Published in Proceedings of the Time Series Workshop of the 29th Neural Information Processing Systems conference, NIPS-2015 which should be cited to refer to this work

Functional learning of time-series models preserving Granger-causality structures

Magda Gregorova
HES-SO & University of Geneva, Switzerland
magda.gregorova@ungie.ch

Francesco Dinuzzo
Amazon
francesco.dinuzzo@amazon.com

Alexandros Kalousis
HES-SO & University of Geneva, Switzerland
alexandros.kalousis@unige.ch

Abstract

We develop a functional learning approach to modelling systems of time series which preserves the ability of standard linear time-series models (VARs) to uncover the Granger-causality links in between the series of the system while allowing for richer functional relationships. We propose a framework for learning multiple output-kernels associated with multiple input-kernels over a structured input space and outline an algorithm for simultaneous learning of the kernels with the model parameters with various forms of regularization including non-smooth sparsity inducing norms. We present results of synthetic experiments illustrating the benefits of the described approach.

1 Introduction

We consider the problem of learning m functions $f_j : \mathcal{X}_1 \times \ldots \times \mathcal{X}_m \to \mathcal{Y}_j$ for one-step-ahead predictions of m time series from the past evolution of the m-dimensional time-series system. The time series are observed at synchronous equidistant time points and the observations are arranged into T sequential input-output pairs $\{\big([\mathbf{x}'_{1t}, \ldots, \mathbf{x}'_{mt}]', y_{jt}\big) : j \in \mathbb{N}_m, t \in \mathbb{N}_T\}$, where $y_{jt} \in \mathcal{Y}_j = \mathbb{R}$ is the observation of time series j at time point t, and $\mathbf{x}_{it} \in \mathcal{X}_i = \mathbb{R}^p$ is the vector of p latest observations of series i preceding the time point t.

The m functions share a common input space $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_m = \mathbb{R}^{d=mp}$ and therefore we combine them into a single-vector valued function $\mathbf{f} : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y} = \cup_{j \in \mathbb{N}_m} \mathcal{Y}_j \in \mathbb{R}^m$ is the joint output space. In the input-output pairs $\{\big(\mathbf{x}_t, \mathbf{y}_t\big) : t \in \mathbb{N}_T\}$, we concatenate the m series so that $\mathbf{x}_t = [\mathbf{x}'_{1t}, \ldots, \mathbf{x}'_{mt}]' \in \mathcal{X} = \mathbb{R}^d$ and $\mathbf{y}_t = [y_{1t}, \ldots, y_{mt}]' \in \mathcal{Y} = \mathbb{R}^m$. For linear functions f_j this is the well known vector autoregressive model (VAR), e.g. [1].

Following the standard function-learning theory ([2]) we propose to learn $\mathbf{f}$ in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ of $\mathcal{Y}$-valued functions endowed with a norm $||.||_{\mathcal{H}}$ and associated with a positive-definite matrix-valued kernel $\mathbf{H}(.,.) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{m \times m}$.

Specifically, we focus on the functional spaces associated with the class of sum of separable kernels [3] used frequently in spatio-temporal modelling as approximations of the more general non-separable kernels (e.g. [4])

$$\mathbf{H}(\mathbf{x}_t, \mathbf{x}_{t'}) = \sum_b^B k^b(\mathbf{x}_t, \mathbf{x}_{t'})\, \mathbf{L}^b, \tag{1}$$

where each $k^b : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive-definite scalar-valued kernel measuring the similarity between the inputs (input-kernel), and $\mathbf{L}^b \in \mathbb{S}_+^m$ is a positive-semidefinite matrix encoding the relationships between the outputs (output-kernel).

Since specifying the appropriate kernels is far from trivial and may lead to serious performance degradation if chosen wrongly, we propose to learn the output-kernels $\mathbf{L}^b$ together with the function $\mathbf{f}$ by solving the joint regularized problem[1].

$$\underset{f\in\mathcal{H}, L^b\in\mathcal{S}_+^m}{\arg\min} \quad \sum_t^T \frac{1}{2\lambda}||\mathbf{y}_t - \mathbf{f}\big(\mathbf{x}_t\big)||_2^2 + \frac{1}{2}||\mathbf{f}||_{\mathcal{H}}^2 + R(\sum_b^B \mathbf{L}^b), \tag{2}$$

While the problem of learning function $\mathbf{f}$ together with a single output-kernel $\mathbf{L}$ has previously been addressed (e.g. [5],[6],[7]), we are not aware of any existing method learning the multiple output-kernel matrices $\mathbf{L}^b$ in the separable kernels (1). Such simultaneous learning is a contribution of our paper.

