Learning Predictive Leading Indicators for Forecasting Time Series Systems with Unknown Clusters of Forecast Tasks

Magda Gregorová[1,2] MAGDA.GREGOROVA@HESGE.CH
Alexandros Kalousis[1,2] ALEXANDROS.KALOUSIS@HESGE.CH
Stéphane Marchand-Maillet[2] STEPHANE.MARCHAND-MAILLET@UNIGE.CH
[1] *Geneva School of Business Administration, HES-SO University of Applied Sciences of Western Switzerland;* [2] *University of Geneva, Switzerland*

Editors: Yung-Kyun Noh and Min-Ling Zhang

Abstract

We present a new method for forecasting systems of multiple interrelated time series. The method learns the forecast models together with discovering leading indicators from within the system that serve as good predictors improving the forecast accuracy and a cluster structure of the predictive tasks around these. The method is based on the classical linear vector autoregressive model (VAR) and links the discovery of the leading indicators to inferring sparse graphs of Granger causality. We formulate a new constrained optimisation problem to promote the desired sparse structures across the models and the sharing of information amongst the learning tasks in a multi-task manner. We propose an algorithm for solving the problem and document on a battery of synthetic and real-data experiments the advantages of our new method over baseline VAR models as well as the state-of-the-art sparse VAR learning methods.

Keywords: Time series forecasting; VAR; Granger causality; structured sparsity; multi-task learning; leading indicators

1. Introduction

Time series forecasting is vital in a multitude of application areas. With the increasing ability to collect huge amounts of data, users nowadays call for forecasts for large systems of series. On one hand, practitioners typically strive to gather and include into their models as many potentially helpful data as possible. On the other hand, the specific domain knowledge rarely provides sufficient understanding as to the relationships amongst the series and their importance for forecasting the system. This may lead to cluttering the forecast models with irrelevant data of little predictive benefit thus increasing the complexity of the models with possibly detrimental effects on the forecast accuracy (over-parametrisation and over-fitting).

In this paper we focus on the problem of forecasting such large time series systems from their past evolution. We develop a new forecasting method that learns sparse structured models taking into account the unknown underlying relationships amongst the series. More specifically, the learned models use a limited set of series that the method identifies as useful for improving the predictive performance. We call such series the *leading indicators*.

In reality, there may be external factors from outside the system influencing the system developments. In this work we abstract from such external con-founders for two reasons.

First, we assume that any piece of information that could be gathered has been gathered and therefore even if an external confounder exists, there is no way we can get any data on it. Second, some of the series in the system may serve as surrogates for such unavailable data and we prefer to use these to the extent possible rather than chase the holy grail of full information availability.

We focus on the class of linear vector autoregressive models (VARs) which are simple yet theoretically well-supported, and well-established in the forecasting practice as well as the state-of-the-art time series literature, e.g. Lütkepohl (2005). The new method we develop falls into the broad category of graphical-Granger methods, e.g. Lozano et al. (2009); Shojaie and Michailidis (2010); Songsiri (2013). Granger causality (Granger, 1969) is a notion used for describing a specific type of dynamic dependency between time series. In brief, a series Z Granger-causes series Y if, given all the other relevant information, we can predict Y more accurately when we use the history of Z as an input in our forecast function. In our case, we call such series Z, that contributes to improving the forecast accuracy, the leading indicator.

For our method, we assume little to no prior knowledge about the structure of the time series systems. Yet, we do assume that most of the series in the system bring, in fact, no predictive benefit for the system, and that there are only few leading indicators whose inclusion into the forecast model as inputs improves the accuracy of the forecasts. Technically this assumption of only few leading indicators translates into a sparsity assumption for the forecast model, more precisely, sparsity in the connectivity of the associated Granger-causal graph.

An important subtlety for the model assumptions is that the leading indicators may not be *leading* for the whole system but only for some parts of it (certainly more realistic especially for lager systems). A series Z may not Granger-cause all the other series in the system but only some of them. Nevertheless, if it contributes to improving the forecast accuracy of a group of series, we still consider it a leading indicator for this group. In this sense, we assume the system to be composed of clusters of series organised around their leading indicators. However, neither the identity of the leading indicators nor the composition of the clusters is known a priori.

To develop our method, we built on the paradigms of multi-task, e.g. Caruana (1997); Evgeniou and Pontil (2004), and sparse structured learning (Bach et al., 2012). In order to achieve higher forecast accuracy our method encourages the tasks to borrow strength from one another during the model learning. More specifically, it intertwines the individual predictive tasks by shared structural constraints derived from the assumptions above.

To the best of our knowledge this is the first VAR learning method that promotes common sparse structures across the forecasting tasks of the time series system in order to improve the overall predictive performance. We designed a novel type of structured sparsity constraints coherent with the structural assumptions for the system, integrated them into a new formulation of a VAR optimisation problem, and proposed an efficient algorithm for solving it. The new formulation is unique in being able to discover clusters of series based on the structure of their predictive models concentrated around small number of leading indicators.

Organisation of the paper The following section introduces more formally the basic concepts: linear VAR model and Granger causality. The new method is described in section 3. For clarity of exposition we start in section 3.1 from a set of simplified assumptions. The full method for learning VAR models with task Clustering around Leading indicators (CLVAR) is presented in section 3.2. We review the related work in section 4. In section 5 we present the results of a battery of synthetic and real-data experiments in which we confirm the good performance of our method as compared to a set of baseline state-of-the-art methods. We also comment on unfavourable configurations of data and the bottlenecks in scaling properties. We conclude in section 6.

2. Preliminaries

Notation We use bold upper case and lower case letters for matrices and vectors respectively, and plain letters for scalars (including elements of vectors and matrices). For a matrix $\mathbf{A}$, the vectors $\mathbf{a}_{i,.}$ and $\mathbf{a}_{.,j}$ indicate its ith row and jth column, $a_{i,j}$ is the (i, j) element of the matrix. $\mathbf{A}'$ is the transpose of $\mathbf{A}$, $diag(\mathbf{A})$ is the matrix constructed from the diagonal of $\mathbf{A}$, $\odot$ is the Hadamard product, $\otimes$ is the Kronecker product, $vec(\mathbf{A})$ is the vectorization operator, and $||\mathbf{A}||_F$ is the Frobenius norm. Vectors are by convention column-wise so that $\mathbf{x} = (x_1, \ldots, x_n)'$ is the n-dimensional vector $\mathbf{x}$. For any vectors $\mathbf{x, y}$, $\langle \mathbf{x}, \mathbf{y} \rangle$ and $||\mathbf{x}||_2$ are the standard inner product and ℓ_2 norms. $\mathbf{1}_K$ is the K-dimensional vector of ones.

