Gaussian Processes with Applications

Mattias Villani

Department of Statistics
Stockholm University

and

Department of Computer and Information Science
Linköping University

- Introduction to **Gaussian process regression**
- Gaussian process all the things - some **applications**
- No time for **Bayesian optimization** :(

Cool in Machine Learning now, but ... Statisticians were rockin' it already in the 70's

Biometrika (1975), **62**, 1, *p.* 79 **79**
With 5 text-figures
Printed in Great Britain

A Bayesian approach to model inadequacy for polynomial regression

By B. J. N. BLIGHT
Department of Statistics, Birkbeck College, London

and L. OTT
Department of Statistics, University of Florida, Gainesville

Cool in Machine Learning now, but ...
Statisticians were rockin' it already in the 70's

J. R. Statist. Soc. B (1978),
40, *No.* 1, *pp.* 1–42

Curve Fitting and Optimal Design for Prediction

By A. O'Hagan

University of Warwick

[Read before the Royal Statistical Society at a meeting organized by the Research Section on Wednesday, October 12th, 1977, Professor J. F. C. Kingman in the Chair]

Summary

A Bayesian approach to density estimation

THORBURN DANIEL

Biometrika, Volume 73, Issue 1, 1 April 1986, Pages 65–75, https://doi-org.e.bibl.liu.se/10.1093/biomet/73.1.65

Published: 01 April 1986 **Article history ▾**

- **Linear regression**

$$y = f(\mathbf{x}) + \epsilon$$

$$f(\mathbf{x}) = \mathbf{x}^T \beta$$

and $\epsilon \sim N(0, \sigma_n^2)$ and iid over observations.

- **Polynomial regression**: $\phi(\mathbf{x}) = (1, x, x^2, x^3, ..., x^k)$:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \beta.$$

- More generally: **splines** with **basis functions**.
- Polynomial and spline models are linear in β. Least squares!

- **Model**: Linear regression for all n observations

$$\underset{n\times 1}{\mathbf{y}} = \underset{n\times p}{\mathbf{X}}\underset{p\times 1}{\beta} + \underset{n\times 1}{\varepsilon}$$

- **Prior**

$$\beta \sim N\left(0, \Sigma_p\right)$$

- Common choice (Ridge regression): $\Sigma_p = \lambda^{-1}\mathbf{I}$.
- **Posterior**

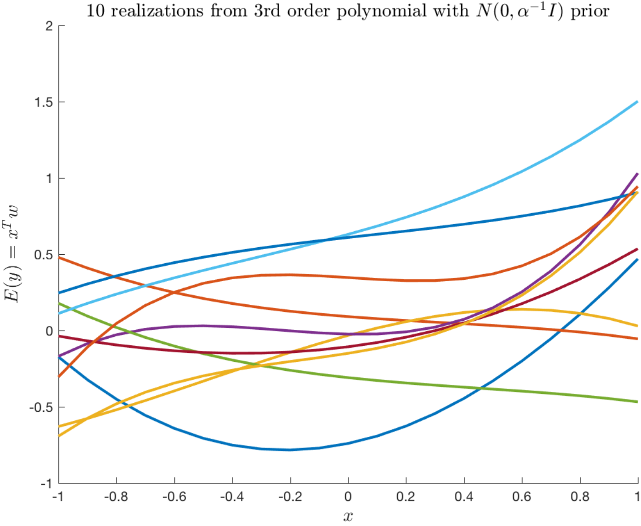
$$\beta|\mathbf{X},\mathbf{y} \sim N\left(\bar{\beta}, \mathbf{A}^{-1}\right)$$

$$\mathbf{A} = \sigma_n^{-2}\mathbf{X}^T\mathbf{X} + \Sigma_p^{-1}$$

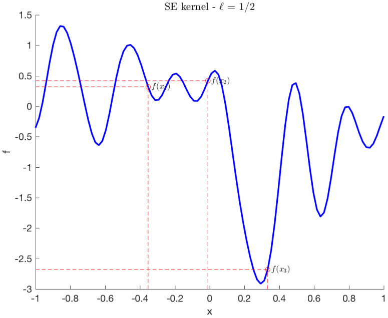
$$\bar{\beta} = \sigma_n^{-2}\left(\sigma_n^{-2}\mathbf{X}^T\mathbf{X} + \Sigma_p^{-1}\right)^{-1}\mathbf{X}^T\mathbf{y}$$

- **Posterior precision = Data Precision + Prior Precision**.

