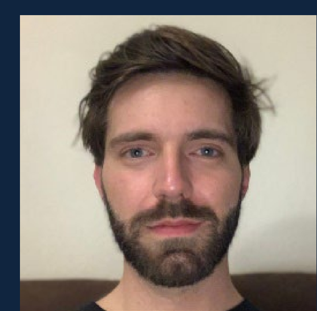
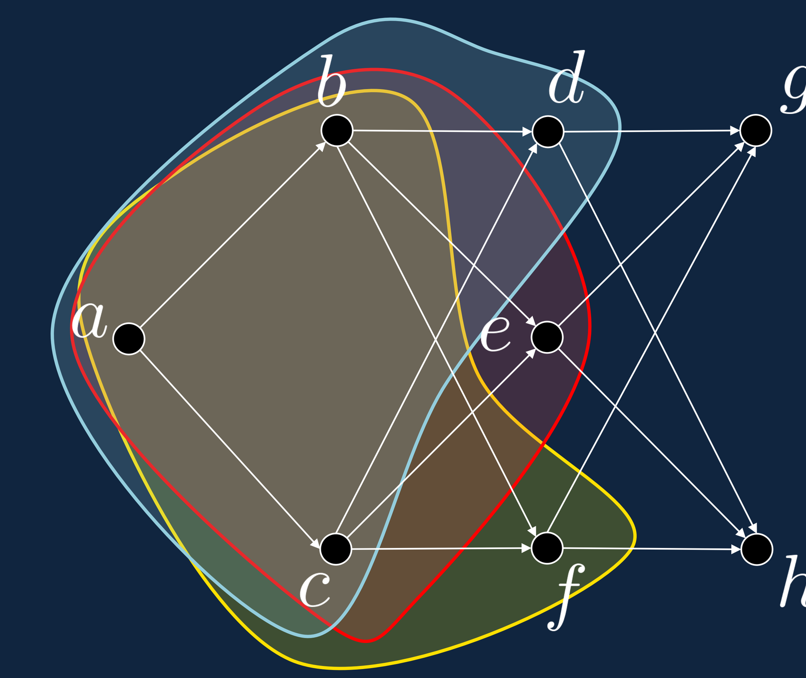
Learning Fourier-Sparse Functions on DAGs

Bastian Seifert, Chris Wendler, Markus Püschel


Department of Computer Science, ETH Zurich

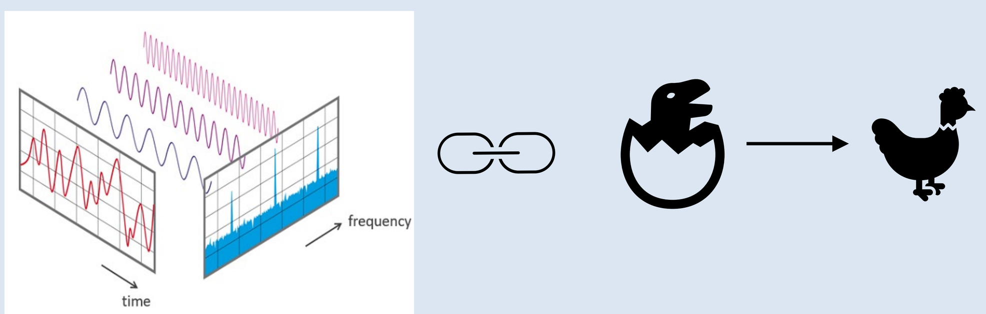


ETH zürich

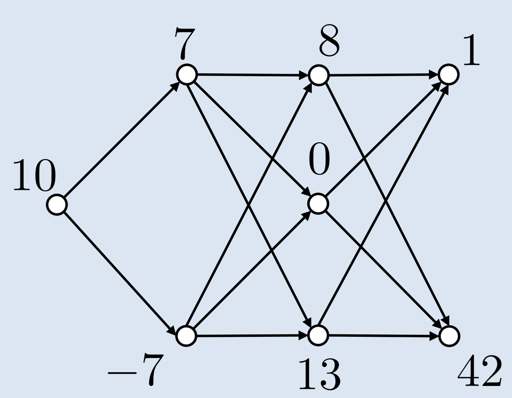
https://acl.inf.ethz.ch/research/ASP/

Goal

Fourier-based learning with **causal** data



Model: Causal data on **DAGs**



Contribution: Novel **Fourier Analysis** on **DAGs** + Prototypical Learning Example