2.1. Vector Autoregressive Model

For a set of K time series observed at T synchronous equidistant time points we write the VAR in the form of a multi-output regression problem as $\mathbf{Y} = \mathbf{X}\mathbf{W} + \mathbf{E}$. Here $\mathbf{Y}$ is the $T \times K$ output matrix for T observations and K time series as individual 1-step-ahead forecasting tasks, $\mathbf{X}$ is the $T \times Kp$ input matrix so that each row $\mathbf{x}_{t,.}$ is a Kp long vector with p lagged values of the K time series as inputs $\mathbf{x}_{t,.} = (y_{t-1,1}, y_{t-2,1}, \ldots, y_{t-p,1}, y_{t-1,2}, \ldots, y_{t-p,K})'$, and $\mathbf{W}$ is the corresponding $Kp \times K$ parameters matrix where each column $\mathbf{w}_{.,k}$ is a model for a single time series forecasting task (see Fig. 1). We follow the standard time series assumptions: the $T \times K$ error matrix $\mathbf{E}$ is a random noise matrix with i.i.d. rows with zero mean and a diagonal covariance; the time series are second order stationary and centred (so that we can omit the intercept).

In principle, we can estimate the model parameters by minimising the standard squared error loss

$$L(\mathbf{W}) := \sum_{t=1}^{T} \sum_{k=1}^{K} (y_{t,k} - \langle \mathbf{w}_{.,k}, \mathbf{x}_{t,.} \rangle)^2 \tag{1}$$

which corresponds to maximising the likelihood with i.i.d. Gaussian errors and spherical covariance. However, since the dimensionality Kp of the regression problem quickly grows with the number of series K (by a multiple of p), often even relatively small VARs suffer from over-parametrisation ($Kp \gg T$). Yet, typically not all the past of all the series is indicative of the future developments of the whole system. In this respect the VARs are typically sparse.

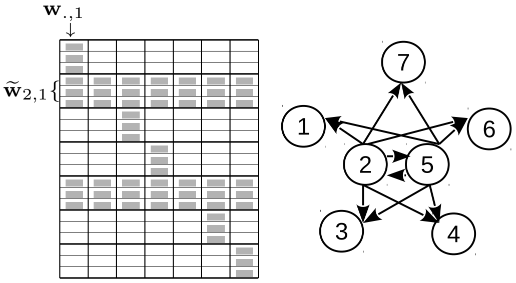
In practice, the univariate autoregressive model (AR) which uses as input for each time series forecast model only its own history (and thus is an extreme sparse version of VAR), is often difficult to beat by any VAR model with the complete input sets. A variety of approaches such as Bayesian or regularisation techniques have been successfully used in the past to promote sparsity and condition the model learning. Those most relevant to our work are discussed in section 4.

2.2. Granger-causality Graphs

Granger (1969) proposed a practical definition of causality in time series based on the accuracy of least-squares predictor functions. In brief, for two time series Z and Y, we say that Z Granger causes if, given all the other relevant information, a predictor function using the history of Z as input can forecast Y better (in the mean-square sense) than a function not using it. Similarly, a set of time series $\{Z_1, \ldots, Z_l\}$ G-causes series Y if it can be predicted better using the past values of the set.

The G-causal relationships can be described by a directed graph $\mathcal{G} = \{\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}}\}$ (Eichler (2012)), where each node $v \in \boldsymbol{\mathcal{V}}$ represents a time series in the system, and the directed edges represent the G-causal relationships between the series. In VARs the G-causality is captured within the $\mathbf{W}$ parameters matrix. When any of the parameters of the k-th task (k-th column of the $\mathbf{W}$) referring to the p past values of the l-th input series is non-zero, we say that the l-th series G-causes series k, and we denote this in the G-causal graph by a directed edge $e_{l,k}$ from v_l to v_k.

Fig. 1 shows a schema of the VAR parameters matrix $\mathbf{W}$ and the corresponding G-causal graph for an example system of $K = 7$ series with the number of lags $p = 3$. In 1(a) the gray cells are the non-zero elements, in 1(b) the circle nodes are the individual time series, the arrow edges are the G-causal links between the series[1]. For example, the arrow from 2 to 1 indicates that series 2 G-causes series 1; correspondingly the cells for the 3 lags in the 2nd block-row and the 1th column are shaded ($\widetilde{\mathbf{w}}_{2,1}$). Series 2 and 5 are the leading indicators for the whole system, their block-rows are shaded in all columns in the $\mathbf{W}$ matrix schema and they have out-edges to all other nodes in the G-graph.



a) $\mathbf{W}$ matrix b) G-causal graph

Figure 1: $\mathbf{W}$ and G-causal graph.

One may question if calling the above notion *causality* is appropriate. Indeed, unlike other perhaps more philosophical approaches, e.g. Pearl (2009), it does not really seek to understand the underlying forces driving the relationships between the series. Instead, the concept is purely technical based on the series contribution to the predictive accuracy, ignoring also possible confounding effects of unobservables. Nevertheless, the term is well established in the time series community. Moreover, it fits very well our purposes, where the primary objective is to learn models with high forecast accuracy that use as inputs only those time series that contribute to improving the accuracy - the leading indicators.

[1] The self-loops corresponding to the block-diagonal elements in $\mathbf{W}$ are omitted for clarity of display.

Therefore, acknowledging all the reservations, we stick to it in this paper always preceding it by Granger or G- to avoid confusion.

3. Learning VARs with Clusters around Leading Indicators

We present here our new method for learning VAR models with task Clustering around Leading indicators (CLVAR). The method relies on the assumption that the generating process is sparse in the sense of there being only a few leading indicators within the system having an impact on the future developments. The leading indicators may be useful for predicting all or only some of the series in the systems. In this respect the series are clustered around their G-causing leading indicators. However, the method does not need to know the identity of the leading indicators nor the cluster assignments a priori and instead learns these together with the predictive models.

In building our method we exploited the multi-task learning ideas (Caruana, 1997) and let the models benefit from learning multiple tasks together (one task per series). This is in stark contrast to other state-of-the-art VAR and graphical-Granger methods, e.g. Arnold et al. (2007); Lozano et al. (2009); Liu and Bahadori (2012). Albeit them being initially posed as multi-task (or multi-output) problems, due to their simple additive structure they decompose into a set of single-task problems solvable independently without any interaction and information sharing during the per-task learning. We, on the other hand, encourage the models to share information and borrow strength from one another in order to improve the overall performance by intertwining the model learning via structural constraints on the models derived from the assumptions outlined above.