Learning multiple $\mathbf{L}^b$'s greatly increases the flexibility of the models as compared to a single output-kernel $\mathbf{L}$. However, the increased complexity calls for a strong regularization via $R(.)$. In this paper we explore the problem of learning the output kernels with the search space limited to PSD diagonal matrices $\mathbf{L}^b \in \mathbb{D}_+^m$ in combination with convex yet possibly non-smooth regularization functions $R(.)$ such as those inducing sparsity [8].

Further, to preserve the ability of our method to capture the Granger causality [9] within the tine-series system that exists in the standard linear models, we propose to use the input-kernels in (1) in the form

$$k^b(\mathbf{x}_t, \mathbf{x}_{t'}) = \kappa^{g_i}(\mathbf{x}_{it}, \mathbf{x}_{it'}), \qquad |\{g_i\}| = B, \tag{3}$$

where each $\kappa^{g_i} : \mathcal{X}_i \times \mathcal{X}_i \to \mathbb{R}$ is a positive-definite scalar-valued kernel with a functional form g operating on the subspace $\mathcal{X}_i$ of the input space $\mathcal{X}$ related to the past of a single time series i. With this input-kernel structure, sparsity promoting regularization over the diagonal output-kernels $\mathbf{L}^b$ encourages learning sparse Granger causality graphs of the systems.

The above formulation is also flexible enough to accommodate various combinations of kernels with varying complexity - from the most straightforward of using a single functional form g for the kernels of all the input subspaces (such as simple linear kernels) to having multiple kernels for each $\mathcal{X}_i$ with possibly different functional forms g across the subspaces. It is thus possible to model differing functional relationships between the input subspaces (single time series history) and the individual outputs (single time series future) within a single model. For example, uneven length in the past dependency can be factored into the model by using decaying input kernels κ^{g_i} with varying speed of decay (e.g. [10]).

2 Learning strategy

Using the representer theorem [2] and introducing the input-kernel gram matrices $\mathbf{K}^b \in \mathbb{R}^{T\times T}$: $K_{i,j}^b = k^b(\mathbf{x}_i, \mathbf{x}_j)$, the output matrix $\mathbf{Y} \in \mathbb{R}^{T\times m}$ and the parameters matrix $\mathbf{C} \in \mathbb{R}^{T\times m}$ we can rewrite problem (2) as an equivalent finite-dimensional problem

$$\underset{C\in\mathbb{R}^{T\times m}, L^b\in\mathcal{D}_+^m}{\arg\min} \frac{||\mathbf{Y} - \sum_b^B \mathbf{K}^b\mathbf{C}\mathbf{L}^b||_F^2}{2\lambda} + \frac{\sum_b^B \langle \mathbf{C}'\mathbf{K}^b\mathbf{C}, \mathbf{L}^b\rangle_F}{2} + R(\sum_b^B \mathbf{L}^b) \tag{4}$$

Instead of solving for multiple diagonal $\mathbf{L}^b$'s we gather the diagonals as columns into a non-negtive matrix $\mathbf{\Theta} = [diag(\mathbf{L}^1), \ldots, diag(\mathbf{L}^B)] \in \mathbb{R}_+^{m\times B}$ and formulate an equivalent learning problem

$$\underset{C\in\mathbb{R}^{n\times m}, \Theta\in\mathbb{R}_+^{m\times B}}{\arg\min} \sum_j^m \Big(\frac{||\mathbf{Y}_{:j} - \sum_b^B \theta_{jb}\mathbf{K}^b\mathbf{C}_{:j}||_2^2}{2\lambda} + \frac{\sum_b^B \theta_{jb}\mathbf{C}'_{j:}\mathbf{K}^b\mathbf{C}_{:j}}{2} \Big) + \Omega(\mathbf{\Theta}), \tag{5}$$

where $\mathbf{C}_{j:}$ and $\mathbf{C}_{:j}$ indicate respectively the jth row and column of matrix $\mathbf{C}$, and $\Omega(\mathbf{\Theta})$ is an equivalent transformation of $R(\sum_b^B \mathbf{L}^b)$ from problem (4).

