
A TITLE

A PREPRINT

Ruoyong Xu

Department of Statistical Sciences
University of Toronto
700 University Ave., Toronto, ON M5G 1Z5, Canada
ruoyong.xu@mail.utoronto.ca

Patrick Brown

Department of Statistical Sciences
University of Toronto
700 University Ave., Toronto, ON M5G 1Z5, Canada
patrick.brown@utoronto.ca

ABSTRACT

some keypoints to mention:

- 1, anisotropic spatial models are important but difficult to evaluate because it consists of many covariance parameters.
- 2, standard approximations of spatial models (GMRF) don't allow for anisotropy (at least as implemented).
- 3, `gpuBatchMatrix` is able to compute the profile LogL for each of the parameters in linear anisotropic spatial models, for dense matrix, use no approximation methods. allow for uncertainty in covariance parameters
- 4, investigated likelihood-based CI's for variance parameters, wider than using information matrix based methods (`gepstatsp`)

Keywords First keyword · Second keyword · More

1 Introduction

- 1- cI for the betas for the `geostapsp` package
- 2- CI's using profile likelihoods from `gpuBatchMatrix` ... second CI's will be wider
- ... simulation study, profile CI's have better coverage

emphasis for paper 2? - anisotropic spatial models are important

- standard approximations of spatial models (GMRF) don't allow for anisotropy (at least as implemented).
- We do dense matrix, no approximations
- also, uncertainty in covariance parameters is important in anisotropic models because there are more of them.

- we should always allow for uncertainty in covariance parameters $\theta = \text{sd, nugget, range, ratio, angle, shape}$, BoxCox

- ... although most frequentist spatial software doesn't, assumes $\theta = \hat{\theta}$
- anisotropic models + boxcox + matern shape means many covariance parameters (??)
- ... important to understand uncertainty in covariance parameters, 2D profile L interesting
- ... allow for uncertainty in covariance parameters to be reflected in CI's for betas

- The easy way: - Hard way:

Likelihood-based CI's for variance parameters

We can estimate shape, range, variance
 ... likelihood might be fairly flat, CI's are wide
 ... but CI's have good coverage

2 The linear geostatistical model

We start with a review of the linear geostatistical model:

$$\begin{aligned} Y_i | U(s_i) &\overset{ind}{\sim} N(\lambda(s_i), \tau^2), \\ \lambda(s_i) &= X(s_i)\beta + U(s_i), \\ (U(s_1), \dots, U(s_n))^T &\sim \text{MVN}(0, \Sigma), \\ \Sigma_{ij} = \text{COV}[U(s_i), U(s_j)] &= \begin{cases} \sigma^2 R(\|s_i - s_j\|/\phi; \kappa), & i \neq j. \\ \sigma^2, & i = j. \end{cases} \end{aligned}$$

where $Y_i : i = 1, \dots, n$ is the observation obtained at location s_i . Given $U(s_i)$, Y_i 's are mutually independently distributed with Gaussian distribution with observation variance τ^2 . $X(s_i)$ is a $1 \times p$ vector of explanatory variables at location s_i , $\beta = (\beta_1, \dots, \beta_p)^T$ is the corresponding vector of regression parameters. $U = (U(s_1), \dots, U(s_n))^T$ is a zero-mean Gaussian random field with covariance matrix Σ . The covariance between locations (s_i) and (s_j) takes the above form, where $R(\cdot)$ is the correlation function, and σ^2 is the spatial variance or variability in residual variation.

