Approximation techniques for Bayesian Logistic Regression

By

MAURO CAMARA ESCUDERO

School of Mathematical Sciences
UNIVERSITY OF SOUTHAMPTON

A dissertation submitted to the University of Southampton in accordance with the requirements for module MATH3031 in the School of Mathematical Sciences.

MAY 2019

ABSTRACT

Many tasks in Bayesian Statistics can be seen as performing expectation with respect to a posterior distribution over the parameter space. This often requires evaluating an integral without a closed-form solution due to intractable likelihoods. This work presents three approximation techniques for such integrals: MCMC, Laplace and Variational. We show that in the context of uni-modal Bayesian Logistic Regression, the Laplace method performs as well as a Random-Walk Metropolis-Hastings algorithm, and outperforms a Gaussian local variational method independently of the number of parameters involved. Interestingly, we show that the Variational method's mode can be closer to the true population mode than the Laplace one for small sample sizes, and could therefore be preferred in situations with small data sets where estimation of the parameter of interest is of most importance.

TABLE OF CONTENTS

List of Figures

Figure **Page**

CHAPTER 1

INTRODUCTION

Historically, the interpretation and definition of probability has engaged many philosophers, scientists and mathematicians. In the early days, the Frequentist interpretation was most prominent because it was in line with one of the core principles of the scientific method: reproducibility. In this interpretation, probabilities have meaning only if defined in the context of reproducible and repeatable experiments. They represent the limiting behavior of the ratio of the number of success over the number of attempts, usually called *trials*, as the number of trials goes to infinity. Although compatible with the working routines of many scientists, this interpretation has two major drawbacks: it does not allow probabilities to be defined for one-off, non-reproducible, experiments, and it often leads to seemingly unintuitive interpretations of the results.

The Bayesian interpretation promises to deliver a more intuitive framework that works also when no long-run frequency is involved. It does so by interpreting probabilities as a way of quantifying the uncertainty around a *statement*, for instance, Bayesian statisticians would be able to give meaning to the probability that the sun will rise tomorrow, whereas this statement would make no sense to a Frequentist statistician. The pitfall of the Bayesian framework is that while it allows greater flexibility and incorporation of subjective knowledge, it can be more difficult to implement computationally due to the requirement to compute integrals that have no closed-form and that are numerically intractable, in practice.

In this introductory chapter we will see how these intractable expressions arise and an overview of the three main methods used to tackle the issue. Chapter 2 will explore stochastic approximations, while Chapter 3 focuses on deterministic approximations instead. Finally, Chapter 4 presents a series of computational results comparing the three methods in the context of Bayesian Logistic Regression.

1.1 Bayesian Setup and Shortcomings

Throughout this thesis I will assume that we are interested in understanding some characteristics of a **population**, from which we have observed data $\mathbf{x} = (x_1, \ldots, x_n)^\top$ as being the realization of a **sample** of *n independent* random variables $\mathbf{X} = (X_1, \ldots, X_n)^\top$. We denote the *unknown* characteristics of the population by the vector of parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)^\top$ taking values in the $p-$dimensional **parameter space** Θ.

Often we have some *a priori* knowledge about the values of the parameters which we can express via a probability distribution $p(\boldsymbol{\theta})$ called the **prior distribution**. It communicates our subjective belief around the distribution of values of the parameters *before* we see the data. Observing the data will likely provide us with additional information that was not available before, it is therefore reasonable to imagine that our prior beliefs might need to be updated in some way. The corresponding **posterior distribution** represents our *revised* beliefs about the characteristic of the population after seeing the data, and we denote it by $p(\boldsymbol{\theta} \mid \mathbf{x})$.

Bayes Theorem provides a direct and intuitive way of updating our prior beliefs:

$$p(\boldsymbol{\theta} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathbf{x})} \tag{1.1}$$

Here $p(\mathbf{x} \mid \boldsymbol{\theta})$ is called **likelihood function** and it is sometimes also denoted by $\mathscr{L}(\boldsymbol{\theta}; \mathbf{x})$. For a vector of values for $\boldsymbol{\theta}$, it gives the *likelihood* that such values represent the actual characteristic of the population, given that we have observed data $\mathbf{x}$. The denominator $p(\mathbf{x})$ is called **model evidence** and it acts as a *normalization constant*, making sure that the posterior distribution integrates to 1 when $\boldsymbol{\theta}$ are continuous random variables, or sums up to 1 when they are discrete.

In other words, Bayes Theorem tells us that our uncertainty after seeing the data is proportional to the product of the information about $\boldsymbol{\theta}$ contained in the data, and our prior knowledge about $\boldsymbol{\theta}$:

$$\underbrace{p(\boldsymbol{\theta} \mid \mathbf{x})}_{\text{posterior}} \propto \underbrace{p(\mathbf{x} \mid \boldsymbol{\theta})}_{\text{likelihood}} \underbrace{p(\boldsymbol{\theta})}_{\text{prior}} \tag{1.2}$$

In practice, many of the problems with the Bayesian approach revolve around intractable integrals. Below we give an overview of some of the most common issues arising in Bayesian statistics [19, 11, 1]:

1. *Inference*: Obtaining the full posterior distribution requires us to compute the model evidence.

$$p(\mathbf{x}) = \int_\Theta p(\mathbf{x} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\, d\boldsymbol{\theta} \tag{1.3}$$

2. *Prediction*: Determining the distribution of new data point.

$$p(\mathbf{x}_{\text{new}} \mid \mathbf{x}) = \int_\Theta p(\mathbf{x}_{\text{new}} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \mathbf{x})\, d\boldsymbol{\theta} \tag{1.4}$$

3. *Posterior Expectation*: Several features of the posterior distribution are also of interest, such as the posterior expectation

$$\mathbb{E}[\boldsymbol{\theta} \mid \mathbf{x}] := \int_\Theta \boldsymbol{\theta}\, p(\boldsymbol{\theta} \mid \mathbf{x})\, d\boldsymbol{\theta} \tag{1.5}$$

4. *Estimation*: Bayes estimator $\widehat{\boldsymbol{\theta}}(\mathbf{X})$ is chosen to minimize the posterior expected loss

$$\rho(\widehat{\boldsymbol{\theta}}(\mathbf{X}) \mid \mathbf{X}=\mathbf{x}) := \mathbb{E}[L(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}(\mathbf{X})) \mid \mathbf{X}=\mathbf{x}] = \int_{\Theta} L(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}(\mathbf{x}))\, p(\boldsymbol{\theta} \mid \mathbf{x})\, d\boldsymbol{\theta} \tag{1.6}$$

   where $L(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}})$ is the loss incurred in estimating $\boldsymbol{\theta}$ with $\widehat{\boldsymbol{\theta}}$, e.g. $L(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}) := (\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})^2$.

5. *Credible Regions*: Credible regions with a chosen confidence level γ are found by

$$C(\mathbf{x}) := \{\boldsymbol{\theta} \in \Theta : p(\boldsymbol{\theta} \mid \mathbf{x}) \geq k\} \qquad \text{where} \qquad \mathbb{P}(p(\boldsymbol{\theta} \mid \mathbf{x}) \geq k \mid \mathbf{x}) = \gamma \tag{1.7}$$

Notice equations (1.3)-(1.7) require us to compute integrals that can be interpreted as the expectation of some function $f(\boldsymbol{\theta})$ with respect to the posterior distribution $p(\boldsymbol{\theta} \mid \mathbf{x})$.

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{x})}[f(\boldsymbol{\theta})] = \int_{\Theta} f(\boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \mathbf{x})\, d\boldsymbol{\theta} \tag{1.8}$$

For instance, one can choose $f(\boldsymbol{\theta}) = \boldsymbol{\theta}$ and $f(\boldsymbol{\theta}) = p(\mathbf{x})$ in (1.5) and (1.3) respectively.
In practice these integrals often have no closed-form solution or are numerically intractable. Developing theory and methodology to overcome these issues is the central focus of an area of Bayesian statistics called *Approximate Inference*.

1.2 Overview of Approximate Inference

1.2.1 Stochastic Approximations

Their drawback is that they require a large computational effort that does not scale well with dimensionality of the parameter space Θ, in other words, they suffer from the *curse of dimensionality*. On the other hand, they have the nice property of asymptotic convergence as sample size increases.

- *Monte Carlo methods*: Approximates expectations by averaging *independent* samples drawn directly[1] from the desired distribution.
- *Markov Chain Monte Carlo methods*: Runs a Markov Chain producing *dependent* samples that asymptotically mimic samples from the desired distribution. These can then be used to approximate any of the integral problems (1.3)-(1.7) using Monte Carlo approximations.

1.2.2 Deterministic Approximations

- *Laplace Approximation*: Approximates $p(\boldsymbol{\theta} \mid \mathbf{x})$ with a multivariate normal distribution centered at the mode of the posterior and having the negative inverse Hessian matrix of $\ln(p(\mathbf{x} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta}))$, evaluated at the mode, as its variance-covariance matrix.
- *Variational methods*: Approximates $p(\boldsymbol{\theta} \mid \mathbf{x})$ with the distribution $q(\boldsymbol{\theta}) \in \mathscr{D}$ that is closest to $p(\boldsymbol{\theta} \mid \mathbf{x})$ according to some objective function, among all the distributions in the family $\mathscr{D}$.

[1] As we will see, sometimes we draw from a proposal distribution, but then correct such samples to make them samples from our desired distribution, see Section 2.2.2.

1.3 Bayesian Logistic Regression

To conclude this chapter, we will consider a key example of Bayesian modelling and inference: Bayesian Logistic Regression.
Consider a Generalized Linear Model, as described in Appendix A, where each response variable follows independently[2] and identically a Bernoulli distribution

$$p(y_i \mid \mathbf{X}_i = \mathbf{x}_i; \pi_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i} \qquad \text{for } y_i \in \{0,1\}$$

and where the link function is given by the **logit** function $g : [0,1] \to \mathbb{R}$ given below.

$$g(\mathbb{E}[y_i \mid \mathbf{X}_i = \mathbf{x}_i]) = g(\pi_i) = \ln\left(\frac{\pi_i}{1-\pi_i}\right) = \mathbf{x}_i^\top \boldsymbol{\beta} \tag{1.9}$$

Equivalently, one can use (1.9) to write the success probability π_i in terms of the explanatory variables and of our parameters of interest $\boldsymbol{\beta}$:

$$\pi_i(\boldsymbol{\beta}) = \frac{\exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}{1+\exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)} \tag{1.10}$$

Leveraging the independence assumption we can write the joint distribution of the response variables (i.e. the likelihood) as a product of marginal probability mass functions

$$p(\mathbf{y} \mid \mathbf{X}_D^{\text{RV}} = \mathbf{X}_D^{\text{obs}}; \boldsymbol{\beta}) = \prod_{i=1}^{n} p(y_i; \mathbf{x}_i, \boldsymbol{\beta}) = \prod_{i=1}^{n} \pi_i(\boldsymbol{\beta})^{y_i}(1-\pi_i(\boldsymbol{\beta}))^{1-y_i} \tag{1.11}$$

For ease of notation we will simply refer to the above expression as $p(\mathbf{y} \mid \boldsymbol{\beta})$ by neglecting the conditioning on the explanatory variables, which will be considered fixed.[3]

Having found the likelihood, we just need to choose a prior that reflects our beliefs about the parameters $\boldsymbol{\beta}$. For simplicity, we will choose a multivariate normal distribution $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ where $\boldsymbol{\mu}_0 \in \mathbb{R}^p$ and $\boldsymbol{\Sigma}_0 \in \mathbb{R}^{p\times p}$, that is

$$p(\boldsymbol{\beta}) = (2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}_0|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)\right) \tag{1.12}$$

Plugging (1.11) and (1.12) into (1.2) we obtain an expression proportional to the posterior distribution

$$p(\boldsymbol{\beta} \mid \mathbf{y}) \propto (2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}_0|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)\right)\prod_{i=1}^{n} \pi_i(\boldsymbol{\beta})^{y_i}(1-\pi_i(\boldsymbol{\beta}))^{1-y_i} \tag{1.13}$$

[2] The term "independent" here means *statistically* independent in the traditional sense. When we say "dependent variable" we mean that we expect $\mathbf{y}$ to change due to a change in $\mathbf{X}_D$ following the pattern of dependence, which includes random and structural components.

[3] The notation $p(\mathbf{y}; \boldsymbol{\beta})$ means that $\boldsymbol{\beta}$ are parameters and if we consider $p(\mathbf{y}; \boldsymbol{\beta})$ as a function of them, then it is called *likelihood*. On the other hand $p(\mathbf{y} \mid \boldsymbol{\pi})$ indicates that we are treating $\boldsymbol{\pi}$ as a random variable. This is the preferred notation in Bayesian Inference.