[1] We simplify the notation here by writing $\mathbf{L}^b$ instead of the set $\{\mathbf{L}^b : b \in \mathcal{N}_B\}$

In this reformulated objective (5) the elements of $\mathbf{\Theta}$ act as weights on the input kernels $\mathbf{K}^b$ (which makes our method closely related to multiple-kernel learning) and therefore sparsity in $\mathbf{\Theta}$ translates into sparsity in the Granger causality graph.

To accommodate various forms of convex though possibly non-smooth regularizers Ω (such as some sparsity inducing norms - examples in section 3 and some further possibilities in section 4) together with the non-negativty constraint on $\mathbf{\Theta}$ we propose to solve problem (5) by the alternating direction method of multipliers (ADMM), [11]. The updates for the parameters matrix $\mathbf{C}$ are the solutions to a system of Sylvester's equations, the update for $\mathbf{\Theta}$ is the proximal operator, and the update for the auxiliary matrix is an NNQP (see section A in the Appending for details).

3 Experiments

To assess the performance of our method and to understand its strengths and weaknesses we have conducted a set of controlled experiments on a synthetic data set. (The section below summarises the main findings, more details on the experimental set-up and results can be found in appendix B.)

We have generated a random realisation from a stable linear VAR with 5 time series. The Grenger-causality structure of the system is sparse (only the 3rd and 5th time series serve as leading indicators for the system), and past dependency varies between 3 to 5 lags. For learning, we set the number of lags to $d_i = 5$ for all the series and model the system as fully dependent in terms of the Granger causality (mimicking the real situation when the true dependency is unknown).

We compare the performance of four variants differing by the complexity of the input kernels κ^{g_i} and the regularizer Ω: KVARLIN - for each input subspace i we use a simple linear kernel ($\kappa^{(lin_i)}$); KVARDEC - for each subspace we use a linear kernel with a decay ($\kappa^{(dec_i)}$); KVARCMB - each subspace uses both the kernels $\kappa^{(lin_i)}$ and $\kappa^{(dec_i)}$. The regularizer Ω for all three is a simple entry-wise ℓ_1 norm. The fourth model KVARSTR uses the same combination of input kernels as KVARCMB but we introduce more domain knowledge into the regularizer Ω by using ℓ_1 only for the off-diagonal elements and applying ℓ_2 on the diagonal (hence self dependency is preserved).

We have followed a standard learning and evaluation procedure: the performance is measured on the *unseen* hold-out sample with models trained on a separate training sample on which also the respective hyper-parameters were tuned by 5-folds cross validation.

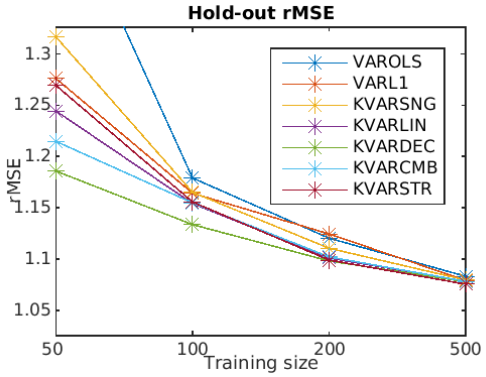

Figure 1: Hold-out forecast root MSE

Figure 1 summarizes the 1-step-ahead forecasting performance of our methods in terms of the average hold-out root MSE for training samples of different sizes against the baselines of a simple linear VAR model fitted by OLS (VAROLS), linear VAR with standard lasso penalty [12] (VARL1) and learning with a single linear input-kernel and a single diagonal output-kernel KVARSNG (i.e. without partitioning the input space). [2]

While for large sample sizes all the methods predict equally well, there are more differences in the small sample size. Overall, KVARDEC performs the best closely followed by the KVARCMB. These two models benefit from the decaying kernels to counter the initial lag-misspecification. While KVARDEC relies on the prior knowledge of the analyst and works only over the decaying kernels, the KVARCMB is less demanding on the initial kernel specification and leverages the multiple kernel selection. It seems, that adding more structure into the regularizer Ω does not bring much benefit as the

[2] Other simple baselines such as predicting mean, random walk or fitting simple AR model performing much worse on this dataset are not included in the graph for clarity of display but are included in the appendix B.

KVARSTR performs worse than all the three methods with the simple ℓ_1 norm. All the simpler linear models oblivious to the input-space structure struggle more in the small sample setting.

