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?

Understanding Causality

In its purest deterministic sense, there is a causal relationship between events if one implies that another has occurred. In reality, such causal relationships can generally only be determined from first principles on a small spatial/temporal scale, with little data involvement. For larger scales and/or larger data involvement, mathematical or statistical models are required from whence causality can be inferred.

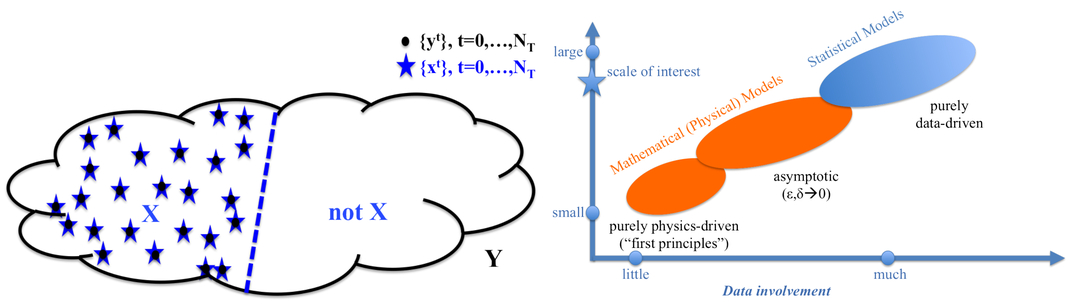

Figure 1: Deterministic Causality (left): X has a causality impact on Y if for any t, event y^t is happening if and only if event x^t happened. In real applications a data model is required, but the realm of applicability (right) determines whether the causality inference is driven by the model or by the data .

Granger Causality: The standard causality inference measures the ability of predicting the future values of a time series (e.g., $y^{t+\tau}$) using past values of another time series (e.g., $x^t, x^{t-\tau}, \ldots, x^{t-q\tau}$), where τ is a time step and $q\tau$ is a maximal time lag [2]. The predictive models that have been deployed first to measure such a causality relation between y and x were linear **A**uto**R**egressive models with e**X**ternal impact factors (ARX-models) [1]:

$$y^{t+\tau} = \mu + \sum_{i=0}^{p} A_i y^{t-i\tau} + \sum_{j=0}^{q} B_j x^{t-j\tau} + \sigma(t), \tag{1}$$

where $\{\mu, A_0, A_1, \ldots, A_p, B_0, B_1, \ldots, B_q\}$ are the ARX-model parameters that can be estimated from the available time series for x and y (e.g., deploying the maximum likelihood method) and $\sigma(t)$ is some stochastic noise process [1]. Then, the variable x has a Granger causality relation to the variable y if and only if at least one of the B_j is statistically-significantly different from zero.

Limitations: Granger causality may lead to biased results. This bias might be amplified and lead to a completely wrong inference of the causality relations from the data, if the underlying model assumptions of standard causality inference methods (e.g., like the intrinsic linearity of the Granger causality measures) are not fulfilled.

Algorithm: approximation through well-posed lower bound

Assuming y^t being statistically-independent in t sequence of binary variables or observed probabilities (conditioned on the knowledge of variables x and u), inference of both the unknown causality vector $\Lambda(t)$ and of the unknown probability process P_y^t for the discrete state model

$$P_y^t = \Lambda^\dagger(t) P_x^t,$$

can be done via a maximisation w.r.t. $\Lambda(t)$ of the following log-likelihood functional:

$$\mathcal{L} = \max_{\Lambda} \left(\sum_{t=0}^{N_T} \left[(1-y^t) \ln\left(1-\Lambda^\dagger(t)P_x^t\right) + y^t \ln\left(\Lambda^\dagger(t)P_x^t\right) \right] \right). \tag{2}$$

This problem is unfortunately ill-posed. The central challenge is to formulate the investigations of Λ_i into a well-posed optimisation problem. We start with the approximation:

$$\Lambda(t) = \sum_{i=1}^{K} \gamma_i^t \Lambda_i \quad \text{with} \quad \sum_{i=1}^{K} \gamma_i^t = 1, \quad \gamma_i^t \geq 0.$$

