Graphs in State-Space Models for Granger Causality

(in the Earth, Climate and Social systems)

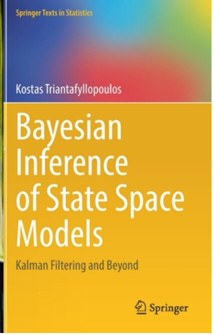
Víctor Elvira | Edinburgh
Émilie Chouzenoux | Paris-Saclay
Jordi Cerdà & Gustau Camps-Valls | València

http://isp.uv.es @isp_uv_es gustau.camps@uv.es

Outline

- Introduction
- GraphEM
- Experiments
- Causeme.net
- Conclusions

Introduction



- **Motivation**:
  - Sequential processing of observed multivariate data is everywhere!
  - Interrelated random processes: one is observed + one is hidden
  - State-space models (SSMs) → Linear-Gaussian state-space model
    - e.g Kalman filter is a simple & efficient inference procedure
- **Challenges:**
  - Inference algorithms in SSMs need model parameters to be known
  - Joint estimation of parameters & transition matrix is difficult

This talk

- **What?**
  - Estimate the transition matrix in the linear-Gaussian SSM
  - Relate the transition matrix to adjacency matrix of a directed graph
  - Connections represent (causal) dependencies between the states
- **How?**
  - Develop an efficient Expectation-Minimization (EM) methodology
  - Estimate transition matrix assuming a sparse graph model
- **What for?**
  - Wide range of problems in Earth, weather, climate, social sciences

The linear-Gaussian Model

- *Deterministic notation*
  - Unobserved state $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{q}_k$
  - Observations $\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{r}_k$

    where $\mathbf{q}_k \sim \mathcal{N}(0, \mathbf{Q})$ and $\mathbf{r}_k \sim \mathcal{N}(0, \mathbf{R})$
- *Probabilistic notation*
  - Hidden state $\mathcal{N}(\mathbf{x}_k; \mathbf{A}\mathbf{x}_{k-1}, \mathbf{Q})$
  - Observations $\mathcal{N}(\mathbf{y}_k; \mathbf{H}\mathbf{x}_k, \mathbf{R})$

On the transition matrix & Granger causality

- *Deterministic notation*
  - Unobserved state $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{q}_k$
  - Observations $\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{r}_k$

  where $\mathbf{q}_k \sim \mathcal{N}(0, \mathbf{Q})$ and $\mathbf{r}_k \sim \mathcal{N}(0, \mathbf{R})$
- *(i,j)* entry in $\mathbf{A}$ encodes the weight in which *j-th* time series in the hidden state affects the *i-th* time series in the next step (0 for no Granger effect)
- $\mathbf{A}$: 1) is high-dimensional, 2) controls the AR process of the hidden state, and 3) related to the inner structure of the system (my prior!)

The linear-Gaussian Model - inference

- **Kalman filter (forward)**
  – predicted and filtered distributions are Gaussian

$$p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) \qquad\qquad p(\mathbf{x}_k | \mathbf{y}_{1:k})$$

- **Rauch-Tung-Striebel (RTS) smoother (backward)**
  – also Gaussian, by processing the observations backward

$$p(\mathbf{x}_k | \mathbf{y}_{1:K})$$

Kalman Filter and RTS smoother

Kalman filter

- Initialize: $\mathbf{m}_0$, $\mathbf{P}_0$
- For $k = 1, \ldots, K$

  Predict stage:

$$\mathbf{x}_k^- = \mathbf{A}\mathbf{m}_{k-1}$$
$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^\top + \mathbf{Q}$$

  Update stage:

$$\mathbf{z}_k = \mathbf{y}_k - \mathbf{H}\mathbf{x}_k^-$$
$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_k^-\mathbf{H}^\top + \mathbf{R}$$
$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^\top\mathbf{S}_k^{-1}$$
$$\mathbf{m}_k = \mathbf{x}_k^- + \mathbf{K}_k\mathbf{z}_k$$
$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top$$

RTS smoother

- For $k = K, \ldots, 1$

  Smoothing stage:

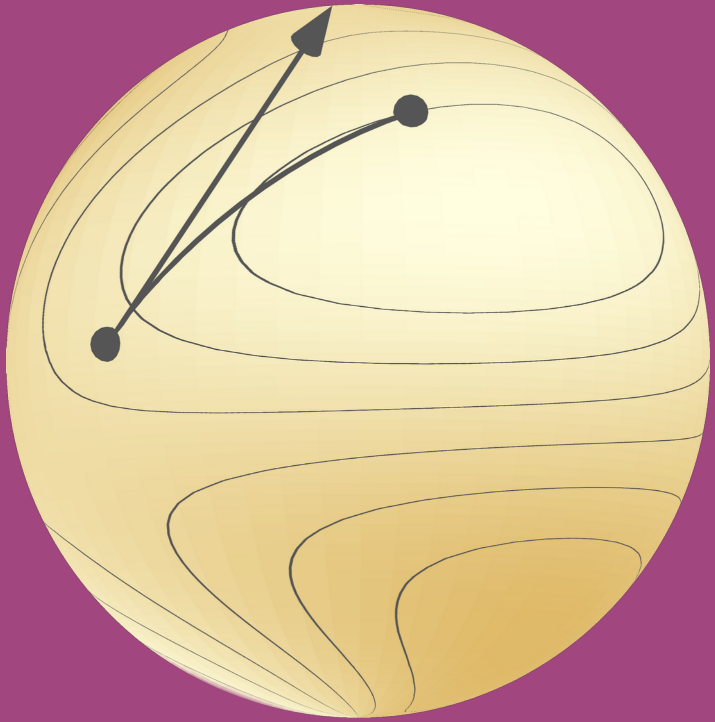
$$\mathbf{x}_{k+1}^- = \mathbf{A}\mathbf{m}_k$$
$$\mathbf{P}_{k+1}^- = \mathbf{A}\mathbf{P}_k\mathbf{A}^\top + \mathbf{Q}$$
$$\mathbf{G}_k = \mathbf{P}_k\mathbf{A}^\top(\mathbf{P}_{k+1}^-)^{-1}$$
$$\mathbf{m}_k^s = \mathbf{m}_k + \mathbf{G}_k(\mathbf{m}_{k+1}^s - \mathbf{x}_{k+1}^-)$$
$$\mathbf{P}_k^s = \mathbf{P}_k + \mathbf{G}_k(\mathbf{P}_{k+1}^s - \mathbf{P}_{k+1}^-)\mathbf{G}_k^\top$$

✓ Filtering distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_k, \mathbf{P}_k)$

✓ Smoothing distribution $p(\mathbf{x}_k|\mathbf{y}_{1:K}) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_k^s, \mathbf{P}_k^s)$

✗ What if the state matrix $\mathbf{A}$ is unknown?