Figure 2 compares the Granger-causality structure of the learned models with the true model structure. The figure shows the heatmaps of the model parameters where a model for each output series is a column and the parameters associated with the individual input series are the rows. For example, the dark square at position (3,5) means that there is a strong Granger-causality link from time series 3 to series 5 (the past of series 3 helps in predicting the future of series 5).

Similarly as with the forecasting performance we see that most of the methods manage to recover the Granger causality fairly well for the largest training sample while this is much less obvious for the small training size, where KVARDEC and KVARCMB are closest to the true structure. Note that KVARSNG is not presented in the figure as it by construction does not have the capability to discover the Granger-causality structure.

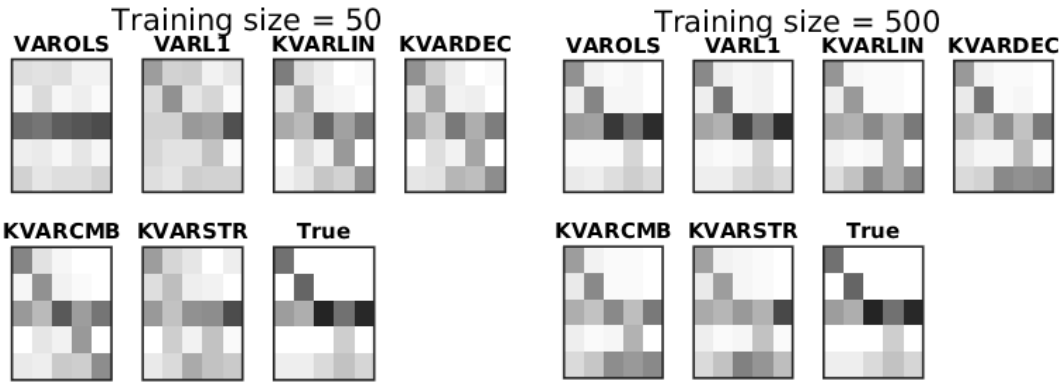

Figure 2: Granger causality structures illustrated by heatmaps of the learned model parameters. A column in the matrix is a model for an output series, rows are parameters assoicated with each series as inputs.

4 Conclusions and future work

We present a new approach to kernel learning for time series forecasting which preserves the input-space structure and therefore allows for uncovering the Granger-causality links in between the series in the system together with learning the forecasting model. We achieve this by using the matrix-valued kernel in the form of a *sum of separable kernels* and learning multiple output-kernels, one for each of the input-kernels operating over an input subspace.

In this paper we limit the output-kernel learning to diagonal matrices. We propose an algorithm for solving the optimisation problem with various forms of convex but possibly non-smooth regularizers and present results of an experiment on synthetic dataset which confirms the usefulness of our multiple input&output-kernel approach.

In the presented experiment we've tested only very simple regularizers (ℓ_1 and ℓ_2 norms). It remains to be explored if using more sophisticated and structured regularization (e.g. along the lines of group lasso [13], exclusive lasso [14] or some low rank structures) would benefit the model learning. Also (and more importantly), we would like to lift the diagonality constraint on the output-kernels (which essentially breaks the contemporaneous links between the models) and explore methods for learning a set of non-diagonal output-kernels. Here the regularization $R(.)$ plays obviously a crucial role.

References

[1] Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer-Verlag Berlin Heidelberg, 2005.

[2] Charles a Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural computation*, 17(1):177–204, jan 2005.

[3] Mauricio a. Alvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for Vector-Valued Functions: a Review. pages 1–37, 2011.

[4] Jaakko Luttinen and Alexander Ilin. Efficient Gaussian Process Inference for Short-Scale Spatio-Temporal Modeling. *International Conference on Artificial Intelligence and Statistics*, XX:741–750, 2012.

[5] Vikas Sindhwani, Ha Quang Minh, and Aurelie Lozano. Scalable Matrix-valued Kernel Learning for High-dimensional Nonlinear Multivariate Regression and Granger Causality. In *UAI*, 2014.

[6] Francesco Dinuzzo. Learning output kernels for multi-task problems. *Neurocomputing*, 118:119–126, 2013.

[7] Néhémy Lim, Florence DAlché-Buc, Cédric Auliac, and George Michailidis. Operator-valued Kernel-based Vector Autoregressive Models for Network Inference. *Machine Learning*, 99(3):1–22, 2014.

