Software tools for space-time modelling

A comparison of WinBUGS and Stan

Pramoda Jayasinghe & Ying C MacNab
School of Population and Public Health
THE UNIVERSITY OF BRITISH COLUMBIA

5th August, 2024
Portland, OR

Motivation

- Different software tools exists for MCMC estimation (Štrumbelj et al., 2024).
  - BUGS, JAGS and Nimble (Gibbs sampling).
  - Stan, ADMB and PyMC (Hamiltonian Monte Carlo).
  - INLA (approximation).
- Selection of feasible tools for spatio-temporal (ST) estimation is unclear.
- Most studies uses one tool and comparison tools ST unavailable.
- Comparisons found:
  - Hazard ratio (HR) network meta analysis (NMA) (Jevdjevic et al., 2022).
  - Models in ecology (Monnahan et al., 2017).

RQ "Is the performance of WinBUGS and Stan similar for ST models?"

Models

$Y_{i,t}$ – Observed cases in area i at time t

$Y_{i,t}^{E}$ – Observed cases in area i at time t

$\gamma_{i,t}$ – Relative risk in area i at time t

μ_t – GLMM intercept

$\psi_{i,t}$ – GLMM random effect

$$Y_{i,t} \sim \text{Poisson}\left(Y_{i,t}^{E}\gamma_{i,t}\right);$$

$$\gamma_{i,t} = \exp\left(\mu_t + \psi_{i,t}\right)$$

Teal coloured parameters are fixed for the simulation.

Models. . .

- 2 forms assumed for GLMM random effects.
- Proper conditional autoregressive (pCAR) assumed for spatial and/or temporal components.

$AR(1)$ assumed:

$$\psi_{i,t} = \rho\psi_{i,t-1} + \epsilon_{i,t},$$
$$\boldsymbol{\epsilon}_t \sim \text{pCAR}\left(\lambda, \sigma_t^2\right).$$

Convolution assumed:

$$\psi_{i,t} = \eta_{i,t} + \zeta_{i,t},$$
$$\boldsymbol{\eta}_i \sim pCAR\left(\lambda^{(t)}, {\sigma^{(t)}}^2\right)$$
$$\boldsymbol{\zeta}_t \sim pCAR\left(\lambda^{(s)}, {\sigma_t^{(s)}}^2\right).$$

- $\boldsymbol{\epsilon}_t$ and $\boldsymbol{\zeta}_t$ for spatial relationships.
- $\boldsymbol{\eta}_i$ for temporal relationships.

Case study

Minnesota data

- 87 county data for Minnesota.
- Weekly aggregated case data ($\boldsymbol{y}$) over 10 weeks during the initial peak (30-Sep-2020 to 02-Dec-2020).
- Expected cases ($\boldsymbol{y}^{\boldsymbol{E}}$) are state-wide weekly cases distributed by population ratios.

$$\boldsymbol{y} = \begin{bmatrix} y_{1,1} & \cdots & y_{1,10} \\ & \vdots & \\ y_{87,1} & \cdots & y_{87,10} \end{bmatrix}_{87\times 10} \qquad \boldsymbol{y}^{\boldsymbol{E}} = \begin{bmatrix} y^E_{1,1} & \cdots & y^E_{1,10} \\ & \vdots & \\ y^E_{87,1} & \cdots & y^E_{87,10} \end{bmatrix}_{87\times 10}$$

Minnesota data...

- Spatial neighbourhood matrix ($\boldsymbol{W}^{(s)}$) extracted from boundaries.
  - Used for ϵ_t and ζ_t.
- Temporal neighbourhood matrix ($\boldsymbol{W}^{(t)}$) can be created.
  - Used for η_i.

$$\boldsymbol{W}^{(t)} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 1 & 0 & 1 & \dots & 0 & 0 \\ & & & \vdots & & \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}_{10\times10}$$

Model fit

AR(1) model:

Software	$\overline{D(\theta)}$	p_{DIC}	DIC	$-2lppd$	p_{WAIC}	$WAIC$
WinBUGS	6,507	656	7,163	6,243	829	7,073
OpenBUGS	6,510	654	7,164	6,246	834	7,079
Stan (full)	6,509	654	7,162	6,245	831	7,075
Stan (sparse)	6,511	657	7,168	6,246	834	7,080

Convolution model:

Software	$\overline{D(\theta)}$	p_{DIC}	DIC	$-2lppd$	p_{WAIC}	$WAIC$
WinBUGS	6,511	669	7,180	6,245	838	7,083
OpenBUGS	6,510	672	7,182	6,245	835	7,080
Stan (full)	6,493	675	7,168	6,235	811	7,046
Stan (sparse)	6,491	673	7,165	6,234	809	7,043

Run times - Based on ESS of RR

AR(1) model (100 thin):

Software	Time (min)[1]	Min ESS/min	Med ESS/min
WinBUGS	12.0	12.46	83.10
OpenBUGS[2]	4.1	36.21	241.40
Stan (full)	248.6	2.53	4.00
Stan (sparse)	7.3	100.10	136.39

Convolution model (300 thin):

