Bayesian and frequentist computation with application to data from the Medical Birth Registry of Norway

Janne Mannseth

Thesis for the degree of Philosophiae Doctor (PhD)
University of Bergen, Norway
2019

UNIVERSITY OF BERGEN

Bayesian and frequentist computation with application to data from the Medical Birth Registry of Norway

Janne Mannseth

Thesis for the degree of Philosophiae Doctor (PhD)
at the University of Bergen

Date of defense: 25.04.2019

Year: 2019

Title: Bayesian and frequentist computation with application to data from the Medical Birth Registry of Norway

Name: Janne Mannseth

Print: Skipnes Kommunikasjon / University of Bergen

Preface

The work of this thesis has been carried out during a 4-year position as a PhD-student in the Department of Mathematics at the University of Bergen, from August 2014 until December 2018. In total, three of these years have been dedicated to my research and to my education in the form of courses, seminars and conferences. One year has consisted of various assignments such as teaching, organizing seminars for students and grading exam papers. For one year I had a 20% leave because I was working part time as a statistician for a national health registry. The fall of 2017 was spent at the Center for Clinical Research at Haukeland University Hospital in Bergen.

Outline

This thesis considers numerical methods that are applied as solutions in Bayesian and frequentist computing. A well-known problem that occurs in Bayesian inference is finding the posterior distribution. A very common approach to this problem is to use Markov chain Monte Carlo (MCMC) methods to generate samples that stem from the posterior distribution. There exist several branches that originate from the MCMC methods. One of these is Hamiltonian Monte Carlo (HMC). Here, HMC is considered with the purpose of improving its performance and efficiency by altering the numerical integration schemes within the algorithm.

Although MCMC methods are popular and effective, they also have weaknesses. For certain latent variable models, MCMC simulations are often computationally inefficient and time consuming. This is especially a problem for high-dimensional cases. For these situations, we consider different numerical techniques. First, we look at a particular property that is important in integrated nested Laplace approximations (INLA). This property will be used in later computations. INLA is significantly faster than MCMC for latent variable models, though we have used a different methodology. We use an automatic differentiation (AD) model builder, `TMB`. With `TMB`, we can combine AD and Laplace approximation in the process of finding the maximum likelihood estimates of the parameters. Finally, we use the stochastic partial differential equations (SPDE) approach to supply a link between Gaussian fields (GFs) and Gaussian Markov random fields (GMRFs). GFs are very common with latent variable and spatial models and GMRFs come with big computational advantages. We use the SPDE approach for spatial and spatio-temporal models. The spatial data come from the Medical Birth registry of Norway.

List of papers

Paper 1 Janne Mannseth, Tore S. Kleppe and Hans J. Skaug (2017). "On the application of improved symplectic integrators in Hamiltonian Monte Carlo." *Communications in Statistics - Simulation and Computation*.

Paper 2 Janne Mannseth, Geir D. Berentsen and Hans J. Skaug. “Robustness of the SPDE approximation under short range spatial correlation.”

Paper 3 Janne Mannseth, Geir D. Berentsen, Hans J. Skaug and Dag Moster. “Joint modeling of Caesarean section and severe maternal hemorrhage using spatio-temporal Gaussian random fields.”

Acknowledgements

First, I want to thank my main supervisor Hans Julius Skaug for introducing me to the very interesting problems that I have worked with in my thesis and for sharing his knowledge about them. I have learned very much from it. You have prioritized my interests and given me valuable opportunities such as getting to spend a semester with the statisticians at the university hospital and the possibility to cooperate with the Department of Global Public Health and Primary Care here in Bergen. This has opened many doors for me, and I am very thankful for that.

I also want to thank my co-supervisor Tore Selland Kleppe. You are the one who introduced me to scientific work when you supervised me for my master's thesis, and that played a big part in my choice of continuing down this path. In addition, you always give precise and quick feedback even though you are in another city. This has been so helpful and I have truly appreciated it. I would also like to thank my second co-supervisor Geir Drage Berentsen. You have taught me much about how to approach a variety of statistical problems, and I have also become better at programming after cooperating with you. I want to thank you for your optimism towards me and for always taking the time to discuss questions that I have had.

I am grateful for being allowed to stay one semester with the statisticians at the Center for Clinical Research. This was such a good experience for me that I enjoyed and learned a lot from.

I want to thank my parents, who support me and who I know that I can always count on. And finally, I would like to thank Simen, who is there for me every single day. You have helped me more than you know and have also given me hope and motivation when I have needed it the most.

Janne Mannseth

Abstract

In chapter 1, the field of statistics is discussed in general terms. Then, Bayes' theorem is presented together with the posterior distribution. Finally, we consider maximum likelihood estimation and its relation to a two-step inference procedure.

The thesis proceeds to introducing several numerical methods in chapter 2. One of the main methods is Hamiltonian Monte Carlo (HMC). In order to understand HMC, which is a type of Markov chain Monte Carlo (MCMC) algorithm, we first study some theory of Markov chains. Furthermore, MCMC and some examples of its well-known algorithms are encountered. As HMC is based on Hamiltonian dynamics (that originate from the field of physics), a short illustration of the dynamics is given. Furthermore, the system of differential equations used in HMC is discussed, as are several different numerical integration schemes and methods for developing them. A numerical integration scheme is a necessary part of the HMC algorithm. Paper 1 covers HMC and finding more efficient numerical integration schemes to be used in the algorithm.

It is clear that there are situations where the MCMC algorithms perform poorly, especially with regards to time consumption. For certain latent variable models we should consider alternative numerical methods to MCMC. For these models, one could use integrated nested Laplace approximation (INLA). Although INLA has not been used in the model specification here, we are interested in some of the theory it applies. INLA is an efficient alternative for latent Gaussian models and uses some important properties of Gaussian fields (GFs) and Gaussian Markov random fields (GMRFs). From INLA, we here more specifically consider the relation between the computationally demanding covariance matrix and the sparse precision matrix. These two matrices are directly connected to GFs and GMRFs respectively, and we examine some the computational advantages that come with GMRFs compared to GFs. We use the template model builder (`TMB`) that uses automatic differentiation and Laplace approximation in the process of finding the maximum likelihood estimation of the parameters of the marginal likelihood. The combination in `TMB` gives the ability to handle complex latent variable models, and this is the methodology that is used for the spatial models in paper 2 and paper 3.

In chapter 3 we look at the field of spatial statistics. This includes its uses and the different types of spatial data that we have. Spatial data are separated by what kind of location information that is available and what type of problem we want to model. Moreover, we discuss the stochastic partial differential equations (SPDE) approach. This is an approach that is able to link GFs and GMRFs using a triangulated grid. This grid, a weak solution to the SPDE and its use with spatial models are examined. To understand the SPDE approach we also consider the Matérn covariance function, which is the covariance function for certain GFs. The SPDE approach is essential for the results of paper 2 and paper 3. The spatial and spatio-temporal models are also explained and some additional results to paper 3 are presented. Finally, we look at health registry data, which is the type of data used in paper 2 and paper 3.

Contents

Preface i

Acknowledgements iii

Abstract v

1 Introduction 1
- 1.1 Bayesian inference . . . 2
- 1.2 A two-step inference procedure . . . 3

2 Numerical integration methods 5
- 2.1 Markov chains . . . 5
- 2.2 Monte Carlo methods . . . 6
- 2.3 Markov chain Monte Carlo . . . 7
- 2.4 Hamiltonian Monte Carlo . . . 9
  - 2.4.1 Numerical integration schemes . . . 11
  - 2.4.2 The Hamiltonian Monte Carlo algorithm . . . 12
- 2.5 Integrated nested Laplace approximation . . . 14
- 2.6 The Template Model Builder . . . 16

3 Spatial modeling 19
- 3.1 Spatial statistics . . . 19
  - 3.1.1 Spatial stochastic processes . . . 20
- 3.2 The stochastic partial differential equations approach . . . 20
- 3.3 The spatial and spatio-temporal models . . . 22
- 3.4 Additional results . . . 24
- 3.5 Health registry data . . . 27

4 Summary of papers 29
- 4.1 Summary of Paper 1: "On the application of improved symplectic integrators in Hamiltonian Monte Carlo" . . . 29
- 4.2 Summary of Paper 2: "Robustness of the SPDE approximation under short range spatial correlation" . . . 30
- 4.3 Summary of Paper 3: “Joint modeling of Caesarean section and severe maternal hemorrhage using spatio-temporal Gaussian random fields” . . . 31

5 Errata 33

6 Papers 37
6.1 On the application of improved symplectic integrators in Hamiltonian Monte Carlo 39
6.2 Robustness of the SPDE approximation under short range spatial correlation 51
6.3 Joint modeling of Caesarean section and severe maternal hemorrhage using spatio-temporal Gaussian random fields 77

Chapter 1

Introduction

In statistics, mathematical knowledge and data analysis are combined with the aim of describing a population. Moreover, to be able to draw useful assumptions about it with a given certainty. The field of statistics has many applications. In actuarial science, a wide range of statistical methods are used to determine risk in the business of insurance and finance. Machine learning uses statistical methods to give computers the ability to generate predictions. Biostatistics consider medical and biological incidents and study them using statistical analysis.

The kind of subjects that are studied can range from a certain group of people to the weather. Regardless of subject, access to data about the phenomenon of interest is often sought when working in the field of statistics. Census data, which is data about the whole (relevant) population, is ideal but often problematic. It is therefore also common to use a subgroup of the population as a representation. This group can be used to describe behaviour and we want this sample to represent the entire population. Thus, statistical inference is applied in order to acquire more knowledge about the sample. We can perform hypothesis testing, parameter estimation and regression to try and describe the population using just this sample. For the data in this thesis, the focus is mainly on health registry data. This type of data often coincides with a large population. Therefore, in addition to the societal benefits, health registry data is highly relevant in research.

Variables that are the subject of statistical analysis can be random (stochastic). This means that they can take several values, based on the result of a random experiment. For instance, coin tossing gives the random variable two possible outcomes. In statistics, a probability distribution is a function that represents all the values that the random variable can take as well as the corresponding probabilities. In the simple coin tossing example, the probability distribution will have probability $\frac{1}{2}$ for both of the possible values the random variable can take.