GraphEM algorithm



Goal and challenge

- **Goal**: find the MAP estimate of $\mathbf{A}$ given the observed data

$$p(\mathbf{A}|\mathbf{y}_{1:K}) \propto p(\mathbf{A})p(\mathbf{y}_{1:K}|\mathbf{A})$$

Equivalent to: minimize

$$\varphi_K(\mathbf{A}) = -\log p(\mathbf{A}) - \log p(\mathbf{y}_{1:K}|\mathbf{A})$$

- **Challenge**: estimating $p(\mathbf{y}_{1:K}|\mathbf{A})$ requires to run Kalman filter

$$\begin{aligned}\varphi_k(\mathbf{A}) &= \varphi_{k-1}(\mathbf{A}) - \log p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \mathbf{A}) \\ &= \varphi_{k-1}(\mathbf{A}) + \frac{1}{2}\log|2\pi\mathbf{S}_k(\mathbf{A})| + \frac{1}{2}\mathbf{z}_k(\mathbf{A})^\top \mathbf{S}_k(\mathbf{A})^{-1}\mathbf{z}_k(\mathbf{A})\end{aligned}$$

Goal and challenge

- **Goal**: find the MAP estimate of $\mathbf{A}$ given the observed data

$$p(\mathbf{A}|\mathbf{y}_{1:K}) \propto p(\mathbf{A})p(\mathbf{y}_{1:K}|\mathbf{A})$$

Equivalent to: minimize

$$\varphi_K(\mathbf{A}) = -\log p(\mathbf{A}) - \log p(\mathbf{y}_{1:K}|\mathbf{A})$$

- **Challenge**: estimating $p(\mathbf{y}_{1:K}|\mathbf{A})$ requires to run Kalman filter

$$\varphi_k(\mathbf{A}) = \varphi_{k-1}(\mathbf{A}) - \log p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \mathbf{A})$$

$$= \varphi_{k-1}(\mathbf{A}) + \frac{1}{2}\log|\mathbf{S}_k(\mathbf{A})| + \frac{1}{2}\mathbf{z}_k(\mathbf{A})^\top \mathbf{S}_k(\mathbf{A})^{-1}\mathbf{z}_k(\mathbf{A})$$

Non-tractable minimization

GraphEM strategy

- **EM strategy**:
  - Minimize a sequence of tractable approximations of φ_K
  - Do it via satisfying a majorizing property
- **LASSO regularization**:
  - choose the prior to favor sparse matrix $\mathbf{A}$
  - reveal interpretable and compact network of interdependencies

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad \varphi_0(\mathbf{A}) = \gamma \|\mathbf{A}\|_1, \qquad \gamma > 0$$

[Elvira et al, 2022]

The E step: Majorizing approximation of φ_K

1) Run Kalman filter/RTS smoother by setting the state matrix to $\mathbf{A}'$

$$\mathbf{\Sigma} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{P}_k^s + \mathbf{m}_k^s(\mathbf{m}_k^s)^\top \qquad \mathbf{\Phi} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{P}_{k-1}^s + \mathbf{m}_{k-1}^s(\mathbf{m}_{k-1}^s)^\top$$

$$\mathbf{C} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{P}_k^s\mathbf{G}_{k-1}^\top + \mathbf{m}_k^s(\mathbf{m}_{k-1}^s)^\top$$

2) Build $\mathcal{Q}(\mathbf{A};\mathbf{A}') = \frac{K}{2}\text{tr}\left(\mathbf{Q}^{-1}(\mathbf{\Sigma} - \mathbf{C}\mathbf{A}^\top - \mathbf{A}\mathbf{C}^\top + \mathbf{A}\mathbf{\Phi}\mathbf{A}^\top)\right) + \varphi_0(\mathbf{A}) + \mathcal{C}$
with the prior $\varphi_0(\mathbf{A}) = -\log p(\mathbf{A})$
such that $\mathcal{Q}(\mathbf{A};\mathbf{A}') \geq \varphi_K(\mathbf{A}), \qquad \mathcal{Q}(\mathbf{A}';\mathbf{A}') = \varphi_K(\mathbf{A}')$ [Sarkka 2013]

The M step: Upper bound optimization

- **Goal**: search for a minimizer of $\mathcal{Q}(\mathbf{A};\mathbf{A}')$ with respect to $\mathbf{A}$

$$\operatorname{argmin}_{\mathbf{A}} \underbrace{\frac{K}{2}\operatorname{tr}\left(\mathbf{Q}^{-1}(\mathbf{\Sigma} - \mathbf{C}\mathbf{A}^\top - \mathbf{A}\mathbf{C}^\top + \mathbf{A}\mathbf{\Phi}\mathbf{A}^\top)\right)}_{f_1(\mathbf{A})} + \underbrace{\gamma\|\mathbf{A}\|_1}_{f_2(\mathbf{A})}$$

- **Problem:** Convex non-smooth minimization problem!

The M step: Upper bound optimization

- **Goal**: search for a minimizer of $\mathcal{Q}(\mathbf{A};\mathbf{A}')$ with respect to $\mathbf{A}$

$$\mathrm{argmin}_{\mathbf{A}} \underbrace{\frac{K}{2}\mathrm{tr}\left(\mathbf{Q}^{-1}(\mathbf{\Sigma} - \mathbf{C}\mathbf{A}^\top - \mathbf{A}\mathbf{C}^\top + \mathbf{A}\mathbf{\Phi}\mathbf{A}^\top)\right)}_{f_1(\mathbf{A})} + \underbrace{\gamma\|\mathbf{A}\|_1}_{f_2(\mathbf{A})}$$

- **Alternative**:
  - Proximal splitting approach [Combettes and Pesquet, 2010]

$$\mathrm{prox}_f(\widetilde{\mathbf{A}}) = \mathrm{argmin}_{\mathbf{A}} \left(f(\mathbf{A}) + \frac{1}{2}\|\mathbf{A} - \widetilde{\mathbf{A}}\|_F^2\right)$$

  - Douglas-Rachford algorithm [Benfenati et al., 2020] - http://proximity-operator.net

The GraphEM in a nutshell

GraphEM algorithm

- Initialization of $\mathbf{A}^{(0)}$.
- For $i = 1, 2, \ldots$

E-step Run the Kalman filter and RTS smoother by setting $\mathbf{A}' := \mathbf{A}^{(i-1)}$ and construct $\mathcal{Q}(\mathbf{A}; \mathbf{A}^{(i-1)})$.

M-step Update $\mathbf{A}^{(i)} = \text{argmin}_{\mathbf{A}} \left(\mathcal{Q}(\mathbf{A}; \mathbf{A}^{(i-1)})\right)$ using Douglas-Rachford algorithm.

