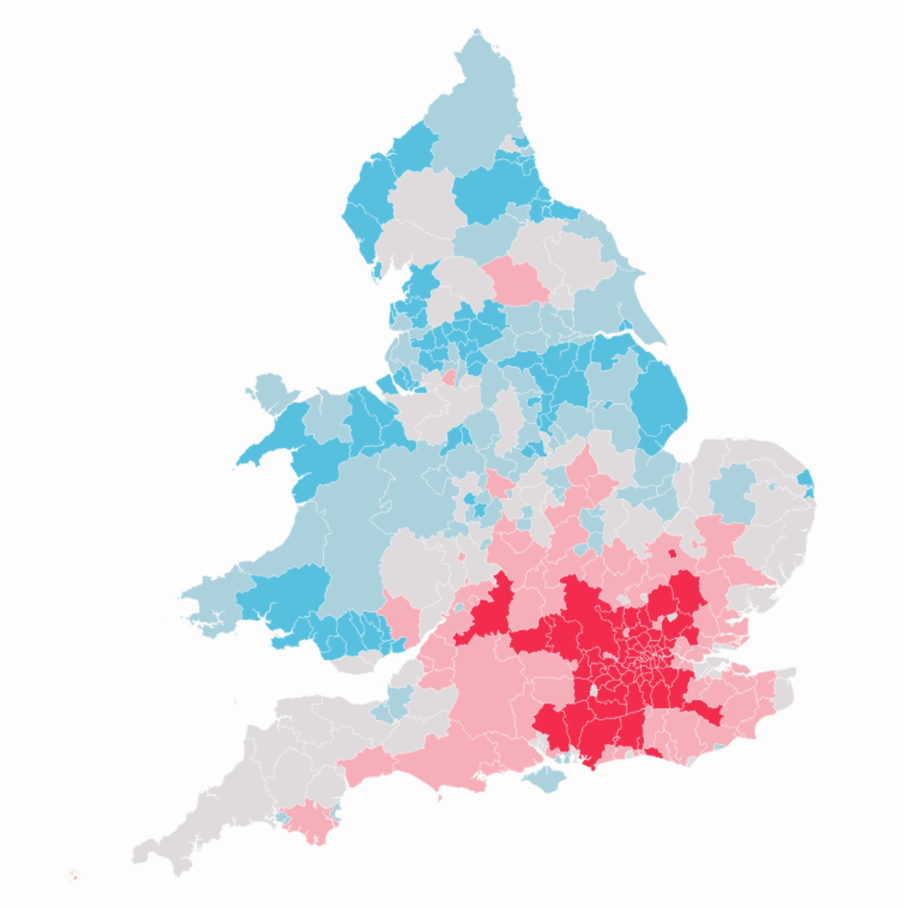
VNNGP: Variational Nearest Neighbor Gaussian Process

Luhuan Wu , Geoff Pleiss, John Cunningham

Columbia University

Problem setup

Motivating example: modeling UK housing price



Latent GP

$$f \sim N(f \mid 0, K_{ff})$$

Observations

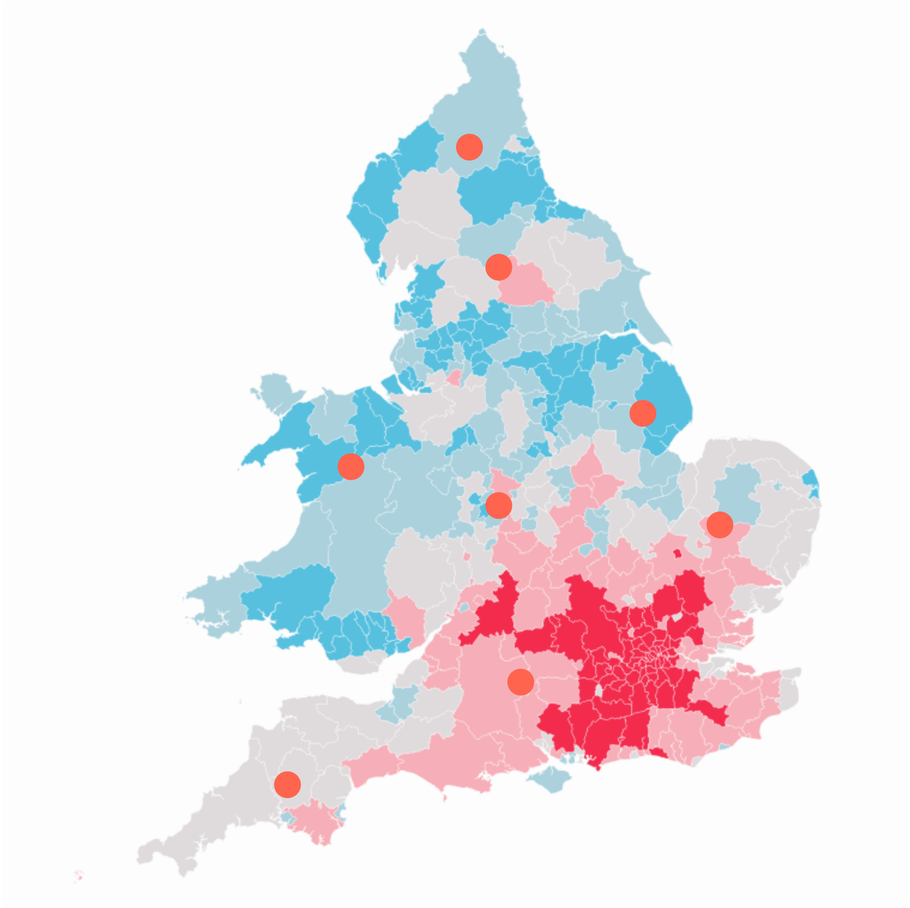
$$y \sim \prod_{i=1}^{N} p(y_i \mid f_i)$$

