# Tensor Graph Regularized Bootstrap for High-Dimensional Bond Yield Reconstructions: Applications in Green Bond Markets

Prof. Dr. Gareth W. Peters

(CStat-RSS, Adv.DSP-RSS, CMath-FIMA, FIOR, FRSS, FIMA, YAS-RSE, SIRM, ISI-Elected Member, IEEE-Senior Fellow)

Professor of Statistics for Risk and Insurance &
Janet and Ian Duncan Endowed Chair in Actuarial Science

University of California Santa Barbara

Dr. Marta Campi
University of Zurich (UZH)
Deep Hearing Lab, ENT & Statistics Departments

December 10, 2025

# Table of contents

# Related Publications



1. Antonian, E. B., Peters, G. W., Chantler, M. J. (2024). Kernel Generalized Least Squares Regression for Network-Structured Data. *Plos One*. https://doi.org/10.1371/journal.pone.0324087
2. Antonian, E. B., Peters, G. W., Chantler, M. J. (2024). Bayesian Reconstruction of Cartesian Product Graph Signals with General Patterns of Missing Data. *Journal of the Franklin Institute*, 361.9 (2024): 106805.
3. **Campi M., Peters G.W., Antonian E.. Multiway Graph Signal Reconstruction: A Statistical Framework for Modeling Green Bond Yield Structures. Preprint.**

Software library for highly efficient large factorised graph computations.

1. Antonian, E. B., Peters, G. W., Chantler, M. J. (2023). PyKronecker: A Python Library for the Efficient Manipulation of Kronecker Products and Related Structures. In *The Journal of Open Source Software, 8(81).*

Ph.D. of Dr. Edward Antonian: Supervised by Prof. Peters (UCSB-HW) and Prof. Chantler (HW)

GitHub for Thesis and Code: https://github.com/nickelnine37

# Table of contents

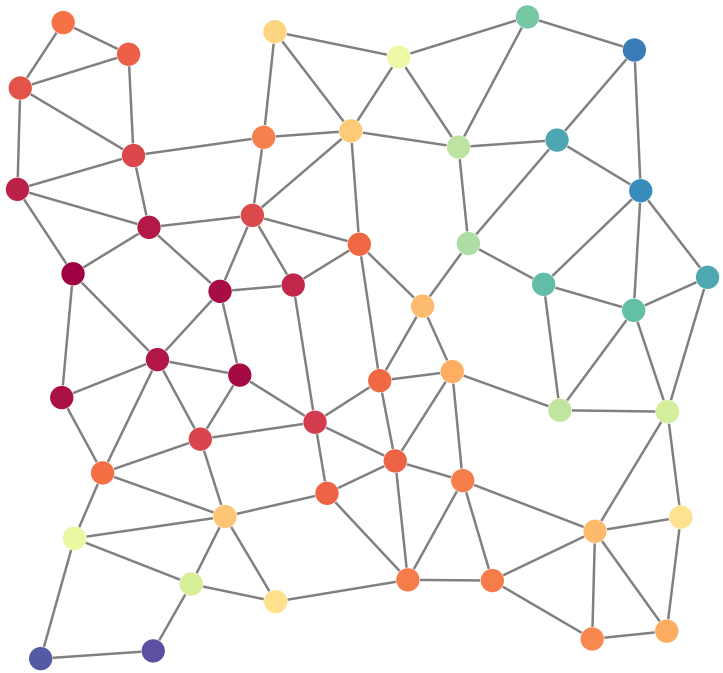
# Graph Signal



A graph signal is a vector $\mathbf{y} \in \mathbb{R}^N$ where each element is understood as being measured on a node of the graph ($N = |\mathcal{V}|$).

An example could be temperature over a network of monitoring stations, or user taste preferences over a social network.

A graphical depiction of a smooth graph signal where the value of signal at each node is displayed in colour (at one time instant).



# Graph Signal

**Formally:** given an undirected weighted graph $G = (\mathcal{V}, \mathcal{E})$, with $|V| = N$, then a graph signal

$$f : \mathcal{V} \to \mathbb{R}$$

is a function defined on the vertices of the graph that maps every vertex $\{v_i\}_{i=1,\ldots,N}$ to a real number $y_i := f(i) \in \mathbb{R}$.

**Intuitively:** we often expect graph signals $(y_i := f(i))$ to be "smooth", ie. neighbouring nodes are likely to be similar.

> Distance $d(y_i, y_j)$ is small when $v_i \sim v_j$ i.e. when graph vertices have a connecting edge association.

Can measure similarity $d(y_i, y_j)$ via squared Total Variation (Dirichlet energy):

$$\text{TV}_2(\mathbf{y}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{A}_{ij}(\mathbf{y}_i - \mathbf{y}_j)^2 = \mathbf{y}^\top(\boldsymbol{D} - \boldsymbol{A})\mathbf{y} = \mathbf{y}^\top\mathbf{L}\mathbf{y}$$

# Laplacian Eigenbasis

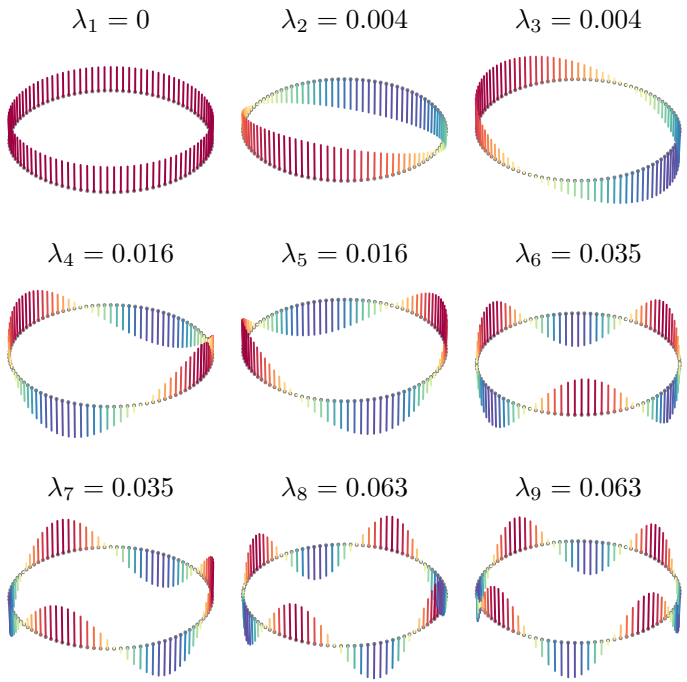
The graph Laplacian $\mathbf{L}$ gives a natural basis in which to analyse the frequency content of a graph signal.

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$$

This provides a direct generalisation of classical Fourier analysis to signals residing on an irregular domain.

$$\begin{aligned}
\mathbf{u}_1 &= \underset{|\mathbf{u}|^2=1}{\operatorname{argmin}} \quad \mathrm{TV}_2(\mathbf{u}) \\
\mathbf{u}_2 &= \underset{|\mathbf{u}|^2=1,\, \perp\mathbf{u}_1}{\operatorname{argmin}} \quad \mathrm{TV}_2(\mathbf{u}) \\
\mathbf{u}_3 &= \underset{|\mathbf{u}|^2=1,\, \perp\mathbf{u}_1,\mathbf{u}_2}{\operatorname{argmin}} \quad \mathrm{TV}_2(\mathbf{u}) \\
\mathbf{u}_4 &= \ldots
\end{aligned}$$

# Laplacian Eigenbasis




The eigenvectors of the graph Laplacian can be understood as sequentially less smooth with respect to the topology of the graph.

The corresponding eigenvalue, referred to as the frequency, gives a value specifying how "rough" each eigenvector is relative to the others, as measured by $TV_2$.