We can simplify this expression by neglecting all those terms that do not depend on $\boldsymbol{\beta}$ and by taking the natural logarithm on both sides; this will also have the advantage to contain potential overflows in the Python implementation shown in Chapter 4.

$$\ln(p(\boldsymbol{\beta} \mid \mathbf{y})) \propto -\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0) + \sum_{i=1}^{n} y_i \ln(\pi_i(\boldsymbol{\beta})) + (1-y_i)\ln(1-\pi_i(\boldsymbol{\beta})) \tag{1.14}$$

This expression cannot be integrated analytically and, for large p, it would be computationally intractable even using standard quadrature techniques.

CHAPTER 2

STOCHASTIC APPROXIMATIONS

Markov Chain Monte Carlo (MCMC) methods are considered the gold standard for obtaining samples from an intractable posterior distribution which can be used to obtain summary statistics such as the posterior mean, or can be used to obtain non-parametric density estimates. In this chapter we will first explore methods that generate independent samples, but that are inefficient in high dimensions, and then we will construct the theory behind MCMC techniques that produce dependent samples but are much more robust in high dimensional settings.

2.1 Introduction to Sampling

Successful application of the theory of Monte Carlo and Markov Chain Monte Carlo methods (MCMC) relies on the assumption that we have a way of sampling a uniform distribution or, in other words, that we are able to generate random numbers within a finite interval. Due to the deterministic nature of our computers, we cannot generate *truly* random numbers, however there exists several methods that produce m numbers within $[0,1]$ such that, when compared through specific tests[1], they mimic the behavior of $V_1, \ldots, V_m \sim \mathcal{U}(0,1)$. Such numbers are called **pseudo random numbers** and, throughout this chapter, we will assume the existence of a generator of pseudo random numbers and consider its output as truly random.

Very often we are interested in obtaining samples from distributions other than the Uniform, for instance, the posterior distribution described in Section 1.1. Luckily, one can show [2] that every *continuous* random variable is a transformation of a uniform random variable.

[1] One can use the Kolmogorov-Smirnov test or time series analysis methods.

Theorem 2.1 (Probability Integral Transform)**.** *Let X be a continuous random variable with cumulative distribution function F. Let Y be a uniform random variable in the interval $[0,1]$. Then $F^{-1}(Y)$ has cdf F. Equivalently, let X be a defined as above. Then $U := F(X)$ follows a uniform distribution in the interval $[0,1]$.*

For instance, suppose that our random variable X follows an exponential distribution with rate parameter λ

$$f(x;\lambda) = \lambda e^{-\lambda x} \qquad ,\lambda > 0$$

then its cdf will be $F(x) = 1 - e^{-\lambda x}$, which is strictly increasing. We can therefore write down its inverse as $F^{-1}(y) = -\lambda \ln(1-y)$. According to the Probability Integral Transform theorem, if $Y \sim \mathcal{U}(0,1)$ then $F^{-1}(Y)$ should follow an exponential distribution with rate parameter λ. Figure 2.1 shows this in the case of $\lambda = 1$:

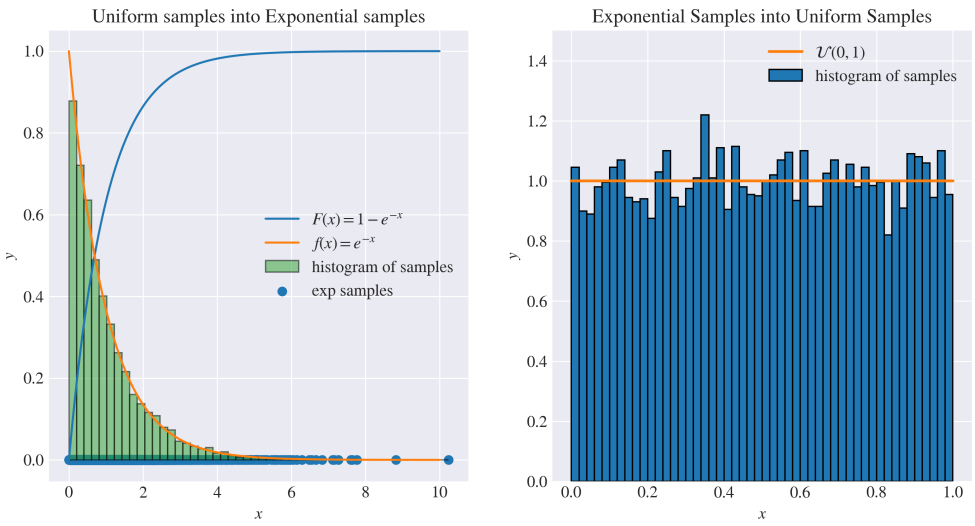

Figure 2.1: Simulation of Probability Integral Transform

2.2 Monte Carlo approximation

Suppose that we have obtained N *independent* samples $\boldsymbol{\theta}_1, \ldots \boldsymbol{\theta}_N$ from our target probability distribution $p(\boldsymbol{\theta} \mid \mathbf{x})$. Expectation (1.8) can then be thought of the *population* mean of $f(\boldsymbol{\theta})$, and it can be approximate by a sample average:

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{x})}\left[f(\boldsymbol{\theta})\right] \approx \overline{f}(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{\theta}_i) \tag{2.1}$$

Then, by the strong law of large numbers we know that $\overline{f}(\boldsymbol{\theta}) \xrightarrow{\text{a.s.}} \mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{x})}\left[f(\boldsymbol{\theta})\right]$ as $N \to \infty$.

2.2.1 Uniform Sampling

In equation (2.1) the samples $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$ are assumed to be drawn directly from $p(\boldsymbol{\theta} \mid \mathbf{x})$, which guarantees that, if N is large enough, their (normalized) histogram will approximate the probability density. In other words, samples are drawn proportionally to $p(\boldsymbol{\theta} \mid \mathbf{x})$.
A different way of obtaining the same estimate as in (2.1) is to draw $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$ *uniformly* in Θ, provided Θ is a bounded space, evaluate the density $p(\boldsymbol{\theta} \mid \mathbf{x})$ at those samples $p(\boldsymbol{\theta}_1 \mid \mathbf{x}), \ldots, p(\boldsymbol{\theta}_N \mid \mathbf{x})$ and use them as weights in a weighted average of $f(\boldsymbol{\theta}_1), \ldots, f(\boldsymbol{\theta}_N)$:

$$\overline{f}(\boldsymbol{\theta}) := \sum_{i=1}^{N} f(\boldsymbol{\theta}_i)\, p(\boldsymbol{\theta}_i \mid \mathbf{x}) \tag{2.2}$$

For most densities, however, the majority of the probability distribution mass will be concentrated in a relatively small area of Θ called **typical set** [14]. It follows that if the samples $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$ are drawn uniformly in Θ, then most of them will be drawn from regions where the mass of the probability distribution is practically zero, which means that a very large number of samples will have a negligible contribution to the sum (2.2).

As an example, suppose that we are interested in solving problem (1.3) the one-dimensional case of a standard normal distribution. Namely, we know that our posterior distribution is proportional to a standard normal distribution

$$p(\theta \mid x) \propto e^{-\frac{x^2}{2}} \qquad \text{or equivalently} \qquad p(x \mid \theta)\, p(\theta) = e^{-\frac{x^2}{2}}$$

and so we can evaluate $p(\theta \mid x)$ for any $\theta \in \mathbb{R}$ up to a proportionality constant (which, in this example, is $\sqrt{2\pi} \approx 2.51$).

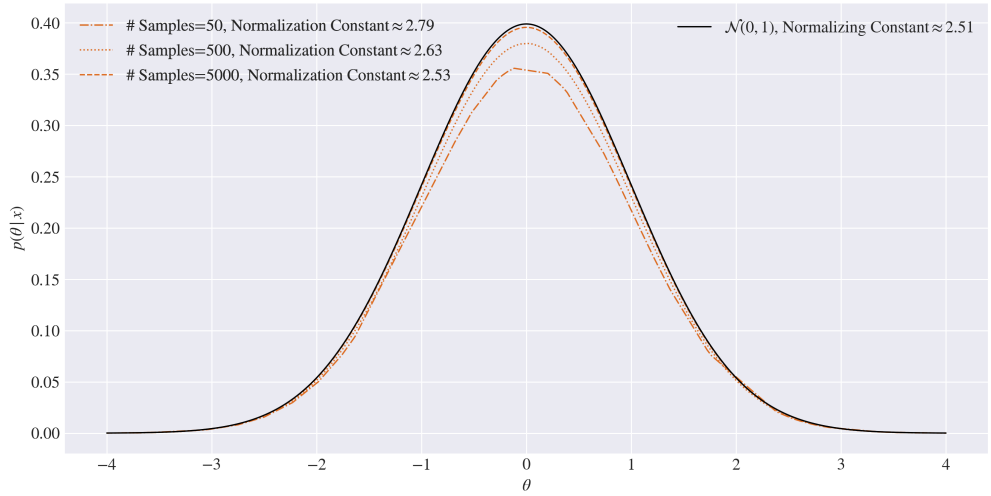

Figure 2.2: Estimation of a Normalization Constant via Uniform Sampling

Figure 2.2 shows how, even by restricting the sample space to $[-4,4]$ rather than the whole of $\mathbb{R}$, uniform sampling requires a large number of samples to obtain a decent approximation for the normalization constant. In practice, we might not be able to restrict ourselves to such a narrow region of Θ and as the dimensionality grows we can expect the number of samples needed to increase drastically.

2.2.2 Importance Sampling

Rather than drawing our samples $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N$ uniformly, we would like to have a mechanism to draw them from regions of Θ where we are confident that the density $p(\boldsymbol{\theta} \mid \mathbf{x})$ has a non-negligible mass, to make the algorithm more efficient. One way in which we can do this, is to draw $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N$ from a non-uniform distribution $q(\boldsymbol{\theta})$ that has a support Θ_q "similar " to Θ and then adjust such samples based on how different $q(\boldsymbol{\theta}_i)$ is to $p(\boldsymbol{\theta}_i \mid \mathbf{x})$. Consequently, we aim to find a **proposal distribution** $q(\boldsymbol{\theta})$ that is:

- "Biased" towards areas of Θ where $p(\boldsymbol{\theta} \mid \mathbf{x})$ has more mass, and $q(\boldsymbol{\theta}) > 0$ whenever $p(\boldsymbol{\theta} \mid \mathbf{x}) > 0$.
- Easy to sample from, and such that we can compute $q(\boldsymbol{\theta})$ for every $\boldsymbol{\theta} \in \Theta$ up to a proportionality constant, i.e. we can compute $\tilde{q}(\boldsymbol{\theta}) = \mathscr{C}_q q(\boldsymbol{\theta})$ for $\mathscr{C}_q := \int_\Theta \tilde{q}(\boldsymbol{\theta}) d\boldsymbol{\theta}$.

In the following we show that even in the case in which we can only evaluate $p(\boldsymbol{\theta} \mid \mathbf{x})$ up to a proportionality constant[2], the mechanism described above correctly approximates equation (1.8).

$$
\begin{aligned}
\mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{x})}\left[f(\boldsymbol{\theta})\right] &= \int_\Theta p(\boldsymbol{\theta} \mid \mathbf{x}) f(\boldsymbol{\theta}) d\boldsymbol{\theta} && \text{LOTUS}^3 \\
&= \int_\Theta q(\boldsymbol{\theta}) \frac{p(\boldsymbol{\theta} \mid \mathbf{x})}{q(\boldsymbol{\theta})} f(\boldsymbol{\theta}) d\boldsymbol{\theta} && q(\boldsymbol{\theta}) \text{ is the proposal. Assume } q(\boldsymbol{\theta}) > 0 \text{ for } \boldsymbol{\theta} \in \Theta \\
&= \frac{\mathscr{C}_q}{p(\mathbf{x})} \int_\Theta q(\boldsymbol{\theta}) \frac{p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\tilde{q}(\boldsymbol{\theta})} f(\boldsymbol{\theta}) d\boldsymbol{\theta} && \\
&= \frac{\mathscr{C}_q}{p(\mathbf{x})} \mathbb{E}_{q(\boldsymbol{\theta})}\left[\frac{p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\tilde{q}(\boldsymbol{\theta})} f(\boldsymbol{\theta})\right] && \text{LOTUS} \\
&\approx \frac{\mathscr{C}_q}{p(\mathbf{x})} \frac{1}{N} \sum_{i=1}^{N} \frac{p(\mathbf{x} \mid \boldsymbol{\theta}_i) p(\boldsymbol{\theta}_i)}{\tilde{q}(\boldsymbol{\theta}_i)} f(\boldsymbol{\theta}_i) && \text{where } \boldsymbol{\theta}_i \sim q(\boldsymbol{\theta})
\end{aligned} \tag{2.3}
$$

