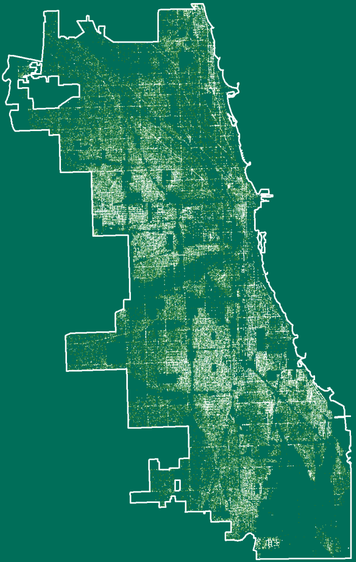


Deep generative modelling aiding GPs and spatial statistics and MCMC (in three chapters)

Elizaveta Semenova
Department of Computer Science
University of Oxford
www.elizaveta-semenova.com

GPSS 2023, Manchester

MACHINE LEARNING & GLOBAL HEALTH NETWORK

UNIVERSITY OF OXFORD

Outline

Introduction: spatial statistics

PriorVAE: encoding random vectors

aggVAE: encoding GP aggregates

PriorCVAE: can we infer hyperparameters?

Introduction: Spatial statistics

Disease mapping and public health

- A map of a three-stage containment field in Italy, 1691

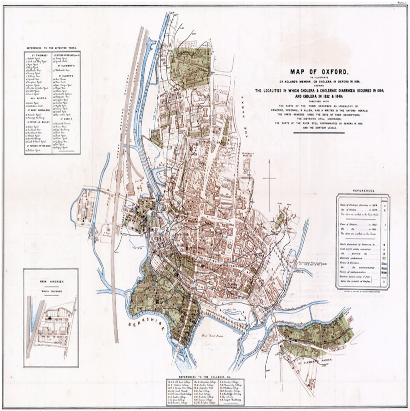
"Disease mapping and innovation: A history from wood-block prints to Web 3.0", Tom Koch (2022)

The map that changed how we fight outbreaks

- Dr. John Snow mapped cholera cases in London, 1854.

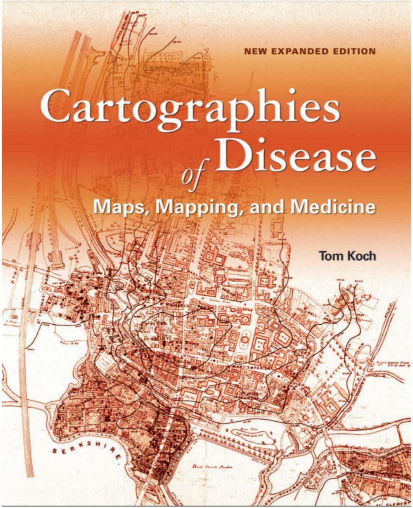
'On the Mode of Communication of Cholera', Second Edition, John Snow (1855c)

Disease mapping and public health



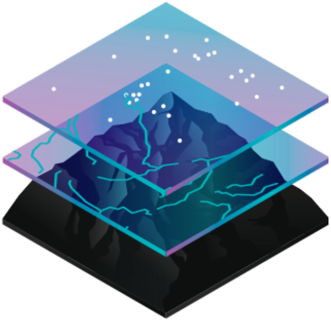
'Memoir on the cholera at Oxford, in the year 1854 : with considerations suggested by the epidemic', Acland (1856)

Disease mapping and technology



"Cartographies of Disease: Maps, Mapping, and Medicine", Tom Koch (2017)