3.1. Leading Indicators for Whole System

For the sake of exposition we first concentrate on a simplified problem of learning a VAR with leading indicators shared by the whole system (without clustering). The structure we assume here is the one illustrated in Fig. 1. We see that the parameters matrix $\mathbf{W}$ is sparse with non-zero elements only in the block-rows corresponding to the lags of the leading indicators for the system (series 2 and 5 in the example in Fig. 1) and on the block diagonal. The block-diagonal elements of $\mathbf{W}$ are associated with the lags of each series serving as inputs for predicting its own 1-step-ahead future. It is a stylised fact that the future of a stationary time series depends first and foremost on its own past developments. Therefore in addition to the leading indicators we want each of the individual series forecast function to use its own past as a relevant input. We bring the above structural assumptions into the method by formulating novel fit-for-purpose constraints for learning VAR models with multi-task structured sparsity.

3.1.1. Learning Problem and Algorithm for Learning without Clusters

We first introduce some new notation to accommodate for the necessary block structure across the lags of the input series in the input matrix $\mathbf{X}$ and the corresponding elements of the parameters matrix $\mathbf{W}$. For each input vector $\mathbf{x}_{t,.}$ (a row of $\mathbf{X}$) we indicate by $\widetilde{\mathbf{x}}_{t,j} = (y_{t-1,j}, y_{t-2,j}, \ldots, y_{t-p,j})'$ the p-long sub-vector of $\mathbf{x}_{t,.}$ referring to the history (the p lagged values preceding time t) of the series j, so that for the whole row we have $\mathbf{x}_{t,.} =$

$(x_{t,1}, \ldots, x_{t,Kp})' = (\widetilde{\mathbf{x}}'_{t,1}, \ldots, \widetilde{\mathbf{x}}'_{t,K})'$. Correspondingly, in each model vector $\mathbf{w}_{.,k}$ (a column of $\mathbf{W}$), we indicate by $\widetilde{\mathbf{w}}_{j,k}$ the p-long sub-vector of the kth model parameters associated with the input sub-vector $\widetilde{\mathbf{x}}_{t,j}$. In Fig. 1, $\widetilde{\mathbf{w}}_{2,1}$ is the block of the 3 shaded parameters in column 1 and rows $\{4, 5, 6\}$ - the block of parameters of the model for forecasting the 1st time series associated with the 3 lags of the 2nd time series (a leading indicator) as inputs. Using these blocks of inputs $\widetilde{\mathbf{x}}_{t,j}$ and parameters $\widetilde{\mathbf{w}}_{j,k}$ we can rewrite the inner products in the loss in (1) as $\langle \mathbf{w}_{.,k}, \mathbf{x}_{t,.} \rangle = \sum_{b=1}^{K} \langle \widetilde{\mathbf{w}}_{b,k}, \widetilde{\mathbf{x}}_{t,b} \rangle$.

Next, we associate each of the parameter blocks with a single non-negative scalar $\gamma_{b,k}$ so that $\widetilde{\mathbf{w}}_{b,k} = \gamma_{b,k}\, \widetilde{\mathbf{v}}_{b,k}$. The $Kp \times K$ matrix $\mathbf{V}$, composed of the blocks $\widetilde{\mathbf{v}}_{b,k}$ in the same way as $\mathbf{W}$ is composed of $\widetilde{\mathbf{w}}_{b,k}$, is therefore just a rescaling of the original $\mathbf{W}$ with the weights $\gamma_{b,k}$ used for each block. With this new re-parametrization the squared-error loss (1) is

$$L(\mathbf{W}) = \sum_{t=1}^{T} \sum_{k=1}^{K} (y_{t,k} - \sum_{b=1}^{K} \gamma_{b,k} \langle \widetilde{\mathbf{v}}_{b,k}, \widetilde{\mathbf{x}}_{t,b} \rangle)^2. \tag{2}$$

Finally, we use the non-negative $K \times K$ weight matrix $\mathbf{\Gamma} = \{\gamma_{b,k} \mid b, k = 1, \ldots, K\}$ to formulate our multi-task structured sparsity constraints. In $\mathbf{\Gamma}$ each element corresponds to a single series serving as an input to a single predictive model. A zero weight $\gamma_{b,k} = 0$ results in a zero parameter sub-vector $\widetilde{\mathbf{w}}_{b,k} = \mathbf{0}$ and therefore the corresponding input sub-vectors $\widetilde{\mathbf{x}}_{t,b}$ (the past lags of series b for each time point t) have no effect in the predictive functions for task k.

Our assumption of only small number of leading indicators means that most series shall have no predictive effect for any of the tasks. This can be achieved by $\mathbf{\Gamma}$ having most of its rows equal to zero. On the other hand, the non-zero elements corresponding to the leading indicators shall form full rows of $\mathbf{\Gamma}$. As explained in section 3.1, in addition to the leading indicators we also want each series past to serve as an input to its own forecast function. This translates to non-zero diagonal elements $\gamma_{i,i} \neq 0$. To combine these two contradicting structural requirements onto $\mathbf{\Gamma}$ (sparse rows vs. non-zero diagonal) we construct the matrix from two same size matrices $\mathbf{\Gamma} = \mathbf{A} + \mathbf{B}$, one for each of the structures: $\mathbf{A}$ for the row-sparse of leading indicators, $\mathbf{B}$ for the diagonal of the own history.

We now formulate the optimisation problem for learning VAR with shared leading indicators across the whole system and dependency on own past as the constrained minimisation

$$\begin{array}{ll} \operatorname{argmin}_{\mathbf{A},\mathbf{V}} & \sum_{t=1}^{T} \sum_{k=1}^{K} (y_{t,k} - \sum_{b=1}^{K} (\alpha_{b,k} + \beta_{b,k}) \langle \widetilde{\mathbf{v}}_{b,k}, \widetilde{\mathbf{x}}_{t,j} \rangle)^2 + \lambda ||\mathbf{V}||_F^2 \\ \text{s.t.} & \mathbf{1}'_K\, \overline{\boldsymbol{\alpha}} = \kappa;\ \overline{\boldsymbol{\alpha}} \geq \mathbf{0};\ \boldsymbol{\alpha}_{.,j} = \overline{\boldsymbol{\alpha}},\ \beta_{j,j} = 1 - \alpha_{j,j}\ \forall j = 1, \ldots, K \end{array} , \tag{3}$$

where the links between the matrices $\mathbf{A}, \mathbf{B}, \mathbf{\Gamma}, \mathbf{V}$ and the parameter matrix $\mathbf{W}$ of the VAR model are explained in the paragraphs above.