In Bayesian statistics, the posterior distribution is the probability distribution of a random variable conditional on some information. The posterior distribution, which we will consider more closely in section 1.1, includes a normalizing factor. This factor requires finding an integral. Solving this integral analytically has proven to be very complex, and often impossible. The solution has instead been to use numerical methods that draw samples from the posterior distribution. Sometimes, we want to evaluate the probability of the data conditional on the model parameters. This is called the likelihood of a model. The values of the parameters that can best explain the data is found using maximum likelihood estimation (MLE), a popular method in frequentist statistics. In this thesis and in the three papers included here, we will see several different numerical techniques that are used in Bayesian and frequentist computing. In paper 1, the use of Markov chain Monte Carlo (MCMC) methods and in particular the Hamiltonian Monte Carlo (HMC) method is discussed. In general, MCMC methods are widely used within Bayesian statistics and MCMC simulation can be used to draw samples from the posterior distribution. However, MCMC can be very computationally demanding and time consuming. This is especially true for complex latent variable models. As an alternative, one could turn to integrated nested Laplace approximations (INLA), which is an efficient method for latent Gaussian models (LGMs). Here, we only use INLA to illustrate some computational advantages that are to be used later. Instead, we use the template model builder (`TMB`). `TMB` is based on a method that finds the parameter estimates using MLE in combination with automatic differentiation (AD) and Laplace approximation. For paper 2 and paper 3, this method is used together with the stochastic partial differential equations (SPDE) approach. This approach is implemented in the same `R` package as INLA. Combining the SPDE approach and methodologies for approximating the complex latent models gives interesting alternatives to the MCMC algorithms.

1.1 Bayesian inference

Bayesian inference can be applied in cases where we want to adjust the probability for an event based on getting more information. Consider first Bayes' theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}, \tag{1.1}$$

Bayes' theorem can be illustrated with the following example. Say we want to find the probability of a patient having lung cancer given that the patient smokes. In this situation, A is the event that the patient has lung cancer, and let us say that we know that this probability is $P(A) = 0.1$. Let B be that the patient smokes. Moreover, let the probability of smoking $P(B) = 0.03$. We have that $P(B \mid A)$ gives the probability that the patient smokes, given that it has lung cancer. Here, this probability is 0.07. Finally, $P(A \mid B)$ will then be equal to 0.23. This means that if the patient smokes, the probability of having lung cancer is 23%, which is more than double the probability of having lung cancer without knowing if the patient smokes. By adding information, we can thus update the probability of an event. There are numerous examples like this, where Bayes' theorem is applied with point probabilities. However, there are also often distributions that follow the events and this is a very common situation in Bayesian inference.

In frequentist statistics (sometimes referred to as classical statistics), the parameter θ is a fixed and unknown quantity. This is opposed to Bayesian statistics, where θ is a quantity whose value varies. Here, θ is treated as a random variable with variation characterized by a prior distribution $\pi(\theta)$. The prior distribution is a probability distribution that is formulated before we have any information from the data y. In other words, it contains information that is available prior to making use of the data. Let $\pi()$ denote a general probability distribution. A random sample $y = y_1, \ldots, y_n$ is drawn from a population indexed by θ. We let these observations have likelihood $\pi(y \mid \theta)$ and we denote the probability distribution for $\theta \mid y$ as the posterior distribution. The posterior distribution is proportional to the prior distribution times the likelihood, as seen in equation (1.2):

$$\pi(\theta \mid y) \propto \pi(y \mid \theta)\pi(\theta). \tag{1.2}$$

Equation (1.2) can be seen as a representation of Bayes' theorem because the posterior distribution can be fully formulated as

$$\pi(\theta \mid y) = \frac{\pi(y \mid \theta)\pi(\theta)}{\pi(y)}, \tag{1.3}$$

where $\pi(y)$ is the marginal distribution of y and expands to

$$\pi(y) = \int \pi(y \mid \theta)\pi(\theta)d\theta. \tag{1.4}$$

The posterior distribution (1.3) yields the possibility of finding out how likely something (here: θ) is, based on the actual data but also accounting for prior knowledge. This distribution is highly relevant in Bayesian inference. Its uses include estimation of parameters as well as prediction.

In general, solving the integral in equation (1.4) analytically can be very demanding, and often infeasible. The solution is to solve the problem numerically. In Bayesian inference, we therefore have a variety of numerical methods that are used to overcome this issue.

1.2 A two-step inference procedure

In frequentist statistics, we consider θ to be a number instead of a random variable. We know that the likelihood function is $\pi(y \mid \theta)$, and that it is the probability distribution for the data given the parameters of the model. We assume that the observations $y = y_1, \ldots, y_n$ are independent. Typically, we want to find the values of the parameters in the model that can best explain the observed data. When θ is considered a fixed quantity, MLE is a popular method for estimating its value. Define the likelihood as $L(\theta \mid y)$. Because of the independence in the data, we have that

$$L(\theta \mid y) = \pi(y \mid \theta) \overset{\text{ind}}{=} \prod_{i=1}^{n} \pi(y_i \mid \theta). \tag{1.5}$$

MLE has three essential steps:

- Define the likelihood based on the probability distribution for the data given the parameters.
- Find the log-likelihood.
- Maximize the log-likelihood with respect to the parameters.

The first step in each case is simply to specify the likelihood for the appropriate probability distribution. In the second step we take the natural logarithm (here: $\log_e$) of $L(\theta \mid y)$, denoted $l(\theta \mid y)$. This will simplify the coming computations greatly, because of the fact that $\log_e \prod_{i=1}^{n} \pi() = \sum_{i=1}^{n} \log_e \pi()$. The natural logarithm is chosen because it is an increasing function, and therefore secures that maximizing the log-likelihood is equal to maximizing the likelihood. Finally, we want to find the MLE values for the parameters. To maximize $l(\theta \mid y)$, we first find its derivative with respect to each parameter. Second, we set each derivative equal to zero and solve the equation for the appropriate parameter. This solution gives the MLE values for the parameters.

A relatively common procedure is to combine MLE and MCMC simulation. First, we find the MLE values for the parameters of a model using the procedure described above. Then, we want to use an MCMC algorithm to obtain samples from the desired distribution. By using the MLE parameter values as starting points for the sampling process, burn-in time can be notably reduced. This procedure is part of the results in paper 2, where we use MLE together with AD and Laplace approximation to obtain the estimated parameter values. These values are afterwards used as initial values in HMC sampling. From the sampling we obtain a distribution of samples for the model parameters.

Chapter 2

Numerical integration methods

The posterior distribution in equation (1.3) is relatively easy to express using the Bayesian theory from section 1.1. The issues arise when we want to evaluate it. In practice, it is very common to encounter a case where direct evaluation is impossible. To be able to say something about the posterior distribution, we make use of numerical methods. These numerical methods are used to obtain approximations that can give information about the posterior distribution. In paper 1 we use HMC and consider how the numerical integration schemes within the algorithm can be improved. We know that HMC is a branch of the well-known MCMC algorithms. For HMC and other MCMC algorithms, a latent variable model that is high-dimensional can cause problems, among other things related to time consumption. INLA is a numerical method that can be applied for LGMs. `TMB` offers efficient implementation of complex latent variable joint likelihoods, and maximizes an approximation of the marginal likelihood using MLE. The SPDE approach can be combined with other numerical methods in order to simplify computations. In paper 2 and paper 3, we use the SPDE approach in combination with `TMB`.

2.1 Markov chains

A Markov chain is defined as a stochastic process that holds a specific property. This property is the Markov property and says that the process is without memory. Markov chains were first introduced by Andrey Markov in the early 1900s.

Consider a stochastic process $x = x_1, x_2, \ldots$. Define the discrete state space as all the values that can be taken by x_i and define p as the distribution of x. By definition the sequence x is a Markov chain if for the states j and $i_1, i_2, \ldots, i_{n-1}, i$:

$$p_{i,j} = p(x_{n+1} = j \mid x_1 = i_1, x_2 = i_2, \ldots, x_n = i) = p(x_{n+1} = j \mid x_n = i), \tag{2.1}$$

where $p_{i,j}$ is a transition probability (i.e. the probability of moving to a certain state given the current state) and n is the n^{th} step. In other words, equation (2.1) says that the distribution of x_{n+1} given all previous states is memoryless and should depend only on the previous state x_n. This trait is what we refer to as the Markov property. We define the transition matrix P as the square matrix that holds all the transition probabilities $p_{i,j}$.

The Markov chain definitions above hold for a discrete state space. In MCMC, we often handle cases with continuous state space (but still discrete time). In this case, we define the continuous state space S and let $x \in S$ and $A \subset S$. Moreover, we define the transition kernel $P(x,\cdot)$ so that $P(x,A)$ gives the probability of reaching the set A from state x.

Then, define the transition density $p(x,\cdot)$ as the distribution of $\cdot$ given that $X_n = x$. We define $p(x,y)$ so that the following holds:

$$P(x,A) = \int_{y \in A} p(x,y)dy, \tag{2.2}$$

$$P(x,S) = \int_{y \in S} p(x,y)dy = 1, \tag{2.3}$$

which means that it is non-negative and that it is a probability density function.

A stationary process is one that is not affected by a jump in time. In a Markov chain, a stationary distribution is a probability distribution that does not change in time. The probability distribution $\pi()$ is a stationary distribution for the Markov chain for all $A \subset S$ when

$$\pi(A) = \int P(x,A)\pi(x)dx. \tag{2.4}$$

This can also be expressed as $\pi P = \pi$. This means that the probability distribution is invariant by the transition kernel. Stationarity is an important factor that we will re-encounter in section 2.3. The same goes for reversibility, which means that there exist a function π that fulfills

$$\pi(x)p(x,y) = \pi(y)p(y,x) \tag{2.5}$$

This means that it is the same to move from x to y in the Markov chain as it is to move from y to x. This trait is sometimes referred to as detailed balance and it implies stationarity.

2.2 Monte Carlo methods

Monte Carlo methods (or Monte Carlo simulation) can in practice be seen as any method within statistical inference that uses numerical simulation to solve problems. Monte Carlo methods can be used to look more closely at traits belonging to a model. When we sample, we aim to generate a large amount of samples that mirror the actual behaviour of the model. Thereafter, we study the traits of the samples in order to say something about the model itself.

Monte Carlo integration is an essential part of Monte Carlo methods. This method uses the law of large numbers to estimate the solution of integrals. The law of large numbers says that the average of a sample converges in probability to the expected value, typically formulated as follows. Let $x_1, x_2, \dots$ be independent and identically distributed random variables that have a finite first moment $E(x_i) = \mu$. Then,

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} x_i \xrightarrow{p} \mu. \tag{2.6}$$

In Monte Carlo integration, we transform an integration problem into an expectation problem and then use the law of large numbers. Say that we want to find $\int_a^b g(x)dx$

where $\{x_i\}$ are random variables distributed as $f(x)$. Then, the mathematical expectation of $E(g(x))$ is $E(g(x)) = \int_{-\infty}^{\infty} g(x)f(x)dx$. By the law of large numbers we now know that

$$\frac{1}{N}\sum_{i=1}^{N} g(x_i) \tag{2.7}$$

will converge in probability to the expected value. This is why we want to transform the integration problem into an expectation problem. As an example, let $x \sim U(a,b)$ for x_i where $i = 1,\dots n$ and find $\int_a^b g(x)dx$. With the expansion

$$\int_a^b g(x)dx = (b-a)\int_a^b g(x)\frac{1}{(b-a)}dx, \tag{2.8}$$

we get that

$$(b-a)\int_a^b g(x)f(x)dx = (b-a)E(g(x)). \tag{2.9}$$

when x is uniformly distributed. We have now transformed the problem and can use the law of large numbers for $E(g(x))$. This is one example of a general approach. Typically, the Monte Carlo methods can therefore use the independent and identically distributed random variables $\{x_i\}$ and generate (many) samples in order to find properties of a function.