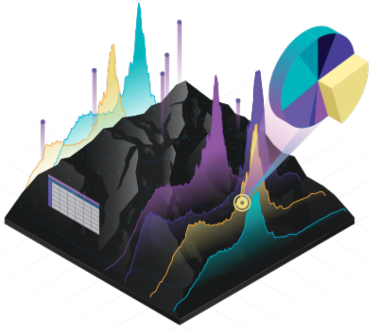
Modern technology for disease mapping



Data

geo-tagged

spatiotemporal



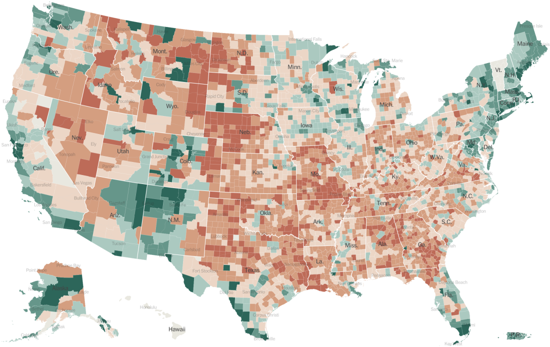
Methods

Bayesian inference + spatial statistics

deep learning

Image Credit: ESRI

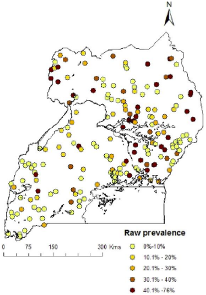
Areal data



US vaccinations at county level.

Credit: The New York Times

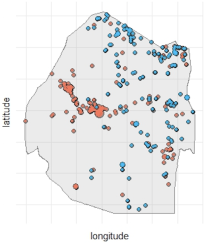
Geostatistical data



Observed malaria prevalence at
survey locations in Uganda.

Credit: J Ssempiira

Point pattern data



Observed local (blue) and imported (red) malaria cases in Eswatini, 2015.

Credit: E Semenova

Methods: classical approach

- Hierarchical Bayesian modelling using Gaussian Processes.

Methods: latent Gaussian models

$y = (y_1, ..., y_n)$	- outcome data over a set of n locations
$y \sim p(y \mid g^{-1}(\eta), \theta)$	- observational model (likelihood)
$\eta = X\beta + f$	- additive model for the mean, combines a fixed effects and random effect terms
$\underline{f \sim p(f \mid \theta)}$	- <u>random effect term: Gaussian process</u>
$\theta \sim p(\theta)$	- hyperparameters

Methods: Bayesian inference

- y - data, θ - parameters,

$$\underbrace{p(\theta|y)}_{\text{posterior}} \propto \underbrace{p(y|\theta)}_{\text{likelihood}} \underbrace{p(\theta)}_{\text{prior}}$$

- Gold standard inference algorithms: Markov chain Monte Carlo (MCMC) - theoretical guarantees; diagnostic tools
- Probabilistic programming languages: Stan, PyMC3, Numpyro, Turing.jl

Stan PyMC Pyro Turing.jl

Probabilistic programming languages (PPLs)

- PPLs allow users to specify probabilistic models and perform inference automatically.
- Users need to specify
  1. prior
  2. likelihood
- Inference is performed by an MCMC algorithm (Gibbs, Metropolis-Hastings, HMC) or Variational Inference

PPLs and software choices

Stan, PyMC	require manual reimplementation of NNs
Pyro + PyTorch	no manual implementation required, but slow
Numpyro + JAX	no manual implementation required, and fast
Turing.jl + Flux.jl	

Analyzing MCMC outputs

- Diagnostics for MCMC samples
  - Gelman-Rubin statistic ($\hat{R}$)
  - Effective sample size (ESS) per second

Methods: Gaussian Processes

- **Definition:** a Gaussian Process (GP) is random function f on a set X such that for any $x_1, \ldots, x_n \in X$, the vector $f_{\text{GP}} = [f(x_1), \ldots, f(x_n)]^T$ is multivariate Gaussian.

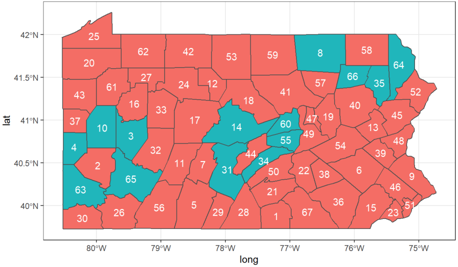
- GPs are characterised by
  - a *mean* function $m(x) = \mathbb{E}(f(x))$,
  - a *kernel (covariance)* function $k(x, x') = Cov(f(x), f(x'))$, e.g.

$$k(x_i, x_j) = \tau \exp\left(-\frac{||x_i - x_j||^2}{2l^2}\right).$$

- Notation: $f \sim \text{GP}(m, k)$.