Time complexity: $O(N^3)$

*image source: https://twitter.com/undertheraedar/status/1388144547268530176

Problem setup

Motivating example: modeling UK housing price



SVGP: augment with $M \ll N$ inducing points

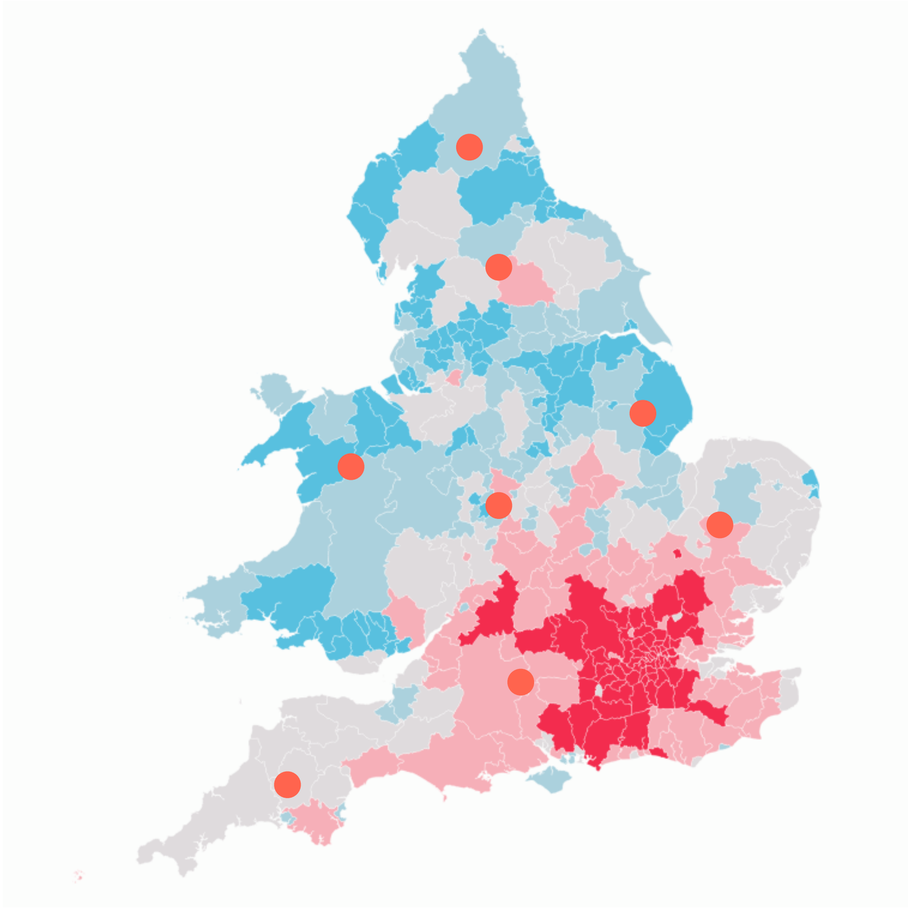
$$\begin{pmatrix} f \\ u \end{pmatrix} \sim N(0, \begin{bmatrix} K_{ff} & K_{fu} \\ K_{uf} & K_{uu} \end{bmatrix})$$

*image source: https://twitter.com/undertheraedar/status/1388144547268530176

*SVGP: Gaussian process for big data. Hensman et al. 2013.

Problem setup

Motivating example: modeling UK housing price



SVGP: augment with $M \ll N$ inducing points

$$\begin{pmatrix} f \\ u \end{pmatrix} \sim N(0, \begin{bmatrix} K_{ff} & K_{fu} \\ K_{uf} & K_{uu} \end{bmatrix})$$

Stochastic variational inference

$$ELBO_{SVGP} = \sum_{i=1}^{N} \mathbb{E}_{p(f_i|u)q(u)} \left[\log p(y_i|f_i)\right] - KL(q(u)\|p(u))$$