- Versatile, valid approach if the proximity operator of f_2 is available
- In practice, Douglas-Rachford iterations need warm-up initializations
- Good properties, e.g. monotonical decrease & convergence

[Elvira et al, 2022]

Experimental results

- Synthetic
- Climate
- Migrations
- Food insecurity

1- Synthetic problems

Synthetic problems

- 4 synthetic datasets with H = Id and block-diagonal matrix $\mathbf{A}$
- Diagonal blocks of $\mathbf{A}$ are randomly set as matrices of AR(1) processes

Dataset	$(b_j)_{1\leq j\leq b}$	$(\sigma_{\mathbf{Q}}, \sigma_{\mathbf{R}}, \sigma_{\mathbf{P}})$
A	$(3,3,3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
B	$(3,3,3)$	$(1,1,10^{-4})$
C	$(3,5,5,3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
D	$(3,5,5,3)$	$(1,1,10^{-4})$

- GraphEM [Elvira 2022] - MLEM [Sarkka 2013] - Pairwise & Cond. GC [Luengo 2019]
- Results are averaged on 50 runs

Synthetic problems

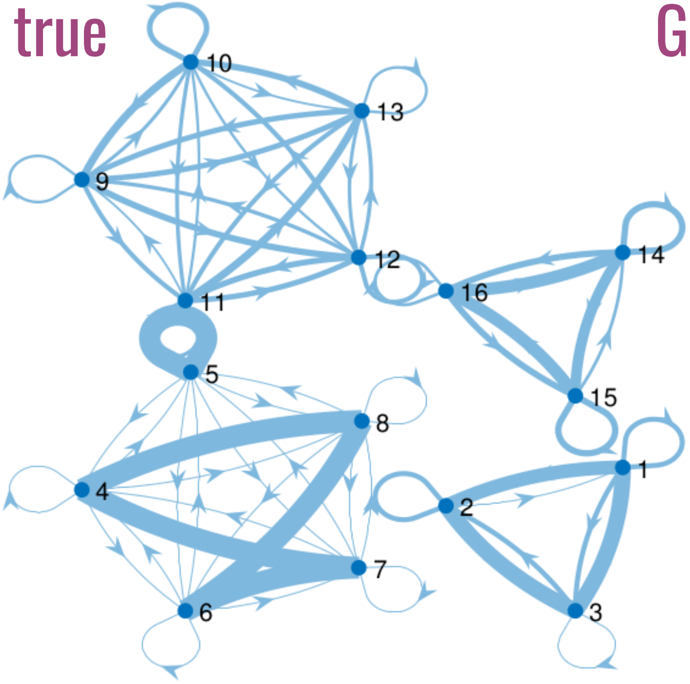
	method	RMSE	accur.	prec.	recall	spec.	F1
A	GraphEM	0.081	0.9104	0.9880	0.7407	0.9952	0.8463
	MLEM	0.149	0.3333	0.3333	1	0	0.5
	PGC	-	0.8765	0.9474	0.6667	0.9815	0.7826
	CGC	-	0.8765	1	0.6293	1	0.7727
B	GraphEM	0.082	0.9113	0.9914	0.7407	0.9967	0.8477
	MLEM	0.148	0.3333	0.3333	1	0	0.5
	PGC	-	0.8889	1	0.6667	1	0.8
	CGC	-	0.8889	1	0.6667	1	0.8
C	GraphEM	0.120	0.9231	0.9401	0.77	0.9785	0.8427
	MLEM	0.238	0.2656	0.2656	1	0	0.4198
	PGC	-	0.9023	0.9778	0.6471	0.9949	0.7788
	CGC	-	0.8555	0.9697	0.4706	0.9949	0.6337
D	GraphEM	0.121	0.9247	0.9601	0.7547	0.9862	0.8421
	MLEM	0.239	0.2656	0.2656	1	0	0.4198
	PGC	-	0.8906	0.9	0.6618	0.9734	0.7627
	CGC	-	0.8477	0.9394	0.4559	0.9894	0.6139

- No RMSE for PGC/CGC as edge-detection methods
- MLEM poor results as no sparsity is encoded
- GraphEM much better, esp. accuracy & F1

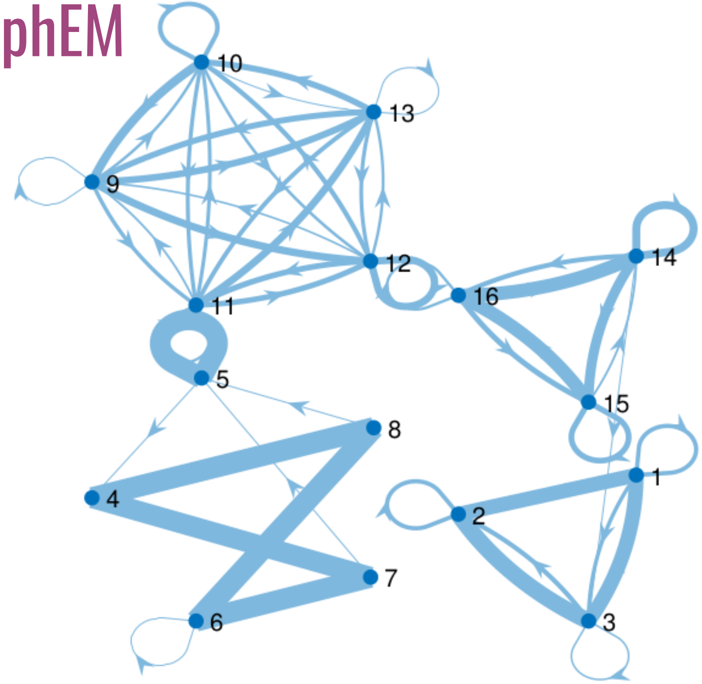
[Elvira et al, 2022]

Synthetic problem “C”