In the last step we have used Monte Carlo approximation (2.1).
Our job is not done because the ratio of normalization constants is unknown. Luckily, we can estimate

[2] The proportionality constant is given by (1.3) as $p(\mathbf{x}) = \int_\Theta p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$.
[3] Law of the Unconscious Statistician. See theorem 4.1.1 in DeGroot and Schervish [8].

the inverse of such ratio by following the work of Bishop [3]

$$\begin{aligned}
\frac{p(\mathbf{x})}{\mathscr{C}_q} &= \frac{1}{\mathscr{C}_q}\int_\Theta p(\mathbf{x}\mid\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} \\
&= \frac{1}{\mathscr{C}_q}\int_\Theta \frac{p(\mathbf{x}\mid\boldsymbol{\theta})p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}q(\boldsymbol{\theta})d\boldsymbol{\theta} \\
&= \int_\Theta \frac{p(\mathbf{x}\mid\boldsymbol{\theta})p(\boldsymbol{\theta})}{\widetilde{q}(\boldsymbol{\theta})}q(\boldsymbol{\theta})d\boldsymbol{\theta} \\
&= \mathbb{E}_{q(\boldsymbol{\theta})}\left[\frac{p(\mathbf{x}\mid\boldsymbol{\theta})p(\boldsymbol{\theta})}{\widetilde{q}(\boldsymbol{\theta})}\right] && \text{LOTUS} \\
&\approx \frac{1}{N}\sum_{i=1}^{N}\frac{p(\mathbf{x}\mid\boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i)}{\widetilde{q}(\boldsymbol{\theta}_i)} && \text{Monte Carlo approximation}
\end{aligned} \tag{2.4}$$

Now plugging (2.4) into (2.3) having defined

$$\widetilde{r}_i := \frac{p(\mathbf{x}\mid\boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i)}{\widetilde{q}(\boldsymbol{\theta}_i)} \qquad \forall i \in \{1,\dots,N\} \tag{2.5}$$

for convenience of notation, we obtain the following estimate:

$$\begin{aligned}
\mathbb{E}_{p(\boldsymbol{\theta}\mid\mathbf{x})}[f(\boldsymbol{\theta})] &\approx \left(\frac{1}{N}\sum_{j=1}^{N}\widetilde{r}_j\right)^{-1}\frac{1}{N}\sum_{i=1}^{N}\widetilde{r}_i f(\boldsymbol{\theta}_i) \\
&= \sum_{i=1}^{N}\frac{\widetilde{r}_i}{\frac{1}{N}\sum_{j=1}^{N}\widetilde{r}_j}f(\boldsymbol{\theta}_i) \\
&:= \sum_{i=1}^{N} w_i f(\boldsymbol{\theta}_i) && \text{define } w_i := \frac{\widetilde{r}_i}{\frac{1}{N}\sum_{j=1}^{N}\widetilde{r}_j}
\end{aligned} \tag{2.6}$$

Where w_i are called **importance weights** and intuitively they compensate the error introduced by sampling from the distribution $q(\boldsymbol{\theta})$ rather than $p(\boldsymbol{\theta}\mid\mathbf{x})$. The introduction of importance weights should make it clear that importance sampling could perform badly if the proposal distribution $q(\boldsymbol{\theta})$ is close to zero in regions of the support Θ where $p(\boldsymbol{\theta}\mid\mathbf{x})$ is large.
Unfortunately, this tends to happen when we deal with high dimensional random variables, making Importance Sampling suffer from the *Curse of Dimensionality*, even though to a much lesser extent than uniform sampling. Indeed, one could think of Uniform sampling as begin a specific case of Importance sampling where the proposal distribution is just a uniform density[4].

2.2.3 Rejection Sampling

Both Uniform and Importance sampling are methods that allow us to approximates expectations such as (1.8), but they are not methods that provide us with samples $\boldsymbol{\theta}_1,\dots,\boldsymbol{\theta}_N$ from $p(\boldsymbol{\theta}\mid\mathbf{x})$ themselves. In theory, one could find the inverse cdf of $p(\boldsymbol{\theta}\mid\mathbf{x})$, apply theorem 2.1 and use a Monte Carlo approximation. However, this is often impractical, especially when we are working with complex

[4] As discussed in Section 2.2.1, one should restrict Θ to a bounded space.

distributions, either because we might not have an explicit expression for F^{-1}, or because we can find a much more efficient method of sampling, such as Rejection Sampling. It is particularly suited when:

1. For any given $\boldsymbol{\theta}$ in the support Θ, it is straightforward to compute $p(\boldsymbol{\theta} \mid \mathbf{x})$ up to some normalizing constant, i.e. we can calculate $p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$ but not $p(\boldsymbol{\theta} \mid \mathbf{x})$.

2. It is difficult and computationally expensive to sample from $p(\boldsymbol{\theta} \mid \mathbf{x})$.

The intuition behind Rejection Sampling is that we want to choose a distribution that it is easy to sample from, as in Importance sampling, and we want to rescale the density so that $p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$ is underneath it. Then, we want to draw samples from this simpler distribution and have a criteria that allows us to "accept" some of those samples as coming from our target distribution $p(\boldsymbol{\theta} \mid \mathbf{x})$.

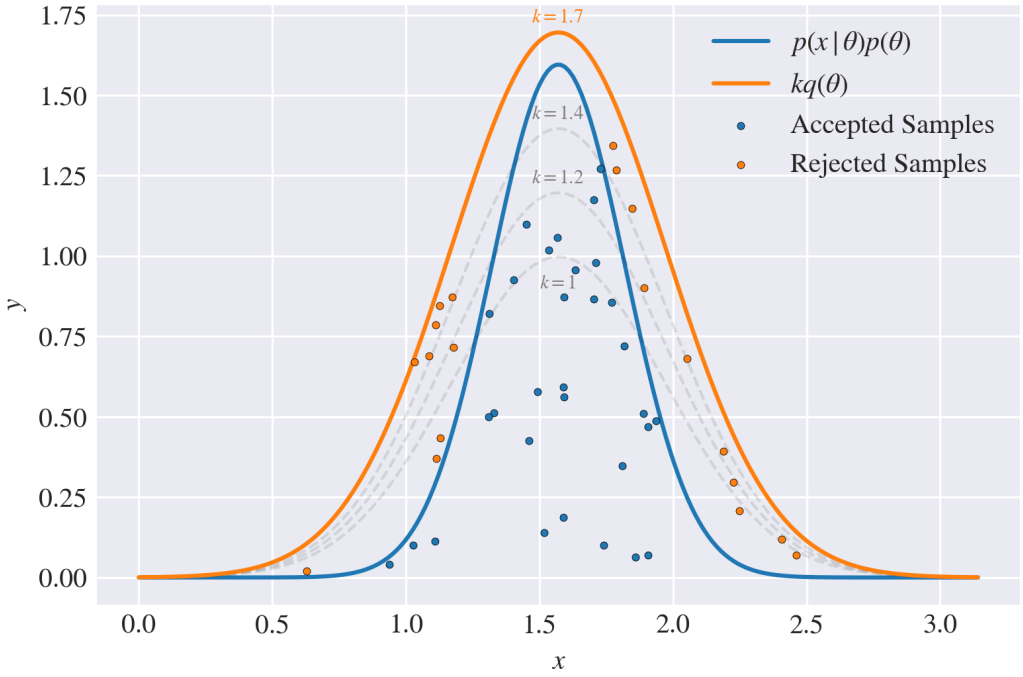

Figure 2.3: Example of a proposal distribution and its comparison function

More formally, let us choose a **proposal distribution** $q(\boldsymbol{\theta})$ that it is easy to sample from, i.e. we can effortlessly generate $\boldsymbol{\theta}_i \sim q(\boldsymbol{\theta})$, and that has compatible support with Θ. Next, we want to rescale this distribution so that it is always above our unnormalized target distribution $p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$. To do this, we look for a value $k \in \mathbb{R}$ such that

$$p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta}) \leq kq(\boldsymbol{\theta}) \qquad \forall \boldsymbol{\theta} \in \Theta \qquad k > 1 \tag{2.7}$$

From Equation (2.7) and Figure 2.3 it should be clear that $kq(\boldsymbol{\theta})$ is not a probability distribution as it does not integrate to one, therefore it is often called **comparison function**.
The Rejection Sampling algorithm to generate samples $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$ from $p(\boldsymbol{\theta} \mid \mathbf{x})$ then reads as follows:

Algorithm 1: Rejection Sampling

for $i = 1, 2, \ldots, N$ **do**:
    Draw a sample from the proposal distribution $\boldsymbol{\theta}_i \sim q(\boldsymbol{\theta})$.
    Draw a uniform sample *underneath* the comparison function $u_i \mid \boldsymbol{\theta}_i \sim \mathcal{U}(0, kq(\boldsymbol{\theta}_i))$.
    if $u_i > p(\mathbf{x} \mid \boldsymbol{\theta}_i) p(\boldsymbol{\theta}_i)$ **do**:
        Reject $\boldsymbol{\theta}_i$.
    else:
        Accept $\boldsymbol{\theta}_i$ as an *independent* sample from $p(\boldsymbol{\theta} \mid \mathbf{x})$.
    end
end

Intuitively, we are generating *uniform* samples under the *graph* of the comparison function,

$$(\boldsymbol{\theta}_i, u_i) \sim \mathcal{U}(\{(\boldsymbol{\theta}, u) : 0 \leq u \leq kq(\boldsymbol{\theta})\})$$

and we are accepting them if they are *also* under the graph of $p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$, as shown in Figure 2.3. To see this, denote $p(\boldsymbol{\theta}, u \mid \mathbf{x})$ the joint density. Then

$$p(\boldsymbol{\theta}, u \mid \mathbf{x}) = \begin{cases} p(u \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathbf{x}) = \mathcal{U}(0, kq(\boldsymbol{\theta})) \times q(\boldsymbol{\theta}) = \frac{1}{kq(\boldsymbol{\theta})} q(\boldsymbol{\theta}) = \frac{1}{k} & \text{if } (\boldsymbol{\theta} \;\; U)^\top \in \{(\boldsymbol{\theta}, u) : 0 \leq u \leq kq(\boldsymbol{\theta})\} \\ 0 & \text{otherwise} \end{cases}$$

which proves the assertion above. A more detailed proof of this statement can be found in [7].
The **acceptance probability** can naturally be found by comparing the area of the unnormalized target distribution $p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$ and the comparison function $kq(\boldsymbol{\theta})$:

$$\mathbb{P}(\text{accept}) := \int \frac{p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{kq(\boldsymbol{\theta})} q(\boldsymbol{\theta}) d\boldsymbol{\theta} = \frac{p(\mathbf{x})}{k} = \frac{\text{area under } p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\text{area under } kq(\boldsymbol{\theta})} \tag{2.8}$$

Following the analysis of Devroye [9] one can model the number M of trials leading to a successful sample $\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta} \mid \mathbf{x})$ as a Geometric distribution with probability equal to $\frac{p(\mathbf{x})}{k}$ and with support $\{1, 2, \ldots\}$

$$\mathbb{P}(M = j) = \underbrace{\left(1 - \frac{p(\mathbf{x})}{k}\right)^{j-1}}_{j-1 \text{ rejections}} \times \overbrace{\frac{p(\mathbf{x})}{k}}^{1 \text{ acceptance}} \qquad \text{for } j \in \{1, 2, \ldots\}$$

From this it is straightforward to find the expected number of draws leading to a successful sample

$$\mathbb{E}_{M \sim \text{Geom}\left(\frac{p(\mathbf{x})}{k}\right)}[M] = \frac{1}{\frac{p(\mathbf{x})}{k}} = \frac{k}{p(\mathbf{x})} \tag{2.9}$$

Equations (2.8) and (2.9) show that to minimize wasteful calculations we need to choose k to be as small as possible, i.e. we require the comparison function to envelope the unnormalized target

distribution as tightly as possible.
It is important to observe that throughout the algorithm we work with $p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$ only, and yet we claim to obtain samples from $p(\boldsymbol{\theta} \mid \mathbf{x})$. The motivation is simple enough: the samples generated $\mathbf{x}_i$ have density proportional to the unnormalized target distribution $p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$, but since $p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta}) = p(\mathbf{x})p(\boldsymbol{\theta} \mid \mathbf{x})$, the samples $\boldsymbol{\theta}_i$ are also proportional to the normalized target distribution $p(\boldsymbol{\theta} \mid \mathbf{x})$ as the normalization constant $p(\mathbf{x})$ is common to all $\boldsymbol{\theta} \in \Theta$ [9].

