Winter Conference in Statistics
Bayesian Machine Learning

Making Use of GPs and Bayesian Optimization

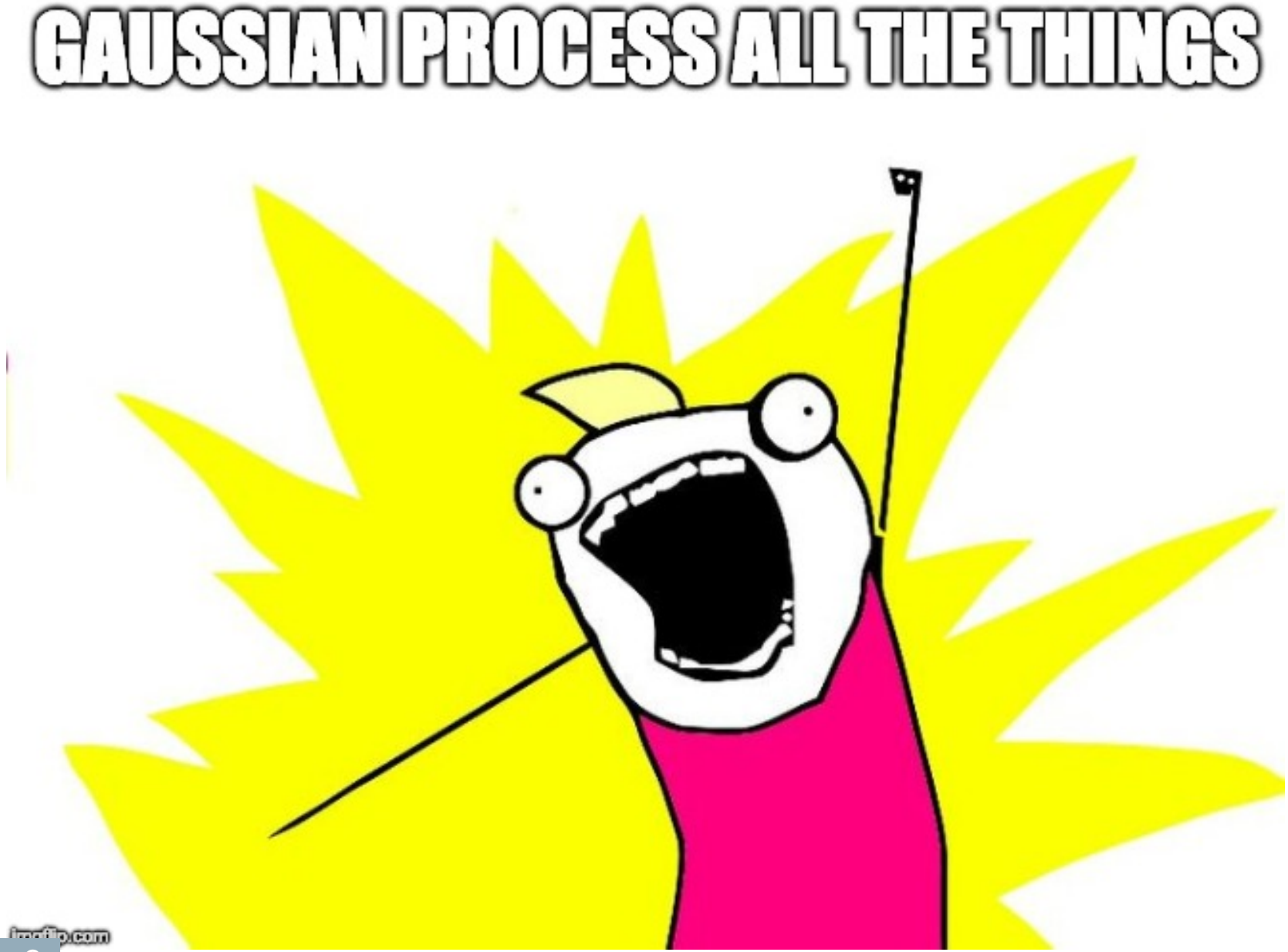
Mattias Villani

Department of Statistics
Stockholm University
and
Department of Computer and Information Science
Linköping University

Overview

- GP classification
- Real-time robotic search
- Flexible hemodynamics modeling for brain imaging
- Modeling the evolution of airline network structures
- Bayesian optimization

Alternative title of lecture



GP Classification

- **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

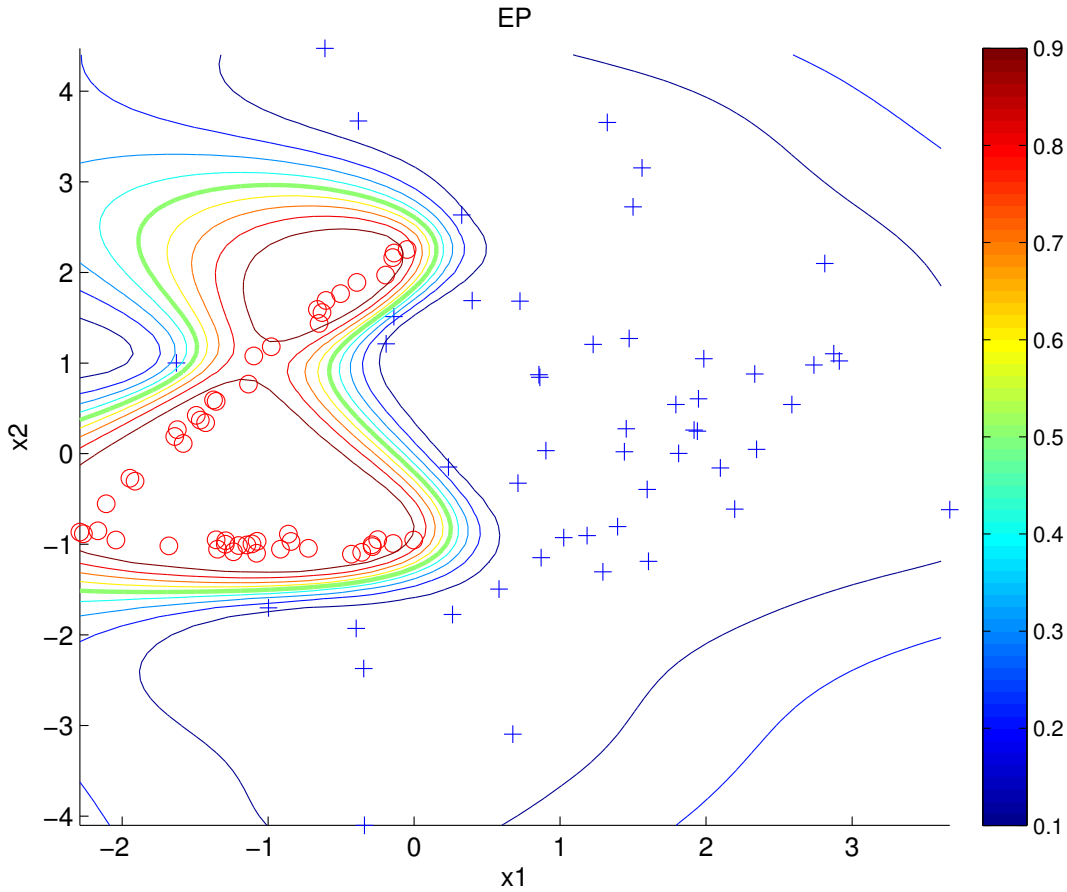
$$f \sim \mathrm{GP}(0, k(\mathbf{x}, \mathbf{x}'))$$

