Graph GPs as emulators for rebalancing Amazon's supply chain network.

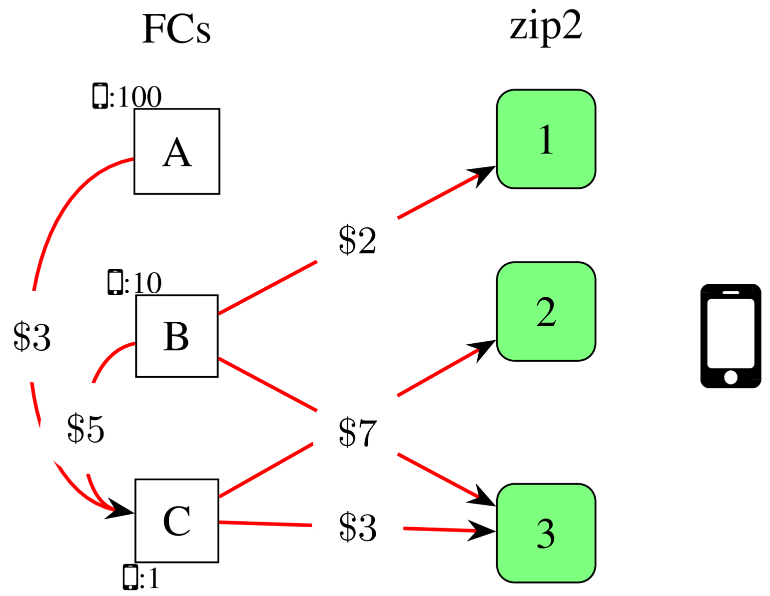
by Stefanos Eleftheriadis (stefele@)

Layman's transfers

$$\min \sum_{\text{all demand scenarios}} (\text{shipping cost} + \text{missed sales penalty}) + \text{transfer cost}$$

$s.t.$

inventory flow is not violated,
outbound shippings and lost sales add up to the total demand,
transferred units between an arc are upper bounded.



Design the emulator

We have a **multi-output** regression:

$$f(\mathbf{x}_n) = \{y_{n,\mathrm{v}}\}_{\mathrm{v}=1}^{V}, \quad \text{V is the set of otuputs/arcs,}$$

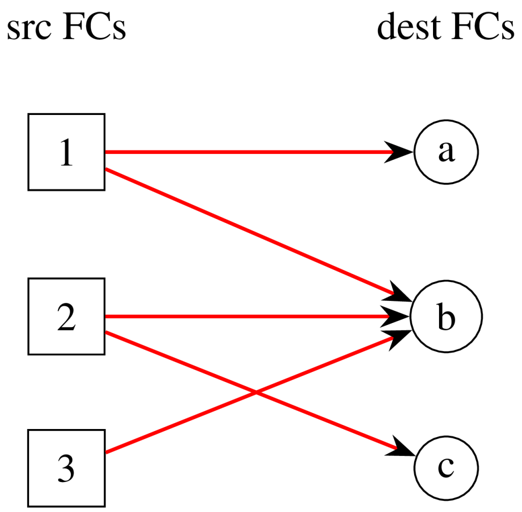
where

$$\mathbf{x}_n = \Bigg[\underbrace{\mathrm{inv}, d, \mathrm{fc}_{\mathrm{lat/long}}}_{\mathrm{FC1}}, \cdots \underbrace{\mathrm{inv}, d, \mathrm{fc}_{\mathrm{lat/long}}}_{\mathrm{FCj}}, \mathrm{pkg}_n\Bigg]^{\top} \in \mathbb{R}^D.$$