**Undirected graph:** first Laplacian eigenvector will always be constant with an eigenvalue of zero $(\lambda_1 = 0)$.

# Table of contents

# Graph Products

Consider two undirected graphs $\mathcal{G}_A = (\mathcal{V}_A, \mathcal{E}_A)$ and $\mathcal{G}_B = (\mathcal{V}_B, \mathcal{E}_B)$ with vertex sets given by $\mathcal{V}_A = \{a \in \mathbb{N} \,|\, a \leq N_A\}$ and $\mathcal{V}_B = \{b \in \mathbb{N} \,|\, b \leq N_B\}$ respectively.

**Graph Product:** A new graph $\mathcal{G}$ can be constructed by taking the product between factor graphs $\mathcal{G}_A$ and $\mathcal{G}_B$.

$$\mathcal{G} = \mathcal{G}_A \diamond \mathcal{G}_B = (\mathcal{V}, \mathcal{E})$$

**Definitions of graph product $\diamond$ :** $\mathcal{G}$ has vertex set $\mathcal{V}$ from factor graphs

$$\mathcal{V} = \mathcal{V}_A \times \mathcal{V}_B = \{(a, b) \in \mathbb{N}^2 \,|\, a \leq N_A \text{ and } b \leq N_B\}$$

Typically, vertices are arranged in lexicographic order: i.e. $(a, b) \leq (a', b')$ iff $a < a'$ or $(a = a'$ and $b \leq b')$.

Then a specific graph product is defined by a consistent rule for constructing $\mathcal{E}$ from the factor edge sets $\mathcal{E}_A$ and $\mathcal{E}_B$.

One can define $2^8 = 256$ possible graph product rules.

# Graph Products

Adjacency & Laplacians of the product graph in terms of factor graph matrices:

| | Adjacency matrix | Laplacian |
|---|---|---|
| Cartesian | $\mathbf{A}_A \oplus \mathbf{A}_B$ | $\mathbf{L}_A \oplus \mathbf{L}_B$ |
| Direct | $\mathbf{A}_A \otimes \mathbf{A}_B$ | $\mathbf{D}_A \otimes \mathbf{L}_B + \mathbf{L}_A \otimes \mathbf{D}_B - \mathbf{L}_A \otimes \mathbf{L}_B$ |
| Strong | $\mathbf{A}_A \otimes \mathbf{A}_B + \mathbf{A}_A \oplus \mathbf{A}_B$ | $\mathbf{D}_A \otimes \mathbf{L}_B + \mathbf{L}_A \otimes \mathbf{D}_B - \mathbf{L}_A \otimes \mathbf{L}_B + \mathbf{L}_A \oplus \mathbf{L}_B$ |
| Lexicographic | $\mathbf{I}_A \otimes \mathbf{A}_B + \mathbf{A}_A \otimes \mathbf{O}_A$ | $\mathbf{I}_A \otimes \mathbf{L}_B + \mathbf{L}_A \otimes \mathbf{O}_B + \mathbf{D}_A \otimes (N_B \mathbf{I}_B - \mathbf{O}_B)$ |

$\mathbf{D}_A$ and $\mathbf{D}_B$ are the diagonal degree matrices, i.e $\mathbf{D}_A = \operatorname{diag} \mathbf{A}_A \mathbf{1}$. $\mathbf{I}_A$ and $\mathbf{O}_A$ are the $(N_A \times N_A)$ identity matrix and matrix of ones respectively.

# Graph Products



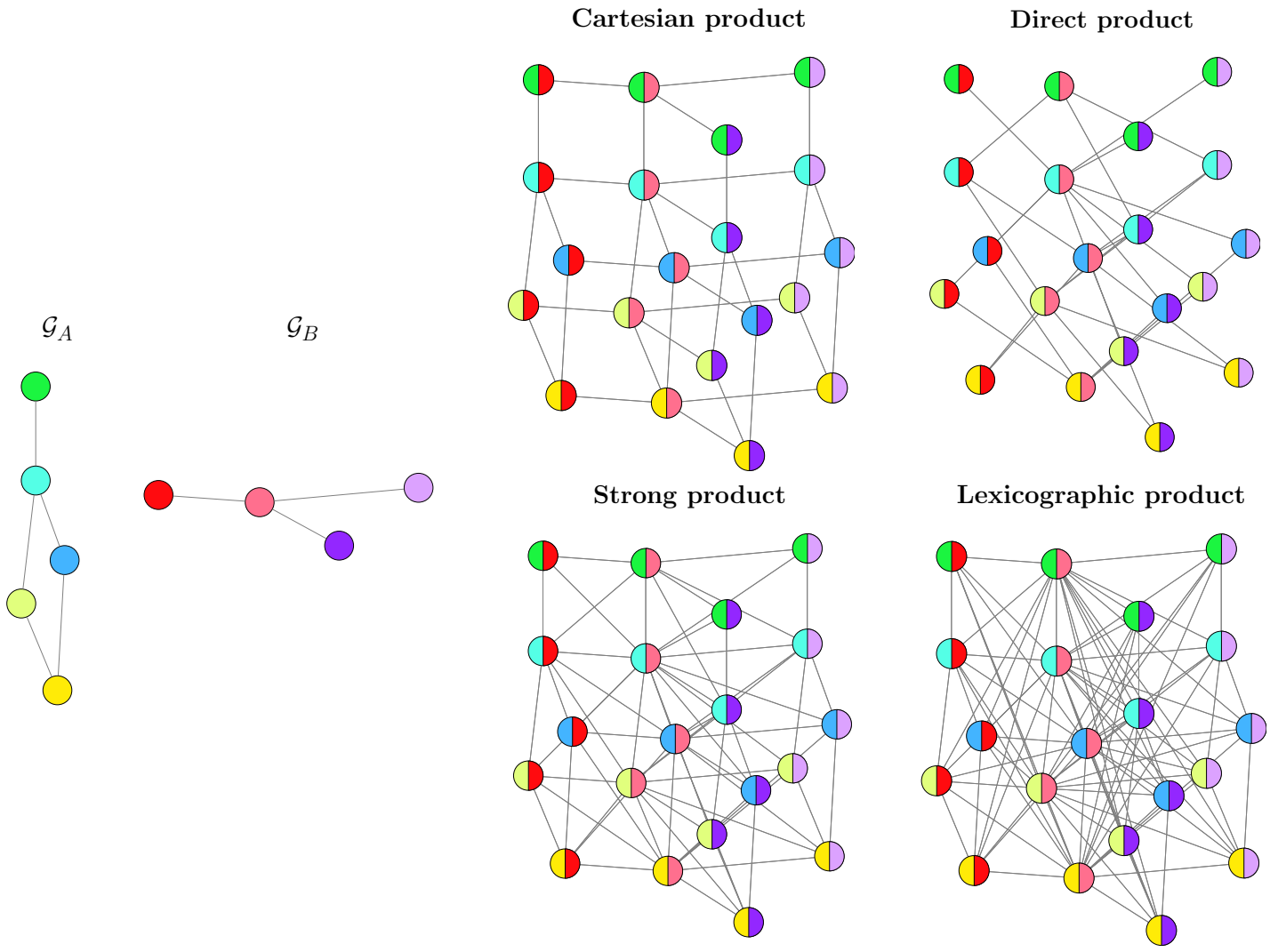



Figure: Depiction of the four standard graph products

Laplacian of a Cartesian product graph satisfies:

$$\begin{aligned}\mathbf{L} &= \mathbf{L}_A \oplus \mathbf{L}_B \\ &= \mathbf{L}_A \otimes \mathbf{I}_B + \mathbf{I}_A \otimes \mathbf{L}_B\end{aligned}$$

Can perform eigen decomposition one each graph product component

$$\begin{aligned}\mathbf{L} &= (\mathbf{U}_A \otimes \mathbf{U}_B)(\mathbf{\Lambda}_A \oplus \mathbf{\Lambda}_B)(\mathbf{U}_A^\top \otimes \mathbf{U}_B^\top) \\ &= \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top\end{aligned}$$