Picture from https://www.nti-audio.com/de/service/wissen/fast-fourier-transformation-fft

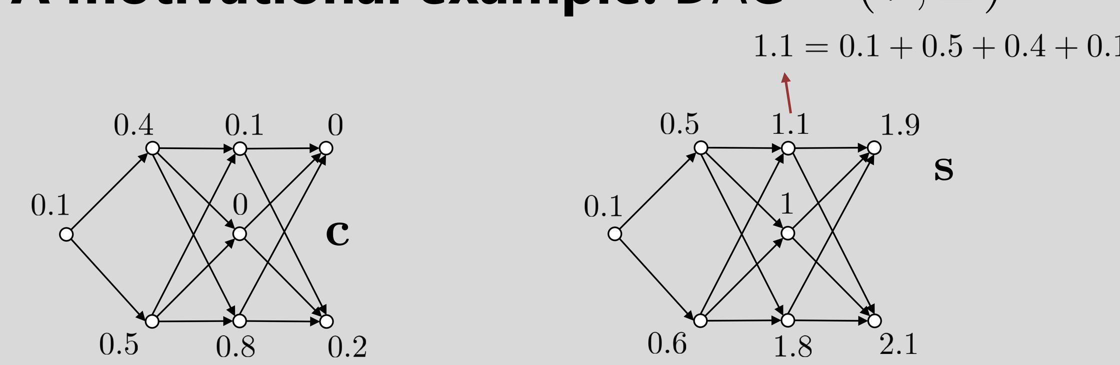
Algebraic Fourier Analysis

Generic Approach (Algebraic Signal Processing, IEEE Trans. Signal Proc., 2008)	**Example:** Discrete Time Signals
Definition of Shift	Time Delay $(s_n)_{n\in\mathbb{Z}} \mapsto (s_{n-1})_{n\in\mathbb{Z}}$
Convolution Linear shift-equivariant mappings	$(H * s)_n = \sum_{k\in\mathbb{Z}} h_n s_{n-k}$
Fourier transform Eigendecomposition of convolution	$\mathcal{F} = \mathrm{DTFT}: \begin{array}{l}(s_n)_{n\in\mathbb{Z}} \mapsto (\hat{s})_{k\in\mathbb{Z}} \\ \hat{s}_k = \sum_{n\in\mathbb{Z}} s_n e^{-ikn}\end{array}$

Key question:
How to define shift on DAGs?

Fourier Analysis on DAGs

A motivational example: DAG $= (V, E)$



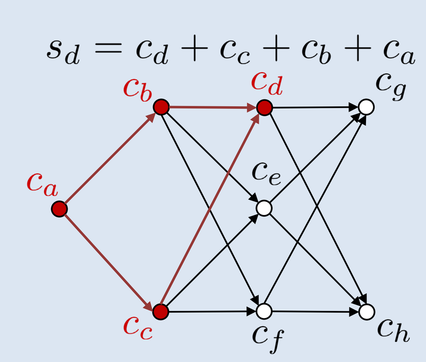
Causes $(c_y)_{y\in V}$: Pollution along river network