In (3) we force all the columns of $\mathbf{A}$ to be equal to the same vector $\overline{\boldsymbol{\alpha}}$[2], and we promote the sparsity in this vector by constraining it onto a simplex of size κ. κ controls the relative weight of each series own past vs. the past of all the neighbouring series. For identifiability reasons we force the diagonal elements of $\mathbf{\Gamma}$ to equal unity by scaling appropriately the diagonal $\beta_{j,j}$ elements. Lastly, while $\mathbf{\Gamma}$ is constructed and constrained to control for the

[2] This does not excessively limit the capacity of the models as the final model matrix $\mathbf{W}$ is the result of combining $\mathbf{\Gamma}$ with the learned matrix $\mathbf{V}$.

structure of the learned models (as per our assumptions), the actual value of the final parameters $\mathbf{W}$ is the result of combining it with the other learned matrix $\mathbf{V}$. To confine the overall complexity of the final model $\mathbf{W}$ we impose a standard ridge penalty (Hoerl and Kennard, 1970) on the model parameters $\mathbf{V}$.

The optimisation problem (3) is jointly non-convex, however, it is convex with respect to each of the optimisation variables with the other variable fixed. Therefore we propose to solve it by an alternating descent for $\mathbf{A}$ and $\mathbf{V}$ as outlined in algorithm 1 below. $\mathbf{B}$ is solved trivially applying directly the equality constraint of (3) over the learned matrix $\mathbf{A}$ as $\mathbf{B} = \mathbf{I} - diag(\mathbf{A})$ which implies $\mathbf{\Gamma} = \mathbf{A} + \mathbf{B} = \mathbf{A} - diag(\mathbf{A}) + \mathbf{I}$.

Algorithm 1: Alternating descent for VAR with system-shared leading indicators

```
Input      : training data $\mathbf{Y}, \mathbf{X}$; hyper-parameters $\lambda, \kappa$
Initialise: $\overline{\boldsymbol{\alpha}}$ evenly to satisfy constraints in all columns of $\mathbf{A}$; $\mathbf{\Gamma} \leftarrow \mathbf{A} - diag(\mathbf{A}) + \mathbf{I}$
repeat                                                          // Alternating descent
    begin Step 1: Solve for $\mathbf{V}$
        foreach task $k$ do
            re-weight input blocks $\mathbf{z}_{t,b}^{(k)} \leftarrow \gamma_{b,k}\, \widetilde{\mathbf{x}}_{t,b}$    $\forall$ time point $t$ and input series $b$
            $\mathbf{v}_{.,k} \leftarrow \text{argmin}_{\mathbf{v}}\, ||\mathbf{y}_{.,k} - \mathbf{Z}^{(k)}\, \mathbf{v}||_2^2 + \lambda ||\mathbf{v}||_2^2$        // standard ridge regression
        end
    end
    begin Step 2: Solve for $\mathbf{A}$ and $\mathbf{\Gamma}$
        foreach task $k$ do
            input products $h_{t,b}^{(k)} \leftarrow \langle \widetilde{\mathbf{v}}_{b,k}, \widetilde{\mathbf{x}}_{t,b} \rangle$    $\forall$ time point $t$ and input series $b$
            task residuals after using own history $r_{t,k} \leftarrow y_{t,k} - h_{t,k}^{(k)}$    $\forall$ time point $t$
            remove own history from input products $h_{t,k}^{(k)} \leftarrow 0$    $\forall$ time point $t$
        end
        concatenate vertically input product matrices $\mathbf{H} = vertcat(\mathbf{H}^{(\cdot)})$
        $\overline{\boldsymbol{\alpha}} \leftarrow \text{argmin}_{\overline{\boldsymbol{\alpha}}}\, ||vec(\mathbf{R}) - \mathbf{H}\, \overline{\boldsymbol{\alpha}}||_2^2$, s.t. $\overline{\boldsymbol{\alpha}}$ on simplex    // projected grad descent
        put $\overline{\boldsymbol{\alpha}}$ to all columns of $\mathbf{A}$; $\mathbf{\Gamma} \leftarrow \mathbf{A} - diag(\mathbf{A}) + \mathbf{I}$
    end
until objective convergence;
```

To foster the intuition behind our method we provide links to other well-known learning problems and methods. First, we can rewrite the weighted inner product in the loss function (2) as $\langle \widetilde{\mathbf{v}}_{b,k}, \gamma_{b,k} \widetilde{\mathbf{x}}_{t,b} \rangle$. In this "feature learning" formulation the weights $\gamma_{b,k}$ act on the original inputs and, hence, generate new task-specific features $\mathbf{z}_{t,b}^{(k)} = \gamma_{b,k}\, \widetilde{\mathbf{x}}_{t,b}$. These are actually used in Step 1 of our algorithm 1. Alternatively, we can express the ridge penalty on $\mathbf{V}$ used in eq. (3) as $||\mathbf{V}||_F^2 = \sum_{b,k} ||\widetilde{\mathbf{v}}_{b,k}||_2^2 = \sum_{b,k} 1/\gamma_{b,k}^2 ||\widetilde{\mathbf{w}}_{b,k}||_2^2$. In this "adaptive ridge" formulation the elements of $\mathbf{\Gamma}$, which in our methods we learn, act as weights for the ℓ_2 regularization of $\mathbf{W}$. Equivalently, we can see this as the Bayesian maximum-a-posteriori with Guassian priors where the elements of $\mathbf{\Gamma}$ are the learned priors for the variance of the model parameters or (perhaps more interestingly) the random errors.

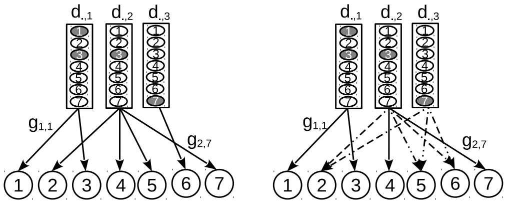
3.2. Leading Indicators for Clusters of Predictive Tasks

After explaining in section 3.1 the simplified case of learning a VAR with leading indicators for the whole system, we now move onto the more complex (and for larger VARs certainly more realistic) setting of the leading indicators being predictive only for parts of the system - clusters of predictive tasks.

To get started we briefly consider the situation in which the cluster structure (not the leading indicators) is known a priori. Here the models could be learned by a simple modification of algorithm 1 where in step 2 we would work with cluster-specific vectors $\overline{\boldsymbol{\alpha}}$ and matrices $\mathbf{H}$ and $\mathbf{R}$ constructed over the known cluster members. In reality the clusters are typically not known and therefore our CLVAR method is designed to learn them together with the leading indicators.

We use the same block decompositions of the input and parameter matrices $\mathbf{X}$ and $\mathbf{W}$, and the structural matrices $\boldsymbol{\Gamma} = \mathbf{A}+\mathbf{B} = \mathbf{A}-diag(\mathbf{A})+\mathbf{I}$ and the rescaled parameter matrix $\mathbf{V}$ defined in section 3.1. However, we need to alter the structural assumptions encoded into the matrix $\mathbf{A}$. In the cluster case $\mathbf{A}$ still shall have many rows equal to zero but it shall no longer have all the columns equal (same leading indicators for all the tasks). Instead, we learn it as a low rank matrix by factorizing it into two lower dimensional matrices $\mathbf{A} = \mathbf{DG}$: the $K \times r$ dictionary matrix $\mathbf{D}$ with the dictionary atoms (columns of $\mathbf{D}$) representing the cluster prototypes of the dependency structure; and the $r \times K$ matrix $\mathbf{G}$ with the elements being the per-model dictionary weights, $1 \leq r \leq K$.