In general, Monte Carlo methods are applied for estimating unknown parameters by using random sampling. We can for instance estimate Type I and II error rates (when the decision about the null hypothesis gives false positive or false negative) and test power (the probability that a false null hypothesis will be rejected). Monte Carlo methods are relevant in several areas such as finance, research, gambling and engineering.

2.3 Markov chain Monte Carlo

The work of Metropolis et al. (1953) introduced what we today know as the Metropolis algorithm in MCMC. Later, Hastings (1970) edited the Metropolis algorithm into what we now call the Metropolis-Hastings algorithm. A third method, the Gibbs sampler, is seen as a special case of the Metropolis-Hastings algorithm and was introduced by Geman and Geman (1987). These algorithms are all part of MCMC.

In fact, MCMC is a framework of methods proposed to be used in situations where we want to draw samples from a target density, for instance the posterior distribution. The overall aim is to perform random sampling by using Markov chains. In the previous section, Monte Carlo methods were used when generating samples by making use of a sequence $\{x_i\}$ of independent and identically distributed random variables. In MCMC, we make use of Markov chains instead. When we want to sample from the target density using MCMC, the first step is to create a Markov chain. Here, the distribution that you want to sample from is the stationary distribution (the one that is discussed in section 2.1). The states of this chain are then considered to be sample proposals from the target distribution. By introducing a criterion that accepts or rejects the proposals, we finally end up with a collection of samples that behave as though they were drawn from the true distribution.

The Metropolis-Hastings algorithm follows the MCMC recipe, and generates a Markov chain $\{x_t\}$ with the stationary distribution mimicking its target distribution $p(x)$. The algorithm proposes to change states from x_t to x_{t+1}. The proposal x^* is drawn from a proposal distribution $g(x^* \mid x_t)$. The Metropolis-Hastings algorithm is set up like in Algorithm 1. Figure 2.1 shows a proposal distribution and a target distribution and illustrates the Metropolis-Hastings sampling procedure. The initial value and first proposal are both exemplified in the figure 2.1.

Algorithm 1 The Metropolis-Hastings algorithm

```
Given initial value $x_0$:
for $t = 1, \ldots, T$ do
    Draw $x^* \sim g(x^* \mid x_t)$
    Define acceptance ratio $r = \frac{\pi(x^*) \times g(x_t|x^*)}{\pi(x_t) \times g(x^*|x_t)}$
    Define acceptance probability $\alpha = \min(r, 1)$
    if $\alpha > \texttt{Uniform}(0, 1)$ then
        $x_{t+1} = x^*$
    else
        $x_{t+1} = x_t$
    end if
end for
```

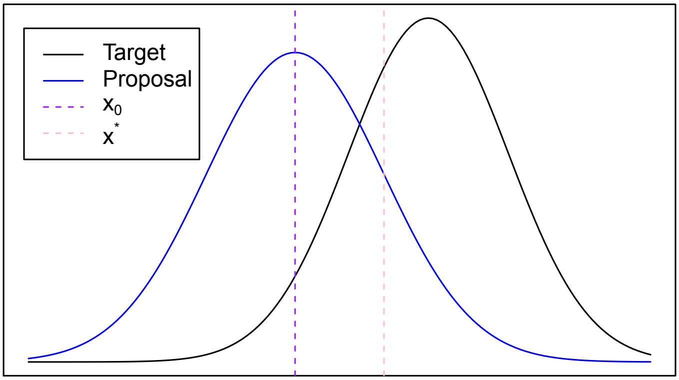

Figure 2.1: Illustration of target and proposal distribution with initial value and proposal sample for Metropolis-Hastings.

A second example of an MCMC algorithm is the Gibbs sampler, a common choice in cases where multivariate target distributions are involved. Let the vector

$x = x_1, \ldots, x_d \in \mathbb{R}^d$. In Gibbs sampling we split x in two so that $x = \{x_i, x_{-i}\}$, where $x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$. We have that $i = 1, \ldots, d$ and sample from each conditional density $f(x_i \mid x_{-i})$. See Algorithm 2 for the structure of the algorithm.

Algorithm 2 The Gibbs sampler

```
Given initial $x = (x_{1,0}, \ldots, x_{d,0})$ at $t = 0$:
for $t = 1, ..., T$ do
    Set $x_1 = x_{1,(t-1)}$
    for $i = 1, \ldots, d$ do
        Generate $x^*_{i,t} \sim f(x_i \mid x_{-i})$
        Update $x_i = x^*_{i,t}$
    end for
    Let $x_t = (x^*_{1,t}, \ldots, x^*_{d,t}))$
end for
```

There are challenges connected to MCMC sampling. One standard issue is the burn-in-period, which refers to the amount of samples in the beginning of the chain that we choose to discard. Because of the initial value guess, we may need a certain amount of time before obtaining "reasonable" proposals. This is solved by selecting a number of burn-in samples and starting to collect samples only after that period. This clearly makes the procedure more time-consuming. In addition, MCMC samples are correlated because each proposal is drawn using the current sample. With larger correlation, the iteration run for reaching the stationary distribution gets longer. A larger correlation is also connected to a larger variance. The correlated samples therefore increase computation time as well as giving more variation in the samples. This will have a negative effect on the quality of the samples. The decrease in quality can be found by considering a measure of how many independent and identically distributed samples that are needed in order to obtain the same variance as with correlated samples. Moreover, we would in theory prefer a quite high acceptance probability in the MCMC algorithms. However, it is common with MCMC that (because of higher correlation) the moves are small when the acceptance probability is higher (Rizzo, 2007). Issues such as these that mentioned here inspire the exploration of new and improved numerical methods for sampling.

2.4 Hamiltonian Monte Carlo

The background of Hamiltonian dynamics originates from the physics genre. Duane et al. (1987) combined Hamiltonian dynamics and MCMC in to hybrid Monte Carlo. Hybrid Monte Carlo is now more commonly known as Hamiltonian Monte Carlo (HMC).

HMC is a branch of MCMC. Just like with MCMC, HMC also generates samples from a distribution where direct sampling is unfeasible. We recall that MCMC methods have correlated samples, and that relatively high acceptance rates (for the next state) are often associated with short moves. This means that many steps are required before reaching a steady state, which is time consuming. Many of the MCMC algorithms (like Metropolis-Hastings and Gibbs) use random walk behaviour in the process of reaching the target distribution. This can also have a negative effect time wise. HMC diminishes the correlation between samples and therefore has the possibility to obtain both high acceptance rates and long moves at the same time (Neal, 2010). Compared to MCMC, HMC can thus use fewer steps to reach the same state. HMC avoids the random walk behaviour that is typical for many of the MCMC algorithms because of the way the HMC algorithm (Algorithm 3) proposes the trajectory of states. All of this can have a positive effect on the computation time, which decreases when steps are longer without exploring random walk.

HMC is often visualized as a puck that can slide without friction throughout a surface where the height varies. The state of the system consists of the position and momentum of the puck, given by two two-dimensional vectors q and p. Furthermore, there exists potential and kinetic energy $U(q)$ and $K(p)$, that together conserves the total energy through movement in the system. When Hamiltonian dynamics are applied outside physics and in the case of HMC, the position vector is the variable that is truly of interest. The momentum vector can be considered as more of an auxiliary variable. In the general case, the two vectors q an p are $d-$dimensional and the system of q and p is described by the Hamiltonian $H(q,p)$. For change in time, this is described by a system of differential equations including the partial derivatives of q and p.

The HMC method is distinguished from MCMC methods by the use of potential and kinetic energy. In addition, the gradient is evaluated in the process of creating the sample proposal for a new state. In HMC, Hamiltonian dynamics hold important properties. These are time reversibility and preservation of volume. The importance and validity of these properties will be explained in section 2.4.2.

The proposal state consists of position $q \in \mathbb{R}^d$ and momentum $p \in \mathbb{R}^d$, where d is dimension. The change in q and p over time is described by the partial derivatives of the Hamiltonian and numerical integration in HMC is carried out with the following system of differential equations:

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}, \tag{2.10}$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}, \tag{2.11}$$

where $i = 1,...,d$. We say that Hamiltonian dynamics are described by the Hamiltonian

$$H(q,p) = U(q) + K(p), \tag{2.12}$$

where the potential energy U is defined as minus the log probability density of the distribution for the position q and the kinetic energy K is defined as

$$K(p) = p^T M^{-1} p/2, \tag{2.13}$$

where M is a symmetric, positive-definite matrix (Neal, 2010). In fact, K is the negative log probability of a multivariate normal distribution (MVN) $\sim \mathtt{MVN}(0,M)$ where M is the covariance matrix. With this, we can rewrite equations (2.10) and (2.11) so that:

$$\frac{dq_i}{dt} = [M^{-1}p]_i, \tag{2.14}$$

$$\frac{dp_i}{dt} = -\frac{\partial U}{\partial q_i}. \tag{2.15}$$

Solving the differential equations is carried out by approximation, making use of a numerical integration scheme.

2.4.1 Numerical integration schemes

The reason for introducing numerical integration schemes is the need to approximate the solution of the differential equations in section 2.4. This is done by discretizing time using the time step ε. A well known approximation method for solving a system such as equations (2.10)-(2.11) is Euler's method where the steps are as follows (for $t = 0$ and when ∇ represents the gradient):

$$\begin{aligned} p(\varepsilon) &= p(0) - \varepsilon \nabla U(q(0)), \\ q(\varepsilon) &= q(0) + \varepsilon p(0). \end{aligned} \tag{2.16}$$

If repeated, we see that after starting at time zero and states being computed at time ε, we can get the states at time 2ε, 3ε and so on. This process can be repeated for L steps. The Euler's method easily encounters problems like divergence or long calculation time. Therefore, an improved version is to prefer, such as a modified version that uses the updated value $p(\varepsilon)$ in stead of $p(0)$ when computing the position variable $q(\varepsilon)$. Because of volume preservation, the problem with divergence decreases for this updated method (Neal, 2010).

From these simple cases we get to know the structural setup of numerical methods that are used to solve a system of differential equations. We also see that there are relatively simple ways to improve a method and increase the accuracy of the proposals, such as the small modification made to the Euler's method. This also includes adding extra partial steps. One such method is the leapfrog method (Neal, 2010). The leapfrog method is the commonly used numerical integration scheme in HMC sampling and is set up as follows (details are given in paper 1):

$$\begin{aligned} p(\varepsilon/2) &= p(0) - \frac{\varepsilon}{2} \nabla U(q(0)), \\ q(\varepsilon) &= q(0) + \varepsilon p(\varepsilon/2), \\ p(\varepsilon) &= p(\varepsilon/2) - \frac{\varepsilon}{2} \nabla U(q(\varepsilon)). \end{aligned} \tag{2.17}$$

It is clear that the leapfrog method (2.17) follows the same procedure as we saw for Euler's method, but with more updates and partial steps.