true



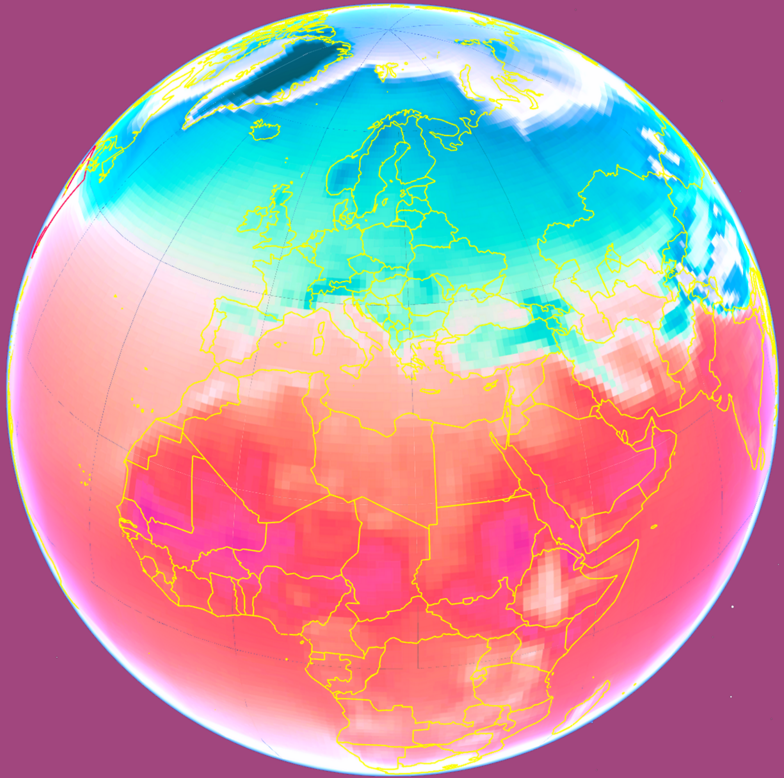
GraphEM

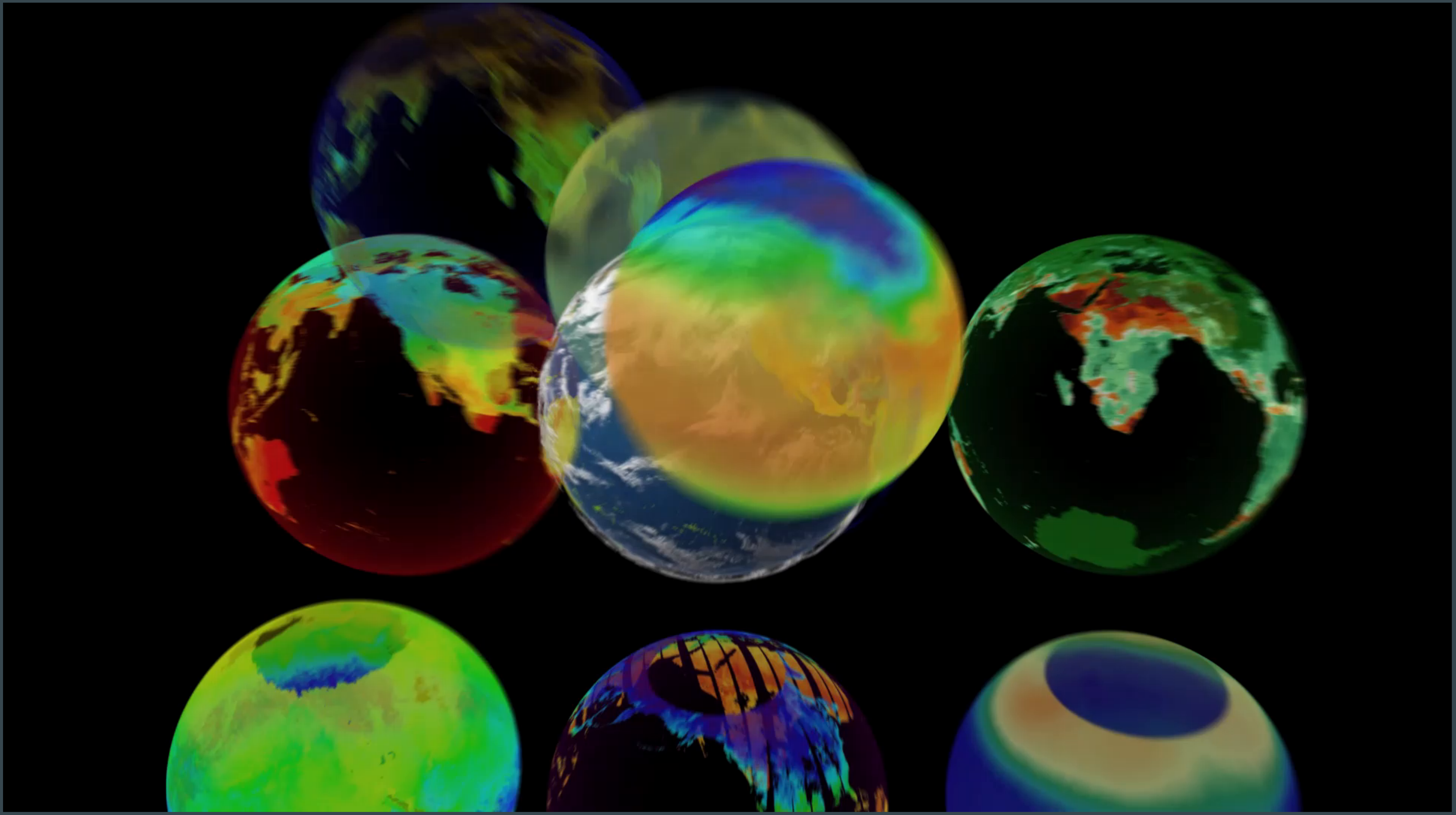


- Both structure of the graph and weight values are well recovered

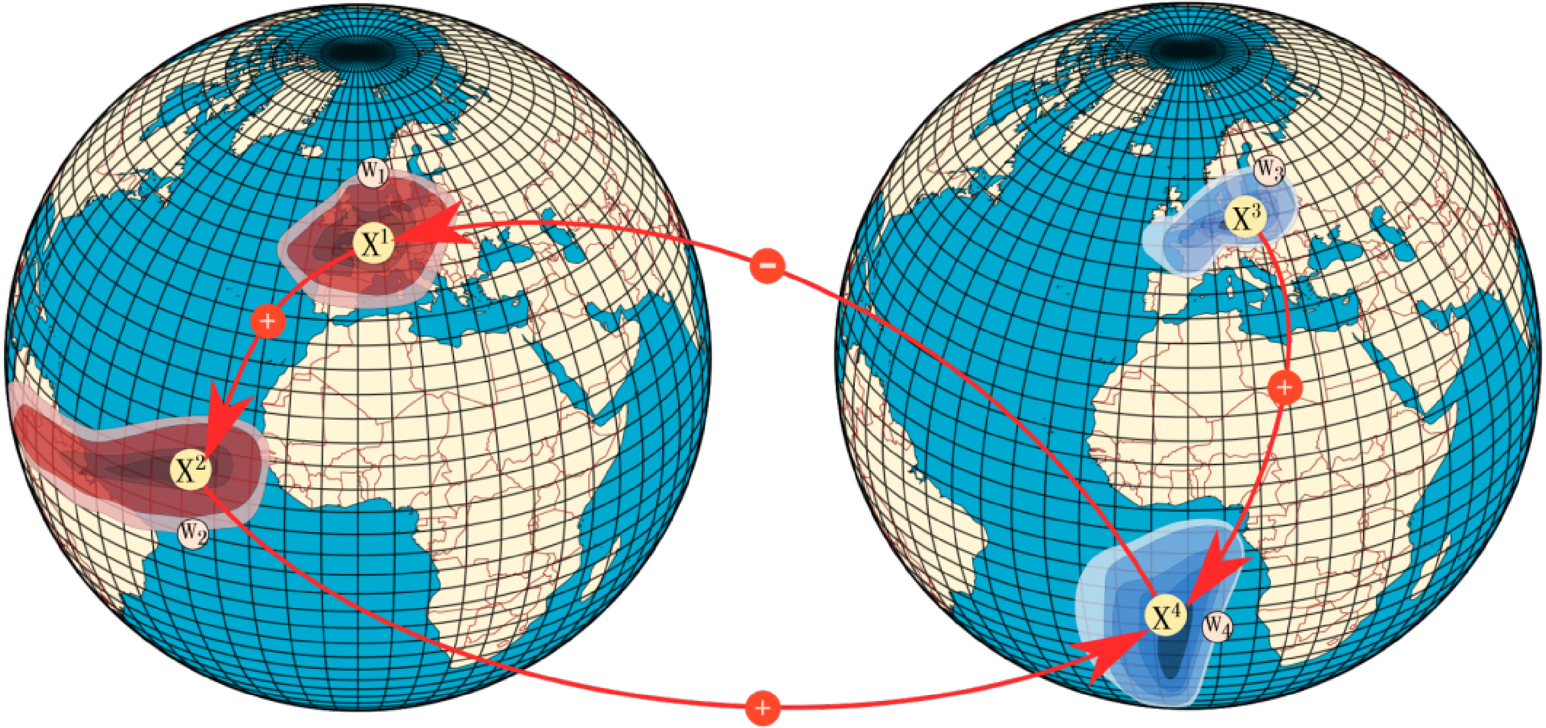
[Elvira et al, 2022]

