Spatial “Data Science” Meets Bayesian Inference

Sudipto Banerjee

University of California, Los Angeles, USA

Department of Biostatistics UCLA

Outline

- Spatial “Data Science”
- Bayesian Geostatistics
- BIG Spatial DATA
- Bayesian Modeling for BIG Spatial Data

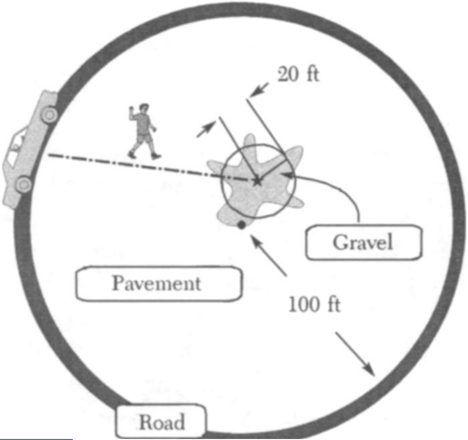
GPS, GIS and Spatial Data Science

Global Positioning Systems: GPS

- Finding the position on the earth from satellite receivers...

  https://www.maa.org/sites/default/files/pdf/cms_upload/Thompson07734.pdf

- 2 satellites should yield 2 equations (circles) that intersect at the point we seek.



- Finding the position on the earth from satellite receivers...

https://www.maa.org/sites/default/files/pdf/cms_upload/Thompson07734.pdf

- But a third satellite may not agree: Error (in watch) between earth time and signal departure time (from satellite).

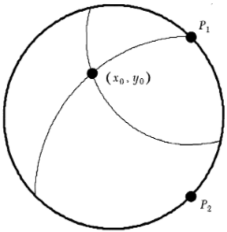


FIGURE 3
Two Messengers.

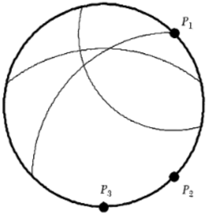


FIGURE 4
Three Messengers.

- Finding the position on the earth from satellite receivers...

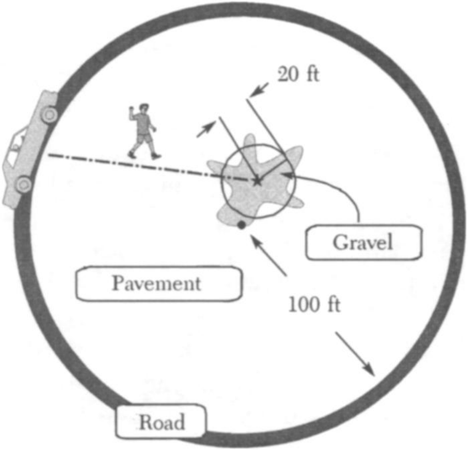
  https://www.maa.org/sites/default/files/pdf/cms_upload/Thompson07734.pdf

- 3 satellites yield 3 equations (circles) to solve for coordinates and error :

$$\|p - r_i\|^2 = d(\Delta t_i, \epsilon)^2 \; ; \quad i = 1, 2, 3 \; ,$$

  where:
  - $p = (x, y)$ is the position vector we wish to find (3 unknowns);
  - r_i is the position vector of satellite i for $i = 1, 2, 3$ (known);
  - Δt_i is the elapsed time for signal to arrive from satellite i (known);
  - ϵ is the error in time clock (1 unknown);
  - $d(\Delta t_i, \epsilon)$ is the distance of p from satellite i (known in terms of ϵ).

Mathematics of Global Positioning Systems in R

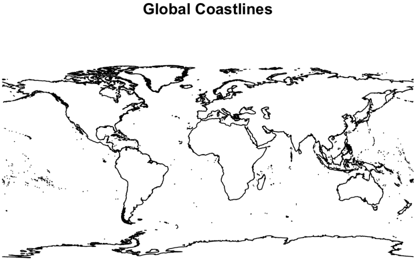


```
library(nleqslv)

funch <- function(d,x){
a <- 20 + (5* (d - x -5))

    return(a)

}

func <- function(x) {
y <- rep(0, times=3)

    y[1] <- (x[1] - 70.7)^2 + (x[2] - 70.7)^2 - (funch(20.2,x[3]))^2
    y[2] <- (x[1] - 70.7)^2 + (x[2] + 70.7)^2 - (funch(29.5,x[3]))^2
    y[3] <- (x[1] - 0.0)^2 + (x[2] + 100.0)^2 - (funch(32.2,x[3]))^2

    return(y)

}

xstart <- matrix(c(15,30,5), ncol=3)

root <- searchZeros(xstart, func, method="Newton", global="dbldog")
```

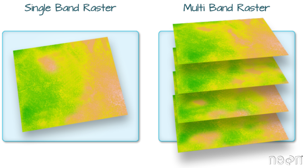
- Computing system for storing and analyzing spatial data.
- Survey data can be directly entered into a GIS from digital data collection systems on survey instruments.
- GPS data be collected and then imported into a GIS.
- Web mining is a method of collecting spatial data using “web crawlers” (programs to aggregate required spatial data from the web).

Intro to spatial data in R
By Leah A. Wasser
https://nceas.github.io/oss-lessons/spatial-data-gis-law/3-mon-intro-gis-in-r.html