2.3 Markov Chains Monte Carlo

Uniform, Importance and Rejection sampling generate samples from the target distribution $p(\boldsymbol{\theta} \mid \mathbf{x})$ that are *independent*. Markov Chain Monte Carlo (MCMC) methods, instead, generate dependent samples that are only asymptotically distributed according to $p(\boldsymbol{\theta} \mid \mathbf{x})$. Put differently, MCMC trades off the *costly* independence requirement and exactness of the samples, for less informative, approximate, dependent samples that are faster to compute.
Essentially, MCMC uses Monte Carlo approximation on samples that have been generated via a Markov Chain exploring Θ, and whose equilibrium distribution is $p(\boldsymbol{\theta} \mid \mathbf{x})$. This guarantees that, provided we run our chain for a long enough time, the samples $\{\boldsymbol{\theta}_t : t \in \mathbb{N}\}$ will be distributed as $p(\boldsymbol{\theta} \mid \mathbf{x})$.
A Markov Chain is simply a sequence of random variables $\{\boldsymbol{\theta}_t : t \in \mathbb{N}\}$ taking values in Θ and for which the **Markov property** holds: the value taken by the random variable $\boldsymbol{\theta}_{t+1}$ depends only on the value taken by the previous random variable $\boldsymbol{\theta}_t$, and not on the past history $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{t-1}$. Suppose that we have constructed the chain in a way in which if the initial random variable $\boldsymbol{\theta}_0$ has a carefully-selected density $r(\boldsymbol{\theta})$ then all the following random variables $\boldsymbol{\theta}_t$ for $t > 0$ also have $r(\boldsymbol{\theta})$ as their *marginal* distribution. We call such a distribution the **invariant distribution** of the chain.
Under suitable conditions, discussed at length in [16, 3, 19, 10, 1], one can show that starting with the random variable $\boldsymbol{\theta}_0$ taking any value in Θ the limiting distribution of the random variables $\{\boldsymbol{\theta}_t : t \in \mathbb{N}\}$ tends to the invariant distribution as $t \to \infty$.
Evidently, the key to MCMC is having a mechanism to construct a chain satisfying the regularity conditions and whose invariant distribution is the target posterior $p(\boldsymbol{\theta} \mid \mathbf{x})$, so that for $t \in \mathbb{N}$ large enough, we can consider $\boldsymbol{\theta}_t \sim p(\boldsymbol{\theta} \mid \mathbf{x})$.
A sufficient condition for $p(\boldsymbol{\theta} \mid \mathbf{x})$ to be the invariant distribution of the Markov chain is to design a **transition density** $t(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$ satisfying the **detailed balance** equation.

$$p(\boldsymbol{\theta}' \mid \mathbf{x})\, t(\boldsymbol{\theta} \mid \boldsymbol{\theta}') = p(\boldsymbol{\theta} \mid \mathbf{x})\, t(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) \tag{2.10}$$

The transition density $t(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$ describes the probability of the next random variable $\boldsymbol{\theta}'$ taking a certain value[5], given the value of the current random variable $\boldsymbol{\theta}$. It follows that a Markov Chain

[5] Since we are working with continuous random variables we should rather consider the probability of $\boldsymbol{\theta}'$ taking value in a (measurable) subset of Θ. For this reason one should define a transition kernel [16]

satisfying the detailed balance is **reversible** as the subsequence $(\boldsymbol{\theta}, \boldsymbol{\theta}')$ has the same probability as the subsequence $(\boldsymbol{\theta}', \boldsymbol{\theta})$.

2.3.1 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm [15, 12, 18] requires the experimenter to choose a starting value for the random variable $\boldsymbol{\theta}_0$ and a **proposal distribution** $q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}_t)$. A candidate for the value of $\boldsymbol{\theta}_1$ is then sampled from the proposal $\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}_0)$ and it is either accepted, with a certain probability, as being the value of $\boldsymbol{\theta}_1$ or it is rejected, in which case we assign the same value taken by $\boldsymbol{\theta}_0$ to $\boldsymbol{\theta}_1$. This process is repeated a sufficiently large number of times to exploit the asymptotic properties discussed previously. The algorithm to generate N Metropolis-Hastings samples works as follows.

Algorithm 2: Metropolis-Hastings

Choose a starting value taken by $\boldsymbol{\theta}_0$ in Θ. Choose a proposal distribution $q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}_i)$.
for $i = 1, 2, \ldots, N$ **do**:
    Draw a sample from the proposal distribution $\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}_i)$.
    Compute the acceptance probability.

$$\alpha\left(\boldsymbol{\theta}^*, \boldsymbol{\theta}_i\right) := \min\left\{1, \frac{p(\boldsymbol{\theta}^* \mid \mathbf{x}) q(\boldsymbol{\theta}_i \mid \boldsymbol{\theta}^*)}{p(\boldsymbol{\theta}_i \mid \mathbf{x}) q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}_i)}\right\} \tag{2.11}$$

    Draw a uniform random sample $u_i \sim \mathcal{U}(0,1)$.
    if $u_i \leq \alpha\left(\boldsymbol{\theta}^*, \boldsymbol{\theta}_i\right)$ **do**:
        Accept $\boldsymbol{\theta}^*$ as the value taken by $\boldsymbol{\theta}_{i+1}$.
    else:
        Reject $\boldsymbol{\theta}^*$ and use the value of $\boldsymbol{\theta}_i$ as the realization of $\boldsymbol{\theta}_{i+1}$.
    end
end

One can easily show that the way in which we have constructed the acceptance probability guarantees that the detailed balance in (2.10) is satisfied, and therefore the posterior distribution is the limiting distribution of the chain. Also, the normalization constants for both the target and the proposal distributions cancel out when calculating the acceptance probability. This means that we can still generate the required samples from $p(\boldsymbol{\theta} \mid \mathbf{x})$ even if we can only obtain samples up to a proportionality constant.

2.3.2 Random-Walk Metropolis-Hastings algorithm

The Random-Walk Metropolis-Hastings algorithm (RWMH) is a special case of Algorithm 2 where we have chosen a *symmetric* proposal distribution $q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}_i) = q(\boldsymbol{\theta}_i \mid \boldsymbol{\theta}^*)$. In this case the acceptance probability becomes

$$\alpha\left(\boldsymbol{\theta}^*, \boldsymbol{\theta}_i\right) = \min\left\{1, \frac{p(\boldsymbol{\theta}^* \mid \mathbf{x})}{p(\boldsymbol{\theta}_i \mid \mathbf{x})}\right\} \tag{2.12}$$

A very common choice for the proposal distribution is a normal distribution centered at the value of the previous random variable $\boldsymbol{\theta}_i$. The experimenter then should take care in choosing the variance[6] of the proposal σ_q^2, as analyzed by O'Hagan and Forster [16] because a value that is too small will lead to candidates $\boldsymbol{\theta}^*$ that are very close to $\boldsymbol{\theta}_i$ and therefore the ratio $p(\boldsymbol{\theta}^* \mid \mathbf{x}) \big/ p(\boldsymbol{\theta}_i \mid \mathbf{x})$ will be nearly 1. The consequence of this is that we will almost always accept samples, however this will slow down our exploration of the parameter space Θ. On the other hand, if the variance is too large then we will be suggesting values for $\boldsymbol{\theta}_{i+1}$ that are spread out across Θ. However, we know from Section 2.2.1 that most of the mass of the density $p(\boldsymbol{\theta} \mid \mathbf{x})$ is concentrated in the typical set and therefore we will be suggesting values where $p(\boldsymbol{\theta}^* \mid \mathbf{x})$ has very low mass, leading to many rejections.
Helpfully, Roberts et al. [20] showed that in a p dimensional parameter space Θ one can choose the following variance-covariance matrix to obtain a sensible acceptance rate that allows the Chain to explore Θ without getting stuck or slowing down too much, provided the posterior dependence is not too great.

$$\sigma_p^2 I := \frac{2.38^2}{p} \times \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \tag{2.13}$$

[6] Or the variance-covariance matrix in more than one dimension.

CHAPTER 3

DETERMINISTIC APPROXIMATIONS

Deterministic approximation methods aim to find a distribution that resembles the posterior distribution, rather than drawing samples from it. One of the main advantage of these methods over stochastic approximation methods seen in the previous chapter, is that we do not have to wait for the Markov Chain to converge to the equilibrium distribution, and for this reason deterministic approximation are more well-suited for large-scale problems. The drawback is that we will only be working with approximations and not with the exact posterior distribution, so we will not have any convergence guarantee.

3.1 Laplace Approximation

3.1.1 A General Framework

The idea of Laplace approximation is to approximate $p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$ by a multivariate normal density centered at the mode. From basic results in optimization we know that the mode $\boldsymbol{\theta}_0 \in \Theta$ will satisfy:

- $\nabla_{\boldsymbol{\theta}} p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} = 0$ (i.e. it will be a stationary point)
- $\nabla^2 p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \prec 0$ (i.e. it will be a maximum)

As usual, since the natural logarithm is a strictly increasing function, $\boldsymbol{\theta}_0$ will also be the maximum of $\ln(p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta}))$, which we can Taylor expand to find a second order approximation, using the first order optimality condition above:

$$\ln(p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})) \approx \ln(p(\mathbf{x} \mid \boldsymbol{\theta}_0)p(\boldsymbol{\theta}_0)) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \nabla^2 \ln(p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta}))\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} (\boldsymbol{\theta} - \boldsymbol{\theta}_0)$$

Exponentiating both sides leads to

$$p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta}) \approx p(\mathbf{x} \mid \boldsymbol{\theta}_0)p(\boldsymbol{\theta}_0) \exp\left\{\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \nabla^2 \ln(p(\mathbf{x} \mid \boldsymbol{\theta})p(\boldsymbol{\theta}))\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} (\boldsymbol{\theta} - \boldsymbol{\theta}_0)\right\}$$

Comparing the expression above with (A.5) we notice that the exponential term can be seen as the exponential part of a multivariate normal distribution with

$$\boldsymbol{\mu} = \boldsymbol{\theta}_0 \in \mathbb{R}^p \qquad \text{and} \qquad \boldsymbol{\Sigma}^{-1} = -\nabla^2 \ln(p(\mathbf{x} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta}))\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \in \mathbb{R}^{p\times p}$$

With this in mind, we can define our Gaussian approximation to be

$$q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_0, -H(\boldsymbol{\theta}_0)^{-1}) \tag{3.1}$$

where $H(\boldsymbol{\theta}_0)$ is the Hessian matrix of $\ln\big(p(\mathbf{x} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\big)$ at the mode $\boldsymbol{\theta}_0$. Simply put, Laplace method approximates $p(\mathbf{x} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})$ with a multivariate normal centered at the mode whose variance-covariance matrix is the observed information matrix evaluated at the mode, provided the prior is not too influential.
Notice that an important drawback of this method is that it bases its approximation solely on the behavior of $p(\mathbf{x} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})$ at a single point $\boldsymbol{\theta}_0$.

3.1.2 Application to Bayesian Logistic Regression

We can apply result (3.1) to the logistic regression log-posterior that was found in Section 1.3 to be

$$\ln(p(\boldsymbol{\beta} \mid \mathbf{y})) \propto -\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0) + \sum_{i=1}^{n} y_i \ln(\pi_i(\boldsymbol{\beta})) + (1-y_i)\ln(1-\pi_i(\boldsymbol{\beta}))$$

First, let's find the variance-covariance matrix. The derivative of the first term of the log-likelihood can be found as follows:

$$\begin{aligned}
\nabla_{\boldsymbol{\beta}}\left(-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)\right) &= -\frac{1}{2}\nabla_{\boldsymbol{\beta}}\left(\boldsymbol{\beta}^\top \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\beta} - \boldsymbol{\beta}^\top \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_0 \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\beta} + \boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right) \\
&= -\frac{1}{2}\left(2\boldsymbol{\beta}^\top \boldsymbol{\Sigma}_0^{-1} - \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right) \\
&= -\boldsymbol{\beta}^\top \boldsymbol{\Sigma}_0^{-1} + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0
\end{aligned}$$

Differentiating again with respect to $\boldsymbol{\beta}$ and multiplying by -1 yields:

$$-\nabla^2_{\boldsymbol{\beta}}\left(-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)\right) = \boldsymbol{\Sigma}_0^{-1} \tag{3.2}$$

In order to differentiate the second term with respect to $\boldsymbol{\beta}$, we need to use the following useful fact about the Sigmoid function:

$$\frac{d\sigma(a)}{da} = \sigma(a)(1-\sigma(a)) \qquad \text{where} \qquad \sigma(a) := \frac{\exp(a)}{1+\exp(a)} \tag{3.3}$$

It then follows from (1.10) that

$$\nabla_{\boldsymbol{\beta}}\pi_i(\boldsymbol{\beta}) = \nabla_{\boldsymbol{\beta}}\sigma(\mathbf{x}_i^\top \boldsymbol{\beta}) = \sigma(\mathbf{x}_i^\top \boldsymbol{\beta})(1-\sigma(\mathbf{x}_i^\top \boldsymbol{\beta}))\nabla_{\boldsymbol{\beta}}(\mathbf{x}_i^\top \boldsymbol{\beta}) = \pi_i(\boldsymbol{\beta})(1-\pi_i(\boldsymbol{\beta}))\mathbf{x}_i$$