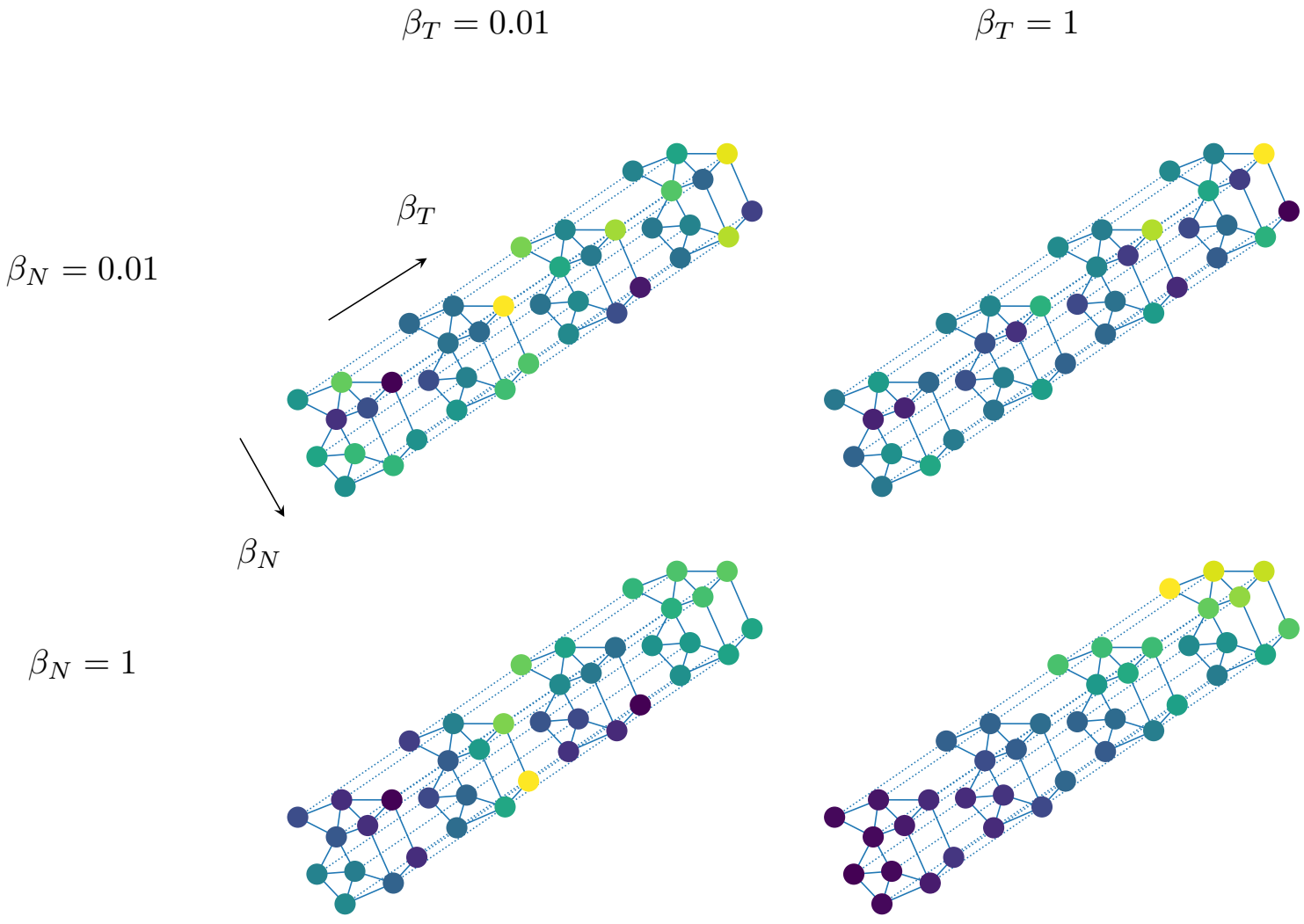
Can develop a definition of filtering in two dimensions.

$$\begin{aligned}\mathbf{H} &= \left(\mathbf{U}_A \otimes \mathbf{U}_B\right) g\left(\mathbf{\Lambda}_A \oplus \mathbf{\Lambda}_B\right) \left(\mathbf{U}_A^\top \otimes \mathbf{U}_B^\top\right) \\ &= \left(\mathbf{U}_A \otimes \mathbf{U}_B\right) \operatorname{diag}\operatorname{vec}\left(\mathbf{G}\right) \left(\mathbf{U}_A^\top \otimes \mathbf{U}_B^\top\right) \\ &= \mathbf{U}\mathbf{D}_{\mathbf{G}}\mathbf{U}^\top\end{aligned}$$

The matrix $\mathbf{G} \in \mathbb{R}^{N_B \times N_A}$ holds the value of the filter function applied to the sum of pairs of eigenvalues.

$$\mathbf{G}_{ba} = g\left(\lambda_a^{(A)}, \lambda_b^{(B)}; \beta_a, \beta_b\right)$$

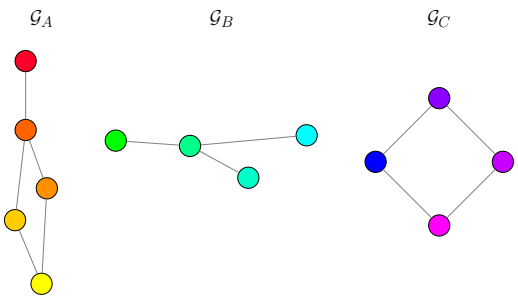
# Cartesian Product Graph Filters

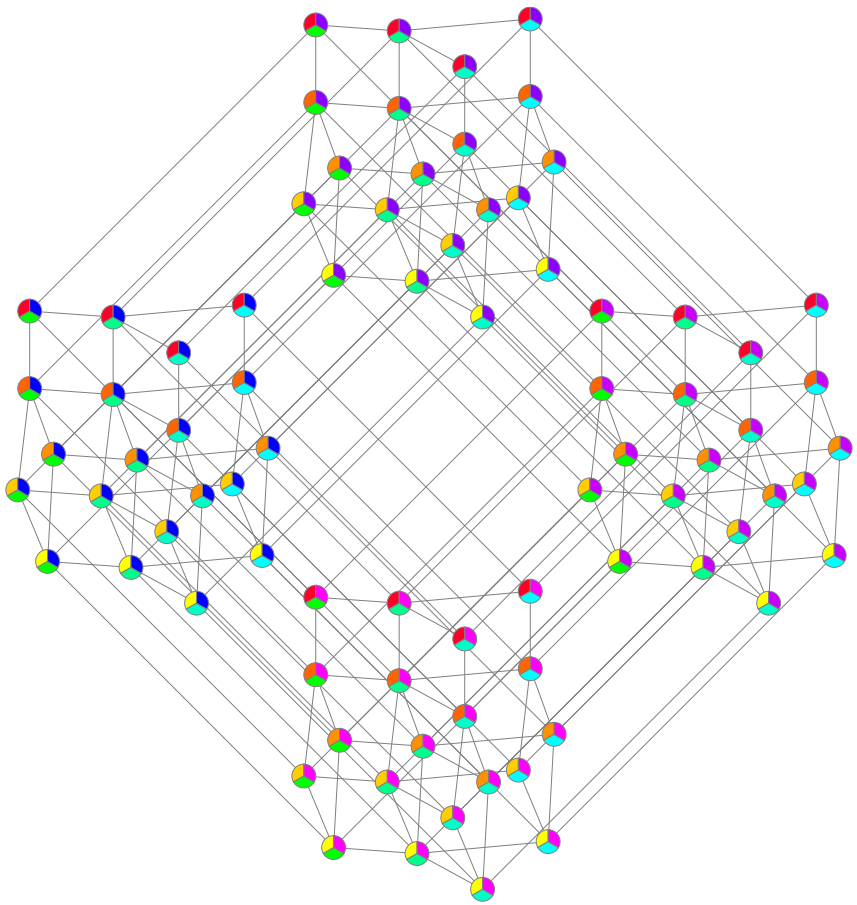


# From Matrices to Tensors

We can easily extend the concept of a two-dimensional product to three or more dimensions.