Complexity: $O(N^3) \rightarrow O(M^3)$

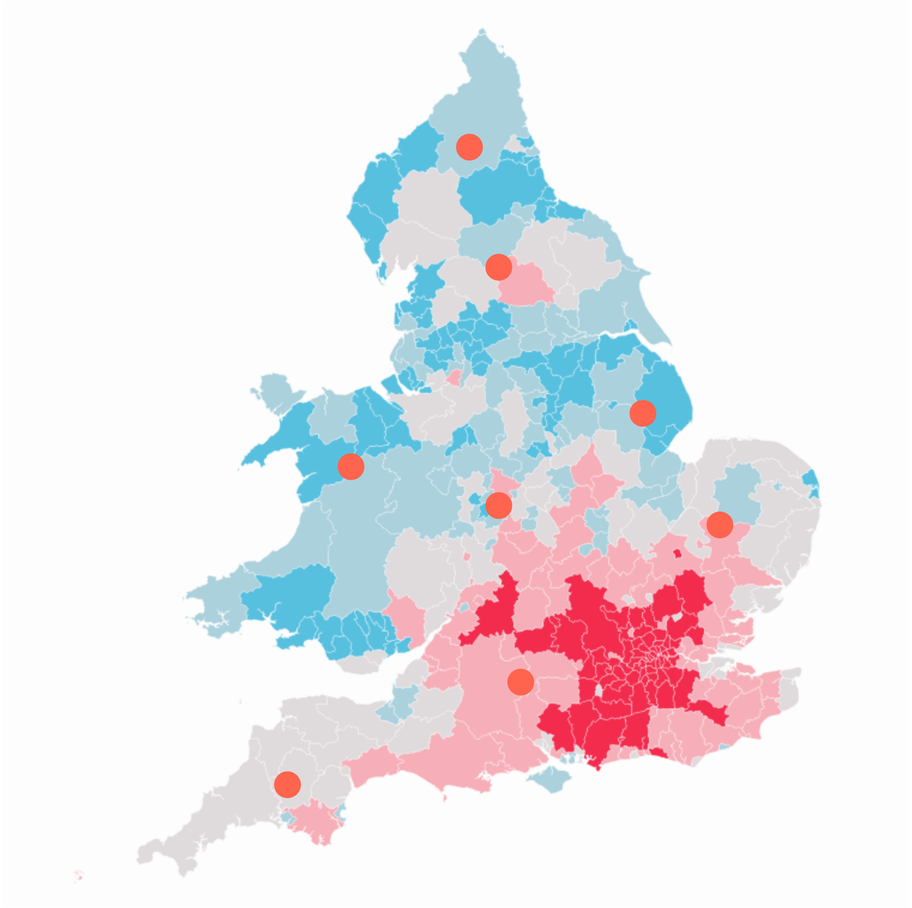
Flexible likelihood models

*image source: https://twitter.com/undertheraedar/status/1388144547268530176

*SVGP: Gaussian process for big data. Hensman et al. 2013.

Problem setup

Motivating example: modeling UK housing price



SVGP: augment with $M \ll N$ inducing points

$$\begin{pmatrix} f \\ u \end{pmatrix} \sim N(0, \begin{bmatrix} K_{ff} & K_{fu} \\ K_{uf} & K_{uu} \end{bmatrix})$$

Bottleneck:

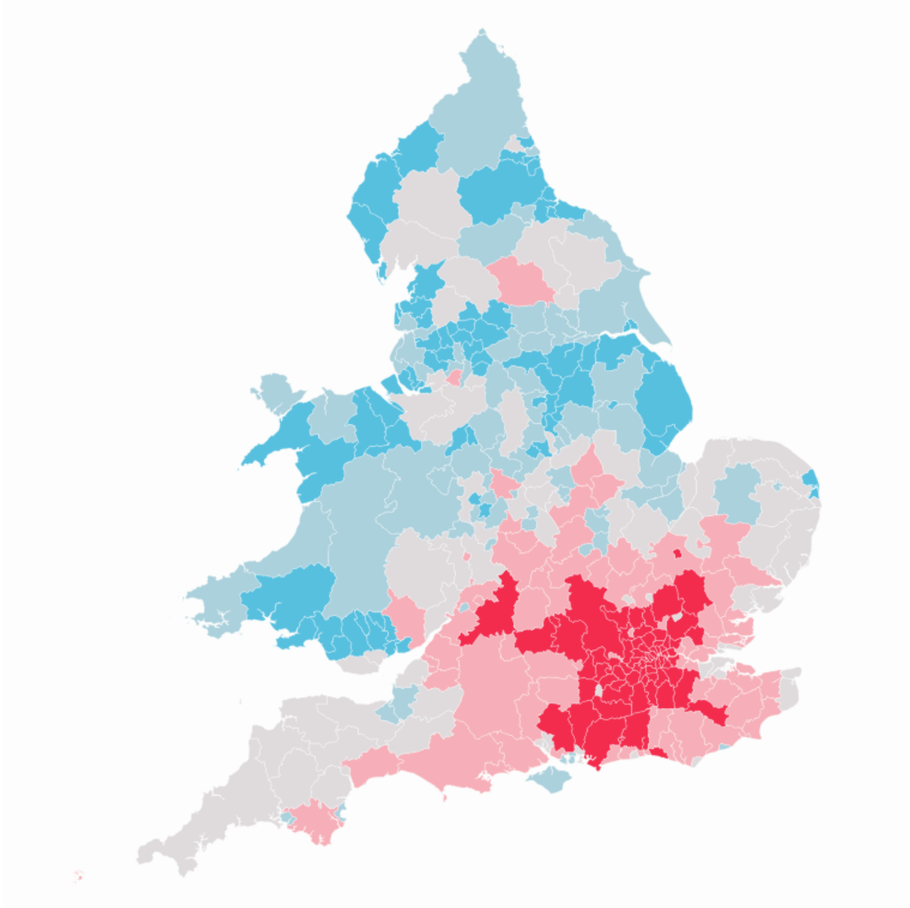
Since $M \ll N$, SVGP may result in **poor approximation quality** for large-scale data that is not inherently low-rank

*image source: https://twitter.com/undertheraedar/status/1388144547268530176

*SVGP: Gaussian process for big data. Hensman et al. 2013.