Software	Time (min)[1]	Min ESS/min	Med ESS/min
WinBUGS	30.0	7.99	33.29
OpenBUGS	28.0	14.63	107.04
Stan (full)	2,529.4	0.29	0.39
Stan (sparse)	252.3	2.78	3.95

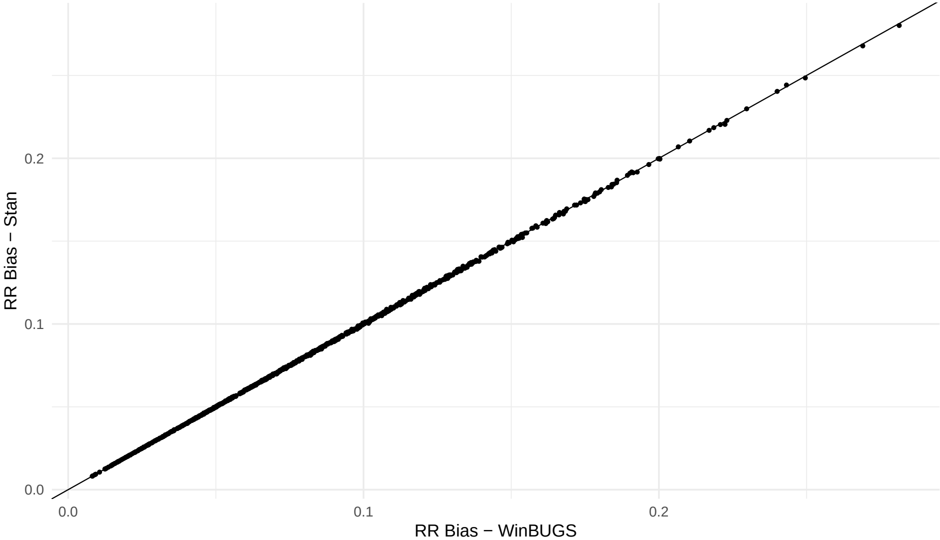
[1] AMD 7900x processor. WinBUGS 1.4.3, OpenBUGS 3.2.3 and CmdStan 2.34.1 executed through R 4.3.2.
[2] Vastly different results between runs. Seen median ESS/min in the 40s.

Simulation

Simulation setup

- Used to obtain performance of each method.
- Parameters selected close to actual data (not required).
- Run 500 simulations.
- Hyper-parameters and expected cases ($\boldsymbol{y}^{\boldsymbol{E}}$) fixed.
- Estimates recorded.
- Data generated from corresponding models.

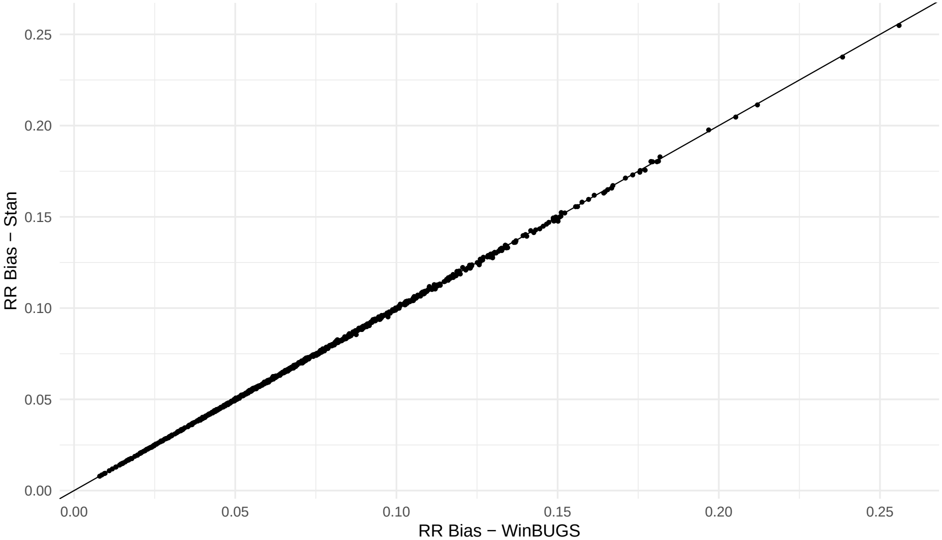
Simulation results - AR(1) model



Simulation results - AR(1) model

	Bias (RMSE)		95% coverage	
	WinBUGS	Stan	WinBUGS	Stan
λ	0.052 (0.052)	0.049 (0.049)	0.95	0.94
ρ	0.023 (0.023)	0.023 (0.023)	0.96	0.95
σ_{t1}	0.053 (0.053)	0.050 (0.050)	0.94	0.94
σ_{t2}	0.050 (0.050)	0.047 (0.047)	0.96	0.96
σ_{t3}	0.046 (0.046)	0.041 (0.041)	0.95	0.95
σ_{t4}	0.048 (0.048)	0.046 (0.046)	0.94	0.95
σ_{t5}	0.033 (0.033)	0.032 (0.032)	0.96	0.94
σ_{t6}	0.034 (0.034)	0.032 (0.032)	0.94	0.95
σ_{t7}	0.040 (0.040)	0.038 (0.038)	0.95	0.95
σ_{t8}	0.049 (0.049)	0.046 (0.046)	0.93	0.94
σ_{t9}	0.034 (0.034)	0.032 (0.032)	0.95	0.95
σ_{t10}	0.045 (0.045)	0.042 (0.042)	0.95	0.96