A prior on β is really a prior over functions



- **Non-parametric regression**: avoid a parametric form for $f(\cdot)$.
- Treat $f(\mathbf{x})$ as **an unknown parameter for every x**.



- A *new* parameter for *every* **x**, you must be joking?
- Instead of restricting to linear, impose **smoothness**.

- **Weight space view**
- Restrict attention to a grid of x-values: $x_1, .., x_k$.
- Put a joint prior on the **vector of k function values**

$$f(x_1), ..., f(x_k)$$

- **Function space view**
- Treat f **as an unknown function**.
- Put a prior over a set of functions.

- A GP implies:

$$\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_k) \end{pmatrix} \sim N(\mathbf{m}, \mathbf{K})$$

- But how do we specify the $k \times k$ **covariance matrix** $\mathbf{K}$?

$$Cov\left(f(x_p), f(x_q)\right)$$

- **Squared exponential covariance function**

$$Cov\left(f(x_p), f(x_q)\right) = k(x_p, x_q) = \sigma_f^2 \exp\left(-\frac{1}{2}\left(\frac{x_p - x_q}{\ell}\right)^2\right)$$

- Nearby x's have highly correlated function ordinates $f(x)$.
- We can compute $Cov\left(f(x_p), f(x_q)\right)$ for *any* x_p and x_q.

Definition

A **Gaussian process** (**GP**) is a collection of random variables, any finite number of which have a multivariate Gaussian distribution.

- A GP is a **probability distribution over functions**.
- A GP is specified by a **mean** and a **covariance function**

$$m(x) = \mathrm{E}\left[f(x)\right]$$

$$k(x, x') = E\left[(f(x) - m(x))\left(f(x') - m(x')\right)\right]$$

  for any two inputs x and x'.
- A **Gaussian process** is denoted by

$$f(x) \sim \mathrm{GP}\left(m(x), k(x, x')\right)$$

- $f(x) \sim \mathrm{GP}$ encodes **prior beliefs** about the unknown $f(\cdot)$.

- Let $r = \|x - x'\|$.
- **Squared exponential** (**SE**) kernel ($\ell > 0, \sigma_f > 0$)

$$K_{SE}(r) = \sigma_f^2 \exp\left(-\frac{r^2}{2\ell^2}\right)$$

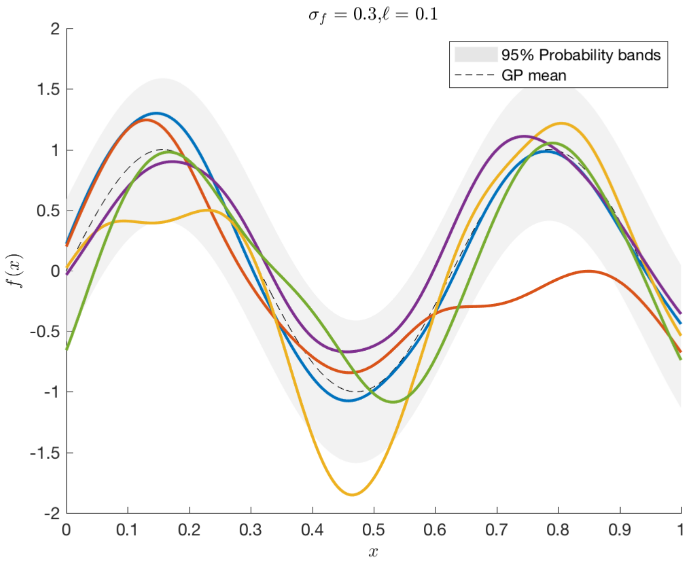
- **Matérn** kernel ($\ell > 0, \sigma_f > 0, \nu > 0$)

$$K_{Matern}(r) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\ell}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu} r}{\ell}\right)$$