The purpose of the numerical integration scheme in HMC is to update the position (and momentum) variable in time through the use of previous time steps and evaluation of the gradient. With this as a purpose, there is clearly room for alterations and innovations in the design of the numerical integration methods that are used in HMC. In paper 1, we compare the performance of the leapfrog method to the performances of three new numerical schemes that also can be used in the HMC algorithm. These three new schemes are developed using the splitting method (Blanes et al., 2014).

To explain the splitting method, let $(q_i, p_i) \rightarrow (q_{i+1}, p_{i+1})$ represent a time step such that it corresponds to a transformation in the phase space:

$$(q_{i+1}, p_{i+1}) = \psi_\varepsilon(q_i, p_i), \tag{2.18}$$

where the mapping ψ is volume preserving and reversible. Moreover, consider a more general version of the transformation:

$$(q^*, p^*) = \psi_\varepsilon(q_i, p_i), \tag{2.19}$$

where (q^*, p^*) are the proposals and (q_i, p_i) are the current position and momentum. With this general transformation, we want to describe a possible transition from a current pair of position and momentum to a proposal state. We already know that in HMC, moving from one state to the next is dependent on using a numerical integration scheme. We can derive from equation (2.19) that ψ represents such a scheme.

We thereafter want each step of the integration scheme to be defined within ψ. Let a φ represent each partial time step (for instance the leapfrog method has three partial time steps). Moreover, let $\circ$ represent function compositions. This way, the leapfrog method can be written as

$$\psi_\varepsilon = \varphi^B_{\varepsilon/2} \circ \varphi^A_\varepsilon \circ \varphi^B_{\varepsilon/2}, \tag{2.20}$$

where each of the three φ refers to one of the three partial time steps of equation (2.17). Each subscript of φ represents the length of the time step, while superscript A and B denotes whether the step belongs to an update in the position or the momentum variable respectively.

Equation (2.20) is specifically defined for the leapfrog method. We can replace it with a more general expression that represents additional numerical integration schemes for the HMC algorithm. There are possible variations, as long as the expression remains palindromic (this means that the expression is the same read forwards and backwards). Blanes et al. (2014) suggest the following general composition:

$$\psi_\varepsilon = \varphi^A_{a_1\varepsilon} \circ \varphi^B_{\varepsilon/2} \circ \varphi^A_{(1-2a_1)\varepsilon} \circ \varphi^B_{\varepsilon/2} \circ \varphi^A_{a_1\varepsilon}. \tag{2.21}$$

The reason for focusing on the variation in equation (2.21), is that it is a more general form of the leapfrog scheme that was represented by (2.20). We see that if $a_1 = 0$, equation (2.21) reduces to exactly the leapfrog method (2.20). In paper 1 we look at the development of numerical integration schemes that are determined by different values of a_1 in (2.21). We consider the performance of these different five-step algorithms and compare them to the leapfrog method. Although the leapfrog method is a three-step algorithm, it is a special case and originates from the same palindromic expression (2.21) as the new numerical schemes. In paper 1, we look at three different numerical schemes that are developed using the splitting method. Evaluation of their performance in the HMC algorithm reveals that there are more efficient schemes than the standard leapfrog method.

2.4.2 The Hamiltonian Monte Carlo algorithm

In the HMC algorithm each iteration goes through two steps. First there is a change in momentum and then a possible change in position and momentum. Let (q^*, p^*) be the proposed state that is accepted with probability α. The HMC algorithm starts with randomly drawing a new p value from the Gaussian distribution. The next step is to obtain the full proposed state, which is done by doing a Metropolis sampling. This is repeated for L steps, where at last position q is then either accepted or rejected based on α. The momentum update is used just for finding α and can be omitted after each step. Algorithm 3 shows the general HMC procedure.

Algorithm 3 The HMC algorithm

```
Given current state $(q_i, p_i)$:
Randomly draw new $p$ from Gaussian distribution
for $i = 1, ..., L$ do
    simulate full position variable $q$ using appropriate integrator
    simulate full momentum variable $p$ using appropriate integrator
end for
$\alpha = \min\{1, \exp[H(q_i, p_i) - H(q^*, p^*)]\}$
if $\alpha > \texttt{Uniform}(0,1)$ then
    $q_{i+1} = q^*$
else
    $q_{i+1} = q_i$
end if
```

By using the leapfrog method or another appropriate numerical integration scheme, the HMC algorithm perform a Metropolis update for the current state and we end up with a trajectory of states that is obtained by the use of Hamiltonian dynamics. Because of the way that we obtain the trajectory of proposal states we can have long moves and high acceptance rates at the same time, whereas the random walk behaviour that many of the MCMC algorithms use to propose states have random direction. This may decrease computation time for HMC compared to MCMC.

In Algorithm 3, the term appropriate integrator simply refers to an integrator with step size ε that holds the properties from section 2.4. For obtaining the best results, the choices of ε and number of steps L are critical. If ε is too large or too small, this will either lead to small acceptance rates or long computation time, a problem similar to the problems for MCMC methods that were discussed previously. Also, the trajectory length εL is important, and an uneducated choice of L could lead to exploration by random walk (Neal, 2010). Originally the values of ε and L are user-specified in the HMC algorithm. The No-U-Turn sampler (NUTS) (Hoffman and Gelman, 2014) is an addition to the standard HMC algorithm that adaptively tunes these values. The step size ε is determined by use of primal-dual averaging. NUTS thereafter eliminates the need to choose L by introducing a recursive algorithm that explores a wide trajectory and stops when it is moving backwards, or thus when it is making a u-turn. One reason why the HMC algorithm has not become more popular can possibly be the user specified parameters, and the problems they may cause. By including NUTS, the process of using HMC is simplified and more robust.

In section 2.4 it is mentioned that the Hamiltonian dynamics hold two important properties. The first is time reversibility. The Metropolis proposal that we have in the HMC algorithm requires reversibility, because that is how it is ensured that the stationary distribution is equal to the target distribution. For HMC, let T_t denote the mapping from (q, p) at time 0 to (q_t, p_t) at time t. If we have reversibility in T_t, it means that T_{-t} must map $(q_t, -p_t)$ to $(q, -p)$. By negating the equations (2.10) and (2.11) we achieve this inverse mapping. The second property is volume preservation. When the Hamiltonian dynamics are volume preserving, the determinant of the Jacobian (matrix of all the first order partial derivatives) of the mapping is equal to one (Beskos et al., 2013). Without volume preservation, we would need to compute the determinant of the Jacobian in relation to the change in volume in the acceptance probability in Algorithm 3. Volume preservation is the same as having a vector field with zero divergence (Neal, 2010). Here, equations (2.10) and (2.11) is the vector field F and we define the divergence of F as $\nabla \bullet F$. For HMC, this becomes

$$\begin{aligned} \text{div } \mathbf{F} &= \sum_{i=1}^{d} [\frac{\partial}{\partial q_i} \frac{\partial}{\partial p_i} \cdot \frac{dq_i}{dt} \frac{dp_i}{dt}] \quad = \sum_{i=1}^{d} [\frac{\partial}{\partial q_i} \frac{dq_i}{dt} + \frac{\partial}{\partial p_i} \frac{dp_i}{dt}] \\ &= \sum_{i=1}^{d} [\frac{\partial}{\partial q_i} \frac{\partial H}{\partial q_i} - \frac{\partial}{\partial p_i} \frac{\partial H}{\partial q_i}] = \sum_{i=1}^{d} [\frac{\partial^2 H}{\partial q_i \partial p_i} - \frac{\partial^2 H}{\partial p_i \partial q_i}] \\ &= 0, \end{aligned} \tag{2.22}$$

and thus we have volume preservation. All the numerical integrations schemes that are used in the HMC algorithm must hold these properties. The schemes in paper 1 that are developed from the splitting method fulfill this requirement.

HMC can clearly be an efficient alternative to the standard MCMC methods. However, there are situations where both HMC and other MCMC methods are time consuming and not an optimal choice. This is for example the case for latent variable models where the joint posterior or likelihood (of latent variables and parameters) is high-dimensional. Some of these models come with high correlation. For algorithms such as the Metropolis-Hastings and the Gibbs sampler the model itself may be quite easy to implement, but the performance is limited (see e.g. Betancourt and Girolami (2015)). When there is correlation in the model there is a need for many iterations in order to fully traverse the target distribution. This leads to random walk and very long computation times. With HMC, the correlation is less of an issue. However, the implementation of the model is more difficult. As we know from Algorithm 3, it requires both integration of the Hamiltonian and the finding derivatives of the target. Although there are suggestions for how to overcome these issues (see e.g. the programming language Stan (Carpenter et al., 2017)), it is also possible to look outside the MCMC framework and find numerical methods that are appropriate and fast for the latent variable models in question.

2.5 Integrated nested Laplace approximation

INLA is implemented in the `R` package `INLA` (from now on: `R-INLA`) as a method for doing Bayesian inference with LGMs (Rue et al., 2009). These models have a latent field that is Gaussian. Because of properties belonging to certain latent fields that will be discussed later in this section, INLA is regarded as a computationally effective alternative in Bayesian inference compared with other numerical methods.

The LGMs are models where MCMC simulation has been a popular choice for approximation method. However, there are several disadvantages to MCMC in these situations. The elements of the latent field are dependent on the hyperparameters of the model. This is a well-known computational issue with MCMC and using MCMC on complex latent variable models is known to be very time consuming. Several of the

MCMC algorithms have been tested with LGMs, but issues remain nonetheless (see e.g. Liu (2008), Robert and Casella (2013) and Gamerman (1997)). INLA, however, offers numerical calculation of the posterior and is arranged to perform well for LGMs. For a high-dimensional joint posterior, INLA will be significantly faster than the MCMC simulations.

First, consider the typical hierarchical form of an LGM:

$$y_i \mid x, \theta \sim \pi(y_i \mid \eta_i, \theta), \tag{2.23}$$

$$\eta_i = \sum_j c_{ij} x_j, \tag{2.24}$$

$$x \mid \theta \sim \texttt{MVN}(0, \Sigma), \tag{2.25}$$

$$\theta \sim \pi(\theta), \tag{2.26}$$

where x is a latent field and θ and y as before are parameter vector and data vector respectively. The linear predictor η has covariate values c_{ij}. We see from (2.25) that the latent field has a covariance matrix Σ. This is a dense matrix. The latent field x is also a Gaussian field (GF) if it is jointly distributed with a multivariate Gaussian distribution with a zero mean vector and covariance matrix Σ, which is the case here. Even though GFs are quite common, they are also computationally demanding to work with. This mainly has to do with the dense covariance matrix. Performing factorizations on Σ (such as Cholesky decomposition) comes with a high cost.