Measured signal $(s_x)_{x\in V}$: accumulated pollution from predecessors

Signal: Sum of its causes

$$s_x = \sum_{y \le x} c_y$$

y predecessor of x

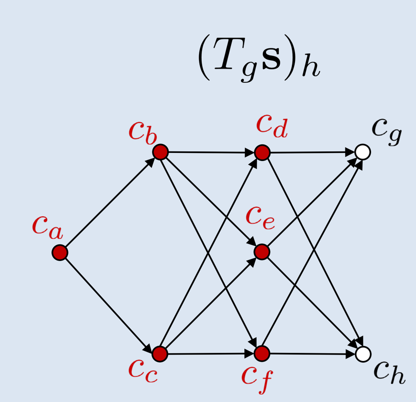


Shift by $q \in V$ **:** Set of common causes

$$(T_q \mathbf{s})_x = \sum_{y\le x \text{ and } y \le q} c_y$$

$$T_q T_p = T_p T_q$$

T_q not invertible



Convolution: See paper, polynomial combinations of shifts

Fourier transform: Eigendecompostion of shifts/convolution

$$\mathcal{F}\colon (s_x)_{x\in V} \mapsto (\hat{s}_y)_{y\in V} = (c_y)_{y\in V}$$

How to calculate? Moebius inversion!

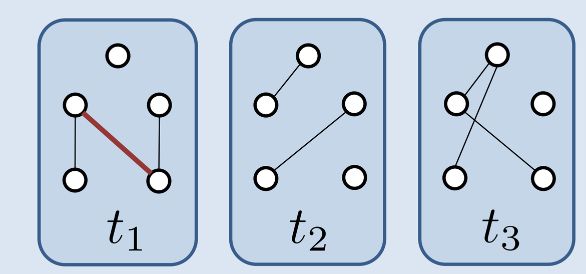
$$\mu(x,x) = 1, \text{ and } \mu(x,y) = -\sum_{x\le z<y} \mu(x,z)$$

Rota, 1964

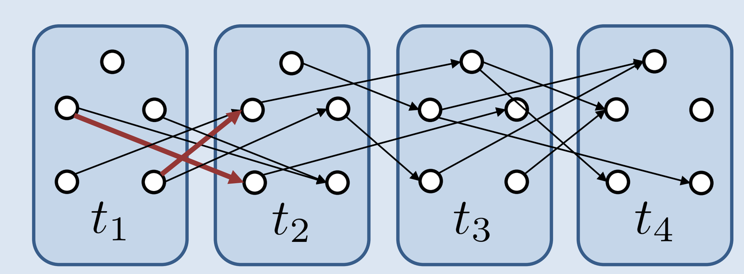
$$\mathcal{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 1 & 0 & 0 \\ -2 & 2 & 2 & -1 & -1 & -1 & 1 & 0 \\ -2 & 2 & 2 & -1 & -1 & -1 & 0 & 1 \end{bmatrix}, \mathcal{F}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Dynamic Network DAGs

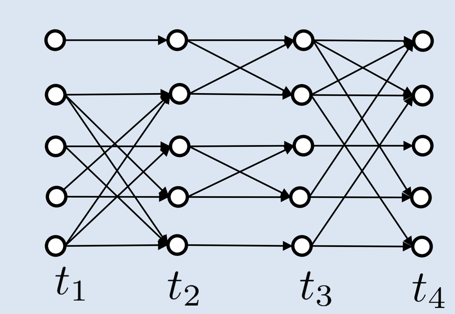
Dynamic Network: A graph where the edges change with time



Dynamic DAG: Nodes of DAG (x,t), edges (x,t) to (y,t+1) if there is edge (x,y) at time t