Let $Y = (Y_1, \dots, Y_n)^T$ be a vector of observations, and $X = (X(s_1), \dots, X(s_n))$ be an $n \times p$ design matrix, we rewrite the model in matrix format

$$Y \sim \text{MVN}(X\beta, \sigma^2 R(\phi, \kappa) + \tau^2 I).$$

Write $V = R(\phi, \kappa) + \nu^2 I$, and $\nu^2 = \tau^2/\sigma^2$, then

$$Y \sim \text{MVN}(X\beta, \sigma^2 V). \quad (1)$$

2.1 Matérn correlation

The powered exponential family, the Matérn family and the spherical family are the three most commonly used families of correlation functions in geostatistics. We favour the Matérn family because of its flexibility, and it is made available in our R package *gpuBatchMatrix*. The form of Matérn correlation used in *gpuBatchMatrix* is

$$R(d; \phi, \kappa) = \frac{2^{\kappa-1}}{\Gamma(\kappa)} (\sqrt{8\kappa} \frac{d}{\phi})^\kappa K_\kappa(\sqrt{8\kappa} \frac{d}{\phi}), \quad \text{where } \phi > 0, \kappa > 0,$$

$K_\kappa(\cdot)$ denotes the modified Bessel function of the second kind of order κ , κ is the shape parameter which determines the smoothness of $U(x)$. For $\kappa = 0.5$, the Matérn correlation function coincides with the exponential correlation $\exp(-d/\phi)$. When $\kappa \rightarrow \infty$, it tends to the Gaussian correlation $\exp\{-2(\|d\|/\phi)^2\}$. ϕ is called the range or scale parameter, it controls the rate of decay of the correlation as d increases.

2.2 Box-Cox transformation

The fit of the linear geostatistical model can often be improved by applying a transformation to the response variable Y . Skewed random fields that mildly deviate from the Gaussian distribution can be modeled by means of the Box-Cox transformation (BCT) [1]. For positive-valued response variable Y , the Box-Cox transformation has the form

$$Y' = \begin{cases} (Y^\lambda - 1)/\lambda : & \lambda \in \mathbb{R} \quad \text{and} \quad \lambda \neq 0, \\ \log Y : & \lambda = 0, \end{cases}$$

where Y' denotes the transformed response. The aim of the BCT is to make data closely resemble the Gaussian distribution, so that the usual assumptions for linear models hold. The BCT cannot guarantee that any probability distribution can be normalized. However, even in cases when an exact mapping to the normal distribution is not possible, BCT may still provide a useful approximation [2].

2.3 Geometric anisotropy

Geometrical anisotropy is defined by two additional parameters in the correlation function. Let (x_1, x_2) represent the original coordinates, which is rotated counterclockwise by φ , then the new coordinates (x'_1, x'_2) is given by

$$\begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} = \begin{pmatrix} 1/\phi_1 & 0 \\ 0 & 1/\phi_2 \end{pmatrix} \cdot \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},$$

where $\varphi \in [-\pi/2, \pi/2]$ is called the anisotropy angle, $\psi = \phi_1/\phi_2 \in [0, 1]$ is the anisotropy ratio.

3 Inference based on the likelihood function

Let $y'^T = (y'_1, \dots, y'_n)$ be the Box-Cox transformed data of the observations $y^T = (y_1, \dots, y_n)$. The most general log-likelihood function for the linear geostatistical model parameters given y' would be

$$-2\ell(\beta, \sigma^2, \lambda, \omega; y') = (y' - X\beta)^T (\sigma^2 V)^{-1} (y' - X\beta) + \log |\sigma^2 V| - 2 * (\lambda - 1) \sum_{i=1}^n \log y_i + n \log(2\pi). \quad (2)$$

the third term on the right-hand side of (2) arises from the Jacobian of the transformation. This log-likelihood function is complex to evaluate as we have several types of parameters to estimate, including the regression parameters β , the transformation parameter λ , and the covariance parameters $(\sigma^2, \phi, \kappa, \nu^2, \varphi, \psi)$. For simplicity we use $\omega = (\phi, \kappa, \nu^2, \varphi, \psi)$ to denote the correlation parameters in the following.