Here, signals become tensors, or multidimensional arrays, and graphs can become extremely large.



# Spectral decomposition

Consider a tensor graph signal $\mathcal{Y} \in \mathbb{R}^{N_1 \times N_2 \times \ldots \times N_d}$ represented in its multi-dimensional array form.

$$\mathbf{y} = \text{vec}_{\text{RM}}(\mathcal{Y}) \quad \Longleftrightarrow \quad \mathcal{Y} = \text{ten}_{\text{RM}}(\mathbf{y})$$

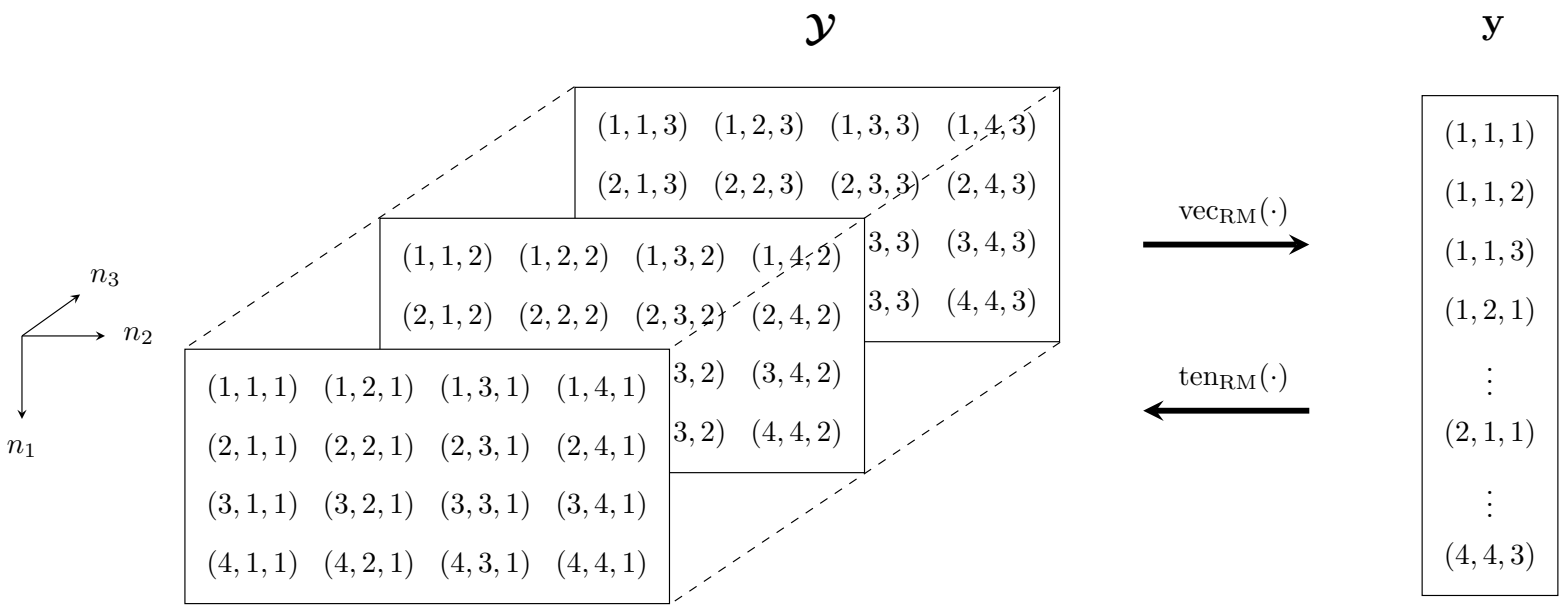



Figure: Depiction of the process of converting an order-3 tensor between its multidimensional array and vector form in **row-major** (RM) order. Note that the elements in the vectorised signal are lexicographically ordered.

# Spectral decomposition

Define the Graph Fourier Transform (GFT) and its corresponding inverse (IGFT) of this signal as follows:

$$\text{GFT}(\mathcal{Y}) = \text{ten}_{\text{RM}}\left(\mathbf{U}^\top \mathbf{y}\right) = \text{ten}_{\text{RM}}\left(\left(\bigotimes_{i=1}^{d} \mathbf{U}^{(i)}\right)^\top \text{vec}_{\text{RM}}\left(\mathcal{Y}\right)\right)$$

$$\text{IGFT}(\mathcal{Y}) = \text{ten}_{\text{RM}}\left(\mathbf{U}\mathbf{y}\right) \quad = \text{ten}_{\text{RM}}\left(\left(\bigotimes_{i=1}^{d} \mathbf{U}^{(i)}\right) \text{vec}_{\text{RM}}\left(\mathcal{Y}\right)\right)$$

**Action of a general spectral operator:** first apply GFT to signal $\mathcal{Y}$ & then apply scaling function to each spectral component. Next transform back into the vertex domain via the IGFT.

$$\mathcal{Y}' = \text{IGFT}\left(\mathcal{G} \circ \text{GFT}(\mathcal{Y})\right)$$

Matrix operator generating this transformation is:

$$\mathbf{H} = \mathbf{U}\, \text{diag}\, \text{vec}_{\text{RM}}\left(\mathcal{G}\right)\mathbf{U}^\top$$

# Table of contents

# Graph Signal Reconstruction Objectives

**Goal of Graph Signal Reconstruction (GSR)**: is to estimate the value of a partially observed graph signal at nodes where no data was collected.

GSR on a Cartesian product graph where available data is partially observed signal $\mathbf{Y} \in \mathbb{R}^{N \times T}$ where only an arbitrary subset $\mathcal{S} = \{(n_1, t_1), (n_2, t_2), \dots\}$ of the signal elements were recorded.

All other missing elements of $\mathbf{Y}$ are set to zero.

Binary Sensing Matrix $\mathbf{S} \in \{0,1\}^{N \times T}$ tracks missing elements of $\mathbf{Y}$:

$$\mathbf{S}_{nt} = \begin{cases} 1 & \text{if } (n,t) \in \mathcal{S} \\ 0 & \text{otherwise.} \end{cases}$$

Data input to GSR problem is as follows:

$$\text{input data} = \left\{ \mathbf{Y} \in \mathbb{R}^{N \times T}, \ \mathbf{S} \in \{0,1\}^{N \times T}, \ \mathbf{A} \in \mathbb{R}^{NT \times NT} \right\}$$

# Graph Signal Reconstruction Objectives

Assume $\mathbf{Y}$ is a noisy partial observation of underlying signal $\mathbf{F} \in \mathbb{R}^{N \times T}$, assumed to be smooth w.r.t. the graph topology:

$$\mathbf{Y} = \mathbf{S} \odot \left( \mathbf{F} + \mathbf{E} \right)$$

Matrix $\mathbf{E}$ represents the model error assumed an independent normal distribution with unit variance.

Distribution of $\mathbf{Y}$ given latent graph signal $\mathbf{F}$ is

$$\text{vec}\left(\mathbf{Y}\right) \mid \mathbf{F} \sim \mathcal{N}\Big( \text{vec}\left(\mathbf{S} \odot \mathbf{F}\right), \ \text{diag}\,\text{vec}\left(\mathbf{S}\right) \Big)$$

Covariance matrix $\text{diag}\,\text{vec}\left(\mathbf{S}\right)$ is semi-positive definite by construction and reflects constraint that some elements of $\mathbf{Y}$ are zero with probability 1.