[8] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured Sparsity through Convex Optimization. *Statistical Science*, 27(4):450–468, nov 2012.

[9] CWJ Granger. Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica: Journal of the Econometric Society*, 37(3):424–438, 1969.

[10] Gianluigi Pillonetto, Francesco Dinuzzo, Tianshi Chen, Giuseppe De Nicolao, and Lennart Ljung. Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3):657–682, 2014.

[11] Stephen Boyd. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

[12] Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.

[13] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[14] Deguang Kong, Ryohei Fujimaki, Ji Liu, Feiping Nie, and Chris Ding. Exclusive Feature Learning on Arbitrary Structures via 1 , 2 -norm. *NIPS*, (1):1–9, 2014.

Appendix

A ADMM

We reformulate the minimisation problem (5) into an equivalent problem with an auxiliary variable $\mathbf{A}$ and an additional equality constraint

$$\begin{aligned} \arg\min_{C,\Theta,A} \quad & \sum_{j}^{m} \Big(\frac{||\mathbf{Y}_{:j} - \sum_b^B \alpha_{jb} \mathbf{K}^b \mathbf{C}_{:j}||_2^2}{2\lambda} + \frac{\sum_b^B \alpha_{jb} \mathbf{C}'_{j:} \mathbf{K}^b \mathbf{C}_{:j}}{2} \Big) + \Omega(\mathbf{\Theta}) \\ \text{s.t.} \quad & \mathbf{A} \geq 0, \quad \mathbf{A} - \mathbf{\Theta} = \mathbf{0} \end{aligned} \tag{6}$$

with $\mathbf{C} \in \mathbb{R}^{n \times m}$, $\mathbf{\Theta}, \mathbf{A} \in \mathbb{R}^{m \times B}$.

The updates at each iteration are (in the scaled-dual-variable version)

$$\mathbf{C}^{k+1} = \arg\min_{C} \mathcal{L}(\mathbf{C}, \mathbf{A}^k) \tag{7}$$

$$\mathbf{\Theta}^{k+1} = \arg\min_{\Theta} \Omega(\mathbf{\Theta}) + \frac{\rho}{2} ||\mathbf{A}^k + \mathbf{U}^k - \mathbf{\Theta}||_F^2 \tag{8}$$

$$\mathbf{A}^{k+1} = \arg\min_{A \geq 0} \mathcal{L}(\mathbf{C}^{k+1}, \mathbf{A}) + \frac{\rho}{2} ||\mathbf{A} + \mathbf{U}^k - \mathbf{\Theta}^{k+1}||_F^2 \tag{9}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \mathbf{A}^{k+1} - \mathbf{\Theta}^{k+1} \tag{10}$$

The $\mathbf{C}$-update is best obtained column-wise as solutions to the system of equations

$$\Big(\sum_b^B \alpha_{jb}^k \mathbf{K}^b + \lambda \mathbf{I} \Big) \mathbf{C}_{:j}^{k+1} = \mathbf{Y}_{:j}, \qquad \forall j \in \mathbb{N}_m, \tag{11}$$

For $\mathbf{\Theta}$ the update is a proximal operator of $\Omega(.)$ around $(\mathbf{A}^k + \mathbf{U}^k)$. The optimality condition depends on $\Omega(.)$ but will in general be

$$0 \in \partial\Omega(\mathbf{\Theta}^{k+1}) + \rho(\mathbf{\Theta}^{k+1} - \mathbf{A}^k - \mathbf{U}^k) \tag{12}$$

The algorithm can accommodate various forms of the regularizer $\Omega(.)$ as long as the associated proximal operator $prox_{\frac{1}{\rho}\Omega}(.)$ is supplied to it.

The $\mathbf{A}$-update is obtained row-wise by solving the non-negative quadratic programmes (NNQP)

$$(\mathbf{A}_{j:}^{k+1})' = \arg\min_{\mathbf{a} \geq 0} \frac{1}{2} \mathbf{a} \mathbf{H} \mathbf{a} + \mathbf{g}' \mathbf{a}, \quad \forall j \in \mathbb{N}_m, \tag{13}$$