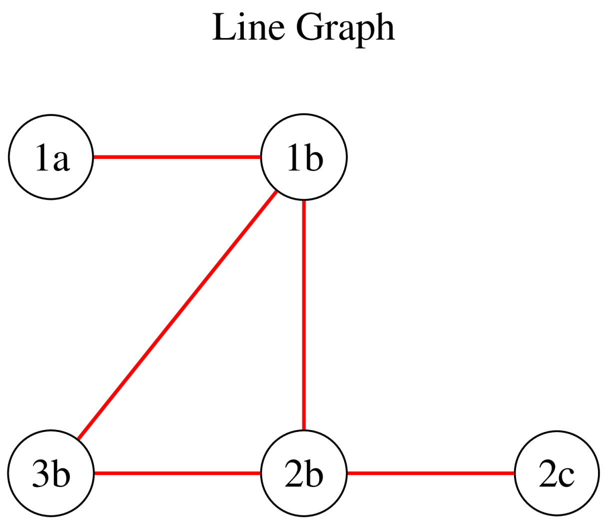
Input features per instance (ASIN-date combination), n:

- **Item properties**: pkg_n (volume/weight),
- **node features**:
  - inv_j (on-hand + in-transit inventory),
  - $\mathrm{fc}_{\mathrm{lat/long}}$ (location),
- **outbound features**: d_j (demand at each node).

Construct the graph from Amazon's network



$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$$



$$L(\mathcal{G}) = \{\mathcal{V}', \mathcal{E}'\}$$

GPs on graphs

We need to define:

$$f_\nu : \mathcal{X} \in \mathbb{R}^{\mathcal{V}' \times D} \to \mathcal{Y} \in \mathbb{R} \quad \text{(glboal funtion for the } \nu \text{th output)}.$$

We start from:

$$g_\nu : \mathbb{R}^D \to \mathbb{R}, \qquad g_\nu(\cdot) \sim \mathcal{GP}(0, k_g(\cdot,\cdot)), \quad \text{(GP on one node)}$$

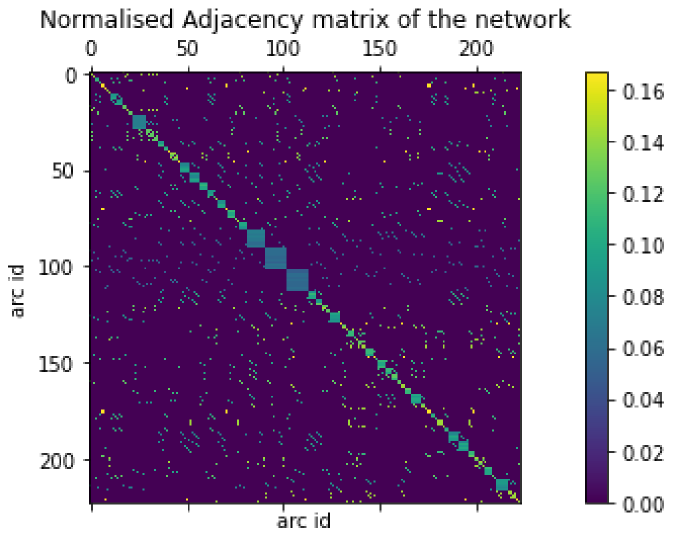
$$f_\nu(\mathbf{x}_n) = \mathbf{w}_\nu{}^\top \begin{bmatrix} g_\nu(\mathbf{x}_n^{[1]}) \\ g_\nu(\mathbf{x}_n^{[2]}) \\ \vdots \\ g_\nu(\mathbf{x}_n^{[V']}) \end{bmatrix}, \quad \mathbf{w}_\nu \in \mathbb{R}^{\mathcal{V}'} \quad \text{(GP on the entire graph)}$$

$$\mathbf{x}_n^{[v]} = \begin{bmatrix} \underbrace{\mathrm{inv}_j, \mathrm{fc}_{j,\mathrm{lat/long}}}_{\mathrm{FC}_j}, \underbrace{\mathrm{inv}_k, d_k, \mathrm{fc}_{k,\mathrm{lat/long}}}_{\mathrm{FC}_k}, \mathrm{pkg}_n \end{bmatrix}^\top \in \mathbb{R}^{D'}.$$

How to choose $\mathbf{W}$?

