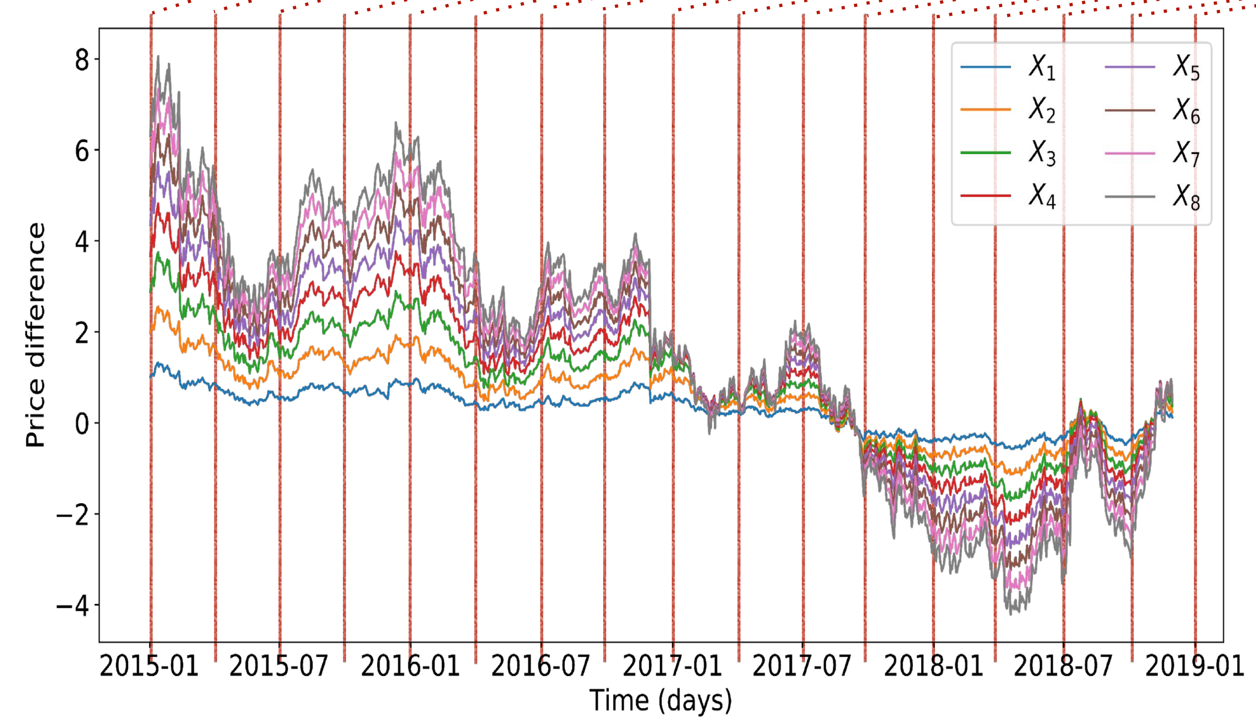
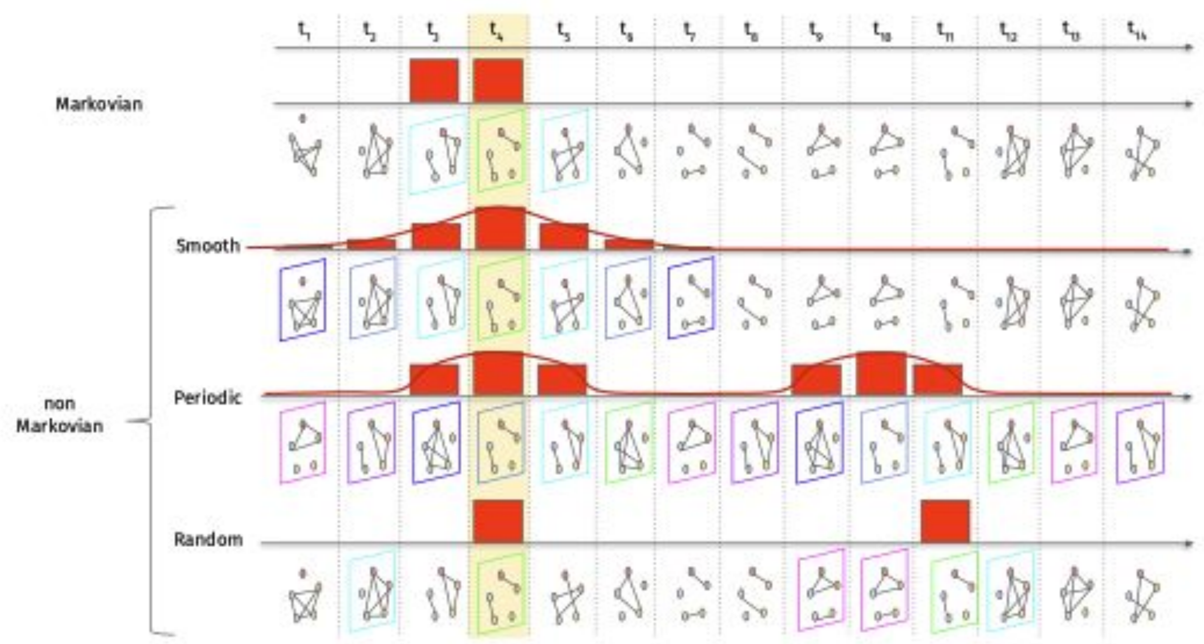
Temporal Pattern Detection in Time-Varying Graphical Models