Problem setup

Motivating example: modeling UK housing price



Nearest neighbor GP:

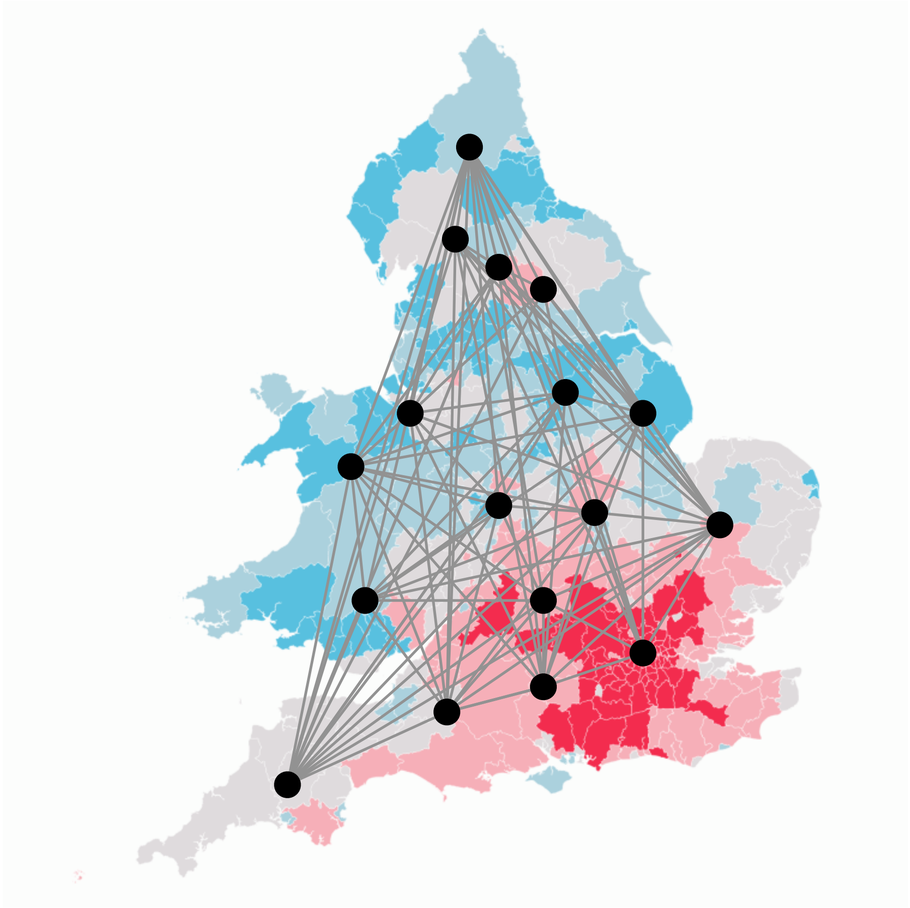
- Assumption: strong local correlations
- Common in spatial / temporal problems

*image source: https://twitter.com/undertheraedar/status/1388144547268530176

*Nearest neighbor GP: Hierarchical nearest-neighbor gaussian process models for large geostatistical datasets. Datta et al., 2016a

Problem setup

Motivating example: modeling UK housing price



Nearest neighbor GP:

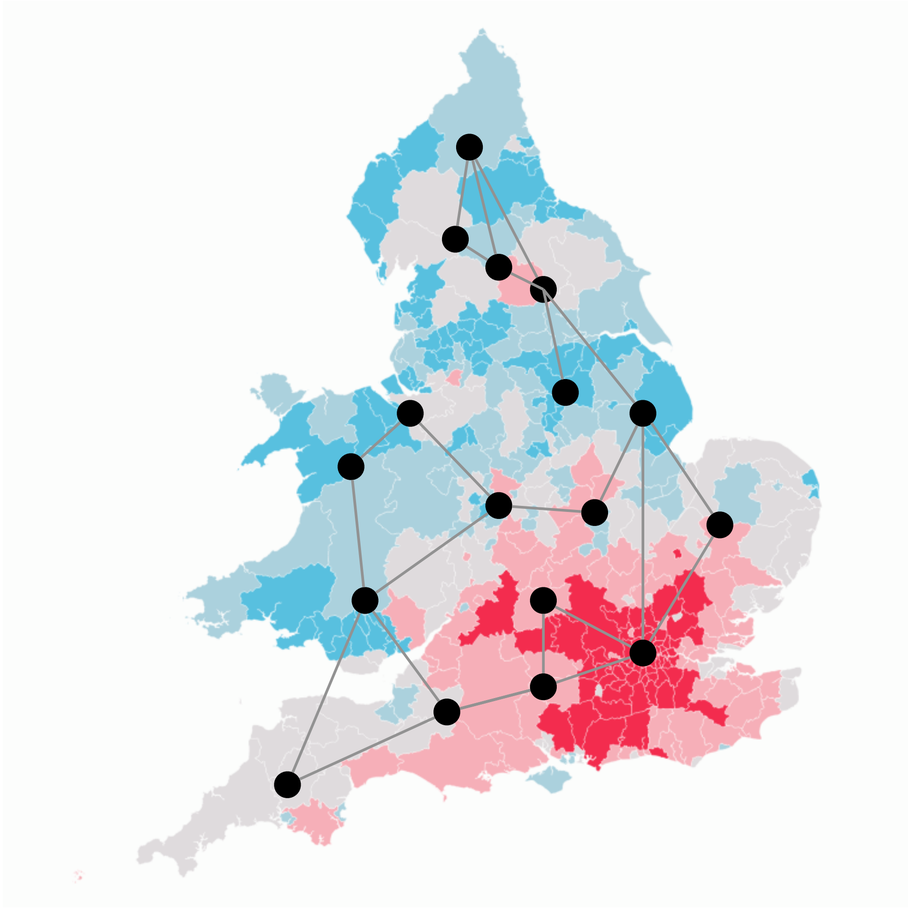
- Assumption: strong local correlations
- Common in spatial / temporal problems
- Instead of making a fully-connected dependency

*image source: https://twitter.com/undertheraedar/status/1388144547268530176

*Nearest neighbor GP: Hierarchical nearest-neighbor gaussian process models for large geostatistical datasets. Datta et al., 2016a

Problem setup

Motivating example: modeling UK housing price



Nearest neighbor GP:

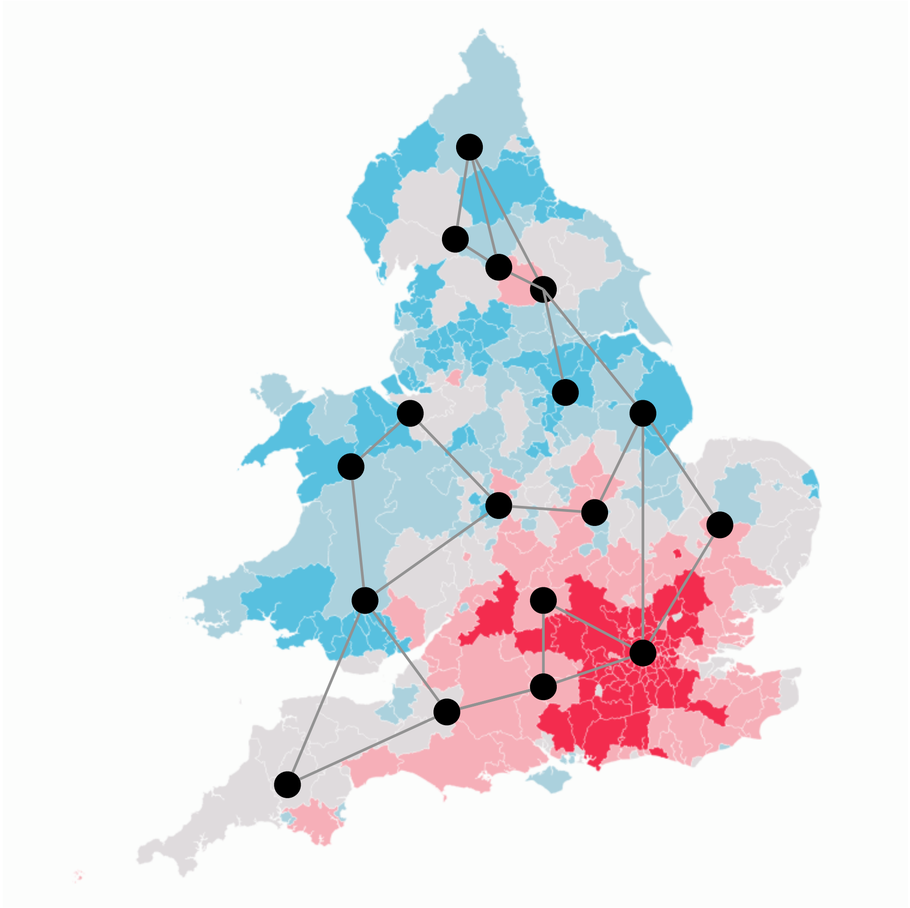
- Assumption: strong local correlations
- Common in spatial / temporal problems
- Instead of making a fully-connected dependency
- Reduce to K-nearest-neighbor dependency

*image source: https://twitter.com/undertheraedar/status/1388144547268530176

*Nearest neighbor GP: Hierarchical nearest-neighbor gaussian process models for large geostatistical datasets. Datta et al., 2016a

Problem setup

Motivating example: modeling UK housing price



Nearest neighbor GP:

- Assumption: strong local correlations
- Common in spatial / temporal problems
- Instead of making a fully-connected dependency
- Reduce to K-nearest-neighbor dependency
- Inference: MCMC, scales $O(NK^3)$
- Current practice is limited to Gaussian likelihoods

*image source: https://twitter.com/undertheraedar/status/1388144547268530176

*Nearest neighbor GP: Hierarchical nearest-neighbor gaussian process models for large geostatistical datasets. Datta et al., 2016a

Can we combine the best of two worlds?

Desiderata:

- Keep the flexibility of variational inference
- Boost the number of inducing points to achieve high-fidelity inference

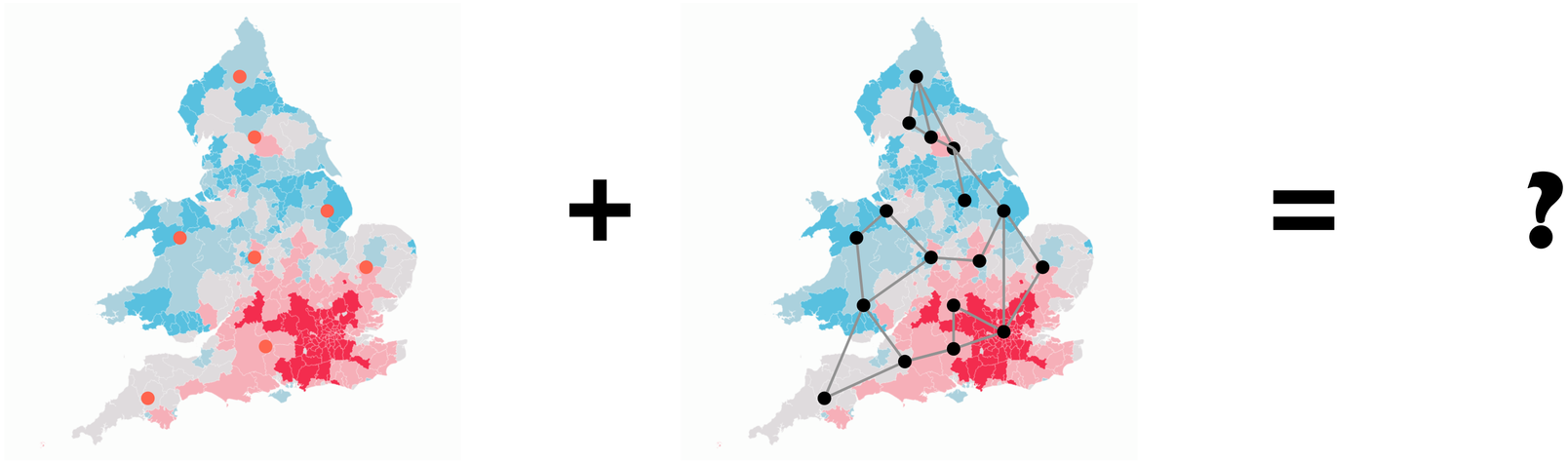


*image source: https://twitter.com/undertheraedar/status/1388144547268530176

VNNGP

Variational nearest neighbor Gaussian process

- Main methodology
  - nearest neighbor approx. + inducing-point GP + variational inference