The graph's normalised adjacency matrix can act as a local smoother:

$$\mathbf{W} = (\mathbf{I} + \mathbf{D})^{-1}(\mathbf{I} + \mathbf{A})$$



Graphs $\longrightarrow$ Convolutions

- graph features $\mathbf{x}_n$ $\rightarrow$ nth "image"
- node features $\mathbf{x}_n^{[v]}$ $\rightarrow$ vth "patch"
- node response $g_\nu(\cdot)$ $\rightarrow$ relu(conv$(\cdot)$)
- graph response $f_\nu(\cdot)$ $\rightarrow$ avg. pooling

ReLUs $\longrightarrow$ Spherical kernels

For some weight $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\sigma_{\text{relu}}(\mathbf{x}^\top \mathbf{w}) = \|\mathbf{x}\|\|\mathbf{w}\| \max(0, \frac{\mathbf{x}^\top \mathbf{w}}{\|\mathbf{x}\|\|\mathbf{w}\|}) = \underbrace{r_x r_w}_{\text{radial}} \underbrace{\sigma_{\text{relu}}(\bar{\mathbf{x}}^\top \bar{\mathbf{w}})}_{\text{angular}} .$$

In the limit, a single layer NN with ReLU converges to a GP with a *spherical* (or *zonal*) kernel:

$$k(\mathbf{x}, \mathbf{z}) = \mathbb{E}_{\mathbf{w}} \left[\sigma_{\text{relu}}(\mathbf{w}^\top \mathbf{x}) \sigma_{\text{relu}}(\mathbf{w}^\top \mathbf{z})\right] = \underbrace{\|\mathbf{x}\|\|\mathbf{z}\|}_{\text{radial}} \underbrace{\frac{1}{\pi}\left(t\left(\pi - \arccos(t)\right) + \sqrt{1 - t^2}\right)}_{\text{angular}}$$

$$= r_x r_z \kappa(t) , \quad t = \bar{\mathbf{x}}^\top \bar{\mathbf{z}} .$$

For L layers the angular part of the equivalent kernel is:

$$\kappa^L(t) := \underbrace{\kappa \circ \cdots \circ \kappa}_{L \text{ times}}(t) .$$

An interlude on spherical harmonics

The eigenfunctions of spherical kernels are spherical harmonics $\phi_\ell^m(\cdot)$ with frequency ℓ and phase m

$$\int_{\mathbb{S}^{d-1}} k(\mathbf{x}, \mathbf{z}) \phi_\ell^m(\mathbf{z}) \, \mathrm{d}\Omega = \lambda_\ell(k) \phi_\ell^m(\mathbf{x}) \, .$$

Spherical harmonics are **orthogonal** wrt to the uniform measure in the sphere:

$$\int_{\mathbb{S}^{d-1}} \phi_\ell^m(\mathbf{x}) \phi_{\ell'}^{m'}(\mathbf{x}) \, \mathrm{d}\Omega = \delta_{\ell\ell'} \delta_{mm'} \, .$$

The Addition theorem connects sphercial harmonics of frequency ℓ to Gegenbauer polynomials of order ℓ:

$$\sum_{m=1}^{N(\ell,d)} \phi_\ell^m(\mathbf{x}) \phi_\ell^m(\mathbf{z}) = \frac{\ell + \alpha}{\alpha} C_\ell^{(\alpha)}(\mathbf{x}^\top \mathbf{z}) \, , \quad \alpha = \frac{d-2}{2} \, .$$

Spherical kernels $\rightarrow$ Polynomials

From Mercer's + Addition thm:

$$k(\mathbf{x}, \mathbf{z}) = r_x r_z \sum_{\ell=0}^{\infty} \sum_{m=0}^{N(\ell,d)} \lambda_\ell \phi_\ell^m(\mathbf{x}) \phi_\ell^m(\mathbf{z}) = r_x r_z \sum_{\ell=0}^{\infty} \frac{\ell + \alpha}{\alpha} \lambda_\ell C_\ell^{(\alpha)}(\mathbf{x}^\top \mathbf{z}) \, .$$

Hint! The eigenvalues decay polynomially as we move to higher frequencies.

