Towards the HPC-inference of causality networks from multiscale economical data

Illia Horenko[1], Patrick Gagliardini[1], William Sawyer[2], Lukáš Pospíšil[1]
[1] Università della Svizzera Italiana (USI Lugano), [2] Swiss National Supercomputing Centre (CSCS/ETH Zurich),
[illia.horenko@usi.ch, patrick.gagliardini@usi.ch, william.sawyer@cscs.ch, lukas.pospisil@usi.ch]

Focus of the project

Analysis of large amounts of economical data and data-driven inference of causality relations between different components of economical systems is one of the central problems in modern computational finance and economics. The task of proper mathematical description and adequate causality understanding for the economical data is hampered by the multiscale nature of the underlying processes, resulting from the presence of different temporal and spatial, i.e. regional, sectorial and global, scales.

Important challenges are:

(i) an investigation of the mutual causality influences of different economic observables and their spatial (e.g., regional) and temporal (e.g., associated with the business cycle) evolution,
(ii) identification of the most important exogenous impact factors that play a role in their dynamics,
(iii) proper mathematical and statistical description of the influences coming from the unresolved/latent scales and factors.

The solution of these problems can be enhanced by analysis of a causality network inferred from the data. This network is a directed weighted graph with edges representing the causality relations between the different economical variables, exogenous factors, etc. (situated at the vertices of this causality graph). Analysis of this graph would allow to understand the most important features of the underlying complex economical system.

Milestone questions about the targeted economical data:

1. Is there a causality relation between different sectors of the economy with respect to the credit risk migrations?'
2. What is the most effective implementation of the multiscale causality inference framework in the embarassingly-parallel case?
3. Are there any statistically-significant causality impacts from other sectors on the companies inside of the 'Banking and Finance' sector?
4. Among all of the considered alternatives and platforms, what is the most scalable implementation for multiscale causality inference?

Non-stationary modelling

We suppose that unknown parameters of the model θ are changing in time

$$\theta^*(t) = \arg\min_{\theta(t)} \int_0^T g(x(t),\theta(t))\ dt \quad \approx \quad \arg\min_{\theta(t)} \sum_{t=0}^{T} g(x_t,\theta(t)),$$

where $g(x(t),\theta)$ represents local model error (or equivalently negative value of maximum likelihood function), for example

$$g(x(t),\theta(t)) = \|x(t)-\theta(t)\|^2 \quad \text{(K-means model)}, \qquad g(x(t),\theta(t)) = \left\| x(t) - \left(\sum_{p=0}^{m} \alpha_p(t) t^p\right)\right\|^2 \quad \text{(polynomial regression)},$$

$$g(x(t),\theta(t)) = \left\| x(t) - \left(\mu(t) + \textstyle\sum_{i=0}^{p} A_i(t)x(t-i\tau) + \sum_{j=0}^{q} B_j(t) y^{t-j\tau}\right)\right\|^2 \quad \text{(autoregressive models)}.$$

Assumptions

1.) $\forall t$ in k-th cluster : $\theta(t) = \theta_k$ (i.e. θ is constant on cluster)

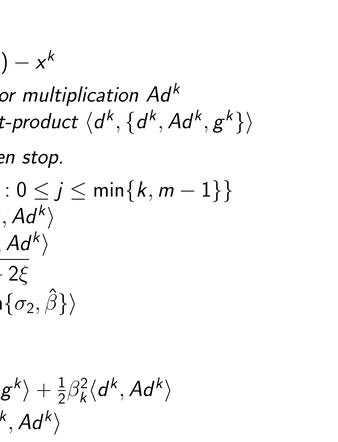
$$[\Theta^*,\Gamma^*(t)] = \arg\min_{\Theta,\Gamma} \int_0^T \sum_{k=1}^{K} \gamma_k(t)\cdot g(x(t),\theta_k)\ dt \quad \approx \quad \arg\min_{\Theta,\Gamma} \sum_{t=0}^{T}\sum_{k=1}^{K} \gamma_{k,t}\cdot g(x_t,\theta_k)$$

where each cluster is determined by

θ_k	vector of cluster stationary model parameters (depends on used model),
γ_k	model indicator function $\gamma_k(t)\in\{0,1\}$, $\gamma_k(t) = \begin{cases} 1 & \text{if } k\text{-th cluster is active in } t \\ 0 & \text{if } k\text{-th cluster is inactive in } t \end{cases}$ and exactly one cluster is active in t

