

RECONSTRUCTION OF PAST EUROPEAN LAND COVER FROM POLLEN DATA

USING SPATIAL STATISTICS AND CRANK-NICOLSON
MONTE CARLO

LOVISA SVENSSON

Master's thesis
2019:E56



LUND INSTITUTE OF TECHNOLOGY
Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

TYP AV DOKUMENT		DOKUMENTBETECKNING
Examensarbete	Kompendium	LUTFMS-3382-2019
Delrapport	Rapport	

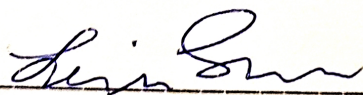
INSTITUTION
Matematikcentrum. Matematisk statistik, Lunds universitet, Box 118, 221 00 LUND
FÖRFATTARE
Lovisa Svensson
DOKUMENTTITEL OCH UNDERTITEL
Reconstruction of past European land cover from pollen data: using spatial statistics and Crank-Nicolson Monte Carlo
Given a pollen data set from Europe over a time period, the aim is to reconstruct the past land cover by interpolating from the pollen data values to a continuous map. The data is on compositional form with three vegetation categories; coniferous forest, broadleaved forest and open land. Reconstruction will be based on a Gaussian Markov random field with separable spatio-temporal structure for the covariance matrix. The spatio-temporal covariance matrix is constructed by Kronecker products which simplifies many matrix computations. The field and parameters for the model are estimated by Markov Chain Monte Carlo, with a Crank Nicolson Langevin proposal to estimate the spatio-temporal field. Crank Nicolson Langevin method works well, although implementation could be technical with a lot of details. Convergence for some of the model parameters is slow with bad mixing. The average compositional distance for the reconstruction and the validation set was 0.71. The model was better at finding temporal structure rather than spatial. Reconstructions from this model could be used as input to other models such as (Strandberg et al. 2014) to investigate how anthropogenic deforestation, and other changes in nature, impacts climate change.
NYCKELORD
DOKUMENTTITEL OCH UNDERTITEL - SVENSK ÖVERSÄTTNING AV UTLÄNDSK ORIGINALTITEL

UTGIVNINGSDATUM	ANTAL SID	SPRÅK
år 2019 mån 08	42	svenska <u>engelska</u> annat

ÖVRIGA BIBLIOGRAFISKA UPPGIFTER	ISSN
	ISBN
	2019 E:56

I, the undersigned, being the copyright owner of the abstract, hereby grant to all reference source permission to publish and disseminate the abstract.

Signature



Date 2019-09-06

Abstract

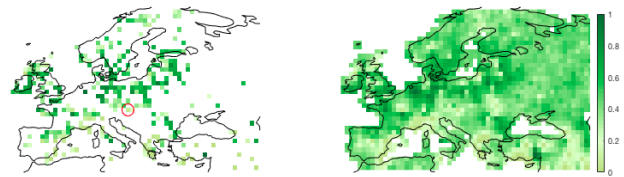
Given a pollen data set from Europe over a time period, the aim is to reconstruct the past land cover by interpolating from the pollen data values to a continuous map. The data is on compositional form with three vegetation categories; coniferous forest, broadleaved forest and open land. Reconstruction will be based on a Gaussian Markov random field with separable spatio-temporal structure for the covariance matrix. The spatio-temporal covariance matrix is constructed by Kronecker products which simplifies many matrix computations. The field and parameters for the model are estimated by Markov Chain Monte Carlo, with a Crank Nicolson Langevin proposal to estimate the spatio-temporal field. Crank Nicolson Langevin method works well, although implementation could be technical with a lot of details. Convergence for some of the model parameters is slow with bad mixing. The average compositional distance for the reconstruction and the validation set was 0.71. The model was better at finding temporal structure rather than spatial. Reconstructions from this model could be used as input to other models such as (Strandberg et al. 2014) to investigate how anthropogenic deforestation, and other changes in nature, impacts climate change.

Keywords: Pollen data, compositional data, Dirichlet distribution, spatio-temporal reconstruction, Kronecker product, Gaussian Markov random field (GMRF), Markov Chain Monte Carlo (MCMC), Metropolis Hastings (MH), Metropolis adjusted Langevin algorithm (MALA).

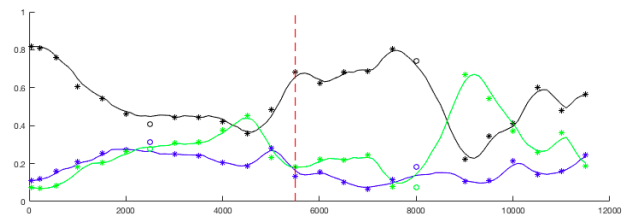
Nyframtagen pollendata används för att återskapa skogslandskap i Europa från senaste istiden fram till modern tid.

Ny pollendata¹ har blivit tillgänglig för att analysera skogslandskap i Europa från istid fram till modern tid. Med datamängder som blir allt större, ställs det krav på tekniken att få fram modeller som kan hantera dem på ett effektivt sätt. Man har nu med stor framgång lyckats återskapa skogslandskap över hela Europa från de senaste 10 000 åren.

Pollendatan kommer från sjöar och våtmarker runt om Europa. Datan har blivit viktad med hänsyn till pollenpartiklars fallhastighet och produktions nivå. Därefter har datan delats in i tre kategorier; barrskog, lövskog och öppet landskap². Varje datapunkt är kopplad till en latitud-longitud-ruta samt en tidpunkt. Problemet är att datan endast finns i omkring 15 % av alla latitud-longitud-rutor som utgör Europa, se figur 1. Utmaningen blir då att, baserat på datan, utvidga perspektivet till att ge en kontinuerlig bild av växtligheten över hela Europa. Vi förfinar också tidsintervallet genom att skatta vegetationen mellan de redan givna tidsnedslagen, se figur 2. Fullständiga skogslandskapsrekonstruktioner kan användas i andra arbeten, till exempel undersökningar om hur mänsklig avskogning och andra landskapsförändringar har påverkat klimatet³. I Figur 1 syns pollendatan till vänster som enstaka pixlar på en karta. Rekonstruktionen av skogslandskapet syns till höger. Här visas mängden lövskog som en procentsats av totalen, det vill säga, barrskog, lövskog och öppet landskap sammanlagt. Denna skogsåterskapning är gjord i mer än hundra tidpunkter, där pollendatan är given i endast 25 tidpunkter, se figur 2. I figur 2 syns tidsskalan för en utvald pixel över hela tidsaxeln, där både pollendatan och rekonstruktion är markerade.



Figur 1: Pollendata (till vänster) och rekonstruktion (till höger), av lövskog från året markerad med röd streckad linje i figur 2. Den röda cirkeln markerar den pixel som beskrivs i figur 2.



Figur 2: Stjärnorna (och cirkelarna) representerar modell- (och validerings-) pollendata för den inringade pixeln i figur 1, över tid. Linjerna representerar rekonstruktionen, där svart linje är öppet landskap, grön linje är lövskog och blå linje är barrskog. Tidsaxeln börjar i modern tid, 1950 v.t., och går bakåt i tiden. Den röda streckade linjen markerar året för tidsfönstret i figur 1.

Tid-rums fältet som utgör växtlighetsskattningarna, är modellerat med ett så kallat GMRF-fält där beroendestrukturen för tid och rum är separabla. Fältet har sedan skattats med MCMC, med en uppdateringsregel kallad Crank-Nicolson-Langevin. Mer om metoden hittas i den fullständiga rapporten⁴.

Skriven av: Lovisa Svensson.

¹Projekt Landclim II, finansierat av SRC och publicerat i databasen PANGEA.

²Trondman, A. K. et al. (2015), 'Pollen-based quantitative reconstructions of holocene regional vegetation cover (plant-functional types and land-cover types) in Europe suitable for climate modelling'

³Strandberg, G, et al. (2014), 'Regional climate model simulations for europe at 6 and 0.2 k bp: sensitivity to changes in anthropogenic deforestation'

⁴Svensson, L. (2019) 'Reconstruction of past European land cover from pollen data: using spatial statistics and Crank-Nicolson Monte Carlo'.

Aknowledgements

First and foremost I want to thank my supervisor Johan Lindström for the trust with this interesting and challenging project. I am very grateful for your patience, time and extensive knowledge in this topic. Thank you to PhD students and staff at the centre for mathematical statistics, for an always joyful time in the fika-room. Thank you to my lovely colleagues and new found friends in Das Gupta, who made this time very precious to me. To family and friends, thank you for always being by my side supporting me.

Contents

1	Introduction	1
1.1	Background	1
1.2	Aim and limitations	1
1.3	The data set	2
2	Model	5
2.1	Compositional data	5
2.2	The link function as the additive log-ratio transform (ALR)	5
2.3	Dirichlet distribution	6
2.4	The reconstruction, $\boldsymbol{\eta}$	6
2.5	Gaussian Markov random field, \boldsymbol{x}	7
2.6	The precision matrix, \boldsymbol{Q}	8
2.6.1	The precision matrices Q_s and Q_t	8
2.6.2	The matrices G_s and G_t	9
2.6.3	Prior distribution for the scaling parameter κ	9
2.6.4	The Kronecker product	10
2.6.5	Approximate minimum degree permutation, (AMD)	11
2.7	The mean coefficient $\boldsymbol{B}\boldsymbol{\beta}$	11
3	Estimating parameters with MCMC	13
3.1	Markov Chain Monte Carlo, (MCMC)	13
3.2	The target density for the Markov chain	13
3.3	Metropolis-Hastings, (MH)	14
3.3.1	Transition density for a random walk	14
3.3.2	Langevin diffusion	15
3.3.3	Metropolis adjusted Langevin algorithm, (MALA)	15
3.3.4	Crank Nicolson Langevin, (CNL)	16
3.3.5	Step size for proposals	17
3.4	Update of $\boldsymbol{\beta}$ and α with MALA	18
3.5	Update of parameters $\kappa_s, \kappa_t, \boldsymbol{\rho}_x$ and $\boldsymbol{\rho}_\beta$	18
3.5.1	Proposal for $\kappa, \boldsymbol{\rho}_x$	18
4	Implementations of the pollen data model with parameter estimation	21
4.1	Validation	22
4.2	Creating G_s and G_t	22
4.3	Reordering of matrices $G_s, G_s G_s$ and \boldsymbol{A}	23
4.4	Iterative update of mean and variance	23
4.5	Implementing CNL-function	24
5	Result	25
5.1	Parameters	25
5.2	Reconstruction	28
5.3	Validation	31

6	Discussion and conclusions	33
6.1	Crank Nicolson Langevin method	33
6.2	Convergence of parameters κ_s , κ_t and α	33
6.3	The average compositional distance	33
6.4	Further extensions of the model	34
A	Acceptance rate for MALA	37
B	More about Crank Nicolson Langevin	37
B.1	Why pCNL instead of CNL	37
B.2	Calculations for pCNL proposal	37
B.3	Acceptance rate for pCNL	38
B.4	Gradient of $\Phi(\mathbf{x})$	40
C	The joint posterior for κ, ρ_x	41
C.1	Posterior for ρ_β	41
D	Creating the observation matrix A	41

1 Introduction

1.1 Background

Based on pollen data, this thesis aims to reconstruct European land cover for the time period from the last Ice Age up to modern time. The pollen data comes from lakes and bogs and have then been analysed and categorised into three vegetation groups; coniferous forest, broadleaved forest and open land (Trondman et al. 2015). The data is on compositional form, i.e., each vegetation category is represented with a percentage.

Imagine a fine grided map over Europe for a specific year or *time window*, the given pollen data exist only in some of the grid cells. The intention is to interpolate over all grid cells given the grid cells with known values. The reconstruction will then be visualised with a heat map for the three vegetation fields. In this project we will extend the reconstruction (Pirzamanbein et al. 2018) from estimating only one time window to estimate many time windows simultaneously, over the given time era. Dependence structures in the estimation will be defined in both space and time, we call it spatio-temporal dependence (Blangiardo & Cameletti 2015). Such interpolation can be performed with a spatio-temporal Gaussian Markov random field. Parameter estimation will be performed with Markov chain Monte Carlo (Metropolis et al. 1953), where methods as Crank Nicolson Langevin will be used (Cotter et al. 2013, Beskos et al. 2008).

Land cover reconstructions are needed in works such as (Strandberg et al. 2014), which investigate how anthropogenic deforestation impacts climate change. Thus, complete data regarding regional vegetation is needed. Spatial interpolation with compositional data has been done before (e.g. Paciorek & McLachlan 2009, Billheimer et al. 2001, Tjelmeland & Lund 2003). In particular, a model with similar pollen data was implemented by Pirzamanbein et al. (2018). This model will serve as a foundation for the model developed in this thesis.

1.2 Aim and limitations

The main goal of this project is to, with spatial statistics and the Crank Nicolson Langevin method, reconstruct land cover in Europe from a given pollen data set. With a proper model we want to present a final reconstruction of the land cover and evaluate the model. We want to extend the previous model (Pirzamanbein et al. 2018) by

- considering temporal and spatial data simultaneously and
- implementing a Crank Nicolson Langevin method to estimate the spatio-temporal structure.

The aim of this thesis is practical rather than theoretical. The purpose here is not to mathematically prove any methods, the reader will find such results in the references.

1.3 The data set

The project will be based on a pollen data set from Landclim II.¹ The pollen data was gathered from lakes and bogs around Europe. Corrections for some biases caused by e.g. different fall speeds and production rates of pollen, have been performed by (Trondman et al. 2015).

The data spans from present time, 1950 CE, until the most recent Ice Age, roughly 11700 years ago. The data is given at 25 intervals during this time period. The midpoint of each time interval is used as time indexing for the modelling. Hence we use time points with start at year 50 until year 11500 before present time. The time steps are irregular. Time steps closest to modern time are smallest at 175 years, then they are increasing in length up to 500 years, for the earliest time steps.

