Graphical Models and Regularization for Sparse Dynamic Network Identification

Xiaozhu Fang
School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen
xiaozhufang@link.cuhk.edu.cn

Abstract

In this work, we explore the learning problem of the large dynamic network, which is composed of sparse nodes. The objective is predicting the time series data of one node in the networks with respect to other nodes. This task becomes difficult if the number of nodes rises up, but is feasible for the sparse case. Here we focus on the autoregressive model because its maximum entropy estimate can be embedded with sparsity on its precision matrix, which is connected to the conditional independence in the undirected graphs. It means if the undirected graph is given and sparse, this prior knowledge can be embedded as a lasso-like penalty, improving the perdition significantly. In the simulation, we use synthetic data and real-world data to successfully valid this new estimate.

1 Introduction

There are lots of signals around us, represented as discrete time series. Some signals are not completely random but motivated by some mechanism, displaying a regular principle. Learning such principles is necessary for many purposes, e.g., smoothing, filtering, predicting, etc. The learning problems can be divided into three categories by data structure: 1) the spatial (or static) problem, e.g., images; 2) the temporal (or dynamic) problems, e.g., audio; 3)the spatial-temporal problem, e.g., video. In this paper, we concentrate on the last one.

Spatial-temporal learning is the sequence to sequence learning in the machine learning perspective. There are two comprehensive and popular methods. The first is the deep neural networks, e.g., recurrent neural network (RNN), long short-term memory (LSTM). The second is based on Gaussian processes. Although deep learning is very powerful for most cases, it lacks the reality and robustness, which are the primary barriers to industrial implementation.

Apart from the deep networks, the Gaussian processes associated with the graph model have been also proven powerful in huge networks [8]. The graphs are significantly simplified by omitting the edge for two conditional independent nodes. However, most graphical models and theorems are based on static graphs. Here we would extend that to the famous spatial-temporal model, autoregressive processes (AR), to see how the time domain makes a large difference.

2 Relative work

The spatial-temporal learning problems have been discussed for decades. The typical applications includes Computer Vision [13], Geology [4], Epidemiology [9], Economy [10]. Moreover, it is always a large problem in the control community when people try to model the complex and large dynamic networks, e.g. traffic networks [14], smart grid [1], etc.

The AR processes identification has a long history, but the modification of sparsity for the large dynamics is dated to Songsiri's work, see [11, 12], which is also the foundation of this paper.

Songsiri's works focus on identifying AR model itself, but this paper extends it to sequence to sequence problems. In the simulation, we also use another the state of the art, kernel-based ridge estimate, which can be founded in [3]. Besides, there are new developments of AR identification for large networks via other techniques, see [17, 15, 16].

The code is almost independently written by the author except for the CVX toolbox in MATLAB [2]. The code is available on Github with the link, `https://github.com/XiaozhuFang/Graphical-Models-and-Regularization-for-Sparse-Dynamic-Network-Identification`

3 Background

In what follows we would show the difference between the static problem and the dynamic problem for the graphical models. The representative models of the two problems are undirected graphs and the Markov chain. Note the Markov chain is the discrete type of the Markov processes with countable nodes. Besides, we also have the AR process with order 1, denoted as AR(1) process. Their difference is illustrated in Fig.1.

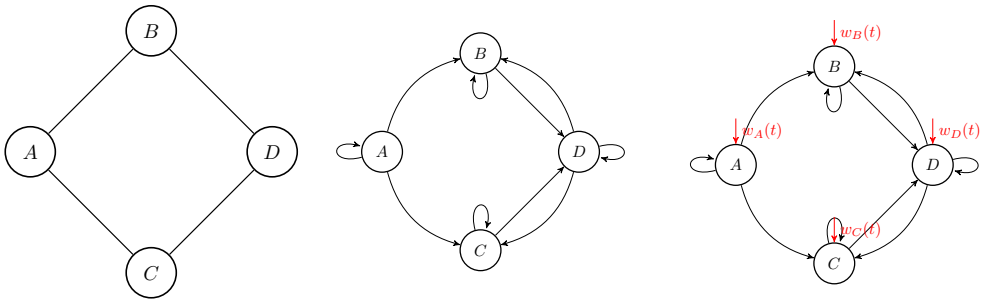

Figure 1: (left)The undirected graph for the static problem; (mid) the Markov processes for the dynamic problem;(right) the AR(1) processes. The red inputs are the exogenous Gaussian noise $\boldsymbol{w} \sim \mathcal{N}(0, \Sigma_w)$ in the purpose of generating the realization $\{x_A(t), x_B(t), x_C(t), x_D(t)\}_{t=1}^N$.

First we refer to the conclusion about the relationship between the Markov process and the AR(1) process.