2.) continuous real-valued functions $\gamma_k(t)\in[0,1]$

$$0\le\gamma_k(t)\le 1, \quad \forall t: \sum_{k=1}^{K}\gamma_k(t) = 1 \qquad (1)$$



3.) smooth $\gamma_k(t)$ $\Rightarrow$ regularisation, $\|\gamma\|_{H1}\le\|x\|_{H1}$

$$\int_0^T \sum_{k=1}^{K}\gamma_k(t)\cdot g(x(t),\theta_k)\ dt + \varepsilon^2 \sum_{k=1}^{K}\int_0^T (\partial_t\gamma_k(t))^2 dt$$

$$\text{(using FEM hat functions)} \qquad \approx \sum_{t=0}^{T}\sum_{k=1}^{K}\gamma_{k,t}\cdot g(x_t,\theta_k) + \varepsilon^2\sum_{k=1}^{K}\sum_{t=0}^{T-1}(\gamma_{k,t+1}-\gamma_{k,t})^2$$

Solving the problem

The optimisation problem is given by

$$[\Theta^*,\Gamma^*] = \arg\min_{\Theta,\Gamma}\sum_{t=0}^{T}\sum_{k=1}^{K}\gamma_{k,t}\cdot g(x_t,\theta_k) + \varepsilon^2\sum_{k=1}^{K}\sum_{t=0}^{T-1}(\gamma_{k,t+1}-\gamma_{k,t})^2 \quad \text{s.t. } 0\le\gamma_{k,t}\le 1,\ \forall t:\sum_{k=1}^{K}\gamma_{k,t} = 1. \qquad (2)$$

This nonconvex problem (2) could be solved iteratively as a sequence of solution of two optimisation problems, see Algorithm 1.

set feasible initial approximation Γ_0

while $\|L(\Gamma_{it},\Theta_{it}) - L(\Gamma_{it-1},\Theta_{it-1})\| \ge \varepsilon$

solve $\Theta_{it} = \arg\min_{\Theta} L(\Theta,\Gamma_{it-1})$ *(with fixed* Γ_{it-1}*)*

solve $\Gamma_{it} = \arg\min_{\Gamma} L(\Theta_{it},\Gamma)$ *(with fixed* Θ_{it}*)*

$it = it+1$

endwhile

Algorithm 1: **Outer algorithm.**

$$\Gamma_{it} = \arg\min \tfrac{1}{2}\gamma^T H\gamma + g^T\gamma$$
$$\text{subject to} \quad \forall t: \sum_{k=1}^{K}\gamma_k^t = 1, \gamma\ge 0. \qquad (3)$$
$$H\in\mathbb{R}^{K\cdot T,K\cdot T}, \quad g,\gamma\in\mathbb{R}^{K\cdot T}$$

$$AIC(L,\Theta,K) = -2\ln L + 2(\text{sizeof}(\Theta)+K) \qquad (4)$$

Given cost function $f:\mathbb{R}^n\to\mathbb{R}$*, initial approximation* $x^0\in\Omega$*, projection onto feasible set* $P_\Omega(x)$*, parameters* $m\in\mathbb{N},\gamma\in(0,1)$*, safeguarding parameters* $\sigma_1,\sigma_2\in\mathbb{R}: 0<\sigma_1<\sigma_2<1$*, precision* $\varepsilon>0$*, and initial step-size* $\alpha_0>0$*.*

$k := 0$
$g^0 := Ax^0 - b$
$f^0 := 1/2\langle g^0 - b, x^0\rangle$
for $k = 0,1,\ldots$
$d^k := P_\Omega(x^k - \alpha_k g^k) - x^k$
compute matrix-vector multiplication Ad^k
compute multiple dot-product $\langle d^k,\{d^k,Ad^k,g^k\}\rangle$
if $\sqrt{\langle d^k,d^k\rangle}\le\varepsilon$ *then stop.*
$f_{\max} := \max\{f(x^{k-j}) : 0\le j\le\min\{k,m-1\}\}$
$\xi := (f_{\max} - f^k)/\langle d^k,Ad^k\rangle$
$\bar\beta := -\langle g^k,d^k\rangle/\langle d^k,Ad^k\rangle$
$\hat\beta := \gamma\bar\beta + \sqrt{\gamma^2\bar\beta^2 + 2\xi}$
choose $\beta_k\in\langle\sigma_1,\min\{\sigma_2,\hat\beta\}\rangle$
$x^{k+1} := x^k + \beta_k d^k$
$g^{k+1} := g^k + \beta_k Ad^k$
$f^{k+1} := f^k + \beta_k\langle d^k,g^k\rangle + \frac{1}{2}\beta_k^2\langle d^k,Ad^k\rangle$
$\alpha_{k+1} := \langle d^k,d^k\rangle/\langle d^k,Ad^k\rangle$
$k := k+1$
endwhile