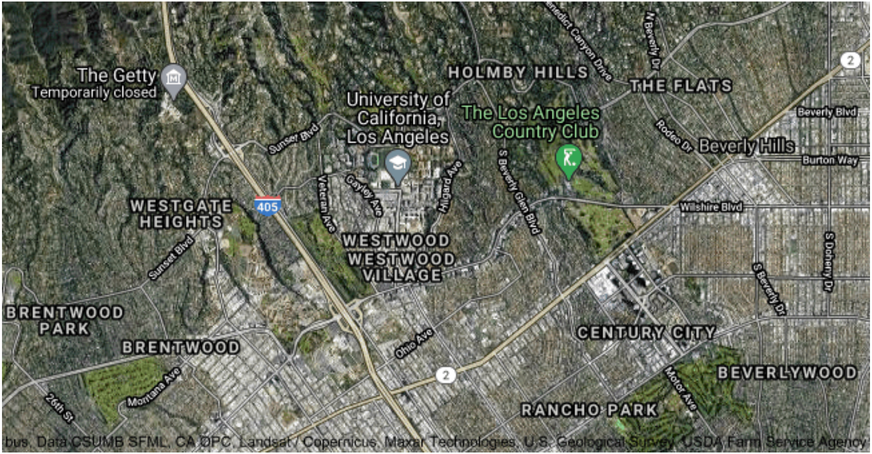
Vector data: points and lines



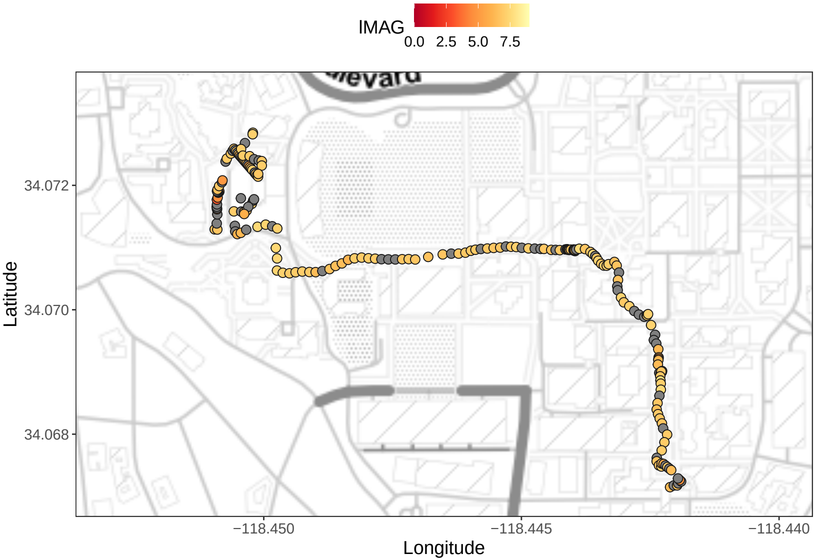
Raster data: pixels



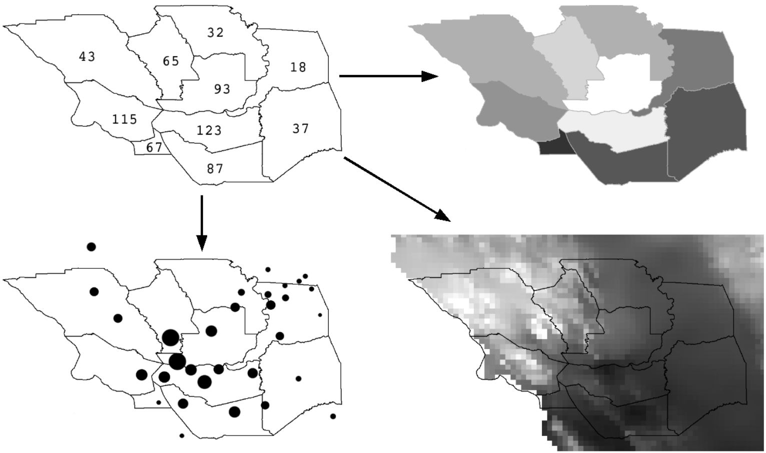
Westwood neighborhood



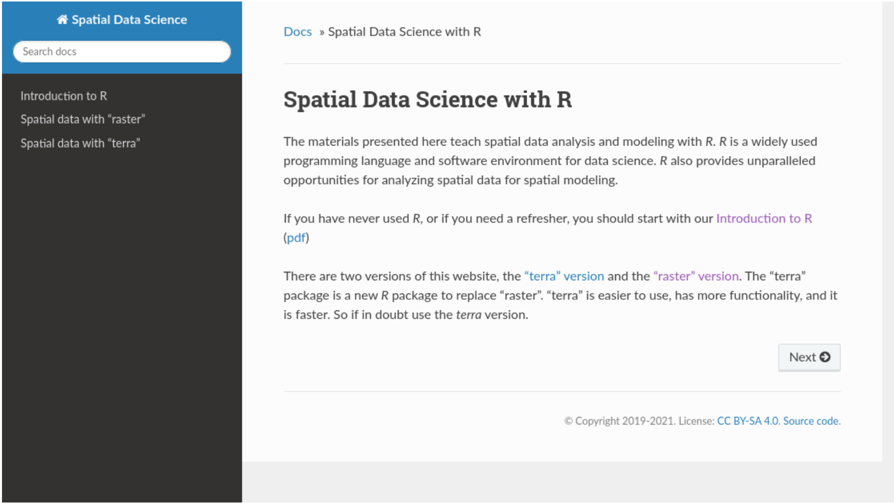
GIS and GPS together...



Different spatial data types: Misalignment & COSP



Spatial Data Science in R: https://rspatial.org/


Spatial Data Science

Search docs

Introduction to R

Spatial data with “raster”

Spatial data with “terra”

Docs » Spatial Data Science with R

Spatial Data Science with R

The materials presented here teach spatial data analysis and modeling with *R*. *R* is a widely used programming language and software environment for data science. *R* also provides unparalleled opportunities for analyzing spatial data for spatial modeling.

If you have never used *R*, or if you need a refresher, you should start with our Introduction to R (pdf)

There are two versions of this website, the “terra” version and the “raster” version. The “terra” package is a new *R* package to replace “raster”. “terra” is easier to use, has more functionality, and it is faster. So if in doubt use the *terra* version.

Next

Bayesian Geostatistics

Point-referenced spatial data

- Each observation is associated with a location (point)
- Data represents a sample from a continuous spatial domain
- Also referred to as geocoded or geostatistical data

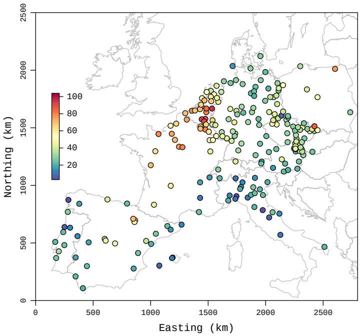
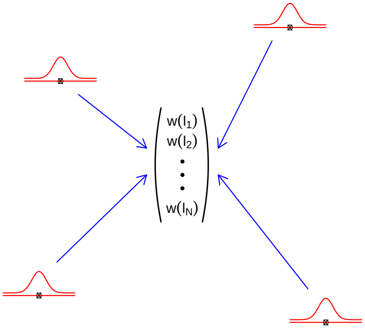


Figure: Pollutant levels in Europe in March, 2009

Hierarchical spatial process models: Cressie & Wikle (2011); Banerjee, Carlin & Gelfand (2014)

$$[\text{data} \mid \text{process, parameters}] \times [\text{process} \mid \text{parameters}] \times [\text{parameters}]$$

What is a process?