Modelling areal data

- State-of-the-art models rely on "borrowing strength" from neighbours and use hierarchical Bayesian models to do so



Neighbors of areas 2, 44 and 58 of Pennsylvania.

Credit: Moraga, "Geospatial Health Data: Modelling and Visualization with R-INLA and Shiny"

Models of areal data

$f \sim \text{MVN}(0, Q^{-1})$ | Q - precision matrix

$Q = \tau I$ | i.i.d.

$Q = \tau(D - \alpha A)$ | CAR: A and D are defined by the neighbourhood structure

$Q = \tau(D - A)$ | ICAR

$Q^{-1} = \tau_1^{-1} I + \tau_2^{-1}(D - A)^{-}$ | BYM

Modelling point pattern data

Log-Gaussian Cox process:

$$
\begin{aligned}
L(s_1, \ldots, s_n; \lambda(s)) &= \exp(-\lambda(D)) \prod_{i=1}^{n} \lambda(s_i), \\
\lambda(D) &= \int_D \lambda(s) ds, \\
\lambda(s) &= \exp(X^T(s)\beta + f(s)), \\
f &\sim \mathrm{GP}(0, k).
\end{aligned}
$$

Computational bottleneck

- Gaussian Processes scale as $O(n^3)$.
- Bayesian inference with MCMC requires $O(n^3)$ calculations for each draw from the posterior.

PriorVAE: encoding random vectors

Goal

$$\underbrace{g(E[y|f_{\mathrm{GP}}])}_{\text{Mean}} = \underbrace{X\beta}_{\text{fixed effect}} + \underbrace{f_{\mathrm{GP}}}_{\text{random effect}}$$

PriorVAE philosophy

$$g(E[y|f_{\mathrm{GP}}]) = X\beta + f_{\mathrm{GP}}$$

- Replace costly evaluation of f_{GP} at inference stage with a cheap approximation learned with deep generative modelling.

Idea: train VAE on GP prior draws

$$g(E[y|f_{\mathrm{GP}}]) = X\beta + f_{\mathrm{VAE}}$$

- Substitute evaluation of the GP with the decoder of a trained variational autoencoder (VAE).

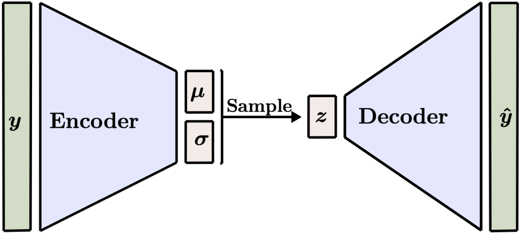
Idea: train VAE on GP priors

- Decoder of a trained variational autoencoder (VAE):

$$\mathrm{ELBO}_{\mathrm{VAE}} = \mathbb{E}_{q(z|y)}\left[\log p(y|z)\right] - KL\left[q(z|y)||p(z)\right],$$

$$p(z) \sim N(0, I)$$

Variational Autoencoder



PriorVAE workflow

- Fix the set of observation locations (i.e. spatial structure or temporal labels),
- Use draws from a GP prior f_{GP} over the observation locations as training data for a VAE,
- Use the trained decoder $\phi_w(.)$ as a drop-in replacement for the GP in the model used for inference.

Pseudocode[1]

```
def decoder_numpy(z, W1, B1, W2, B2):
    def linear(z, W, B):
        lin_out = jnp.matmul(z, W) + B
        return lin_out

    return linear(jax.nn.relu(linear(z, W1, B1)), W2, B2)

def numpyro_model(z_dim, y):
    z = numpyro.sample("z",
      npdist.Normal(jnp.zeros(z_dim), jnp.ones(z_dim)))

    f = numpyro.deterministic("f",
      decoder_numpy(z, W1, B1, W2, B2))
    sigma = numpyro.sample("sigma", npdist.HalfNormal(1))

    y = numpyro.sample("y", npdist.Normal(f, sigma),
      obs=y)
```

[1] colab demo: https://tinyurl.com/priorcvae

Why does it work?