Federico Tomasi, Spotify
Veronica Tozzo, Massachusetts General Hospital, Harvard Medical School
Annalisa Barla, Università di Genova

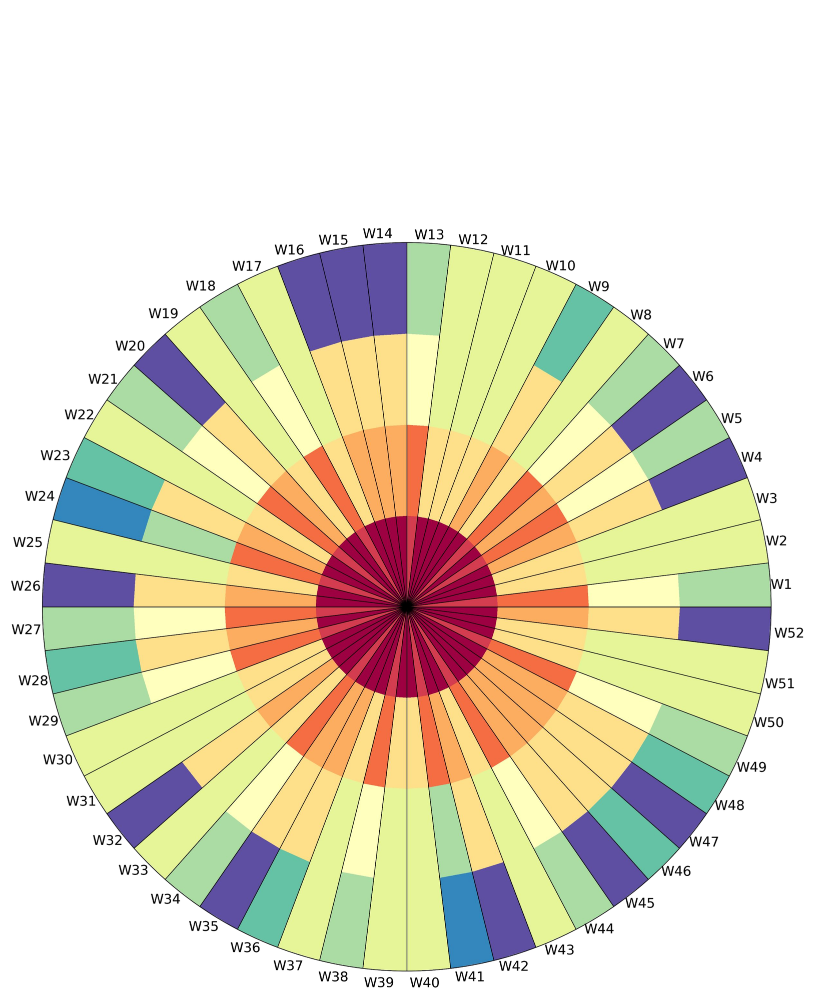
Real-world phenomena often include variables changing their behaviour in time. Such change can be captured by looking at their interactions.



Dependencies in time can be captured by kernel functions, capturing evolving relationships between not necessarily contiguous time steps.

