# Thoughts about future work

MG

May 4, 2015

**Abstract**

This is to help me capture and structure possible ideas for future work that shall not be forgotten. The advantage of writing these down and developing the notation etc. is that they can be quickly picked up and worked out into something more serious. Each section has a topic possibly rather unrelated to the other sections. The details developed in the subsections may be somewhat messy at this stage. Several sections have very little text yet, these are place-holders for raised ideas not to be forgotten (though perhaps completely unrealistic or wrong).

## Contents

# 1 Granger causality in panel time-series (last updated 2/5/2015)

The idea here is to extend our paper on learning Granger-causality [1] into panel data. In this setting the multiple time series are observed over several *Note: independent?* cross-sections *Note: is this the right term?* . For example, in the Kaggle Wallmart competition (`https://www.kaggle.com/c/walmart-recruiting-sales-in-stormy-weather`), the time series are the sales of various products (bread, milk, umbrella), the panel dimension is added by having these over multiple shops. The assumption here would be that for every shop you can learn a VAR but the VARs across the sections shall be similar to one another in terms of their Granger-causality. Possibly, the individual VARs could also be constrained to be focalised (see [1]) but this is perhaps not necessary in the 1st version.

This ideas is developed in this paper submitted to ICML2015 workshop on demand forecasting on 1/5/2015.

## 1.1 Panel F-VAR, SF-VAR

This bit is not in the above workshop paper so is left here. In [1] we have developed methods to learn VAR models whose Granger-causality graphs are concentrated around a few focal series. We could extend this to the panel setting. Following the development and logic of [1] we rewrite the loss function in terms of the p-dimensional blocks in the input and parameter matrices $\mathbf{X}_z, \mathbf{W}_z$.

$$L(\mathbf{W}^{3d}) := \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{z=1}^{Z}(y_{t,k,z} - \sum_{b}^{K}\langle \tilde{\mathbf{w}}_{b,k,z}, \tilde{\mathbf{x}}_{t,b,z}\rangle)^2 \tag{1}$$

and further by decomposing $\tilde{\mathbf{w}}_{b,k,z} = \gamma_{b,k,z}\tilde{\mathbf{v}}_{b,k,z}$. The $K \times K \times Z$ 3d-tensor $\mathbf{\Gamma}^{3d}$ will be used to control the sparsity and similarity between the models and the $Kp \times K \times Z$ 3d-tensor $\mathbf{V}^{3d}$ allows for learning task-section specific parameters.

We decompose each of the $\mathbf{\Gamma}_z$ (cuts of $\mathbf{\Gamma}^{3d}$ through the $Z$ dimension) into the common $\mathbf{A}_z$ and task-specific $\mathbf{B}_z$ parts as in [1] so that $\mathbf{\Gamma}_z = \mathbf{A}_z - diag(\mathbf{A}_z) + \mathbf{B}_z$ where we set $\mathbf{B}_z = \mathbf{I}$. To enforce the similarity of the models across the sections, we further set $\mathbf{\Gamma}_z = \mathbf{\Gamma}$ (and $\mathbf{A}_z = \mathbf{A}$) so that we learn just a single matrix $\mathbf{\Gamma}$ common for all the cross-sections.

$$L(\mathbf{W}^{3d}) := \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{z=1}^{Z}(y_{t,k,z} - \sum_{b}^{K}\gamma_{b,k}\langle \tilde{\mathbf{v}}_{b,k,z}, \tilde{\mathbf{x}}_{t,b,z}\rangle)^2 \tag{2}$$

The constraints follow from [1] in analogy.

- for F-VAR: $||\mathbf{V}_z||_2^2 < \epsilon$; $\alpha_{.,k} = \overline{\alpha}$; $\sum_b \overline{\alpha}_b = 1$; $\overline{\alpha}_b > 0$
- for SF-VAR: $||\mathbf{V}_z||_2^2 < \epsilon$; $\sum_b \alpha_{b,k} = 1$; $\alpha_{b,k} > 0$; $rank(\mathbf{A}) < r$.

# 2 Convex relaxation of F-VAR and SF-VAR (last updated 29/4/2015)

In section 2.3 of [1] it is noted that the we can rewrite optimisation of F-VAR and SF-VAR as a weighted ridge regression problem where $R(\mathbf{W}) = ||\mathbf{V}||_2^2 = \sum_{b,k} 1/\gamma_{b,k}^2 ||\tilde{\mathbf{w}}_{b,k}||_2^2$ with further constraints on $\mathbf{\Gamma}$. Based on the note from Francesco [2] the optimisation problem for F-VAR can be simplified to be expressed only in terms of $\mathbf{W}$.