We can then use this result to find the derivative of the second term of the log-posterior.

$$\begin{aligned}\nabla_{\boldsymbol{\beta}}\left(\sum_{i=1}^{n} y_i \ln(\pi_i(\boldsymbol{\beta})) + (1-y_i)\ln(1-\pi_i(\boldsymbol{\beta}))\right) &= \sum_{i=1}^{n} \nabla_{\boldsymbol{\beta}}\left(y_i \ln(\pi_i(\boldsymbol{\beta})) + (1-y_i)\ln(1-\pi_i(\boldsymbol{\beta}))\right) \\ &= \sum_{i=1}^{n}\left[\frac{y_i}{\pi_i(\boldsymbol{\beta})}\nabla_{\boldsymbol{\beta}}\pi_i(\boldsymbol{\beta}) - \frac{1-y_i}{1-\pi_i(\boldsymbol{\beta})}\nabla_{\boldsymbol{\beta}}\pi_i(\boldsymbol{\beta})\right] \\ &= \sum_{i=1}^{n}\left[y_i(1-\pi_i(\boldsymbol{\beta}))\mathbf{x}_i - (1-y_i)\pi_i(\boldsymbol{\beta})\mathbf{x}_i\right] \\ &= \sum_{i=1}^{n}\left(y_i - \pi_i(\boldsymbol{\beta})\right)\mathbf{x}_i\end{aligned}$$

Differentiating again with respect to $\boldsymbol{\beta}$ and multiplying by -1 yields

$$-\nabla^2_{\boldsymbol{\beta}}\left(\sum_{i=1}^{n} y_i \ln(\pi_i(\boldsymbol{\beta})) + (1-y_i)\ln(1-\pi_i(\boldsymbol{\beta}))\right) = \sum_{i=1}^{n}\pi_i(\boldsymbol{\beta})(1-\pi_i(\boldsymbol{\beta}))\mathbf{x}_i\mathbf{x}_i^\top \tag{3.4}$$

Combining (3.2) and (3.4) gives an expression for observed Fisher information matrix

$$-H(\boldsymbol{\beta}_0) = \boldsymbol{\Sigma}_0^{-1} + \sum_{i=1}^{n}\pi_i(\boldsymbol{\beta}_0)(1-\pi_i(\boldsymbol{\beta}_0))\mathbf{x}_i\mathbf{x}_i^\top$$

where $\boldsymbol{\beta}_0$ is the mode of the posterior, which in practice is often found using an optimization routine. The multivariate normal Laplace approximation to the posterior becomes:

$$q(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta} \mid \boldsymbol{\beta}_0, -H(\boldsymbol{\beta}_0)^{-1}) \tag{3.5}$$

3.2 Variational Methods

3.2.1 A General Framework

As noted at the end of Section 3.1.1, Laplace approximation can look simplistic because it approximates the posterior using only information at the mode of the posterior distribution. On the other hand, Variational Inference aims to use more information to approximate $p(\boldsymbol{\theta} \mid \mathbf{x})p(\boldsymbol{\theta})$ with a similar distribution.

There are infinitely many distributions that we can choose from and considering all of them would render Variational Inference inefficient and ultimately useless. To overcome this issue, one has to specify a family of distributions $\mathscr{D}$ that is flexible enough to allow distributions very similar to $p(\boldsymbol{\theta} \mid \mathbf{x})p(\boldsymbol{\theta})$ while being restrictive enough so that we don't have to consider every possible probability distribution available. Once the family of distributions $\mathscr{D}$ has been defined, the next step is to choose the distribution $q(\boldsymbol{\theta}) \in \mathscr{D}$ that is closest to our target distribution, according to some objective function. Such an objective function should be able to compare how similar two distributions are, and a good candidate for comparing two continuous[1] distributions $q(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta} \mid \mathbf{x})$ is the

[1] Notice that the KL-divergence can equally be defined for discrete distributions by replacing integrals with summations.

Kullback-Leiber divergence, or KL-divergence, described in Appendix A.4.

$$\mathrm{KL}\big(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})\big) := \mathbb{E}_{q(\boldsymbol{\theta})}\left[\ln\left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \mathbf{x})}\right)\right] = \int_{\Theta} q(\boldsymbol{\theta}) \ln\left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \mathbf{x})}\right) d\boldsymbol{\theta} \tag{3.6}$$

It is important to notice a few properties of this objective function:

- It is *not* symmetric $\mathrm{KL}\big(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})\big) \neq \mathrm{KL}\big(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})\big)$.
- It is non-negative $\mathrm{KL}\big(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})\big) \geq 0$.
- It is minimized when the two distributions are equal, i.e. $\mathrm{KL}\big(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})\big) = 0$ if and only if $q(\boldsymbol{\theta}) \overset{\text{a.e.}}{=} p(\boldsymbol{\theta} \mid \mathbf{x})$

With this in mind, we can concisely rewrite the variational approach as

$$q^*(\boldsymbol{\theta}) = \underset{q(\boldsymbol{\theta}) \in \mathscr{D}}{\arg\min}\, \mathrm{KL}\big(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})\big)$$

Unfortunately, using the KL-divergence is often impractical because it would require knowing the normalization constant for $p(\boldsymbol{\theta} \mid \mathbf{x})$, which is likely to be unknown. Indeed one can rewrite (3.6) as:

$$\begin{aligned}
\mathrm{KL}\big(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})\big) &:= \mathbb{E}_{q(\boldsymbol{\theta})}\left[\ln\left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \mathbf{x})}\right)\right] \\
&= \mathbb{E}_{q(\boldsymbol{\theta})}\left[\ln\big(q(\boldsymbol{\theta})\big) - \ln\left(\frac{p(\mathbf{x} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathbf{x})}\right)\right] \\
&= \mathbb{E}_{q(\boldsymbol{\theta})}\big[\ln\big(q(\boldsymbol{\theta})\big)\big] - \mathbb{E}_{q(\boldsymbol{\theta})}\big[\ln\big(p(\mathbf{x} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\big)\big] + \mathbb{E}_{q(\boldsymbol{\theta})}\big[\ln\big(p(\mathbf{x})\big)\big] \\
&= \mathbb{E}_{q(\boldsymbol{\theta})}\big[\ln\big(q(\boldsymbol{\theta})\big)\big] - \mathbb{E}_{q(\boldsymbol{\theta})}\big[\ln\big(p(\mathbf{x} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\big)\big] + p(\mathbf{x})
\end{aligned}$$

A simple remedy to this, is to "shift" the KL-divergence to not depend on the model evidence $p(\mathbf{x})$ anymore. Thus, we define a new objective function called **Evidence Lower Bound** (ELBO), which takes the following form [4]:

$$\begin{aligned}
\mathrm{elbo}(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})) &= -\mathrm{KL}\big(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})\big) + \ln\big(p(\mathbf{x})\big) \\
&= \mathbb{E}_{q(\boldsymbol{\theta})}\big[\ln\big(p(\mathbf{x} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\big)\big] - \mathbb{E}_{q(\boldsymbol{\theta})}\big[\ln\big(q(\boldsymbol{\theta})\big)\big]
\end{aligned} \tag{3.7}$$

Its name comes from the fact that it provides a lower bound for the model evidence. This can be seen by rewriting the formula above as

$$\ln\big(p(\mathbf{x})\big) = \mathrm{elbo}(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})) + \mathrm{KL}\big(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})\big)$$

and using the fact that the KL-divergence is non negative leads to

$$\ln\big(p(\mathbf{x})\big) \geq \mathrm{elbo}(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x}))$$

Having defined this new objective function, one can observe that since $\ln\big(p(\mathbf{x})\big)$ is constant in $\mathscr{D}$, maximizing the ELBO is equivalent to minimizing the KL-divergence.

For a better insight into what it practically means to maximize the evidence lower bound, we can rewrite (3.7) following the work of Blei et al. [4]:

$$\begin{aligned} \text{elbo}(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})) &= \mathbb{E}_{q(\boldsymbol{\theta})}\left[\ln\left(p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})\right)\right] - \mathbb{E}_{q(\boldsymbol{\theta})}\left[\ln\left(q(\boldsymbol{\theta})\right)\right] \\ &= \mathbb{E}_{q(\boldsymbol{\theta})}\left[\ln\left(p(\mathbf{x} \mid \boldsymbol{\theta})\right)\right] + \mathbb{E}_{q(\boldsymbol{\theta})}\left[\ln\left(p(\boldsymbol{\theta})\right)\right] - \mathbb{E}_{q(\boldsymbol{\theta})}\left[\ln\left(q(\boldsymbol{\theta})\right)\right] \\ &= \mathbb{E}_{q(\boldsymbol{\theta})}\left[\ln\left(p(\mathbf{x} \mid \boldsymbol{\theta})\right)\right] - \text{KL}\left(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta})\right) \end{aligned} \tag{3.8}$$

From (3.8) we can deduce that maximizing the evidence lower bound means

- Maximizing the expected log-likelihood. This can be interpreted as encouraging distributions of parameters $\boldsymbol{\theta}$ that explain the data $\mathbf{x}$ the best.
- Minimizing the KL divergence between $q(\boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$. This means that we require the approximate distribution $q^*(\boldsymbol{\theta})$ to be close to the prior.

3.2.2 Local Variational Methods

In the previous section we have seen that variational methods aim to maximizing a lower bound on the model evidence. The lower bound is found by bounding the whole posterior distribution $p(\boldsymbol{\theta} \mid \mathbf{x})$, so such methods are called *global* variational methods.
A simpler, but often equally powerful, strategy is to bound individual variables or terms within the posterior distribution and then combine those bounds to obtain one for the model evidence. Naturally, such methods are referred to as *local* variational methods [3].

In this thesis, we are interested in Bayesian logistic regression, so the model evidence is given by

$$p(\mathbf{y}) = \int_{\mathbb{R}^p} p(\mathbf{y} \mid \boldsymbol{\beta}) p(\boldsymbol{\beta}) d\boldsymbol{\beta}$$

where $p(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and the likelihood is given by (1.11) as:

$$p(\mathbf{y} \mid \boldsymbol{\beta}) = \prod_{i=1}^{n} p(y_i; \mathbf{x}_i, \boldsymbol{\beta}) = \prod_{i=1}^{n} \pi_i(\boldsymbol{\beta})^{y_i} (1 - \pi_i(\boldsymbol{\beta}))^{1-y_i}$$

where $\pi_i(\boldsymbol{\beta}) = \sigma(\mathbf{x}_i^\top \boldsymbol{\beta})$. The strategy adopted by Jaakkola and Jordan [13], is to approximate $\ln(\pi_i(\boldsymbol{\beta}))$ by a lower bound which is a quadratic function $\eta_i = \mathbf{x}_i^\top \boldsymbol{\beta}$, resulting in a Gaussian posterior. Notice $\pi_i(\boldsymbol{\beta})$ is a sigmoid transformation of the linear predictor, see Figure 3.1, and Bishop [3], and Jaakkola and Jordan [13] show that

$$\pi_i(\boldsymbol{\beta}) = \sigma(\mathbf{x}_i^\top \boldsymbol{\beta}) \geq \sigma(\xi_i) \exp\left\{ \frac{\mathbf{x}_i^\top \boldsymbol{\beta} - \xi_i}{2} - \lambda(\xi_i)((\mathbf{x}_i^\top \boldsymbol{\beta})^2 - \xi_i^2) \right\} \tag{3.9}$$

where each ξ_i is a **variational parameter** which has to be optimized for the bound to be optimal, and

$$\lambda(\xi_i) := \frac{1}{2\xi_i} \left[\sigma(\xi_i) - \frac{1}{2} \right] \tag{3.10}$$