Spherical kernels → Polynomials

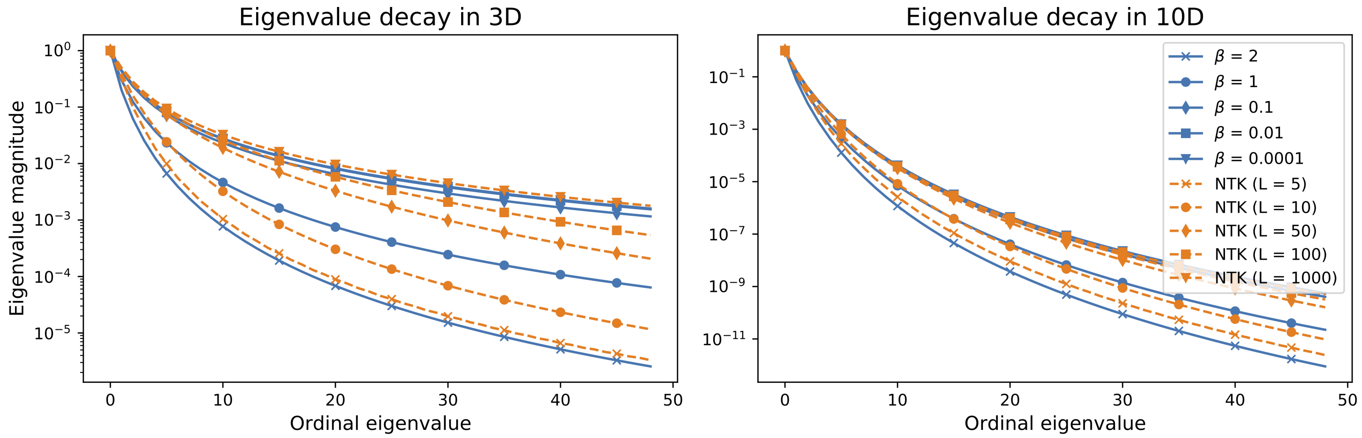
From Mercer's + Addition thm:

$$k(\mathbf{x}, \mathbf{z}) = r_x r_z \sum_{\ell=0}^{\infty} \sum_{m=0}^{N(\ell,d)} \lambda_\ell \phi_\ell^m(\mathbf{x}) \phi_\ell^m(\mathbf{z}) = r_x r_z \sum_{\ell=0}^{\infty} \frac{\ell + \alpha}{\alpha} \lambda_\ell C_\ell^{(\alpha)}(\mathbf{x}^\top \mathbf{z}) \, .$$

Hint! The eigenvalues decay polynomially as we move to higher frequencies.

$$k(\mathbf{x}, \mathbf{z}) = r_x r_z \sum_{\ell=0}^{\infty} \frac{\ell + \alpha}{\alpha} \ell^{-\beta} C_\ell^{(\alpha)}(\mathbf{x}^\top \mathbf{z}) \, .$$

Spherical kernel with "continuous" depth



Back to (sparse) GPs

Inducing points	Inducing features
$u = f(\mathbf{z})$	$u = \langle f, \phi_\ell^m \rangle_{\mathcal{H}}$
$\mathrm{cov}(f(\mathbf{x}), u)$ $= k(\mathbf{x}, \mathbf{z})$	$\mathrm{cov}(f(\mathbf{x}), u)$ $= \phi_\ell^m(\mathbf{x})$
$\mathrm{cov}(u, u')$ $= k(\mathbf{z}, \mathbf{z}')$	$\mathrm{cov}(u, u')$ $= \frac{\delta_{\ell\ell'}\delta_{mm'}}{\lambda_\ell}$