3.1 Profile log-likelihood functions

The profile log-likelihood for parameter φ is defined as $\ell_p(\varphi; y) = \sup_{\psi} \ell(\varphi, \psi; y)$. Given ω and λ , The log-likelihood function (2) is maximized at

$$\hat{\beta}_{\omega, \lambda} = (X^T V^{-1} X)^{-1} X^T V^{-1} y', \quad \text{and} \quad (3)$$

$$\hat{\sigma}_{\omega, \lambda}^2 = \frac{1}{n} (y' - X \hat{\beta}_{\omega, \lambda})^T V^{-1} (y' - X \hat{\beta}_{\omega, \lambda}). \quad (4)$$

Substitute (3) and (4) back into (2), we have the profile log-likelihood for ω and λ

$$-2\ell_p(\omega, \lambda; y') = n \log \frac{(y' - X \hat{\beta}_{\omega, \lambda})^T V^{-1} (y' - X \hat{\beta}_{\omega, \lambda})}{n} + \log |V| - 2(\lambda - 1) \sum_{i=1}^n \log y_i + n \log(2\pi) + n, \quad (5)$$

The main problem of maximum likelihood estimation is that variance is underestimated. Think of the simple case when the covariance matrix of Y is $\sigma^2 I$, then

$$\hat{\sigma}^2 = \frac{1}{n} (y - X \hat{\beta})^T (y - X \hat{\beta}),$$

$$\mathbb{E}[\hat{\sigma}^2] = \frac{n-p}{n} \sigma^2 < \sigma^2,$$

where $p = \text{rank}(X)$, is the number of elements in β . The estimation bias comes from not taking into account the degree of freedom used for estimating β . Restricted maximum likelihood (REML) developed by [3] is an approach that produces unbiased estimators or less biased estimators than MLE in general. REML eliminates the influence of X on $\hat{\sigma}^2$ and $\hat{\omega}$. In this method, we find a matrix A of full rank such that $AX = 0$. The $n \times n$ projection matrix $S = I - X(X^T X)^{-1} X^T$ satisfies $SX = 0$, however, S is degenerate as $\text{rank}(S) = n - p$, so we take just $n - p$ linearly independent rows from S to make the matrix $A_{(n-p) \times n}$. Transform the data linearly to $Y^* = AY$, then $\mathbb{E}(Y^*) = AX\beta = 0$. Recall in (1) we obtained $Y \sim \text{MVN}(X\beta, \sigma^2 V)$, thus

$$Y^* \sim \text{MVN}(0, \sigma^2 A V A^T).$$

The principle of the REML method is to estimate the variance parameter σ^2 by maximizing the restricted log-likelihood function below, where y^* represents a realization of Y^* ,

$$-2\ell^*(\sigma^2, \omega; y^*) = y^{*T} (\sigma^2 A V A^T)^{-1} y^* + \log |\sigma^2 A V A^T| + (n - p) \log(2\pi), \quad (6)$$

Differentiating (6) with regard to σ^2 and setting it equal to zero gives the unbiased estimate for σ^2

$$\hat{\sigma}_{reml}^2(\omega) = \frac{y^{*\top}(AVA^\top)^{-1}y^*}{n-p}.$$

The REML criterion is based on the likelihood of Y^* , in which X does not appear. REML bypasses estimating β and can therefore, produce unbiased estimates for σ^2 .

We can write (6) in terms of the original observed data y using the two identities derived by [4] in Section 5.1:

$$|A(\sigma^2 V)A^\top| = |\sigma^2 V| |X^\top(\sigma^2 V)^{-1}X|$$

and

$$y^*(\sigma^2 AVA^\top)^{-1}y^* = (y - X\hat{\beta})^\top(\sigma^2 V)^{-1}(y - X\hat{\beta}),$$

where $\hat{\beta} = (X^\top V^{-1}X)^{-1}X^\top V^{-1}y$. Then the restricted log-likelihood of y is

$$-2\ell^*(\sigma^2, \omega; y) = (y - X\hat{\beta})^\top(\sigma^2 V)^{-1}(y - X\hat{\beta}) + (n-p)\log\sigma^2 + \log|V| + \log|X^\top V^{-1}X| + n\log(2\pi).$$

Applying the BCT on y we have the restricted log-likelihood in terms of the transformed response data y'

$$\begin{aligned} -2\ell^*(\sigma^2, \omega, \lambda; y') &= (y' - X\hat{\beta})^\top(\sigma^2 V)^{-1}(y' - X\hat{\beta}) + (n-p)\log\sigma^2 + \log|V| + \log|X^\top V^{-1}X| + n\log(2\pi) - \\ &\quad 2(\lambda-1)\sum_{i=1}^n \log y_i, \end{aligned} \quad (7)$$

which does not depend on the choice of matrix A .