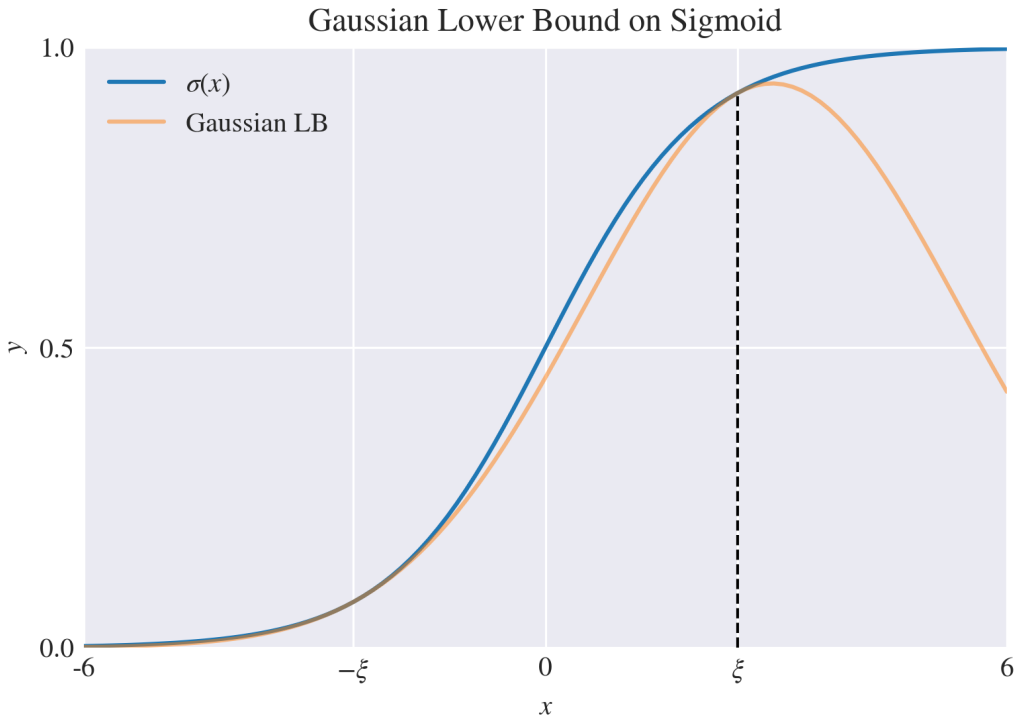

Figure 3.1: Example of Jaakkola and Jordan's lower bound

Our aim is to use (3.9) to bound each factor $p(y_i;\mathbf{x}_i,\boldsymbol{\beta})$ in (1.11). In order to do this, we can rewrite each term as follows:

$$
\begin{aligned}
p(y_i;\mathbf{x}_i,\boldsymbol{\beta}) &= \pi_i(\boldsymbol{\beta})^{y_i}(1-\pi_i(\boldsymbol{\beta}))^{1-y_i} \\
&= \sigma(\mathbf{x}_i^\top\boldsymbol{\beta})^{y_i}(1-\sigma(\mathbf{x}_i^\top\boldsymbol{\beta}))^{1-y_i} \\
&= \left(\frac{\exp\{\mathbf{x}_i^\top\boldsymbol{\beta}\}}{1+\exp\{\mathbf{x}_i^\top\boldsymbol{\beta}\}}\right)^{y_i}\left(1-\frac{\exp\{\mathbf{x}_i^\top\boldsymbol{\beta}\}}{1+\exp\{\mathbf{x}_i^\top\boldsymbol{\beta}\}}\right)^{1-y_i} \\
&= \frac{\exp\{\mathbf{x}_i^\top\boldsymbol{\beta}y_i\}}{\left(1+\exp\{\mathbf{x}_i^\top\boldsymbol{\beta}\}\right)^{y_i}}\frac{1}{1+\exp\{\mathbf{x}_i^\top\boldsymbol{\beta}\}}\left(1+\exp\{\mathbf{x}_i^\top\boldsymbol{\beta}\}\right)^{y_i} \\
&= \exp\{\mathbf{x}_i^\top\boldsymbol{\beta}y_i\}\frac{1}{1+\exp\{\mathbf{x}_i^\top\boldsymbol{\beta}\}} \\
&= \exp\{\mathbf{x}_i^\top\boldsymbol{\beta}y_i\}\frac{\exp\{-\mathbf{x}_i^\top\boldsymbol{\beta}\}}{1+\exp\{-\mathbf{x}_i^\top\boldsymbol{\beta}\}} \\
&= \exp\{\mathbf{x}_i^\top\boldsymbol{\beta}y_i\}\,\sigma(-\mathbf{x}_i^\top\boldsymbol{\beta}) \qquad (3.11)
\end{aligned}
$$

Now it is straightforward to apply the bound in (3.9) to (3.11)

$$p(y_i;\mathbf{x}_i^\top,\boldsymbol{\beta}) \geq \exp\{\mathbf{x}_i^\top\boldsymbol{\beta} y_i\}\,\sigma(\xi_i)\exp\left\{-\frac{\mathbf{x}_i^\top\boldsymbol{\beta}+\xi_i}{2}-\lambda(\xi_i)((\mathbf{x}_i^\top\boldsymbol{\beta})^2-\xi_i^2)\right\}$$
$$=\sigma(\xi_i)\exp\left\{\mathbf{x}_i^\top\boldsymbol{\beta} y_i-\frac{\mathbf{x}_i^\top\boldsymbol{\beta}+\xi_i}{2}-\lambda(\xi_i)((\mathbf{x}_i^\top\boldsymbol{\beta})^2-\xi_i^2)\right\}$$

and consequently we can bound the likelihood

$$p(\mathbf{y}\mid\boldsymbol{\beta}) \geq \tilde{p}\left(\mathbf{y}\mid\boldsymbol{\beta};\boldsymbol{\xi}\right) := \prod_{i=1}^{n}\sigma(\xi_i)\exp\left\{\mathbf{x}_i^\top\boldsymbol{\beta} y_i-\frac{\mathbf{x}_i^\top\boldsymbol{\beta}+\xi_i}{2}-\lambda(\xi_i)((\mathbf{x}_i^\top\boldsymbol{\beta})^2-\xi_i^2)\right\} \tag{3.12}$$

where $\boldsymbol{\xi} := (\xi_1,\ldots,\xi_n)^\top$ is a vector of variational parameters[2].
Recall that in Section 3.1.1 we have found a bound on $p(\mathbf{x}\mid\boldsymbol{\theta})p(\boldsymbol{\theta})$ that was an exponential of a quadratic function of $\boldsymbol{\theta}$, and then we have obtained a Gaussian approximation to the full posterior by normalizing it to become a proper normal distribution. Similarly, we can find the following bound

$$p(\mathbf{y}\mid\boldsymbol{\beta})\,p(\boldsymbol{\beta}) \geq \tilde{p}\left(\mathbf{y}\mid\boldsymbol{\beta};\boldsymbol{\xi}\right)p(\boldsymbol{\beta})$$

on the unnormalized posterior distribution. Since the logarithm is a strictly increasing function, we can equally write

$$\begin{aligned}
\ln\left(p(\mathbf{y},\boldsymbol{\beta})\right) &= \ln\left(p(\mathbf{y}\mid\boldsymbol{\beta})p(\boldsymbol{\beta})\right)\\
&\geq \ln\left(\tilde{p}\left(\mathbf{y}\mid\boldsymbol{\beta};\boldsymbol{\xi}\right)p(\boldsymbol{\beta})\right)\\
&= \ln\left(\tilde{p}\left(\mathbf{y}\mid\boldsymbol{\beta};\boldsymbol{\xi}\right)\right)+\ln\left(p(\boldsymbol{\beta})\right)\\
&= \sum_{i=1}^{n}\left[\ln\left(\sigma(\xi_i)\right)+\mathbf{x}_i^\top\boldsymbol{\beta} y_i-\frac{\mathbf{x}_i^\top\boldsymbol{\beta}+\xi_i}{2}-\lambda(\xi_i)\left((\mathbf{x}_i^\top\boldsymbol{\beta})^2-\xi_i^2\right)\right]-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^\top\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)+\text{const}\\
&= \sum_{i=1}^{n}\left[\mathbf{x}_i^\top\boldsymbol{\beta}\left(y_i-\frac{1}{2}\right)-\lambda(\xi_i)(\mathbf{x}_i^\top\boldsymbol{\beta})^2\right]-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^\top\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)+\text{const}\\
&= \sum_{i=1}^{n}\left[\boldsymbol{\beta}^\top\mathbf{x}_i\left(y_i-\frac{1}{2}\right)-\lambda(\xi_i)(\boldsymbol{\beta}^\top\mathbf{x}_i)^2\right]-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^\top\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)+\text{const}\\
&= \boldsymbol{\beta}^\top\sum_{i=1}^{n}\mathbf{x}_i\left(y_i-\frac{1}{2}\right)-\boldsymbol{\beta}^\top\left[\sum_{i=1}^{n}\lambda(\xi_i)\mathbf{x}_i\mathbf{x}_i^\top\right]\boldsymbol{\beta}-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^\top\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)+\text{const}\\
&= \boldsymbol{\beta}^\top\sum_{i=1}^{n}\mathbf{x}_i\left(y_i-\frac{1}{2}\right)-\boldsymbol{\beta}^\top\left[\sum_{i=1}^{n}\lambda(\xi_i)\mathbf{x}_i\mathbf{x}_i^\top\right]\boldsymbol{\beta}-\frac{1}{2}\boldsymbol{\beta}^\top\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\beta}+\boldsymbol{\beta}^\top\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0+\text{const}\\
&= \boldsymbol{\beta}^\top\left[\sum_{i=1}^{n}\mathbf{x}_i\left(y_i-\frac{1}{2}\right)+\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right]-\frac{1}{2}\boldsymbol{\beta}^\top\left[2\sum_{i=1}^{n}\lambda(\xi_i)\mathbf{x}_i\mathbf{x}_i^\top+\boldsymbol{\Sigma}_0^{-1}\right]\boldsymbol{\beta}+\text{const}
\end{aligned} \tag{3.13}$$

where we have absorbed all terms not depending on $\boldsymbol{\beta}$ into a constant and rewritten the entire expression in terms of powers of $\boldsymbol{\beta}$. We can then complete the square using equations (A.6)-(A.8) to

[2] In equation (3.12) we have defined $\tilde{p}\left(\mathbf{y}\mid\boldsymbol{\beta};\boldsymbol{\xi}\right) = \tilde{p}(\mathbf{y}\mid\mathbf{X}_D^{\text{RV}}=\mathbf{X}_D^{\text{obs}},\boldsymbol{\beta};\boldsymbol{\xi})$ to be equal to the right-hand side of the equation. This is not a density because it is not normalized. If we were to normalize it, it would cease to be a bound [3].

obtain the form of our variational Gaussian approximation

$$q(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta} \mid \boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v) \tag{3.14}$$

where the mean and variance-covariance matrix are given by

$$\boldsymbol{\Sigma}_v^{-1} = 2\sum_{i=1}^{n} \lambda(\xi_i)\mathbf{x}_i\mathbf{x}_i^\top + \boldsymbol{\Sigma}_0^{-1} \tag{3.15}$$

$$\boldsymbol{\mu}_v = \boldsymbol{\Sigma}_v\left(\sum_{i=1}^{n} \mathbf{x}_i\left(y_i - \frac{1}{2}\right) + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right) \tag{3.16}$$

So far, we have bounded each success probability $\pi_i(\boldsymbol{\beta})$ with a lower bound depending on the variational parameter ξ_i, then we have used the independence assumption to factorize our likelihood and we have used the bound on the success probability to find a lower bound for the whole likelihood. Next, we managed to find a normal approximation to the posterior distribution by first bounding $\ln\big(p(\mathbf{y} \mid \boldsymbol{\beta})p(\boldsymbol{\beta})\big)$ and then completing the square. Looking at equations (3.15)-(3.16) we quickly spot that they depend on the vector of variational parameters $\boldsymbol{\xi}$. Recall that in general Variational Inference our aim is to maximize the evidence lower bound. In local Variational Inference we maintain the same spirit, but instead of maximizing the ELBO, we maximize a different bound. In particular, the parameters $\boldsymbol{\xi}$ have to be determined via maximization of the following bound

$$p(\mathbf{y}) = \int_{\mathbb{R}^p} p(\mathbf{y} \mid \boldsymbol{\beta})p(\boldsymbol{\beta})d\boldsymbol{\beta} \geq \int_{\mathbb{R}^p} \widetilde{p}\left(\mathbf{y} \mid \boldsymbol{\beta}; \boldsymbol{\xi}\right) p(\boldsymbol{\beta})d\boldsymbol{\beta}$$

Once again, since the logarithm is a strictly increasing function, maximizing the bound above is just equivalent to maximizing

$$\mathscr{B}(\boldsymbol{\xi}) := \ln\left(\int_{\mathbb{R}^p} \widetilde{p}\left(\mathbf{y} \mid \boldsymbol{\beta}; \boldsymbol{\xi}\right) p(\boldsymbol{\beta})d\boldsymbol{\beta}\right). \tag{3.17}$$

There are two ways of maximizing (3.17). One method uses the EM algorithm, the second uses straight forward integration, luckily both lead to the same result.