- Continuous data:

$$
\begin{aligned}
y(s_i) \,|\, \mu(s_i),\, \tau^2 &\overset{ind}{\sim} N(\mu(s_i), \tau^2)\ ; \quad i = 1, 2, \ldots, n\ ; \\
\mu(s_i) &= \beta_0 + \beta_1 x_1(s_i) + \beta_2 x_2(s_i) + \cdots + \beta_p x_p(s_i) + w(s_i)\ ; \\
\beta_j &\overset{ind}{\sim} N(0, \sigma_\beta^2)\ ; \quad j = 0, 1, \ldots, p\ ; \\
w = (w(s_1), w(s_2), \ldots, w(s_n))^\top &\sim N(0, \sigma^2 R_w(\phi))\ ; \\
1/\tau^2 \sim \text{Gamma}(a_\tau, b_\tau)\ ; &\quad 1/\sigma^2 \sim \text{Gamma}(a_\sigma, b_\sigma)\ ; \\
&\qquad \phi \sim \text{Unif}(a_\phi, b_\phi)\ .
\end{aligned}
$$

- $R_w(\phi)$ is $n \times n$ spatial correlation matrix.

- Count data:

$$
\begin{aligned}
y(s_i) &\sim Poi(\lambda(s_i))\ ; \quad i = 1, 2, \ldots, n\ ; \\
\log\lambda(s_i) &= \beta_0 + \beta_1 x_1(s_i) + \beta_2 x_2(s_i) + \cdots + \beta_p x_p(s_i) + w(s_i)\ ; \\
\beta_j &\overset{ind}{\sim} N(0, \sigma_\beta^2)\ ; \quad j = 0, 1, \ldots, p\ ; \quad w \sim N(0, \sigma^2 R_w)\ ; \\
1/\sigma^2 &\sim \text{Gamma}(a_\sigma, b_\sigma)\ ; \quad \phi \sim \text{Unif}(a_\phi, b_\phi)\ .
\end{aligned}
$$

- $R_w(\phi)$ is $n \times n$ spatial correlation matrix.

- Binary data:

$$
\begin{aligned}
y(s_i) &\sim Ber(p(s_i))\ ; \quad i = 1, 2, \ldots, n\ ; \\
\log\left(\frac{p(s_i)}{1-p(s_i)}\right) &= \beta_0 + \beta_1 x_1(s_i) + \beta_2 x_2(s_i) + \cdots + \beta_p x_p(s_i) + w(s_i)\ ; \\
\beta_j \overset{ind}{\sim} N(0, \sigma_\beta^2)\ ; &\quad j = 0, 1, \ldots, p\ ; \quad w \sim N(0, \sigma^2 R_w)\ ; \\
1/\sigma^2 &\sim \mathrm{Gamma}(a_\sigma, b_\sigma)\ ; \quad \phi \sim \mathrm{Unif}(a_\phi, b_\phi)\ .
\end{aligned}
$$

- $R_w(\phi)$ is $n \times n$ spatial correlation matrix.

Spatial random effects: Gaussian process

- We say that $w(s) \sim GP(0, \sigma^2\rho(\cdot))$:

$$w = (w(s_1), w(s_2), \ldots, w(s_n))^\top \sim N(0, \sigma^2 R_w)\ ;$$

- R_w is $n \times n$ spatial correlation matrix:

$$R_w[i,j] = \rho(s_i, s_j)\ .$$

- The correlation function is parametrized to capture strength of association as a function of distance. Practical choice (works well for a variety of situations):

$$\rho(s_i, s_j) = \exp(-\phi\|s_i - s_j\|)\ .$$

- Step-I: Estimate parameters (MCMC) by sampling from

$$[\beta, w, \tau^2, \sigma^2, \phi \,|\, y, X]$$

- Step-II: Estimate the latent process $w(s_0)$ at new location s_0 by sampling from

$$[w(s_0) \,|\, w, \sigma^2, \phi]$$

  for each sampled value of w, σ^2 and ϕ obtained in Step-I.
- Step III: Obtain posterior samples of $\mu(s_0)$:

$$\mu(s_0) = \beta_0 + \beta_1 x_1(s_0) + \beta_2 x_2(s_0) + \cdots + \beta_p x_p(s_0) + w(s_0)\ .$$

- Step-IV: Predict $y(s_0)$ by drawing its value from

$$N(\mu(s_0), \tau^2)$$

  for each sampled $\mu(s_0)$ (from Step-III) and τ^2 (from Step-I).

Bayesian inference for spatial count data

- Step-I: Estimate parameters (MCMC) by sampling from

$$[\beta, w, \sigma^2, \phi \,|\, y, X]$$

- Step-II: Estimate the latent process $w(s_0)$ at new location s_0 by drawing from

$$[w(s_0) \,|\, w, \sigma^2, \phi]$$

  for each sampled value of w, σ^2 and ϕ obtained in Step-I.
- Step III: Obtain posterior samples of $\lambda(s_0)$:

$$\lambda(s_0) = \exp\left(\beta_0 + \beta_1 x_1(s_0) + \beta_2 x_2(s_0) + \cdots + \beta_p x_p(s_0) + w(s_0)\right) \; .$$

- Step-IV: Predict $y(s_0)$ by drawing its value from

$$Poi(\lambda(s_0))$$

  for each sampled $\lambda(s_0)$ in Step-III.