VNNGP

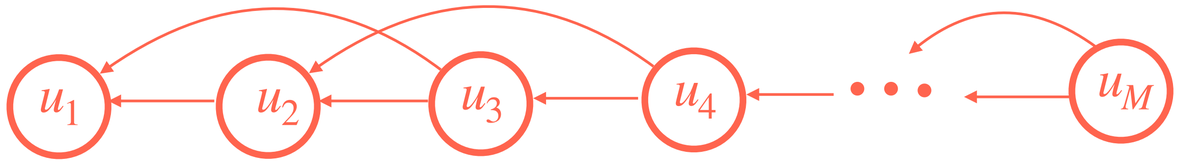
Variational nearest neighbor Gaussian process

- Main methodology
  - nearest neighbor approx. + inducing-point GP + variational inference
- Advantages
  - Enables stochastic optimization over data points **and** inducing points
  - Scales up the size of inducing points to match the size of data
  - Achieves better predictive performance

VNNGP: methodology

Generative process

- Make a K-nearest neighbor approximation for $p(u)$



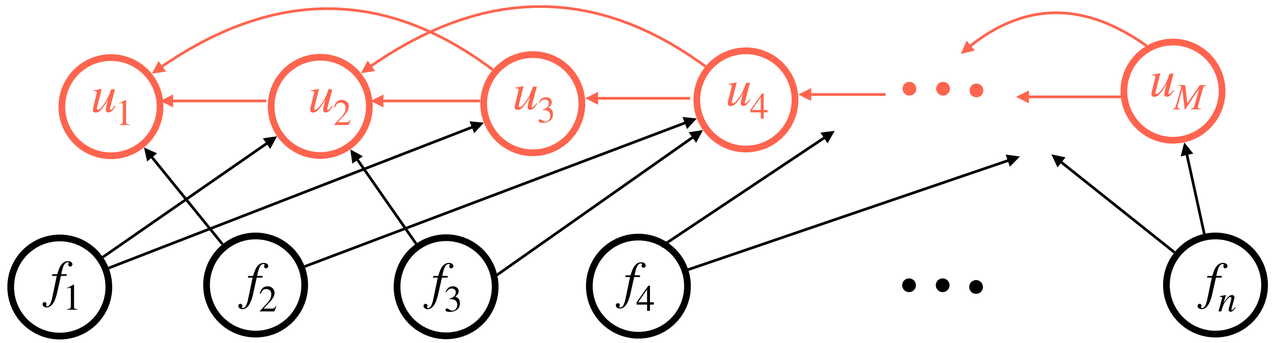
$$p(u) \approx \prod_{j=1}^{M} p(u_j \mid u_{n(j)})$$

nearest neighbor set

VNNGP: methodology

Generative process

- Make a K-nearest neighbor approximation for $p(u)$
- And then make observations only dependent on K nearest neighbor inducing points



$$p(u) \approx \prod_{j=1}^{M} p(u_j \mid u_{n(j)})$$

nearest neighbor set

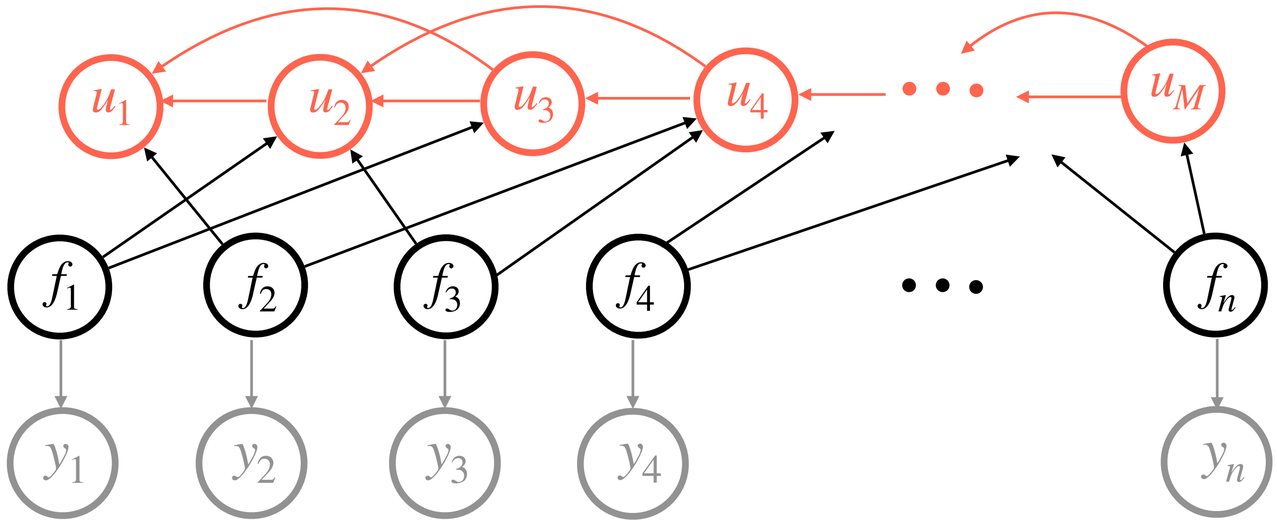
$$p(f \mid u) \approx \prod_{i=1}^{N} p(f_i \mid u_{n(i)})$$