3.2.2.1 Optimization of Variational Parameters via EM Algorithm

The EM algorithm is a procedure that can be used to find maximum likelihood estimates of a parameter $\boldsymbol{\xi}$ governing a model of observed[3] data $\mathbf{Y} = \mathbf{y}$ and unobserved or latent variables $\boldsymbol{\beta}$. Namely, suppose we aim to find the maximum likelihood estimate $\widehat{\boldsymbol{\xi}}_{\text{MLE}}$ maximizing the marginal likelihood for observed data given below

$$\mathscr{L}(\boldsymbol{\xi}; \mathbf{y}) = p(\mathbf{y} \mid \boldsymbol{\xi}) = \int_{\mathbb{R}^p} p(\mathbf{y}, \boldsymbol{\beta} \mid \boldsymbol{\xi})d\boldsymbol{\beta} \tag{3.18}$$

which is found by marginalizing the complete-data likelihood $p(\mathbf{y}, \boldsymbol{\beta} \mid \boldsymbol{\xi})$ over the latent variables $\boldsymbol{\beta}$. The EM-algorithm then proceeds as follows:

[3] Recall from Appendix A.2 that we consider our explanatory variables $\mathbf{X}_D^{\text{obs}}$ as being fixed constants. By observed data we mean the observations of the random variable $\mathbf{Y}$ of response variables.

Algorithm 3: EM Algorithm

1 Initialize maximum likelihood estimate $\boldsymbol{\xi}^{(0)} \in \mathbb{R}^p$ with a guess.
2 Until convergence:
3 **E-step**: Define the function $Q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^{(t)})$ as the expected value of complete-data log-likelihood $\ln\big(p(\mathbf{y}, \boldsymbol{\beta} \mid \boldsymbol{\xi})\big)$ with respect to $p(\boldsymbol{\beta} \mid \mathbf{y}, \boldsymbol{\xi}^{(t)})$, the conditional distribution over the latent variables, given the observed data $\mathbf{y}$ and the current estimate of the maximum likelihood estimator $\boldsymbol{\xi}^{(t)}$.

$$Q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^{(t)}) := \mathbb{E}_{p(\boldsymbol{\beta} \mid \mathbf{y}, \boldsymbol{\xi}^{(t)})}\left[\ln\big(p(\mathbf{y}, \boldsymbol{\beta} \mid \boldsymbol{\xi})\big)\right]$$

4 **M-step**: Maximize the function $Q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^{(t)})$ with respect to $\boldsymbol{\xi}$ to find the new estimate of the maximum likelihood estimator $\boldsymbol{\xi}^{(t+1)}$

$$\boldsymbol{\xi}^{(t+1)} := \arg\max_{\boldsymbol{\xi}} Q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^{(t)})$$

end

Notice that the complete-data likelihood in (3.18) here is nothing more than $\widetilde{p}\big(\mathbf{y} \mid \boldsymbol{\beta}; \boldsymbol{\xi}\big)\, p(\boldsymbol{\beta})$ encountered in Section 3.2.2 where $\widetilde{p}\big(\mathbf{y} \mid \boldsymbol{\beta}; \boldsymbol{\xi}\big)$ is defined in equation (3.12). Also notice that in the discussion in the previous section, we state the our goal is to maximize the bound on the marginal likelihood given by

$$p(\mathbf{y}) \geq p(\mathbf{y} \mid \boldsymbol{\xi}) := \int_{\mathbb{R}^p} \widetilde{p}\big(\mathbf{y} \mid \boldsymbol{\beta}; \boldsymbol{\xi}\big)\, p(\boldsymbol{\beta}) d\boldsymbol{\beta}$$

with respect to the variational parameters $\boldsymbol{\xi}$. Since logarithms are strictly increasing we have that $p(\mathbf{y}) \geq p(\mathbf{y} \mid \boldsymbol{\xi})$ implies

$$\ln\big(p(\mathbf{y})\big) \geq \ln\big(p(\mathbf{y} \mid \boldsymbol{\xi})\big) = \ln\left(\int_{\mathbb{R}^p} \widetilde{p}\big(\mathbf{y} \mid \boldsymbol{\beta}; \boldsymbol{\xi}\big)\, p(\boldsymbol{\beta}) d\boldsymbol{\beta}\right) =: \mathscr{B}(\boldsymbol{\xi}).$$

where $\mathscr{B}(\boldsymbol{\xi})$ was defined in the same way in equation (3.17).
It follows, that we can perform local variational inference and find the values of $\boldsymbol{\xi}$ maximizing the evidence lower bound by performing the EM-algorithm described in Algorithm 3 as this will deliver the value of $\boldsymbol{\xi}$ maximizing the marginal likelihood $p(\mathbf{y} \mid \boldsymbol{\xi})$, and since the marginal likelihood $p(\mathbf{y} \mid \boldsymbol{\xi})$ is exactly our bound on the model evidence $p(\mathbf{y})$, the two methods will be equivalent. Notice we will use the variational approximate posterior $q(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta} \mid \boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)$ as the conditional distribution for the Expectation, where its mean and variance-covariance matrix are given by (3.16) and (3.15) respectively.

Algorithm 3 can be simplified because, in this case, it is easy to find an expression for $Q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^{(t)})$ and

maximizing it with respect to ξ_i. Let us rewrite $Q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^{(t)})$ as follows

$$
\begin{aligned}
Q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^{(t)}) &= \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\ln\left(p(\mathbf{y},\boldsymbol{\beta} \mid \boldsymbol{\xi})\right)\right] \\
&= \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\ln\left(\widetilde{p}\left(\mathbf{y} \mid \boldsymbol{\beta};\boldsymbol{\xi}\right) p(\boldsymbol{\beta})\right)\right] \\
&= \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\ln\left(\widetilde{p}\left(\mathbf{y} \mid \boldsymbol{\beta};\boldsymbol{\xi}\right)\right)\right] + C && \text{no } \boldsymbol{\xi} \text{ dependence} \\
&= \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\sum_{i=1}^{n}\left(\ln\left(\sigma(\xi_i)\right) + \mathbf{x}_i^\top \boldsymbol{\beta} y_i - \frac{\mathbf{x}_i^\top \boldsymbol{\beta} + \xi_i}{2} - \lambda(\xi_i)\left((\mathbf{x}_i^\top \boldsymbol{\beta})^2 - \xi_i^2\right)\right)\right] + C && \text{by (3.12)} \\
&= \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\sum_{i=1}^{n}\left(\ln\left(\sigma(\xi_i)\right) + \boldsymbol{\beta}^\top \mathbf{x}_i y_i - \frac{\boldsymbol{\beta}^\top \mathbf{x}_i + \xi_i}{2} - \lambda(\xi_i)\left(\mathbf{x}_i^\top \boldsymbol{\beta}\boldsymbol{\beta}^\top \mathbf{x}_i - \xi_i^2\right)\right)\right] + C \\
&= \sum_{i=1}^{n}\left(\ln\left(\sigma(\xi_i)\right) + \boldsymbol{\mu}_v^\top \mathbf{x}_i\left(y_i - \frac{1}{2}\right) - \frac{\xi_i}{2} - \lambda(\xi_i)\mathbf{x}_i^\top \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\boldsymbol{\beta}^\top\right]\mathbf{x}_i + \lambda(\xi_i)\xi_i^2\right) + C && \text{by (3.14)} \\
&= \sum_{i=1}^{n}\left(\ln\left(\sigma(\xi_i)\right) - \frac{\xi_i}{2} - \lambda(\xi_i)\mathbf{x}_i^\top \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\boldsymbol{\beta}^\top\right]\mathbf{x}_i + \lambda(\xi_i)\xi_i^2\right) + C
\end{aligned}
$$

where C at each step we have defined C to be a constant absorbing all terms not depending on $\boldsymbol{\xi}$. Now we can take the derivative of this expression with respect to ξ_k for $k \in \{1,\dots,n\}$

$$
\begin{aligned}
\frac{\partial Q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^{(t)})}{\partial \xi_k} &= \frac{\partial}{\partial \xi_k}\left(\sum_{i=1}^{n}\left(\ln\left(\sigma(\xi_i)\right) - \frac{\xi_i}{2} - \lambda(\xi_i)\mathbf{x}_i^\top \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\boldsymbol{\beta}^\top\right]\mathbf{x}_i + \lambda(\xi_i)\xi_i^2\right) + C\right) \\
&= \frac{\partial}{\partial \xi_k}\left(\ln\left(\sigma(\xi_k)\right) - \frac{\xi_k}{2} - \lambda(\xi_k)\mathbf{x}_k^\top \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\boldsymbol{\beta}^\top\right]\mathbf{x}_k + \lambda(\xi_k)\xi_k^2\right) \\
&= \frac{1}{\sigma(\xi_k)}\sigma(\xi_k)\left(1-\sigma(\xi_k)\right) - \frac{1}{2} - \lambda'(\xi_k)\mathbf{x}_k^\top \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\boldsymbol{\beta}^\top\right]\mathbf{x}_k + \lambda'(\xi_k)\xi_k^2 + 2\lambda(\xi_k)\xi_k && \text{by (3.3)} \\
&= \frac{1}{2} - \sigma(\xi_k) - \lambda'(\xi_k)\mathbf{x}_k^\top \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\boldsymbol{\beta}^\top\right]\mathbf{x}_k + \lambda'(\xi_k)\xi_k^2 + 2\xi_k\frac{1}{2\xi_k}\left(\sigma(\xi_k) - \frac{1}{2}\right) && \text{by (3.10)} \\
&= -\lambda'(\xi_k)\left(\mathbf{x}_k^\top \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\boldsymbol{\beta}^\top\right]\mathbf{x}_k - \xi_k^2\right) && (3.19)
\end{aligned}
$$

We can now set this equal to zero to find an update formula for ξ_k. Before doing so, we can show the following facts:

- The bound found by Jaakkola and Jordan and given in equation (3.9) is symmetric around 0. This will allow us to consider only positive variational parameters ξ_i, so that we can take the positive square root later on.
- The function $\lambda(\xi_i)$ is strictly decreasing for $\xi_i > 0$. This will allow us to divide through by $\lambda'(\xi_k)$ in (3.19).

First, we can notice that the Sigmoid function has the following useful property

$$\sigma(-x) = 1 - \sigma(x) \tag{3.20}$$

We can then use it to show that $\lambda(\xi_i)$ is symmetric around zero:

$$\begin{aligned}
\lambda(-\xi_i) &= \frac{1}{2(-\xi_i)}\left(\sigma(-\xi_i) - \frac{1}{2}\right) \\
&= -\frac{1}{2\xi_i}\left(1 - \sigma(\xi_i) - \frac{1}{2}\right) \\
&= \frac{1}{2\xi_i}\left(\sigma(\xi_i) - \frac{1}{2}\right) \\
&= \lambda(\xi_i) \qquad (3.21)
\end{aligned}$$

Finally, we can show that bound (3.9) is indeed symmetric around $\xi_i = 0$

$$\begin{aligned}
\sigma(-\xi_i)\exp\left(\frac{\mathbf{x}_i^\top\boldsymbol{\beta} - (-\xi_i)}{2} - \lambda(-\xi_i)\left((\mathbf{x}_i^\top\boldsymbol{\beta})^2 - \xi_i^2\right)\right) &= \sigma(-\xi_i)\exp\left(\frac{\mathbf{x}_i^\top\boldsymbol{\beta} + \xi_i}{2} - \lambda(\xi_i)\left((\mathbf{x}_i^\top\boldsymbol{\beta})^2 - \xi_i^2\right)\right) \quad \text{by (3.21)} \\
&= \frac{e^{-\xi_i}}{1+e^{-\xi_i}}\exp\left(\frac{\mathbf{x}_i^\top\boldsymbol{\beta} + \xi_i}{2} - \lambda(\xi_i)\left((\mathbf{x}_i^\top\boldsymbol{\beta})^2 - \xi_i^2\right)\right) \quad \text{def. of } \sigma(\xi_i) \\
&= \frac{1}{1+e^{-\xi_i}}\exp\left(\frac{\mathbf{x}_i^\top\boldsymbol{\beta} - \xi_i}{2} - \lambda(\xi_i)\left((\mathbf{x}_i^\top\boldsymbol{\beta})^2 - \xi_i^2\right)\right) \\
&= \frac{e^{\xi_i}}{1+e^{\xi_i}}\exp\left(\frac{\mathbf{x}_i^\top\boldsymbol{\beta} - \xi_i}{2} - \lambda(\xi_i)\left((\mathbf{x}_i^\top\boldsymbol{\beta})^2 - \xi_i^2\right)\right) \\
&= \sigma(\xi_i)\exp\left(\frac{\mathbf{x}_i^\top\boldsymbol{\beta} - \xi_i}{2} - \lambda(\xi_i)\left((\mathbf{x}_i^\top\boldsymbol{\beta})^2 - \xi_i^2\right)\right)
\end{aligned}$$

which finishes the proof. One can also inspect the following bound graphically as both ξ_i and $\mathbf{x}_i^\top\boldsymbol{\beta}$ change, as illustrated by Figure 3.2. Since the bound is symmetric, we can restrict ourselves to considering values of ξ_i that are non-negative.