- Step-I: Estimate parameters (MCMC) by sampling from

$$[\beta, w, \sigma^2, \phi \,|\, y, X]$$

- Step-II: Estimate the latent process $w(s_0)$ at new location s_0 by drawing from

$$[w(s_0) \,|\, w, \sigma^2, \phi]$$

  for each sampled value of w, σ^2 and ϕ obtained in Step-I.
- Step III: Obtain posterior samples of $p(s_0)$:

$$p(s_0) = \text{logit}^{-1} \left(\beta_0 + \beta_1 x_1(s_0) + \beta_2 x_2(s_0) + \cdots + \beta_p x_p(s_0) + w(s_0)\right) \; .$$

- Step-IV: Predict $y(s_0)$ by drawing its value from

$$Ber(p(s_0))$$

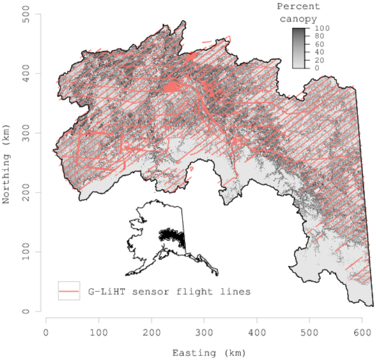
  for each sampled $p(s_0)$ in Step-III.

Setting Priors

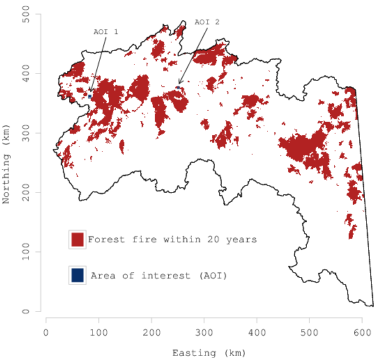
- For regression slopes we customarily assign non-informative priors.
- For the variance component σ^2 (partial sill), we customarily choose an Inverse-Gamma (or, equivalently, Gamma prior for $1/\sigma^2$)—the shape parameter is taken to be 2 and the scale parameter is chosen so that the prior mean is equal to the scale parameter. This value can be set from an exploratory variogram analysis. Strategy for τ^2 (nugget) is similar.
- For the range parameter ϕ, we usually set it so that the effective range (distance where spatial correlation drops to 0.05) is between some small number and does not exceed about 50% of the maximum inter-site distance. For example, with the exponential correlation function we solve $\rho(d;\phi) = 0.05$ and see that $\phi \approx 3/d$, where d is the effective spatial range. We bound $d \in (d_{\min}, d_{\max})$ and this suggests $\phi \sim \text{Unif}(3/d_{\max}, 3/d_{\max})$.

BIG Spatial DATA

Example: Alaska Tanana Valley Forest Height Data



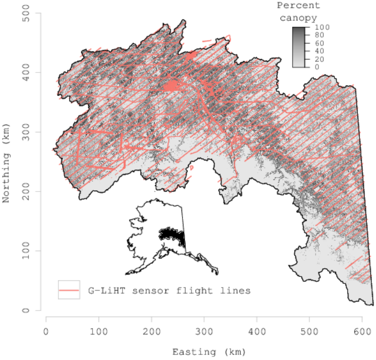
(a) Forest height and tree cover



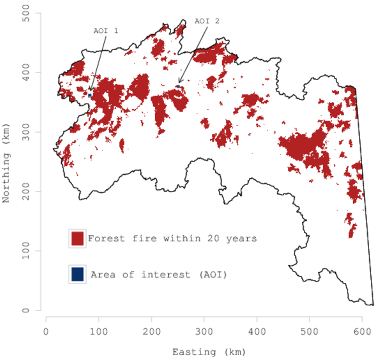
(b) Forest fire history

- Forest height (red lines) data from LiDAR at 5×10^6 locations
- Knowledge of forest height is important for biomass assessment, carbon management etc

Example: Alaska Tanana Valley Forest Height Data



(c) Forest height and tree cover



(d) Forest fire history

- Goal: High-resolution domainwide prediction maps of forest height
- Covariates: Domainwide tree cover (grey) and forest fire history (red patches) in the last 20 years

Models used:

- Non-spatial regression: $y_{FH} = \beta_0 + \beta_{tree} x_{tree} + \beta_{fire} x_{fire} + \epsilon$

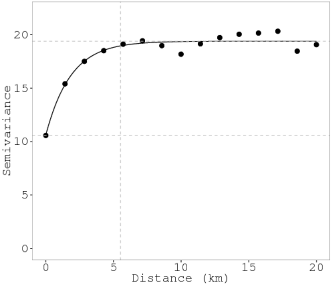


Figure: Variogram of the residuals from non-spatial regression indicates strong spatial pattern

Geostatistical models: Example of spatial process models

- $y_{FH}(\ell) = \beta_0 + \beta_{tree} x_{tree}(\ell) + \beta_{fire} x_{fire}(\ell) + w(\ell) + \epsilon(\ell)$
- $w(\ell) \sim GP(0, C(\cdot, \cdot \,|\, \theta)); \theta = \{\sigma^2, \rho, \nu\}$.
- Example of a covariance function:

$$C(\ell, \ell' \,|\, \theta) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{\|\ell - \ell'\|}{\rho} \right)^{\nu} K_{\nu} \left(\sqrt{2\nu} \frac{\|\ell - \ell'\|}{\rho} \right) .$$

- Data y_{FH}, X observed over $\{\ell_1, \ell_2, \ldots, \ell_n\}$.
- $y_{FH} \sim N(X\beta, K_\theta)$ where K_θ is the spatial covariance matrix:

$$K_\theta = C_{(\sigma,\phi)} + \tau^2 I \ , \quad \text{where} \;\; \theta = \{\sigma, \phi, \tau\}$$

where $C_{(\sigma^2,\phi)} = [C(\ell_i, \ell_j \,|\, \sigma^2, \phi)]$ is the $n \times n$ covariance matrix.

- Computing: (i) $\texttt{chol}(K_\theta) = LDL^\top$, (ii) $v = \texttt{trsolve}(L, y - X\beta)$,

$$-\frac{1}{2}\sum_{i=1}^{n} \log d_{ii} - \frac{1}{2}\sum_{i=1}^{n} v_i^2/d_{ii}$$

Computational Details

- Compute the quadratic form and determinant (for any given $\{\beta, \theta\}$):

$$\begin{array}{ll} \text{Cholesky:} & \texttt{chol}(K_\theta) = LDL^\top \ \text{(expensive)}\ ; \\ \text{Solve for } v\text{:} & v = \texttt{trsolve}(L, y - X\beta)\ \text{(cheap)}\ ; \\ \text{Quadratic form:} & v^\top D^{-1} v = \sum_{i=1}^n v_i^2/d_{ii}\ \text{(cheap)}\ ; \\ \text{Determinant:} & \log\det(K_\theta) = \sum_{i=1}^n \log d_{ii}\ \text{(cheap)}\ . \end{array}$$

- Log-likelihood (up to a constant):

$$-\frac{1}{2}\sum_{i=1}^n \log d_{ii} - \frac{1}{2}\sum_{i=1}^n v_i^2/d_{ii}$$

- Bayesian inference: Priors on $\{\beta, \theta\}$
- Bayesian interpolation: $p(w(\ell_0) \mid y_{FH})$ is well-defined.
- Bayesian prediction: $p(y_{FH}(\ell_0) \mid y_{FH})$ is well-defined.
- Requires iterative algorithms (e.g., MCMC or variants; INLA; VB).