In [4] the following lower bound for $\mathcal{L}$ was proved:

$$\mathcal{L} \geq I^\epsilon = \max_{\gamma_i^t, \Lambda_i} \left(\sum_{i=1}^{\mathbf{K}} \left[\sum_{t=0}^{N_T} \gamma_i^t g(t, \Lambda_i) - \epsilon^2 \sum_{j=1}^{n} \lambda_i^{(j)} \right] \right). \tag{3}$$

with and subject to the constraints:

$$\sum_{i=1}^{\mathbf{K}} \gamma_i^t = 1, \gamma_i^t \geq 0 \text{ for all } t \text{ and } i, \quad 0 < \sum_{j=1}^{n} \lambda_i^{(j)} < 1, \quad 0 \leq \lambda_i^{(j)} \leq 1, \text{ for all } i \text{ and } j, \quad \sum_{t_1,t_2=0}^{N_T} |\gamma_i^{t_1} - \gamma_i^{t_2}| \leq \bar{\mathbf{C}}(N_T), \text{ for all } i. \tag{4}$$

The maximisation of I^ϵ belongs to the well-posed class of FEM-BV-problems.

High Performance Computing Implementation

We start the development of our new HPC Causality library implementing more simplier models. The Granger Causality could be used for modelling general whole set of non-stationary models solving the general optimisation problem with *average cluster functional*

$$L(\theta_1, \ldots, \theta_K, \Gamma) = \sum_{t=m}^{T} \sum_{k=1}^{K} \gamma_k^t g(x^t, \ldots, x^{t-m}, \theta_k) \quad \to \min \tag{5}$$

where θ_k represents parameters of the model on k-th cluster (unknown) and γ_k are *model indicator functions* (unknown). These functions represent the presence of appropriate model on k-th cluster. The error of the model on k-th cluster is computed via *model error functions* g. For complete review and applications see Metzner et. al [5]. Also presented problem (2) could be considered as a special case of optimisation problem (5). The nonconvex problem (5) could be solved iteratively as a sequence of solution of two optimisation problems, see Algorithm 1.

set feasible initial approximation Γ_0

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

$$\Gamma_{it} = \arg\min \tfrac{1}{2}\gamma^T H \gamma + g^T \gamma \quad \text{subject to} \quad \forall t : \sum_{k=1}^{K} \gamma_k^t = 1, \gamma \geq 0. \tag{6}$$

$$AIC(L, \Theta, K) = -2 \ln L + 2(\text{sizeof}(\Theta) + K) \tag{7}$$

Given initial approximation $x^0 \in \Omega$, *parameters* $m \in \mathbb{N}, \gamma \in (0,1)$, *safeguarding parameter* $\sigma_2 \in (0,1)$, *and initial step-size* $\alpha_0 > 0$.

$k := 0$
$g^0 := Ax^0 - b$
$f^0 := 1/2\langle g^0 - b, x^0 \rangle$
while $\|\widetilde{g}_\alpha(x)\|$ *is not sufficiently small*
  $d^k := P(x^k - \alpha_k g^k) - x^k$
  $f_{\max} := \max\{f(x^{k-j}) : 0 \leq j \leq \min\{k, m-1\}\}$
  $\xi := (f_{\max} - f^k)/\langle Ad^k, d^k \rangle$
  $\bar{\beta} := -\langle g^k, d^k \rangle / \langle Ad^k, d^k \rangle$
  $\hat{\beta} := \gamma\bar{\beta} + \sqrt{\gamma^2\bar{\beta}^2 + 2\xi}$
  choose $\beta_k \in [\sigma_1, \min\{\sigma_2, \hat{\beta}\}]$
  $x^{k+1} := x^k + \beta_k d^k$
  $g^{k+1} := g^k + \beta_k Ad^k$
  $f^{k+1} := 1/2\langle g^{k+1} - b, x^{k+1} \rangle$
  $\alpha_{k+1} := \langle d^k, d^k \rangle / \langle Ad^k, d^k \rangle$
  $k := k + 1$
endwhile

Return approximation of solution x^k.

Algorithm 2: **Spectral projected gradient method for QP (SPG-QP).**

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.

Quadratic Programming problem

Please notice that the second problem (i.e. Γ-problem) is independent on selected model. The model indicator functions Γ represented by vector of dimension $K \cdot T$ have to fullfill conditions (4) with so-called BV norm. However, instead of using BV-norm, we can use Euclidean norm and appropriate FEM regularisation, see Horenko [3]. The obtained optimisation problem is Quadratic Programming problem with SPS Hessian matrix and feasible set defined by simplex (6). Solving this problem is the most time-consuming operation.
In our library, we are using a Spectral Projected Gradient method (see Martinez et al. [7]) simplified for QP problems (see Algorithm 2) developed by Pospisil [6]. 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.