Return approximation of solution x^k*.*

Algorithm 2: **Spectral projected gradient method for QP.**

Please notice that the second problem in Algorithm 1 (i.e. Γ-problem) is independent of selected model. The model indicator functions Γ are represented by vector of dimension $K\cdot T$ and they have to fullfill conditions (1). The obtained optimisation problem is Quadratic Programming problem with SPS Hessian matrix and feasible set defined by simplex (3). Solving this problem is the most time-consuming operation.

In our library, we are using a Spectral Projected Gradient method (see Martinez et al. [8]) simplified for QP problems (see Algorithm 2) developed by Pospisil [7]. The algorithm is based on the solving the sequence of projection problems and since the feasible set is described by separable simplex constraints (of dimension K), this system extends the granularity of the solution process and it is suitable for GPU.

Since the number of clusters K is unknown, the optimisation problem (2) with different choice of these parameters has to be solved. Also different choise of initial Γ_0 leads to different results. These problems are completely independent and leads to the straightforward parellelisation. In the end of solution process, the solution with the lowest AIC number (4) is chosen as a solution of the original problem.

Acknowledgements

This work is supported by Platform for Advanced Scientific Computing (PASC). We would like to thank Olga Kaiser, Dimitri Igdalov, Ganna Marchenko, Ben Cumming, and Patrick Sanan for their collaboration in the project.

Parallelisation

The basic layout of the data distribution is presented in Figure 1. We naturally split data x of the problem through the largest parameter of the problem - the length of time-series T. In the same way, we also split corresponing cluster indicator functions γ. Since the number of model parameters is small (the aim of the optimisation is not only to fit the model to obtain the smallest error but also use as few parameters as possible, see AIC (4)). Therefore, each node could own the whole vector of parameters of all models θ. However, using this approach, nodes have to communicate during the inner optimisation problem objects assembly process in every outer iteration. The most time-consuming operation (the projection onto feasible set in QP) is performed locally. Since the matrix of the QP is block diagonal Laplace matrix, there is only a small communication during multiplication process. Moreover, this parallelisation aproach provides us an opportunity to manipulate with data which cannot be stored locally (long time-series).

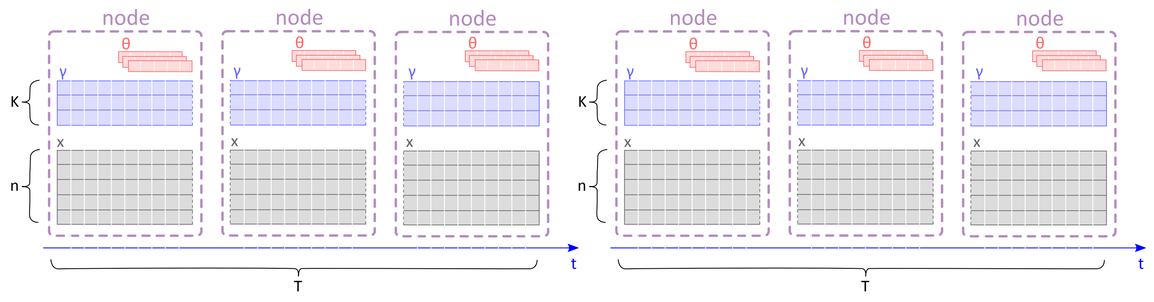


Figure 1: Computation on large-scale time-series data. We naturally split the data and indicator functions through the time-series. There are running several parallel jobs (computational units), which solve the problem with different initial conditions / number of clusters / integer model parameters.

In furture work, we create a management on the top of this approach. Each computing unit will consist of the set of nodes operating together on one data vector solving one particular problem of given K and the set of random initial values of γ.
Also this parallel time-series splitting layout brings naturally the idea of the domain decomposition methods for solving not only inner QP , but also a whole original optimisation problem (2).