Prediction and interpolation

- Conditional predictive density

$$p(y(\ell_0) \,|\, y, \theta, \beta) = N\left(y(\ell_0) \,\middle|\, \mu(\ell_0), \sigma^2(\ell_0)\right) \;.$$

- "Kriging" (spatial prediction/interpolation)

$$\begin{aligned} \mu(\ell_0) &= \mathrm{E}[y(\ell_0) \,|\, y, \theta] = x^\top(\ell_0)\beta + k_\theta^\top(\ell_0) K_\theta^{-1}(y - X\beta) \;, \\ \sigma^2(\ell_0) &= \mathrm{var}[y(\ell_0) \,|\, y, \theta] = K_\theta(\ell_0, \ell_0) - k_\theta^\top(\ell_0) K_\theta^{-1} k_\theta(\ell_0) \;. \end{aligned}$$

- Bayesian "kriging" computes (simulates) posterior predictive density:

$$p(y(\ell_0) \,|\, y) = \int p(y(\ell_0) \,|\, y, \theta, \beta) p(\beta, \theta \,|\, y) \mathrm{d}\beta \mathrm{d}\theta$$

Computational Details for Prediction

- Compute the mean and variance (for any given $\{\beta, \theta\}$ and ℓ_0):

$$
\begin{array}{ll}
\text{Cholesky:} & \texttt{chol}(K_\theta) = LDL^\top \; ; \\
\text{Solve for } v\text{:} & v = \texttt{trsolve}(L, k_\theta(\ell_0)) \; ; \\
\text{Solve for } u\text{:} & u = \texttt{trsolve}(L^\top, D^{-1}v) \; ; \\
\text{Predictive mean:} & x^\top(\ell_0)\beta + u^\top(y - X\beta) \; ; \\
\text{Predictive variance:} & K_\theta(\ell_0, \ell_0) - u^\top k_\theta(\ell_0) \; .
\end{array}
$$

- Primary bottleneck is chol$(\cdot)$

- Bayesian spatial interpolation also yields posterior of $\{w(\ell_0), w\}$:

$$
p(w(\ell_0), w \,|\, y) = \int p(w(\ell_0) \,|\, w, \theta, \beta) p(w \,|\, y, \theta, \beta) p(\theta, \beta \,|\, y) \mathrm{d}\beta \mathrm{d}\theta
$$

Bayesian Inference for BIG Spatial Data

Bayesian regression for BIG DATA

- Conjugate Bayesian hierarchical linear model:

$$y_i \,|\, \beta, \sigma^2 \stackrel{ind}{\sim} N(x_i^\top \beta, \sigma^2)\ ,\ i = 1, 2, \ldots, n\ ;$$
$$\beta \,|\, \sigma^2 \sim N(\mu_\beta, \sigma^2 V_\beta)\ ; \quad \sigma^2 \sim IG(a, b)\ .$$

- Exact Bayesian inference:

$$\sigma^2 \,|\, y \sim IG(a^*, b^*) \quad \beta \,|\, \sigma^2, y \sim N(Mm, \sigma^2 M)\ , \quad \text{where}$$
$$m = V_\beta^{-1}\mu_\beta + X^\top y\ , \quad M^{-1} = V_\beta^{-1} + X^\top X\ ,$$
$$a^* = a + n/2\ , \quad b^* = \mu_\beta^\top V_\beta^{-1}\mu_\beta + y^\top y - m^\top M^{-1} m\ .$$

- What if the data cannot be stored/loaded into available workspace?

Bayesian updating with independent data

- Sequential update for $\theta = \{\beta, \sigma^2\}$: update posterior for $k = 1, 2, \ldots, n$

$$p(\theta \,|\, y_1, y_2, \ldots, y_k) \propto p(\theta \,|\, y_1, y_2, \ldots, y_{k-1}) \times p(y_k \,|\, \theta)\ .$$

- Posterior becomes the prior for the next k.
- Divide & Conquer with cloud computing (e.g., HADOOP).

- Partition data as $\{y_k, X_k\}$, $k = 1, 2, \ldots, K$, where each y_k is $n_k \times 1$, X_k is $n_k \times p$ and $N = \sum_{k=1}^{K} n_k$.
- For each subset compute:

$$m_k = V_\beta^{-1} + X_k^\top y_k \text{ and } M_k^{-1} = V_\beta^{-1} + X_k^\top X_k \;.$$

- Then compute

$$m = \sum_{k=1}^{K} (m_k - (1 - 1/K)\mu_\beta) \;;$$
$$M^{-1} = \sum_{k=1}^{K} (M_k^{-1} - (1 - 1/K)V_\beta^{-1}) \;.$$

- Crucially depends on independence of observations.
- Meta-Kriging (Guhaniyogi and Banerjee, *Technometrics*, 2018): find convex combination of subset-posteriors closest to the full posterior.