and squash f through logistic function

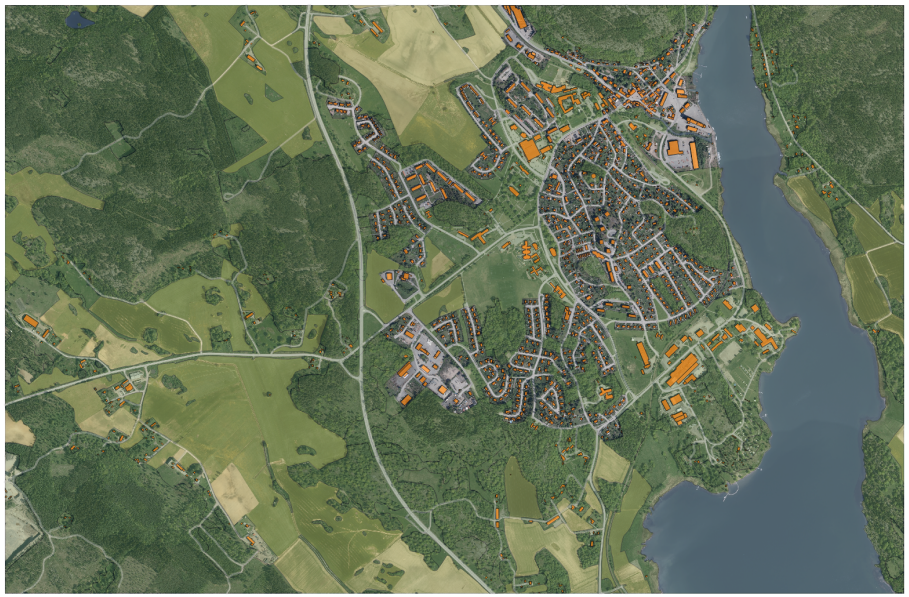
$$\Pr(y = 1|\mathbf{x}) = \lambda(f(\mathbf{x}))$$

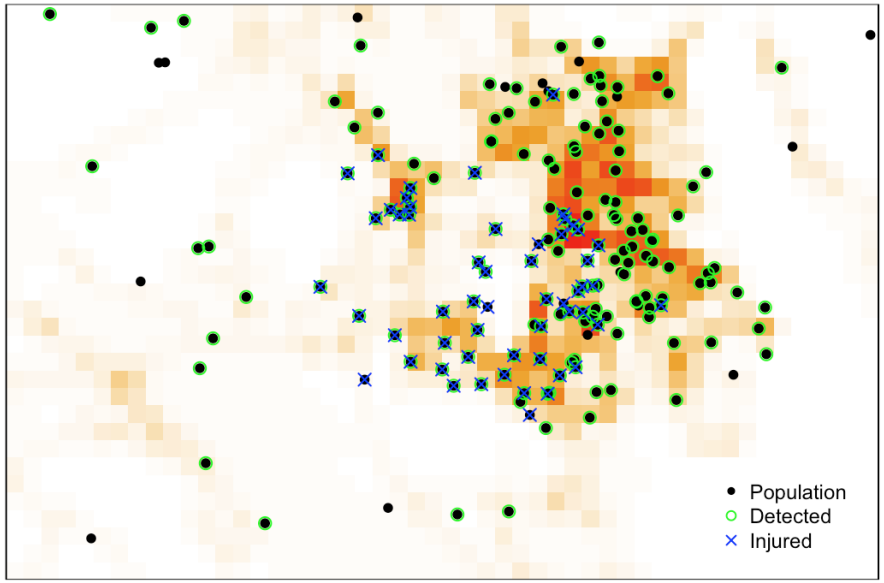
- Nonparametric **flexible decision boundaries**.

GP Classification on simulated data



Simulated data over Gamleby[1]





[1]Work with Olov Andersson, Per Sidén, Johan Dahlin, Patrick Doherty.

Hierarchical Spatial Point Process

- **Log Gaussian Cox Process** (LGCP) for **number of persons** in the subset $\tilde{S} \subset S$.

$$N_{y^\star}(\tilde{S})|\lambda \sim \text{Poisson}\left(\int_{\mathbf{s}\in\tilde{S}} \lambda(\mathbf{s})d\mathbf{s}\right)$$

$$\log\lambda(\mathbf{s}) = \alpha_\lambda + \underbrace{\mathbf{x}_\lambda^\top(\mathbf{s})}_{GIS}\beta_\lambda + \underbrace{\xi_\lambda(\mathbf{s})}_{GP\text{ in }2D}$$

- The **number of detected** persons by a thinned LGCP

$$N_y(\tilde{S})|r,\lambda \sim \text{Poisson}\left(\int_{\mathbf{s}\in\tilde{S}} r(\mathbf{s})\lambda(\mathbf{s})d\mathbf{s}\right)$$