2- Climate science





Climate teleconnections



[Runge et al, 2020]

Climate data

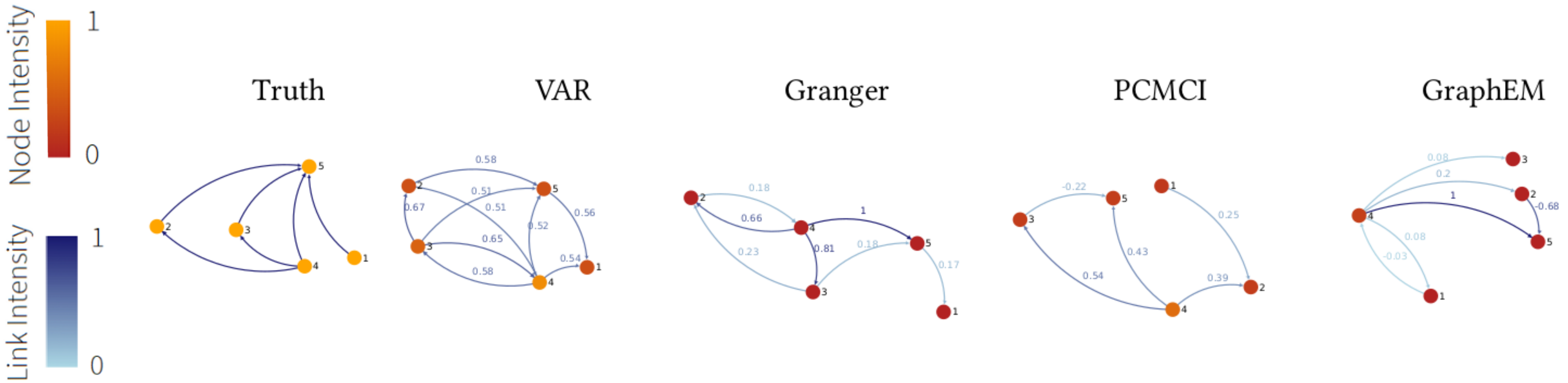
- Synthetic data generation [Runge et al, 2020]
  - Climate model simulations of pre-industrial (stationary) control runs
    - 15 vars: hfls, hfss, huss, rlds, rlus, rlut, ta, tas, tasmax, ...
  - Varimax projected onto 5 PCs
  - VAR modeling & clipped coefficients
  - Averaged time series (5-day resolution) + add noise
- GraphEM [Elvira 2022] - VAR [Sarkka 2013] - GC [Luengo 2019] - PCMCI [Runge, 2019]
- Results are averaged on 100 runs

Climate results

method	best hyperparameters	accur.	prec.	recall	spec.	F1
GraphEM [12]	$\sigma_{\mathbf{R}} = 0.1, \sigma_{\mathbf{P}} = 10^{-4}, \gamma_1 = 50$	**0.72**	0.75	**0.55**	0.86	**0.63**
VAR [32]	$\ell = 8$	0.56	0.50	0.46	0.64	0.48
Granger [14]	$\ell = 8$	0.6	0.57	0.36	0.79	0.44
PCMCI [25]	$\tau_{\max} = 8, \alpha_{\mathrm{PC}} = 0.05$, ParCorr	0.72	**0.83**	0.45	**0.93**	0.59

- GraphEM outperforms VAR and GC in all performance metrics
- GraphEM outperforms PCMCI in recall and overall F1 score

Climate results

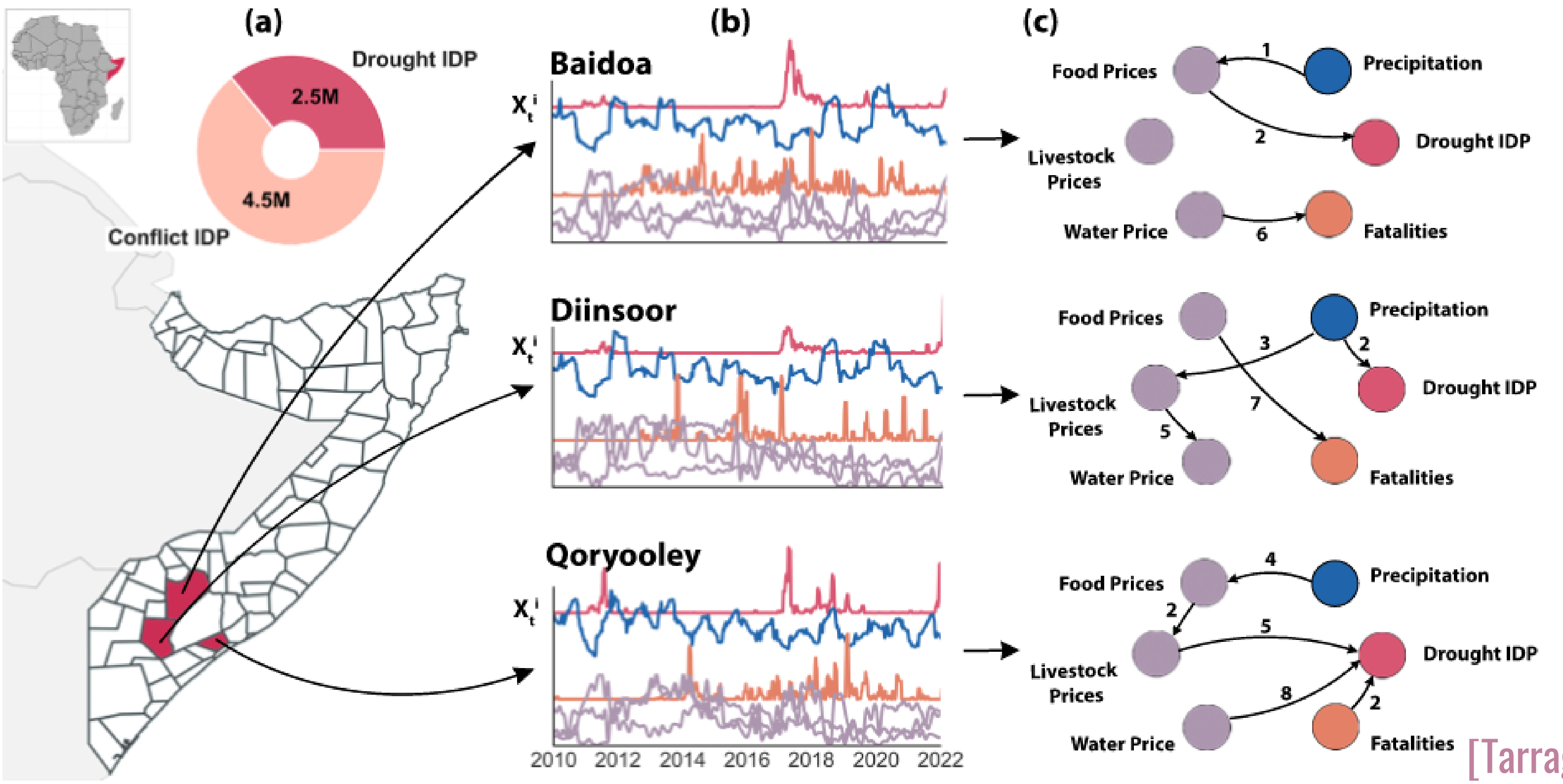


- Good detection of links {2, 4} → 5 unlike PCMCI
- Much sparser (and less convoluted) solution unlike VAR and GC

