic data demonstrate the effectiveness of

our approaches, in terms of clustering accuracy. In addition, our tests on real GDP data uncover interesting relationships among countries. In this work, we only consider non-overlapping clusters. However, many real-world datasets have inherently overlapping clusters. We plan to investigate this problem in the future.

## VIII. Acknowledgment

This work is partially supported by NIH 1R01CA141090-0109.

## References

[1] W. H. Tsen, "Exports, domestic demand and economic growth in china: Granger causality analysis," in *An international conference on WTO, China, and the Asian Economies, IV: Economic Integration and Economic Development*, 2006.

[2] A. Arnold, Y. Liu, and N. Abe, "Temporal causal modeling with graphical granger methods," in *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, 2007, pp. 66–75.

[3] W. Xue, C. Yonghong, B. S. L, and D. Mingzhou, "Granger causality between multiple interdependent neurobiological time series: blockwise versus pairwise methods," *International journal of neural systems*, vol. 17, no. 2, pp. 71–8, 2007.

[4] C. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, vol. 37, pp. 424–438, 1969.

[5] M. Eichler, "Granger causality and path diagrams for multivariate time series," *Journal of Econometrics*, vol. 137, pp. 334–353, 2007.

[6] S. Haufe, K.-R. Muller, G. Nolte, and N. Kramer, "Sparse causal discovery in multivariate time series," in *Proceedings of the NIPS'08 workshop on causality*, 2008.

[7] R. Jin, N. Ruan, S. McCallen, and V. Lee, "Dynamic module discovery in temporal complex networks," in *International workshop on analysis of dynamic networks at the SIAM international conference on Data Mining*, 2009.

[8] W. Liao, "Clustering of time series data–a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, November 2005.

[9] A. J. Bagnall and G. J. Janacek, "Clustering time series from ARMA models with clipped data," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 49–58.

[10] J. Lin, M. Vlachos, E. Keogh, and D. Gunopulos, "Iterative incremental clustering of time series," in *In EDBT*, 2004, pp. 106–122.

[11] X. Wang, A. Wirth, and L. Wang, "Structure-based statistical features and multivariate time series clustering," in *ICDM*, 2007, pp. 351–360.

[12] D. P. Bertsekas, *Constrained optimization and Lagrange Multiplier methods*. Academic Press, 1982.

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[14] I. Chang, W.-Z. Shao, and H.-H. Teh, "Heuristic solutions for the general maximum independent set problem with applications to expert system design," in *Computer Software and Applications Conference, COMPSAC 88*, 1988.

[15] L. Hars, "Hybrid heuristic for the maximum weighted independent set problem," Institut für Ökonometrie und Operations Research, University of Bonn, Tech. Rep., 1989.

[16] G. Kumfert, A. Pothen, P. Heggernes, P. Heggernes, S. C. Eisenstat, and S. C. Eisenstat, "The computational complexity of the minimum degree algorithm," in *Proceedings of 14th Norwegian Computer Science Conference, NIK 2001, University of Troms, Norway*, 2001, pp. 98–109.

[17] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 22, no. 8, pp. 888–905, 2000.

[18] J. H. D. Zhou and B. Schölkopf, "Learning from labeled and unlabeled data on directed graph," in *22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

[19] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci. USA*, vol. 99, pp. 7821–7826, 2002.

[20] P. Hore, L. O. Hall, and D. B. Goldgof, "A scalable framework for cluster ensembles," *Pattern Recogn.*, vol. 42, no. 5, pp. 676–688, 2009.

[21] M. Kalisch and P. Buehlmann, "Estimating high-dimensional directed acyclic graphs with the pc algorithm," ETH Zurich, Tech. Rep., 2005.

[22] A. Moneta and P. Spirtes, "Graphical models for the identification of causal structures in multivariate time series models." *Fifth international conference on computation intelligence in economics and finance*, 2006.

[23] D. Heckerman, "A tutorial on learning with baysian networks," Cambridge, MA, USA, 1999.

[24] J. Pearl, *Causality*. Cambridge University Press, Cambridge, UK, 2000.

[25] M. Eichler, "Graphical modelling of multivariate time series with latent variables," University of Heidelberg Tech. Rep., 2006.