Differentiate (7) with respect to σ^2 and set it equal to zero, we have the expression for $\hat{\sigma}_{reml}^2$ in terms of the Box-Cox transformed data given ω and λ ,

$$\hat{\sigma}_{reml}^2(\omega, \lambda) = \frac{(y' - X\hat{\beta})^\top V^{-1}(y' - X\hat{\beta})}{n-p}. \quad (8)$$

Substitute (8) into (7), leading to the profile restricted log-likelihood for (ω, λ)

$$-2\ell_p^*(\omega, \lambda; y') = (n-p)\log\hat{\sigma}^2 + \log|V| + \log|X^\top V^{-1}X| - 2(\lambda-1)\sum_{i=1}^n \log y_i + n\log(2\pi) + n-p. \quad (9)$$

Compared to the “ml” profile log-likelihood given in (5), the effective dimension is reduced from n of $\ell(\cdot)$ to $(n-p)$ of $\ell_p^*(\cdot)$, this distinction is important if p is large.

3.2 Parameter estimation

Geostatsp's methods

The following are two important theorems in likelihood-based inference, which lead to two ways of calculating likelihood-based confidence intervals (CIs). We write $I(\theta)$ for the expected Fisher's information matrix, and θ is the true parameter value.

Theorem 3.1. $\hat{\theta}(Y) \sim \text{MVN}(\theta, I(\theta)^{-1})$.

Theorem 3.2. $2\{\ell(\hat{\theta}(Y); Y) - \ell(\theta; Y)\} \sim \chi_p^2$, where p is the dimension of θ .

The *lgm()* function in **geostatsp** package estimates the correlation parameters $\omega = (\phi, \kappa, \nu^2, \varphi, \psi)$ and the transformation parameter λ by maximization of the likelihood. Specifically, *lgm()* uses the *optim* function to optimize the equation (5) (if *reml()*=FALSE) numerically over ω and λ to find $(\hat{\omega}, \hat{\lambda}) = \text{argmax}_{\omega, \lambda} \ell_p(\omega, \lambda; y')$. **geostatsp** treats $\hat{\omega}$ and $\hat{\lambda}$ as fixed and back substitute them into (3) and (4) to obtain the maximum likelihood estimates (MLEs) $\hat{\beta}_{\omega, \lambda}$ and $\hat{\sigma}_{\omega, \lambda}^2$. The *optim()* function returns the Hessian evaluated at $\hat{\omega}$ and $\hat{\lambda}$. Conventionally, *lgm()* calculates CIs based on

Theorem 3.1. The 95% confidence intervals for each of the parameters are calculated by $\hat{\theta}_i \pm 1.96\sqrt{\text{Var}(\hat{\theta}_i)}$, where

θ_i represents any of the parameter in $(\beta, \sigma, \omega, \lambda)$, and $\text{Var}(\hat{\theta}_i)$ is approximated by the diagonals of the inverse of the negative Hessian. This way of lnaelcac does not take into account of uncertainty in correlation parameters.

There is the other function *profLlgm()* in *geostatsp* that calculates confidence intervals or joint confidence regions based on the likelihood-ratio method 3.2. If c represents the β -quantile of χ_p^2 , i.e., $\mathbb{P}(\chi_p^2 \leq c) = \beta$, then the set $\{\theta : \ell(\theta) \geq \ell(\hat{\theta}) - c/2\}$ is asymptotically, a β -level confidence set for θ . For a fixed ϕ value, *optim()* evaluates (5) for different combinations of $(\kappa, \nu^2, \varphi, \psi)$ and finds the optim Log-likelihood value, repeat this process for a number of ϕ values to finally get the profile log-likelihood and MLE of ϕ . *profLlgm()* runs *optim()* a great many of times and thus is very slow.

gpuBatchMatrix's method

The hard way is by computing the profile log-likelihood $\ell_p(\beta) = \ell(\beta, \hat{\omega}_\beta, \hat{\sigma}_\beta, \hat{\lambda}_\beta)$. (QUESTION: cannot geostatsp uses optim for beta? is it too slow so geostatsp does not do that?)