INLA uses properties of Gaussian Markov random fields (GMRFs). For the GF x to be a GMRF it must hold two essential properties that both are assumed for INLA (Rue et al., 2009). First, there must be conditional independence in x. This means that x must hold the pairwise Markov property. For $i \neq j$, this means that the variables x_i and x_j are independent conditional on all the other variables. The second property is that the number of hyperparameters is small. When this holds, the GF x is a GMRF with precision matrix Q and the conditional independence also ensures that Q is sparse (Rue and Held, 2005). There is a relationship between the covariance matrix and the precision matrix such that

$$\Sigma = Q^{-1}. \tag{2.27}$$

Clearly, we prefer numerical computation with sparse matrices rather than dense matrices. INLA avoids the dense covariance matrix completely because of the relationship (2.27) and this is a fundamental part of its efficiency. In section 3.2, the efficiency of using GMRFs and the precision matrix is made use of in the SPDE approach for spatial statistics. This approach, which creates a link between GFs and GMRFs and which is implemented in `R-INLA`, is relevant for paper 2 and paper 3. For the actual approximation of the marginal, we have not used INLA or `R-INLA` (see section 2.6). The approximation methods of INLA are therefore just mentioned briefly below.

Recall from section 1.1 that we have to use numerical approximation when we are interested in finding the posterior (1.3). With INLA we want to find the marginal posteriors for the latent field, which are:

$$\pi(\theta_k \mid y) = \int \pi(\theta \mid y) d\theta_{-k} \tag{2.28}$$

and

$$\pi(x_j \mid y) = \int \pi(x_j \mid \theta, y) \pi(\theta \mid y) d\theta. \tag{2.29}$$

We see that the posterior marginals are nested, and that we need to find $\pi(\theta \mid y)$ and $\pi(x_j \mid \theta, y)$. The former of these two is the posterior and is approximated as

$$\tilde{\pi}(\theta \mid y) \propto \frac{\pi(y \mid x, \theta)\pi(x \mid \theta)\pi(\theta)}{\tilde{\pi}(x \mid \theta, y)}\Big|_{x=x^*(\theta)},$$

where $x^*(\theta)$ is the mode and $\tilde{\pi}(x \mid \theta, y)$ is the Laplace approximation (see section 2.6) for the latent field full conditional. The other marginal posterior, $\pi(x_j \mid \theta, y)$, is found using yet another Laplace approximation.

2.6 The Template Model Builder

In paper 2 and paper 3, the use of GMRFs and the precision matrix is essential. Even though INLA gives a methodology for approximating marginal posteriors, it has not been applied in these papers. Instead, we have used AD and Laplace approximation combined in the `R` package `TMB` (Kristensen et al., 2016) for optimization of the marginal likelihood. `TMB` is used for complex latent variable models. In practice, the user specifies a joint log-likelihood in `C++`, while the rest of the implementations are carried out in `R`.

`TMB` operates in the following manner: we use MLE for the Laplace approximation of the marginal likelihood, while the latent variables are integrated out automatically using AD. AD is in short a set of computer techniques that can find the derivative of a function. Any complicated function consists of a combination of simple mathematical operations such as subtraction, division and logarithm. This also means that the function can be arranged into an sequence of functions of simple operations. Each of these will have a simple derivative. By using the chain rule, a computer program can find the derivative of a complex function automatically. The derivatives we obtain are used in the Laplace approximation of the marginal likelihood. Calculation of these derivatives may be very tedious for the user without AD, which is the case for INLA. The optimization of the Laplace approximation is carried out in `R`. `TMB` is relevant for several types of models, but in paper 2 and paper 3 we focus on GMRF models and make use of the sparse precision matrix.

The Laplace approximation is a special form of Taylor series expansion where we use the first three terms. Let us first consider a simple case where we approximate the logarithm (for simplicity) of a function $f(x)$ around its global maximum x_0. The expansion is written as

$$\log f(x) \approx \log f(x_0) + \frac{\partial \log f(x_0)}{\partial x}(x - x_0) + \frac{\partial^2 \log f(x_0)}{2\partial x^2}(x - x_0)^2. \tag{2.30}$$

The first order derivative of $\log f$ in the maximum x_0 is equal to zero. The second order derivative of $\log f$ in x_0 is negative. Inserting this, the new expression becomes

$$\log f(x) \approx \log f(x_0) - \frac{1}{2}|\frac{\partial^2}{\partial x^2}\log f(x_0)|(x - x_0)^2. \tag{2.31}$$

Define $\hat{\sigma}^2 = \frac{1}{|\frac{\partial^2 \log f(x_0)}{\partial x}|}$. By inserting $\hat{\sigma}^2$ and taking the integral and exponential on both sides, we get the following new expression:

$$\begin{aligned}\int f(x)dx &\approx \int \exp(\log f(x_0) - \frac{1}{2\hat{\sigma}^2}(x-x_0)^2)dx \\ &= c\int \exp(-\frac{1}{2\hat{\sigma}^2}(x-x_0)^2)dx,\end{aligned} \tag{2.32}$$

where c is a constant. We now see that the final expression in equation (2.32) is similar to the known integral of a normal distribution $\mathcal{N}(x_0, \hat{\sigma}^2)$, and by using the appropriate normalizing factor this can thus be used as an approximation. This is a simple introduction to the idea behind the Laplace approximation. However, the idea can be extended to approximating the marginal likelihood of the latent variable models that we use in TMB. In fact, the sparse precision matrix of the GMRF models enables the Laplace approximation for models with very many latent variables (Kristensen et al., 2016).

Let us for now redefine x as the latent variables $x \in \mathbb{R}^n$. If we then let $f(x, \theta)$ be the negative log-likelihood, this is the function that is specified by the user and written out in C++. We want to use MLE and Laplace approximation for the marginal likelihood $L(\theta)$. Here, we therefore define the MLE for θ as the one that maximizes $L(\theta)$ where

$$L(\theta) = \int_{\mathbb{R}^n} \exp(-f(x,\theta))dx. \tag{2.33}$$

Let thus $\hat{x}(\theta)$ define the value that minimizes $f(x,\theta)$ with respect to x. If we now continue by doing the Taylor expansion around $\hat{x}(\theta)$, we will find that equation (2.33) is starting to look similar to a version of (2.32). Moreover, we define the Hessian of $f(x,\theta)$ with respect to x and calculated in the minimizing value $\hat{x}(\theta)$ as $H(\theta)$. Inserting the Taylor expansion and the Hessian gives the equation

$$L(\theta) \approx C\int \exp\{-\frac{1}{2}[x-\hat{x}(\theta)]^T H(\theta)[x-\hat{x}(\theta)]\}dx, \tag{2.34}$$

where C is a constant. Equation (2.34) now is similar to the MVN distribution. With the the appropriate normalizing factor, we get that the Laplace approximation for $L(\theta)$ is defined as

$$L^*(\theta) = (2\pi)^{n/2}|H(\theta)|^{-\frac{1}{2}}\exp(-f(\hat{x},\theta)). \tag{2.35}$$

The final step is to find the estimate of θ that minimizes the negative logarithm of $L^*(\theta)$. While TMB returns the negative log-likelihood, this is done with an appropriate optimizer in R.

Chapter 3

Spatial modeling

3.1 Spatial statistics

The field of spatial statistics includes all areas where there exist a spatial relationship in the data and where this relationship is used in the statistical calculations. Based on this, we can define spatial data as data that in some way are based on location and where the data points can be mapped in geographical relation to each other. The study of spatial data is common in very many academic areas, such as epidemiology and econometrics. The interest lies in gaining insight in factors that are affected or determined by spatial location. A famous example in spatial epidemiology is one that goes back to 1854 and Dr. John Snow's theory that cholera was transmitted through the drinking water. Another example from 1948 is Blum (1948) who saw sunlight as a factor in skin cancer. In spatial epidemiology, the use of statistical models can among other things enable research on risks for getting a certain disease (or similar outcome) or the probability of how and where a disease is spreading. Such knowledge, using estimation and prediction, gives the opportunity of taking necessary action or preparing for possible demanding outcomes.

We can typically distinguish between three types of spatial data. These are:

- Area data.
- Geostatistical data.
- Spatial point patters.

Consider area data first. If we let D be a domain $\in \mathbb{R}^d$, then area data is found in the cases when the data is spread in a closed region within D. This region is again divided in several parts (based on smaller regions or similar systems) that are subject for the spatial study. Typically, neighboring structures and disease mapping (discrete data) are common here (Blangiardo and Cameletti, 2015).

Geostatistical data (also known as point-referenced data) are the data where each observation is connected to one specific location, for instance coordinates belonging to certain houses or all hospitals within a region.

Finally, spatial point patterns refers to the cases where the locations (within a region) are random, based on the event taking place or not taking place (Blangiardo and Cameletti, 2015).

3.1.1 Spatial stochastic processes

A stochastic process is defined as a collection of random variables

$$\{Z(t) \mid t \in T\}, \tag{3.1}$$

indexed by time t in the set T. Intuitively, $Z(t)$ can be seen as the state of the process at time t and is a random variable holding a certain probability distribution. The whole collection of these states is what we call the stochastic process, for both discrete and continuous cases. The state space of the process is the set of possible values for each state in the process.

We can turn to spatial modeling when our data includes any type of the geographical information that is presented in section 3.1. In other words, this means that there is some sort of location $s = s_1, \ldots, s_n$, connected to each observation and that the different locations are in geographical reference to each other (s_n being the final location). We can then let the spatial data be represented by a stochastic process

$$U(s) = \{u(s),\ s \in D\}, \tag{3.2}$$

where D is still a domain (Blangiardo and Cameletti, 2015).

In the case of the three different types of spatial data that were established in section 3.1, we have for area data that $\{u(s)\}$ represent aggregate data within the boundaries of area $s \in D$. For geostatistical data, the process $\{u(s)\}$ represent outcomes at given locations. Here, s is normally a coordinate vector. When data come as spatial point patterns, $\{u(s)\}$ define if an event has happened or not. In such cases, all the values of s are random.

3.2 The stochastic partial differential equations approach

As mentioned in Chapter 1, the MCMC simulation of the integral that gives the the marginal distribution (1.4) in the posterior distribution can be extremely demanding computationally. The same holds for complex latent variable models. Methods that are efficient in overcoming these obstacles are therefore always of interest. Section 2.5 and 2.6 explain how we can use different numerical methods to find good and fast approximations to marginal posteriors and likelihoods.

GFs are very important in spatial statistics. However, they are also linked to computational issues. This comes from the fact that the GF has a dense covariance matrix Σ. To perform operations on Σ, like finding the inverse, there is a great computational cost. In section 2.5 we saw that having a latent field that is a GMRF instead of a GF is a huge advantage because of the sparse precision matrix connected to it. In this section, we will discuss the SPDE approach (Lindgren et al., 2011). This approach can be combined with both INLA and `TMB` by utilizing a link between GFs and GMRFs. The SPDE approach proposes to make a GMRF representation of the GF that has a certain SPDE as its solution. In paper 2 and paper 3, we use the SPDE approach to get GMRF representations of the latent fields. Then, we evaluate the log-likelihood with the methodology from `TMB` that is explained in section 2.6.