Bayesian low rank models (Wikle, *HSS*, 2010)

- Hierarchical Bayesian regression models are *naturally* low-rank:

$$
\begin{aligned}
y \,|\, \beta, z, \theta, \tau &\sim N(X\beta + B_\theta z, D_\tau)\ ; \\
z \,|\, \theta &\sim N(0, V_{z,\theta})\ ; \\
\beta \,|\, \mu_\beta, V_\beta &\sim N(0, V_\beta)\ ; \\
\theta, \tau &\sim p(\theta, \tau) = p(\theta) \times p(\tau)\ .
\end{aligned}
$$

- Posterior distribution:

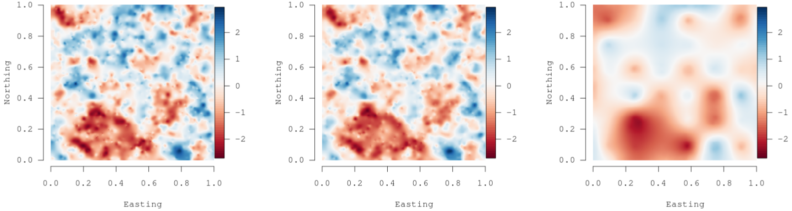
$$
p(\theta) \times p(\tau) \times N(\beta \,|\, \mu_\beta, V_\beta) \times N(z \,|\, 0, V_{z,\theta}) \times N(y \,|\, X\beta + B_\theta z, D_\tau)\ .
$$

- $B_\theta z$? Start with a *parent process* $w(\ell)$ and construct $\tilde{w}(\ell)$

$$
w(\ell) \approx \tilde{w}(\ell) = \sum_{j=1}^{r} b_\theta(\ell, \ell_j^*) z(\ell_j^*) = b_\theta^\top(\ell) z.
$$

- Example: $\tilde{w}(\ell) = \mathrm{E}[w(\ell) \,|\, w^*] = \sum_{j=1}^{r} b_\theta(\ell, \ell_j^*) w(\ell_j^*)$

Oversmoothing in low rank models (Banerjee, *Bayesian Anal.*, 2017)



(a) True w (b) Full GP (c) PPGP 64 knots

Figure: Comparing full GP vs low-rank GP with 2500 locations. Figure (3(c)) exhibits oversmoothing by a low-rank process (with $r = 64$)

- Can be explained: $P_{[B_1:B_2]} = P_{B_1} + P_{[(I-P_{B_1})B_2]}$
- Fixes and improvements: MRA (Katzfuss, *JASA*, 2016).
- Sparse approximations or sparsity-inducing processes.

Burgeoning literature on scalable GPs for large data

- Vecchia, A.V. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 50, 297–312. DOI: https://doi.org/10.1111/j.2517-6161.1988.tb01729.x
- Wikle, C.K and Cressie, N. (1999). A dimension-reduced approach to space-time Kalman filtering. *Biometrika*, 86, 815–829. DOI: https://doi.org/10.1093/biomet/86.4.815
- Stein, M.L., Chi, Z. and Welty, L.J. (2004), Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66, 275–296. DOI: https://doi.org/10.1046/j.1369-7412.2003.05512.x
- Banerjee, S., Gelfand, A.E., Finley, A.O. and Sang, H. (2008). Gaussian predictive process models for large spatial datasets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70, 825–848.
- Cressie, N. and Johannesson, G. (2008), Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70, 209-226. DOI: https://doi.org/10.1111/j.1467-9868.2007.00633.x
- Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., and Sain, S. (2015). A multiresolution gaussian process model for the analysis of large spatial datasets. *Journal of Computational and Graphical Statistics*, 24, 579–599.
- Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. (2016a). Hierarchical Nearest-Neighbor Gaussian Process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111, 800–812. DOI: https://doi.org/10.1080/01621459.2015.1044091.
- Datta, A., Banerjee, S., Finley, A. O., Hamm, N. A. S., and Schaap, M. (2016b). Non-separable dynamic Nearest-Neighbor Gaussian Process models for large spatio-temporal data with an application to particulate matter analysis. *Annals of Applied Statistics*, 10, 1286–1316.
- Stroud, J.R., Stein, M.L. and Lysen, S. (2017) Bayesian and maximum likelihood estimation for Gaussian processes on an incomplete lattice, *Journal of Computational and Graphical Statistics*, 6, 108–120. DOI: https://doi.org/10.1080/10618600.2016.1152970.
- Katzfuss, M. (2017). A multi-resolution approximation for massive spatial datasets. *Journal of the American Statistical Association*, 112, 201–214.
- Banerjee, S. (2017). High-dimensional Bayesian geostatistics. *Bayesian Analysis*, 12, 583–614. DOI: https://doi.org/10.1214/17-BA1056R
- Finley, A.O., Datta, A., Cook, B.C., Morton, D.C. Andersen, H.E. and Banerjee, S. (2019). Efficient algorithms for Bayesian nearest-neighbor Gaussian processes. *Journal of Computational and Graphical Statistics*, 28, 401–414. DOI: https://doi.org/10.1080/10618600.2018.1537924.
- Heaton, M., Datta, A., Finley, A., Furrer, R., Guhaniyogi, R., Gerber, F., Hammerling, D., Katzfuss, M., Lindgren, F., Nychka, D., and Zammit-Mangion, A. (2019). A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological and Environmental Statistics*, 24, 398-425. DOI: https://doi.org/10.1007/s13253-018-00348-w
- Katzfuss, M. and Guinness, J. (2021). A general framework for Vecchia approximations of gaussian processes. *Statistical Science*, 36, 124–141. DOI: https://doi.org/10.1214/19-STS755
- Katzfuss, M., Stroud, J.R. and Wikle, C.K. (in press). Ensemble Kalman methods for high-dimensional hierarchical dynamic space-time models. *Journal of the American Statistical Association*, DOI: https://doi.org/10.1080/01621459.2019.1592753
- Peruzzi, M., Banerjee, S. and Finley, A.O. (in press). Highly scalable Bayesian geostatistical modeling via meshed Gaussian processes on partitioned domains. *Journal of the American Statistical Association*, DOI: https://doi.org/10.1080/01621459.2020.1833889

https://cemse.kaust.edu.sa/stsds/2021-kaust-competition-spatial-statistics-large-datasets

KAUST CEMSE Join Us News Events Calendar

STSDS Spatio-Temporal Statistics & Data Science

People Publications Talks Teaching Software STSDS Servers KAUST Competitions on Spatial Statistics Contacts

2021 KAUST Competition on Spatial Statistics for Large Datasets

Introduction

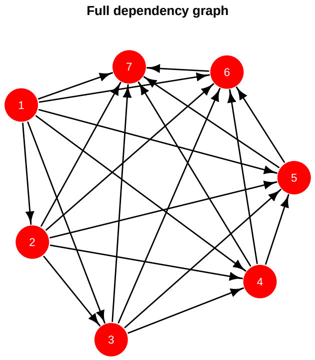
With the development of observing techniques and computing devices, it has become easier and more common to obtain large datasets. Statistical inference in spatial statistics becomes computationally challenging. For decades, various approximation methods have been proposed to model and analyze large-scale spatial data when the exact computation is infeasible. However, in the literature, the performance of the statistical inference using those proposed approximation methods was usually assessed with small and medium datasets only, for which the exact solution can be obtained. Then, for real-world large datasets, the exact computation was no longer feasible. The inference with approximation methods was often validated empirically or via prediction accuracy with the fitted model.