Note: we also extended these methods to exponential family models & categorical and ordinal data settings on graphs!

# Graph Signal Reconstruction Objectives

Bayesian estimation sets a prior on the latent random graph signal $\mathbf{F}$ which is specified as smooth with respect to the topology of the graph:

w.l.o.g. assume the prior mean for $\mathbf{F}$ is zero across all elements.

$$\text{vec}\,(\mathbf{F}) \sim \mathcal{N}\left(\mathbf{0},\, \gamma^{-1}\mathbf{H}^2\right)$$

Given observations $\mathbf{Y}$ then Bayes' rule gives posterior over $\mathbf{F}$:

$$\pi\left(\text{vec}\,(\mathbf{F}) \mid \mathbf{Y}\right) = \frac{\pi\left(\text{vec}\,(\mathbf{Y}) \mid \mathbf{F}\right)\pi(\mathbf{F})}{\pi(\mathbf{Y})}.$$

In the Gaussian-Gaussian model this produces the posterior mean:

$$\widehat{\text{vec}\,(\mathbf{F})} = \mathbb{E}\left[\text{vec}\,(\mathbf{F}) \mid \mathbf{Y}\right] = \left(\text{diag}\,\text{vec}\,(\mathbf{S}) + \gamma\mathbf{H}^{-2}\right)^{-1}\text{vec}\,(\mathbf{Y})$$

# Graph Signal Reconstruction Solvers

**Calculating:** $\left(\text{diag vec}\,(\mathbf{S}) + \gamma \mathbf{H}^{-2}\right)^{-1}$

Coefficient matrix defining the system is size $NT \times NT$
$\Rightarrow$ direct methods e.g. Gaussian elimination infeasible
$\Rightarrow$ direct computational cost of inversion $O\left(N^3 T^3\right)$ arithmetic operations.

Alternative solvers include: stationary iterative methods; Krylov methods; or multigrid methods.

In the following work:

> Antonian, E. B., Peters, G. W., Chantler, M. J. (2024). Bayesian Reconstruction of Cartesian Product Graph Signals with General Patterns of Missing Data. *Journal of the Franklin Institute*, 361.9 (2024): 106805.

we propose a stationary iterative method and a Krylov method and compare their relative behaviour.

Convergence behaviour of each is derived theoretically and verified numerically.

In both cases, we show that each step of the iterative process can be completed in $O(N^2 T + NT^2)$ operations, making a solution feasible for relatively large graph problems.

# Table of contents

# Testing for Utilty of GSR

When is Graph Signal Reconstruction effective?

i.e. Is the partially observed signal exhibiting smoothness with respect to the available graph?

Illustrated for single graph setting...

1. Downsample partially observed signal $\mathbf{y} \in \mathbb{R}^N$ into a vector $\widetilde{\mathbf{y}} \in \mathbb{R}^{\widetilde{N}}$ s.t. only observed elements kept.
2. Adjust adjacency matrix to produce $\widetilde{\mathbf{A}} \in \mathbb{R}^{\widetilde{N} \times \widetilde{N}}$, which represents the connections between the available nodes
3. Create corresponding Laplacian $\widetilde{\mathbf{L}} \in \mathbb{R}^{\widetilde{N} \times \widetilde{N}}$.
4. Pre-whiten $\widetilde{\mathbf{y}}$ for mean of zero and unit variance.
5. Compute the total square variation of $\widetilde{\mathbf{y}}$ as

$$\mathrm{TV}_2(\widetilde{\mathbf{y}}) = \widetilde{\mathbf{y}}^\top \widetilde{\mathbf{L}} \widetilde{\mathbf{y}} = \sum_{n=1}^{\widetilde{N}} \widetilde{\lambda}_n \left(\widetilde{\mathbf{U}}^\top \widetilde{\mathbf{y}}\right)_n^2$$

which can be used for a test.

We define the following hypothesis test:

**Null hypothesis** $\mathcal{H}_0$: The vector $\widetilde{\mathbf{y}}$ is spherically distributed in the Fourier domain.

**Alternative hypothesis** $\mathcal{H}_1$: $\widetilde{\mathbf{y}}$ is biased towards low-frequency Fourier components.

**Under $\mathcal{H}_0$:** $(\widetilde{\mathbf{U}}^\top \widetilde{\mathbf{y}})_n \sim \mathcal{N}(0,1)$ and $\text{Cov}\left((\widetilde{\mathbf{U}}^\top \widetilde{\mathbf{y}})_i, (\widetilde{\mathbf{U}}^\top \widetilde{\mathbf{y}})_j\right) = \delta_{ij}$.

$\Rightarrow \text{TV}_2(\widetilde{\mathbf{y}})$ is the sum of $\widetilde{N}$ independent gamma random variables, where

$$\widetilde{\lambda}_n (\widetilde{\mathbf{U}}^\top \widetilde{\mathbf{y}})_n^2 \sim \Gamma(k = \frac{1}{2}, \theta = 2\widetilde{\lambda}_n)$$

Hence:

$$\text{E}\left[\widetilde{\lambda}_n (\widetilde{\mathbf{U}}^\top \widetilde{\mathbf{y}})_n^2\right] = \widetilde{\lambda}_n, \quad \text{Var}\left[\widetilde{\lambda}_n (\widetilde{\mathbf{U}}^\top \widetilde{\mathbf{y}})_n^2\right] = 2\widetilde{\lambda}_n^2$$

so by Lyapunov central limit theorem, under the null hypothesis, $\text{TV}_2(\widetilde{\mathbf{y}})$ will be approximately distributed as $(\widetilde{N} \to \infty)$ by

$$\text{TV}_2(\widetilde{\mathbf{y}})|\mathcal{H}_0 \sim \mathcal{N}\left(\sum_{n=1}^{\widetilde{N}} \widetilde{\lambda}_n, 2\sum_{n=1}^{\widetilde{N}} \widetilde{\lambda}_n^2\right) = \mathcal{N}\left(\text{Tr}(\widetilde{\mathbf{L}}), 2\,\text{Tr}(\widetilde{\mathbf{L}}^2)\right)$$

# Table of contents

# Green Bond Markets: The Challenge



**Problem:** Pricing securities across multiple dimensions when observations are extremely sparse.

Green bond yields depend on **multiple characteristics simultaneously**:

Environmental purpose (solar, wind, water, transportation...)
Credit rating (AAA, AA+, AA, ... BBB+)
Maturity (1 year to 50 years)
Tax treatment (federal/state exempt, AMT, taxable...)

**Extreme sparsity:** 99.27% of characteristic combinations have no observable prices

**Market size:** \$1.5 trillion globally — accurate pricing matters!

$\Rightarrow$ **Neither traditional econometrics nor standard ML handles this well.**

# Data: US Municipal Green Bonds

**Dataset:** Weekly yield-to-maturity from Bloomberg

- **829 distinct bonds** with sufficient trading history
- **429 weekly observations** (June 2018 – March 2023)
- Period includes: COVID-19 disruption, Fed rate hikes

**Five-dimensional tensor structure:**

| Dimension | Size | Graph Structure |
|---|---|---|
| Time | 429 | Chain (weekly) |
| Environmental purpose | 161 | Hierarchical tree (IEA/Bloomberg) |
| Credit rating | 8 | Ordered chain (AAA $\rightarrow$ BBB+) |
| Maturity | 8 | Chain (0-1y $\rightarrow$ 40-50y) |
| Tax status | 7 | Custom graph (legal provisions) |

**Theoretical tensor size:** $429 \times 161 \times 8 \times 8 \times 7 \approx$ **31 million** values

# Why Graph Structure?



**Credit Ratings:** Ordered chain

- AAA $\leftrightarrow$ AA+ $\leftrightarrow$ AA $\leftrightarrow$ ... $\leftrightarrow$ BBB+
- Adjacent ratings = similar default risk