In the space-plane, the data is divided into $1^\circ \times 1^\circ$ longitude/latitude grid cells. A space plane at one time point will be called a *time window*. There are almost 400 unique ($1^\circ \times 1^\circ$)-grid cells containing pollen data. However, each grid cell does not always have values for every time window. In total there are 7663 data points or *observations* in the time- space plane.

In figure 1, one pixel for all time windows is visible, the red circles in figure 2 specifies which pixel. Note how the time line in figure 1 is reversed, i.e., starts at present time and goes backwards in time. Here we see how there are more time points closer to modern time. One can also see that this time series is not complete since there is one sample missing at 8500 years before present. The observations are on compositional form i.e., the three vegetation categories in each pixel are represented by a value in the interval $(0, 1)$ and the three values sum up to one. All observations in the time window specified with the dashed red line in figure 1, are illustrated in figure 2.

¹Landclim project, funded by SRC and published in the PANGEA database.

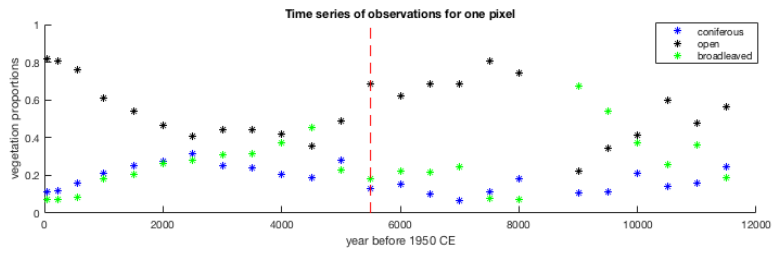


Figure 1: The time series of the three vegetation values for the circled pixel in figure 2. Note that the time line on the x-axis is reversed, i.e., starts at present time and ends in the earliest time era at 11500 years before present time. The dashed red line marks the time window shown in figure 2.

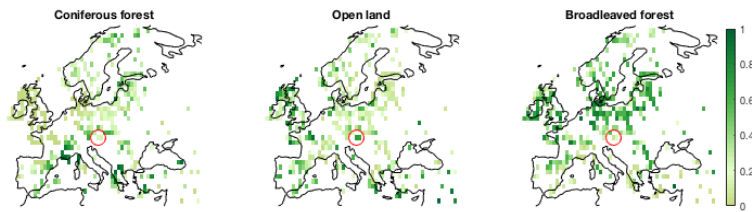


Figure 2: An illustration of pollen data from the time window represented with the dashed red line in figure 1. The three vegetation categories have values on compositional form. The red circles indicate the pixel illustrated in figure 1.

2 Model

We are going to set up a statistical model to estimate a reconstruction, $\boldsymbol{\eta}$, of the land cover. To measure how good the estimated reconstruction is, a probabilistic comparison will be done with the given observations, \boldsymbol{y} , of the pollen data set, see Section 2.3.

The reconstruction $\boldsymbol{\eta}$, discussed more in Section 2.4, consists of an abstract \boldsymbol{x} -field with spatio-temporal structure and mean value coefficients $\boldsymbol{\beta}$ which will be discussed in Sections 2.5 and 2.7 respectively. Additional parameters needed for the reconstruction will be presented along the way. For the abstract \boldsymbol{x} -field, there will be a detailed description of its covariance matrix, wherein the spatio-temporal structure lies, as accounted for in Section 2.6.

The focus throughout Section 2 will both be on explaining the role of each parameter of the model but also to present their parameter distributions, which will be of importance in Section 3, where the estimation of the parameters will be described.

2.1 Compositional data

There are N_{obs} number of observations in \boldsymbol{y} , which each have $D = 3$ different vegetation categories. The observations, \boldsymbol{y} , will consequently be of size $(N_{obs} \times D)$. One observation will represent one *pixel* having two space coordinates and one time coordinate.

The observation values for the vegetation categories are on a compositional form. Hence, the value that represents each class lies in the interval $(0, 1)$ and the three class values for each pixel sum up to one. If $\boldsymbol{y}_s = (y_{s,1}, y_{s,2}, \dots, y_{s,D})$, where $s = 1, \dots, N_{obs}$ indicates the site, then

$$y_{s,k} \in (0, 1) \text{ and } \sum_{k=1}^D y_{s,k} = 1.$$

2.2 The link function as the additive log-ratio transform (ALR)

When we do estimations for the reconstruction, $\boldsymbol{\eta}$, we do not want to have any limitation on $\boldsymbol{\eta}$ such as those caused by the compositional data. Instead we want $\eta_i \in \mathbb{R}$. Since the D number of compositional fields sum up to one, it is sufficient to only estimate $d = D - 1$ fields. To go from the reconstruction, $\boldsymbol{\eta}$, with d fields, to a reconstruction, \boldsymbol{z} , on compositional form with D fields, we have a link function

$$f(\eta_1, \dots, \eta_d) = z_1, \dots, z_d, z_D, \quad f : \mathbb{R}^d \mapsto [0, 1]^D.$$

There are many options for this link function. Here we follow Pirzamanbein et al. (2018) and use the additive log-ratio transform

$$z_i = f(\boldsymbol{\eta}) = \begin{cases} \frac{\exp \eta_i}{1 + \sum_i^d \exp \eta_i} & \text{for } i = 1, \dots, d \\ \frac{1}{1 + \sum_i^d \exp \eta_i} & \text{for } i = D \end{cases}, \quad \text{and} \quad (1)$$

$$\eta_i = f^{-1}(\mathbf{z}) = \log \frac{z_i}{z_D} \quad \text{for } i = 1, \dots, d. \quad (2)$$

2.3 Dirichlet distribution

It will be crucial to have a measure of how good the reconstruction, \mathbf{z} , is, i.e., how well it matches the observations, \mathbf{y} . This measure will be a probability. We will assume the observations \mathbf{y} , to be independent draws from a multivariate Dirichlet distribution (Pirzamanbein et al. 2018), conditioned on the latent field \mathbf{z} . Hence $\mathbf{y}|\mathbf{z}, \alpha \sim Dir(\alpha\mathbf{z})$, where α is a Dirichlet scale parameter. The Dirichlet probability density function for a single observation is given by

$$p(y_s|\alpha, z_s) = \frac{\Gamma(\alpha)}{\prod_{k=1}^D \Gamma(\alpha z_{s,k})} \prod_{k=1}^D y_{s,k}^{\alpha z_{s,k} - 1}, \quad \alpha > 0. \quad (3)$$

The assumption that each observation is independent, conditioned on the latent field \mathbf{z} , gives the following total probability for all of the observations

$$p(\mathbf{y}|\alpha, \mathbf{z}) = \prod_{s=1}^{N_{obs}} p(y_s|\alpha, z_s). \quad (4)$$

The scale parameter α , acts as an inverse variance parameter, $V(\mathbf{y}) \approx 1/\alpha$, with a low value of α indicating more uncertainty in the observations. We assume a gamma prior for α to get a Bayesian hierarchical model. Hence the parameter can be drawn from

$$\alpha \sim \Gamma(a_\alpha, b_\alpha). \quad (5)$$

Due to lack of intuition for α , the priors will be set to uninformative values; $a_\alpha = 1.5, b_\alpha = 0.1$ (Pirzamanbein et al. 2018).

2.4 The reconstruction, $\boldsymbol{\eta}$

The reconstructed field, $\boldsymbol{\eta}_{all}$, will consist of two main parts; the spatiotemporal structure, \mathbf{x} , and a mean parameter, $\mathbf{B}\boldsymbol{\beta}$. We have the complete estimated field as

$$\boldsymbol{\eta}_{all} = \mathbf{x} + \mathbf{B}\boldsymbol{\beta}. \quad (6)$$

If N is the total number of pixels to be estimated, the components $\boldsymbol{\eta}_{all}$, \boldsymbol{x} and $\mathbf{B}\boldsymbol{\beta}$ will be of size $(Nd \times 1)$. Figure 3 shows how all pixels in \boldsymbol{x} are stored and sorted after spatial, temporal and vegetation category indexing. Both \boldsymbol{x} and $\mathbf{B}\boldsymbol{\beta}$ will be further discussed in Sections 2.5 and 2.7.

The number of observed pixels, N_{obs} , is smaller than the total number of pixels, i.e., $N_{obs} \leq N$. Whenever we have a full reconstructed field $\boldsymbol{\eta}_{all}$, we need to extract only the observed pixels from it, in order to measure how good the estimation is. We extract the observed pixels in the reconstruction with an observation matrix \mathbf{A} , as follows

$$\boldsymbol{\eta}_{obs} = \mathbf{A}\boldsymbol{\eta}_{all},$$

where \mathbf{A} is a $N_{obs}d \times Nd$ sparse matrix with ones on the places where we have observations. How to construct \mathbf{A} is discussed in Appendix D. The reconstruction, $\boldsymbol{\eta}_{obs}$, can be transformed to compositional form by $\boldsymbol{z} = f(\boldsymbol{\eta}_{obs})$, which will be the conditioned latent field for the observations in the Dirichlet distribution (3).

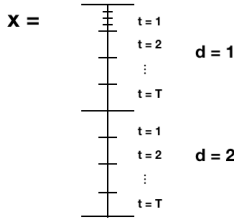


Figure 3: A diagram over the structure of \boldsymbol{x} . The vegetation fields, d , is the outer structure. The shortest horizontal lines in the cell $t = 1$ represent the spatial field for the first vegetation layer in the first time window, followed by the spatial field for the second time window, $t = 2$, etc.

2.5 Gaussian Markov random field, \boldsymbol{x}

The reconstruction (6) has one term, \boldsymbol{x} , representing the spatio-temporal structure. The structure will be modelled as a Gaussian Markov random field (GMRF). Thus, the field, \boldsymbol{x} , will be seen as draws from an underlying Gaussian Random field where the dependence between pixels is based on a certain neighbourhood structure including a Markov property, stored in an inverse covariance matrix \mathbf{Q} . We get the following prior model for \boldsymbol{x} ,

$$\boldsymbol{x} \sim \mathcal{N}(0, \mathbf{Q}^{-1}), \quad (7)$$

where \mathbf{Q}^{-1} is the covariance matrix (Rue & Held 2005). It follows that the probability density function for \boldsymbol{x} will be $p(\boldsymbol{x}) \propto \exp(-\frac{1}{2}\boldsymbol{x}^T \mathbf{Q}\boldsymbol{x})$, and will later on serve as a prior for the conditioned draws $\boldsymbol{x}|\boldsymbol{y}$.

2.6 The precision matrix, Q

The covariance matrix, Q^{-1} , in (7), or inverse covariance (precision) matrix, Q , will consist of three parts; spatial structure, temporal structure and the dependence between the fields describing different vegetation components. The field dependence will be a $d \times d$ covariance matrix called ρ_x , which captures dependency among and within the different fields. We denote the spatio-temporal covariance $Q_{s,t}^{-1}$. We assume the same dependence for all components and with the Kronecker product we get $Q^{-1} = \rho_x \otimes Q_{s,t}^{-1}$, or as a precision matrix, $Q = \rho_x^{-1} \otimes Q_{s,t}$ (Pirzamanbein et al. 2018). Further, we assume a separable spatio-temporal dependence in $Q_{s,t}$. Hence, the precision matrix, $Q_{s,t}$, can be decomposed with a Kronecker product between two other precision matrices, one consisting purely of spatial structure, Q_s , and one of temporal structure, Q_t . The Kronecker product gives the full spatio-temporal precision matrix, $Q_{s,t} = Q_t \otimes Q_s$ (Blangiardo & Cameletti 2015). The full precision matrix, including field dependence, then become

$$Q = \rho_x^{-1} \otimes Q_t \otimes Q_s. \quad (8)$$

We will in Sections 2.6.1 and 2.6.4 discuss construction of Q_s and Q_t and important calculation rules that apply to the Kronecker product. Firstly, we declare which prior assumptions are needed for the covariance matrix, ρ_x . The Inverse Wishart distribution is a commonly used as conjugate prior for covariance matrices and will hence serve as prior distribution for ρ_x . We draw

$$\rho_x \sim IW(a_\rho, b_\rho I_{d \times d}), \quad (9)$$

where $I_{d \times d}$ is the $d \times d$ identity matrix. The parameters of the prior, a_ρ and b_ρ , will be chosen as $a_\rho = 1$ and $b_\rho = 10$ (Pirzamanbein et al. 2018).

2.6.1 The precision matrices Q_s and Q_t

For a GMRF field defined on a regular grid, the precision matrix

$$Q = \frac{1}{\kappa^{2\nu}} (\kappa^4 I + 2\kappa^2 G + GG), \quad (10)$$

approximates fields with stationary Matérn covariance function, $r(\mathbf{h})$, where \mathbf{h} is all relative positions of two locations (Lindgren et al. 2011). Here, I is the identity matrix, κ, ν some constants and G is a finite difference approximation of the negative Laplacian (Lindgren et al. 2011). With one Q_s for the spatial structure and another Q_t for the temporal structure we will have different constants and matrices $\kappa_s, \kappa_t, \nu_s, \nu_t, G_s$ and G_t for the two precision matrices. The constant, ν , is given by $\nu = 2 - d_\kappa/2$ where d_κ is dimension of the coordinate space on which the GMRF is defined, e.g., $d_\kappa = 2$ for space and $d_\kappa = 1$ for time. The interpretation of κ as range, and how to estimate the parameter, are given in Section 2.6.3.

2.6.2 The matrices G_s and G_t

The finite difference approximation of the negative Laplacian, G , represent the spatial or temporal structure. Its size will equal the number of pixels to estimate; hence G_t is of size $n_t \times n_t$ and G_s of size $n_s \times n_s$. It will have number of neighbours for each pixel at the diagonal and -1 at the locations where each pixel has its neighbours. Hence the matrix will be symmetric and each row and column will sum to zero. The positive parameter, $\kappa^2 > 0$, guaranties that Q is a positive definite matrix. The G_t matrix will have the following structure

$$G_t = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & & & \\ 0 & -1 & 2 & \ddots & & \\ \vdots & & \ddots & \ddots & & \\ 0 & & & & 2 & -1 \\ 0 & & & & -1 & 1 \end{pmatrix}. \quad (11)$$

The shape of G_s will be the same as for Q_s , shown in figure 4 and will mainly have fours on the diagonal since almost all pixels have four neighbours. To create both G_t and G_s there are pre-written Matlab functions that we will use, as is further described in Section 4.2.