SHOW A PROFILE LOG-LIKELIHOOD PLOT FOR BETA'S. (MEUSE DATA OR SWISSRAIN DATA?)
SHOW ALL LIKELIHOODS PLOT VS BETA'S

"The results R1 and R2 are invoked routinely in modern statistical practice, but they rely on assumptions that are not always satisfied. In a geostatistical context, the two most common situations in which these assumptions do not hold are boundary problems and dimensional ambiguities. "

3.3 CI's comparison for model parameters

SHOW A TABLE OF CI compare between *geostatsp* and *gpuBatchMatrix*

```
R> library(gpuRandom)
R> library(gpuR)
R> library(data.table)
R> library("geostatsp")

R> params
##      shape  range variance nugget anisoRatio anisoAngleRadians
## [1,]  1.83 20000         1  0.000      10.00          0.08
## [2,]  4.00 38619         1 15.000      16.00          0.70
## [3,]  3.00 90000         1  0.137      25.00          0.30
## [4,]  2.00 250000        1 20.000       8.09          0.50
## [5,]  0.20 15000         1  0.020      65.00          6.50
## [6,]  0.30 20000         1  0.080      45.00         12.50

## Error in gpuRandom::likfitLgmCov2d(data = swissRain, formula = rain ~ : unused
arguments (paramToEstimate = NULL, minustwotimes = FALSE, df = 1)
## Error in eval(expr, envir, enclos): object 'result5' not found
## Error in eval(expr, envir, enclos): object 'result5' not found
## Error in eval(expr, envir, enclos): object 'result5' not found
## Error in eval(expr, envir, enclos): object 'result5' not found
## Error in eval(expr, envir, enclos): object 'result5' not found
## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting
a method for function 'as.matrix': object 'result5' not found
## Error in as.vector(result5$detVar): object 'result5' not found
## Error in matrix(rep(detVar, Ndata), nrow = Nparam): object 'detVar' not found
## Error in as.vector(result5$jacobian): object 'result5' not found
## Error in integer(n): object 'Nparam' not found
```

```
## Error in eval(expr, envir, enclos): object 'Ncov' not found
## Error in eval(expr, envir, enclos): object 'XVYXVX' not found
## Error in eval(expr, envir, enclos): object 'XVYXVX' not found
## Error in eval(expr, envir, enclos): object 'Nparam' not found
## Error in eval(expr, envir, enclos): object 'Nparam' not found
## Error in matrix(XTVinvX[-selectedrows, -a], ncol = Ucov, byrow = TRUE): object
'XTVinvX' not found
## Error in matrix(XTVinvX[selectedrows, -a], nrow = Nparam, ncol = Ucov): object
'XTVinvX' not found
## Error in eval(expr, envir, enclos): object 'XVY' not found
## Error in eval(expr, envir, enclos): object 'XVY' not found
## Error in eval(expr, envir, enclos): object 'XTVinvX' not found
## Error in matrix(0, nrow = Nparam, ncol = Ndata): object 'Nparam' not found
## Error in matrix(0, nrow = Nparam, ncol = Ndata): object 'Nparam' not found
## Error in matrix(0, nrow = Nparam, ncol = Ndata): object 'Nparam' not found
## Error in matrix(0, nrow = Nparam, ncol = Ndata): object 'Nparam' not found
## Error in matrix(0, nrow = Nparam, ncol = Ndata): object 'Nparam' not found
## Error in eval(expr, envir, enclos): object 'Nparam' not found
## Error in matrix(0, nrow = m, ncol = Nparam): object 'Nparam' not found
## Error in eval(expr, envir, enclos): object 'ssqY' not found
## Error in colnames(All_min2loglik_forthisbeta) <- c("1", "2", "3", "4", : object
'All_min2loglik_forthisbeta' not found
## Error in matplot(Betas, All_min2loglik_forthisbeta, type = "l", lty = 1, : object
'All_min2loglik_forthisbeta' not found
## Error in colnames(All_min2loglik_forthisbeta): object 'All_min2loglik_forthisbeta'
not found
```

gpuBatchMatrix give wider ci and more accurate MLE's, especially for the shape parameter.