Variational approximation

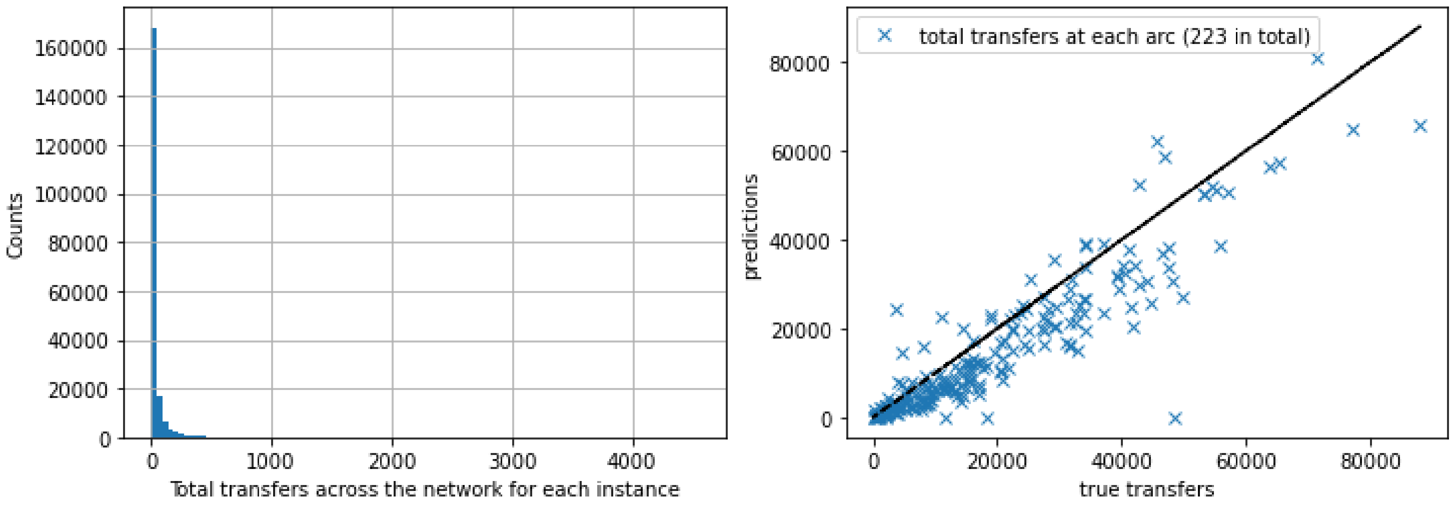
$$\text{ELBO} = \mathbb{E}_{q(f(\cdot))}\left[\log p(y|f(\cdot))\right] - \text{KL}\left[q(f(\cdot))||p(f(\cdot))\right]$$

$$\begin{aligned}
q(f_\nu(\cdot)) &= \mathcal{GP}\left(m_\nu(\cdot), \sigma_\nu(\cdot,\cdot)\right) \\
m_\nu(\cdot) &= k_{f_\nu}(\cdot,\mathbf{Z})\mathbf{K}^{-1}\boldsymbol{\mu}_\nu \rightarrow \phi(\cdot)^\top \text{diag}(\boldsymbol{\lambda})\boldsymbol{\mu}_\nu \\
\sigma_\nu(\cdot,\cdot) &= k_{f_\nu}(\cdot,\cdot) - k_{f_\nu}(\cdot,\mathbf{Z})\mathbf{K}^{-1}k_{f_\nu}(\mathbf{Z},\cdot) + k_{f_\nu}(\cdot,\mathbf{Z})\mathbf{K}^{-1}\mathbf{S}_\nu\mathbf{K}^{-1}k_{f_\nu}(\mathbf{Z},\cdot) \\
&\rightarrow k_{f_\nu}(\cdot,\cdot) - \phi(\cdot)^\top \text{diag}(\boldsymbol{\lambda})\phi(\cdot) + \phi(\cdot)^\top \text{diag}(\boldsymbol{\lambda})\mathbf{S}_\nu \text{diag}(\boldsymbol{\lambda})\phi(\cdot)\,,
\end{aligned}$$

$$\text{with} \quad k_{f_\nu}(\cdot,\cdot) = \mathbf{w}_\nu^\top k_{g_\nu}(\cdot,\cdot)\mathbf{w}_\nu \quad \text{and} \quad k_{f_\nu}(\cdot,\mathbf{Z}) = \mathbf{w}_\nu^\top k_{g_\nu}(\cdot,\mathbf{Z}).$$