To better understand the clustering effect of the low-rank decomposition, Fig. 2 illustrates it for an imaginary system of $K = 7$ time series with rank $r = 3$. The $\mathbf{d}_{.,j}$ j={1,2,3} columns in the top are the sparse cluster prototypes (the non-zero elements for the leading indicators are shaded). The circles in the bottom are the individual learning tasks and the arrows are the per-model dictionary weights $g_{i,j}$. Solid arrows have weight 1, missing arrows have weight zero, dashed arrows have weight between 0 and 1. So for example, the solid arrow from the 2nd column to the 7th circle in Fig. 2(*a*) is the $g_{2,7}$ element of matrix $\mathbf{G}$. Since it is a full arrow, it is equal to 1. The arrow from the 3rd column to the 2nd circle in Fig. 2(*b*) is the $g_{3,2}$ element of $\mathbf{G}$. Since the arrow is dashed, we have $0 < g_{3,2} < 1$.



(*a*) hard cluster assignments

(*b*) soft cluster assignments

Figure 2: Roles of $\mathbf{D}$ and $\mathbf{G}$ matrices in the low-rank decomposition $\mathbf{A}$

Fig. 2(*a*) uses a binary matrix $\mathbf{G}$ (no dashed arrows) reflecting hard clustering of the tasks consistent with our initial setting of a priori known clusters. Each task (circle at the bottom) is associated with only one cluster prototype (columns of $\mathbf{D}$ in the top). In contrast, Fig. 2(*b*) uses matrix $\mathbf{G}$ with elements between 0 and 1 to perform soft clustering of the tasks. Each task (circle at the bottom) may be associated with more than one cluster prototype (columns of $\mathbf{D}$ in the top). Our CLVAR is based on this latter approach of soft-clustering of the forecast tasks.

3.2.1. Learning Problem and Algorithm for CLVAR

We now adapt the minimisation problem (3) for the multi-cluster setting

$$\begin{aligned} &\text{argmin}_{\mathbf{D},\mathbf{G},\mathbf{V}} \ \textstyle\sum_{t=1}^{T} \sum_{k=1}^{K} (y_{t,k} - \sum_{b=1}^{K} (\sum_{j=1}^{K} d_{b,j} g_{j,k} + \beta_{b,k}) \langle \widetilde{\mathbf{v}}'_{b,k}, \widetilde{\mathbf{x}}'_{t,j} \rangle)^2 + \lambda ||\mathbf{V}||_F^2 \\ &\qquad\qquad \text{s.t. } \mathbf{1}'_K \, \mathbf{d}_{.,j} = \kappa; \ \mathbf{d}_{.,j} \geq \mathbf{0}; \ \mathbf{1}'_r \, \mathbf{g}_{.,j} = 1; \ \mathbf{g}_{.,j} \geq \mathbf{0}, \ \ \beta_{j,j} = 1 - \alpha_{j,j} \ \forall j \ \ . \end{aligned} \tag{4}$$

The relations of the optimisation matrices $\mathbf{D}, \mathbf{G}, \mathbf{V}$ to the parameter matrix $\mathbf{W}$ of the VAR model are as explained in the paragraphs above. The principal difference of the formulation (4) as compared to problem (3) is the low-rank decomposition of matrix $\mathbf{A} = \mathbf{DG}$ using the fact that $a_{b,k} = \sum_{j=1}^{K} d_{b,j} g_{j,k}$. Similarly as for the single column $\overline{\boldsymbol{\alpha}}$ in (3) we promote sparsity in the cluster prototypes $\mathbf{d}_{.,j}$ by constraining them onto the simplex. And we use the probability simplex constraints to sparsify the per-task weights in the columns of $\mathbf{G}$ so that the task are not based on all the prototypes.

```
Algorithm 2: CLVAR - VAR with leading indicators for clusters of predictive tasks
Input      : training data $\mathbf{Y}, \mathbf{X}$; hyper-parameters $\lambda, \kappa, r$
Initialise: $\mathbf{D}, \mathbf{G}$ evenly to satisfy the constraints; $\mathbf{A} \leftarrow \mathbf{DG}$; $\mathbf{\Gamma} \leftarrow \mathbf{A} - diag(\mathbf{A}) + \mathbf{I}$
repeat                                                        // Alternating descent
    begin Step 1: Solve for $\mathbf{V}$
        same as in algorithm 1
    end
    begin Step 2: Solve for $\mathbf{D}, \mathbf{G}$ and $\mathbf{\Gamma}$
        foreach task $k$ do
            same as in algorithm 1
            $\mathbf{g_{.,k}} \leftarrow \text{argmin}_{\mathbf{g}} ||\mathbf{r}_{.,k} - \mathbf{H}^{(k)} \, \mathbf{g}||_2^2$, s.t. $\mathbf{g}$ on simplex    // projected grad desc
        end
        concatenate vertically input product matrices $\mathbf{H} = vertcat(\mathbf{H}^{(.)})$
        expand matrices to match dictionary vectorization $\widehat{\mathbf{G}} \leftarrow \mathbf{G}' \otimes \mathbf{1}_T \mathbf{1}'_K$; $\widehat{\mathbf{H}} = \mathbf{1}'_r \otimes \mathbf{H}$
        $vec(\mathbf{D}) \leftarrow \text{argmin}_{\mathbf{D}} ||vec(\mathbf{R}) - \widehat{\mathbf{G}} \odot \widehat{\mathbf{H}} \, vec(\mathbf{D})||_2^2$        // projected grad desc
                                    s.t. $\mathbf{d}_{.,j}$ on simplex $\forall j$
        $\mathbf{A} = \mathbf{DG}$; $\mathbf{\Gamma} \leftarrow \mathbf{A} - diag(\mathbf{A}) + \mathbf{I}$
    end
until objective convergence;
```

We propose to solve problem (4) by alternating descent algorithm 2. While non-convex, the alternating approach for learning the low-rank matrix decomposition is known to perform well in practice and has been recently supported by new theoretical guarantees, e.g. Park et al. (2016). We solve the two sub-problems in step 2 by projected gradient descent with FISTA backtracking line search (Beck and Teboulle, 2009). The algorithm is $\mathcal{O}(T)$ for increasing number of observation and $\mathcal{O}(K^3)$ for increasing number of time series. However, one needs to bear in mind that with each additional series the complexity of the VAR model itself increases by $\mathcal{O}(K)$. Nevertheless, the expensive scaling with K is an important bottleneck of our method and we are investigating options to address it in our future work.