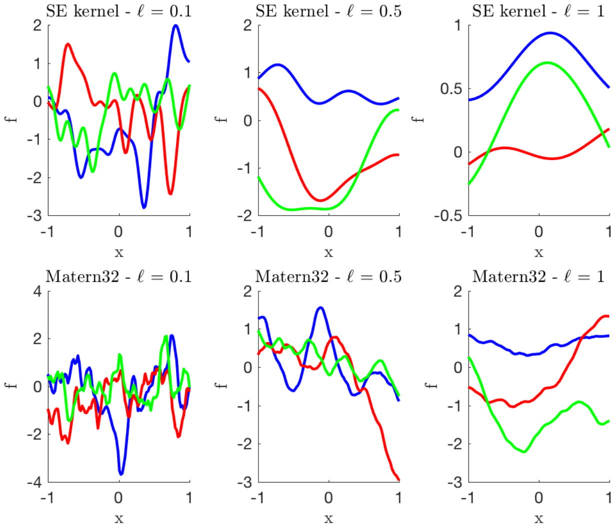
- **Simulate draw** from $f(x) \sim \mathrm{GP}(m(x), k(x, x'))$ by:
  - form a grid $\mathbf{x}_* = (x_1, ..., x_n)$
  - simulate function values from multivariate normal:

$$f(\mathbf{x}_*) \sim N(m(\mathbf{x}_*), K(\mathbf{x}_*, \mathbf{x}_*))$$

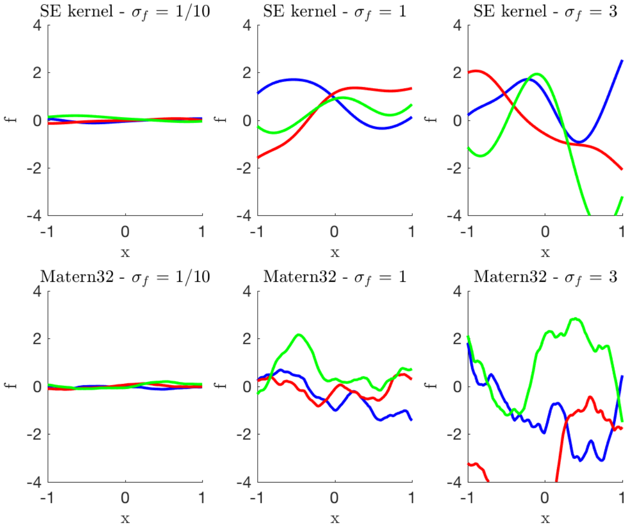
Simulating a GP



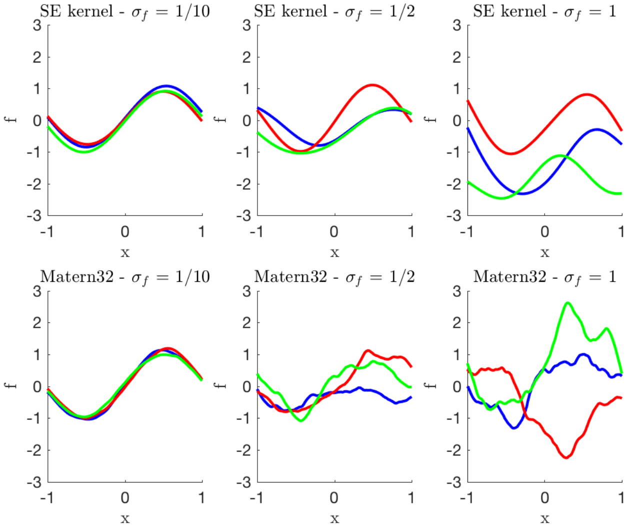
The length scale ℓ determines the smoothness



The scale factor σ_f determines the variance



The mean can be sin(3x). Or whatever.



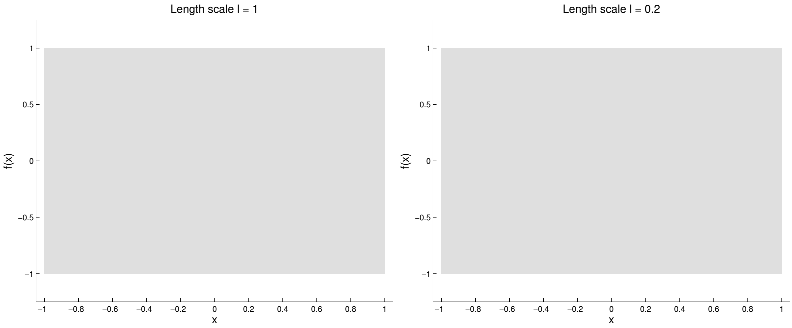
- The joint way: Choose a grid $x_1, ..., x_k$. Simulate the k-vector

$$\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_k) \end{pmatrix} \sim N(\mathbf{m}, \mathbf{K})$$