2.6.3 Prior distribution for the scaling parameter κ

The scaling parameter, κ , decides how strong dependence there is between each pixel and its neighbours. Low values of κ increases the dependence among neighbours. For estimation of κ , we will use the following scaled exponential probability density function as a prior

$$f_\kappa(\kappa; \lambda) = \left(\frac{1}{\kappa}\right)^{d_\kappa/2-1} \exp(-\lambda\kappa^{d_\kappa/2}), \quad (12)$$

with $d_\kappa = 2$ for the two dimensional spatial case and $d_\kappa = 1$ for the one dimensional temporal case, as suggested by Fuglstad et al. (2018). We have that λ is a constant decided by

$$\lambda = -\log(\alpha_0) \cdot \left(\frac{\rho_0}{\sqrt{8\nu}}\right)^{d_\kappa/2},$$

where $\alpha_0 = 0.01$, $\nu = 2 - d_\kappa/2$. The range, ρ_0 , will be set to $\rho_{0,s} = 5$ for the spatial case and $\rho_{0,t} = 10$ for the temporal case, in our model. The constant λ is determined by considering how unlikely short ranges are i.e., the prior probability of ranges less than ρ_0 is α_0 (Fuglstad et al. 2018).

2.6.4 The Kronecker product

Taking the Kronecker product of two matrices A and B , of sizes $m_A \times n_A$ and $m_B \times n_B$, gives a $m_A m_B \times n_A n_B$ matrix. The product is performed as; if

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad \text{then} \quad A \otimes B = \begin{pmatrix} aB & bB \\ cB & dB \end{pmatrix}. \quad (13)$$

for any matrix B (Blangiardo & Cameletti 2015, Fernandes & Plateau 1998). It follows that if A and B are symmetric, then $A \otimes B$ is symmetric. Some of the properties which apply for the Kronecker product is the associative law and the compatibility with ordinary matrix inversion and transpose

$$\begin{aligned} A \otimes (B \otimes C) &= (A \otimes B) \otimes C, \\ (A \otimes B)^{-1} &= A^{-1} \otimes B^{-1}, \\ (A \otimes B)^T &= A^T \otimes B^T. \end{aligned} \quad (14)$$

A Cholesky factor, R_A , of the Hermitian positive definite matrix, A , is a decomposition on the form $A = R_A^T R_A$. If \mathbf{A} is a Kronecker product $\mathbf{A} = B \otimes C$, then

$$\mathbf{R}_A = R_B \otimes R_C, \quad (15)$$

where the cholesky factors, R_B and R_C , can be calculated e.g. with the Matlab function `chol.m`. Further, the determinant for two square matrices on Kronecker form is given by

$$\det(A \otimes B) = \det(A)^{n_B} \det(B)^{n_A}. \quad (16)$$

where n_A and n_B are the sizes of A and B .

Solving matrix equations including a Kronecker product in the matrix, can be done without computing the full Kronecker product as in (13). Assume now matrices A , B , and X of sizes $m_A \times n_A$, $m_B \times n_B$, and $n_B \times n_A$. The solution is then given by

$$(A \otimes B) \cdot \text{vec}(X) = \text{vec}(BXA^T), \quad (17)$$

where $\text{vec}(X)$ is a column stacked vector of the matrix X . The solution BXA^T will be of size $m_B \times m_A$, but column stacked, i.e., $\text{vec}(BXA^T)$, which implies a vector $m_B m_A \times 1$. If A and B are quadratic, we can, using (14) and (17), solve the following equation

$$(A \otimes B)^{-1} \cdot \text{vec}(X) = (A^{-1} \otimes B^{-1}) \cdot \text{vec}(X) = \text{vec}(B^{-1} X A^{-T}). \quad (18)$$

There exist algorithms (e.g. Fernandes & Plateau 1998), that solves (17) and (18) for Kronecker products containing more then two matrices. The Matlab functions `kronmult.m` and `kronsolve.m`, written by Matthias Kredler, solves equation (17) and (18) based on algorithms from Fernandes & Plateau (1998).

2.6.5 Approximate minimum degree permutation, (AMD)

Before doing a Cholesky factorisation of a sparse matrix, it is very wise to do an Approximate minimum degree of permutation (AMD). The AMD, permutes rows and columns of the matrix in a beneficial way, which makes the calculations for the Cholesky factorisation faster. The Matlab function `amd.m`, written by Timothy A. Davis and others, could be used to extract a vector, p , indicating how the matrix should be permuted. In this project, we will perform AMD on the biggest and least sparse matrix $G_s G_s$. The index vector will be used to reorder G_s , $G_s G_s$, and the observation matrix \mathbf{A} , see more in Sections 4 and 4.3 how AMD is used in the model. In figure 4 we see the effects of reordering Q_s with help of the Matlab function `amd.m`. The number of elements in the Cholesky factorisation with reordered Q_s is reduced by a factor of ten.

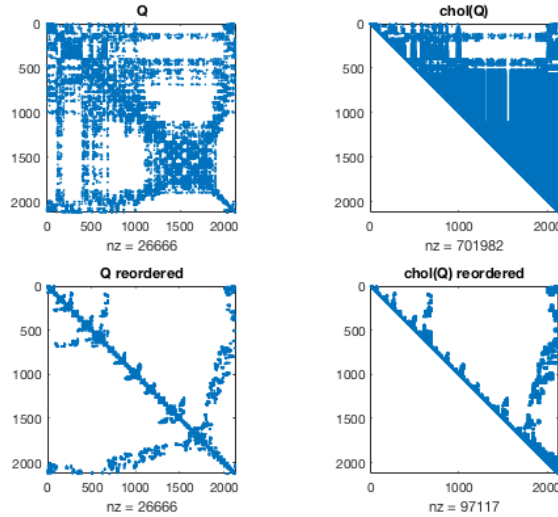


Figure 4: The symmetric sparse matrix Q_s (to the upper left) and its Cholesky decomposition (to the upper right). Reordered Q_s and Cholesky decomposed Q_s , according to the index vector $p = chol(Q_s)$, are given in the second row.

2.7 The mean coefficient $B\beta$

The parameter β , will be a mean value for each spatial field and represent the regression component for the reconstruction (6). Since we have n_t different temporal fields and each field has d vegetation fields, there will be a total of $n_t d$, β coefficients, hence β will be of size $n_t d \times 1$. We will assume a Gaussian distribution as a prior model for β ,

$$\beta \sim \mathcal{N}(0, \rho_\beta \otimes G_t G_t). \quad (19)$$

Here we use ρ_β , distinct from ρ_x , since we allow for different correlations in the spatio-temporal field, \mathbf{x} , and the mean regression coefficient, β . The covariance matrix, ρ_β , will be drawn in the same way as ρ_x , given in (9). The matrix $G_t G_t$ in (19) is equivalent with $Q_t(\kappa_t = 0)$, where $\kappa = 0$ corresponds to smoothing of the β , essentially an assumption of a slowly varying average vegetation composition.

The parameter β is of size $(n_t d \times 1)$ whilst η and \mathbf{x} is of size $(Nd \times 1)$. The \mathbf{B} matrix in (6) duplicates and maps the right β_k to every x_k in the reconstruction (6). Hence \mathbf{B} will be of size $(Nd \times dn_t)$ and is created as

$$\mathbf{B} = I_{n_t d \times n_t d} \otimes \mathbb{1}_{n_s},$$

where n_s is the number of spatial pixels in one time window, $I_{n_t d \times n_t d}$ is an identity matrix of size $n_t d \times n_t d$, and $\mathbb{1}_{n_s}$ is a column vector of ones of length n_s . An illustration of \mathbf{B} is given in figure 5. The \mathbf{B} matrix is a block diagonal matrix with column vectors of ones on the "diagonal" and zeros at the off-diagonals.

Figure 5: The \mathbf{B} matrix is a diagonal matrix with column vectors of ones at the "diagonal" and zeros at all other places. Here the vertical lines at the diagonal of \mathbf{B} represent column vectors of ones. Each column of ones at the diagonal maps one β_k to one time window stored in \mathbf{x} .

3 Estimating parameters with MCMC

In Section 2 we introduced all model components, including the parameters $\mathbf{x}, \boldsymbol{\beta}, \alpha, \kappa_s, \kappa_t, \boldsymbol{\rho}_x, \boldsymbol{\rho}_\beta$, and \mathbf{y} , and their probability distributions. Bayes theorem will be used as a link between all of the parameters and stating their dependence of each other. To get a good estimation of these parameters we will sample them many times from their distributions, we use Markov chain Monte Carlo (MCMC) with Metropolis Hasting proposals (Metropolis et al. 1953, Hastings 1970).

Section 3.1 gives a brief explanation regarding the principles of Markov Chain Monte Carlo. It is followed by Section 3.2 where the target density for the complete chain is described. In Section 3.3 with subsections, necessary theory for the sampling is presented. The theory will be essential when designing algorithms of how to update the parameters.

3.1 Markov Chain Monte Carlo, (MCMC)

The principle of Markov Chain Monte Carlo (MCMC) is to approximate a distribution by sampling many times from it. Each sample, u_k , for an arbitrarily parameter, is drawn from the assumed stationary distribution $\pi(u_{k+1} \in A|u_k)$. Together, all samples for one parameter create a Markov chain $\{u_n\}$. Reversibility in the chain implies stationarity. Stationarity together with ergodicity implies convergence for the mean of the chain. After convergence, the law of large numbers guarantees that the mean of the chain goes towards the true value. Hence the mean of the converged part of the chain, will be used as an estimation for a parameter.

We will sample from a *target density*, for which we can not directly guarantee stationarity. With a *transition density* for the Markov chain, we can make the target density match the stationary distribution.

3.2 The target density for the Markov chain

The target distribution, π , for the MCMC estimation in our model, is the complicated posterior distribution for all parameters given the observations, i.e., $\pi(\mathbf{x}, \boldsymbol{\beta}, \alpha, \boldsymbol{\kappa}, \boldsymbol{\rho}|\mathbf{y}) = p(\mathbf{x}, \boldsymbol{\beta}, \alpha, \boldsymbol{\kappa}, \boldsymbol{\rho}|\mathbf{y})$. With Bayes theorem, the target distribution can be written as

$$\begin{aligned} p(\mathbf{x}, \boldsymbol{\beta}, \alpha, \boldsymbol{\kappa}, \boldsymbol{\rho}|\mathbf{y}) &\propto p(\mathbf{y}|\mathbf{x}, \boldsymbol{\beta}, \alpha) \cdot p(\mathbf{x}, \boldsymbol{\beta}, \alpha, \boldsymbol{\kappa}, \boldsymbol{\rho}) \\ &\propto p(\mathbf{y}|\mathbf{x}, \boldsymbol{\beta}, \alpha) \cdot p(\mathbf{x}, \boldsymbol{\beta}|\boldsymbol{\kappa}, \boldsymbol{\rho}) \cdot p(\alpha, \boldsymbol{\kappa}, \boldsymbol{\rho}) \\ &\propto p(\mathbf{y}|\mathbf{x}, \boldsymbol{\beta}, \alpha) \cdot p(\mathbf{x}|\kappa_s, \kappa_t, \boldsymbol{\rho}_x) \cdot p(\boldsymbol{\beta}|\boldsymbol{\rho}_\beta) \cdot p(\alpha) \cdot p(\kappa_s)p(\kappa_t)p(\boldsymbol{\rho}_x)p(\boldsymbol{\rho}_\beta). \end{aligned}$$

Applying Gibbs sampling (Geman & Geman 1984), we divide the target density into three main blocks, in order to sample each block separately. The main blocks are

1. $\pi(\mathbf{x}) \propto \pi(\mathbf{x}, \boldsymbol{\beta}, \alpha, \kappa, \boldsymbol{\rho}|\mathbf{y}) \propto p(\mathbf{x}|\kappa, \boldsymbol{\rho}_x) \cdot p(\mathbf{y}|\mathbf{x}, \boldsymbol{\beta}, \alpha)$
2. $\pi(\alpha, \boldsymbol{\beta}) \propto \pi(\mathbf{x}, \boldsymbol{\beta}, \alpha, \kappa, \boldsymbol{\rho}|\mathbf{y}) \propto p(\boldsymbol{\beta}|\boldsymbol{\rho}_\beta) \cdot p(\alpha) \cdot p(\mathbf{y}|\mathbf{x}, \boldsymbol{\beta}, \alpha)$
3. $\pi(\kappa, \boldsymbol{\rho}) \propto \pi(\mathbf{x}, \boldsymbol{\beta}, \alpha, \kappa, \boldsymbol{\rho}|\mathbf{y}) \propto p(\mathbf{x}|\kappa, \boldsymbol{\rho}_x) \cdot p(\boldsymbol{\beta}|\boldsymbol{\rho}_\beta) \cdot p(\kappa_s)p(\kappa_t)p(\boldsymbol{\rho}_x)p(\boldsymbol{\rho}_\beta)$.

All three blocks will be sampled with the Metropolis-Hastings algorithm but with different kinds of proposals.