$$\log r(\mathbf{s}) = \mathbf{x}_r^\top(\mathbf{s})\beta_r$$

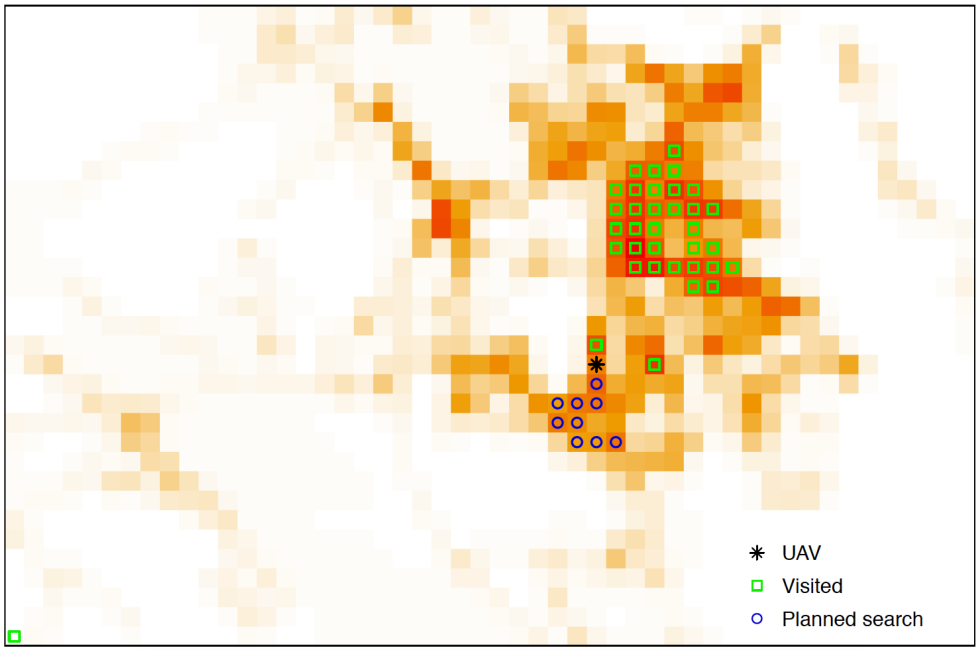
- **Probability of injury**

$$w_i|q \sim \text{Bernoulli}\left(q(\mathbf{y}_i)\right),$$

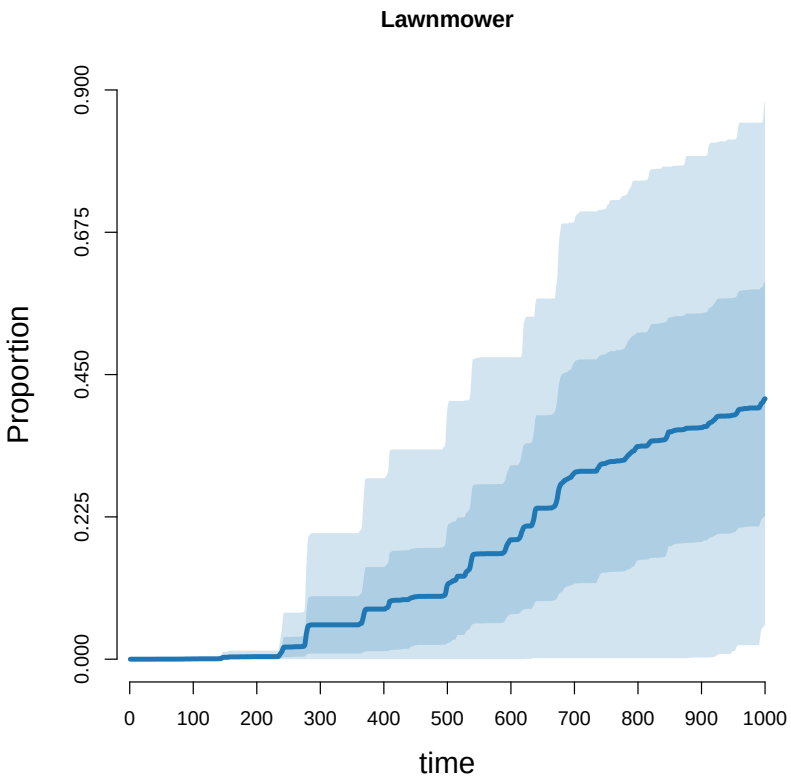
$$\log\left(\frac{q(\mathbf{s})}{1-q(\mathbf{s})}\right) = \alpha_q + \mathbf{x}_q^\top(\mathbf{s})\beta_q + \xi_q(\mathbf{s})$$

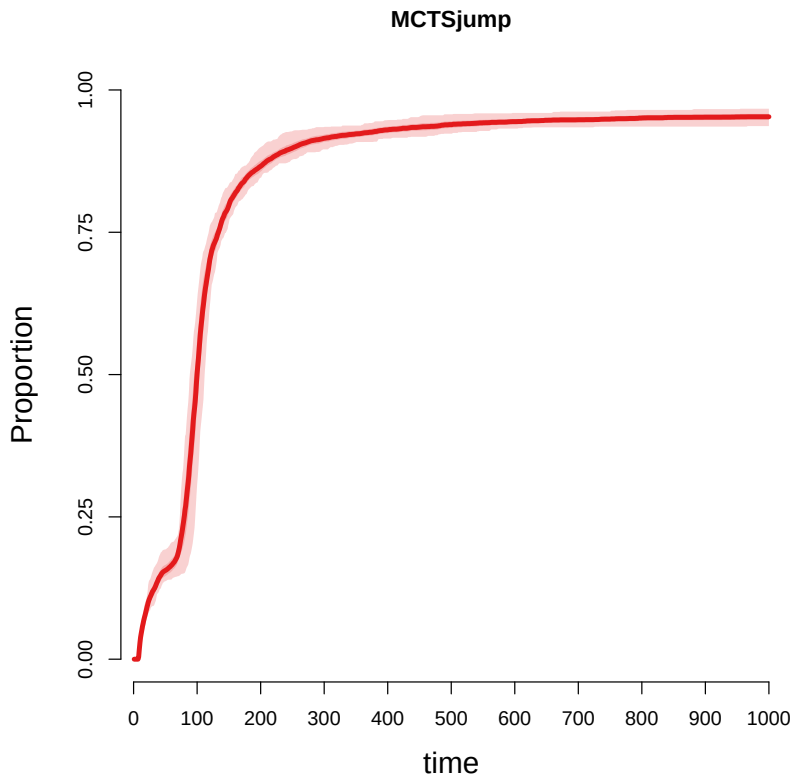
Real-time decision making under uncertainty

- Challenges
  1. **missing data** - point pattern is only partially observed
  2. **real-time sequential inference over spatial fields**
  3. **real-time decision making** under uncertainty
- Solutions
  1. Strong priors based on **GIS data**
  2. **Warm-started INLA** for inference
  3. Tailored **Monte Carlo Tree Search** for decision