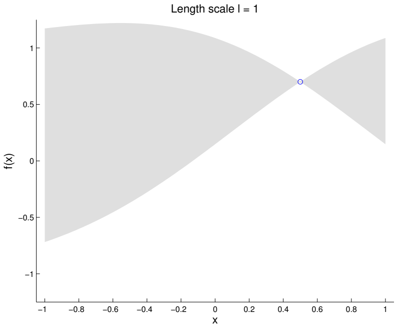
- More intuition from the conditional decomposition

$$\begin{aligned} p\left(f(x_1), f(x_2), ..., f(x_k)\right) &= p\left(f(x_1)\right) p\left(f(x_2)|f(x_1)\right) \cdots \\ &\times p\left(f(x_k)|f(x_1), ..., f(x_{k-1})\right) \end{aligned}$$

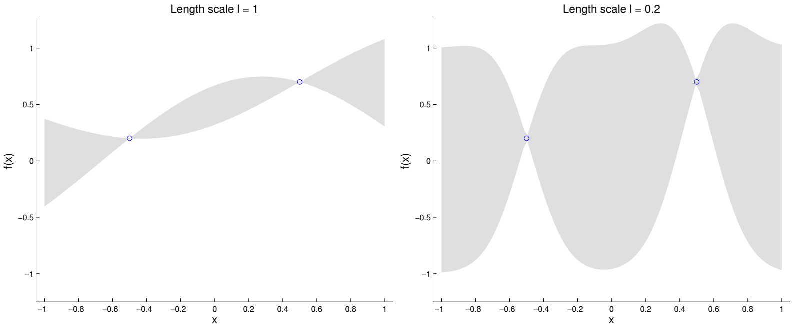
Simulating from $p\left(f(x_1)\right)$



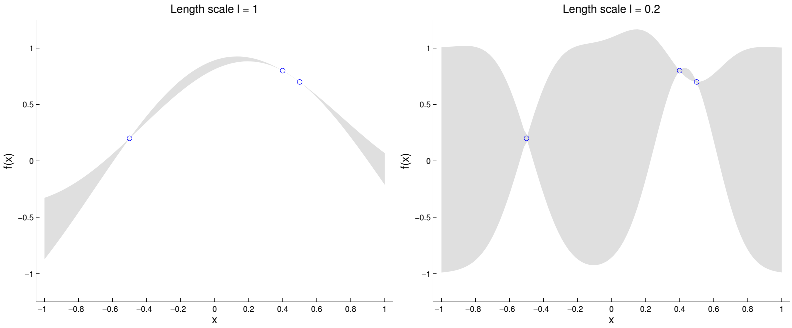
SIMULATING FROM $p\left(f(x_2)|f(x_1)\right)$



Simulating from $p\left(f(x_3)|f(x_1), f(x_2)\right)$



SIMULATING FROM $p\left(f(x_4) | f(x_1), f(x_2), f(x_3)\right)$



- **Model**

$$y_i = f(x_i) + \varepsilon_i, \quad \varepsilon \overset{iid}{\sim} N(0, \sigma_n^2)$$

- **Prior**

$$f(x) \sim GP\left(0, k(x, x')\right)$$

- **Observed**: $\mathbf{x} = (x_1, ..., x_n)^T$ and $\mathbf{y} = (y_1, ..., y_n)^T$.
- **Goal**: posterior of $f(\cdot)$ over a grid of x-values: $\mathbf{f}_* = \mathbf{f}(\mathbf{x}_*)$.
- **Posterior**

$$\begin{aligned}
\mathbf{f}_* | \mathbf{x}, \mathbf{y}, \mathbf{x}_* &\sim N\left(\bar{\mathbf{f}}_*, \mathrm{cov}(\mathbf{f}_*)\right) \\
\bar{\mathbf{f}}_* &= K(\mathbf{x}_*, \mathbf{x})\left[K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I\right]^{-1} \mathbf{y} \\
\mathrm{cov}(\mathbf{f}_*) &= K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{x})\left[K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I\right]^{-1} K(\mathbf{x}, \mathbf{x}_*)
\end{aligned}$$