3- Climate-human interactions

Understanding climate-induced migrations

Fundación BBVA PE

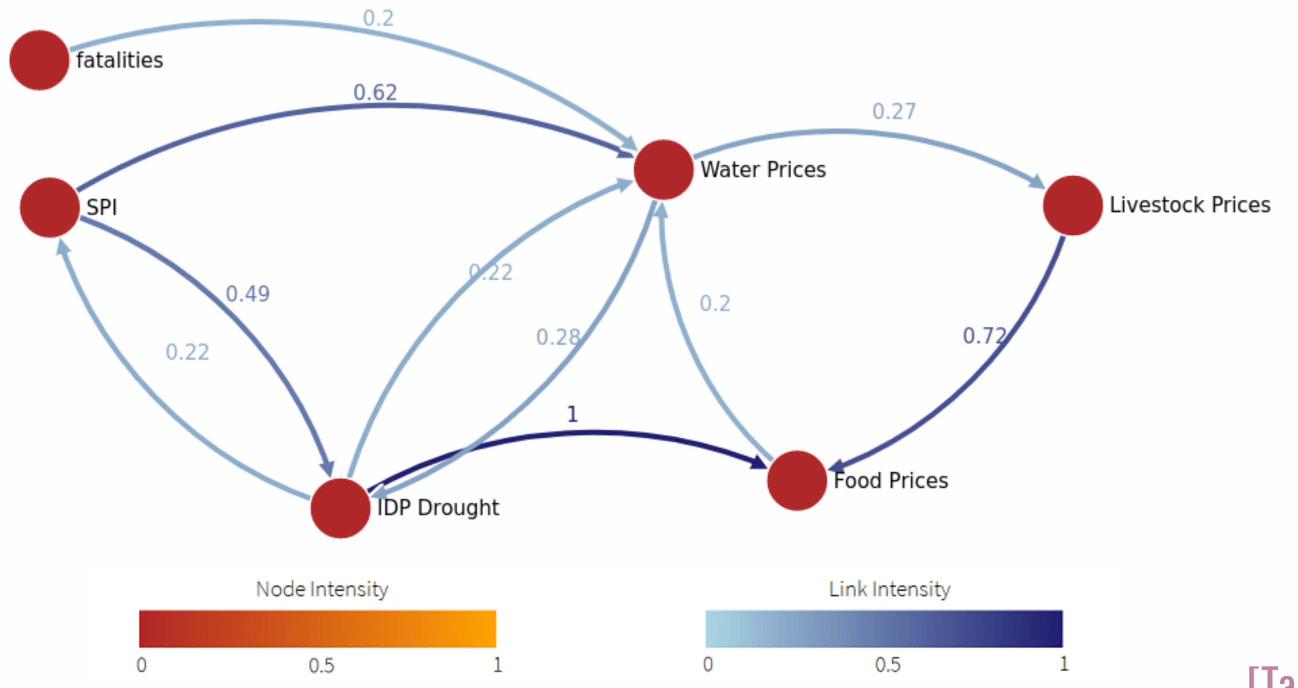


[Tarraga et al, 2022]

Understanding climate-induced migrations

Fundación BBVA PE

Granger causality

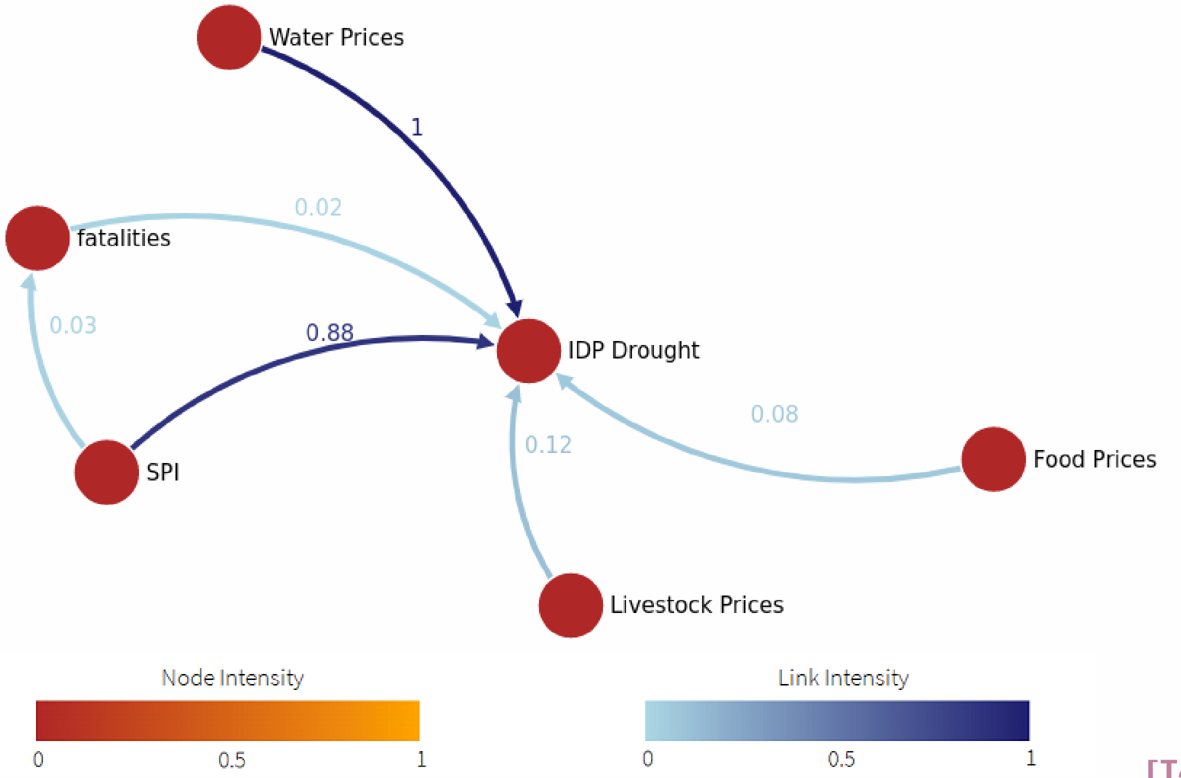


[Tarraga et al, 2022]

Understanding climate-induced migrations

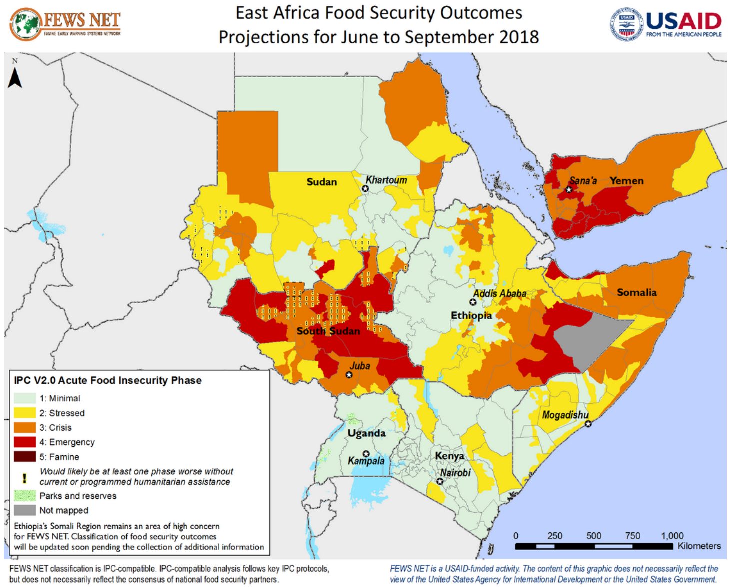
Fundación BBVA PE

GraphEM



[Tarraga et al, 2022]