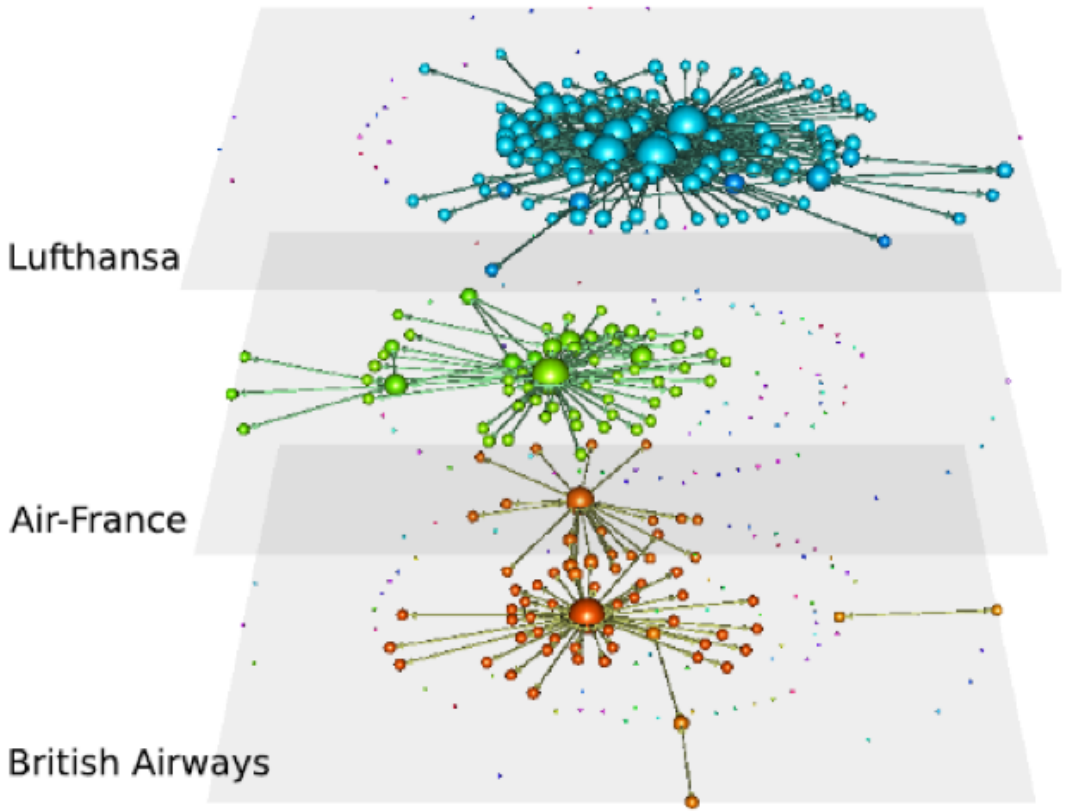
We find injured a lot faster than lawnmower





Airline network evolution

- **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**.



Dynamic networks driven by latent variables

- **Static Bernoulli model** for adjacency matrix **Y**

$$Y_{uv}(t)|\pi \sim \text{Bern}(\pi)$$

- **Dynamic Bernoulli** with **global latent Gaussian process**

$$\begin{aligned} Y_{uv}(t)|\pi(t) &\sim \text{Bern}\left(\pi(t)\right) \\ \text{Logit}\left[\pi(t)\right] &= z(t), \\ z(t) &\sim \text{GP}(\mu(t), K(t', t)) \end{aligned}$$

- **Dynamic Bernoulli** with **latent Gaussian processes at nodes**

$$\begin{aligned} Y_{uv}(t)|(t) &\sim \text{Bern}\left[\pi_{uv}(t)\right] \\ \text{Logit}\left[\pi_{uv}(t)\right] &= z(t) - \left\|x_u(t) - x_v(t)\right\|, \\ z(t) &\sim \text{GP}(\mu_z(t), K_z(t', t)) \\ x_u(t) &\sim \text{GP}(\mu_u(t), K_u(t', t)), \ u = 1, ..., N. \end{aligned}$$

Dynamic multi-layer networks

- **Dynamic multi-layer Bernoulli model**[2]

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

$$\text{Logit}\left[\pi_{uv}^{(k)}(t)\right] = z(t) - \|x_u(t) - x_v(t)\| - \left\|x_u^{(k)}(t) - x_v^{(k)}(t)\right\|,$$

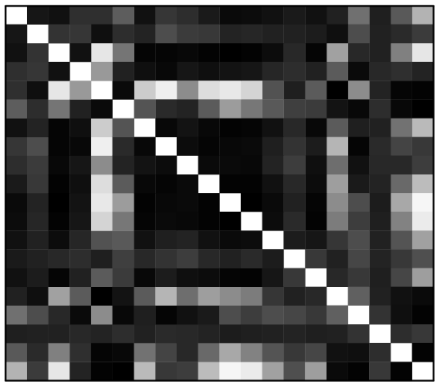
where

- **Global GP** across airport and airlines: $z(t)$
- **Airport-specific GPs**, but common across airlines: $x_u(t)$.
- **Airport/Airline-specific** GPs: $x_u^{(k)}(t)$.

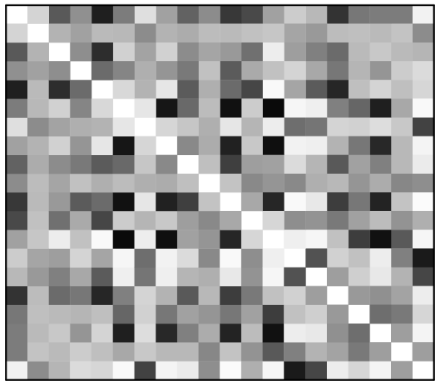
[2] Durante et al (2017). Bayesian learning of dynamic multilayer networks, JMLR.

Learning a dynamic multi-layer networks

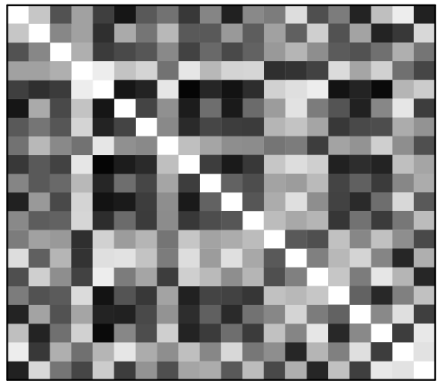
Sampled Link Probabilities at Layer 1, Time 1



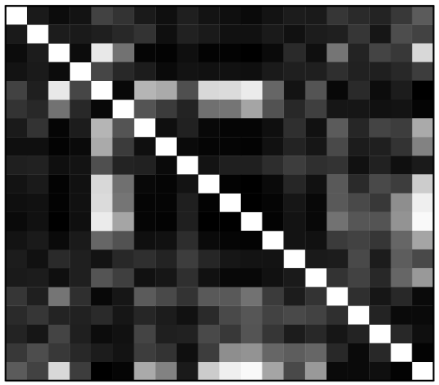
Sampled Link Probabilities at Layer 1, Time 10