- Idea: obtain joint $p(\mathbf{y}, \mathbf{f}_*)$ and then $p(\mathbf{f}_*|\mathbf{y})$ by conditioning.
- **Model**

$$y_i = f(\mathrm{x}_i) + \varepsilon_i, \quad \varepsilon \overset{iid}{\sim} N(0, \sigma_n^2)$$

- **Prior**

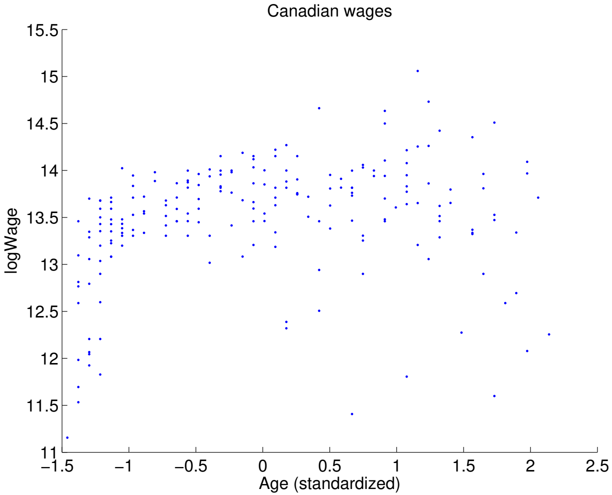
$$f(x) \sim GP\left(0, k(x, x')\right)$$

- Joint distribution of $(\mathbf{y}, \mathbf{f}_*)$

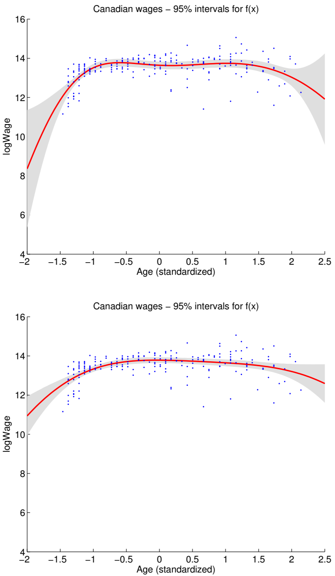
$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim \mathrm{N}\left[\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I & K(\mathbf{x}, \mathbf{x}_*) \\ K(\mathbf{x}_*, \mathbf{x}) & K(\mathbf{x}_*, \mathbf{x}_*) \end{pmatrix}\right]$$

- Result: conditional distributions from multivariate normal are normal.

Example - Canadian wages



Posterior of f - $\ell = 0.2, 0.5, 1, 2$



- Kernel depends on **hyperparameters** θ. Example SE kernel $[\theta = (\sigma_f, \ell)^T]$

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\ell^2}\right)$$

- Common: maximize the **marginal likelihood** wrt θ:

$$p(\mathbf{y}|\mathbf{X}, \theta) = \int p(\mathbf{y}|\mathbf{X}, \mathbf{f}, \theta)p(\mathbf{f}|\mathbf{X}, \theta)d\mathbf{f}$$

  $\mathbf{f} = f(\mathbf{X})$ is a vector of function values in the training data.
- For **Gaussian process regression**:

$$\log p(\mathbf{y}|\mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^T\left(K + \sigma_n^2 I\right)^{-1}\mathbf{y} - \frac{1}{2}\log\left|K + \sigma_n^2 I\right| - \frac{n}{2}\log(2\pi)$$

- Proper **Bayesian inference for hyperparameters**

$$p(\theta|\mathbf{y}, \mathbf{X}) \propto p(\mathbf{y}|\mathbf{X}, \theta)p(\theta).$$

- **Binary** or multi-class **response**. Aim: $\Pr(y_i = 1|\mathbf{x}_i)$.
- **Logistic regression**

$$\Pr(y_i = 1|\mathbf{x}_i) = \lambda(\mathbf{x}_i^T \beta), \text{ where } \lambda(z) = \frac{1}{1 + \exp(-z)}.$$

- $\lambda(z)$ 'squashes' the linear prediction $\mathbf{x}^T \beta \in \mathbb{R}$ into $[0, 1]$.
- **Linear decision boundaries** because of linear predictor $\mathbf{x}^T \beta$.
- **GP classification**: replace $\mathbf{x}^T \beta$ by $f(\mathbf{x})$ where