Toy model: Geometrical time-dependent clustering with K-means model

Geometrical clustering problem represents the most basic modelling functions - constant function. We are trying to model the given data using the one value in every cluster in least-square sence

$$\forall t\in T_k: x_t = \theta_k + \epsilon_t \quad L(\theta_1,\ldots,\theta_K,\Gamma) = \sum_{t=0}^{T}\sum_{k=1}^{K}\gamma_k(t)\|x_t-\theta_k\|^2 \ \to \min$$

The early Matlab implementation revealed the main algorithm challenges, see Figure 2. The inner problems in Algorithm 1 could reuse the solution from previous outer iteration as an initial approximation in new solution process. Moreover, it is not necessary to solve problems exactly and adaptive precision control should be proposed and implemented.

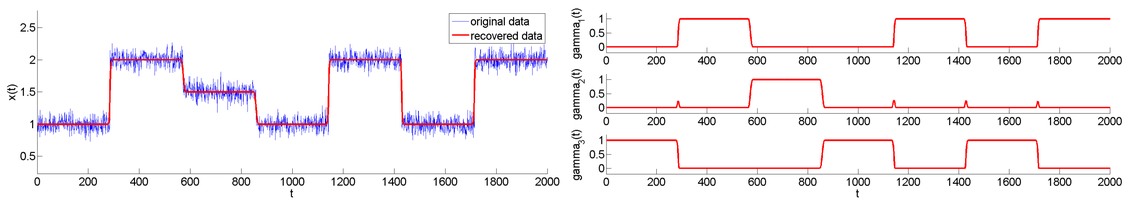


Figure 2: Initial example: one-dimensional K-means problem with $K=3, T=2000$ solved by FEM-BV-H1 in Matlab. The problem is solved in 4 outer iterations, the inner QP problem (of dimension $K\cdot N = 6000$) is solved by Matlab *quadprog* solver. Let us remark that the Matlab implementation of *'interior-point-convex'* algorithm is not able to use approximation of solution from previous outer iteration as an initial guess of the solution. Moreover, it is not able to control the precision based on the decrease of the objective function. In projected gradient methods (like SPG-QP), we are able to control the decrease in every iteration as well as use initial approximation.

We implement Algorithm 1 and Algorithm 2 in PETSc framework and solve K-means problem on 2 nodes on PIZ Daint machine (Intel Xeon E5-2670 (8 cores, 32GB) with NVIDIA Tesla K20X (2688 cores, 6GB)) . The results are presented in Figures 3, 4, 5, and 7.

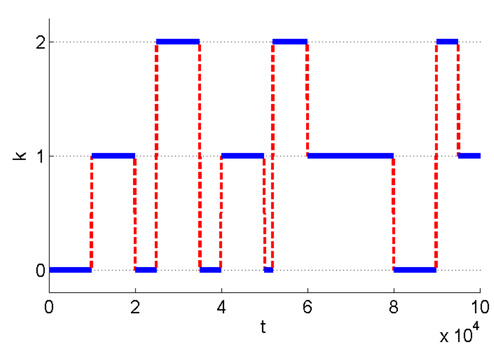


Figure 3: Large-scaled example: Switching schema for generating testing benchmark.

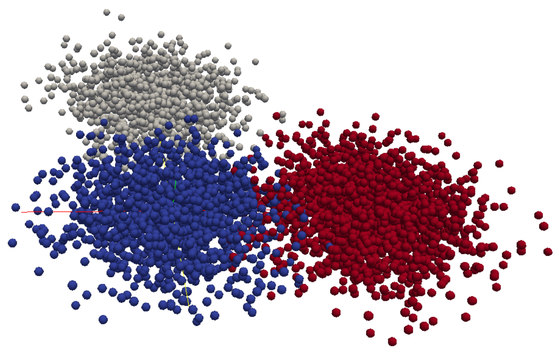


Figure 4: The solution: clustered data. The affiliation to the cluster is decided by the maximum value of indicator functions $\gamma_0,\gamma_1,\gamma_2$ (see Figure 5). For visualisation, we used VTK format openable in Paraview.