$z_n \sim N(0, I_n)$

$f_{\text{GP}} = L_\theta z_n$

Linear operation, but θ needs to be inferred.

Complexity: $O(n^3)$.

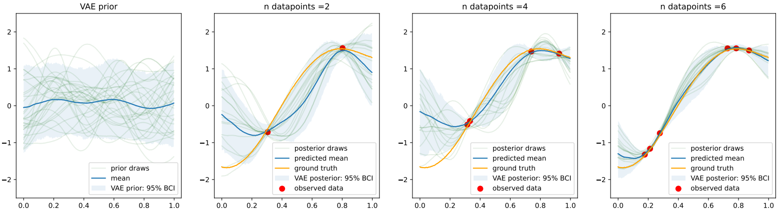
$z_d \sim N(0, I_d), \quad d < n$

$f_{\text{VAE}} = \phi_w(z_d)$

Non-linear operation, but deterministic transformation.

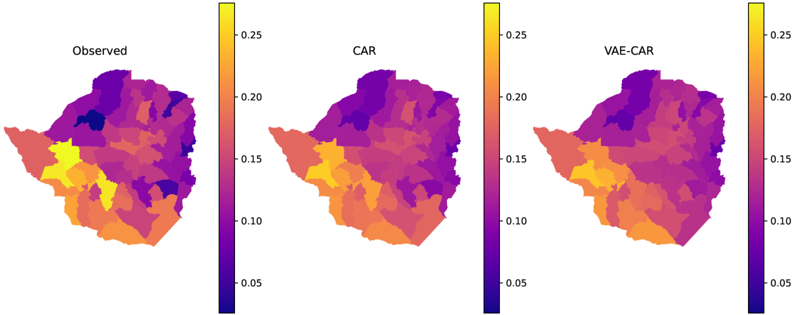
Complexity: $O(dn)$.

PriorVAE: one-dimensional GP inference



Making inference using the learned prior on a regular grid, $n = 400$

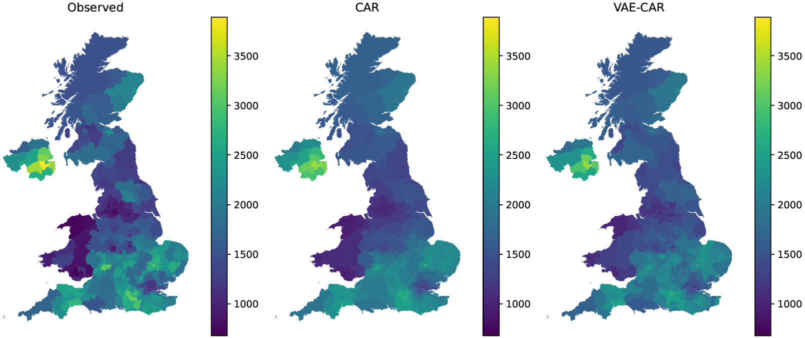
PriorVAE: HIV prevalence in Zimbabwe



70x speedup (ESS per second):

Model	Effective sample size (ESS)	Elapsed time, s	ESS per second
CAR	120	13	9
VAE-CAR	2600	4	650

PriorVAE: projected COVID-19 incidence in the UK



350x speedup (ESS per second):

Model	Effective sample size (ESS)	Elapsed time, s	ESS per second
CAR	317	277	1.14
VAE-CAR	3188	8	398

PriorVAE: Discussion

Advantages:

- Fast inference because of uncorrelated parameters in low dimensional space
- No need to retain training data
- Can be utilized for a variety of problems like time-series data, fixed spatial data
- Very efficient MCMC inference

Disadvantages:

PriorVAE: Discussion

Advantages:

- Fast inference because of uncorrelated parameters in low dimensional space
- No need to retain training data
- Can be utilized for a variety of problems like time-series data, fixed spatial data
- Very efficient MCMC inference

Disadvantages:

- Output is not conditioned on the input
- Input locations needs to be fixed for all prior training functions