From [1] we have $\mathbf{\Gamma} = \mathbf{A} - diag(\mathbf{A}) + I$. So for the elements $\gamma_{b,k} = \alpha_{b,k} + \delta_{b,k}(1 - \alpha_{b,k})$, where $\delta_{b,k} = 1$ for $b = k$ and is zero otherwise.

For the regularizor we get

$$\begin{aligned}
R(\mathbf{W}) &= \sum_{b,k} 1/\gamma_{b,k}^2 ||\tilde{\mathbf{w}}_{b,k}||_2^2 \\
&= \sum_{b,k} ||\tilde{\mathbf{w}}_{b,k}||_2^2 \ / \ \big(\alpha_{b,k} + \delta_{b,k}(1 - \alpha_{b,k})\big)^2 \\
&= \sum_{b,k \neq b} ||\tilde{\mathbf{w}}_{b,k}||_2^2 \ / \ \alpha_{b,k}^2 + \sum_b ||\tilde{\mathbf{w}}_{b,b}||_2^2 \qquad (3)
\end{aligned}$$

The full optimisation problem (in the Lagrange form) can be written as

$$J(\mathbf{W}) = L(\mathbf{W}) + \lambda_1 ||\mathbf{V}||_2^2 + \sum_k \lambda_{k+1} \sum_b \alpha_{b,k}, \qquad (4)$$

where

$$L(\mathbf{W}) := \sum_{t=1}^{T} \sum_{k=1}^{K} (y_{t,k} - \langle \mathbf{w}_{.,k}, \mathbf{x}_{t,.} \rangle)^2 \qquad (5)$$

and with further positivity constraints $\alpha_{b,k} \geq 0$ and for SF-VAR $rank(\mathbf{A}) \leq r$.

## 2.1 F-VAR

Following the assumptions of F-VAR, the columns in the $\mathbf{A}$ matrix are identical so that $\alpha_{.,k} = \overline{\alpha}$. Using (3) we can rewrite (4) as

$$J(\mathbf{W}) = L(\mathbf{W}) + \lambda_1 \Big( \sum_{b,k \neq b} ||\tilde{\mathbf{w}}_{b,k}||_2^2 \ / \ \overline{\alpha}_b^2 + \sum_b ||\tilde{\mathbf{w}}_{b,b}||_2^2 \Big) + \lambda_2 \sum_b \overline{\alpha}_b \qquad (6)$$

Minimising with respect to $\overline{\alpha}$ by equating $\partial J(\mathbf{W}) / \partial \overline{\alpha}_b = 0$ we get

$$\begin{aligned}
-2 \ \lambda_1 \sum_{k \neq b} ||\tilde{\mathbf{w}}_{b,k}||_2^2 \ \overline{\alpha}_b^{-3} + \lambda_2 &= 0 \\
2 \ \lambda_1 \sum_{k \neq b} ||\tilde{\mathbf{w}}_{b,k}||_2^2 &= \lambda_2 \ \overline{\alpha}_b^3 \quad \text{note that } \overline{\alpha}_b \geq 0 \\
(\frac{2 \ \lambda_1}{\lambda_2})^{1/3} \ (\sum_{k \neq b} ||\tilde{\mathbf{w}}_{b,k}||_2^2)^{1/3} &= \overline{\alpha}_b \qquad (7)
\end{aligned}$$

Plugging this back to (6) we get

$$\begin{aligned}
J(\mathbf{W}) &= L(\mathbf{W}) + (\lambda_2/2)^{2/3} \lambda_1^{1/3} \sum_b \Big( \sum_{k \neq b} ||\tilde{\mathbf{w}}_{b,k}||_2^2 \Big)^{1/3} + \lambda_1 \sum_b ||\tilde{\mathbf{w}}_{b,b}||_2^2 \\
&= L(\mathbf{W}) + \kappa_1 \sum_b ||\tilde{\mathbf{W}}_{b,k \neq b}||_2^{2/3} + \kappa_2 \sum_b ||\tilde{\mathbf{w}}_{b,b}||_2^2, \qquad (8)
\end{aligned}$$

where $\tilde{\mathbf{W}}_{b,k \neq b}$ is the $p \times K - 1$ matrix constructed from $\mathbf{W}$ by taking the $b$-th block of rows and leaving out the $b$-th column.