**Maturity:** Ordered chain

- 0-1y $\leftrightarrow$ 1-2y $\leftrightarrow$ ... $\leftrightarrow$ 40-50y
- Term structure smoothness

**Environmental Purpose:** Hierarchy

- Solar $\sim$ Wind (both renewable)
- Renewable $\neq$ Water infrastructure

**Tax Status:** Custom graph

- Edges connect statuses sharing legal provisions
- Similar tax $\Rightarrow$ similar investor base

**Key insight:** Encode domain knowledge through graphs, don't learn from sparse data!

# Validating Graph Structure: Isotropy Test



**Key question:** Is the assumed graph structure actually appropriate?

**Isotropy test:** Compare observed smoothness to random signals

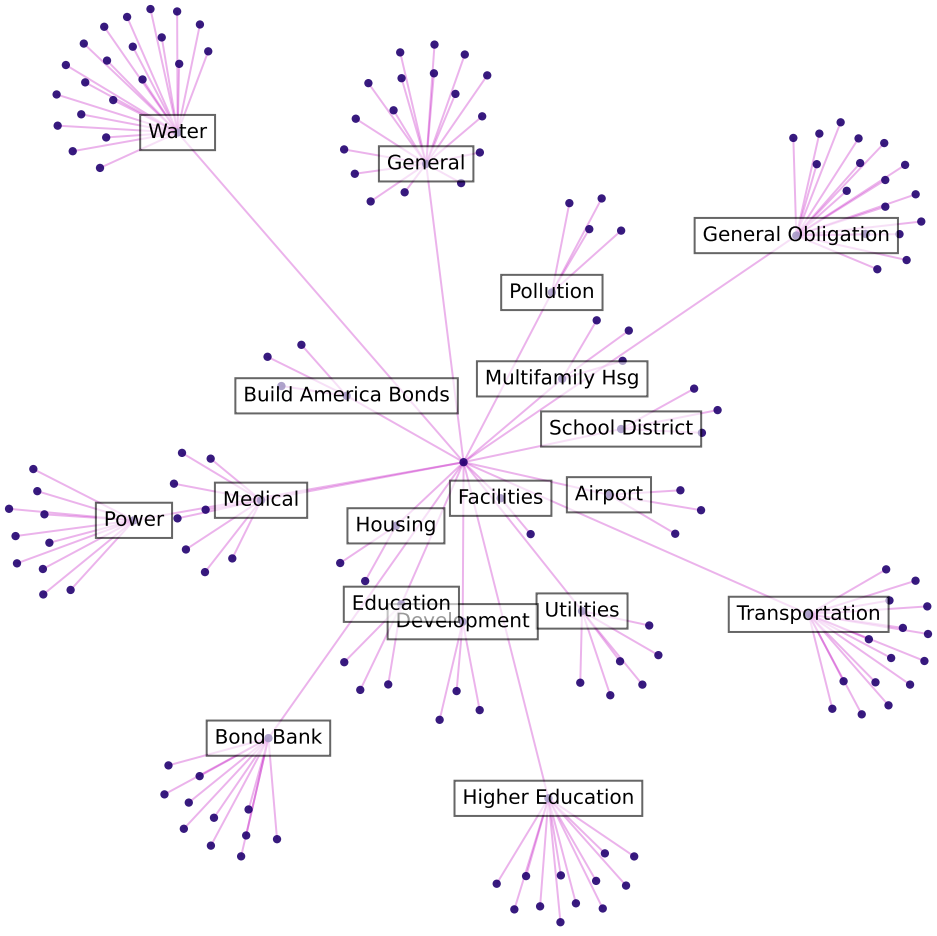
$$Z = \frac{\mathrm{TV}_2(\tilde{\mathbf{y}}) - \mathbb{E}[\mathrm{TV}_2]}{\sqrt{\mathrm{Var}[\mathrm{TV}_2]}}$$

| Maturity Bucket | Mean $Z$-score | Aggregate $Z$ | $p$-value |
|---|---|---|---|
| 5–10 years | $-0.16$ | $-2.58$ | 0.005 |
| 10–20 years | $-0.83$ | $-13.06$ | $< 0.001$ |
| 20–30 years | $-0.77$ | $-12.13$ | $< 0.001$ |
| 30–40 years | $-0.15$ | $-2.25$ | 0.012 |

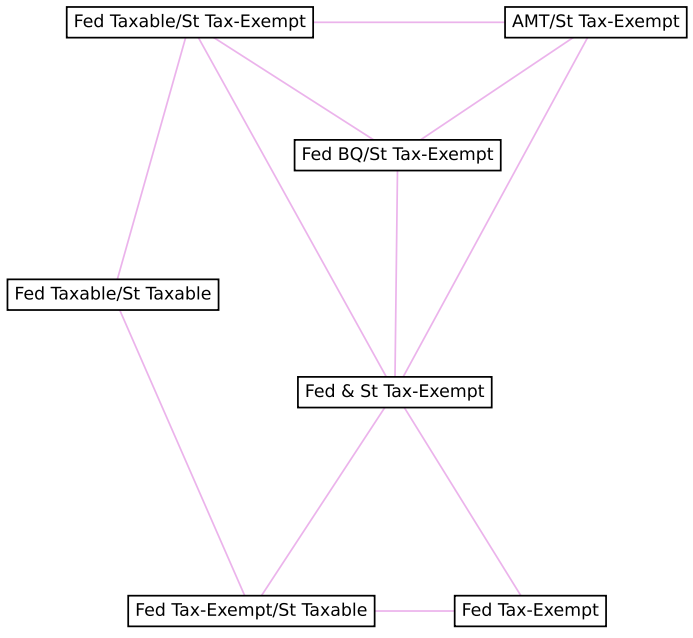
**Result:** All buckets reject $H_0$ ($p < 0.02$). Yields ARE smoother than random on the rating chain — graph structure is empirically validated!

# Graph Structures





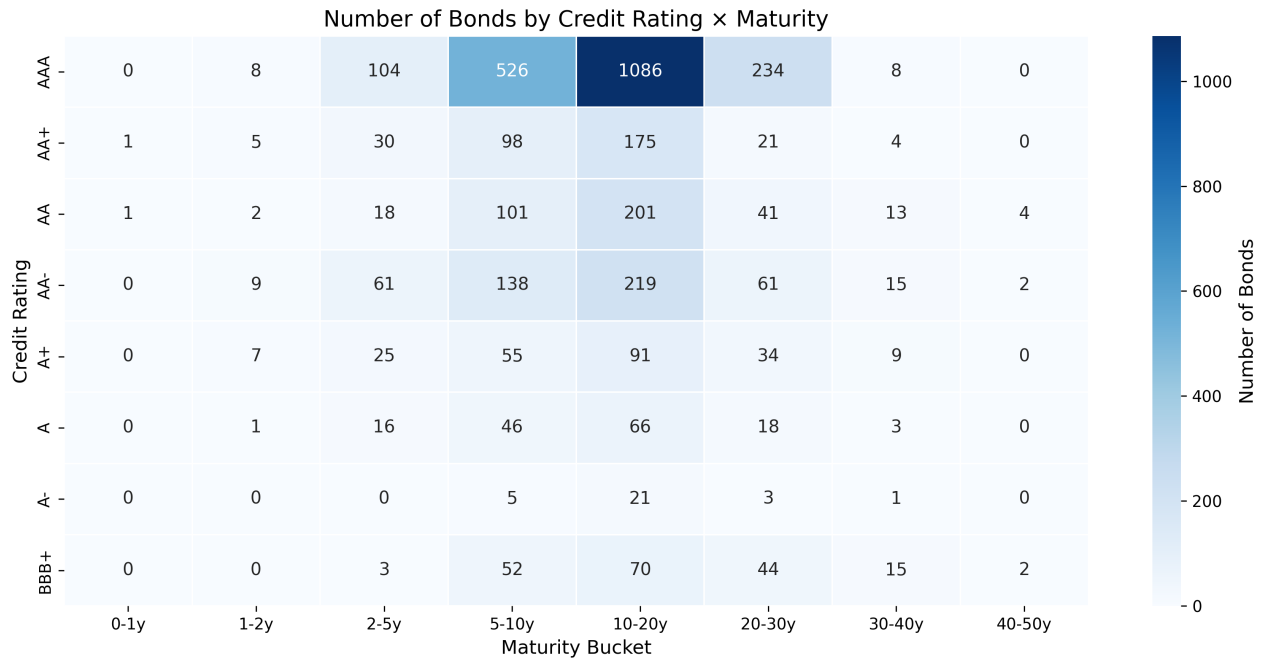

Environmental purpose hierarchy (IEA/Bloomberg taxonomy)