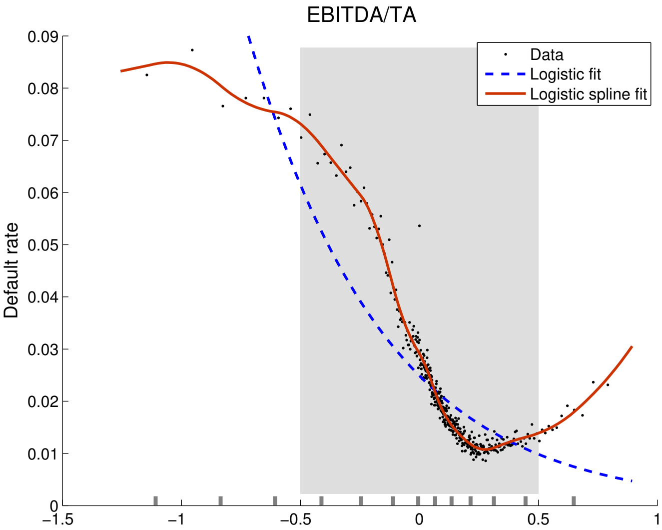
$$f \sim \mathrm{GP}(0, k(\mathbf{x}, \mathbf{x}'))$$

  and squash f through logistic function

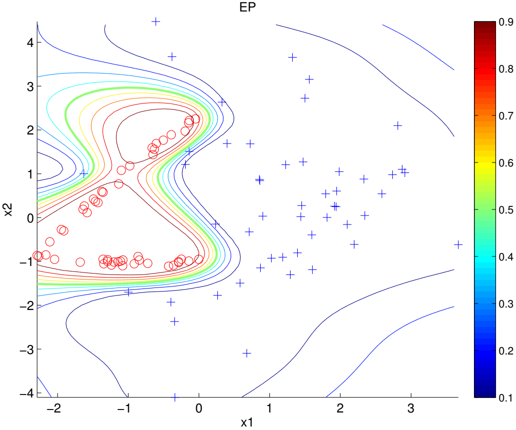
$$\Pr(y = 1|\mathbf{x}) = \lambda(f(\mathbf{x}))$$

- Nonparametric **flexible decision boundaries**.

Classification - firm bankruptcy



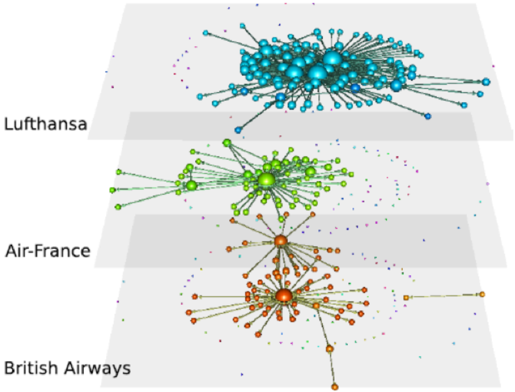
GP Classification on simulated data



- **Aim**: **Real-time prediction** of road travel times.
- **Data**: noisy GPS from a large number of taxis. SL bus data.
- **Model**: **Structured Multivariate Gaussian Process**:
  - **GPs over time** for each street section. Periodic kernels.
  - **Hierarchical** factor-like model. Traffic on main roads feed the small roads.
  - **Spatial**, **topological**, dependence between street sections.

AIRLINE NETWORK PREDICTIONS

- **Aim**: **Predict** the evolution of airline **networks over time**.
- **Data**: Quarterly world-wide networks for all airlines.
- **Model**: **Dynamic multi-layered networks** driven by **GPs**.



- **Bernoulli model** with **bilinear latent Gaussian processes**:

$$Y_{uv}^{(k)}(t)|\pi_{uv}^{(k)}(t) \sim \mathrm{Bern}\left[\pi_{uv}^{(k)}(t)\right]$$

$$\mathrm{Logit}\left[\pi_{uv}^{(k)}(t)\right] = \mu(t) + \sum_{r=1}^{R} \bar{x}_{ur}(t)\bar{x}_{vr}(t) + \sum_{h=1}^{H} x_{uh}^{(k)}(t)x_{vh}^{(k)}(t),$$

  where
  - **Global GP** across actors and layers: $\mu(t)$
  - **Actor-specific GPs**, but common across layers: $\bar{x}_{ur}(t)$.
  - **Actor-specific and layer-specific** GPs: $x_{uh}^{(k)}(t)$.

- Scalability.

Thanks!