In this competition, the goal is to reassess existing approximation methods on large spatial datasets in a uniform way that guarantees a fair comparison. The results will be compared to the exact solution provided by the ExaGeoStat software. We generated a collection of synthetic datasets on a large scale from a set of selected true models. We aim at validating the statistical performance of the state-of-the-art approximation methods in terms of modeling, inference, and prediction. The selected true models cover disparate spatial properties to ensure a fair comparison among all the competitors' methods.

Burgeoning literature on DAG-based modeling...

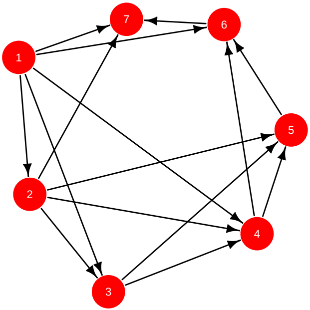
- Vecchia, A.V. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 50, 297–312. DOI: https://doi.org/10.1111/j.2517-6161.1988.tb01729.x
- Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. (2016). Hierarchical Nearest-Neighbour Gaussian Process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111, 800–812. DOI: https://doi.org/10.1080/01621459.2015.1044091.
- Katzfuss, M. and Guinness, J. (2021). A general framework for Vecchia approximations of gaussian processes. *Statistical Science*, 36, 124–141. DOI: https://doi.org/10.1214/19-STS755
- Peruzzi, M., Banerjee, S. and Finley, A.O. (in press). Highly scalable Bayesian geostatistical modeling via meshed Gaussian processes on partitioned domains. *Journal of the American Statistical Association*, DOI: https://doi.org/10.1080/01621459.2020.1833889

Simple method of introducing sparsity (e.g. graphical models)



$$p(y_1)p(y_2 \mid y_1)p(y_3 \mid y_1, y_2)p(y_4 \mid y_1, y_2, y_3)$$
$$\times\, p(y_5 \mid y_1, y_2, y_3, y_4)p(y_6 \mid y_1, y_2, \ldots, y_5)p(y_7 \mid y_1, y_2, \ldots, y_6)\ .$$

Simple method of introducing sparsity (e.g. graphical models)

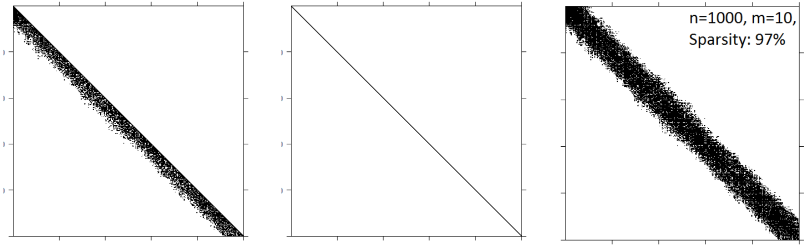


$$p(y_1)p(y_2 \mid y_1)p(y_3 \mid y_1, y_2)p(y_4 \mid y_1, y_2, y_3)$$
$$p(y_5 \mid \cancel{y_1}, y_2, y_3, y_4)p(y_6 \mid y_1, \cancel{y_2}, \cancel{y_3}, y_4, y_5)p(y_7 \mid y_1, y_2, \cancel{y_3}, \cancel{y_4}, \cancel{y_5}, y_6)$$

Sparse precision matrices (e.g., graphical Gaussian models)

$$p(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} p(w_i \,|\, w_{<i}) \approx \prod_{i=1}^{n} p(w_i \,|\, w_{\partial_i})$$

$$N(w \,|\, 0, K_\theta) \approx N(w \,|\, 0, \tilde{K}_\theta)\ ;\ \tilde{K}_\theta^{-1} = (I - A)^\top D^{-1} (I - A)$$



(a) $I - A$ (b) D^{-1} (c) $\tilde{K}_\theta^{-1}$

- $\det(\tilde{K}_\theta^{-1}) = \prod_{i=1}^{n} D_{ii}^{-1}$, $\tilde{K}_\theta^{-1}$ is sparse with $O(nm^2)$ entries

Pseudocodes for A and D

- Computing A and D

```
for(i in 1:(n-1) {
  Pa = N[i+1] # neighbors of i+1
  a[i+1,Pa] = solve(K[Pa,Pa], K[i+1, Pa])
  d[i+1,i+1] = K[i+1,i+1] - dot(K[i+1, Pa],a[i+1,Pa])
}
```

- We need to solve $n-1$ linear systems of size at most $m \times m$. Trivially parallelizable!
- Quadratic form:

```
qf(u,v,A,D) = u[1]*v[1] / D[1,1]
for(i in 2:n) {
    qf(u,v,A,D) = qf(u,v,A,D)
      + (u[i] - dot(A[i,N(i)], u[N(i)]))
        *(v[i] - dot(A[i,N(i)], v[N(i)]))/D[i,i]
    }
```

- Determinant: $\det(\tilde{K}_\theta) = \prod_{i=1}^{n}$ d[i,i]

Conjugate latent NNGP models: Finley et al., *JCGS*, 2019

$$[\text{data} \,|\, \text{process}] \times [\text{process} \,|\, \text{parameters}] \times [\text{parameters}] \,.$$

$$y(\ell_i) \overset{ind}{\sim} N(x(\ell_i)^\top \beta + w(\ell_i), \sigma^2 \delta^2)\ , i = 1, 2, \ldots, n$$
$$w = \{w(\ell_i)\} \sim N(0, \sigma^2 \tilde{M}); \quad \{\beta, \sigma^2\} \sim NIG(\mu_\beta, V_\beta, a_\sigma, b_\sigma)$$

Hierarchical linear model:

$$\underbrace{\begin{bmatrix} \frac{1}{\delta} y \\ L_\beta^{-1} \mu_\beta \\ 0 \end{bmatrix}}_{y_*} = \underbrace{\begin{bmatrix} \frac{1}{\delta} X & \frac{1}{\delta} I_n \\ L_\beta^{-1} & O \\ O & D^{-\frac{1}{2}}(I - A) \end{bmatrix}}_{X_*} \underbrace{\begin{bmatrix} \beta \\ w \end{bmatrix}}_{\gamma} + \underbrace{\begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix}}_{\eta}$$