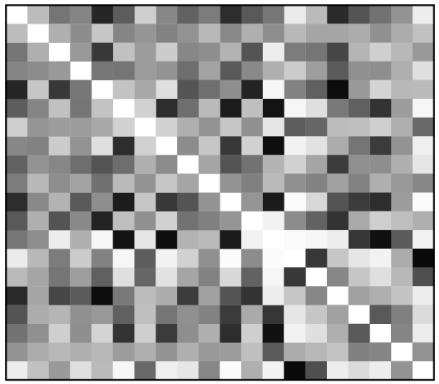
Sampled Link Probabilities at Layer 1, Time 22



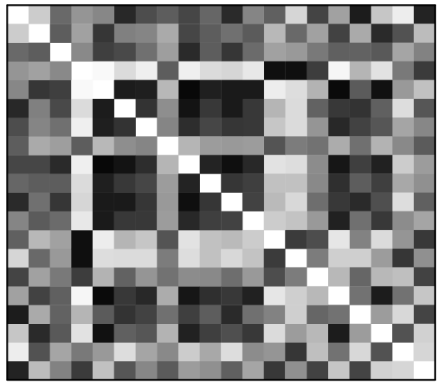
Estimated Link Probabilities at Layer 1, Time 1



Estimated Link Probabilities at Layer 1, Time 10



Estimated Link Probabilities at Layer 1, Time 22



Stochastic block multi-layer networks

- Number of GPs: $1 + KN(N-1)/2$ for N nodes and K layers.
- Airline prediction. N in thousands, K in hundreds ...
- **Stochastic block model**[3]

$$Y_{uv}^{(k)}(t) | s_u = a, s_v = b \sim \text{Bern}\left(\pi_{ab}^{(k)}(t)\right)$$

$$\text{Logit}\left[\pi_{ab}^{(k)}(t)\right] = z(t) - \|x_a(t) - x_b(t)\| - \left\|x_a^{(k)}(t) - x_b^{(k)}(t)\right\|,$$

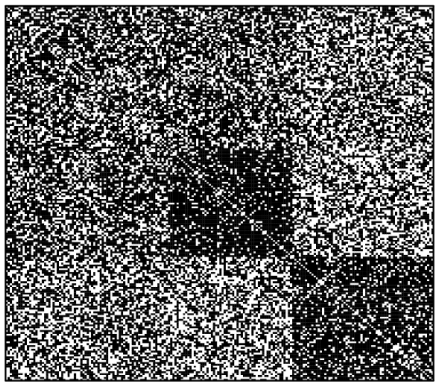
$$s_u \sim \text{Categorical}(\omega_1, ..., \omega_B)$$

- Number of GPs: $1 + KB(B+1)/2$ for B blocks. $B \ll N$.
- Need to learn the latent block indicators, s_u, for $u = 1, ..., N$.

[3] Ongoing work with Hector Rodriguez-Deniz at LiU.

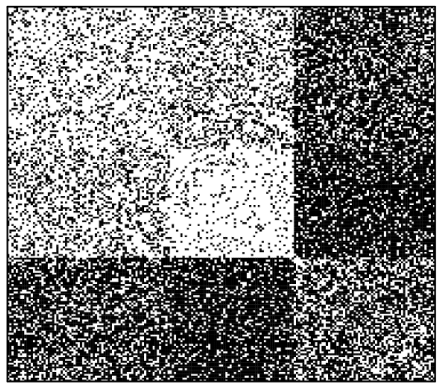
Block-structured multi-layer networks

Simulated Adjacency Matrix at Layer 1, Time 1

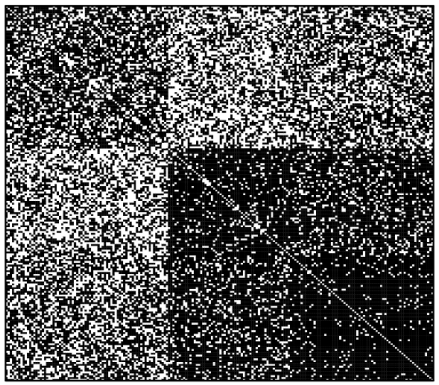


Simulated Adjacency Matrix at Layer 1, Time 9

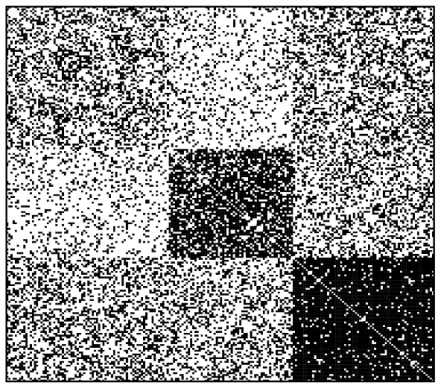
Simulated Adjacency Matrix at Layer 1, Time 18



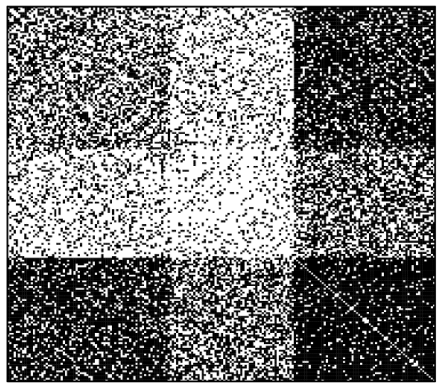
Simulated Adjacency Matrix at Layer 2, Time 1