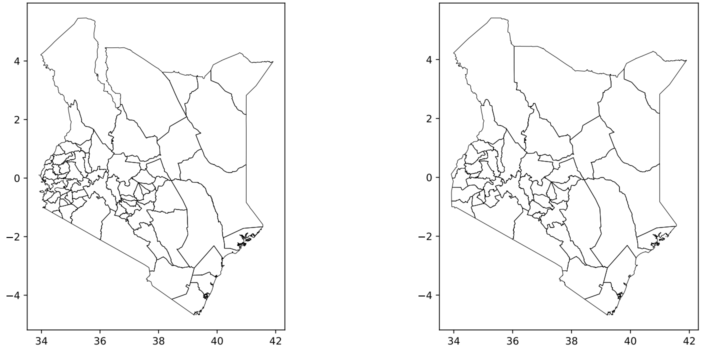
Encodes random vectors, not random functions.

Source code:

PriorVAE `https://github.com/elizavetasemenova/PriorVAE`

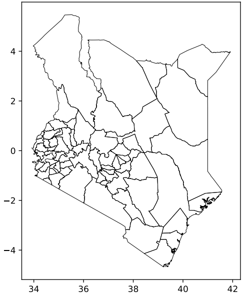
aggVAE: encoding GP aggregates and change-of-support problem

Kenya: boundaries before and after 2010



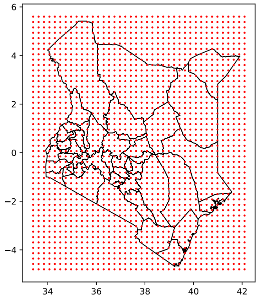
aggVAE: what are we solving?

- Adjacency-based models assume heterogeneity.
- Changing boundaries: change-of-support.



Computational grid

- Create fine spatial grid $\{g_1, ...g_n\}$ over the domain of interest:

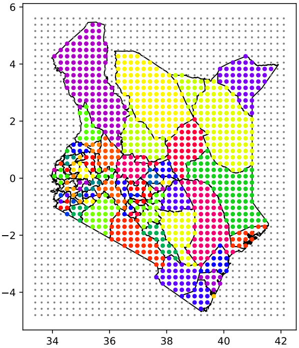


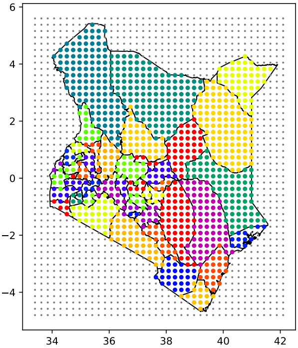
Computational grid

- Draw GP evaluations over the grid:

$$
\begin{aligned}
f &= \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} \sim \mathrm{MVN}(0, \Sigma), \\
f_j &= f(g_j), \\
\Sigma_{jk} &= \sigma^2 \exp\left(-\frac{d_{jk}^2}{2l^2}\right), \\
d_{jk} &= ||g_j - g_k||
\end{aligned}
$$

Attribution of grid points over polygons





Computing GP aggregates over polygons

For each district (polygon) $p_i, i = 1, ..., K$, compute

$$f_{\text{aggGP}}^{p_i} = \int_{p_i} f(s)ds \approx c \sum_{g_j \in p_i} f_j = c\bar{f}_{\text{aggGP}}^{p_i}.$$

Spatial random effect:

$$f_{\text{aggGP}} = \begin{pmatrix} f_{\text{aggGP}}^{p_1} \\ \vdots \\ f_{\text{aggGP}}^{p_K} \end{pmatrix} = Mf \in \mathbb{R}^K,$$

$$M : \quad m_{ij} = I_{\{g_j \subset p_i\}}.$$

Joint encoding of priors

To tackle the the change-of-support problem, encode $\bar{f}^{\text{old}}_{\text{aggGP}}$ and $\bar{f}^{\text{new}}_{\text{aggGP}}$ jointly:

$$\bar{f}^{\text{joint}}_{\text{aggGP}} = \begin{pmatrix} \bar{f}^{p_1^{\text{old}}}_{\text{aggGP}} \\ \cdots \\ \bar{f}^{p_{K_1}^{\text{old}}}_{\text{aggGP}} \\ - - -- \\ \bar{f}^{p_1^{\text{new}}}_{\text{aggGP}} \\ \bar{f}^{p_{K_2}^{\text{new}}}_{\text{aggGP}} \end{pmatrix} = \begin{pmatrix} M^{\text{old}} f \\ M^{\text{new}} f \end{pmatrix} \in \mathbb{R}^{K_1+K_2}.$$