Theorem 1. *The Markov process is equivalent to the AR(1) process if the exogenesou noise is i.i.d.*

Then we consider the undirected graph and the Markov processes. The Markov processes can reduce to the undirected model by projecting data along the temporal axis. In details, we consider the realization of the Markov process is $\{x_A(t), x_B(t), x_C(t), x_D(t)\}_{t=1}^N$, which can also be regarded as the realization of undirected model. Here we call this undirected graph is the projected undirected graph from the Markov process.

Assume the Gaussian distribution for all models. For the undirected graphs, we have the covariance matrix, Σ. Equivalently, the covariance matrix is the inverse of the precision $\Lambda := \Sigma^{-1}$. For the Markov process, we use the transition matrix P. Then we ask the following two questions.

1. Can the zeros in P imply the zeros in Λ ?
2. Can the zeros in Λ imply the zeros in P?

We find only the second statement is true. The first one is conditionally true only if both one entry and its symmetric one are zeros, e.g., $p_{AB} = 0$ and $p_{BA} = 0$. Thus, we can conclude

Assumption 2. *Without loss of generality, we assume all nodes in the Markov process is self autoregressive, i.e., $p_{ii} \neq 0$. If not, these nodes can be deleted and its impact is equivalent to the exogenous noise $\boldsymbol{w} \sim \mathcal{N}(0, \Sigma_w)$.*

Theorem 3. *With Assumption 2 and $\boldsymbol{w}$ is i.i.d., the void edges in the Markov process are equivalent to the void edges in the undirected model if the undirected graph is the projected graph from the Markov process.*

Proof. Assume p_{AB} is non-zero, and p_{AA} is non-zero by Assumption 1.

$$x_B(t+1) = p_{AB}x_A(t), \quad x_A(t+1) = p_{AA}x_A(t), \forall t, \tag{1}$$

such that $x_B(t+1)$ and $x_A(t+1)$ are directly correlated for all k. Then x_A and x_B are correlated in the undirected graph. □

Corollary 4. *If Σ_w is not restricted to be i.i.d., the undirected graph would be influenced by both the Markov process and Σ_w. Then the void edges in the Markov process is only the necessary condition for the void edges in its projected undirected graphs.*

This is a very interesting conclusion, which tells if we observe the topology of projected undirected graphs, we can conclude some information of the Markov process. This would be very helpful for the learning of the Markov process because most time the spatial topology is easily accessible for spatial-temporal problems.

However, the Markov process in Fig.1 is a too simple model for the spatial-temporal problems because it assumes time-invariant dynamics, i.e., the transition matrix P is time-invariant. As we mention, it is just AR(1) processes, which means the current state only depends on the last one state. For the complex system, it is proper to extend it to AR(n) processes. In this paper, we would discuss Theorem.3 also works, or how to modify Theorem.3, for AR(n) processes.

To sum up, in this paper we explore the question that if the spatial topology is beneficial for learning the spatial-temporal models, especially for the information about the sparsity.

4 Problem formulation

We build our question on the very comprehensive problem, the sequence to sequence learning. The learning problem is the typical problem, learning the structure and parameters from the observation data, which cam regarded as the realization of models, e.g., $\{x_A(t), x_B(t), x_C(t), x_D(t)\}_{t=1}^N$. For learning dynamic problems, we also call it the identification problem. The motivation of identification is the prediction in the time domain.

The original problem is formulated as following. For convenience, we would take Fig. 1 as an example. First, we aims to the sequence to sequence prediction. Based on the training data $\{x_A(t), x_B(t), x_C(t), x_D(t)\}_{t=1}^N$, we require the one-step ahead prediction $x_A(t), x_B(t), x_C(t), x_D(t)$ with respect to $\{x_B(t-k), x_C(t-k), x_D(t-k)\}_{k=1}^\infty$, where $x_A(t)$ is the special node without the history data. Here we only consider to predict $x_A(t)$ by linear modes, i.e., the finite impulse response model (FIR)

$$\hat{x}_A(t) = \sum_{k=1}^{\infty} h_{AB}(k)x_B(t-k) + \sum_{k=1}^{\infty} h_{AC}(k)x_C(t-k) + \sum_{k=1}^{\infty} h_{AD}(k)x_D(t-k), \tag{2}$$

where $\{h_{AB}(k)\}_{k=1}^\infty, \{h_{AC}(k)\}_{k=1}^\infty, \{h_{AD}(k)\}_{k=1}^\infty$ are the impulse response coefficient to be estimated from the data.