4. Related Work

We explained in section 2.2 how our search for leading indicators links to the Granger causality discovery in VARs. As shows the list of references in the survey of Liu and Bahadori (2012), this has been a rather active research area over the last several years. While the traditional approach for G-discovery was based on pairwise testing of candidate models or the use of model selection criteria such as AIC or BIC, inefficiency of such approaches for builidng predictive models of large time series system has long been recognised[3], e.g. Doan et al. (1984).

As an alternative, variants of so-called graphical Granger methods based on regularization for parameter shrinkage and thresholding (along the lines of Lasso (Tibshirani, 2007)) have been proposed in the literature. We use the two best-established ones, the lasso-Granger (VARL1) of Arnold et al. (2007) and the grouped-lasso-Granger (VARLG) of Lozano et al. (2009), as the state-of-the-art competitors in our experiments. More recent adaptations of the graphical Granger method address the specific problems of determining the order of the models and the G-causality simultaneously (Shojaie and Michailidis, 2010; Ren et al., 2013), the G-causality inference in irregular (Bahadori and Liu, 2012) and subsampled series (Gong et al., 2015), and in systems with instantaneous effects (Peters et al., 2013). However, neither of the above methods considers or exploits any common structures in the G-causality graphs as we do in our method.

Common structures in the dependency are assumed by Jalali and Sanghavi (2012) and Geiger et al. (2015) though the common interactions are with unobserved variables from outside the system rather then within the system itself. Also, the methods discussed in these have no clustering ability. Songsiri (2015) considers common structures across several datasets (in panel data setting) instead of within the dynamic dependencies of a single dataset. Huang and Schneider (2012) assume sparse bi-clustering of the G-graph nodes (by the in- and out- edges) to learn fully connected sub-graphs in contrast to our shared sparse structures. Most recently, Hong et al. (2017) proposes to learn clusters of series by Laplacian clustering over the sparse model parameters. However, the underlying models are treated independently not encouraging any common structures at learning.

More broadly, our work builds on the multi-task (Caruana, 1997) and structured sparsity (Bach et al., 2012) learning techniques developed outside the time-series settings. Similar block-decompositions of the feature and parameter matrices as we use in our methods have been proposed to promote group structures across multiple models (Argyriou et al., 2007; Swirszcz and Lozano, 2012). Although the methods developed therein have no clustering capability. Various approaches for learning model clusters are discussed in Bakker and Heskes (2003); Xue et al. (2007); Jacob et al. (2009); Kang et al. (2011); Kumar and Hal Daume III (2012) of which the latest uses similar low-rank decomposition approach as our method. Nevertheless, neither of these approaches learns sparse models and builds the clustering on similar structural assumptions as our method does.

[3] Due to the lack of domain knowledge to support the model selection and combinatorial complexity of exhaustive search.

5. Experiments

We present here the results of a set of experiments on synthetic and real-world datasets. We compare to relevant baseline methods for VAR learning: univariate auto-regressive model AR (though simple, AR is typically hard to beat by high-dimensional VARs when the domain knowledge cannot help to specify a relevant feature subset for the VAR model), VAR model with standard ℓ_2 regularisation VARL2 (controls over-parametrisation by shrinkage but does not yield sparse models), VAR model with ℓ_1 regularisation VARL1 (lasso-Granger of Arnold et al. (2007)), and VAR with group lasso regularisation VARLG (grouped-lasso-Granger of Lozano et al. (2009)). We implemented all the methods in Matlab using standard state-of-the-art approaches: trivial analytical solutions for AR and VARL2, FISTA proximal-gradient (Beck and Teboulle, 2009) for VARL1 and VARLG. The full code together with the datasets amenable for full replication of our experiments is available from `https://bitbucket.org/dmmlgeneva/var-leading-indicators`.

In all our experiments we simulated real-life forecasting exercises. We split the analysed datasets into training and hold-out sets unseen at learning and only used for performance evaluation. The trained models were used to produce one-step ahead forecasts by sliding through all the points in the hold-out. We repeated each experiments over 20 random re-samples. The reported performance is the averages over these 20 re-samples. The construction of the re-samples for the synthetic and real datasets is explained in the respective sections below. We used 3-folds cross-validation with mean squared error as the criterion for the hyper-parameter grid search. Unless otherwise stated below, the grids were: $\lambda \in$ 15-elements grid $[10^{-4} \dots 10^{3}]$ (used also for VARL2, VARL1 and VARLG), $\kappa \in \{0.5, 1, 2\}$, rank $\in \{1, 0.1K, 0.2K, K\}$. We preprocessed all the data by zero-centering and unit-standardization based on the training statistics only.

For all the experiments and all the tested methods we fixed the lag of the learned models to $p = 5$. While the search for the best lag p has in the past constituted an important part of time series modelling[4], in high-dimensional settings the exhaustive search through possible sub-set model combinations is clearly impractical. Modern methods therefore focus on using VARs with sufficient number of lags to cater for the underlying time dependency and apply Bayesian or regularization methods to control the model complexity, e.g. Koop (2013). In our case, this is achieved by the ridge shrinkage on the parameter matrix $\mathbf{V}$.

5.1. Synthetic Experiments

We designed six generating processes for systems varying by number of series and the G-causal structure. The first three are small systems with $K = 10$ series only, the next three increase the size to $K = \{30, 50, 100\}$. Systems 1 and 2 are unfavourable for our method, generated by processes not corresponding to our structural assumptions: in the 1st each series is generated from its own past only and therefore can be best modelled by a simple univariate AR model (the G-causal graph has no links); the 2nd is a fully connected VAR (all series are leading for the whole system). The 3rd system consists of 2 clusters with 5 series each, both depending on 1 leading indicator. Systems 4-6 are composed of $\{3, 5, 10\}$ clusters respectively, each with 10 series concentrated around 2 leading indicators[5].