Simulated Adjacency Matrix at Layer 2, Time 9

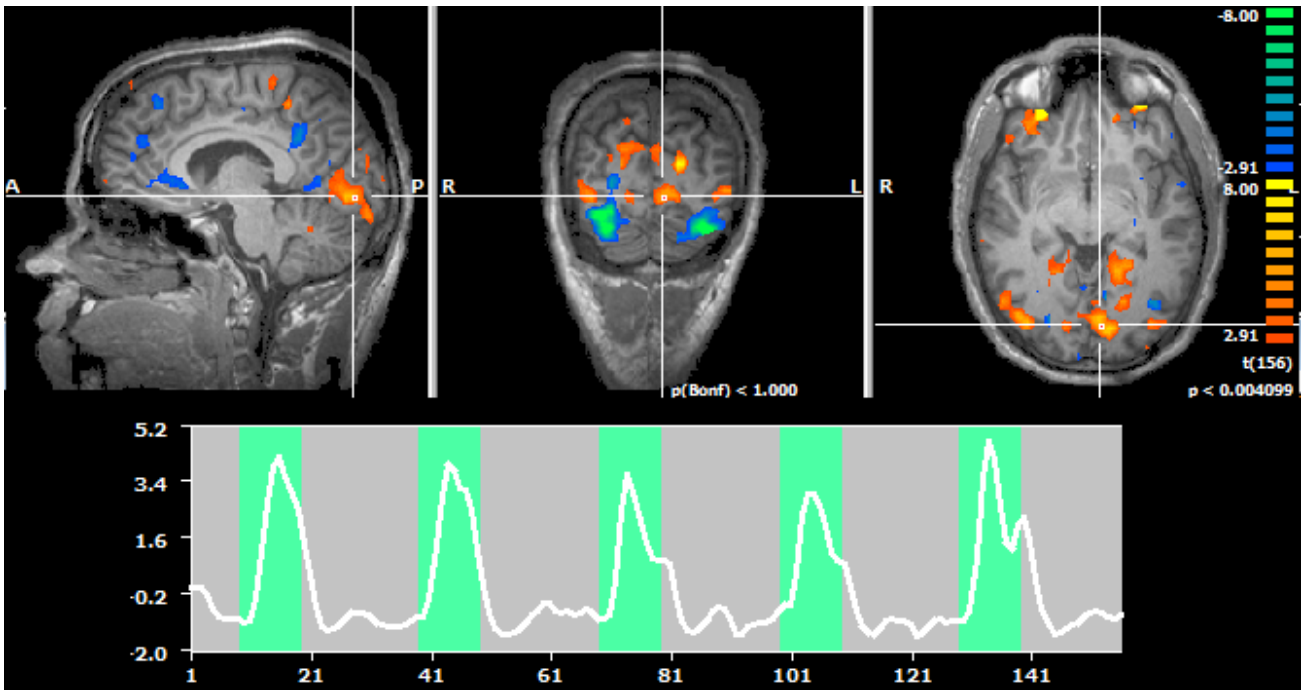


Simulated Adjacency Matrix at Layer 2, Time 18



fMRI - LOCALIZING ACTIVITY IN THE BRAIN

- **functional Magnetic Resonance Imaging**
- 3D images of brain activity over many time periods.
- 128 x 128 x 30 **voxels** observed over 180 time periods
- So 491,520 **spatially correlated time series**.
- Repeated on 10-30 individuals
- Which **brain regions** are **activated by stimulus** (e.g. pain)?