In statistics, the variance tells us how a variable varies. The covariance is seen as a measure of the relationship between two variables and how changes that occur in one variable relate to changes that occur in the other one. In other words, the covariance show how the two vary together. The covariance function describes the spatial covariance of a stochastic process or GF. With the covariance function, we get the covariance of the process at every two locations. The covariance function can be stationary, which means that we only consider the position that two locations have relative (not absolute) to each other. The covariance function can also either be non-isotropic or isotropic, referring to whether or not the covariance is dependent on spatial direction. This means that for an isotropic covariance function, we consider only the absolute value of the distance between the two locations.

The Matérn covariance function is a stationary and isotropic covariance function that provides the covariance between spatial coordinates. The Matérn covariance function is given as

$$\mathrm{Cov}(d) = \frac{\sigma_u^2}{\Gamma(\lambda)2^{\lambda-1}}(\kappa d)^{\lambda}\mathrm{K}_{\lambda}(\kappa d), \tag{3.3}$$

$$\sigma_u^2 = \frac{\Gamma(\lambda)}{\lambda(\alpha)(4\pi)^{r/2}\kappa^{2\lambda}\tau^2}, \tag{3.4}$$

where d is the absolute distance between two locations, K is the modified Bessel function of order λ, κ is a scaling parameter and the parameter σ_u^2 refers to the marginal variance of the spatial random effects. Parameter τ controls the marginal variance. We have that the default value of $\lambda = 1$ and that α is determined by $\alpha = \lambda + r/2$ (Blangiardo and Cameletti, 2015). Because r is the dimension of the location space, we are only interested in $r = 2$ in paper 2 and paper 3. This means that $\alpha = 2$ in these cases.

The starting point of the whole process is the following SPDE:

$$(\kappa^2 - \Delta)^{\alpha/2}(\tau u(s)) = \Omega(s), \tag{3.5}$$

where $s \in \mathbb{R}^2$, Δ is the Laplacian, α controls smoothness and Ω(s) is a Gaussian spatial white noise process. Here, the GF u with Matérn covariance is a solution to the SPDE (3.5).

Following the SPDE approach, we first construct an irregular grid of triangles. Each triangle has three vertices (i.e. edges) and every two triangles meet in one or zero vertices. We want most of the vertices near where the observations are located, and then have supporting vertices added around them. Thereafter, the aim is to construct a GMRF representation of the GF on the grid. To do this, we begin with defining the weak solution of the SPDE (3.5) as

$$\int \zeta_j(s)(\kappa^2 - \Delta)^{\alpha/2}\tau u(s)ds \overset{d}{=} \int \zeta_j(s)\Omega(s)ds, \tag{3.6}$$

where $\overset{d}{=}$ means equal in distribution. This should hold for every finite set of test functions $\{\zeta_j(s)\}$ when $j = 1, \dots, m$ (Lindgren et al., 2011).

With the weak solution above, we want to use the finite element method to represent it. This is done by letting

$$u(s) = \sum_{n=1}^{N} a_n(s)w_n, \tag{3.7}$$

where $\{a_n\}$ are basis functions, $\{w_n\}$ are Gaussian weights with mean zero and N is the number of vertices on the irregular grid. Moreover, the basis functions are such that $a_n = 1$ at vertex n and 0 otherwise.

By letting $m = N$, we move forward in the solution by finding the distribution for $\{w_n\}$ that solves (3.6) for only a finite set of test functions. For our case $\alpha = 2$, we let $\zeta_n = a_n$. This is the approximation to the solution of the SPDE. A main result of the SPDE approach is to use Neumann boundary conditions (here: the normal derivative is zero at the surface) to find the precision matrix Q for the Gaussian weights $\{w_n\}$ (Lindgren et al., 2011). The reason why this is so valuable, is that we now have $u(s)$ represented as a GMRF with the inverse of the sparse precision matrix Q^{-1}. Like for INLA in section 2.5, this means that by using the SPDE approach we avoid computations with Σ completely. This is very valuable, because the SPDE approach can be combined with other numerical methods such as `TMB`. In the SPDE approach, Q consists of three matrices and is defined as

$$\tau^2(K_{\kappa^2}C^{-1}K_{\kappa^2}), \tag{3.8}$$

where $(K_{\kappa^2})_{ij} = \kappa^2 C_{ij} + G_{ij}$. If we define the inner product $< f, g >$ as $\int f(s)g(s)ds$, then the matrix C consists of inner products of the basis functions. The matrix G consists of inner products of the gradients of the basis functions.

The SPDE approach is implemented in the `R-INLA` package. The triangulated grid is called a mesh and can be created simply based on a set of locations (coordinates) provided by the user. As discussed, the coordinates are placed in proper reference to each other on vertices inside the grid, together with additional supporting vertices. Together, this forms the mesh. Figure 3.1 shows an example of the structure of a triangulated mesh. After creating the mesh, its structure is used in the process of finding the approximate solution to the SPDE (3.5) and thereby the inverse of the precision matrix. Because the mesh by construction holds a GMRF representation of the field, we can extract the three matrices in (3.8) from it.

We know that working with a GMRF with a sparse precision matrix rather than a GF with a dense covariance matrix is very beneficial computationally. By introducing the SPDE approach, we get an efficient way of representing GFs as GMRFs and the covariance matrix Σ as Q^{-1}. Merging the SPDE approach with methodologies such as `TMB` can therefore be very advantageous.

3.3 The spatial and spatio-temporal models

In paper 2 and paper 3 we find spatial and spatio-temporal models. The first model considers birth weight and the spatial correlation between hospitals. The second model covers caesarean section (CS) and CS rates in municipalities over a given time period, in addition to spatial correlation between the municipalities. The third model is a joint model between CS and severe maternal hemorrhage (SMH). This model is used to evaluate the correlation between the two responses. These models, that all contain spatial random effects, are built up in a hierarchical manner.

In paper 2, we use a normally distributed spatial model for birth weight $y_i = \eta_{ij} + \varepsilon_{ij}$, where j is hospital and i represent each individual. Then, η is the mean and is equal to

$$\eta_{ij} = X_i^T\beta + u_j + v_j. \tag{3.9}$$

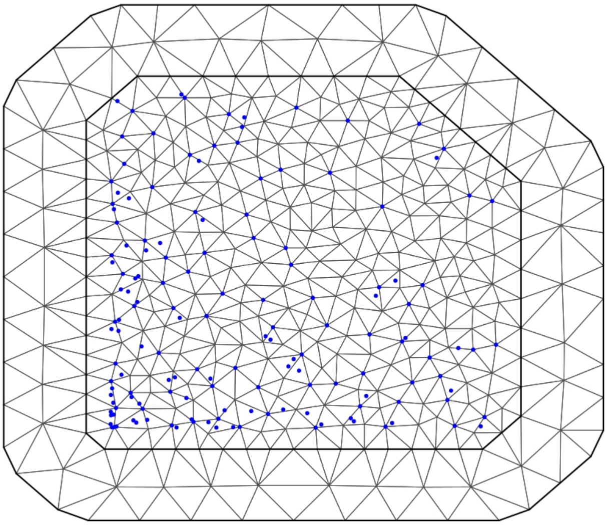

Figure 3.1: Example of a mesh constructed for a given set of coordinates (blue). The mesh is created using `R-INLA` function: `inla.mesh.2d()` with parameter settings `cutoff` $= 0.05$ and `max.edge` $= \mathrm{c}(0.05, 0.2)$.

While X and β are covariate vector and covariate effect vector, the hospital effect v_j has a normal distribution with mean 0 and variance σ_v^2. The spatial random effect u has a MVN distribution with mean vector $(0,\dots,0)$ and covariance matrix Σ.

In paper 3, there are two spatio-temporal models. We let t denote the time, j represent the municipalities and i the individual. The model for CS, y_{ijt}, is Bernoulli distributed with probability p_{ijt}. This probability is connected to a linear predictor η_{ijt} via a logit link function

$$\log\left(\frac{p_{ijt}}{1-p_{ijt}}\right) = \eta_{ijt} = X_i^T\beta + u_{jt} + \omega t, \tag{3.10}$$

where X and β are covariate vector and covariate effects vector respectively. The spatial random effect u has a MVN distribution with mean vector $(0,\dots,0)$ and covariance matrix Σ and also, u_{jt} is specific to time and municipality and follows an AR(1) process. The parameter ω is a linear time trend. The joint model between CS and SMH is bivariate Bernoulli with probabilities $p_{ijt}^{(1)}$ and $p_{ijt}^{(2)}$. The CS probability $p_{ijt}^{(1)}$ is connected to a linear predictor $\eta_{ijt}^{(1)}$ via the logit link function

$$\log\left(\frac{p_{ijt}^{(1)}}{1-p_{ijt}^{(1)}}\right) = \eta_{ijt}^{(1)} = x_i^T\beta^{(1)} + u_{jt}^{(1)} + \omega^{(1)}t + u_{jt}^{(3)}, \tag{3.11}$$

and the SMH probability $p_{ijt}^{(2)}$ is connected to $\eta_{ijt}^{(2)}$ via a logit link

$$\log\left(\frac{p_{ijt}^{(2)}}{1-p_{ijt}^{(2)}}\right) = \eta_{ijt}^{(2)} = x_i^T\beta^{(2)} + u_{jt}^{(2)} + \omega^{(2)}t + \xi u_{jt}^{(3)}. \tag{3.12}$$

When $k = 1,2$, $\beta^{(k)}$ and $\omega^{(k)}$ are covariate vectors and linear time effects. Let the three spatial random effects be denoted $u^{(1)}, u^{(2)}$ and $u^{(3)}$, where the shared spatial random effect is connected by ξ. Then, parameter ξ plays a role in the model correlation $\mathrm{corr}(u^{(1)} + u^{(3)}, u^{(2)} + \xi u^{(3)})$.

All of the models above are examples of latent variable models where the latent field is distributed with a dense covariance matrix Σ. As explained earlier, it would be likely to experience computational issues with MCMC methods for models like this. With the SPDE approach we can use GMRFs with a sparse precision matrix for a latent field u. With `TMB` we can find approximations for the MLE values of the model parameters. For these latent variable models, this gives a huge computational advantage, especially for high-dimensional cases.

3.4 Additional results

This section covers additional results related to the models in paper 3. Central elements and ideas from paper 3 will be considered known here.

The CS model in paper 3 with linear predictor (3.10) includes two time effects. The first is the linear time trend ωt, which is positive and independent of municipality. The second is u_{jt}, a spatial random effect that follows an AR(1) process and that has separate values for each pair of year and municipality (this means that every individual in municipality j and year t share u_{jt}). If there is a linear time trend, it will be present in ω. If there is a time trend that is non-linear, it should be identified in the AR(1) process. Recall that the process is defined as

$$u_{jt} = \alpha u_{j,t-1} + \varepsilon_{jt}. \tag{3.13}$$

Figure 3.2 shows the the development of u_{jt} from year 2001 to year 2014 for four large municipalities in Norway. From paper 3 we know that the autocorrelation in the AR(1) function $\alpha = 0.92$. This value of α indicates that we do not expect to see a very large variation in u_{jt}, especially for the municipalities that are large such as those in Figure 3.2. Some of the very small municipalities (that is: very small in population size) are much more prone to experience a year with an unusual high or low CS rate. We see in Figure 3.2 that there is just small variation in u_{jt} for each of the four municipalities and no clear trends to explain.