Tax status graph (legal provisions)

# Data Sparsity: Bond Distribution






Number of bonds by Credit Rating × Maturity. Note extreme sparsity at short maturities (0–2y) and lower ratings (A-, BBB+). This is why standard methods fail.

# Predictive Performance: GSR vs Baselines



| | Validation | | Test | |
|---|---|---|---|---|
| **Model** | MSE | $R^2$ | MSE | $R^2$ |
| **GSR (ours)** | 0.332 | 0.819 | **0.319** | **0.823** |
| **KGR (ours)** | 0.333 | 0.820 | **0.317** | **0.824** |
| Ridge | 0.501 | 0.729 | 0.497 | 0.725 |
| Lasso | 0.482 | 0.739 | 0.506 | 0.720 |

- **36% reduction** in prediction error vs Ridge/Lasso
- Test set: 83 bonds **never seen** during training
- Graph structure prevents overfitting despite 99% missing data
- Learned $\beta_{\text{env}} = 2.0$ (strongest) reveals market fragmentation

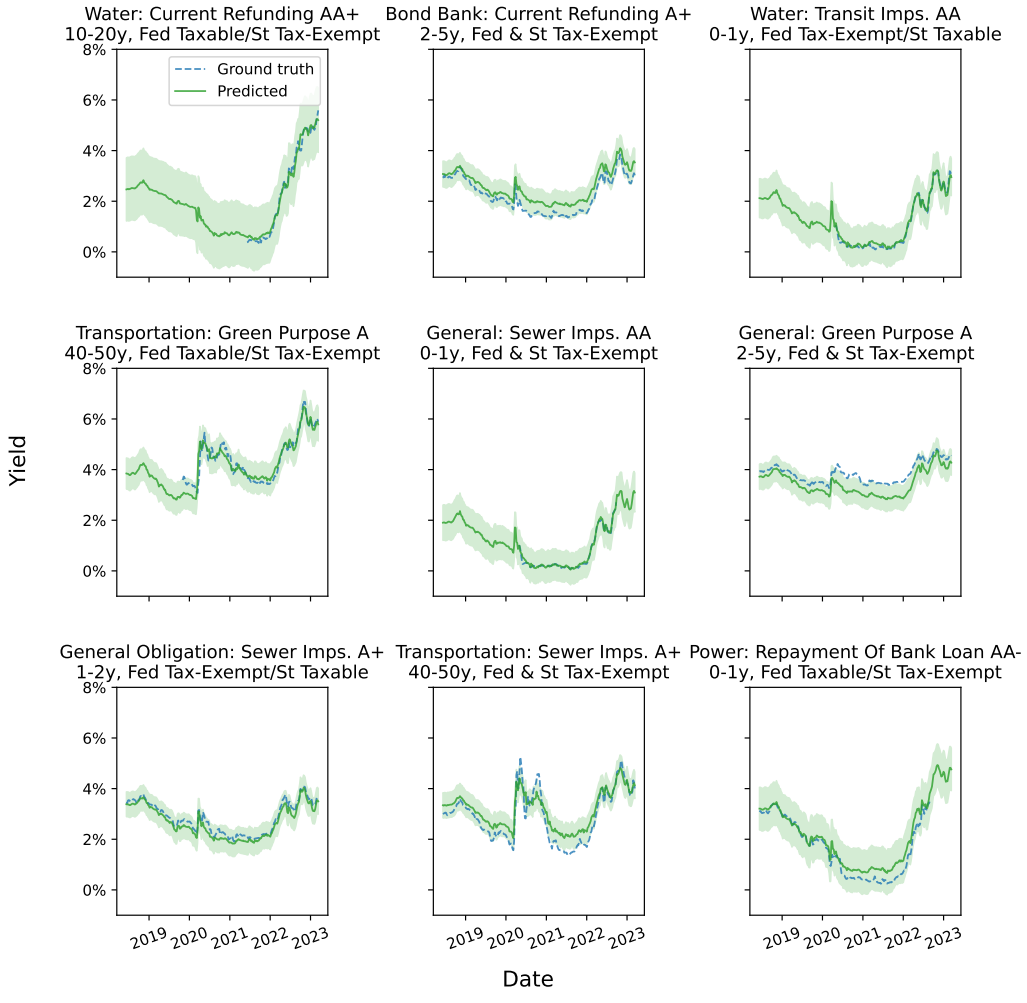
# Out-of-Sample Predictions: GSR




GSR predictions (orange) vs actual yields (blue) for 9 test bonds. Shaded regions show $\pm 2\sigma$ uncertainty. Model captures both levels and dynamics despite never seeing these bonds.