In the control community, such multiple sequences to the single sequence prediction in Eqn.(2) is also named as multiple-input single-output (MISO) identification. The MISO identification is actually the sub-problem of the multiple-input multiple-output (MIMO) identification. Stack all the nodes and denote $x(t) := [x_A(t), x_B(t), x_C(t), x_D(t)]^T \in \mathbb{R}^4$. The arbitrary MISO prediction with form in Eqn.(2) would be solved if the totally dynamics is solved,

$$x(t) = f(x(t-1), x(t-2), \cdots, x(t-\infty)), \tag{3}$$

where $f(\cdot)$ is the prescribed model and it is stated as AR model in this paper. Thus, the AR identification is actually the more general problem of MISO identification.

To solve the sequences to sequence prediction problem (or MISO identification), we would first discuss the AR identification in the following sections, and second, embed the prior knowledge of sparsity to obtain a further accurate and robust model.

5 Autoregressive processes identification

In this section, we review the method of AR processes identification. Denote p the order the AR processes. Then AR(p) processes $\{x(t)\}$ satisfies

$$x(t) = -\sum_{k=1}^{p} A_k x(t-k) + w(t), \quad w(t) \sim \mathcal{N}(0, \Sigma), \tag{4}$$

where $x(t) \in \mathbb{R}^m$ is a m-dimensional vector, $A_k \in M^{m\times m}, k = 1, \cdots, p$ are all $m\times$ m matrices, $w(t)$ is the zero-mean Gaussian noise with variance Σ, independent with $x(t-k), k = 1, 2, \cdots, \infty$.

Another representation of AR(p) processes is multiplying the square root of noise variance on both sides of Eqn.(4). Denote $B_0 = \Sigma^{-1/2}, B_k = \Sigma^{-1/2} A_k, k = 1, \cdots, p$, and $B = [B_0\ B_1 \cdots B_p]$. Then we have

$$B_0 x(t) = -\sum_{k=1}^{p} B_k x(t-k) + v(t), \quad v(t) \sim \mathcal{N}(0, I). \tag{5}$$

Eqn.(4) and Eqn.(5) are equivalent representations for AR processes

5.1 Estimates for AR processes

The output error $e(t)$ is used to indicate the one-step ahead error between the real output and estimate one, i.e.,

$$e(t) = x(t) - \hat{x}(t) = x(t) - (-\sum_{k=1}^{p} \hat{A}_k x(t-k)), \tag{6}$$

where $\hat{\boldsymbol{A}} = [I\ \hat{A}_1 \cdots \hat{A}_p]$ represents the AR coefficient estimated from the training data.

According to different assumptions and criteria, or objective functions, There are different estimates of the AR model via the observation, e.g., the least square (LS) estimate, the maximum likelihood (ML) estimate, and the maximum entropy (ME) estimate. Interestingly, these three estimates have consistent results for AR processes once the data length is large enough.

5.1.1 the LS estimate for AR processes

The LS estimate is derived by minimizing the output error $e(t)$ in a sense of square norm, we obtain

$$\hat{\boldsymbol{A}} = \underset{A_1,\cdots,A_p}{argmin} \boldsymbol{E} \parallel e(t) \parallel_2^2 = \underset{A_1,\cdots,A_p}{argmin}\ Tr(AT(R)A^T), \tag{7}$$

where

$$A = [I\ A_1 \cdots A_p], \quad R = [R_0\ R_1 \cdots R_p], \quad R_k = \boldsymbol{E} x(t+k) x(t)^T, \tag{8}$$

and $T(R)$ is the block-Toeplitz matrix constructed by mapping $T:\ M^{n,p} \to S^{n(p+1)}$

$$T(R) = \begin{bmatrix} X_0 & X_1 & \cdots & X_p \\ X_1^T & X_0 & \cdots & X_{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ X_p^T & X_{p-1}^T & \cdots & X_0 \end{bmatrix}. \tag{9}$$

We estimate variance $\hat{R}_k$ from observation, so we use the sample covariance matrix C to replace T(R), i.e., $C := T(\hat{R})$. AR processes identification has the least square (LS) estimate, which is derived from the constrained quadratic optimization

$$\underset{A}{\text{minimize}} \quad tr(ACA^T). \tag{10}$$

5.1.2 the ML estimate for AR processes

The LS estimate is also the maximum likelihood estimate because we embed the Gaussian distribution to the noise $w(t)$, i.e., $w(t) \sim \mathcal{N}(0, \Sigma)$. By Gaussian processes, we conclude AR processes $\{x(t)\}$ is the joint Gaussian distribution, whose maximum likelihood is

$$\underset{A,\Sigma}{\text{minimize}} \quad \frac{1}{(2\pi)^n det(\Sigma)^{(N-p)/2}} exp(-\frac{1}{2}\sum_{t=p+1}^{N} x(t)^T A^T \Sigma^{-1} A x(t)). \tag{11}$$