The second addition is to explore the sensitivity of the CS model. This can be done by altering the value of u_{j0} to ensure a stationary AR(1) process for the spatial random effects. The process is defined in (3.13), and we can see from it that the variance for the GMRF process becomes $(1-\alpha^2)Q$ (where Q as we recall is the precision matrix). For the first year of the process, each u_{j0} has to be assigned a value. For the CS model in paper 3, we started the process simply with variance Q for the GMRF. Now, we start

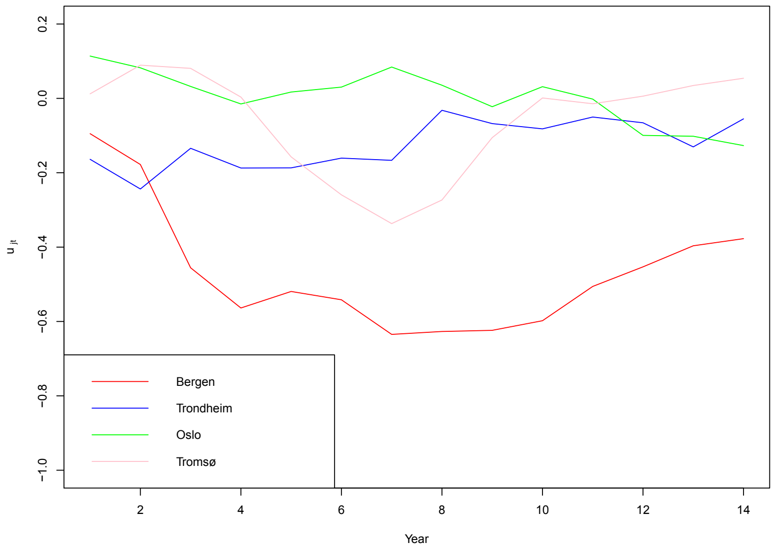

Figure 3.2: Spatial random effect u_{jt} for a selection of municipalities from year 2001 to year 2014.

the process with variance $(1-\alpha^2)Q$ for u_{j0} as well. This gives a stationary variance throughout the process. We now call this model the stationary model.

Other than the changing starting point, the stationary model is fitted in the exact same way as for the CS model in paper 3. The new estimates for the model parameters are found in Table 3.1. Table 3.1 show some differences in results compared to the results for the corresponding CS model in paper 3. The estimated values of the linear time trend ω, the scaling parameter κ and marginal variance σ_u are somewhat smaller in the stationary case. The change in κ also brings a small increase in ρ. The estimated value of α (which was already high) has increased slightly. The new map of the estimated CS rates for year 2001 and year 2014 is found in Figure 3.3 The overall impression of the maps in Figure 3.3 is that there is much similarity to the map in paper 3. However, there are also some clear differences. In year 2001, we see that Fig-

Parameter	Value	SD
σ_u	0.087096	0.006453
α	0.940640	0.010768
ω	0.022460	0.004798
κ	$2.2\times e^{-5}$	$2.0\times e^{-6}$
	Derived parameter	
ρ	130.366	141.914

Table 3.1: Estimated parameter values with standard errors for spatial and time parameters for stationary model. The value of ρ is given in kilometers.

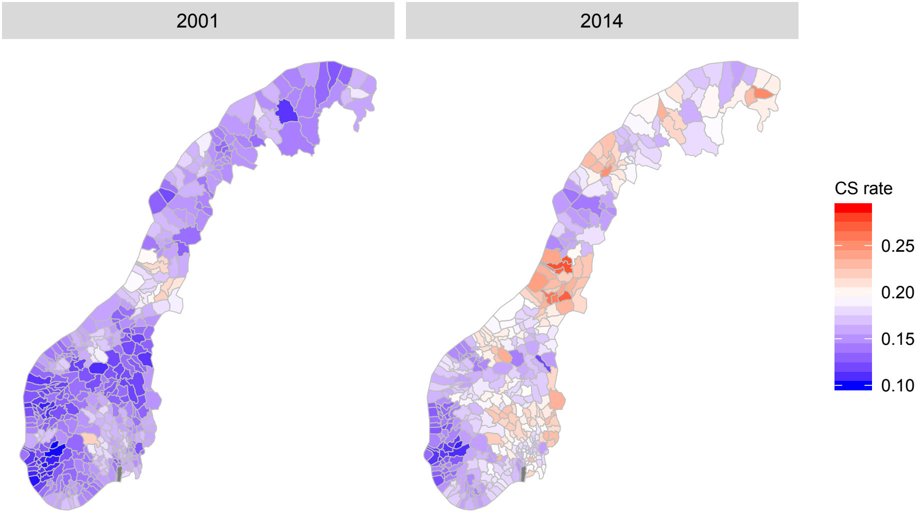

Figure 3.3: Estimated CS rates in Norway by municipality for year 2001 and year 2014.

ure 3.3 show a somewhat higher estimated CS rate in the east and middle of the map, while having a somewhat smaller CS rate in the west. For year 2014, it is harder to spot any differences. This is expected, because we have changed the variance of the first year spatial random effects. The stationarity could easily be implemented for the joint CS and SMH model in paper 3 as well. However, it has not been executed in this case because the fitting of the joint model is particularly time consuming. In addition, the small differences in the results for the stationary CS model and original CS model indicate that it would have little significance.

3.5 Health registry data

For the spatial models that are used in two of the papers that are part of this thesis, registry data from a Norwegian health registry have been used. In Norway there exist several different health registries. Among others, there is a birth registry, a cardiovascular disease registry and a prescription registry. The main objectives of the national registries are to analyze data, to enable research and to contribute in improving healthcare. Based on the interest of each separate registry, various information about individuals is collected and stored.

Some of the data the registries collect are both sensitive and identifiable. This means that the data itself makes it possible to identify individuals in a data set, for instance with personal identification number or a collection of information that is sufficient to recognize an individual. Moreover, this also applies to the instances where the data itself is considered to be sensitive information. Examples are race, religion and certain health related data. This clearly makes the way we handle the registry data very crucial, and there are several ways to tackle the issue. Certain registries require patient consent, which means that data will be collected only for those individuals who agree to be included. Other registries are allowed to collect data based on compulsory notification. This means that the data is collected for every relevant incident.

The Medical Birth Registry of Norway (MBRN) is an example of a registry that uses compulsory notification. All maternity units report information on the same variables, such as complications during pregnancy and labour and mother's health, to the registry. The MBRN also stores personal identification numbers. Individuals like parents, child and siblings are linked within the registry. The MBRN was founded in 1967 and now contains information on around 3 million births (Norwegian Institute of Public Health, 2016). Both the data about birth weight, and the data about CS and SMH are collected from the MBRN, and thus paper 2 and paper 3 both contain results obtained from accessing these data. The MBRN has given access to two different data sets, but they contain a lot of similar information. However, the birth weight data set is considered less sensitive than the CS data set, because while the former only contains the hospital where the birth takes place, the latter also includes the municipality where the birth mother lives.

Registry data is accessible to the public and to researchers in various forms. Variables that makes the data identifiable can be recoded so that only those with the encryption key will be able to see the true value. This is for instance done with identification numbers. However, even though identification might not be possible directly, the data can still be considered sensitive. It is always crucial that the data is stored in ways that are secure. For researchers who use data from the Norwegian health registries there are several alternatives, based on how sensitive the data they request are considered to be. An easy first alternative is to use a data set that only contains variables that has the appropriate level of anonymity, meaning that it is impossible to identify an individual based on information that is found in the data set. In these cases, the data is prepared by the registry and sent to the researcher. The only involvement of the data owners is that they will inspect the article before you can publish anything containing analysis of the data. This is the case for the data in paper 2. However, it is sometimes necessary to acquire data that contains more details and that cannot be made unidentifiable. In these cases, the data will not be given to the researcher to use on a personal computer. Instead, it is necessary to use a secure server with a two factor authentication procedure. The data is stored on this server, and all work that requires using the data must be carried out on the server. Within the server, there is access to all the programs you normally use for working with health registry data like `R` and `SPSS`. In paper 3 the data is considered sensitive because, among other things, the information that gives the location of each individual is so accurate that it in combination with other variables makes some of the data identifiable. Therefore, all results in paper 3 have been obtained on a secure server.

The necessity of handling health registry data with care is explained in the previous paragraph. However, this caution typically leads to tedious processes for accessing the sensitive data. Because of the need for numerous applications and approvals, researchers often are hesitant to enter the process. To simplify this practice, a national collaboration was the startup of the Health Registries for Research (HRR). They want to meet the needs of the research community by offering easier access to data, statistical support and secure servers (HRR, 2018). For instance, the HRR has developed a metadata database (containing no sensitive data) that gives an outline of the different registries and their content, and a possibility to download some data for testing purposes. Solutions like this makes it easier to test the data you are interested in. One purpose of registry data, is to enable research. If the process of acquiring data is too demanding it will lead to fewer research projects and fewer relevant results. Therefore, the HRR and similar initiatives hold an important task in making health registry data more accessible.

Chapter 4

Summary of papers

4.1 Summary of Paper 1: ”On the application of improved symplectic integrators in Hamiltonian Monte Carlo”

Janne Mannseth, Tore Selland Kleppe and Hans Julius Skaug. "On the application of improved symplectic integrators in Hamiltonian Monte Carlo." Communications in Statistics - Simulation and Computation (2017).

This paper evaluates how the leapfrog method, the commonly used numerical integration scheme in Hamiltonian Monte Carlo (HMC), performs compared to a selection of new integrations schemes.

HMC is a Markov chain Monte Carlo (MCMC) method. We go through the important steps of how to use Hamiltonian dynamics that originate from physics to obtain the HMC method. Using HMC, which requires two tuning parameters for the leapfrog method (step size ε and number of steps L), gives potential of outperforming standard MCMC methods.

Blanes et al. (2014) proposes a method for developing new numerical integration schemes to be used in HMC. These schemes could be more efficient than the leapfrog method. The particular schemes that are considered in paper 1 have the same underlying properties as the leapfrog method, but with more partial steps and thus different intervals between the time steps. Using this approach, paper 1 contains the development of three new schemes.

Using a standard Gaussian $d-$dimensional model, the efficiency of the four numerical schemes is compared. One scheme especially stands out and is able to reach a high expected acceptance probability using only approximately $\frac{1}{9}$ of the steps required when using the leapfrog method. The two other new methods also clearly exceed the performance of the leapfrog method for the standard Gaussian model.

As the first results only hold for the specific Gaussian case, paper 1 also considers a second approach. By introducing the No-U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014), the need for user specified tuning parameters disappears. This sampler contains methods for tuning L and ε and enables the use of new models with the numerical integration schemes.