3.3 Metropolis-Hastings, (MH)

Markov chain Monte Carlo is a collection name for a set of different algorithms. One very commonly used MCMC algorithm is the Metropolis-Hasting (MH). The basic idea behind the Metropolis-Hastings (MH) algorithm is to generate a proposal, \hat{x} , given the previous sample x_t , and then either accept or reject the proposal with a certain probability α_{acc} . The new sample, x_{t+1} , thus become

$$x_{t+1} = \begin{cases} \hat{x} & \text{with probability } \alpha_{acc} \\ x_t & \text{otherwise} \end{cases} .$$

The proposal is generated from your choice of transition density $q(\hat{x}|x_t)$. Transition densities could be chosen in many ways. It is however important that the function α_{acc} is chosen such that the Markov chain, $\{x_1, \dots, x_t, x_{t+1}\}$, is reversible with respect to the target density π . A reversible chain implies that the target density is stationary (Rosenthal 2010). The Markov chain must also be ergodic to ensure convergence. The transition density together with the target density π , creates the acceptance probability α_{acc} according to

$$\alpha_{acc}(\hat{x}|x_t) = \min \left(\frac{\pi(\hat{x})q(x_t|\hat{x})}{\pi(x_t)q(\hat{x}|x_t)}, 1 \right). \quad (20)$$

3.3.1 Transition density for a random walk

A standard proposal when doing MH is to use your previous sample plus some noise, i.e., $\hat{x} = x_t + \xi$, where $\xi \sim \mathcal{N}(0, \sigma^2)$ (Roberts et al. 1997). This is also called *a random walk*. The transition density becomes

$$q(\hat{x}|x_t) \propto \exp \left(\frac{-1}{2\sigma^2} (\hat{x} - x_t)^2 \right). \quad (21)$$

For a random walk, it follows that the transition density becomes symmetric, i.e., $q(\hat{x}|x_t) = q(x_t|\hat{x})$. A symmetric proposal implies that the acceptance rate reduces to

$$\alpha_{acc}(\hat{x}|x_t) = \min \left(\frac{\pi(\hat{x})}{\pi(x)}, 1 \right). \quad (22)$$

It is important to find a good scaling for the proposal. A rule of thumb is to scale the proposal variance σ^2 in (21) so that the average acceptance rate is around 1/4 (Gelman et al. 1996, Roberts et al. 1997). With a too low variance, the chain converges and mixes slowly whilst a large variance leads to a high proportion of the proposed moves being rejected.

3.3.2 Langevin diffusion

When considering other proposals than random walk for MH, the optimal proposal scaling can be increased. With Langevin diffusion in the proposal, the optimal asymptotic acceptance rate proved to be 0.57 (Roberts & Rosenthal 1998). The following stochastic partial differential equation (SPDE) defines a diffusion with π as stationary density

$$d\mathbf{x} = \mathcal{K}\nabla \log \pi(\mathbf{x}) + \sqrt{2\mathcal{K}}\xi, \quad (23)$$

where \mathcal{K} is a symmetric positive definite matrix and ξ is Brownian white noise i.e., $\xi \sim \mathcal{N}(0, 1)$ (Roberts & Rosenthal 1998, Roberts & Stramer 2002, Rosenthal 2010). The SPDE is known to be reversible with respect to π and is further more assumed to be ergodic (Beskos et al. 2008). More about conditions for ergodicity of (23) is discussed in Roberts & Stramer (2002).

3.3.3 Metropolis adjusted Langevin algorithm, (MALA)

A common way of discretising differential equations is to use the forward Euler step

$$\frac{dx}{dt} \approx \frac{x_{t+1} - x_t}{h} = f(x_t), \quad (24)$$

where h is the step size and f is the diffusion to be discretised. Discretising (23) with a forward Euler step, where we use the shorter notations $\hat{\mathbf{x}} = \mathbf{x}_{t+1}$ and $\mathbf{x} = \mathbf{x}_t$, gives

$$\frac{\hat{\mathbf{x}} - \mathbf{x}}{h} = \mathcal{K}\nabla \log \pi(\mathbf{x}) + \sqrt{2\mathcal{K}}\xi. \quad (25)$$

We set the step size to $h = \delta^2/2$, and rearrange the terms, which gives the Metropolis adjusted Langevin algorithm (MALA)

$$\hat{\mathbf{x}} = \mathbf{x} + \frac{\delta^2}{2}\mathcal{K}\nabla \log \pi(\mathbf{x}) + \delta\sqrt{\mathcal{K}}\xi. \quad (26)$$

The step size, $\delta^2/2$, in the last term, was evolved to its squared root, $\delta/\sqrt{2}$, in order to be on right scale for the variance of the proposal. The proposal is then given by

$$\hat{\mathbf{x}}|\mathbf{x} \sim \mathcal{N}\left(\mathbf{x} + \frac{\delta^2}{2}\mathcal{K}\nabla \log \pi(\mathbf{x}), \delta^2\mathcal{K}\right) \quad (27)$$

The preconditioner, \mathcal{K} , can be chosen as any positive definite matrix. Common choices are either the identity matrix or some approximation of the curvature in $\log \pi$, e.g. the expected Fisher information, which gives $\mathcal{K}^{-1} = -E(\Delta \log \pi(\mathbf{x}))$ (Pirzamanbein et al. 2018). In Appendix A, you can read more about how the acceptance rate is derived. One can also see how the preconditioner for MALA becomes computational heavy when \mathbf{x} is big. The MALA proposal will be used as the update rule for parameters, α and β , corresponding the second block of the stationary distribution in Section 3.2. In Section 3.4 we explain further the specific proposal and acceptance rate for this.

3.3.4 Crank Nicolson Langevin, (CNL)

An alternative to the Metropolis adjusted Langevin algorithm, MALA, is instead the Crank Nicolson Langevin (CNL) algorithm. For CNL, we need a specific assumption for the stationary density, $\log \pi$, in (23). We will work specifically with the first block for the target density in Section 3.2,

$$\pi(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x}), \quad (28)$$

where the prior $p(\mathbf{x})$ is recognised from Section 2.5 to be $p(\mathbf{x}) \propto \exp(-\frac{1}{2}\mathbf{x}^T \mathbf{Q} \mathbf{x})$ and the likelihood $p(\mathbf{y}|\mathbf{x})$ is the Dirichlet distribution in (4). We will introduce the following expression for $p(\mathbf{y}|\mathbf{x})$,

$$p(\mathbf{y}|\mathbf{x}) \propto \exp(-\Phi(\mathbf{x})), \quad (29)$$

where $\Phi(\mathbf{x})$ can be seen as a potential (Cotter et al. 2013, Beskos et al. 2008). The posterior in (28) can then be written as

$$p(\mathbf{x}|\mathbf{y}) \propto \exp(-\Phi(\mathbf{x}) - \frac{1}{2}\mathbf{x}^T \mathbf{Q} \mathbf{x}). \quad (30)$$

We let $p(\mathbf{x}|\mathbf{y})$ in (30) be the stationary density, π , in (23). We derive $\nabla \log \pi = \mathbf{Q} \mathbf{x} + \nabla \Phi(\mathbf{x})$ and get the following SPDE for CNL

$$d\mathbf{x} = -\mathcal{K}(\mathbf{Q} \mathbf{x} + \nabla \Phi(\mathbf{x})) + \sqrt{2\mathcal{K}}\xi. \quad (31)$$

In the same way as for MALA, the Crank Nicolson Langevin method (CNL) aims to find a good Metropolis Hasting proposal by discretisation of a SPDE. This time we discretise (31) but with a Crank Nicolson step. A Crank Nicolson step takes an Euler's half forward and half backward step i.e.,

$$\frac{d\mathbf{x}}{dt} \approx \frac{\hat{\mathbf{x}} - \mathbf{x}}{h} = \frac{1}{2}[f(\mathbf{x}) + f(\hat{\mathbf{x}})],$$

with the the shorter notations $\mathbf{x} = \mathbf{x}_t$ and $\hat{\mathbf{x}} = \mathbf{x}_{t+1}$ (Crank & Nicolson 1947). Discretising only the linear part, $\mathbf{Q} \mathbf{x}$, in (31) with a Crank Nicolson step gives

$$\frac{\hat{\mathbf{x}} - \mathbf{x}}{h} = -\mathcal{K}(\mathbf{Q} \frac{1}{2}(\mathbf{x} + \hat{\mathbf{x}}) + \nabla \Phi(\mathbf{x})) + \sqrt{2\mathcal{K}}\xi,$$

whereas for the non linear part, $\nabla\Phi(\mathbf{x})$, we used the Euler forward step (Cotter et al. 2013, Beskos et al. 2008). Rearranging of the terms gives

$$\hat{\mathbf{x}} = \mathbf{x} + h \left(-\mathcal{K} \left(\mathbf{Q} \frac{1}{2} (\mathbf{x} + \hat{\mathbf{x}}) + \nabla\Phi(\mathbf{x}) \right) \right) + \sqrt{2h\mathcal{K}}\xi,$$

where the step size, h , in the last term, was evolved to its squared root in order to be on right scale for the proposal variance. Rearranging the terms again gives

$$\left(I + \frac{h}{2} \mathcal{K} \mathbf{Q} \right) \hat{\mathbf{x}} = \left(I - \frac{h}{2} \mathcal{K} \mathbf{Q} \right) \mathbf{x} - h \mathcal{K} \nabla\Phi(\mathbf{x}) + \sqrt{2h\mathcal{K}}\xi. \quad (32)$$

For the preconditioner, \mathcal{K} , in (32), we consider either $\mathcal{K} = \mathbf{I}$ or $\mathcal{K} = \mathbf{Q}^{-1}$. The first choice will be referred to as Crank Nicolson Langevin (CNL), and the other as preconditioned Crank Nicolson Langevin (pCNL). With CNL, the Kronecker form of big matrices cease due to the $(I + \frac{h}{2}\mathbf{Q})$ -term, hence pCNL is a better choice where the Kronecker form of the big matrices is preserved, see Appendix B.1. Therefore we will continue deriving the transition density only for pCNL. With $\mathcal{K} = \mathbf{Q}^{-1}$ we derive (32) in Appendix B.2 and get the following proposal

$$\hat{\mathbf{x}}|\mathbf{x} \sim \mathcal{N}(\zeta\mathbf{x} - (1 - \zeta)\mathbf{Q}^{-1}\nabla\Phi(\mathbf{x}), \omega^2\mathbf{Q}^{-1}), \quad (33)$$

where

$$\omega = \frac{2\sqrt{2h}}{2+h}, \quad \zeta = \sqrt{1 - \omega^2}.$$

The acceptance rate for the pCNL proposal (33), is derived in Appendix B.3 and the gradient, $\nabla\Phi(\mathbf{x})$, is derived in Appendix B.4. The pCNL proposal with corresponding acceptance rate, will be the update rule for the spatio-temporal field \mathbf{x} .

3.3.5 Step size for proposals

In the MALA and CNL proposal, there are discrete time steps, δ and h , that must be defined. The step length scales the proposal in a desired way. With a too small step the chain converges slowly whilst a large step leads to a high proportion of proposed moves being rejected. The step length is adjusted to obtain the target acceptance rate, which for MALA and CNL is $\alpha^* = 0.57$. A reasonable step length is received with the following adaptive MCMC update rule (Pirzamanbein et al. 2018)

$$h_{t+1} = h_t + \frac{1}{t^{1/2}}(\alpha_{acc} - \alpha^*) \quad (34)$$

where t is the index of which iteration we are at and α_{acc} is the acceptance probability at the current step.

3.4 Update of β and α with MALA

We will use MH sampling with a MALA proposal to update α and β . Hence the second block in section 3.2,

$$\pi(\beta, \alpha) \propto p(\beta|\rho_\beta) \cdot p(\alpha) \cdot p(\mathbf{y}|\mathbf{x}, \beta, \alpha), \quad (35)$$

will serve as target distribution for the MALA proposal. The MALA becomes

$$\hat{\theta}|\mathbf{y} \sim \mathcal{N}\left(\theta^T + \frac{\delta^2}{2}\mathcal{K}(\theta)\nabla \log \pi(\beta, \alpha), \delta^2\mathcal{K}(\theta)\right),$$

where $\theta^T = [\beta^T \alpha]$ is a vector. The three priors in (35) are recognised as (19), (5) and (3). The gradient $\nabla \log \pi(\beta, \alpha)$ and preconditioner, $\mathcal{K}(\theta)$, as the expected fisher information, were computed by Pirzamanbein et al. (2018). The step length, δ , is updated as in (34). Acceptance rate is accounted for in Appendix A, where we also discuss why MALA not is a good choice for updating of the \mathbf{x} -field in our model.

3.5 Update of parameters $\kappa_s, \kappa_t, \rho_x$ and ρ_β

The last block to update from Section 3.2 is

$$3. \quad \pi(\kappa, \rho) \propto p(\mathbf{x}|\kappa_s, \kappa_t, \rho_x) \cdot p(\beta|\rho_\beta) \cdot p(\kappa_s)p(\kappa_t)p(\rho_x)p(\rho_\beta). \quad (36)$$

This block can be divided into three smaller blocks

3. (a) $\pi(\kappa_s, \rho_x) \propto p(\mathbf{x}|\kappa_s, \kappa_t, \rho_x) \cdot p(\kappa_s)p(\rho_x)$
- (b) $\pi(\kappa_t, \rho_x) \propto p(\mathbf{x}|\kappa_s, \kappa_t, \rho_x) \cdot p(\kappa_t)p(\rho_x)$
- (c) $\pi(\rho_\beta) \propto p(\beta|\rho_\beta) \cdot p(\rho_\beta)$.

Block 3.a and 3.b, will be updated with the same Metropolis Hastings random walk proposal, described in Section 3.5.1. The target density, $\pi(\rho_\beta)$, in block 3.c will be updated at every iteration with draws from its posterior distribution, which is stated in Appendix C.1

3.5.1 Proposal for κ, ρ_x

We will work with the two posteriors $p(\kappa|\mathbf{x}) \propto (\mathbf{x}|\kappa, \rho_x)p(\kappa)$ and $p(\rho_x|\mathbf{x}) \propto (\mathbf{x}|\kappa, \rho_x)p(\rho_x)$. In Appendix C we see how we can marginalise over ρ_x by $p(\kappa|\mathbf{x}) \propto \int (\mathbf{x}|\kappa, \rho_x) \cdot p(\kappa)p(\rho_x)d\rho_x$. Therefore only $p(\kappa|\mathbf{x})$ will be sampled with a Metropolis-Hastings random walk, whilst $p(\rho_x|\mathbf{x}, \kappa)$ will be updated each time κ gets updated. We use a MH random walk in log-scale which gives the following proposal for κ

$$\log \hat{\kappa} = \log \kappa_i + h \quad (37)$$

where $h \sim \mathcal{N}(0, \sigma_\kappa^2)$. The variance, σ_κ^2 , is updated in a similar way as in (34) and with target acceptance rate $\alpha^* = 0.4$ (Pirzamanbein et al. 2018). The acceptance rate for the proposal can be found in Appendix C. If $\hat{\kappa}$ gets accepted, we update $\boldsymbol{\rho}_x$ by a drawing from the inverse Wishart distribution given in Appendix C, equation (52).