The posterior distribution of γ and σ^2 is

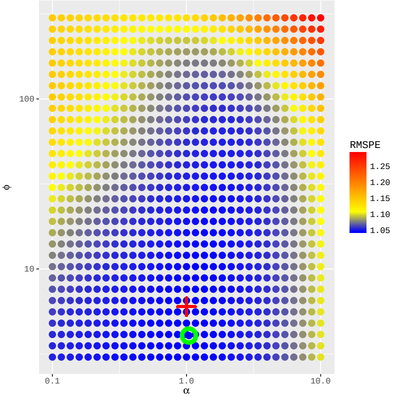
$$p(\gamma, \sigma^2 \,|\, y) \propto IG(\sigma^2 \,|\, a_*, b_*) \times N(\gamma \,|\, \hat{\gamma}, \sigma^2 (X_*^\top X_*)^{-1})$$

Storage and computational complexity $O(n(m+1)^2)$.

Choosing δ^2 and ϕ

- Fix spatial range ϕ and noise-to-signal ratio $\delta^2 = \tau^2/\sigma^2$
- ϕ and δ^2 are chosen using K-fold cross validation over a grid of possible values
- Unlike MCMC, cross-validation can be completely parallelized
- Resolution of the grid for ϕ and δ^2 can be decided based on computing resources available
- In practice, a reasonably coarse grid often suffices

Choosing δ^2 and ϕ



(d) RMSPE

Figure: Simulation experiment: True value (+) of (δ^2, ϕ) and estimated value (○) using 5-fold cross validation

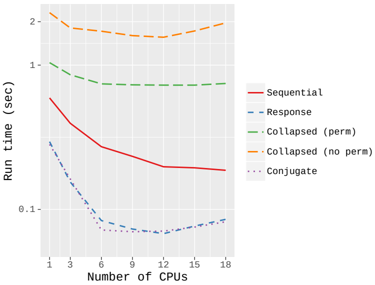
Alaska Tanana Valley data (Finley et al., *JCGS*, 2019)

	Conjugate NNGP	Collapsed NNGP	Response NNGP
β_0	2.51	2.41 (2.35, 2.47)	2.37 (2.31,2.42)
β_{TC}	0.02	0.02 (0.02, 0.02)	0.02 (0.02, 0.02)
β_{Fire}	0.35	0.39 (0.34, 0.43)	0.43 (0.39, 0.48)
σ^2	23.21	18.67 (18.50, 18.81)	17.29 (17.13, 17.41)
τ^2	1.21	1.56 (1.55, 1.56)	1.55 (1.54, 1.55)
ϕ	3.83	3.73 (3.70, 3.77)	4.15 (4.13, 4.19)
CRPS	0.84	0.86	0.86
RMSPE	1.71	1.73	1.72
time (hrs.)	0.002	319	38

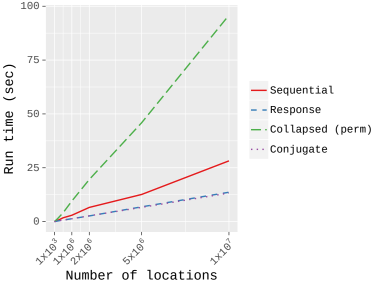
Table: Parameter estimates and model comparison metrics for the Tanana valley dataset

- Conjugate model produces estimates and model comparison numbers very similar to the MCMC based NNGP models
- For 5×10^6 locations, conjugate model takes 7 seconds

Comparison of computing times for different NNGP algorithms (Finley et al., *JCGS*, 2019)



(a)



(b)

Figure: (a) Run time required for one sampler iteration using n=5×10^4 by number of CPUs (y-axis is on the log scale). (b) Run time required for one sampler iteration by number of locations.

Concluding remarks

- Model-based solution for spatial "BIG DATA"
- Available R packages: spNNGP; BRISC GpGp; GpVecchia and meshed.
- Other softwares: exageostat
- Algorithms: Gibbs, RWM, HMC, VB or INLA; HMC is especially promising on RStan.
- Multivariate Geostatistics for large data:
  - Spatial factor models for large data sets (Taylor-Rodriguez et al., *Statistica Sinica*, 2019)
  - Conjugate NNGP models using Matrix-variate Normal-IW family (Zhang and Banerjee, *Biometrics*, 2021)
  - Graphical Gaussian Processes: (Dey et al., *Biometrika*, 2022)
- Enhance scalability using META-KRIGING approaches (Guhaniyogi and Banerjee, 2018)
- Challenges: Nonstationary models; High-dimensional outcomes; High-dimensional domains; Smoother process approximations.

NNGP using Hamiltonian Monte Carlo
http://mc-stan.org/users/documentation/case-studies/nngp.html

- The Metropolis-Hastings algorithm: Sample from any *target* probability density, e.g., posterior density $p(\theta \mid y) \propto p(\theta) \times f(y \mid \theta)$
- Start with a initial value for $\theta = \theta^{(0)}$. Repeat for $j = 1, 2, \ldots, M$:
  1. Propose $\theta^* \sim Q(\cdot \mid \theta^{(j-1)})$. For example, $Q(\cdot \mid \theta^{(j-1)}) = N(\cdot \mid \theta^{(j-1)}, \nu)$.
  2. Compute
  $$A(\theta^* \mid \theta^{(j-1)}) = \min\left(1, \frac{p(\theta^* \mid y) Q(\theta^{(j-1)} \mid \theta^*)}{p(\theta^{(j-1)} \mid y) Q(\theta^* \mid \theta^{(j-1)})}\right)$$
  3. *Accept* $\theta^{(j)} = \theta^*$ with probability $A(\theta^* \mid \theta^{(j-1)})$.
- MH works because it leaves the target invariant (satisfies detailed balance):
$$p(\theta \mid y) T(\theta' \mid \theta) = p(\theta' \mid y) T(\theta \mid \theta')$$
- Hamiltonian Monte Carlo: Use (discretized) Hamiltonian dynamics using *symplectic integrators* to propose in MH.

Thank You!