# Out-of-Sample Predictions: KGR

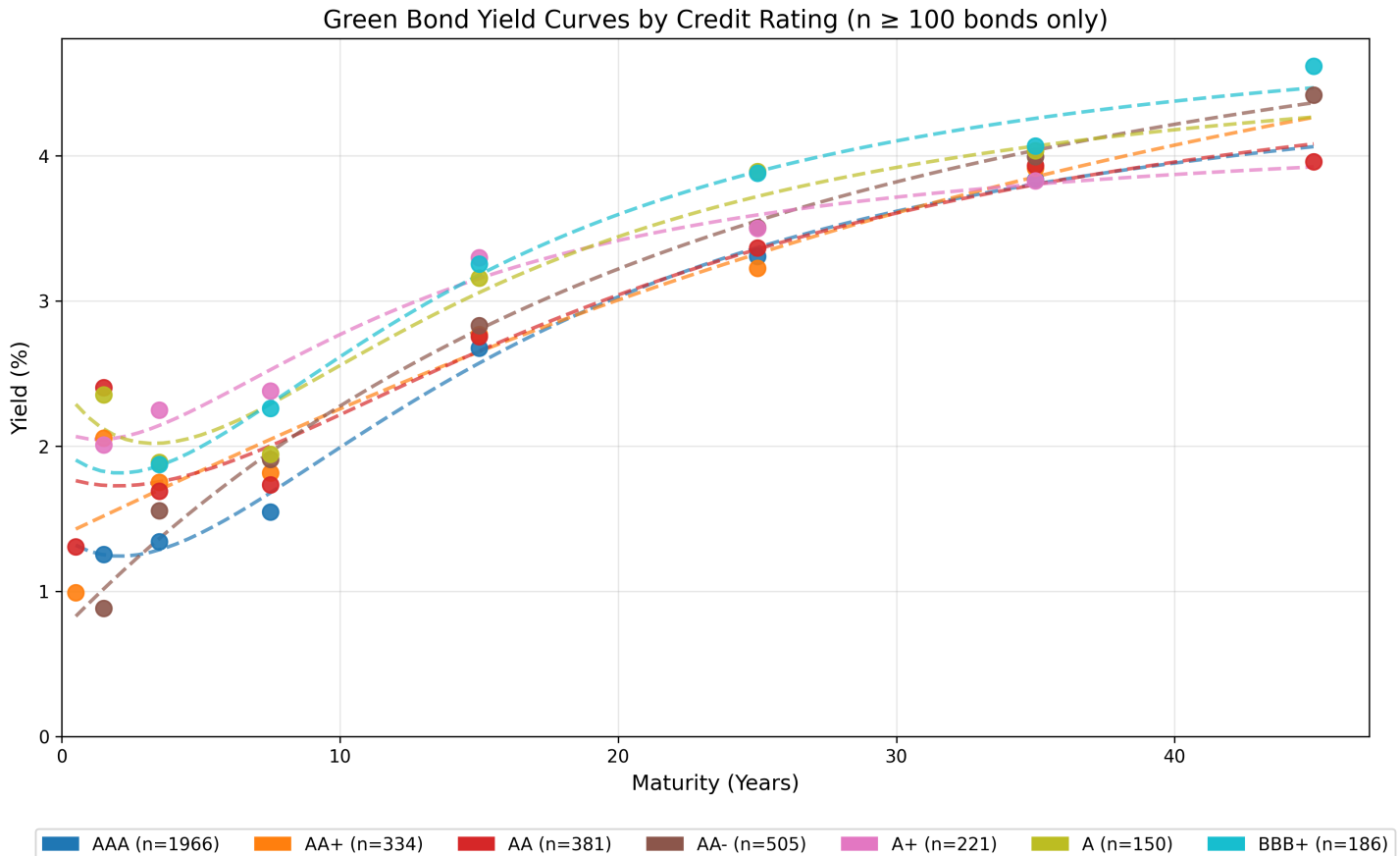





KGR predictions (green) incorporate macroeconomic covariates. Better captures sharp market movements (2022 Fed rate hikes). Uncertainty bands adapt to changing conditions.

# Nelson-Siegel: Fitted Yield Curves

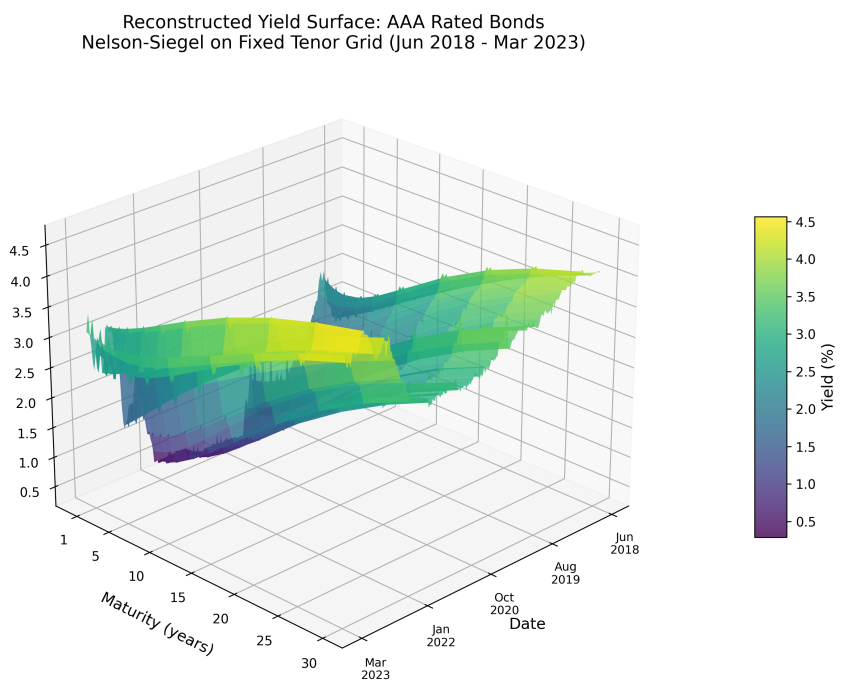




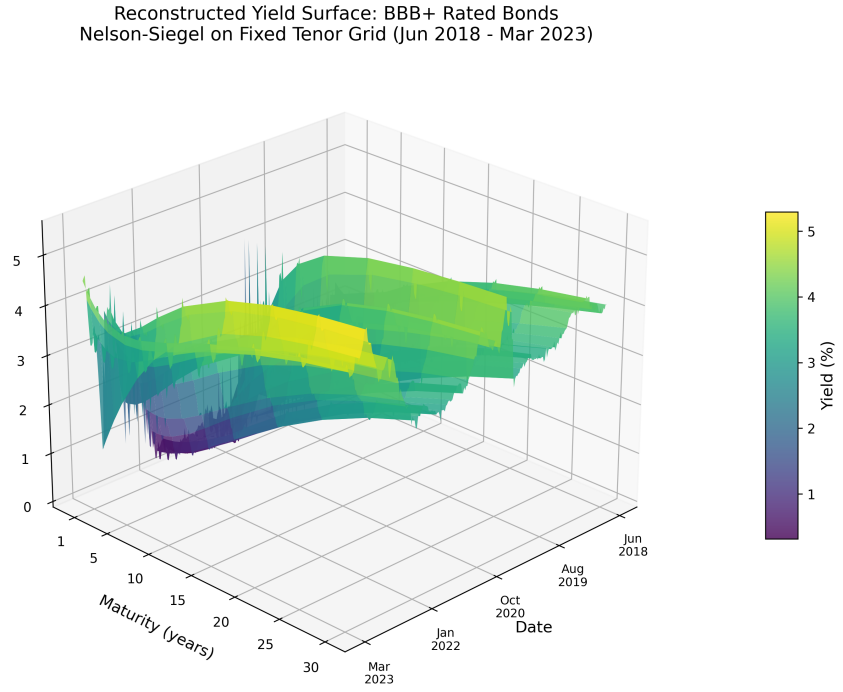

Yield curves by credit rating (ratings with $n \geq 100$ bonds). Dashed lines = NS fit; points = observed averages. Enables yield interpolation at **any** maturity.

# Yield Surface Dynamics






AAA-rated bonds




BBB+ rated bonds

Time evolution of yield curves. COVID decline (March 2020), Fed rate hikes (2022). BBB+ shows higher levels and greater volatility than AAA.

# Table of contents

# Conclusions



- **Problem:** 99% missing data in high-dimensional bond market
- **Solution:** Graph-regularized learning with validated structure
  - Isotropy tests confirm smoothness ($p < 0.001$)
  - Credit chains, maturity sequences, environmental hierarchies
- **Results:** $R^2 = 0.82$ vs 0.72 for Ridge/Lasso (**36% error reduction**)
- **Practical workflow:** GSR $\rightarrow$ Nelson-Siegel $\rightarrow$ actionable yield curves
- **Extensions:** Corporate bonds, MBS, private equity, real estate

# Table of contents

# References



1. Antonian, E. B., Peters, G. W., Chantler, M. J. (2023). PyKronecker: A Python Library for the Efficient Manipulation of Kronecker Products and Related Structures. In *The Journal of Open Source Software, 8(81).*
2. Antonian, E. B., Peters, G. W., Chantler, M. J. (2024). Kernel Generalized Least Squares Regression for Network-Structured Data. Submitted to *Plos One* (In Review).
3. Antonian, E. B., Peters, G. W., Chantler, M. J. (2024). Bayesian Reconstruction of Cartesian Product Graph Signals with General Patterns of Missing Data. *Journal of the Franklin Institute*, 361.9 (2024): 106805.
4. **Campi M., Peters G.W., Antonian E.. Multiway Graph Signal Reconstruction: A Statistical Framework for Modeling Green Bond Yield Structures. Preprint.**

Ph.D. of Dr. Edward Antonian: Supervised by Prof. Peters (UCSB-HW) and Prof. Chantler (HW)

GitHub for Thesis and Code: `https://github.com/nickelnine37`