4 Implementations of the pollen data model with parameter estimation

In this section we provide the reader with more practical details concerning how the model was implemented. Below follows a list representing the implementation of the complete model with parameter estimation

1. Divide data into model and validation set.
2. Create G_s and G_t based on grids to which pollen data should be implemented.
3. Create \mathbf{A} based on model data, see Appendix D.
4. Reorder G_s , $G_s G_s$ and \mathbf{A} based on the vector $p = \text{amd}(G_s G_s)$.
5. Initialise priors and parameters
 - (a) Save desired parameters. Here every 1000th sample was saved due to memory management.
 - (b) Iteratively calculate mean and variance of desired parameters.
 - (c) Calculate \mathbf{Q} as in equation (8) and calculate \mathbf{R} by equation (15).
 - (d) With CNL, update \mathbf{x} .
 - (e) With MALA, update α and β .
 - (f) With MH-random walk, update κ_s .
 - (g) With MH-random walk, update κ_t and ρ_x .
 - (h) Update ρ_β .
7. Undo reorder of $\bar{\mathbf{x}}$
8. Create the final reconstruction $\boldsymbol{\eta}_{all} = \bar{\mathbf{x}} + \mathbf{B}\bar{\boldsymbol{\beta}}$ and transform the reconstruction to compositional form $\mathbf{z} = f(\boldsymbol{\eta}_{all})$ by equation (1).
9. Plot and validate the result.

Some of the steps in the list will be explained further in sections that follows. For the initialisation in step 4, the \mathbf{x} -field and the β parameters were initialised as draws from $\mathcal{N}(0, 0.1)$, remaining parameters were initialised with some arbitrarily but reasonable values. Details for the implementation of step 5.d will be accounted for in Section 4.5. Steps 5.e-5.h were implemented according to algorithms described in Section 3.4 and 3.5

4.1 Validation

In order to measure how well the model performs, the data set was divided into model data and validation data. To measure the difference in temporal and spatial estimation, two complete time windows and 19 complete time series were picked out for validation. Out of the 25 time windows, the 7th and 18th time window were picked out for validation. The 19 time series were picked at random. In total, 12.78 % of all pollen data was used for validation.

After estimations based only on the model data, the validation data set was used for comparison. The comparison was done by computing the average compositional distances (ACD). The ACD, for each location is given by

$$ACD(\eta, u) = ((\eta - u)^T J^{-1} (\eta - u))^{1/2}$$

where η is the reconstruction, u is the additive log-ratio transform of the validation observation, $u = f^{-1}(y_{val})$ and J is a $d \times d$ matrix with 2 on the diagonal and 1 on the off-diagonals (Pirzamanbein et al. 2018). The ACD was then averaged over all locations.

4.2 Creating G_s and G_t

Two separated grids were created, one two-dimensional spatial grid and another one-dimensional temporal grid. The size of the spatial grid was created out of the minimum and maximum longitude and latitude in the pollen data set. We got the sizes

$$\begin{aligned} n_{s1} &= \max_{lat} - \min_{lat} + 1 = 70.5 - 35.5 + 1 = 36 \\ n_{s2} &= \max_{long} - \min_{long} + 1 = 47.5 - (-10.5) + 1 = 59 \\ n_s &= n_{s1} * n_{s2} = 36 * 59 = 2124. \end{aligned}$$

A rectangular grid was created out of the two vectors $(1, 2, \dots, n_{s1})$ and $(1, 2, \dots, n_{s2})$. The coordinates of this grid were then extracted by a Matlab function `ndgrid.m`, to create the $n_s \times n_s$ Matérn precision matrix G_s , with help of another pre-written Matlab function `matern_prec_matrices.m` created by Johan Lindström. The pollen data span over the time $\max_{time} = 11500$ and $\min_{time} = 50$, which is given in years before present time 1950 CE. For the temporal grid, we chose a time step of 100 years. The total number of time windows to estimate became

$$n_t = \max_{time}/100 + 1. \tag{38}$$

The $n_t \times n_t$ matrix G_t is then created by pre-written Matlab function `createG.m` by Johan Lindström.

4.3 Reordering of matrices G_s , $G_s G_s$ and \mathbf{A}

As shown in Section 2.6.5, the AMD permutation speeds up the complete model estimations by about ten times, which is why this is something important to include in the implementation. AMD permutation was performed on the matrix $G_s G_s$ of size $n_s \times n_s$. With the index vector, $p = \text{amd}(G_s G_s)$, we rearranged the matrices as $G_s(p, p)$ and $G_s G_s(p, p)$. The observation matrix \mathbf{A} , also needed to be reordered, is however of size $N_{obs}d \times Nd$. Hence the vector p of length n_s needs to be extended, which was done in the following way

1. $p = \text{amd}(G_t G_t)$
2. $p_{t,s} = p$
3. Loop $n_t - 1$ times:
 - (a) $p_{t,s} = [p_{t,s} \quad p + \text{length}(p_{t,s})]$
4. $p_{t,s,d} = [p_{t,s} \quad p_{t,s} + \text{length}(p_{t,s})]$

The columns of \mathbf{A} was then reordered with this new indexing. After the MCMC estimation, the field \mathbf{x} was reordered back to normal order.

4.4 Iterative update of mean and variance

Estimation with MCMC of fields and parameters, were done 10^6 times, hence it became impossible to store all samples. We implemented an iterative update of mean and variance for the parameters in following way

$$\begin{aligned}\bar{x}_{t+1} &= \bar{x}_t + (x_{t+1} - \bar{x}_t)/t, \\ \sigma_{t+1}^2 &= \sigma_t^2 + ((x_{t+1} - \bar{x}_t) * (x_{t+1} - \bar{x}_{t+1}) - \sigma_t^2) / t,\end{aligned}$$

with t being iteration after burn-in.

4.5 Implementing CNL-function

All necessary calculations for the CNL proposal for the \mathbf{x} -field have been given in previous sections. However, a summary below of the necessary steps in the CNL algorithm will be presented. At one iteration for the MCMC, the CNL function consist of the following steps

1. Calculate $\boldsymbol{\eta}_{obs} = \mathbf{A}(\mathbf{x} + \mathbf{B}\boldsymbol{\beta})$.
2. Calculate $\mathbf{z} = f(\boldsymbol{\eta}_{obs})$ by equation (1).
3. Calculate $\Phi(\mathbf{x}) = -\log p(\mathbf{y}|\mathbf{z}(\mathbf{x}), \alpha)$ by equation (4) and $\nabla\Phi(\mathbf{x})$ as in Appendix B.4.
4. Draw $\boldsymbol{\xi} \sim \mathcal{N}(0, 1)$
5. Calculate $\hat{\mathbf{x}}$ by (44) in Appendix B.2.
6. Calculate $\hat{\boldsymbol{\eta}}_{obs} = \mathbf{A}(\hat{\mathbf{x}} + \mathbf{B}\boldsymbol{\beta})$.
7. Calculate $\hat{\mathbf{z}} = f(\hat{\boldsymbol{\eta}}_{obs})$ by (1).
8. Calculate $\Phi(\hat{\mathbf{x}})$ by (4) and $\nabla\Phi(\hat{\mathbf{x}})$ as in Appendix B.4.
9. Calculate α_{acc} as described in Section B.3.
10. Update h_{i+1} according to (34) for the next iteration.
11. Return $\hat{\mathbf{x}}$ with α_{acc} probability, else return \mathbf{x} .

5 Result

Because of poor convergence of the parameters κ_s, κ_t and α , three different model versions were tried

- Model A: All parameters were estimated
- Model B: Parameter α was fixed.
- Model C: Parameters κ_s and α were fixed.

When fixating parameters, we used results from Pirzamanbein et al. (2018); $\kappa_s = 0.22$ and $\alpha = 10.22$. In section 5.1, convergence plots for all parameters are presented. In Section 5.2 the reconstructions of all models are illustrated and in Section 5.3, model validation is discussed. Since Model A had the best ADC, the focus will lie on that model.

5.1 Parameters

In figure 6, the convergence of the three parameters κ_s, κ_t and α , for the three models A, B and C are shown. For Model A, the parameter α , initially reaches a high magnitude before decreasing again. The α -parameter shows good mixing even though convergence is slow. Both κ parameters show poor mixing and poor convergence. The parameter κ_s increases to a value, > 1 , whilst κ_t goes towards something closer to 0.

Because of the high α -value in Model A, Model B were designed with fixed $\alpha = 10.22$. Another model, Model C, was then designed to try the model with κ_s fixed at a relatively low value $\kappa_s = 0.22$, here α was kept fixated at $\alpha = 10.22$. For Model B, the parameter, κ_s , gets smaller but is still around 1, whilst κ_t gets to an even lower value than for Model A. For Model C, we succeed in raising the κ_t -value. No parameters manage to converge for any of the models.

Figure 7 shows the convergence and mixing for ρ_x, ρ_β , one x -pixel, and one β -value, for Model A. There is good mixing in both \mathbf{x} and β . The covariance matrix, ρ_x , reaches a peak relatively fast but is then sinking some. This could be a consequence of the non convergence of the parameters κ_s and κ_t , which ρ_x is conditioned on. The covariance matrix, ρ_β , only conditioned on β , converges immediately. In the plot for ρ_β , the first iteration is cut out. In figure 8, adaptive step lengths for CNL and MALA and the proposal variance for κ_s and κ_t are illustrated.

The mean of the parameters and 95 % confidence interval (CI) is shown in table 1 for Model A. To illustrate how strong the spatial and temporal dependence become for Model A, one can study figure 9, where the spatial and temporal dependence for the middle pixel and the middle time window are given.

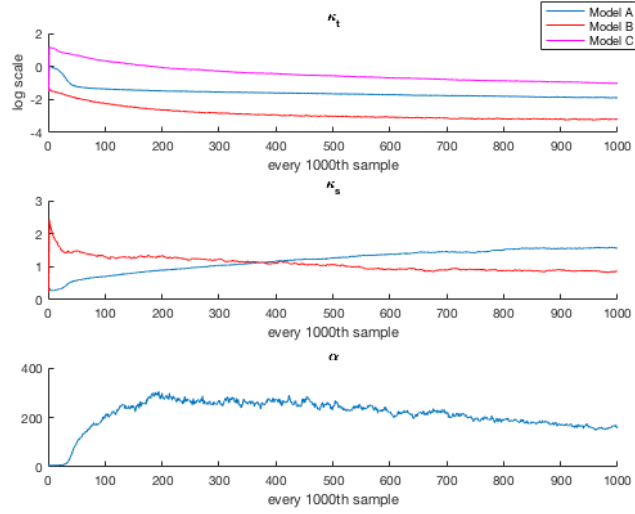


Figure 6: Parameter estimation for the three models A, B and C.

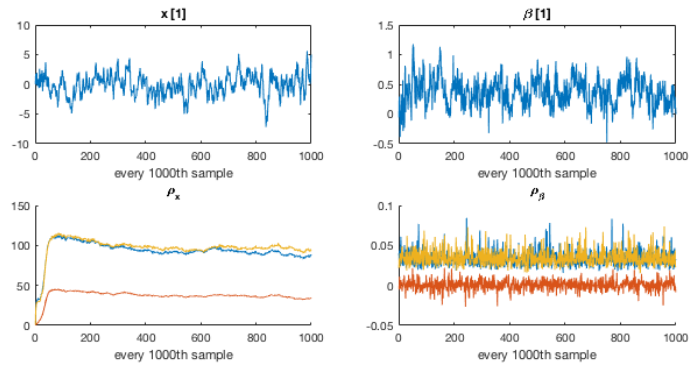


Figure 7: Parameter estimation for every 1000th sample for Model A. In the plots for ρ_x and ρ_β , the diagonal elements are shown with blue and yellow colour and the off-diagonal element is shown in red.

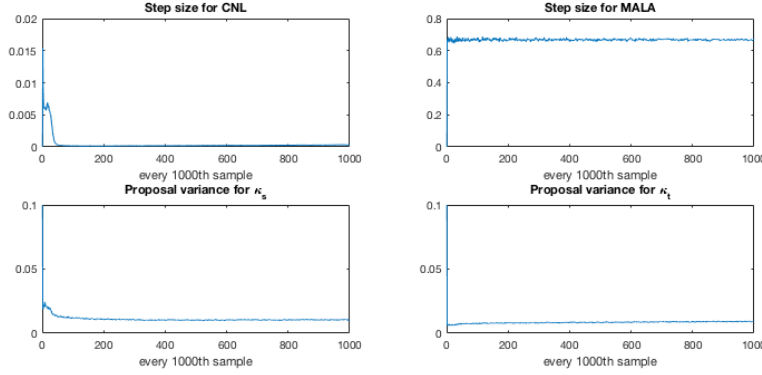


Figure 8: Adaptive step sizes for proposals in Model A.

Table 1: Parameter estimation for Model A; mean and 95 % confidence interval.