Simulation results - Convolution model



Simulation results - Convolution model

	Bias (RMSE)		95% Coverage	
	WinBUGS	Stan	WinBUGS	Stan
λ_t	0.086 (0.086)	0.095 (0.095)	0.99	0.95
σ_t	0.015 (0.015)	0.015 (0.015)	1.00	0.96
λ_s	0.101 (0.101)	0.095 (0.095)	0.97	0.95
σ_{s1}	0.063 (0.063)	0.061 (0.061)	0.96	0.94
σ_{s2}	0.052 (0.052)	0.050 (0.050)	0.97	0.95
σ_{s3}	0.044 (0.044)	0.046 (0.046)	0.99	0.97
σ_{s4}	0.046 (0.046)	0.043 (0.043)	0.97	0.95
σ_{s5}	0.023 (0.023)	0.023 (0.023)	1.00	0.99
σ_{s6}	0.032 (0.032)	0.037 (0.037)	1.00	0.98
σ_{s7}	0.032 (0.032)	0.031 (0.031)	0.97	0.97
σ_{s8}	0.040 (0.040)	0.040 (0.040)	0.96	0.95
σ_{s9}	0.035 (0.035)	0.034 (0.034)	0.98	0.96
σ_{s10}	0.047 (0.047)	0.044 (0.044)	0.97	0.95

Discussion

Summary

- Stan gives the lowest DIC and WAIC for the convolution model. WinBUGS (marginally) for the AR(1) model.

Summary

- Stan gives the lowest DIC and WAIC for the convolution model. WinBUGS (marginally) for the AR(1) model.
- Stan is computationally efficient for AR(1) and WinBUGS for convolution[1].

[1] Ignoring OpenBUGS.

Summary

- Stan gives the lowest DIC and WAIC for the convolution model. WinBUGS (marginally) for the AR(1) model.
- Stan is computationally efficient for AR(1) and WinBUGS for convolution[1].
- OpenBUGS is unpredictable in computational efficiency.

[1] Ignoring OpenBUGS.

- Stan gives the lowest DIC and WAIC for the convolution model. WinBUGS (marginally) for the AR(1) model.
- Stan is computationally efficient for AR(1) and WinBUGS for convolution[1].
- OpenBUGS is unpredictable in computational efficiency.
- Both tools fitted simulated data well, with Stan having slight more bias for λ_t in convolution.

[1] Ignoring OpenBUGS.

Summary

- Stan gives the lowest DIC and WAIC for the convolution model. WinBUGS (marginally) for the AR(1) model.
- Stan is computationally efficient for AR(1) and WinBUGS for convolution[1].
- OpenBUGS is unpredictable in computational efficiency.
- Both tools fitted simulated data well, with Stan having slight more bias for λ_t in convolution.
- Convolution model is harder to fit than the AR(1) model.

[1] Ignoring OpenBUGS.

Fundamental differences

- WinBUGS uses the conditional distribution to sample.
  - The conditional representation of the pCAR is computationally efficient.
  - Uses "local" information (only from neighbours).
  - Is slower (more iterations) to converge.
- Stan uses the joint distribution.
  - Using the full distribution is slow.
  - Efficiency improvements depends on the case (not general).
  - Uses "global" information (neighbours and non-neighbours).
  - Faster (less iterations) than WinBUGS for the same representation.
- WinBUGS codes are comparatively easier for beginners.
- Debugging is easier in Stan.

Some future work

- Fit the AR(1) model for the convolution data and vise-versa.
  - Cannot compare hyper-parameters as done here.
  - Can compare corresponding relative risks.
- Add covariates to measure effect of included covariates.
- Use of variational Bayes in Stan for complex and high dimensional ST models.

References

Jevdjevic, M., J. Youn, S. Petersohn, A. Gittfried, C. Ainsworth, and M. Piena (2022, December). MSR110 A Comparison of Stan Versus WinBUGS Software for Conducting Bayesian Hazard Ratio-Based Network Meta-Analysis. *Value in Health 25*(12), S371.

Joseph, M. (2016, August). Exact sparse CAR models in Stan.

MacNab, Y. C. (2022, August). Bayesian disease mapping: Past, present, and future. *Spatial Statistics 50*, 100593.

Monnahan, C. C., J. T. Thorson, and T. A. Branch (2017, March). Faster estimation of Bayesian models in ecology using Hamiltonian Monte Carlo. *Methods in Ecology and Evolution 8*(3), 339–348.

Štrumbelj, E., A. Bouchard-Côté, J. Corander, A. Gelman, H. Rue, L. Murray, H. Pesonen, M. Plummer, and A. Vehtari (2024, February). Past, Present and Future of Software for Bayesian Inference. *Statistical Science 39*(1).