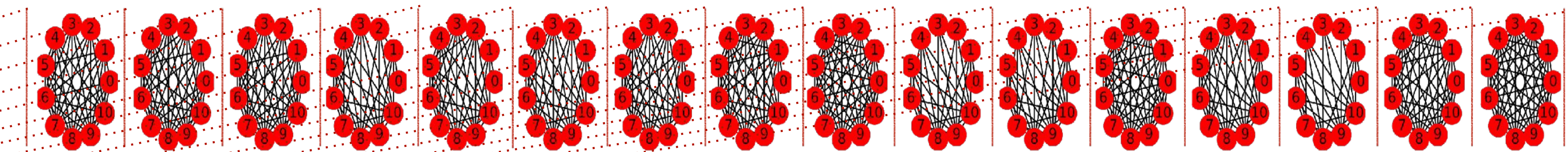


If the kernel function is not known, automatically inferring it from data can achieve clustering of temporal series.



The problem

At each time point t, we have n_t observations where each sample is a d-dimensional vector drawn from a multivariate Gaussian distribution. The goal is to infer a series of precision matrices $\mathbf{\Theta} = (\Theta_1, \ldots, \Theta_T)$ by guiding the inference through the imposition of consistency among time points that are dependent.



Temporal consistency and dependency

We decouple consistency and dependency by using a penalty that consists in a distance function $\mathbf{\Psi}$ and a kernel $\boldsymbol{\kappa}$. The former one models how similar the networks should be in time points that are dependent. The latter models different types of dependencies as gaussian, periodic or random.

$$P_{\Psi,\kappa}(\mathbf{\Theta}) = \sum_{t'>t}^{T} \kappa^{\Psi}(t,t')\ \Psi(\Theta_{t'} - \Theta_t) = \sum_{m=1}^{T-1}\sum_{t=1}^{T-m} \kappa^{\Psi}(m,t)\ \Psi(\Theta_{m+t} - \Theta_t)$$

Model

The problem then becomes the minimization of the following functional where the first part is the negative Gaussian log likelihood.

$$\underset{\Theta_t \in \mathcal{S}^d_{++}}{\text{minimize}} \sum_{t=1}^{T} \Big[-n_t\ \ell(S_t, \Theta_t) + \alpha \|\Theta_t\|_{od,1} \Big] + P_{\Psi,\kappa}(\mathbf{\Theta}),$$

if the temporal dependency pattern is known than the optimization problem is convex and it it optimized through the Alternating Direction Method of Multipliers. If the kernel is not know, it can be inferred from data with an alternating minimization procedure where we look at the pairwise similarity among networks

$$S^{\iota}[t,t'] = 1 - \frac{\Psi(K_t - K_{t'}) - \min\limits_{m,m'=1,\ldots,T} \Psi(K_m - K_{m'})}{\max\limits_{m,m'=1,\ldots,T} \Psi(K_m - K_{m'})}.$$

and we then use it as input to a hierarchical clustering algorithm which provides us with a connectivity matrix

$$C^{\iota}[t,t'] = \begin{cases} 1, & \text{if } t' \text{ and } t \text{ belong to the same cluster} \\ 0, & \text{otherwise} \end{cases}$$

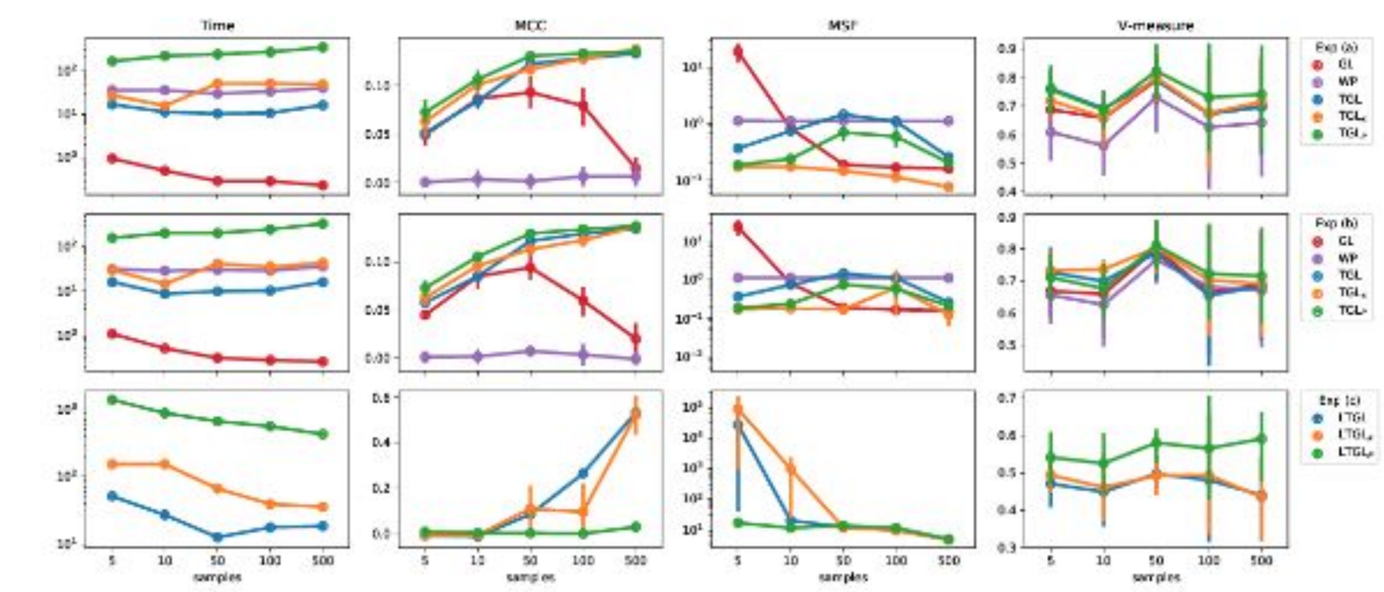
that is then use to build a kernel that guides the inference. The learned kernel also provides intuition on how the analysed process evolves in time.