Parameter	mean		(CI)	
α	199		(145, 253)	
κ_s	1.46		(1.29, 1.64)	
κ_t	0.17		(0.15, 0.19)	
ρ_x	91.2	36.3	(85.2, 97.3)	(33.3, 39.3)
	36.3	96.3	(33.3, 39.3)	(91.8, 100.9)
ρ_β	$3.38 \cdot 10^{-2}$	$3.47 \cdot 10^{-4}$	$(1.61, 5.18) \cdot 10^{-2}$	$(-1.22, 1.28) \cdot 10^{-2}$
	$3.47 \cdot 10^{-4}$	$3.45 \cdot 10^{-2}$	$(-1.22, 1.28) \cdot 10^{-2}$	$(1.62, 5.26) \cdot 10^{-2}$

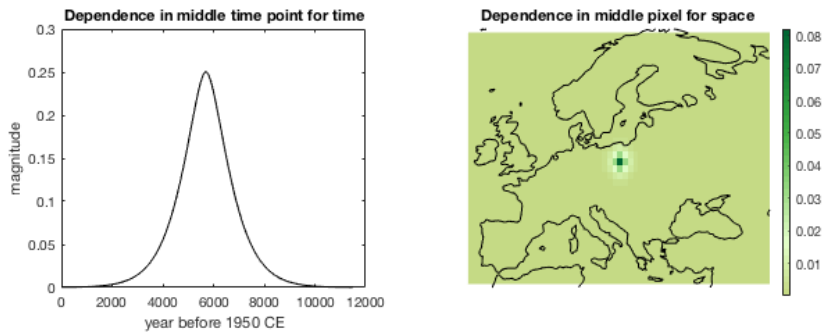


Figure 9: Covariance for time (to the left) and space (to the right) for Model A.

5.2 Reconstruction

Figure 10 shows the reconstruction of one time window for Model A, together with the model data. The high value of α implies low uncertainty in the observations. This, together with low spatial dependence due to high value of κ_s , causes little smoothness in the fields. In figure 11 and 12 the corresponding reconstructions for model B and C are shown for comparison.

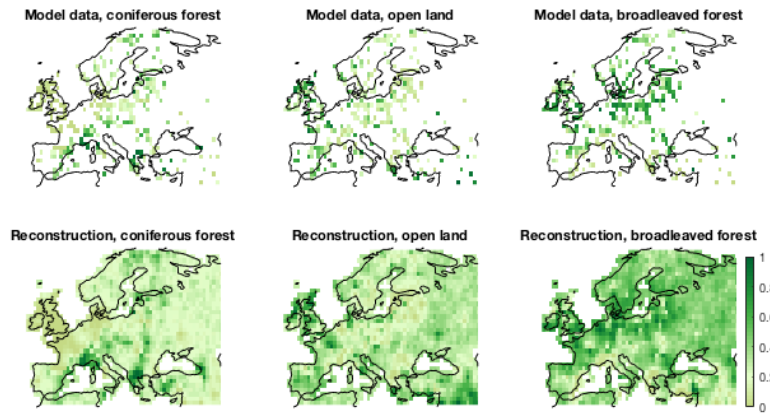


Figure 10: The reconstruction, z , in one time window for Model A together with the model data.

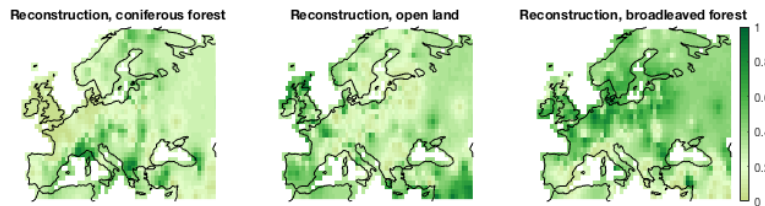


Figure 11: The reconstruction, z , in one time window for Model B.

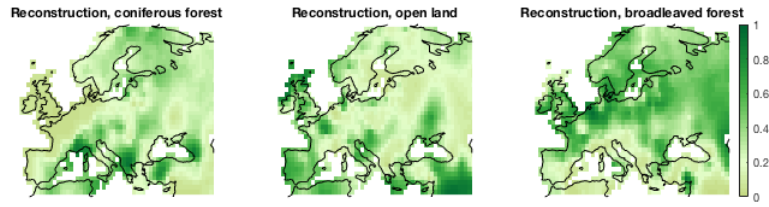


Figure 12: The reconstruction, z , in one time window for Model C

In figure 13 one can see the time series for the z -reconstruction in one pixel for Model A. The high value of α implying high certainty in the observations, together with high temporal dependence due to the low value of κ_t , are causing good smoothness in the reconstruction where the curve hits almost every observation. The reconstruction comes close to the two validation data points marked as circles in the plots. In figure 14 and 15 the corresponding reconstructions for model B and C are shown for comparison.

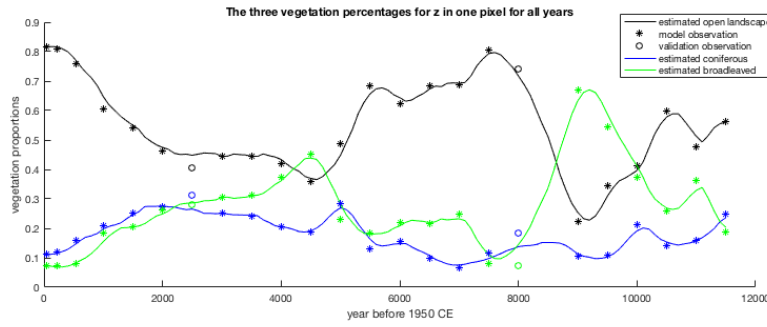


Figure 13: The reconstruction, z , in a time series for one pixel for Model A.

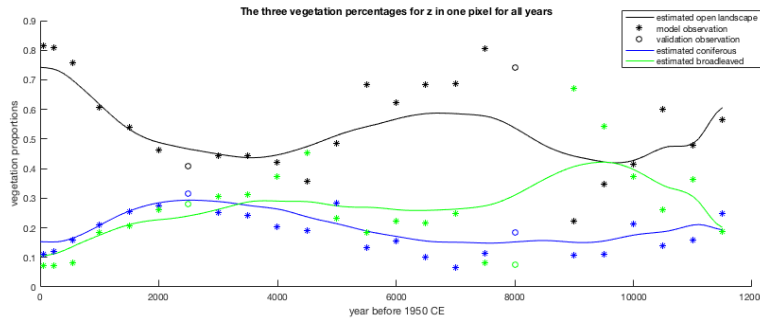


Figure 14: The reconstruction, z , in a time series for one pixel for Model B.

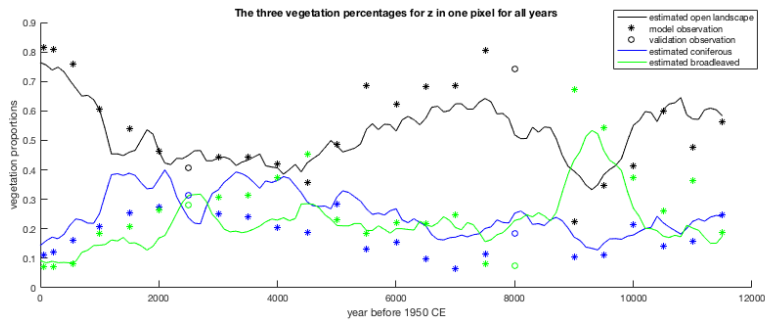


Figure 15: The reconstruction, z , in a time series for one pixel for Model C.

In figure 16, one can see the mean parameter β , in the reconstruction z , for Model A. In figure 17, the variance for the x -fields in one time window is visible. Figure 17 shows lower variance in the observed pixels and higher variance in the pixels at the edges, which is reasonable.

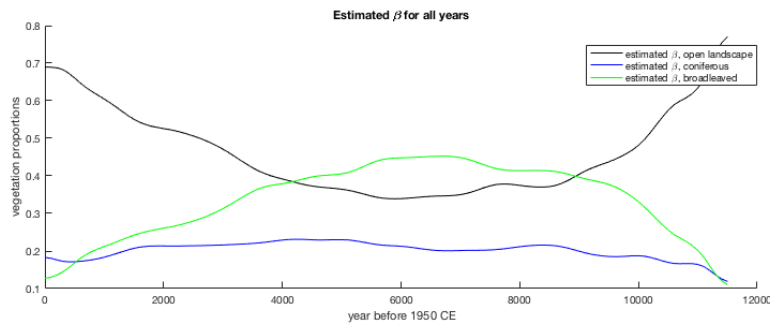


Figure 16: The mean, β , in the reconstruction z for Model A.

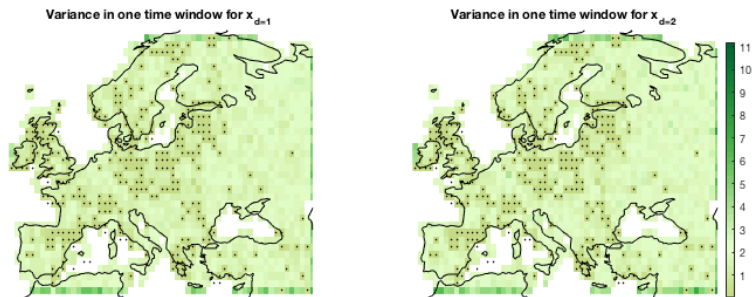


Figure 17: Variance for the x -fields in one time window for Model A.

5.3 Validation

In table 2, validation for the three models are shown. Model A has the best total ACD for the validation data, 0.71. The ACD = 1.04 for the validation data in the 19 time series, is much higher than the ACD = 0.51 for the 2 time windows. This can be explained by the model finding it easier to adjust to the temporal structure. Hence all validation data having neighbours in the time domain will be better estimated. The low ACD for the model data for Model A, confirms that the model has very little uncertainty in the observations, hence coming close to the observation values, but still performs well out of sample, showing good ability to generalise.

Table 2: ACD of the reconstructions for the three models.

	Model A	Model B	Model C
ACD for validation data in the 19 time series	1.04	1.05	1.01
ACD for validation data in the 2 time windows	0.51	0.79	1.20
ACD for all validation data	0.71	0.89	1.13
ACD for the model data	0.16	0.73	0.77

Figure 18 and 19 shows the reconstruction z , for one time window and one time series with validation data for Model A.

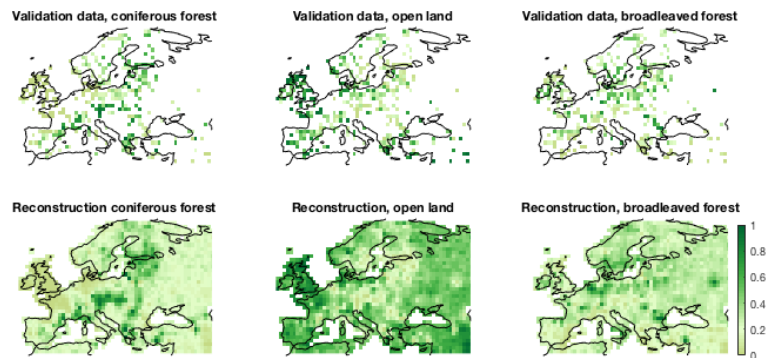


Figure 18: The reconstruction, z , in one time window for Model A, together with validation data.

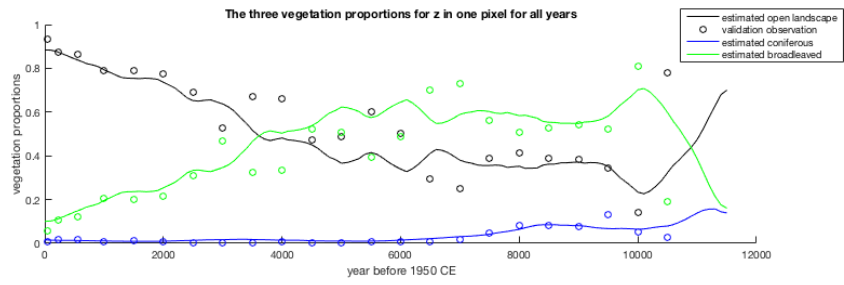


Figure 19: The reconstruction, z , in a time series for one pixel for Model A together with validation data.

6 Discussion and conclusions

6.1 Crank Nicolson Langevin method

The Crank Nicolson Langevin method works well for big data problems, such as the spatio-temporal data set in this project. The preconditioned Crank Nicolson Langevin proposal makes it possible to work with covariance matrices on Kronecker form which simplifies many matrix computations. The \mathbf{x} -fields show a good mixing during the estimation and the fields adapt to what ever parameters the model proposes, as seen in the different reconstructions made by model A, B, and C.

One disadvantage with the Crank Nicolson Langevin method is that it is somehow complicated to derive and implement. There is a lot of details in the derivations of the acceptance rate and the smallest mistake will be devastating when implementing the model. For example, an error in signs for the acceptance rate did not have any particular effect for only temporal modelling but when expanding the model to spatial domain or spatio-temporal domain, this error caused big issues and was hard to recognise.

6.2 Convergence of parameters κ_s , κ_t and α

The three parameters κ_s , κ_t and α , show poor mixing and converge very slowly or not at all. Some actions were taken to improve the convergence. The variance separation, $1/\kappa^{2\nu}$, for \mathbf{Q} and the more precise prior, λ , for the drawings of κ , are two examples. It did improve the convergence for κ_s and κ_t a little. The problem was also tried to be avoided by fixating those parameters with bad convergence, as was done in Model B and C. However, those models did not give any better results.

6.3 The average compositional distance

The reconstruction and the validation data had an average compositional distance of 0.71. When splitting the validation data into the two groups, time windows and time series, the two ACD:s became 0.51 and 1.04 respectively. This is a noticeable difference. The model tends to choose either strong dependence in the spatial or the temporal domain, when it estimates parameters. Here it chooses the temporal domain primarily for parameter estimation. Since there is a clear structure in the data, with the data coming in time windows, one might suspect that this could influence the estimations. We tried to loosen up the structure by removing 50 % of the observations at random. This gave no particular effect on the result.

6.4 Further extensions of the model

One challenge for future work is of course to find better methods of estimating the parameters κ_s , κ_t and α . The parameter α which became very large, could e.g. imply that the assumption of Dirichlet observations is bad. Except from improving convergence of κ_s , κ_t and α , there are other things one also might think of as possible extensions for this model. That could be adding some covariates, e.g. elevation, to the model. The model could also be extended to have more vegetation categories. Then one might want to consider another link function than the additive log-ratio, which does not allow for zero-values in any of the vegetation groups. Zero-values would also be a problem for the Dirichlet distribution.