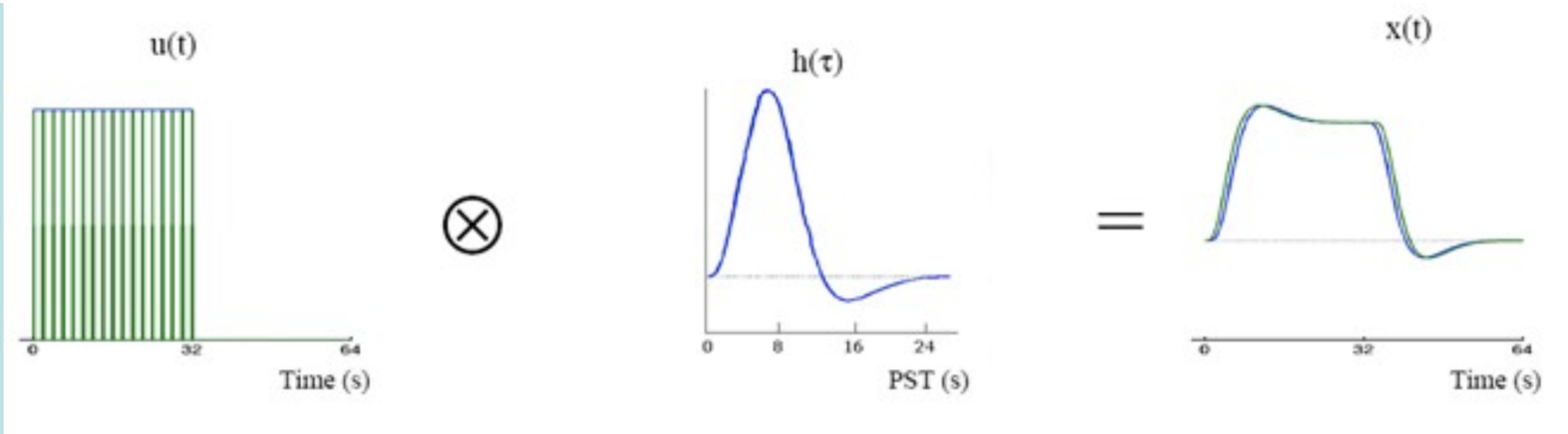


fMRI - LOCALIZING ACTIVITY IN THE BRAIN

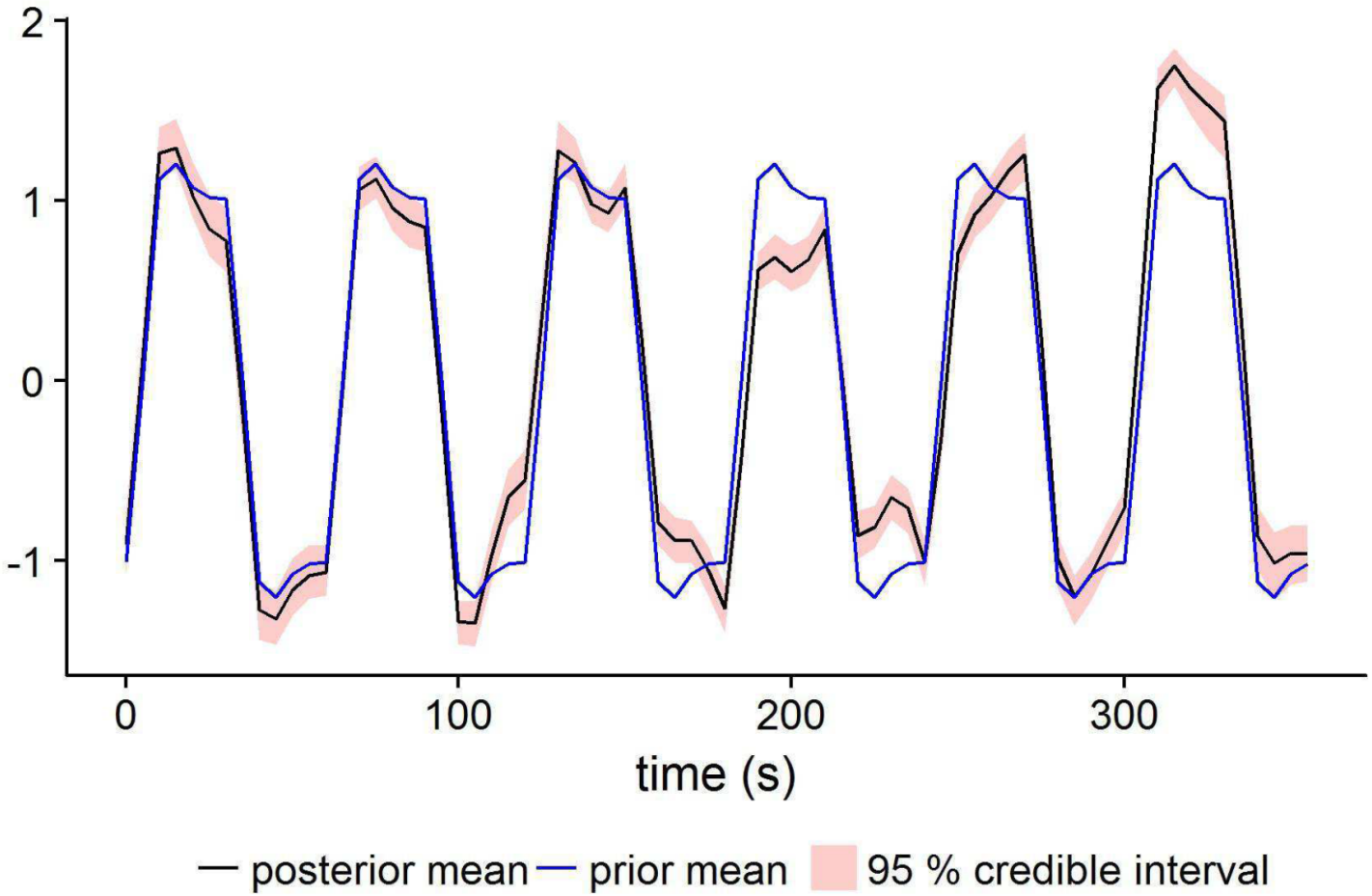
- fMRI **measures activity indirectly via blood flows** (**BOLD**).
- Mapping $\text{Neuronal activity} \rightarrow \text{blood flows}$ is sluggish and modulated by **hemodynamics response function** (**HRF**) $h(t)$.
- Stimulus $\otimes$ HRF = Predicted BOLD $= x_t$.
- Basic model for a voxel is linear regression on with AR noise

$$y_t = x_t\beta + \mathbf{z}_t^T\gamma + v_t, \quad v_t \sim \text{ AR process}$$

- Idea: replace x_t by a **latent GP** with prior mean based on **physiological model** for the hemodynamics.
- No assumption of convolution. No assumptions on the HRF.

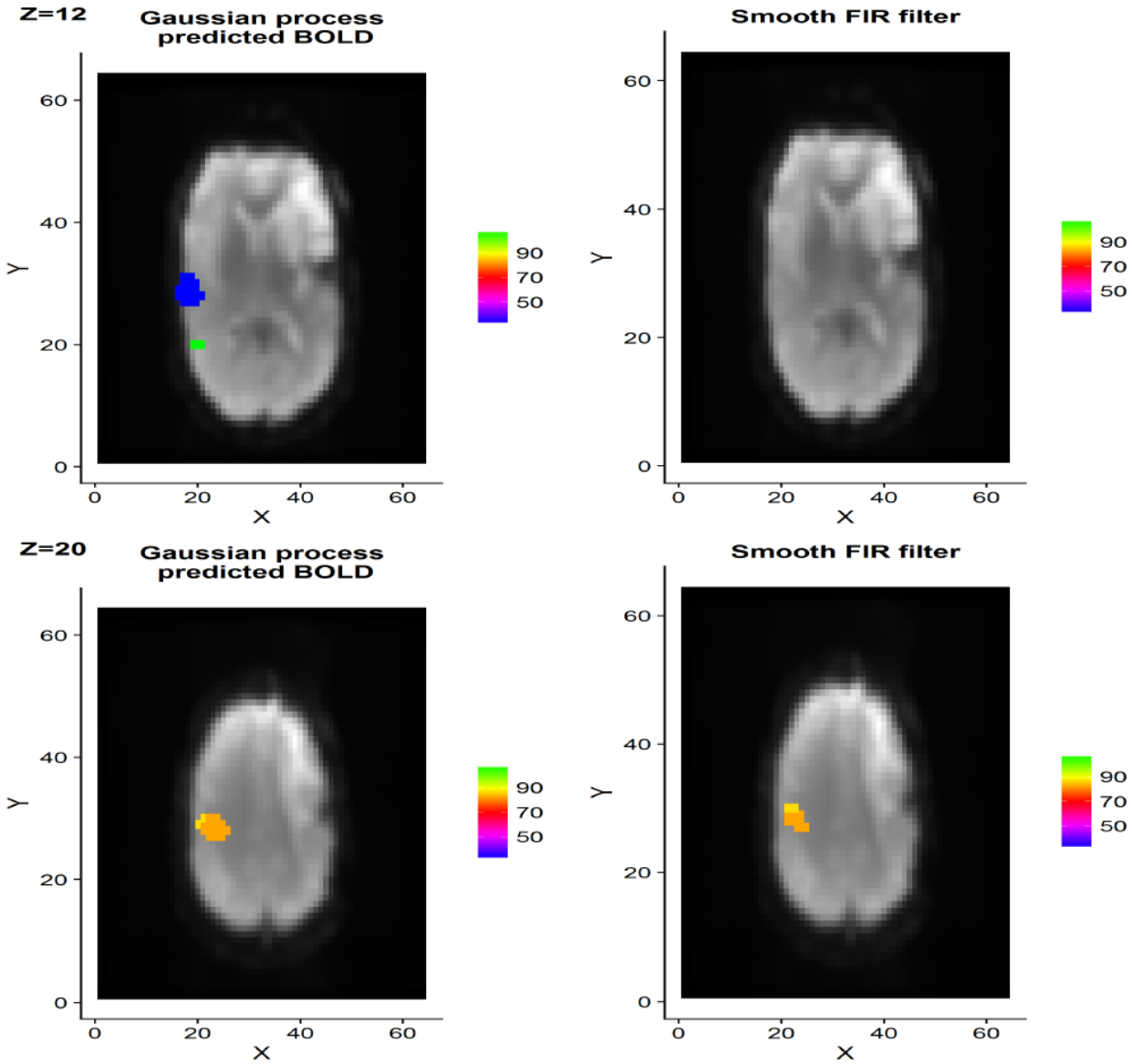


GP PREDICTED BOLD[4]



[4]Work by my PhD student Josef Wilzén and Anders Eklund. Physiological Gaussian Process Priors for the Hemodynamics in fMRI Analysis, arXiv.