limit the nearest neighbor set among inducing points

VNNGP: methodology

Generative process

- Make a K-nearest neighbor approximation for $p(u)$
- And then make observations only dependent on K nearest neighbor inducing points
- I.I.D. likelihood model



$$p(u) \approx \prod_{j=1}^{M} p(u_j \mid u_{n(j)})$$

$$p(f \mid u) \approx \prod_{i=1}^{N} p(f_i \mid u_{n(i)})$$

$$p(y \mid f, u) = \prod_{i=1}^{N} p(y_i \mid f_i)$$

VNNGP: methodology

Variational Inference

- Consider the following variational family for $q(u, f) = q(u)q(f \mid u)$
  - Inducing points: mean-field distribution $q(u) = \prod_{i=1}^{M} \mathcal{N}(u_j \mid m_j, s_j)$
    - Could be extended to a more expressive form, see appendix in paper

VNNGP: methodology

Variational Inference

- Consider the following variational family for $q(u, f) = q(u)q(f \mid u)$
  - Inducing points: mean-field distribution $q(u) = \prod_{i=1}^{M} \mathcal{N}(u_j \mid m_j, s_j)$
    - Could be extended to a more expressive form, see appendix in paper
  - Data points: nearest neighbor GP predictive $q(f \mid u) = \prod_{i=1}^{N} p(f_i \mid u_{n(i)})$

VNNGP: methodology

Variational Inference

- Consider the following variational family for $q(u, f) = q(u)q(f \mid u)$
  - Inducing points: mean-field distribution $q(u) = \prod_{i=1}^{M} \mathcal{N}(u_j \mid m_j, s_j)$
    - Could be extended to a more expressive form, see appendix in paper
  - Data points: nearest neighbor GP predictive $q(f \mid u) = \prod_{i=1}^{N} p(f_i \mid u_{n(i)})$

$$ELBO_{VNNGP} = \sum_{i=1}^{N} \mathbb{E}_{q(f_i|u)p(u)} \left[\log p(y_i \mid f_i)\right] - KL(q(u) \| p(u))$$

"model fit" "regularization"

VNNGP: methodology

$$\textbf{model fit} = \sum_{i=1}^{N} \mathbb{E}_{q(f_i|u)p(u)} \left[\log p(y_i \,|\, f_i)\right]$$

By the choice of variational posterior being the NN prior $q(f_i \,|\, u) = p(f_i \,|\, u_{n(i)})$

$$= \sum_{i=1}^{N} \mathbb{E}_{p(f_i|u_{n(i)})q(u_{n(i)})} \left[\log p(y_i \,|\, f_i)\right]$$

$O(K^3)$ computation

VNNGP: methodology

KL regularization $= KL(q(u) \| p(u))$

$$= KL(\prod_{j=1}^{M} q(u_j) \| \prod_{i=1}^{M} p(u_j \mid u_{n(j)}))$$ by mean-field posterior + NN prior

by def. of KL

$$= \mathbb{E}_{q(u)} \left[\sum_{j=1}^{M} \log \frac{q(u_j)}{p(u_j \mid u_{n(j)})} \right]$$

tower property

$$= \sum_{j=1}^{M} \mathbb{E}_{q(u_{n(j)})} \left[\mathbb{E}_{q(u_j)} \left[\log \frac{q(u_j)}{p(u_j \mid u_{n(j)})} \right] \right]$$

by def. of conditional KL

$$= \sum_{j=1}^{M} \mathbb{E}_{q(u_{n(j)})} \left[KL(q(u_j) \| p(u_j \mid u_{n(j)})) \right]$$

KL factorizes over inducing points!

$O(K^3)$ computation

VNNGP: methodology

Advantages

- Stochastic optimization over data points **and** inducing points

$$\mathscr{L}_{VNNGP} \approx \frac{N}{|I|} \sum_{i \in I} \mathbb{E}_{q(f_i)} \left[\log p(y_i \,|\, f_i) \right] - \frac{M}{|J|} \sum_{j \in J} \mathbb{E}_{q(u_{n_j})} \left[KL[q(u_j) \| p(u_j \,|\, u_{n(j)}) \right]$$

where I and J are random mini-batches of training data indices, and induing point indices

VNNGP: methodology

Advantages

- Stochastic optimization over data points **and** inducing points

$$\mathscr{L}_{VNNGP} \approx \frac{N}{|I|} \sum_{i \in I} \mathbb{E}_{q(f_i)} \left[\log p(y_i \,|\, f_i) \right] - \frac{M}{|J|} \sum_{j \in J} \mathbb{E}_{q(u_{n_j})} \left[KL[q(u_j) \| p(u_j \,|\, u_{n(j)}) \right]$$

where I and J are random mini-batches of training data indices, and induing point indices

- Training complexity is now independent of # of data points N and # of inducing points M

VNNGP: methodology

Advantages

- Stochastic optimization over data points **and** inducing points

$$\mathscr{L}_{VNNGP} \approx \frac{N}{|I|} \sum_{i \in I} \mathbb{E}_{q(f_i)} \left[\log p(y_i | f_i) \right] - \frac{M}{|J|} \sum_{j \in J} \mathbb{E}_{q(u_{n_j})} \left[KL[q(u_j) \| p(u_j | u_{n(j)}) \right]$$

where I and J are random mini-batches of training data indices, and induing point indices