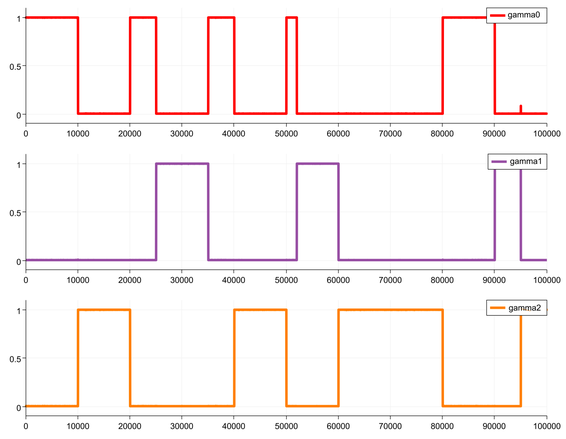


Figure 5: The solution: indicator model function in FEM-BV-H1 model found using our new library. Library generates files suitable for ParaView.

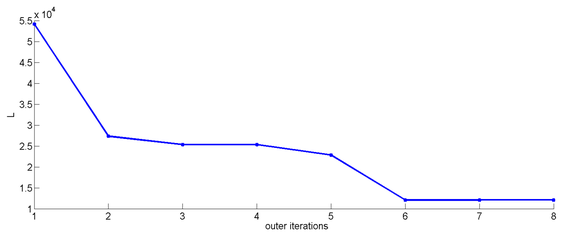
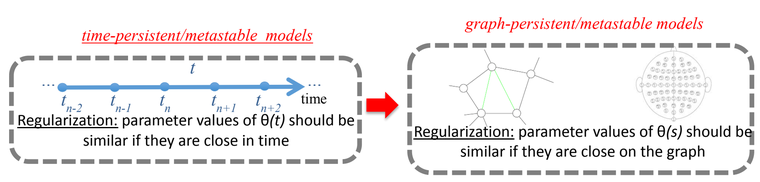


Figure 6: Monotone decrease of global objective function during outer iterations. Our algorithm is terminated when the difference of objective function is less than 10^{-4}.

Future work: from Time-Regularisation to Graph-Regularisation



$$H = 2\text{diag}\left(\textstyle\sum|W_{i,:}|\right) - 2W, \quad W = G\circ B, \quad B = \text{3-diag}(1,1,1)$$
$$G - \text{ graph connectivity matrix}, \quad \circ - \text{ Kronecker product}$$

Matrix-vector multiplication still can be written in matrix-free form and effectivelly performed on GPU. Moreover, the constraints of the QP problem remain the same, therefore the projection problem could be solved in full-parallel way.

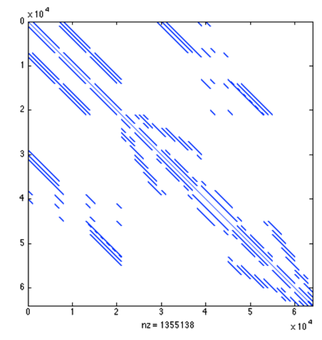


Figure 7: Example: pattern of Hessian matrix in Graph-regularisation.

References

[1] P. Brockwell and R. Davis. *Introduction to Time Series and Forecasting*. Springer, Berlin, 2002.
[2] C. Granger. *Investigating causal relations by econometric models and cross-spectral methods*. Econometrica, 37:424–438, 1969.
[3] I. Horenko. *Finite Element Approach to Clustering of Multidimensional Time Series*. SIAM J. Sci. Comp. 32, 62–83, 2010.
[4] S. Gerber and I. Horenko. *On inference of causality for discrete state models in a multiscale context*. Proc. Natl. Acad. Sci. USA (PNAS), 2014.
[5] S. Gerber and I. Horenko. *Improving Clustering by Imposing Network Information*. Sciences Advances (AAAS), 1(7):e1500163, 2015
[6] P. Metzner, L. Putzig, and I. Horenko. *Analysis of persistent non-stationary time series and applications*. CAMCoS 7, 175–229, 2012.
[7] L. Pospisil. *Development of Algorithms for Solving Minimizing Problems with Convex Quadratic Function on Special Convex Sets and Applications*. PhD thesis, supervised by Z. Dostal, VSB-TU Ostrava, 2014.
[8] E.G. Birgin, J.M Martinez, and M.M. Raydan. *Nonmonotone spectral projected gradient methods on convex sets*. SIAM Journal on Optimization 10, 1196–1211, 2000.
[9] Y. Chen, X. Ye. *Projection Onto A Simplex*. Arxiv. 1101.6081, 2012.