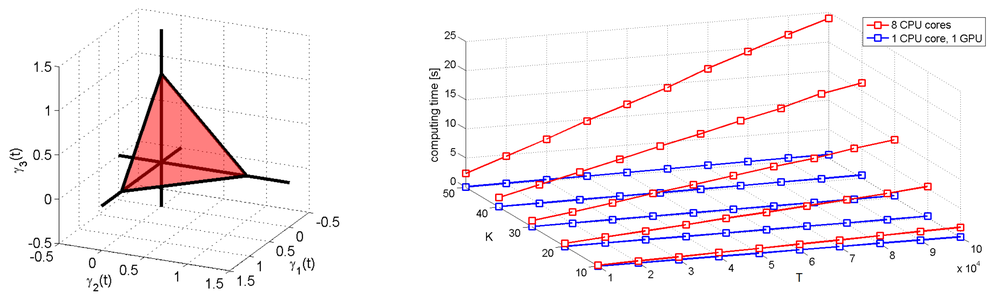

Figure 2: Projection onto feasible set: the feasible set in Quadratic Programming problem (Γ-problem in Algorithm 1) consists of separable simplexes; example for $K = 3$ is presented on left figure. We solve the problem of projection onto simplex using algorithm presented by Chen and Ye [8]. The number of iterations of this algorithm is upper bounded by K. Right figure shows the computing time of 100 random vectors of length $K \cdot T$ using one node on PIZ Daint machine (Intel Xeon E5-2670 (8 cores, 32GB) with NVIDIA Tesla K20X (2688 cores, 6GB)).

Parallelisation

Since the number of clusters K and the memory parameters p, q in model (1) are unknown, the optimization problem (5) 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 (7) is chosen as a solution of the original problem.

At the beginning of our implementation, we focus on more complicated problem. We suppose that the original data of time series are such large that it cannot be stored and operated on one node with shared memory, see Figure 3. However, the data of inner optimisation problems (Γ-problem and Θ-problem) have to be assembled in each outer iterations from data of time-series. Therefore, the communication (scattering of time-series) has to be handled. Our first implementation supposes that inner optimisation problems have still such a small dimension to be solved on one processor (and/or on one GPU card).

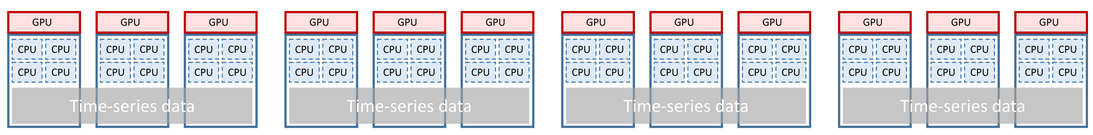

Figure 3: Computation on large-scale time-series data. We suppose that the data of time-series cannot be operated on one single node, therefore the communication is necessary. However, each node is still solving its own problem with given parameters.

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, but still each node will be able to compute its own optimisation problem with given K and given initial approximation Γ_0. Actual implementation consist of only one computing node, therefore in this time, we are still able to solve the problems on large data sets sending a independent computing jobs on supercomputer.

Numerical experiments - Geometrical clustering with Kmeans 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

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

The early Matlab implementation revealed the main algorithm challenges, see Figure 4. 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 implemented.

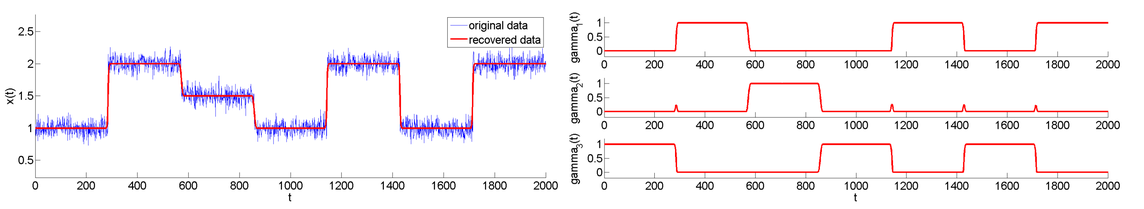

Figure 4: Initial example: one-dimensional K-means problem with $K = 3, T = 2000$ solved by FEM-BV-H1 in Matlab. The problem is solved in 4 outer iterations, the inner QP problem (of dimension $K \cdot N = 6000$) is solved by Matlab *quadprog* solver. Let us remark that the Matlab implementation of *'interior-point-convex'* 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 Kmeans 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 5, 6, 7, and 8.