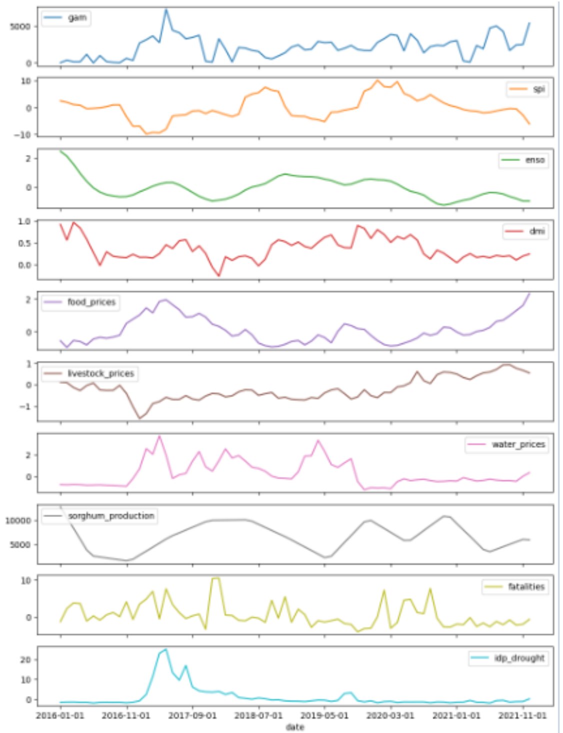
Understanding food insecurity



[Cerdà et al, 2022]

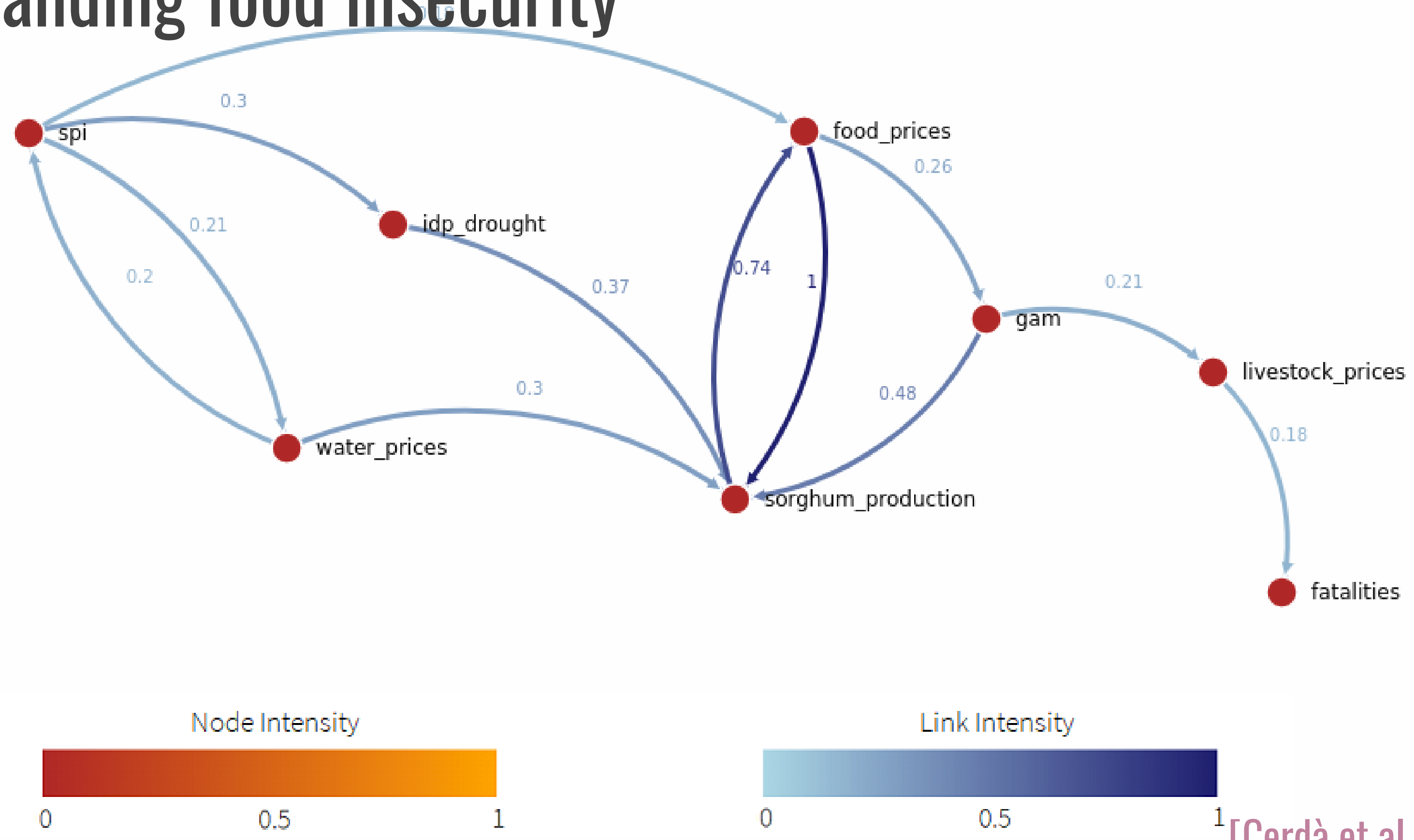
Understanding food insecurity - Data

- Monthly sampled for 37 districts in Somalia, 5 years, 70 points each
- Market/food/livestock/water prices, displaced people, **malnutrition**, fatalities, climate variables, humanitarian aid
- $\mathbf{A}$ constrained with (un)reasonable connections



[Cerdà et al, 2022]