Experiments

Results show how a temporal kernel can better approximate the evolution of a system.

TGL_κ and TGL_P (and their counterparts with latent variables) outperform the competitors in terms of MCC, MSE and V-measure across almost all experiments. We emphasise how, especially when $n \ll d$ (as in this case) prior information on the behaviour of the network is crucial for a reliable inference, and allows to keep a low mean MSE.

TGL_κ has high performance scores when data exhibit a clear pattern (experiment *(a)*). Indeed, the imposition of a specific periodic kernel improves average precision and MSE. Nonetheless, TGL_P (with automatic pattern discovery) outperforms in almost all measures both our competitors and TGL_κ by increasing the accuracy in structure inference and reducing the error in the estimation of the dynamical network



experiment		method	balanced accuracy	average precision	MCC	MSE	V-measure
(a) Periodic-pattern		GL	0.505 ± 0.002	0.029 ± 0.003	0.093 ± 0.017	0.190 ± 0.003	0.790 ± 0.096
		TGL	0.558 ± 0.005	0.301 ± 0.014	0.122 ± 0.003	1.470 ± 0.015	0.791 ± 0.089
		WP	0.498 ± 0.002	0.022 ± 0.001	0.002 ± 0.007	1.133 ± 0.003	0.730 ± 0.137
	Ours	TGL_κ	0.560 ± 0.008	**0.372 ± 0.046**	0.117 ± 0.005	**0.148 ± 0.007**	0.800 ± 0.090
	Ours	TGL_P	**0.577 ± 0.003**	0.341 ± 0.014	**0.130 ± 0.002**	0.707 ± 0.211	**0.822 ± 0.113**
(b) Random-pattern		GL	0.505 ± 0.001	0.029 ± 0.003	0.094 ± 0.014	0.191 ± 0.002	0.802 ± 0.088
		TGL	0.560 ± 0.004	0.299 ± 0.011	0.122 ± 0.004	1.475 ± 0.012	0.788 ± 0.093
		WP	0.497 ± 0.003	0.022 ± 0.001	0.007 ± 0.005	1.130 ± 0.006	0.766 ± 0.099
	Ours	TGL_κ	0.553 ± 0.007	0.284 ± 0.033	0.113 ± 0.005	**0.173 ± 0.017**	0.804 ± 0.088
	Ours	TGL_P	**0.574 ± 0.004**	**0.331 ± 0.015**	**0.130 ± 0.003**	0.758 ± 0.181	**0.811 ± 0.081**
(c) Conditioned-random pattern		LTGL	0.509 ± 0.002	0.278 ± 0.003	0.082 ± 0.010	12.229 ± 0.279	0.497 ± 0.037
	Ours	$LTGL_\kappa$	**0.521 ± 0.016**	**0.299 ± 0.045**	**0.107 ± 0.117**	**11.861 ± 0.483**	0.492 ± 0.058
	Ours	$LTGL_P$	0.500 ± 0.000	0.251 ± 0.002	-0.001 ± 0.002	13.711 ± 0.117	**0.580 ± 0.047**