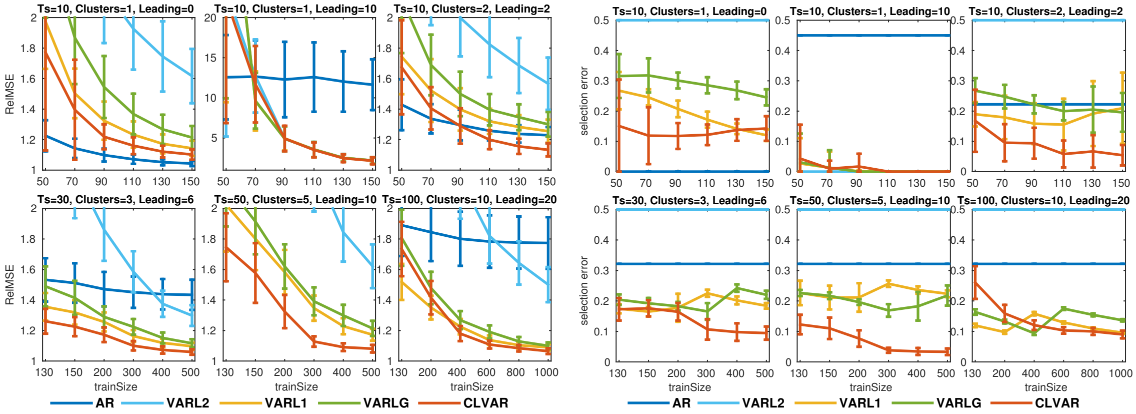
[4] Especially for univariate models within the context of the more general ARMA class (Box et al., 1994).

[5] For the last two, we fixed the rank in CLVAR training to the true number of clusters.

For each of the 6 system designs we first generated a random matrix of VAR coefficients with the required structure. We ensured the processes are stationary by controlling the roots of the model characteristic polynomials. We then generated 20 random realisation of the VAR processes with uncorrelated standard-normal noise. In each, we separated the last 500 observations into a hold-out set and used the previous T observations for training. Once trained, the same model was used for the 1-step-ahead forecasting of the 500 hold-out points by sliding forward through the dataset.

The predictive performance of the methods in the 6 experimental settings for multiple training sizes T is summarised in Fig. $3(a)$[6]. We measure the predictive accuracy by the mean square error of 1-step-ahead forecasts relative to the forecasts produced by the VAR with the true generative coefficients (RelMSE). Doing so we standardize the MSE by the irreducible error of each of the forecast exercises. The closer to 1 (the gold standard) the better. The plots display the average RelMSE over the twenty replications of the experiments, the error bars are at ±1 standard deviation.

In all the experiments the predictive performance improves with the increasing training size and the differences between the methods diminish. CLVAR outperforms all the other methods in the experiments with sparse structures as per our assumptions (mostly markedly). But CLVAR behaves well even in the unfavourable conditions of the first two systems. It still performs better than the other two sparse methods VARL1 and VARLG and the non-sparse VARL2 in the 1st completely sparse experiment[7], and it is on par with the other methods in the 2nd full VAR experiment.



(a) Relative MSE over true model

(b) Selection error of G-causal links

Figure 3: Results for synthetic experiments averaged over 20 experimental replications

In Fig. $3(b)$ we show the accuracy of the methods in selecting the true generative G-causal links between the series in the system. The selection error (the lower the better) is measured as the average of the false negative and false positive rates. We plot the averages with ±1 standard deviation over the 20 experimental replications. The CLVAR typically learned models structurally closer to the true generating process than the other tested methods, in most cases with substantial advantage.

[6] Numerical results behind the plots are listed in the Supplement.

[7] The AR model is in an advantage here since it has the true-process structure by construction.

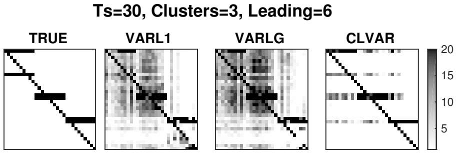


Figure 4: Synthesis of model parameters **W**

To better understand the behaviour of the methods in terms of the structure they learn, we chart in Fig. 4 a synthesis of the model matrices **W** learned by the sparse learning methods for the largest training size in the 4th system[8]. The displayed structures correspond to the schema of the **W** matrix presented in Fig.1. For the figure, the matrices were binarised to simply indicate the existence (1) or non-existence (0) of a G-causal link. The white-to-black shading reflects the number of experimental replications in which this binary indicator is active (equal to 1). So, a black element in the matrix means that this G-causal link was learned in all the 20 re-samples of the generating process. White means no G-causality in any of the re-samples. Though none of the sparse method was able to clearly and systematically recover the true structures, VARL1 and VARLG clearly suffer from more numerous and more frequent over-selections than CLVAR which matches the true structure more closely and with higher selection stability (fewer light-shaded elements).

Finally, we explored how the CLVAR scales with increasing sample size T and the number of time series K. The empirical results correspond to the complexity analysis of section 3.2: the run-times increased fairly slowly with increasing sample size T but were much longer for systems with higher number of series K. Further details are deferred to the Supplement. Overall, the synthetic experiments confirm the desired properties of CLVAR in terms of improved predictive accuracy and structural recovery.

5.2. Real-data Experiments

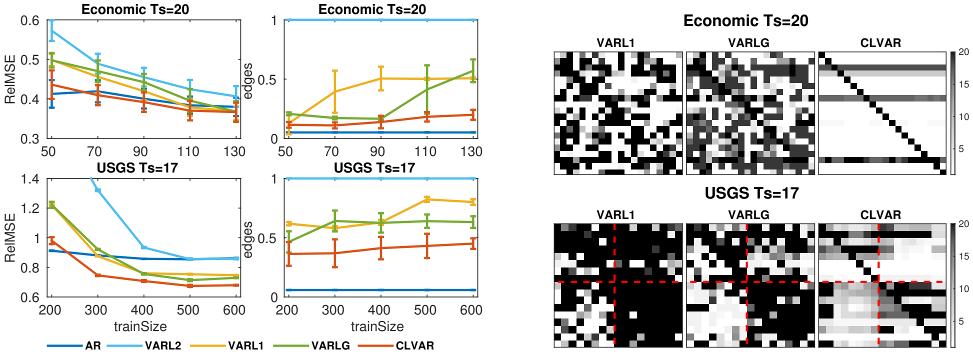
We used two real datasets very different in nature, frequency and length of available observations. First, an USGS dataset of daily averages of water physical discharge[9] measured at 17 sites along the Yellowstone (8 sites) and Connecticut (9 sites) river streams (source: Water Services of the US geological survey `http://www.usgs.gov/`). Second, an economic dataset of quarterly data on 20 major US macro-economic indicators of Stock and Watson (2012) frequently used as a benchmark dataset for VAR learning methods. More details on the datasets can be found in the Supplement.

We preprocessed the data by standard stationary transformations: we followed Stock and Watson (2012) for the economic dataset; by year-on-year log-differences for the USGS. For the short economic dataset, we fixed the hold-out length to 30 and the training sizes from 50 to 130. For the much longer USGS dataset, the hold-out is 300 and the training size increases from 200 to 600. The re-samples are constructed by dropping the latest observation from the data and constructing the shifted train and hold-out from this curtailed dataset.