Let $B = [B_0\ B_1 \cdots B_p]$, $B_0 = \Sigma^{-1/2}$, $B_k = \Sigma^{-1/2} A_k, k = 1, \cdots, p$. The ML estimate in Eqn.11 becomes

$$\underset{B}{\text{minimize}} \quad -2\log det(B_0) + tr(CB^T B). \tag{12}$$

5.1.3 the ME estimate for AR processes

The ME estimate is introduced by [2],

$$\underset{S(w)}{\text{maximize}} \quad \frac{1}{2\pi}\int_{-\pi}^{\pi} \log\det S(\omega) d\omega, \tag{13}$$

$$\text{subject to} \quad \frac{1}{2\pi}\int_{-\pi}^{\pi} S(\omega) e^{jk\omega} d\omega = \hat{R}_k, \quad 0 \le k \le p, \tag{14}$$

where $S(w)$ is the power spectrum density related to R_k with the following Fourier transform.

$$S(w) = \sum_{k=-\infty}^{\infty} R_k e^{-jk\omega}, \quad R_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} S(w) e^{jk\omega} dw. \tag{15}$$

The ME estimate is also the covariance matrix completion problem. The problem aims to complete $S(\omega)$ by the limited number of $\hat{R}_k$.

The following paragraphs would show the dual problem ME estimate in Eqn.(13) has the equivalent form of the ML estimate. Denote Lagrange multiplier $Y_0 \in S^n$ and $Y_k \in R^{n\times n}, k = 1, \cdots, p$. The Lagrange becomes

$$\mathcal{L}(S(\omega), Y_0, Y_k) = -\frac{1}{2\pi}\int_{-\pi}^{\pi} \log\det S(\omega) d\omega + tr(Y_0^T(R_0 - \hat{R}_0)) + 2\sum_{k=1}^{p} tr(Y_k^T(R_k - \hat{R}_k)). \tag{16}$$

Then we differentiate Lagrange with R_k, which are the components of $S(w)$ in Eqn.15. We obtain the equality that

$$Y_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} S^{-1}(\omega) d\omega, 0 \le k \le p, \quad S^{-1}(\omega) = \sum_{k=-\infty}^{\infty} Y_k e^{-jk\omega}. \tag{17}$$

Then the dual form of Eqn.(13) is

$$\underset{Y}{\text{minimize}} \quad -\frac{1}{2\pi}\int_{-\pi}^{\pi} \log\det S^{-1}(\omega) d\omega + tr(Y_0^T \hat{R}_0) + 2\sum_{k=1}^{p} tr(Y_k^T \hat{R}_k) - n. \tag{18}$$

Refer to Kolmogorov's formula in [7] and the substitution $Y_k = \sum_{i=0}^{p-k} B_i^T B_{i+k}$ for $0 \le k \le p$, we have

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} \log\det S^{-1}(\omega) d\omega = \log\det(B_0^T B_0) \tag{19}$$

$$tr(Y_0^T \hat{R}_0) + 2\sum_{k=1}^{p} tr(Y_k^T \hat{R}_k) = tr(T(\hat{R}) B^T B), \tag{20}$$

such that the dual form of the ME estimate, Eqn.(18), becomes the ML estimate, Eqn.(13).

6 The modified estimates for sparse networks

In this section, we assume the graphical model in space is given as the prior knowledge for the learning problem. Particularly, the number of edges is limited such that the graphical model is sparse. Note the edges in the graphical model represent the conditional independence of nodes.

6.1 Maximum entropy estimate for the known sparse graph

If we know there are lots of conditional independence in space, e.g., $x_1 \perp x_2|x_3 \cdots x_m$, we can impose the sparsity in the precision matrix. Thus, this is an essential relationship between the graphical model and the Gaussian model. To illustrate that, we present an example in Fig.2.

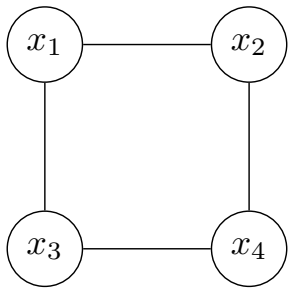

Figure 2: The graphical model indicates the conditional independence $x_1 \perp x_4|x_2, x_3$, and $x_2 \perp x_3|x_1, x_4$, leading to zeros in precision matrix, i.e., $\Lambda_{14} = \Lambda_{41} = \Lambda_{23} = \Lambda_{32} = 0$.

Fig.2 only shows the graphs with the small number of nodes. If the nodes' number rises up but the edge is sparse, i.e., the graph is not fully connected, we could obtain the very sparse precision matrix Λ with numerous zeros. This is what we learn from the static model and the result is applied in the famous algorithm, graphical lasso [5]. For dynamics models like AR processes, however, we have many covariance matrices $R_k := E[x(t+k)x(t)^T]$ with different lags. We must find another precision matrix to impose the sparsity constraints.