'aggVAE' workflow

- Fix spatial structure of areal units as a collection of polygons $P = \{p_1, \ldots, p_k\}$.
- Create an aritificial computational grid of sufficient granularity $G = \{g_1, \ldots, g_n\}$.
- Pre-compute the matrix of indicators $M, \quad m_{ij} = I_{\{g_j \subset p_i\}}$.
- Draw GP evaluations over G using a selected kernel $k(.,.)$: $f = (f_1, \ldots f_n)^T$.
- Compute GP aggregates at the level of $P : f_{\text{aggGP}} = cMf$
- Train PriorVAE on f_{aggGP} draws to obtain f_{aggVAE} priors.
- Use f_{aggVAE} at inference stage within MCMC.

Mapping malaria prevalence in Kenya

- **Model** Malaria prevalence $\theta_i, i \in 1, \ldots K$ is inferred using the Negative Binomial distribution

$$\begin{cases} n_i^{\text{pos}} & \sim \text{NegBin}(n_i^{\text{tests}}, \theta_i), \\ \text{logit}(\theta_i) & = b_0 + f_{\text{aggGP}}^{p_i}. \end{cases}$$

  where n_i^{tests} and n_i^{pos} are the number of total and positive RDT tests, correspondingly.

- **Inference.** Perform MCMC inference using f_{aggVAE} instead of f_{aggGP}.

Results

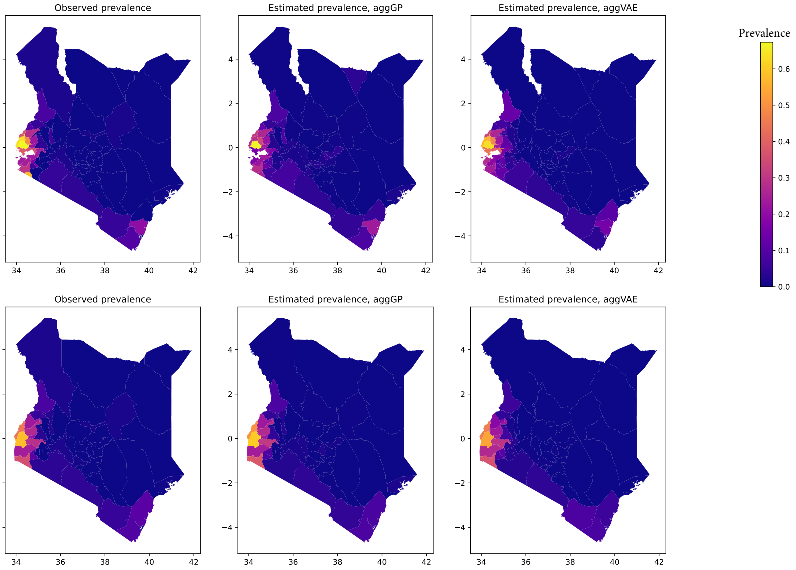
Comparison of MCMC for models with f_{aggGP} and f_{aggVAE} using 200 warm-up steps and 1000 iterations:

Model of the spatial random effect	Elapsed time	Average effective sample size of the random effects
aggGP	**15h***	129
aggVAE	**5s**	231

Table: Model comparison.

* aggGP model has not converged: $\hat{R} = 1.4$.