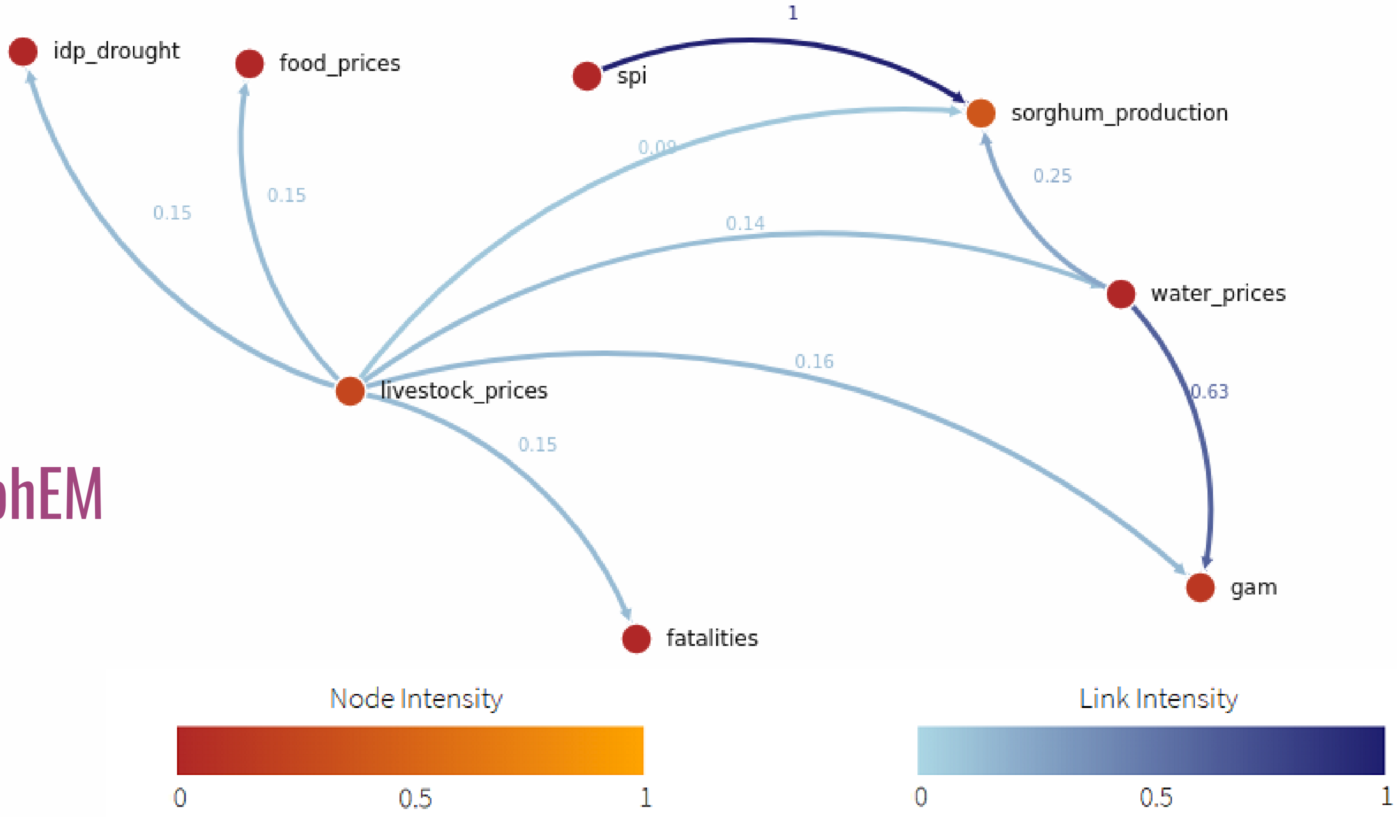
Understanding food insecurity



Granger causality

[Cerdà et al, 2022]

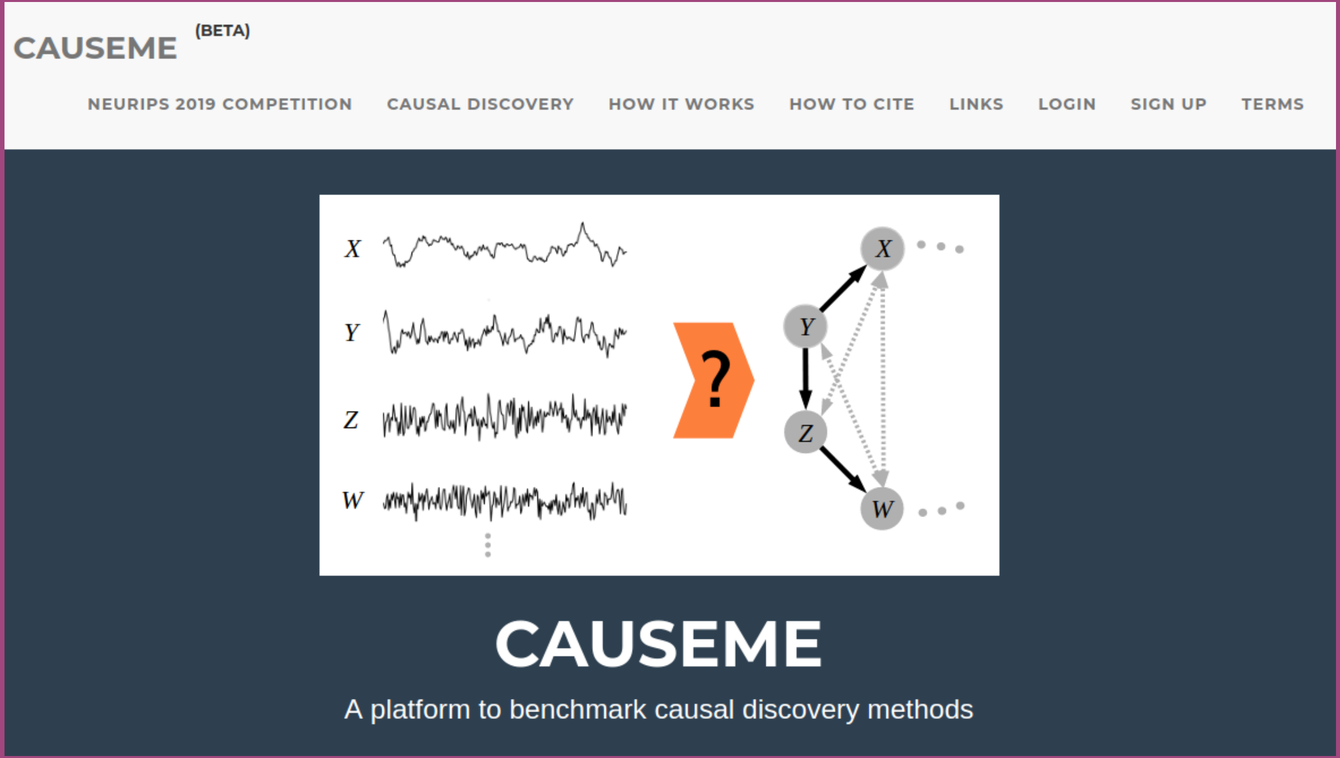
Understanding food insecurity



GraphEM

[Cerdà et al, 2022]

causeme.net



CAUSEME (BETA)
NEURIPS 2019 COMPETITION
CAUSAL DISCOVERY
HOW IT WORKS
HOW TO CITE
LINKS
LOGIN
SIGN UP
TERMS
X
Y
Z
W
?
X
Y
Z
W
CAUSEME
A platform to benchmark causal discovery methods

Conclusions

Conclusions

- **GraphEM**: EM method for inferring the linear hidden state relationships in a linear-Gaussian state-space model
  - **Proximal splitting** w/ convergence guarantees to solve the M-step
  - **LASSO penalization** to model and represent the state entries interactions as a compact and interpretable graph + prior-enforcing
- **Good numerical performance** in synthetic and real problems
- **Use & contribute causeme.net !**
- **Nonlinear extensions** through kernels & particle filtering