Let $X_{-i,-j}$ be the set incorporating all x except x_i and x_j. Given other nodes, the conditional independence in space, $x_i \perp x_j|X_{-i,-j}$ cannot imply $(R_k)^{-1}_{ij} = 0, k = -\infty, \infty$. Instead, $x_i \perp x_j|X_{-i,-j}$ is related to the power spectrum $S(w)^{-1}_{ij}$, see [2].

$$x_i \perp x_j|x_{-i,-j} \quad \Longleftrightarrow \quad (S(w)^{-1})_{ij} = 0, \forall\omega. \tag{21}$$

Take the graphical model in Fig.2 as an example. If it is the spatial relation of AR processes, we can conclude that, $\forall\omega$, $S(\omega)^{-1}_{14} = S(\omega)^{-1}_{41} = S(\omega)^{-1}_{23} = S(\omega)^{-1}_{32} = 0$. This is the essential statement for sparse AR processes. Note here the sparsity is in the sense of spatial domain rather than the temporal domain.

Combining Eqn.(17) and Eqn.(21), we have

$$0 = (S^{-1}(w))_{ij} = (\sum_{k=-\infty}^{\infty} Y_k e^{-jk\omega})_{ij}, \forall\omega, \Longleftrightarrow (Y_k)_{ij} = 0, \forall k. \tag{22}$$

Note we use the substitution $Y_k = \sum_{i=0}^{p-k} B_i^T B_{i+k}$ for $0 \le k \le p$ to bridge the dual of the ME estimate and ML estimate. We further conclude

$$x_i \perp x_j|x_{-i,-j} \Longleftrightarrow (S^{-1}(w))_{ij} = 0 \Longleftrightarrow (Y_k)_{ij} = 0, \forall k \Longleftrightarrow (\sum_{n=0}^{p-k} B_n^T B_{n+k})_{ij} = 0, \forall k, \tag{23}$$

In conclusion, we can modify the ME estimate by adding our prior knowledge got from the graphical model. Denote $\mathcal{V}$ as the set of pairs that has no edge in the given graphical model. Then the ME

estimate is modified with the additional constraints.

$$\underset{B}{\text{minimize}} \quad -\log det(B_0^T B_0) + tr(CB^T B) \tag{24}$$

$$\text{subject to} \quad (\sum_{n=0}^{p-k} B_n^T B_{n+k})_{ij} = 0, \forall k, \text{ for } (i,j) \in \mathcal{V}. \tag{25}$$

Due to the constraint, it is not the ML estimate anymore, but it is still the ME estimate because the prior knowledge from the graphical model outweighs the empirical estimate of the covariance, $\hat{R}_k$.

6.2 Lasso regularization for the unknown sparse graph

In this section, we are only told the graphical model is sparse, but the details of graphical model is not clear any more, i.e., the edge's location set $\mathcal{V}$ is unknown.

This is not a new type of problem for the regression problem. The straightforward method is remove the constraint of the optimization to the objective function as the penalty. Then the constrained regression would become a regularized regression. Since our constraint is enforcing the entries to zero, we transform it into the ℓ_1-norm penalty onto the objective function. This is exactly what the graphical lasso did for the static problem [5].

$$\underset{B}{\text{minimize}} \quad -\log det(B_0^T B_0) + tr(CB^T B) + \gamma h_\infty(B^T B) \tag{26}$$

$$h_\infty(B^T B) = \sum_{i<j} \max\{\max_{k=1,\cdots,p} |(\sum_{n=0}^{p-k} B_n^T B_{n+k})_{ij}|, \max_{k=1,\cdots,p} |(\sum_{n=0}^{p-k} B_n^T B_{n+k})_{ji}|, |(\sum_{n=0}^{p} B_n^T B_n)_{ij}|\},$$

where γ is a hyperparameter that tunes the penalty. This is the $\ell - 1$-norm regularized ML estimate, whose constraint is also convex. The solution can be solved by CVX package in MATLAB.

7 Simulation

In the simulation, we would test the performance of penalized estimates in Eqn.(26). As the baseline, we also add the basic LS estimate for the AR model, which does not have any penalty. Moreover, we introduce another estimate penalized by the kernel-based ridge regularization, see details in Appendix. A. These three estimates are listed in the table below.

Table 1: Estimates used for simulations.

Estimate	notation	penalty	implementation
Maximum likelihood estimate	ML	no	
Spatial lasso estimate	MEs	ℓ_1 norm	CVX[6]
Temporal ridge estimate	MEt	kernel-based ℓ_2 norm	KRM[3]

The final estimate is used for the MISO prediction. After the transformation, all estiamtes above leads to the same form as

$$\hat{x}_i(t) = \sum_{i \neq j} \left(\sum_{k=1}^{m} h_j(k) x_j(t-k) \right), \tag{27}$$

where m is the length of the model, which should be large enough for accuracy. Each estimate is evaluate by the mean square error (MSE) in the test data,

$$\text{MSE} := \frac{1}{N} \sum_{t=1}^{N} \parallel x_i(t) - \hat{x}_i(t)) \parallel^2 . \tag{28}$$

Here is the trick that $t = 1$ is not the beginning of the prediction because we got rid of the warm-up period.