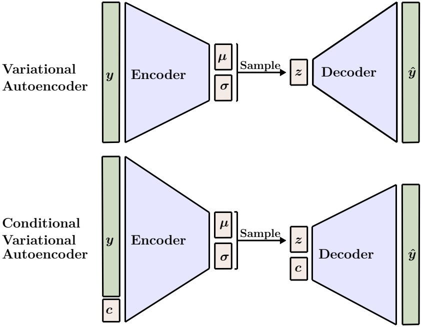
Results



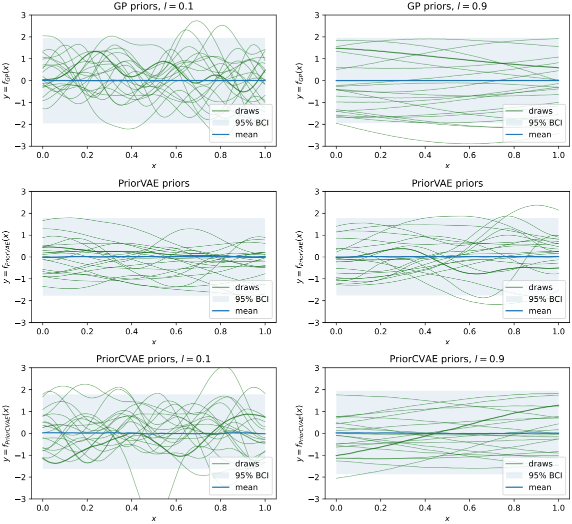
Observed prevalence
Estimated prevalence, aggGP
Estimated prevalence, aggVAE
Observed prevalence
Estimated prevalence, aggGP
Estimated prevalence, aggVAE
Prevalence
0.0
0.1
0.2
0.3
0.4
0.5
0.6

Can we infer hyperparameters? PrioCVAE!

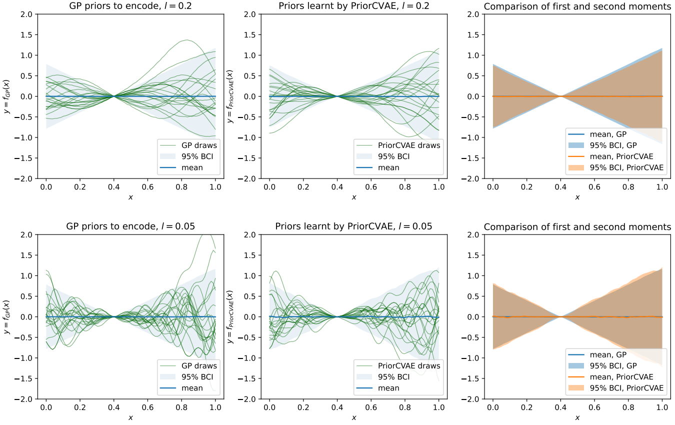
PriorCVAE: use hyperparameter(s) as condition c



PriorCVAE: lengthscale as a condition $c = l$

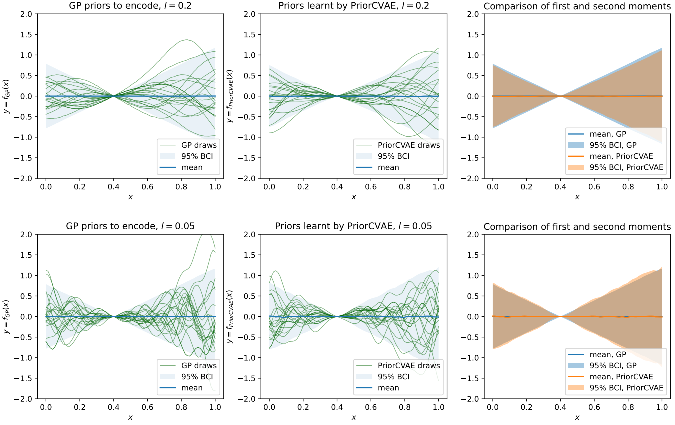


PriorCVAE: non-stationary kernels



PriorCVAE trained on hyperpriors $l \sim \mathcal{U}(0.01, 0.4)$

PriorCVAE: extrapolation w.r.t. hyperparameters



Extrapolating away from $l \in (0.01, 0.4)$

NUTS, Laplace, ADVI - any luck?