References

- Beskos, A., Roberts, G., Stuart, A. & Voss, J. (2008), ‘MCMC methods for diffusion bridges’, *Stoch. Dyn.* **8**, 319–350.
- Billheimer, D., Guttorp, P. & Fagan, W. F. (2001), ‘Statistical interpretation of species composition’, *J. Amer. Statist. Assoc.* **96**, 1205–1214.
- Blangiardo, M. & Cameletti, M. (2015), *Spatial and spatio-temporal Bayesian models with R-INLA*, John Wiley and sons, Ltd, UK.
- Cotter, S. L., Roberts, G. O., Stuart, A. M. & White, D. (2013), ‘MCMCMethods for Functions: Modifying Old Algorithms to Make Them Faster’, *Stat. Sci.* **28**, 424–446.
- Crank, J. & Nicolson, P. (1947), ‘A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type’, *Mathematical Proceedings of the Cambridge Philosophical Society* **43**(1), 50–67.
- Fernandes, P. & Plateau, B. (1998), ‘Efficient descriptor-vector multiplications in stochastic automata networks’, *JACM* **45**, 381–414.
- Fuglstad, G.-A., Simpson, D., Lindgren, F. & Rue, H. (2018), ‘Constructing priors that penalize the complexity of Gaussian random fields’, **0**(0), 1–8.
- Gelman, A., Roberts, G. & Gilks, W. (1996), Efficient metropolis jumping rules, in J. M. Bernardo, J. Berger, A. P. Dawid & A. M. Smith, eds, ‘Bayesian Statistics 5’, Oxford University Press, pp. 599–607.
- Geman, S. & Geman, D. (1984), ‘Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images’, **6**, 721–741.
- Hastings, W. (1970), ‘Monte Carlo sampling methods using Markov chains and their applications’, **57**, 97–109.
- Lindgren, F., Rue, H. & Lindström, J. (2011), ‘An explicit link between Gaussian fields and Gaussian Markov random fields: The SPDE approach’, *J. R. Stat. Soc. Ser.B Stat. Methodol.* **73**, 423–498.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E. (1953), ‘Equations of state calculations by fast computing machines’, **21**, 1087–1092.
- Paciorek, C. J. & McLachlan, J. S. (2009), ‘Mapping ancient forests: bayesian inference for spatio-temporal trends in forest composition using the fossil pollen proxy record’, *J. Amer. Statist. Assoc.* **104**, 608–622.
- Pirzamanbein, B. et al. (2018), ‘Modelling Spatial Compositional Data: Reconstructions of past land cover and uncertainties’, *Spatial Statistics* **24**, 14–31.
- Roberts, G., Gelman, A. & Gilks, W. (1997), ‘Weak convergence and optimal scaling of random walk Metropolis algorithms’, *Ann. Probab.* **7**, 110–120.

- Roberts, G. O. & Stramer, O. (2002), ‘Langevin Diffusions and Metropolis-Hastings Algorithms’, *Appl. Probab.* **4**, 337–357.
- Roberts, G. & Rosenthal, J. S. (1998), ‘Optimal Scaling of Discrete Approximations to Langevin Diffusions’, *J. R. Stat. Soc. B* **760**, 255–268.
- Rosenthal, J. S. (2010), Optimal proposal distributions and adaptive mcmc, *in* S. Brooks, A. Gelman, G. L. Jones & X. Meng, eds, ‘Handbook of Markov Chain Monte Carlo’, Chapman Hall/CRC, pp. 93–111.
- Rue, H. & Held, L. (2005), *Gaussian Markov Random Fields; Theory and Applications*, Vol. 104 of *Monographs on Statistics and Applied Probability*, Chapman & Hall/CRC.
- Strandberg, G., Kjellström, A., Poska, E., Wagner, S., Gaillard, M. J., Trondman, A. K., Mauri, A., Davis, B. A. S., Kaplan, J. O., Birks, H. J. B., Bjune, A. E., Fyfe, R., Giesecke, T., Kalnina, L., Kangur, M., van der Knaap, W. O., Kokfelt, U., Kunes, P., Latałowa, M., Marquer, L., Mazier, F., Nielsen, A. B., Smith, B., Seppä, H. & Sugita, S. (2014), ‘Regional climate model simulations for europe at 6 and 0.2 k bp: sensitivity to changes in anthropogenic deforestation’, *Clim. Past.* **10**, 661–860.
- Tjelmeland, H. & Lund, K. V. (2003), ‘Bayesian modelling of spatial compositional data’, *J. Appl. Stat.* **30**, 87–100.
- Trondman, A. K. et al. (2015), ‘Pollen-based quantitative reconstructions of holocene regional vegetation cover (plant-functional types and land-cover types) in europe suitable for climate modelling’, *Global Change Biology* **21**, 676–697.

A Acceptance rate for MALA

With the notations $\mathcal{K} = \mathcal{I}^{-1} = \mathcal{I}^{-1/2}\mathcal{I}^{-1/2}$ for the preconditioner, we rewrite the MALA proposal in (26) as

$$\hat{\mathbf{x}} = \mathbf{x} + \delta \mathcal{I}(\mathbf{x})^{-1/2} \left(\frac{\delta}{2} \mathcal{I}(\mathbf{x})^{-1/2} \nabla \log \pi(\mathbf{x}) + \boldsymbol{\xi} \right). \quad (39)$$

For the acceptance rate $\alpha_{acc} = \frac{\pi(\hat{\mathbf{x}})q(\mathbf{x}|\hat{\mathbf{x}})}{\pi(\mathbf{x})q(\hat{\mathbf{x}}|\mathbf{x})}$ we identify

$$\frac{q(\mathbf{x}|\hat{\mathbf{x}})}{q(\hat{\mathbf{x}}|\mathbf{x})} = \frac{|\mathcal{I}(\hat{\mathbf{x}})|^{1/2} \exp\left(-\frac{1}{2\delta^2}[\mathbf{x} - \boldsymbol{\mu}(\hat{\mathbf{x}})]^T \mathcal{I}(\hat{\mathbf{x}})[\mathbf{x} - \hat{\boldsymbol{\mu}}(\mathbf{x})]\right)}{|\mathcal{I}(\mathbf{x})|^{1/2} \exp\left(-\frac{1}{2\delta^2}[\hat{\mathbf{x}} - \boldsymbol{\mu}(\mathbf{x})]^T \mathcal{I}(\mathbf{x})[\hat{\mathbf{x}} - \boldsymbol{\mu}(\mathbf{x})]\right)} \quad (40)$$

where $\boldsymbol{\mu}(\hat{\mathbf{x}}) = \mathbf{x} + \frac{\delta^2}{2} \mathcal{I}(\mathbf{x})^{-1} \nabla \log \pi(\mathbf{x})$. Hence computations of the following quantities are needed

$$\mathcal{I}(\mathbf{x}), \quad |\mathcal{I}(\mathbf{x})|^{1/2}, \quad \mathcal{I}(\mathbf{x})^{-1/2}, \quad \mathcal{I}(\hat{\mathbf{x}}), \quad |\mathcal{I}(\hat{\mathbf{x}})|^{1/2}. \quad (41)$$

As mentioned before, we use the expected Fisher information as a preconditioner. With $\pi(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{x})p(\mathbf{y}|\mathbf{x})$, the Fisher information will be on the form $\mathcal{I} = -\Delta \log p(\mathbf{x})p(\mathbf{y}|\mathbf{x}) = \mathbf{Q} + \Delta \log p(\mathbf{y}|\mathbf{x})$. This will be computational heavy when \mathbf{x} , hence \mathcal{I} is big, since all matrices and determinants in (41) must be computed. The computations becomes even more heavy when \mathcal{I} does not have a Kronecker structure since the Kronecker form of \mathbf{Q} ceases with the addition $\mathcal{I}(\mathbf{x}) = \mathbf{Q} + \Delta \log p(\mathbf{y}|\mathbf{x})$.

B More about Crank Nicolson Langevin

B.1 Why pCNL instead of CNL

For CNL with preconditioner $\mathcal{K} = \mathbf{I}$, we get from (32) the CNL proposal

$$\left(\mathbf{I} + \frac{h}{2}\mathbf{Q}\right)\hat{\mathbf{x}} = \left(\mathbf{I} - \frac{h}{2}\mathbf{Q}\right)\mathbf{x} - h\nabla\Phi(\mathbf{x}) + \sqrt{2h}\boldsymbol{\xi}. \quad (42)$$

where $\boldsymbol{\xi} \sim \mathcal{N}(0,1)$. This proposal implies that we have to solve the equation system $\tilde{\mathbf{Q}}^{-1}\mathbf{x}$, where $\tilde{\mathbf{Q}} = \left(\mathbf{I} + \frac{h}{2}\mathbf{Q}\right)$, does not have the Kronecker structure. Hence we cannot take advantage of the solutions for the matrix equations with Kronecker product in (17) and (18). With pCNL we will see how the Kronecker formation in \mathbf{Q} does not get disrupted by any additions.

B.2 Calculations for pCNL proposal

With $\mathcal{K} = \mathbf{Q}^{-1}$, equation (32) is written as

$$\frac{2+h}{2}\hat{\mathbf{x}} = \frac{2-h}{2}\mathbf{x} - h\mathbf{Q}^{-1}\nabla\Phi(\mathbf{x}) + \sqrt{2h\mathbf{Q}^{-1}}\boldsymbol{\xi},$$

which then can be rewritten as

$$\hat{\mathbf{x}} = \frac{2-h}{2+h}\mathbf{x} - \frac{2h}{2+h}\mathbf{Q}^{-1}\nabla\Phi(\mathbf{x}) + \frac{2\sqrt{2h\mathbf{Q}^{-1}}}{2+h}\xi. \quad (43)$$

For further calculations we find help in the following shorter notations

$$\omega = \frac{2\sqrt{2h}}{2+h}, \quad \omega^2 = \frac{8h}{(2+h)^2}, \quad \zeta = \sqrt{1-\omega^2} = \frac{2-h}{2+h}.$$

With the shorter notations above, equation (43) is

$$\hat{\mathbf{x}} = \zeta\mathbf{x} - (1-\zeta)\mathbf{Q}^{-1}\nabla\Phi(\mathbf{x}) + \omega\sqrt{\mathbf{Q}^{-1}}\xi,$$

alternatively, using the Choleskey factorisation $\mathbf{Q}^{-1} = \mathbf{R}^{-1}\mathbf{R}^{-T}$,

$$\hat{\mathbf{x}} = \zeta\mathbf{x} + \mathbf{R}^{-1}\left(- (1-\zeta)\mathbf{R}^{-T}\nabla\Phi(\mathbf{x}) + \omega\xi\right). \quad (44)$$

The proposal for pCNL becomes

$$\hat{\mathbf{x}}|\mathbf{x} \sim \mathcal{N}(\zeta\mathbf{x} - (1-\zeta)\mathbf{Q}^{-1}\nabla\Phi(\mathbf{x}), \omega^2\mathbf{Q}^{-1}), \quad (45)$$

where all computations can utilise the Kronecker structure of \mathbf{Q} .

B.3 Acceptance rate for pCNL

To be able to calculate the acceptance rate $\alpha_{acc} = \frac{\pi(\hat{\mathbf{x}})q(\mathbf{x}|\hat{\mathbf{x}})}{\pi(\mathbf{x})q(\hat{\mathbf{x}}|\mathbf{x})}$. Due to symmetry in the transition density we have

$$\mathbf{x}|\hat{\mathbf{x}} \sim \mathcal{N}(\zeta\hat{\mathbf{x}} - (1-\zeta)\mathbf{Q}^{-1}\nabla\Phi(\hat{\mathbf{x}}), \omega^2\mathbf{Q}^{-1}). \quad (46)$$

The form of a multivariate normal distribution is

$$f_X(\mathbf{x}) \propto \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right),$$

where μ is the mean and Σ the covariance matrix. With μ and Σ as in (46) we get

$$\Sigma^{-1} = \frac{1}{\omega^2}\mathbf{Q}$$

and

$$(\mathbf{x} - \mu) = \mathbf{x} - \zeta\hat{\mathbf{x}} + (1-\zeta)\mathbf{Q}^{-1}\nabla\Phi(\hat{\mathbf{x}})$$

The transition density thus becomes

$$q(\mathbf{x}|\hat{\mathbf{x}}) \propto \exp\left(-\frac{1}{2\omega^2}[\mathbf{x} - \zeta\hat{\mathbf{x}} + (1-\zeta)\mathbf{Q}^{-1}\nabla\Phi(\hat{\mathbf{x}})]^T \mathbf{Q}[(\mathbf{x} - \zeta\hat{\mathbf{x}} + (1-\zeta)\mathbf{Q}^{-1}\nabla\Phi(\hat{\mathbf{x}}))]\right),$$

We will continue working with the log scaled density, $\log q(\mathbf{x}|\hat{\mathbf{x}})$, since this is more convenient. Before we continue, we remind ourselves that \mathbf{Q} is a symmetric matrix, thus $\mathbf{Q} = \mathbf{Q}^T$, and $\hat{\mathbf{x}}^T \mathbf{Q} \mathbf{x}$ is a scalar, thus $\hat{\mathbf{x}}^T \mathbf{Q} \mathbf{x} = (\hat{\mathbf{x}}^T \mathbf{Q} \mathbf{x})^T = \mathbf{x}^T \mathbf{Q} \hat{\mathbf{x}}$. Continuing the derivation of $\log q(\mathbf{x}|\hat{\mathbf{x}})$ we get

$$\begin{aligned} \log q(\mathbf{x}|\hat{\mathbf{x}}) \propto & -\frac{1}{2\omega^2} \left((\mathbf{x} - \zeta \hat{\mathbf{x}})^T \mathbf{Q} (\mathbf{x} - \zeta \hat{\mathbf{x}}) \right. \\ & + 2(1 - \zeta) (\mathbf{x} - \zeta \hat{\mathbf{x}})^T \mathbf{Q} (\mathbf{Q}^{-1} \nabla \Phi(\hat{\mathbf{x}})) \\ & \left. + (1 - \zeta)^2 (\mathbf{Q}^{-1} \nabla \Phi(\hat{\mathbf{x}}))^T \mathbf{Q} (\mathbf{Q}^{-1} \nabla \Phi(\hat{\mathbf{x}})) \right). \end{aligned}$$