Using NUTS and data from Girolami and Calderhead (2011), the four schemes are compared using efficient sample size (ESS) per CPU time as a measure, both for a logistic regression model and a multivariate student t distribution. For the logistic model, all the schemes developed using the Blanes et al. (2014) method have advantage over the leapfrog method. For the student t model, the new schemes mostly obtain higher ESS and step size than the leapfrog method. Still, the results of ESS per CPU time are more nuanced, perhaps because of cost related to the increased number of gradient evaluations for the new methods. However, one of the new methods still outperforms the leapfrog method as dimension increases.

4.2 Summary of Paper 2: ”Robustness of the SPDE approximation under short range spatial correlation”

Janne Mannseth, Geir Drage Berentsen and Hans Julius Skaug.

In paper 2, we evaluate short-range spatial correlation. The purpose of this paper is to show the behaviour of the stochastic partial differential equations (SPDE) approach when the data have little or no spatial correlation.

This paper uses data from the Medical Birth Registry of Norway, more specifically births that have been registered at a Norwegian hospital. The response, birth weight, is approximately normally distributed. Moreover, the model includes random effects. These effects are the spatial random effect u and the hospital effect, v. The spatial information available for the data is the coordinates of the hospital at which the birth took place.

The spatial effect is distributed with a multivariate normal distribution and a spatially structured covariance matrix Σ. The calculation of Σ is computationally difficult.

Two approaches are considered in this paper. The first is the SPDE approach. Spatial data can be represented as spatial processes and these processes are Gaussian fields (GFs) (Blangiardo and Cameletti, 2015). The issue with the use of GFs, is that it requires inverting the large dense covariance matrix Σ . It is as a proposed solution to this problem that the SPDE approach is so relevant. It instead uses a sparse precision matrix of which the inverse is seen as a representation of Σ. The SPDE approach is implemented in the `R` package `R-INLA`.

These specific data about birth weight include information on location down to the level of hospital. In Norway there is mostly just one hospital in each city, which means that nearly all inhabitants of this city will give birth in the same hospital. Moreover, birth weight is not expected to vary geographically within Norway. Because of this, the spatial correlation in the model is expected to be small. In paper 2, we find that the SPDE approach yields inaccurate results when the spatial correlation is close to zero. We compare the SPDE approach with the results that come from including the exact covariance matrix in the model. This is a possibility for these data because the dimension of Σ (which is based on the number of hospitals) allowed it to be inverted without being too computationally demanding. Comparison of the two approaches gives that the SPDE approach differs quite much from the exact case for certain parameter estimates, especially in the marginal variance of the spatial random effect. It also seems that the SPDE approach experiences aliasing in the variance of the hospital effect and random effect. This means that it is random if the variance is put in u, v or a mixture of both of them when there is short-range spatial correlation.

4.3 Summary of Paper 3: "Joint modeling of Caesarean section and severe maternal hemorrhage using spatio-temporal Gaussian random fields"

Janne Mannseth, Geir Drage Berentsen, Hans Julius Skaug and Dag Moster.

In paper 3, we consider spatial correlation for Caesarean section (CS) between municipalities. We also consider the relationship between CS and a medically associated outcome, severe maternal hemorrhage (SMH). The data comes from the Medical Birth Registry of Norway.

The SPDE approach uses a link between GFs and Gaussian Markov random fields (GMRFs). GMRFs have great computational advantages. We make use of the SPDE approach (Lindgren et al., 2011) to enable computation with GMRFs and the sparse precision matrix Q instead of GFs and the covariance matrix Σ. We use `TMB` to implement the likelihood. `TMB` is based on a method that uses Laplace approximation for the marginal likelihood, where automatic differentiation is used in the Laplace approximation. Finally, maximum likelihood estimation is used to find the parameter estimates.

The first model is a Bernoulli distributed model for CS. We know that the CS rates vary between hospitals in Norway. Using the CS model, we get estimates of the covariate effects and the linear time trend. We find that the linear time trend is positive (which is expected). In addition, the CS model includes a spatial random effect following an $\mathrm{AR}(1)$ process. These are used to find the estimated CS rates for each pair of year and municipality, which means that every individual in the same year and municipality share the same CS rate. We also estimate the spatial correlation between the municipalities. We use a range measure that gives the radius at which the spatial correlation is going below 0.1. For the CS model, this radius is large enough to cover multiple municipalities. We find that there is spatial correlation, and that the CS rates are somewhat clustered with geographical location.

The second model is a joint bivariate Bernoulli model for the two responses CS and SMH. This model has many similarities to the CS model, and each response has its own spatial random effect that follow an $\mathrm{AR}(1)$ process. In addition, a new spatial random effect that is shared between CS and SMH is added to the model. We estimate the model correlation, and find that it is very small. The joint model is set up to work for any Bernoulli response that is medically associated with CS, but the optimization of the joint model is very time consuming. This makes it more difficult to test many different variations of the optimization, such as different initial values or even another second response.

Chapter 5

Errata

Below follows a list of errata.

- p.6: "a stationary distribution is a probability distribution that does not change as time changes" is changed to "a stationary distribution is a probability distribution that does not change in time".

- p.6: "the same goes for reversibility, which means that there exist a function h that fulfills $h(x)p(x,y) = h(y)p(y,x)$" is changed to "the same goes for reversibility, which means that there exist a function π that fulfills $\pi(x)p(x,y) = \pi(y)p(y,x)$".

- p.9: "Hybrid Monte Carlo is now more commonly known as HMC" is changed to "Hybrid Monte Carlo is now more commonly known as Hamiltonian Monte Carlo (HMC)".

- p.13: "see e.g. the programming language Stan (Carpenter et al., 2017)" is changed to "(see e.g. the programming language Stan (Carpenter et al., 2017))".

- p.16: "The former of these two is the posterior and is defined as $\pi(\theta \mid y) \propto \frac{\pi(y|x,\theta)\pi(x|\theta)\pi(\theta)}{\tilde{\pi}(x|\theta,y)}|_{x=x^*(\theta)}$" is changed to "The former of these two is the posterior and is approximated as $\tilde{\pi}(\theta \mid y) \propto \frac{\pi(y|x,\theta)\pi(x|\theta)\pi(\theta)}{\tilde{\pi}(x|\theta,y)}|_{x=x^*(\theta)}$".

- p.17: "If we then let $f(x,\theta)$ be the joint log-likelihood" is changed to "If we then let $f(x,\theta)$ be the negative joint log-likelihood".

- p.21: "This means that for an isotropic covariance function, we consider only the absolute value of the position that two locations have relative to each other" is changed to "This means that for an isotropic covariance function, we consider only the absolute value of the distance between the two locations".

- p.21: "$(K_{\kappa^2}) = \kappa^2 C_{ij} + G_{ij}$" is changed to "$(K_{\kappa^2})_{ij} = \kappa^2 C_{ij} + G_{ij}$".

- p.30: "The response, birth weight, is normally distributed" is changed to "The response, birth weight, is approximately normally distributed".

Bibliography

Beskos, A., N. Pillai, G. Roberts, J.-M. Sanz-Serna, A. Stuart, et al. (2013). Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli 19*(5A), 1501–1534. 2.4.2

Betancourt, M. and M. Girolami (2015). Hamiltonian Monte Carlo for hierarchical models. *Current trends in Bayesian methodology with applications 79*, 30. 2.4.2

Blanes, S., F. Casas, and J. Sanz-Serna (2014). Numerical integrators for the hybrid Monte Carlo method. *SIAM Journal on Scientific Computing 36*(4), A1556–A1580. 2.4.1, 2.4.1, 4.1

Blangiardo, M. and M. Cameletti (2015). *Spatial and spatio-temporal Bayesian models with R-INLA*. John Wiley & Sons. 3.1, 3.1.1, 3.2, 4.2

Blum, H. F. (1948). Sunlight as a causal factor in cancer of the skin of man. *Journal of the National Cancer Institute 9*(3), 247–258. 3.1

Carpenter, B., A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell (2017). Stan: A probabilistic programming language. *Journal of statistical software 76*(1). 2.4.2

Duane, S., A. D. Kennedy, B. J. Pendleton, and D. Roweth (1987). Hybrid Monte Carlo. *Physics letters B 195*(2), 216–222. 2.4

Gamerman, D. (1997). Sampling from the posterior distribution in generalized linear mixed models. *Statistics and Computing 7*(1), 57–68. 2.5

Geman, S. and D. Geman (1987). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. In *Readings in Computer Vision*, pp. 564–584. Elsevier. 2.3

Girolami, M. and B. Calderhead (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(2), 123–214. 4.1

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*(1), 97–109. 2.3

Hoffman, M. D. and A. Gelman (2014). The No-U-Turn Sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research 15*(1), 1593–1623. 2.4.2, 4.1

HRR (2018). Facilitating the use and security of norwegian health registries in research. https://hrr.w.uib.no/. [Accessed on December 11, 2018]. 3.5

Kristensen, K., A. Nielsen, C. W. Berg, H. Skaug, and B. M. Bell (2016). TMB: Automatic Differentiation and Laplace Approximation. *Journal of Statistical Software 70*(5), 1–21. 2.6, 2.6

Lindgren, F., H. Rue, and J. Lindström (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(4), 423–498. 3.2, 3.2, 3.2, 4.3

Liu, J. S. (2008). *Monte Carlo strategies in scientific computing*. Springer Science & Business Media. 2.5

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics 21*(6), 1087–1092. 2.3

Neal, R. M. (2010). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo 54*, 113–162. 2.4, 2.4, 2.4.1, 2.4.2

Norwegian Institute of Public Health (2016). Overview of the national health registries. https://www.fhi.no/en/more/access-to-data/about-the-national-health-registries2/. [Accessed on December 11, 2018]. 3.5

Rizzo, M. L. (2007). *Statistical computing with R*. Chapman & Hall/CRC. 2.3

Robert, C. and G. Casella (2013). *Monte Carlo statistical methods*. Springer Science & Business Media. 2.5

Rue, H. and L. Held (2005). *Gaussian Markov random fields: theory and applications*. CRC press. 2.5

Rue, H., S. Martino, and N. Chopin (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology) 71*(2), 319–392. 2.5, 2.5

Chapter 6

Papers

Paper I

6.1 On the application of improved symplectic integrators in Hamiltonian Monte Carlo

Janne Mannseth, Tore Selland Kleppe and Hans Julius Skaug

Communications in Statistics - Simulation and Computation (2017)

Paper II

6.2 Robustness of the SPDE approximation under short range spatial correlation

Janne Mannseth, Geir Drage Berentsen and Hans Julius Skaug

Paper III

6.3 Joint modeling of Caesarean section and severe maternal hemorrhage using spatio-temporal Gaussian random fields

Janne Mannseth, Geir Drage Berentsen, Hans Julius Skaug and Dag Moster

Graphic design: Communication Division, UiB / Print: Skipnes Kommunikasjon AS

uib.no

ISBN: 9788230855058 (print)
9788230852231 (PDF)