Final DAG: Add edges (x,t) to (x,t+1) to model self-influence

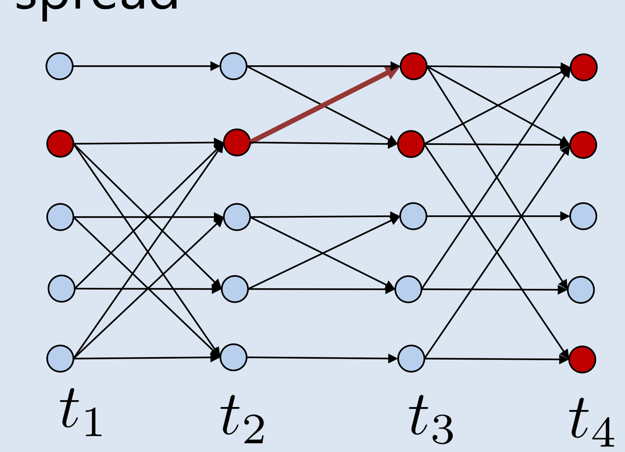


Kim & Anderson, Phys. Rev. E, 2012

Infection Spreading

Haslemere Data Set: 462 participants monitored over 3 days using smartphones

Dynamic DAG with signal: Simulation of infection spread



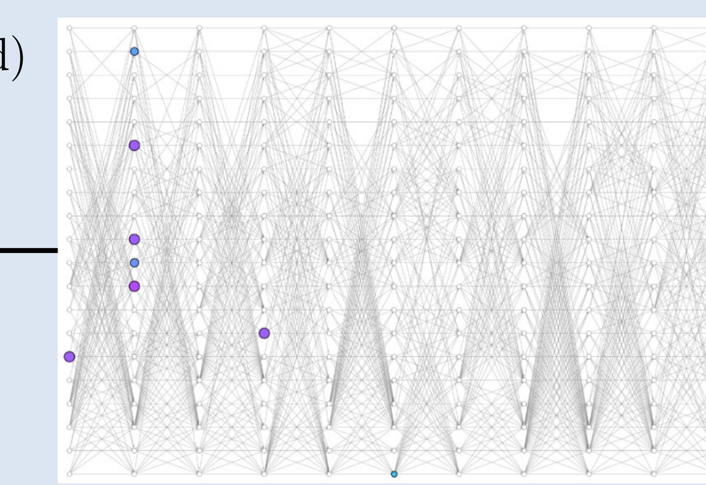
Disease spreads with some probability to persons nearby

Kissler et al., bioRxiv:479154v2, 2018

Fourier-Sparse Learning

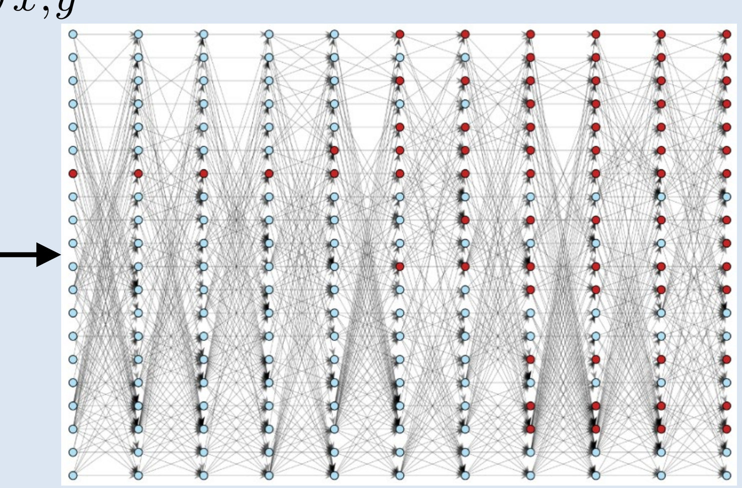
$$\text{prob}(x \text{ is infected}) = \sigma\Big(\sum_{y\in V} \hat{r}_y f_x^y\Big)$$

$\hat{\mathbf{r}}$



$$\mathcal{F}^{-1} = (f_x^y)_{x,y}$$

$\mathbf{s}$



L^1-regularized logistic regression:

$$\min_{\hat{\mathbf{r}}\in\mathbb{R}^{|V|}} -\sum_{i=1}^{n} \big(s_i \log p(x_i) + (1-s_i)\log(1-p(x_i))\big) + \lambda \sum_{y\in V} |\hat{r}_y|$$

Results