Then we implement two simulations. The first simulation is for the synthetic data, which is generated by the consistent AR processes. The second simulation is for the real data, the international stock market, whose real model is complex and inaccessible.

7.1 Synthetic data simulation

Let $x(t) := [x_1(t), \cdots, x_n(t)]^T \in \mathbb{R}^n$. The data $\{x(t)\}_{t=1}^N$ is generated by the exact AR model.

$$x(t) = -\sum_{k=1}^{p} A_k^0 x(t-k) + w(t), \quad w(t) \sim \mathcal{N}(0, \Sigma^0), \tag{29}$$

where A_k^0 and Σ^0 are the true ground of the parameters. The value of the parameters is set as.

$$p = 1, A_1^0 = toeplitz([0.5\ 0.2, 0, \cdots]) + \begin{bmatrix} 0.1 & 0 & 0.4 & 0.5 & \cdots \\ 0 & 0.2 & 0.3 & 0 & \\ 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0.2 & \\ \vdots & & & & 0 \end{bmatrix}, \Sigma = I_n. \tag{30}$$

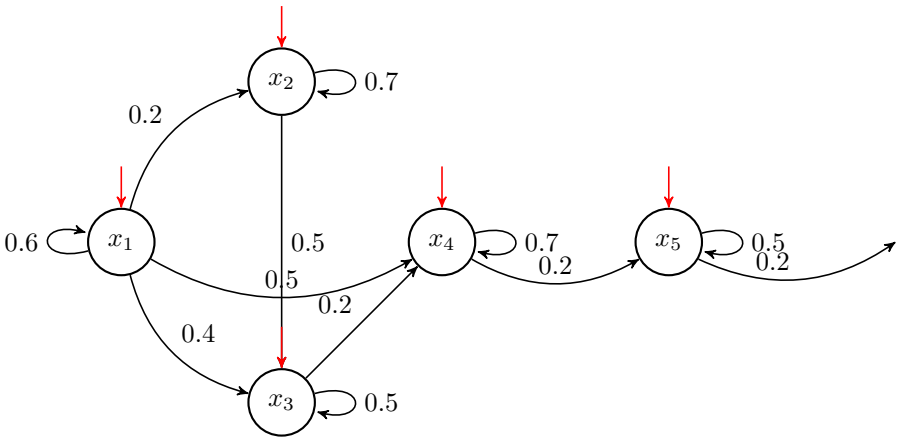

Figure 3: The topology of the true AR(1) process to generate $\{x(t)\}$. The red lines are exogenous zero-mean Gaussian noise with variance I_n. The value on the edge do not represent the transition matrix of the Markov processes because they are not summed to 1.

After setting up the environment, we make the data length N=100, and the model length m=10. Moreover, the spatial dimension is n=5 or n=15. Here n=5 is the dense-nodes condition whose undirected graphical model. According to the Fig.3, we observe that the topology is more sparse if we increase n. The n=15, from n_4 to n_{15} is a long linear chain without any connection to x_1, x_2 and x_3.

The results of estimates show in Fig.4, and we attach the MSE onto the title. Note we must adjust the hyperparameter γ for MEs, so we plot multiple predictors of MEs with respect to different γ. The result is consistent with our expectation: the lasso penalty is significantly helpful for a large number of sparse nodes.

Figure 4: The predicted $\hat{x}_1(t)$ by different estimates and the true ground (dash line) for the testing data. For various choices of hyperparameter γ, we obtain multiple curves of MEs estimates

7.2 International stock market

In this simulation, we use the real-world data. There are 7 daily stock indexes obtained from Yahoo, All Ords (Australia), Hang Seng Index(Hong Kong), Nikkei 225(Japan), DAX (Germany), NYSE Composite (US), Dow Jones Industrial Average(US), and S&P 500 (US). The daily record starts from 2012-07. For this problem we set $n = 7, p = 5, m = 10, N = 500$ for the training. All the stock index is normalized respectively. After the training, we test the predictor on another N=500 testing data. The result of the predictor shows in Fig.5.