Problem (8) is now formulated only in terms of the elements of $\mathbf{W}$ but the middle term is non-convex (and non-differentiable) $\ell_{p,q}$ operator with $q = 2$ and $p = 2/3$.

### 2.1.1 Convex relaxation for F-VAR

We could change the original simplex constraint on $\alpha_{.,k}$ into a $\ell_2$ ball one so that $\sum_b \alpha_{b,k}^2 = 1$. Problem (6) in the Lagrangian form then is

$$J(\mathbf{W}) \quad = \quad L(\mathbf{W}) + \lambda_1 \Big( \sum_{b,k \neq b} ||\tilde{\mathbf{w}}_{b,k}||_2^2 \ / \ \overline{\alpha}_b^2 + \sum_b ||\tilde{\mathbf{w}}_{b,b}||_2^2 \Big) + \lambda_2 \sum_b \overline{\alpha}_b^2 \tag{9}$$

Minimising solution for $\overline{\alpha}_b$

$$\begin{aligned} -2 \ \lambda_1 \sum_{k \neq b} ||\tilde{\mathbf{w}}_{b,k}||_2^2 \ \overline{\alpha}_b^{-3} + \lambda_2 \ \overline{\alpha}_b &= 0 \\ 2 \ \lambda_1 \sum_{k \neq b} ||\tilde{\mathbf{w}}_{b,k}||_2^2 &= \lambda_2 \ \overline{\alpha}_b^4 \quad \text{note that } \overline{\alpha}_b \geq 0 \\ (\frac{2 \ \lambda_1}{\lambda_2})^{1/2} \ (\sum_{k \neq b} ||\tilde{\mathbf{w}}_{b,k}||_2^2)^{1/2} &= \overline{\alpha}_b^2 \qquad (10) \end{aligned}$$

and after plugging this back

$$\begin{aligned} J(\mathbf{W}) &= L(\mathbf{W}) + (\lambda_2/2)^{1/2} \lambda_1^{1/2} \sum_b \Big( \sum_{k \neq b} ||\tilde{\mathbf{w}}_{b,k}||_2^2 \Big)^{1/2} + \lambda_1 \sum_b ||\tilde{\mathbf{w}}_{b,b}||_2^2 \\ &= L(\mathbf{W}) + \kappa_1 \sum_b ||\tilde{\mathbf{W}}_{b,k \neq b}||_2 + \kappa_2 \sum_b ||\tilde{\mathbf{w}}_{b,b}||_2^2, \qquad (11) \end{aligned}$$

The middle term in (11) now reduces to the $\ell_{1,2}$ operator and hence we recover the convex group-lasso formulation across the $\tilde{\mathbf{W}}_{b,k \neq b}$ groups.

Note that in the experimental part of [1] one of the baseline models indicated as GroupVAR had similar formulation to (11). However, it used only a single tuning parameter $\kappa_1 = \kappa_2$ and (probably more importantly) the last term was $\sum_b ||\tilde{\mathbf{w}}_{b,b}||_2$ - an $\ell_{1,2}$ operator across the $\tilde{\mathbf{w}}_{b,b}$ instead of the simple $\ell_2$ norm on the concatenation of $\tilde{\mathbf{w}}_{b,b}$.

## 2.2 SF-VAR

In SF-VAR, in addition to the non-negativity $\alpha_{b,k} \geq 0$ we also have the low-rank constraint on $\mathbf{A}$ which in [1] is achieved by matrix factorization $\mathbf{A} = \mathbf{UL}$ so that $\alpha_{b,k} = \sum_j^r u_{b,j} \ l_{j,k}$, where $r$ is the rank of $\mathbf{A}$. This has a nice interpretation as soft-clustering of the models. We can express the regularized loss (4) using the non-negative $\mathbf{U}$ and $\mathbf{L}$

$$J(\mathbf{W}) \quad = \quad L(\mathbf{W}) + \lambda_1 \Big( \sum_{b,k \neq b} \frac{||\tilde{\mathbf{w}}_{b,k}||_2^2}{\big(\sum_j^r u_{b,j} \ l_{j,k}\big)^2} + \sum_b ||\tilde{\mathbf{w}}_{b,b}||_2^2 \Big) + \lambda_2 \sum_{b,j} u_{b,j} + \lambda_3 \sum_{j,k} l_{j,k} \tag{12}$$

where we use the same $\lambda_2$ and $\lambda_3$ for all the simplex constraints on columns of $\mathbf{U}$ and $\mathbf{L}$.