Developing the matrix multiplication in the first line and cancelling $\mathbf{Q}\mathbf{Q}^{-1}$ in the second and third lines give

$$\begin{aligned} \log q(\mathbf{x}|\hat{\mathbf{x}}) \propto & -\frac{1}{2\omega^2} \left(\mathbf{x}^T \mathbf{Q} \mathbf{x} - 2\zeta \mathbf{x}^T \mathbf{Q} \hat{\mathbf{x}} + \zeta^2 \hat{\mathbf{x}}^T \mathbf{Q} \hat{\mathbf{x}} \right. \\ & + 2(1 - \zeta) (\mathbf{x} - \zeta \hat{\mathbf{x}})^T \nabla \Phi(\hat{\mathbf{x}}) \\ & \left. + (1 - \zeta)^2 \nabla \Phi(\hat{\mathbf{x}})^T \mathbf{Q}^{-1} \nabla \Phi(\hat{\mathbf{x}}) \right). \end{aligned}$$

From symmetry in the proposals we have

$$\begin{aligned} \log q(\hat{\mathbf{x}}|\mathbf{x}) \propto & -\frac{1}{2\omega^2} \left(\hat{\mathbf{x}}^T \mathbf{Q} \hat{\mathbf{x}} - 2\zeta \mathbf{x}^T \mathbf{Q} \hat{\mathbf{x}} + \zeta^2 \mathbf{x}^T \mathbf{Q} \mathbf{x} \right. \\ & + 2(1 - \zeta) (\hat{\mathbf{x}} - \zeta \mathbf{x})^T \nabla \Phi(\mathbf{x}) \\ & \left. + (1 - \zeta)^2 \nabla \Phi(\mathbf{x})^T \mathbf{Q}^{-1} \nabla \Phi(\mathbf{x}) \right). \end{aligned}$$

Subtracting the two the log densities of the proposals gives

$$\begin{aligned} \log q(\mathbf{x}|\hat{\mathbf{x}}) - \log q(\hat{\mathbf{x}}|\mathbf{x}) = & -\frac{1}{2\omega^2} \left((1 - \zeta^2) \mathbf{x}^T \mathbf{Q} \mathbf{x} + (\zeta^2 - 1) \hat{\mathbf{x}}^T \mathbf{Q} \hat{\mathbf{x}} \right. \\ & + 2(1 - \zeta) [(\mathbf{x} - \zeta \hat{\mathbf{x}})^T \nabla \Phi(\hat{\mathbf{x}}) - (\hat{\mathbf{x}} - \zeta \mathbf{x})^T \nabla \Phi(\mathbf{x})] \\ & \left. + (1 - \zeta)^2 [\nabla \Phi(\hat{\mathbf{x}})^T \mathbf{Q}^{-1} \nabla \Phi(\hat{\mathbf{x}}) - \nabla \Phi(\mathbf{x})^T \mathbf{Q}^{-1} \nabla \Phi(\mathbf{x})] \right). \end{aligned}$$

Further simplifications can be done for the first line with $\frac{1-\zeta^2}{\omega^2} = 1$. For second line we note that

$$\frac{1 - \zeta}{\omega^2} = \frac{1}{2} + \frac{h}{4}, \quad \frac{\zeta(1 - \zeta)}{\omega^2} = \frac{1}{2} - \frac{h}{4}, \quad (47)$$

which we use to simplify it down (some computations steps are shortened out from the text). In the third line we use $\frac{(1-\zeta)^2}{\omega^2} = \frac{h}{2}$, and with $\mathbf{Q}^{-1} = \mathbf{R}^{-1} \mathbf{R}^{-T}$ we can rewrite several expressions using euclidean norms. With all this we get

$$\begin{aligned} \log q(\mathbf{x}|\hat{\mathbf{x}}) - \log q(\hat{\mathbf{x}}|\mathbf{x}) = & -\frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \frac{1}{2} \hat{\mathbf{x}}^T \mathbf{Q} \hat{\mathbf{x}} \\ & + \frac{1}{2} (\mathbf{x} - \hat{\mathbf{x}})^T (\nabla \Phi(\mathbf{x}) + \nabla \Phi(\hat{\mathbf{x}})) + \frac{h}{4} (\hat{\mathbf{x}} + \mathbf{x})^T (\nabla \Phi(\mathbf{x}) - \nabla \Phi(\hat{\mathbf{x}})) \\ & - \frac{h}{4} \|\mathbf{R}^{-1} \nabla \Phi(\hat{\mathbf{x}})\|_2^2 + \frac{h}{4} \|\mathbf{R}^{-1} \nabla \Phi(\mathbf{x})\|_2^2. \end{aligned}$$

We notice that the first line corresponds to $\log p(\mathbf{x}) - \log p(\hat{\mathbf{x}})$, which cancel against the $\log p(\hat{\mathbf{x}})/\log p(\mathbf{x})$ term in the acceptance rate, giving

$$\log \frac{p(\hat{\mathbf{x}})q(\mathbf{x}|\hat{\mathbf{x}})}{p(\mathbf{x})q(\hat{\mathbf{x}}|\mathbf{x})} = \frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}})^T(\nabla\Phi(\mathbf{x}) + \nabla\Phi(\hat{\mathbf{x}})) + \frac{h}{4}(\hat{\mathbf{x}} + \mathbf{x})^T(\nabla\Phi(\mathbf{x}) - \nabla\Phi(\hat{\mathbf{x}})) \quad (48)$$

$$- \frac{h}{4}\|\mathbf{Q}^{-1/2}\nabla\Phi(\hat{\mathbf{x}})\|_2^2 + \frac{h}{4}\|\mathbf{Q}^{-1/2}\nabla\Phi(\mathbf{x})\|_2^2. \quad (49)$$

The complete logged acceptance probability, with target density $\pi(\mathbf{x}) = p(\mathbf{x}|\mathbf{y})$, is given by

$$\log \alpha_{acc} = \log \frac{p(\hat{\mathbf{x}}|\mathbf{y})q(\mathbf{x}|\hat{\mathbf{x}})}{p(\mathbf{x}|\mathbf{y})q(\hat{\mathbf{x}}|\mathbf{x})} = \log \frac{p(\mathbf{y}|\hat{\mathbf{x}})p(\hat{\mathbf{x}})q(\mathbf{x}|\hat{\mathbf{x}})}{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})q(\hat{\mathbf{x}}|\mathbf{x})} = \log \frac{p(\mathbf{y}|\hat{\mathbf{x}})}{p(\mathbf{y}|\mathbf{x})} + \log \frac{p(\hat{\mathbf{x}})q(\mathbf{x}|\hat{\mathbf{x}})}{p(\mathbf{x})q(\hat{\mathbf{x}}|\mathbf{x})},$$

where the first term is

$$\log \frac{p(\mathbf{y}|\hat{\mathbf{x}})}{p(\mathbf{y}|\mathbf{x})} = -\Phi(\hat{\mathbf{x}}) + \Phi(\mathbf{x}),$$

and the second term is given in the two lines (48) and (49).

B.4 Gradient of $\Phi(\mathbf{x})$

To derive $\nabla\Phi(\mathbf{x})$, we start with clarifying how the log-density of a Dirichlet distribution looks

$$\begin{aligned} -\Phi(\mathbf{x}) &= \log p(\mathbf{y}|\alpha, \mathbf{z}(\mathbf{x})) = \log \prod_{s=1}^{N_{obs}} \left(\frac{\Gamma(\alpha)}{\prod_{k=1}^D \Gamma(\alpha z_{s,k})} \prod_{k=1}^D y_{s,k}^{\alpha z_{s,k} - 1} \right) \\ &= \sum_{s=1}^{N_{obs}} \log \Gamma(\alpha) - \sum_{s=1}^{N_{obs}} \sum_{k=1}^D \log \Gamma(\alpha z_{s,k}) + \sum_{k=1}^D (\alpha z_{s,k} - 1) \log y_{s,k}. \end{aligned}$$

The elements of the gradient with respect to $\boldsymbol{\eta}$ (Pirzamanbein et al. 2018) are

$$\frac{\partial \log p(\mathbf{y}|f(\boldsymbol{\eta}), \alpha)}{\partial \eta_{s,k}} = \sum_{l=1}^D (-\alpha \psi(\alpha z_{s,l}) + \alpha \log y_{s,l}) \frac{\partial z_{s,l}}{\partial \eta_{s,k}} \quad (50)$$

where $\psi(z)$ is the digamma function $\psi(z) = \frac{d}{dz} \log \Gamma(z)$ and the derivative of the ALR-transform is

$$\frac{\partial z_k}{\partial \eta_i} = \begin{cases} z_k(1 - z_k) & \text{if } k = i, \\ -z_k z_i & k \neq i \end{cases}. \quad (51)$$

C The joint posterior for κ, ρ_x

For the posterior $p(\rho_x | \mathbf{x}, \kappa)$, we have the conjugate prior for ρ_x as given in (9). The posterior hence becomes

$$\rho_x | \kappa, \mathbf{x} \propto IW(a_\rho I + \mathbf{x}^T \mathbf{Q}_{s,t} \mathbf{x}, N + b_\rho), \quad (52)$$

where \mathbf{x} is reshaped on the form $(N \times d)$ and $\mathbf{Q}_{s,t} = Q_t \otimes Q_s$ (Pirzamanbein et al. 2018). The conjugacy makes it possible to marginalize over the covariance matrix ρ_x and integrate it out from the joint posterior $p(\kappa, \rho_x | \mathbf{x})$ (Pirzamanbein et al. 2018). In this manner the posterior for $\kappa | \mathbf{x}$ becomes

$$p(\kappa | \mathbf{x}) \propto \int (\mathbf{x} | \kappa, \rho_x) \cdot p(\kappa) p(\rho_x) d\rho_x \propto \frac{a_\rho^{\frac{d-b_\rho}{2}} |\mathbf{Q}_{s,t}|^{\frac{d}{2}}}{|a_\rho I + \mathbf{x}^T \mathbf{Q}_{s,t} \mathbf{x}|^{\frac{N+b_\rho}{2}}} \cdot p(\kappa_s), \quad (53)$$

The determinant $|\mathbf{Q}_{s,t}|$, is according to (16), given by

$$\det(Q_t \otimes Q_s) = \det(Q_t)^{n_s} \det(Q_s)^{n_t},$$

where n_s and n_t are the length of the square matrices Q_s and Q_t . The acceptance rate for the joint transition density $q(\hat{\kappa}, \hat{\rho} | \kappa, \rho) = p(\hat{\rho} | \mathbf{x}, \hat{\kappa}) \cdot q(\hat{\kappa} | \kappa)$ is given as

$$\begin{aligned} \alpha_{acc} &= \min \left(1, \frac{p(\hat{\rho} | \mathbf{x}, \hat{\kappa}) \cdot p(\hat{\kappa} | \mathbf{x})}{p(\rho | \mathbf{x}, \kappa) \cdot p(\kappa | \mathbf{x})} \cdot \frac{p(\rho | \mathbf{x}, \kappa) \cdot q(\kappa | \hat{\kappa})}{p(\hat{\rho} | \mathbf{x}, \hat{\kappa}) \cdot q(\hat{\kappa} | \kappa)} \right) \\ &= \min \left(1, \frac{p(\hat{\kappa} | \mathbf{x})}{p(\kappa | \mathbf{x})} \cdot \frac{\hat{\kappa}}{\kappa} \right), \end{aligned}$$

where $p(\hat{\kappa} | \mathbf{x})$ is given in (53).

C.1 Posterior for ρ_β

The posterior for ρ_β will be similar to the posterior ρ_x given in (52). The prior for β in (19) gives the posterior

$$\rho_\beta | \beta \propto IW(a_\rho I + \beta^T G_t^T G_t \beta, N + b_\rho), \quad (54)$$

where β is reshaped on the form $(n_t \times d)$.

D Creating the observation matrix \mathbf{A}

The observation matrix \mathbf{A} is created based on the indexing of the model data. Unlike G_s and G_t where spatial and temporal indexing are separated, the observation matrix \mathbf{A} store all indexing together. The original indices for the pollen

data concerning longitude, latitude and time window are stored in vectors we call \mathbf{i}_{lat} , \mathbf{i}_{lon} and \mathbf{i}_{year} . It is more convenient to use indexing that starts at 1. For spatial indices we will use

$$\begin{aligned}\mathbf{i}_{row} &= \mathbf{i}_{lat} - \min_{lat} + 1 & \text{and} \\ \mathbf{i}_{col} &= \mathbf{i}_{lon} - \min_{lon} + 1.\end{aligned}$$

For the indexing in time we also adjust to the new step length of 100 years. It gives the following new indexing for the time windows

$$\mathbf{i}_{time} = \min(\mathbf{floor}((\mathbf{i}_{year} - \min_{time})/100) + 1, n_t - 1) \quad (55)$$

where `floor.m` is a Matlab function that rounds the elements to the nearest integers towards minus infinity. The indexing for the $N_{obs} \times N$ observation matrix, A , was then created as

$$\mathbf{i}_A = \mathbf{i}_{row} + (\mathbf{i}_{col} - 1) * n_{s1} + \mathbf{i}_{time} * n_s$$

Then a $N_{obs} \times N$ sparse matrix A can be created with the pre-written Matlab function `sparse.m`, which takes the vector \mathbf{i}_A and the sizes N_{obs} and N as input parameters. The extended observation matrix \mathbf{A} for d number of fields is then created as the Kronecker product

$$\mathbf{A} = I_{d \times d} \otimes A$$

Another observation matrix A_{val} was created and used for plotting of the validation data.

Master's Theses in Mathematical Sciences 2019:E56
ISSN 1404-6342
LUTFMS-3382-2019
Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden
<http://www.maths.lth.se/>