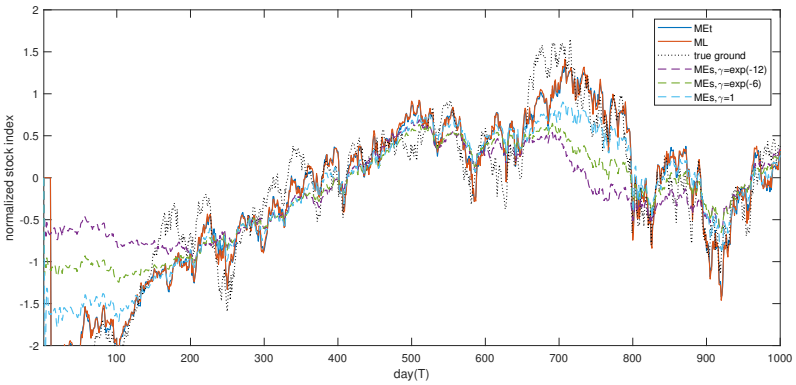

Figure 5: The result of prediction of All Ords by other six stock index. The training data is N=1:500, while the testing data is N=501:100. The terrible prediction at the beginning ($N \leq 200$) is due to the cold start.

For the predictor in Fig. 5, we has the MSE as follows. To get rid of the initial boundary condition, we calculate the training MSE by N=201 rather than N=1.

Table 2: Mean square error of the stock predictor.

MSE table	ML	MEt	MEs γ=exp(-12)	MEs γ =exp(-6)	MEs γ =1
traning(from N=201 to 500)	0.0627	0.0642	0.0971	0.0970	0.0891
testing(from N=501 to 1000)	0.0867	0.0902	0.2944	0.2039	0.1290

According to the Table.2 and Fig.5, we find MEs estimate also competitive with other estimates. Note there are only seven stock indexes and the model is not so sparse. Comparing the training and testing of MEs, the highly penalized $MEs(\gamma = 1)$ is more robust than other choices of γ.

Another advantage of MEs is that ℓ_1 norm would distinguish zero and other, which tell edges of the graph. For this data, we find $\hat{A}_i, i = 2, \cdots, 7$ is negligible to $\hat{A}_1$, which means it is almost AR(1) processes. So we only present $\hat{A}_1$ and its first row $\hat{A}_1(1,:)$, which is used to predict $x_1(t)$.

$$\hat{A}_1(1,:) = \underset{Au}{[0.810} \quad \underset{Jp}{0.091} \quad \underset{HK}{0.043} \quad \underset{Ge}{0.125} \quad \underset{US}{0.151} \quad \underset{US}{0.111} \quad \underset{US}{0.139]}, \tag{31}$$

where the first element represents the AR feature, and other elements represent the weight of edge, i.e., the correlation between Australia stock index and other stock indexes. Interestingly, we find the Australia stock is more related to Germany and the US rather than Japan and Hong Kong. This result is very well explained by the time zone:

> The U.S. is from UTC-6 to UTC-10, and Germany is UTC+1, whose delay for predicting Australia UTC+8 is less than 12 hours. For Hong Kong UTC+8 and Japan UTC+9, it is too close to Australia UTC+8. However, we consider yesterday stock for one-step-ahead prediction such that their time delay is nearly 24 hours. Thus, the Austria stock is conditional independent with Hong Kong and Japan if the U.S. stock is given.

7.3 Conclusion and future work

After all , we can conclude the following statements based on two simulations.

- When the sample length/model length is limited($N/(n * p)$ is small),over-fitting makes ML estimate terrible. This is the motivation why we add the regularization, either ridge or lasso. Particularly, lasso would have superior performance for the large sparse node case.
- The largest advantages of MEs estimate is displayed in the sparse spatial graph (large n). It enforces sparsity so as to amplify the neighbouring nodes. For other estimates, however, they still overfit all nodes in space.
- For MEs, we need tuning hyperparameter γ, e.g. by cross-validation. For MEt, the hyperparameter γ can be tuned automatically. Both MEt and MEs add the penalty to the regression. The difference is because the ridge regression is also maximum a posterior (MAP) estimate and we can use the empirical Bayes (EB) method.
- The research gap still exists,i.e., combining MEt and MEs. When we enforce spatial sparsity, we fixed the time model as the hard constraint. When we enforce the temporal sparsity, we assume the space is full. How to embed both prior knowledge into the regularization is still an open question.

References

[1] Ricardo J Bessa, Artur Trindade, and Vladimiro Miranda. Spatial-temporal solar power forecasting for smart grids. *IEEE Transactions on Industrial Informatics*, 11(1):232–241, 2014.

[2] John Parker Burg. *Maximum entropy spectral analysis.* Stanford University, 1975.

[3] Tianshi Chen, Henrik Ohlsson, and Lennart Ljung. On the estimation of transfer functions, regularizations and gaussian processes—revisited. *Automatica*, 48(8):1525–1535, 2012.

[4] David Roxbee Cox and Valerie Isham. A simple spatial-temporal model of rainfall. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 415(1849):317–328, 1988.

[5] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[6] Michael Grant and Stephen Boyd. Cvx: Matlab software for disciplined convex programming, version 2.1, 2014.