I try following the same trick as in section 2.1 to minimise first for the elements of $\mathbf{U}$ and $\mathbf{L}$.

$$\frac{\partial J(\mathbf{W})}{\partial u_{b,j}} \quad = \quad -2\lambda_1 \sum_{k \neq b} \frac{||\tilde{\mathbf{w}}_{b,k}||_2^2 \ l_{j,k}}{\big(\sum_j^r u_{b,j} \ l_{j,k}\big)^3} + \lambda_2 \tag{13}$$

$$\tag{14}$$

$$\frac{\partial J(\mathbf{W})}{\partial l_{j,k}} \quad = \quad -2\lambda_1 \sum_{b \neq k} \frac{||\tilde{\mathbf{w}}_{b,k}||_2^2 \ u_{b,j}}{\big(\sum_j^r u_{b,j} \ l_{j,k}\big)^3} + \lambda_3 \tag{15}$$

I don't think this can really help anything or at least I don't see it. Not sure if using nuclear norm instead of the **UL** decomposition would help. Perhaps yes ... *Todo: Look at nuclear norm minimisation*

# 3 Restrict directly (partial-)covariance instead of W (last updated 2/5/2015)

Yule-Walker equations? Does this even make sense? What does it really mean "constraining the covariances/partial covariances"?

The original definition of Granger-causality was for 2 variables (or 2 n-dimensional processes) only. That is can I improve prediction of $z_t$ given the past of $x_t$? I'm playing with many variables - the problem here is that some may improve prediction of $z_t$ by passing through another variable. Eg $y_t$ may influence $x_t$ and this in turn $z_t$. How is this treated in the model, the Granger graphs and what would I like to see?

## 3.1 GPs again

Aha?! So .. what if I formulate the problem as Yule-Walker but instead of putting constraints on norm in **W** I directly put some constraints on the covariance estimates?

Perhaps could be more obvious through Bayesian priors? Prior for $\gamma_{ij}(h) \sim \mathcal{N}(0, \sigma)$. Is this in fact a hyperprior on the covariance matrix of the original multi-variate Gaussian process (Gaussian random field)?

In fact, given that the covariance in GP and kernel in the kernel learning theory coincide, imposing sparsity constraints on the covariance somehow relates to Francesco's problem of learning sparse output kernel.

I can look at the vector prediction problem in VAR as at a scalar prediction problem where the specification of the task $l$ is an input for the prediction function.

**The following is mainly based on [4] and [5]** In the multiple time series prediction problem we have got a data sample $\{y_{i=t \times l} : t \in \mathbb{N}_T, l \in \mathbb{N}_m, i \in \mathbb{N}_{Tm}\}$ which we consider to be a realization of a real Gaussian process $f(.) \sim GP(\mu(.), k(.,.))$ with mean function $\mu(.) = E[f(.)]$ and covariance function $k(.,.) = E[(f(.) - \mu(.))'(f(.) - \mu(.))]$ taking as inputs the 2-dimensional vectors $[t,l]_i$ of all possible time and task combinations (GP is a distribution over function $f : \mathbb{N}_T \times \mathbb{N}_m \to \mathbb{R}$).

We will look at $f(.)$ as at an infinite dimensional vector (whose distribution is given by the GP) but in fact we're interested only in the $Tm$ long vector corresponding to our sample (or its parts) and its joint probability distribution which is a multivariate Gussian distribution (by the marginalisation property of Gaussians) $\mathbf{y} \sim N(\mu, \mathbf{K})$, $\mu_i = \mu([t,l]_i)$, $K(i,j) = k_\xi([t,l]_i, [t,l]_j), \forall i,j \in \mathbb{N}_i$. Here $\mu(.) : \mathbb{N}^2 \to \mathbb{R}$ is the mean function, $k_\xi(.,.) : \mathbb{N}^2 \times \mathbb{N}^2 \to \mathbb{R}$ is the covariance function (or kernel) with hyper-parameters $\xi$, and the inputs are the time/task indeces $\{[t,l]_i : i \in \mathbb{N}_{Tm}\}$.