- Training complexity is now independent of # of data points N and # of inducing points M

- Scale up M, eventually putting every inducing point at data locations s.t. $M = N$

Experiment evaluations

Example: UKHousing dataset

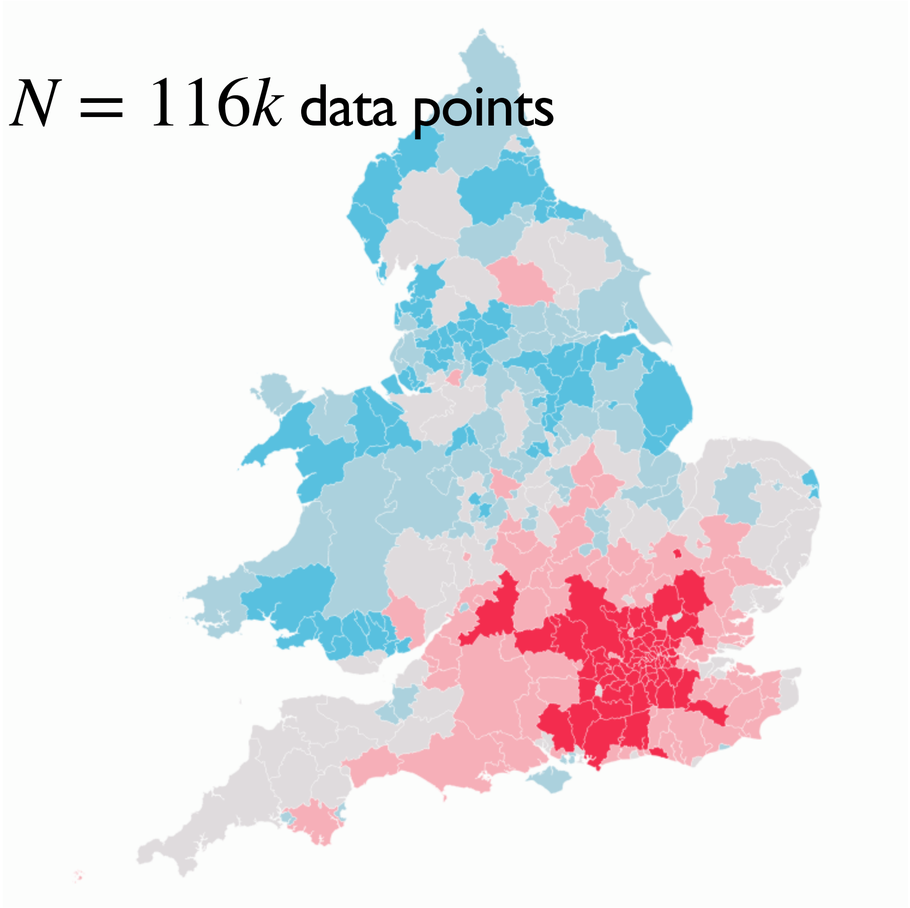
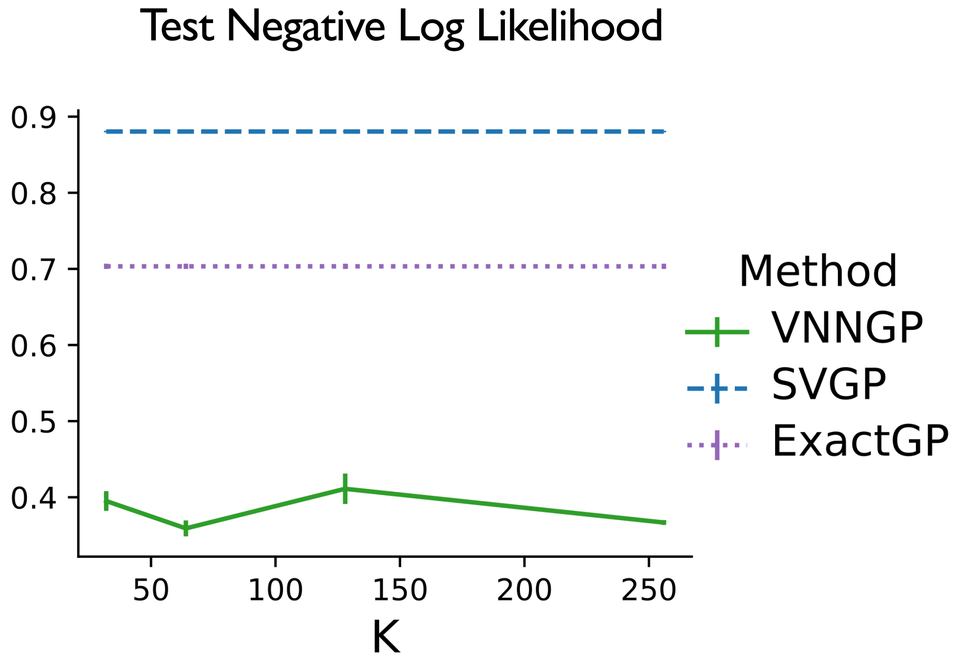


*image source: https://twitter.com/undertheraedar/status/1388144547268530176



VNNGP with $K = 32$ nearest neighbors already outperforms SVGP that uses 1024 inducing points.

- Our code is available on GPyTorch

  https://docs.gpytorch.ai/en/stable/examples/04_Variational_and_Approximate_GPs/VNNGP.html

Thank you! Please come to our poster.