[7] Edward James Hannan. *Multiple time series*, volume 38. John Wiley & Sons, 2009.

[8] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[9] Philippe Lemey, Marc Suchard, and Andrew Rambaut. Reconstructing the initial global spread of a human influenza pandemic: a bayesian spatial-temporal model for the global spread of h1n1pdm. *PLoS currents*, 1, 2009.

[10] R Kelley Pace, Ronald Barry, Otis W Gilley, and CF Sirmans. A method for spatial–temporal forecasting with an application to real estate prices. *International Journal of Forecasting*, 16(2):229–246, 2000.

[11] Jitkomut Songsiri, Joachim Dahl, and Lieven Vandenberghe. Graphical models of autoregressive processes., 2010.

[12] Jitkomut Songsiri and Lieven Vandenberghe. Topology selection in graphical models of autoregressive processes. 11:2671–2705, 2010.

[13] Shuangjie Xu, Yu Cheng, Kang Gu, Yang Yang, Shiyu Chang, and Pan Zhou. Jointly attentive spatial-temporal pooling networks for video-based person re-identification. In *Proceedings of the IEEE international conference on computer vision*, pages 4733–4742, 2017.

[14] He-Sheng Zhang, Yi Zhang, Zhi-Heng Li, and Dong-Cheng Hu. Spatial-temporal traffic data analysis based on global data management using mas. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):267–275, 2004.

[15] Mattia Zorzi. Empirical bayesian learning in ar graphical models. *Automatica*, 109:108516, 2019.

[16] Mattia Zorzi. Autoregressive identification of kronecker graphical models. *Automatica*, 119:109053, 2020.

[17] Mattia Zorzi and Rodolphe Sepulchre. Ar identification of latent-variable graphical models. *IEEE Transactions on Automatic Control*, 61(9):2327–2340, 2015.

A Ridge regularization for MISO problems

Recall our task is the MISO identification. The state of the art methods is the kernel-based regularization method (KRM), which uses ℓ_2 norm, ridge penalty, in the regression. If we copy the original MISO model as

$$\hat{x}_A(t) = \sum_{k=1}^{\infty} h_{AB}(k)x_B(t-k) + \sum_{k=1}^{\infty} h_{AC}(k)x_C(t-k) + \sum_{k=1}^{\infty} h_{AD}(k)x_D(t-k), \tag{32}$$

where the summation to infinite is truncated to n in practice. Denote $h_{AB} = [h_{AB}(1), \cdots, h_{AB}(n)] \in \mathbb{R}^n$, so are h_{AC} and h_{AD}

$$\begin{aligned}\hat{h}_{AB}, \hat{h}_{AC}, \hat{h}_{AD} = \underset{h_{AB}, h_{AC}, h_{AD}}{\operatorname{argmin}} \sum_{t=1}^{N} \| x_A(t) - \hat{x}_A(t) \|^2 + \gamma_1 h_{AB}^T P_1(\alpha_a)^{-1} h_{AB} \\ + \gamma_2 h_{AB}^T P_2(\alpha_2)^{-1} h_{AC} + \gamma_3 h_{AD}^T P_3(\alpha_3)^{-1} h_{AD},\end{aligned} \tag{33}$$

where $P_i(\alpha_i)$ is kernel matrix, the weight matrix for the ridge penalty. Ridge regression is the maximum absolute posterior (MAP) estimator. If you have the priori of h_k, $P_k(\alpha_k)$ is induced by

Reproducing Kernel Hilbert Space(RKHS). Here we set all the kernel matrix $P_k(\alpha_k)$ to be truncated correlated (DC) kernel matrix,

$$P_k(\alpha_k)_{ij}^{-1} = \begin{bmatrix} \frac{1}{\lambda(1-\rho^2)} & -\frac{\rho}{\lambda^{3/2}(1-\rho^2)} & 0 & \cdots & 0 \\ -\frac{\rho}{\lambda^{3/2}(1-\rho^2)} & \frac{1-\rho^4}{\lambda^2(1-\rho^2)^2} & -\frac{\rho}{\lambda^{5/2}(1-\rho^2)} & \cdots & 0 \\ 0 & -\frac{\rho}{\lambda^{5/2}(1-\rho^2)} & \frac{1-\rho^4}{\lambda^3(1-\rho^2)^2} & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & \cdots & \cdots & -\frac{\rho}{\lambda^{(2n-1)/2}(1-\rho^2)} & \frac{1-\rho^4}{\lambda^n(1-\rho^2)^2} \end{bmatrix}. \quad (34)$$

This is also a designed structure (tridiagonal matrix) for ML estimate, which is also penalized form of the maximum entropy estimate, i.e., ME estimate has the hard constraints but here the regularization has the soft constraints.