4 A simulation study

```
R> rm(list = ls(all = TRUE))
R> library("geostatsp")
R> Nsim = 35
R> set.seed(77)
R> mydat = SpatialPointsDataFrame(cbind(seq(100 * 1000, 140 * 1000, len = 500),
+   seq(100 * 1000, 120 * 1000, len = 500)), data = data.frame(cov1 = stats::rnorm(500,
+   120, 30), cov2 = stats::rpois(500, 50)))
R> # simulate a random field
R> trueParam = c(range = 1000, shape = 2, variance = 0.5^2, tauSq = 1^2,
+   anisoRatio = 1, anisoAngleRadians = 0)
R> nugget = trueParam["tauSq"]/trueParam["variance"]
R> names(nugget) <- "nugget"
R> nugget #4
## nugget
```

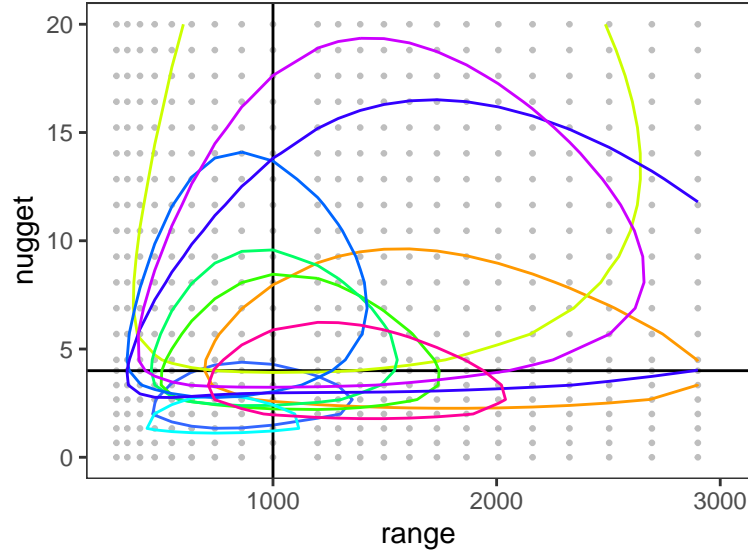


Figure 1: something

```
##      4
R> boxcox = c(boxcox = 1)
R> options(useRandomFields = FALSE)
R> for (i in 1:Nsim) {
+   set.seed(i * 8 - 1)
+   U <- geostatsp::RFsimulate(model = trueParam[setdiff(names(trueParam),
+     "tauSq")], x = mydat@data
+   # simulate response
+   Y <- 3 + 1 * mydat$cov1 + 0.5 * mydat$cov2 + U + stats::rnorm(length(mydat),
+     0, sd = sqrt(trueParam["tauSq"]))
+   mydat@data <- cbind(mydat@data, Y)
+ }
R> colnames(mydat@data) <- c("cov1", "cov2", paste("Y", c(1:Nsim), sep = ""))

R> count/Nsim
## [1] 0.714
```

5 Discussion

References

- [1] George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243, 1964.
- [2] Dionissios T Hristopulos. *Random fields for spatial data modeling: a primer for scientists and engineers*. Springer Nature, 2020.
- [3] H Desmond Patterson and Robin Thompson. Recovery of inter-block information when block sizes are unequal. *Biometrika*, 58(3):545–554, 1971.
- [4] SR Searle, RL Quaas, et al. A notebook on variance components: A detailed description of recent methods of estimating variance components, with applications in animal breeding. 1978.