Observe that this formulation of the covariance function $k$ is very similar *Note: identical?* to the $\mathcal{Y}$-valued kernel formulation $H(x,x')_{rc} = H((x,r)(x',c))$ where $x$'s correspond to the time indeces $t$ and $r, c$ to the task indeces $l$.

$$\begin{gathered} y_i = f([t,l]_i) + e_i, \quad \forall i \in \{1 \dots Tm\} \\ f \sim GP(\mu, k), \quad e_i \sim N(0, \sigma^2) \end{gathered} \tag{16}$$

[4] actually builds the GP theory and then goes onto Yule-Walker as well but only in the one-dimensional case, he does not go into any details for the multi-dimensional case.

### 3.2 Sparse covariance in GPs

Now that I've formulated the problem as a GP, the standard procedure would be to learn the kernel $k$. Here, I would like to learn a sparse kernel but not just somehow sparse, but rather specifically sparse to fit the time-series setting:

- for the time dimension the covariances (correlations) should diminish with increasing distance
- for the task dimension the ordering is to a great degree meaningless and so, in fact, is the distance (other than equal / not equal). The form of the kernel function should already reflect that. But preferably it should also be simply sparse (to begin with) or somehow group sparse (to do something like I'm doing in [1].
- perhaps a strange idea - we could somehow learn the task ordering/proximity (instead or learning sparse covariance) - or would thit be just simple reordering of the learned covariance matrix so that it diminishes in this direction
- moreover, while the process can be expected to be stationary in $t$ it is certainly not stationary in $l$. So is this whole idea of treating $l$ as simply another input a stupid idea?

### 3.3 Shape of kernels - NEW!

These are thoughts about the shape of the kernels (kernel functions) so that they work well for VARs and what I want to do with them.

- A stationary kernel for the time dimension shall be a toeplitz matrix with diminishing elements. In this way, I know the kernel value for a new observations since it is just an extension of the toeplitz matrix.
- There could be some non-stationarity brought into the time dimension by for example making the Toeplitz diagonals some smooth functions
- In vector-output kernel learning in e.g. [2] the output kernel $L$ (which corresponds in my case to the task dimension) is considered to be constant across all the inputs. Perhaps this is a too strong assumption - the relations between the outputs may also develop and change with time.
- In vector-output kernel learning the $H$ kernel is assumed to be separable into the product $H(.,.) = K(.,.)\,L$. I'm not convinced that this is the right assumption for my setting (even though obviously it makes things easy).

### 3.4 Ideas from group meeting 4/5/2015 - NEW!

- work with single time series but for multi-step ahead prediction
- similarity between input and output kernel via the fact that kernel = $XX'$ and covariance = $X'X$ are linked by the eigen-decomposition (share the same eigenvalues etc.)

## 4 Not yet developed

### 4.1 Online learning of sparse VARs

Normally, the VAR is learned in batch mode by sliding window without any control over the smoothness for the models across the windows. The smoothness should be implicit given the stationarity of the processes. But with sparsity norms such as $\ell_1$ which are known to be very unstable in the support selection this may not be so obvious, especially if the window shifts are bigger then single observation. Possibly even less in multi-task learning.

Moreover, the stationarity assumption is very unrealistic for any real-time series which is likely to have gone through some concepts changes. Well, I guess the non-smoothness would in fact be an indication of such concept drift appearing in the new window.

### 4.2 Instantenous covariance and Granger causality

Structural VARs A-model, B-model, AB-model, (see [3])

### 4.3 Granger causality for multi-step ahead prediction

Why, what would be the assumptions

### 4.4 Input-output kernel similarity in times series

Unsupervised learning, kernel PCA?

### 4.5 Matrix output for multi-step

An extension from vector output to matrix output. Fairly obvious since the theory is build over vector-spaces (matrix space is ok). Multi-linear models?

### 4.6 Projected gradient for optimising my problem

Solve least squares by projected gradient descent. Should be easy (I hope) but perhaps too slow.

## References

[1] Magda Gregorova, Alexandros Kalousis, Stephane Marchand-Maillet, Jun Wang: Learning vector autoregressive models with focalised Granger causality graphs, submitted for ICML2015

[2] Francesco Dinuzzo: Note to Magda about equivalent formulation, 17 April 2015

[3] Lütkepohl, Helmut: New Introduction to Multiple Time Series Analysis. Berlin: New York: Springer, 2005.

[4] Turner, Ryan Darby: Gaussian Processes for State Space Models and Change Point Detection, University of Cambridge, 2012.

[5] Rasmussen, Carl Edward, and Christopher K. I. Williams: Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press, 2006.