GP PREDICTED BOLD DETECTS MISSED ACTIVATION?



Gaussian Process Optimization (GPO)

- Aim: **minimization of expensive function**

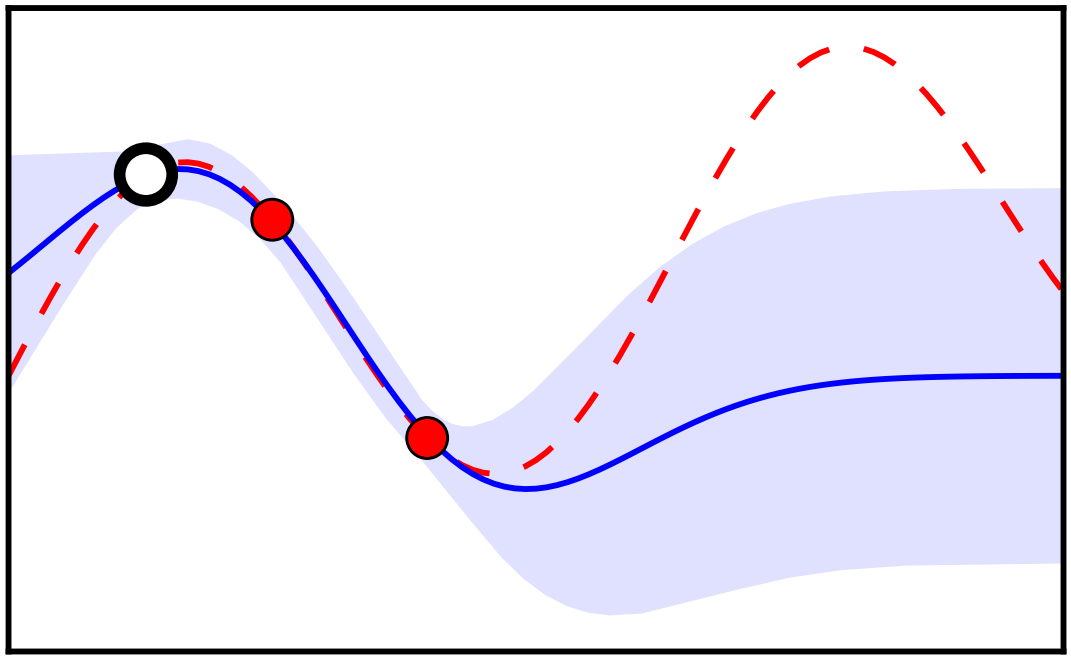
$$\text{argmin}_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

- Typical applications: **hyperparameter estimation**.

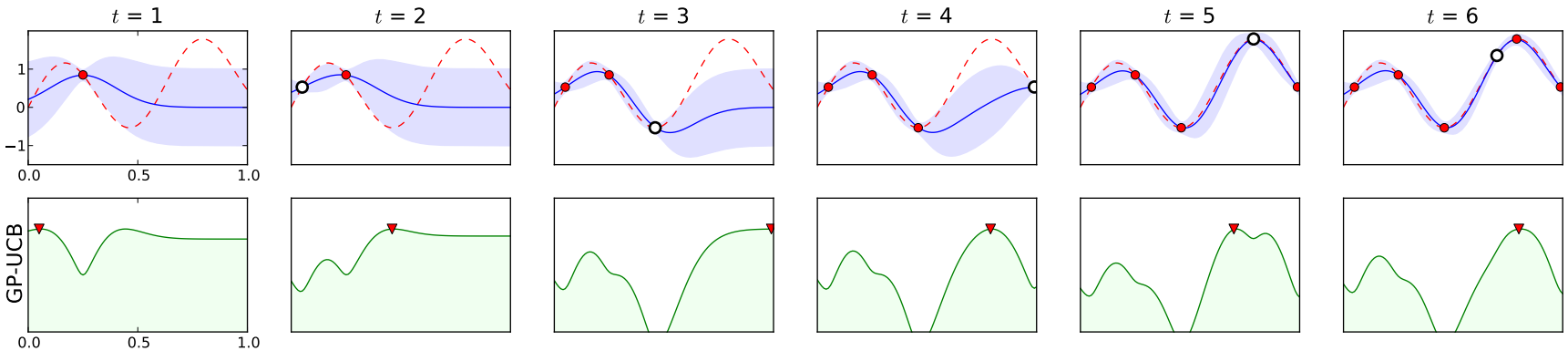
- **GPO idea**:
  - Assume $f \sim \text{GP}$
  - Evaluate f at $x_1, x_2, ..., x_n$.
  - Update to posterior $f|x_1, ..., x_n \sim GP(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*))$.
  - Use posterior of f to find a new x_{n+1}. **Explore** vs **Exploit**.
  - Iterate until convergence.

- **Bayesian Optimization**. Probabilistic numerics.

Explore-Exploit Illustration



Acquisition functions from Brochu et al



Acquisition functions

- **Probability of Improvement** (**PI**)

$$a_{PI}\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right) \equiv \Pr\left(f(\mathbf{x}) < f(\mathbf{x}_{best})\right) = \Phi(\gamma(\mathbf{x}))$$

where

$$\gamma(\mathbf{x}) = \frac{f(\mathbf{x}_{best}) - \mu\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right)}{\sigma\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right)}$$

- **Expected Improvement** (**EI**)

$$a_{EI}\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right) = \sigma\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right)\left[\gamma(\mathbf{x})\Phi(\gamma(\mathbf{x})) + \mathcal{N}\left(\gamma(\mathbf{x}); 0, 1\right)\right]$$

- **Lower Confidence Bound** (**LCB**)

$$a_{EI}\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right) = \mu\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right) - \kappa \cdot \sigma\left(\mathbf{x}; \{\mathbf{x}_n, y_n\}, \theta\right)$$

- Need to maximize the acquisition function to choose $\mathbf{x}_{next}$. Non-convex, but cheaper and simpler than original problem.

ConvNets - Snoek et al (NIPS, 2012)