where

$$\begin{aligned} \mathbf{H} &= {}_{\mathbf{j}}\mathbf{\Psi}'\, {}_{\mathbf{j}}\mathbf{\Psi} + \rho\lambda \mathbf{I}_B \\ \mathbf{g}' &= -\mathbf{y}'\, {}_{\mathbf{j}}\mathbf{\Psi} + \frac{\lambda}{2} {}_{\mathbf{j}}\mathbf{z}' + \rho\lambda(\mathbf{U}_{j:}^k - \mathbf{\Theta}_{j:}^{k+1}) \\ {}_{\mathbf{j}}\mathbf{\Psi} &= \begin{bmatrix} \mathbf{K}^1 \mathbf{C}_{:j} & \mathbf{K}^2 \mathbf{C}_{:j} & \cdots & \mathbf{K}^B \mathbf{C}_{:j} \end{bmatrix}, \quad {}_{j}\Psi_{ib} = \mathbf{K}_{i:}^b \mathbf{C}_{:j} \\ {}_{\mathbf{j}}\mathbf{z} &= \begin{bmatrix} \mathbf{C}'_{:j} \mathbf{K}^1 \mathbf{C}_{:j} \\ \mathbf{C}'_{:j} \mathbf{K}^2 \mathbf{C}_{:j} \\ \vdots \\ \mathbf{C}'_{:j} \mathbf{K}^B \mathbf{C}_{:j} \end{bmatrix} \qquad {}_{j}z_b = \mathbf{C}'_{:j} \mathbf{K}^b \mathbf{C}_{:j} \end{aligned}$$

Finally, the $\mathbf{U}$-updates for the scaled dual variable are

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \mathbf{A}^{k+1} - \mathbf{\Theta}^{k+1} \tag{14}$$

To improve convergence we also use the over-relaxation strategy [11] and replace $\mathbf{\Theta}^{k+1}$ with $\beta\mathbf{\Theta}^{k+1} + (1-\beta)\mathbf{A}^k$ with $\beta \in [1.5, 1.8]$ in the last two update steps (13) and (14).

B Experiments

B.1 Data and experimental protocol

For the experimental evaluation of our method we've generated a 5-dimensional stable linear VAR

$$\mathbf{y}'_t = \mathbf{y}'_{t-1}A_1 + \mathbf{y}'_{t-2}A_2 + \mathbf{y}'_{t-3}A_3 + \mathbf{y}'_{t-4}A_4 + \mathbf{y}'_{t-5}A_5 + \mathbf{e}'_t, \tag{15}$$

with an i.i.d. $\mathbf{e}_t \sim \mathcal{MN}(\mathbf{0}, I)$ and the parameter matrices

$$\begin{aligned}
A_1 &= \begin{bmatrix} 0.96 & 0 & 0 & 0 & 0 \\ 0 & 0.73 & 0 & 0 & 0 \\ -0.6 & -0.79 & 1.005 & 1.26 & 9.12 \\ 0 & 0 & 0 & -0.02 & 0 \\ 0.1 & 0.15 & -0.28 & -0.24 & -1.78 \end{bmatrix} \\
A_2 &= \begin{bmatrix} 0.16 & 0 & 0 & 0 & 0 \\ 0 & 0.38 & 0 & 0 & 0 \\ -0.15 & -0.19 & 0.91 & 0.31 & 2.22 \\ 0 & 0 & 0 & 0.71 & 0 \\ 0.08 & 0.12 & -0.23 & -0.2 & -0.65 \end{bmatrix} \\
A_3 &= \begin{bmatrix} -0.27 & 0 & 0 & 0 & 0 \\ 0 & -0.27 & 0 & 0 & 0 \\ 0.22 & 0.28 & -0.8 & -0.45 & -3.3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.04 & 0.15 & 0.27 \end{bmatrix} \\
A_4 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0 & 0 \\ 0 & 0 & -0.3 & 0 & -0.19 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & 0.12 \end{bmatrix} \\
A_5 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & -0.15 & 0 & 0.12 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned} \tag{16}$$

We have separated the last 50 observations into the hold-out sample for forecast evaluation and used the previous data-points to create training samples of length 50, 100, 200 and 500 respectively.

In each of the training samples we've used 5-folds cross validation to tune the hyperparameter λ for the regularized methods. The grids were unified for all the tested methods to length 20, logarithmically spaced in $4 * \{10^{-6}, \ldots, 10\}$.