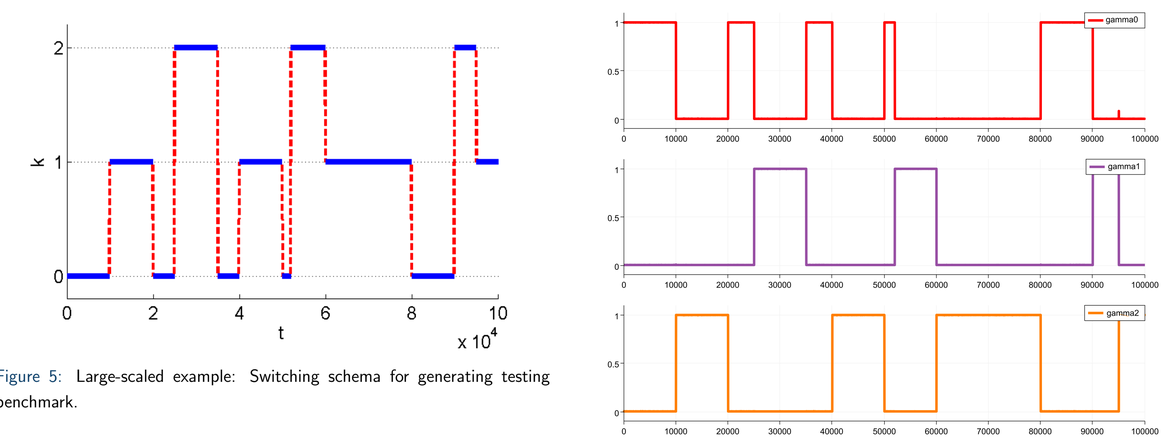

Figure 5: Large-scaled example: Switching schema for generating testing benchmark.

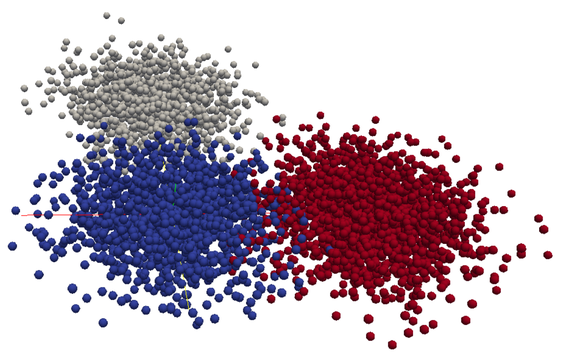

Figure 6: The solution - clustered data. The affiliation to the cluster is decided by the maximum value of indicator functions $\gamma_0, \gamma_1, \gamma_2$ (see Figure 7). For visualisation, we used VTK format openable in Paraview.

Figure 7: The solution - indicator model function in FEM-BV-H1 model found using our new library. Library generates files suitable for ParaView.

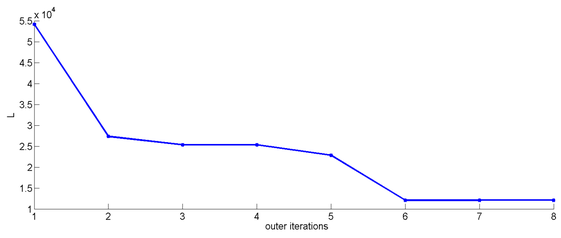

Figure 8: Monotone decrease of global objective function during outer iterations. Our algorithm is terminated when the difference of objective function is less than 10^{-4}.

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] P. Metzner, L. Putzig, and I. Horenko. *Analysis of persistent non-stationary time series and applications*. CAMCoS 7, 175–229, 2012.
[6] 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.
[7] 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.
[8] Y. Chen, X. Ye. *Projection Onto A Simplex*. Arxiv. 1101.6081, 2012.