The results of the two sets of experiments are presented in Fig. 5. The true parameters of the generative processes are unknown here. Therefore the predictive accuracy is measured in terms of the MSE relative to a random walk model (the lower the better), and the structural recovery is measured in terms of the proportion of active edges in the G-causal

[8] For space reasons, results for the other experiments are deferred to the Supplement.

[9] USGS parameter code 00060 - physical discharge in cubic feet per second.



(*a*) MSE and G-causal edges (*b*) Synthesis of parameters **W**

Figure 5: Results for real-data experiments averaged over 20 experimental replications

graph (the lower the better), always averaged across the 20 re-samples with ±1 standard deviation errorbar.

Similarly as in the synthetic experiments, the predictive performance improves with increasing training size and the differences between the methods get smaller. In both experiments, the non-sparse VARL2 has the worst forecasting accuracy (which corresponds to the initial motivation that real large time-series systems tend to be sparse). CLVAR outperformed the other two sparse learning methods VARL1 and VARLG in predictive accuracy as well as sparsity of the learned G-causal graphs. In the economic experiment, the completely (by construction) sparse AR achieved similar predictive accuracy. CLVAR clearly outperforms all the other methods on the USGS dataset.

Fig. 5(*b*) explores the effect of the structural assumptions on the final shape of the model parameter matrices **W** in the same manner as in Fig. 4. The CLVAR matrices are much sparser than the VARL1 and VARLG matrices, organised around a small number of leading indicators. In the economic dataset, the CLVAR method identified three leading indicators for the whole system. In the USGS dataset, the dashed red lines delimit the the Yellowstone (top-left) from the Connecticut (bottom-right) sites. In both these sets of experiments the recovered structure helped improving the forecasts beyond the accuracy achievable by the other tested learning methods.

6. Conclusions

We presented here a new method for learning sparse VAR models with shared structures in their Granger causality graphs based on the leading indicators of the system, a problem that had not been previously addressed in the time series literature.

The new method has multiple learning objectives: good forecasting performance of the models, and the discovery of the leading indicators and the clusters of series around them. Meeting these simultaneously is not trivial and we used the techniques of multi-task and structured sparsity learning to achieve it. The method promotes shared patterns in the structure of the individual predictive tasks by forcing them onto a lower-dimensional sub-space spanned by sparse prototypes of the cluster centres. The empirical evaluation confirmed the efficacy of our approach through favourable results of our new method as compared to the state-of-the-art.

References

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. *NIPS*, 2007.

Andrew Arnold, Yan Liu, and Naoki Abe. Temporal causal modeling with graphical granger methods. *Proceedings of the 13th ACM SIGKDD - KDD '07*, 2007.

Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured Sparsity through Convex Optimization. *Statistical Science*, 2012.

Mohammad Taha Bahadori and Yan Liu. On Causality Inference in Time Series. *2012 AAAI Fall Symposium Series*, 2012.

Bart Bakker and Tom Heskes. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 2003.

Amir Beck and Marc Teboulle. Gradient-based algorithms with applications to signal recovery. *Convex Optimization in Signal Processing and Communications*, 2009.

George E. P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. *Time Series Analysis: Forecasting and Control.* Prentice-Hall International, Inc., 3rd edition, 1994.

Rich Caruana. *Multitask Learning.* PhD thesis, Carnegio Mellon University, 1997.

Thomas Doan, Robert Litterman, and Christopher Sims. Forecasting and conditional projection using realistic prior distributions. *Econometric reviews*, 3(1):1–100, 1984.

Michael Eichler. Graphical modelling of multivariate time series. *Probability Theory and Related Fields*, 2012.

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. *Proceedings of the 10th ACM SIGKDD*, 2004.

P. Geiger, K. Zhang, M. Gong, D. Janzing, and B. Schölkopf. Causal Inference by Identification of Vector Autoregressive Processes with Hidden Components. *ICML*, 2015.

Mingming Gong, Kun Zhang, Dacheng Tao, Philipp Geiger, and Intelligent Systems. Discovering Temporal Causal Relations from Subsampled Data. In *ICML*, 2015.

CWJ W J Granger. Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica: Journal of the Econometric Society*, 1969.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 1970.

Dezhi Hong, Quanquan Gu, and Kamin Whitehouse. High-dimensional Time Series Clustering via Cross-Predictability. *AISTATS*, 2017.

TK Huang and Jeff Schneider. Learning bi-clustered vector autoregressive models. 2012.

Laurent Jacob, Francis Bach, and JP P Vert. Clustered multi-task learning: A convex formulation. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

Ali Jalali and Sujay Sanghavi. Learning the dependence graph of time series with latent factors. In *International Conference on Machine Learning (ICML)*, 2012.

Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with Whom to Share in Multi-task Feature Learning. In *ICML*, 2011.

Gary Koop. Forecasting with medium and large Bayesian VARs. *Journal of Applied Econometrics*, 203(28):177–203, 2013.

Abhishek Kumar and Hal Daume III. Learning task grouping and overlap in multi-task learning. In *International Conference on Machine Learning (ICML)*, 2012.

Yan Liu and MT Bahadori. A Survey on Granger Causality: A Computational View. Technical report, University of Southern California, 2012.

Aurélie C. Lozano, Naoki Abe, Yan Liu, and Saharon Rosset. Grouped graphical Granger modeling for gene expression regulatory networks discovery. *Bioinformatics*, 2009.

Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer, 2005.

D. Park, A. Kyrillidis, C. Caramanis, and S. Sanghavi. Finding low-rank solutions to smooth convex problems via the Burer-Monteiro approach. In *Allerton Conference on Communication, Control, and Computing*, 2016.

Judea Pearl. *Causality*. Cambridge University Press, 2009.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Causal Inference on Time Series using Restricted Structural Equation Models. In *NIPS*, 2013.

Yunwen Ren, Zhiguo Xiao, and Xinsheng Zhang. Two-step adaptive model selection for vector autoregressive processes. *Journal of Multivariate Analysis*, 2013.

Ali Shojaie and George Michailidis. Discovering graphical Granger causality using the truncating lasso penalty. *Bioinformatics (Oxford, England)*, 2010.

Jitkomut Songsiri. Sparse autoregressive model estimation for learning Granger causality in time series. In *Proceedings of the 38th ICASSP*, 2013.

Jitkomut Songsiri. Learning Multiple Granger Graphical Models via Group Fused Lasso. In *IEEE Asian Control Conference (ASCC)*, 2015.

James H. Stock and Mark W. Watson. Generalized Shrinkage Methods for Forecasting Using Many Predictors. *Journal of Business & Economic Statistics*, 2012.

Grzegorz Swirszcz and Aurelie C Lozano. Multi-level Lasso for sparse multi-task regression. In *International Conference on Machine Learning (ICML)*, 2012.

Robert Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 2007.

Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, 2007.