B.2 Results

We have evaluated the predictive performance of the methods by measuring the root mean squared error (rMSE) of one-step-ahead predictions on the hold-out sample

$$rMSE = \sqrt{\sum_{j=1}^{5}\sum_{t=1}^{50} ||\hat{y}_{jt} - y_{jt}||_2^2/(5 * 50)} \tag{17}$$

and calculated the standard deviation in the rMSE across the holdout samples. Table 1 lists the two measures for all the compared methods and training samples.

We have based the comparisons of the ability of the methods to uncover the underlying Granger-causality structure on the learned parameters of the models. Because the parameter matrices of the linear model (15) are only directly comparable with the baseline linear models VAROLS and VARL1, we've used a proxy measure summarising and normalising the matrices as follows: for the

Table 1: Summary comparison of experimental results

	rMSE				Hold-out std(rMSE)			
	50	100	200	500	50	100	200	500
Mean	8.050	8.050	8.050	8.050	3.801	3.801	3.801	3.801
RW	13.77	13.77	13.77	13.77	7.192	7.192	7.192	7.192
AR	5.301	5.064	4.783	4.799	2.477	2.410	2.384	2.411
VAROLS	1.472	1.179	1.120	1.083	0.456	0.321	0.347	0.329
VARLIN	1.276	1.164	1.124	1.079	0.403	0.359	0.350	0.330
KVARSNG	1.317	1.165	1.110	1.080	0.408	0.346	0.348	0.330
KVARLIN	1.244	1.154	1.101	1.078	0.336	0.321	0.324	0.321
KVARDEC	1.186	1.133	1.098	1.078	0.328	0.330	0.317	0.325
KVARCMB	1.215	1.154	1.102	1.078	0.325	0.324	0.316	0.325
KVARSTR	1.269	1.156	1.099	1.076	0.345	0.321	0.335	0.322

linear models, we have calculated an overall dependency matrix $\mathbf{D}^{VAR}$ as

$$\mathbf{D}^{VAR} = \sum_{i=1}^{5} |\mathbf{A}_i| \tag{18}$$

for the kernel models (5) we have calculated the dependency matrix $\mathbf{D}^{KVAR}$ as

$$\mathbf{D}_{ij}^{KVAR} = \sum_{g} \theta_{ig_j} \tag{19}$$

For both, we have then normalized each column (corresponding to function f_j) to sum to 1. Heatmaps of these dependency matrices for the two sample sizes not included in the main text are in figure 3.

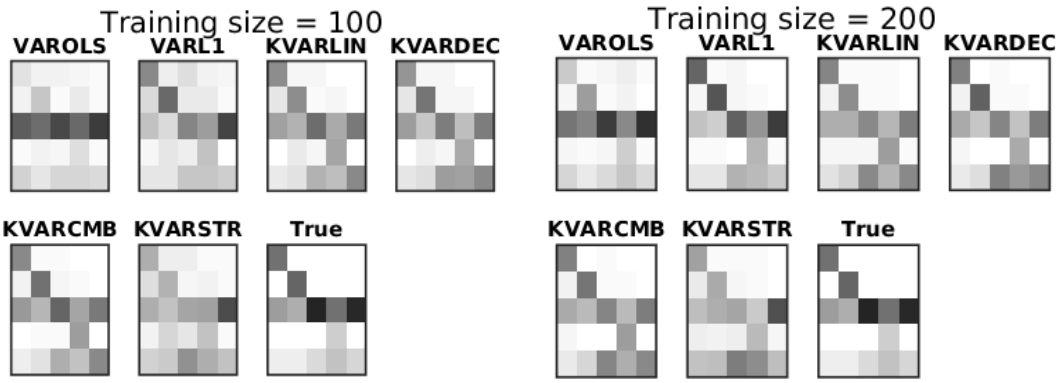

Figure 3: Granger causality structures illustrated by heatmaps of the learned model parameters. A column in the matrix is a model for an output series, rows are parameters assoicated with each series as inputs.

The distances (Frobenious norms) between the dependency matrices of the learned models and the true dependency matrix calculated from the parameters listed in (16) are summarised in table 2

Table 2: Summary of Granger-causality structure distance

	Dependency matrices distance			
	50	100	200	500
VAROLS	0.807	0.729	0.516	0.245
VARL1	0.773	0.579	0.409	0.227
KVARLIN	0.679	0.651	0.788	0.793
KVARDEC	0.764	0.745	0.793	0.822
KVARCMB	0.609	0.622	0.772	0.812
KVARSTR	0.719	0.805	0.887	0.778