The emulator in action

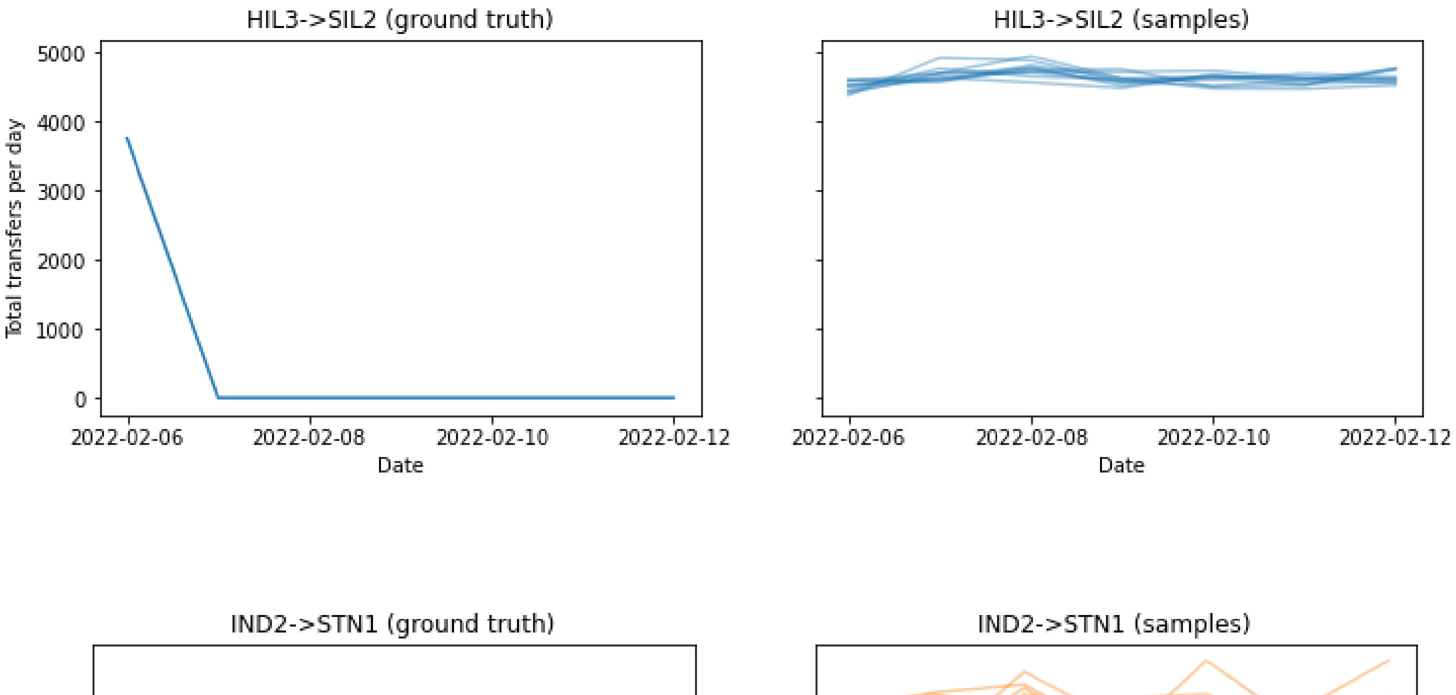
Ugly data, good results



Dive into each arc

Same plot ordered by volume (descending)

Best vs worst performing arc



Important papers

- Ng et al. Bayesian Semi-supervised Learning with Graph Gaussian Processes.
- Kipf & Welling. Semi-Supervised Classification with Graph Convolutional Networks.
- van der Wilk et al. Convolutional Gaussian Processes.
- Dutordoir et al. Sparse Gaussian Process with Spherical Harmonic features.
- Bietti & Bach. Deep Equals Shallow for ReLU Networks in Kernel Regimes.
- Belfer et al. Spectral Analysis of the Neural Tangent Kernel for Deep Residual Networks.