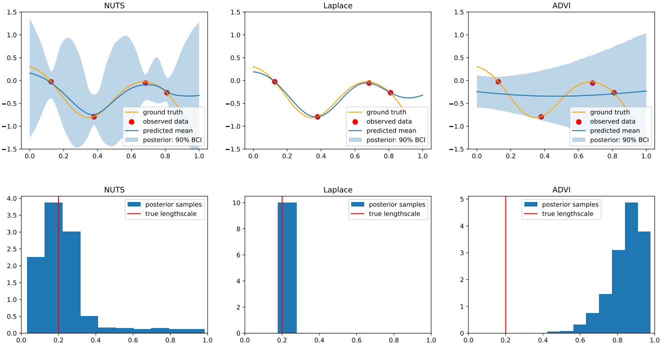
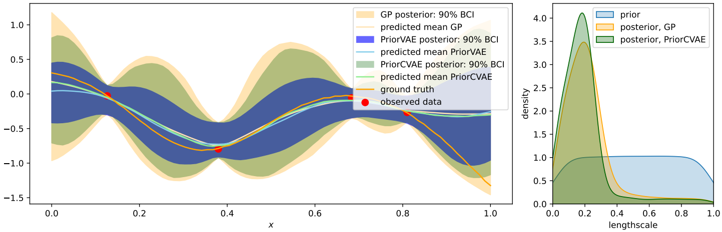


Figure: Top: inferred mean and 90% BCI, bottom: inferred lengthscale.

GP, PriorVAE, PriorCVAE



10K x speedup (ESS per second):

Model	Effective sample size (ESS)	Elapsed time, s	ESS per second
PriorVAE	31115	8	3889
PriorCVAE	34725	17	**2043**
GP	1496	7150	**0.2**

Deep generative modelling for MCMC

Elizaveta Semenova, Yidan Xu, Adam Howes, Theo Rashid, Samir Bhatt, Swapnil Mishra, and Seth Flaxman.
PriorVAE: encoding spatial priors with variational autoencoders for small-area estimation.
Journal of the Royal Society Interface, 19(191):20220094, 2022.

Elizaveta Semenova, Swapnil Mishra, Samir Bhatt, Seth Flaxman, and H Juliette T Unwin.
Deep learning and MCMC with aggVAE for shifting administrative boundaries: mapping malaria prevalence in Kenya.
UAI 2023 workshop "Epistemic Uncertainty in Artificial Intelligence", 2023.

Elizaveta Semenova, Max Cairney-Leeming, and Seth Flaxman.
PriorCVAE: scalable MCMC parameter inference with Bayesian deep generative modelling.
arXiv preprint arXiv:2304.04307, 2023.

Future work

- improve quality of samples

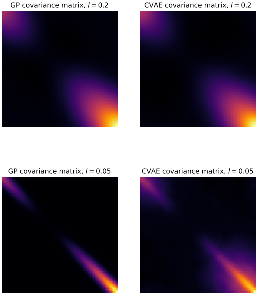


Figure: Empirical covariance matrices

- applications: population genetics, spatial weather extremes
- geometry: sphere, graphs

Related work

- πVAE method
  - Mishra et al, 2022, *Statistics and Computing*
  - it actually existed before PriorVAE
- An application of PriorVAE to Hawkes process
  - Miscouridou et al, 2022, *TMLR*
  - uses PriorVAE to make GP calculations feasible

Code

- **PriorVAE**, JAX (but ugly)
  GitHub: `https://github.com/elizavetasemenova/PriorVAE`
  Colab: `https://tinyurl.com/PriorVAE`
- **PriorCVAE**, PyTorch (manually implement NN for Numpyro)
  GitHub: `http://github.com/elizavetasemenova/PriorCVAE`
  Colab: `https://tinyurl.com/PriorCVAE`
- **PriorCVAE**, JAX (seemless NN and Numpyro integration)
  GitHub: `https://github.com/MLGlobalHealth/PriorCVAE`

Collaborators

Machine Learning & Global Health (MLGH) network

MACHINE LEARNING
& GLOBAL HEALTH NETWORK

- Elizaveta Semenova, Seth Flaxman, Max Cairney-Leeming (University of Oxford)
- Adam Howes, Theo Rashid, Bob Verity (Imperial College London)
- Juliette Unwin (University of Bristol)
- Prakhar Veema (Aalto University)
- Swapnil Mishra (National University of Singapore)
- Samir Bhatt (University of Copenhagen/Imperial College)

Thank you!

- www.elizaveta-semenova.com