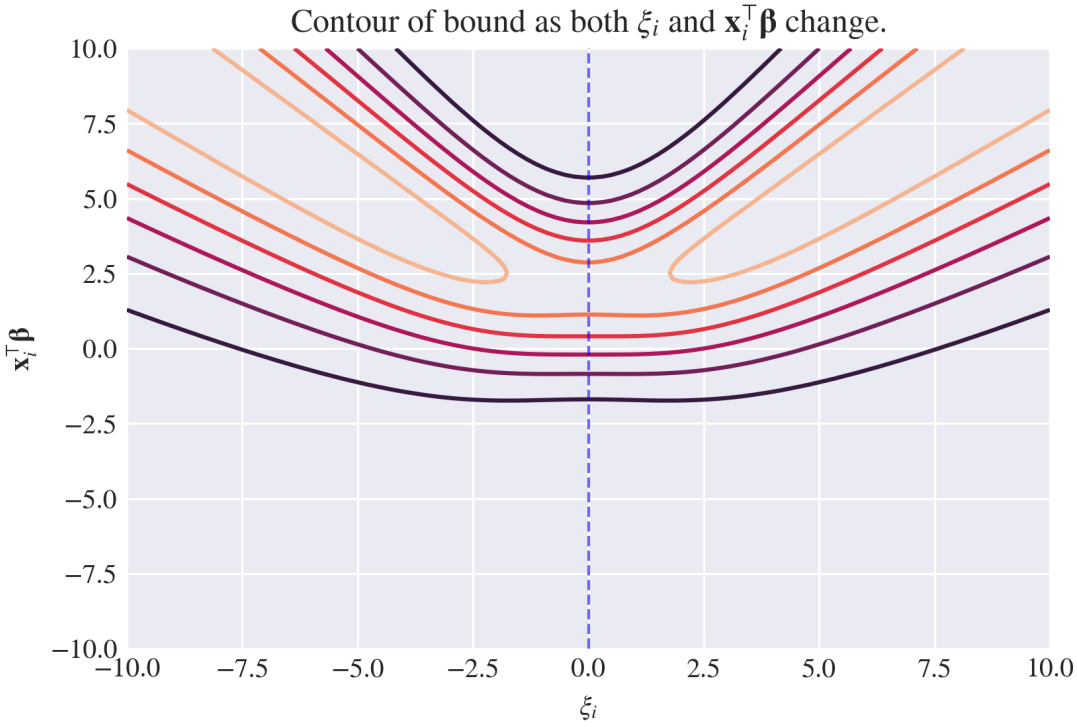

Figure 3.2: Illustration of Symmetry for variational bound

Now consider $\xi_i > 0$. We can take the derivative of $\lambda(\xi_i)$ and show that it is strictly negative for all $\xi_i > 0$, and thus it is a strictly decreasing function. Indeed we have

$$\begin{aligned}
\frac{d\lambda(\xi_i)}{d\xi_i} &= \frac{d}{d\xi_i}\left(\frac{1}{2\xi_i}\left(\sigma(\xi_i) - \frac{1}{2}\right)\right) \\
&= -\frac{1}{2\xi_i^2}\left(\frac{e^{\xi_i}}{1+e^{\xi_i}} - \frac{1}{2}\right) + \frac{1}{2\xi_i}\left(\frac{e^{\xi_i}(1+e^{\xi_i}) - e^{2\xi_i}}{(1+e^{\xi_i})^2}\right) \\
&= -\frac{1}{4\xi_i^2}\left(\frac{e^{\xi_i}-1}{1+e^{\xi_i}}\right) + \frac{e^{\xi_i}}{2\xi_i(1+e^{\xi_i})^2} \\
&= \frac{-e^{2\xi_i} + 1 + 2\xi_i e^{\xi_i}}{4\xi_i^2(1+e^{\xi_i})^2}
\end{aligned}$$

Now consider the numerator and the series expansion of the exponential function:

$$\begin{aligned}
1+2\xi_i\sum_{k=0}^{\infty}\frac{\xi_i^k}{k!}-\sum_{k=0}^{\infty}\frac{(2\xi_i)^k}{k!} &= 1+\sum_{k=0}^{\infty}\frac{2\xi_i^{k+1}}{k!}-1-\sum_{k=1}^{\infty}\frac{2^k\xi_i^k}{k!} \\
&= \sum_{k=0}^{\infty}\frac{2\xi_i^{k+1}}{k!}-\sum_{k=1}^{\infty}\frac{2^k\xi_i^k}{k!} \\
&= \sum_{k=0}^{\infty}\frac{2\xi_i^{k+1}}{k!}-\sum_{j=0}^{\infty}\frac{2^{j+1}\xi_i^{j+1}}{(j+1)!} && \text{define } j := k-1 \\
&= \sum_{k=0}^{\infty}\frac{2\xi_i^{k+1}}{k!}\left(1-\frac{2^k}{k+1}\right) \\
&< 0 && \text{By Bernoulli inequality, since } \xi_i > 0
\end{aligned}$$

Therefore $\lambda'(\xi_i) < 0$ for $\xi_i > 0$, and in particular $\lambda'(\xi_i) \neq 0$ for $\xi_i > 0$. In follows that

$$\frac{\partial Q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^{(t)})}{\partial \xi_k} = 0 \qquad \Longrightarrow \qquad \xi_k^2 = \mathbf{x}_k^\top \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\boldsymbol{\beta}^\top\right]\mathbf{x}_k$$

Since the bound is symmetric around 0 it follows that we can consider the positive square root so that the updated parameter is given by

$$\xi_k^{(t+1)} = \sqrt{\mathbf{x}_k^\top \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\boldsymbol{\beta}^\top\right]\mathbf{x}_k}$$

Now we can find the expectation above by recalling our variational distribution in (3.14) and by using

$$\mathrm{Var}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\right] = \mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\boldsymbol{\beta}^\top\right] - \left(\mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\right]\right)^2$$

to obtain

$$\mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\boldsymbol{\beta}^\top\right] = \mathrm{Var}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\right] + \left(\mathbb{E}_{p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi}^{(t)})}\left[\boldsymbol{\beta}\right]\right)^2 = \boldsymbol{\Sigma}_v + \boldsymbol{\mu}_v\boldsymbol{\mu}_v^\top$$

which immediately leads to a computable update formula for the variational parameters

$$\xi_k^{(t+1)} = \sqrt{\mathbf{x}_k^\top\left(\boldsymbol{\Sigma}_v + \boldsymbol{\mu}_v\boldsymbol{\mu}_v^\top\right)\mathbf{x}_k}$$

Putting everything together we obtain

Algorithm 4: Variational Approximation

Initialize $\boldsymbol{\xi} = (\xi_1^{(1)}, \ldots, \xi_n^{(1)})^\top \in \mathbb{R}^n$ randomly.

for $j = 1, 2, \ldots, \Delta$:

  for $i = 1, 2, \ldots, n$ **do**:

    Find mean and variance-covariance matrix and update variational parameters.

$$\boldsymbol{\Sigma}_v = \left(2 \sum_{i=1}^{n} \lambda(\xi_i^{(j)}) \mathbf{x}_i \mathbf{x}_i^\top + \boldsymbol{\Sigma}_0^{-1} \right)^{-1}$$

$$\boldsymbol{\mu}_v = \boldsymbol{\Sigma}_v \left(\sum_{i=1}^{n} \mathbf{x}_i \left(y_i - \frac{1}{2} \right) + \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0 \right)$$

$$\xi_i^{(j+1)} = \sqrt{\mathbf{x}_i^\top \left(\boldsymbol{\Sigma}_v + \boldsymbol{\mu}_v \boldsymbol{\mu}_v^\top \right) \mathbf{x}_i}$$

  end

end

Notice that one could use several stopping criteria however, in practice, this algorithm converges after just a very small number of iterations Δ. In the algorithm we write $\xi_i^{(j)}$ to indicate the approximation at iteration $j \in \{1, \ldots, \Delta\}$ of the variational parameter associated with the i-th observation.

Chapter 4

Implementation and Results

This last chapter starts by briefly describing how the various approximations seen in the previous chapters have been coded in Python. Full details of such functions or classes can be found in the GitHub repository linked in Appendix B. Successively, we will see some of the results obtained in different Bayesian logistic regression case studies and how RWMH, Laplace approximation and Variational methods compare in such circumstances.

4.1 Implementation Details

4.1.1 Datasets

In the following sections we will explore different Bayesian Regression case studies where the data has been simulated using the function `generate_bernoulli(size, params)`. It takes the data set size and the parameters[1] as arguments, and outputs two arrays of length $n =$`size`. The first is the design matrix $\mathbf{X}_D^{\text{obs}}$ with the first column containing only 1s and the other columns containing (standard)-normally distributed random observations. The second array contains the binary Bernoulli observations $\mathbf{y}$ where each success probability is given by

$$p_i = \sigma\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right) \qquad \text{for } i = 1, \dots, n$$

4.1.2 Target Distribution

The unnormalized posterior distribution shown in equation (1.13) leads to very unstable and lengthy calculations due to the many products in the likelihood. For a better implementation, I have coded

[1] Recall the number of parameters p is equal to the number of explanatory variables plus 1.

the log-posterior in equation (1.14) instead. This can be found as the method `log_posterior` of the class `ExplanatoryVariables` contained in the file `explanatory_variables`.

4.1.3 Random-Walk Metropolis-Hastings Implementation

I have implemented a RWMH algorithm in Python in order to obtain samples from the posterior distribution for Bayesian logistic regression, as discussed in Section 1.3. In order to speed up and improve the algorithm several features were implemented.

1. **Normalization constants**: As noted at the end of Section 2.3.1 $p(\boldsymbol{\theta} \mid \mathbf{x})$ and $q(\cdot \mid \boldsymbol{\theta})$ can both be coded up to a normalization constant as this will cancel out in the ratio of the acceptance probability.

2. **Pre-computing**: It is much faster to generate all the N uniform samples $u_1, \dots, u_N \sim \mathcal{U}(0,1)$ at once at the start of the algorithm, rather than generating each u_i at the corresponding iteration.

3. **Affine Property of Normal Distributions**: Let $\mathbf{X} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$. Then the random variable defined as $\mathbf{Y} := \mathbf{a} + \mathbf{B}\mathbf{X}$ is still normally distributed $\mathbf{Y} \sim \mathcal{N}\left(\mathbf{a} + \mathbf{B}\boldsymbol{\mu}, \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^\top\right)$. In particular, we can make the algorithm more efficient by using the same principle as in 2 and generate, at the start, N samples $\boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_N^* \sim \mathcal{N}\left(\mathbf{0}, \sigma_p^2 I\right)$ and then, at each iteration i, we can "shift" the corresponding sample so that it is centered at the current value taken by $\boldsymbol{\theta}_i$, i.e. the candidate is given by $\boldsymbol{\theta}_i + \boldsymbol{\theta}_i^*$ which will be a realization of a random variable distributed as $\mathcal{N}\left(\boldsymbol{\theta}_i, \sigma_p^2 I\right)$.

4. **Log-scale**: The acceptance probability ratio can become very unstable to compute because the likelihood, which appears both in the numerator and denominator, will consist in a product of terms that can be very small or very large, leading to underflow or overflow respectively. To avoid this issue, we can use the fact that the logarithm is a strictly increasing function and therefore we can work with the acceptance probability on the log scale:

$$\ln\left(\alpha\left(\boldsymbol{\theta}^*, \boldsymbol{\theta}_i\right)\right) = \min\left\{0, \ln\left(p\left(\boldsymbol{\theta}^* \mid \mathbf{x}\right)\right) - \ln\left(p\left(\boldsymbol{\theta}_i \mid \mathbf{x}\right)\right)\right\}$$

   Then the condition $u_i \leq \alpha(\boldsymbol{\theta}^*, \boldsymbol{\theta}_i)$ in Algorithm 2 becomes $\ln(u_i) \leq \ln\left(\alpha\left(\boldsymbol{\theta}^*, \boldsymbol{\theta}_i\right)\right)$. Actually, we can simplify this condition even more by noticing that if $u_i \in [0,1]$, then $\ln(u_i) \leq 0$. This means that we can accept $\boldsymbol{\theta}^*$ as the value of $\boldsymbol{\theta}_{i+1}$ whenever $\ln(u_i) \leq \ln\left(p\left(\boldsymbol{\theta}^* \mid \mathbf{x}\right)\right) - \ln\left(p\left(\boldsymbol{\theta}_i \mid \mathbf{x}\right)\right)$.

5. **Recycling calculations**: Even in the log-scale, each evaluation of the log-posterior might be expensive so we want to avoid to re-compute the same value multiple times. For this reason, before calculating $\ln\left(p\left(\boldsymbol{\theta}^* \mid \mathbf{x}\right)\right) - \ln\left(p\left(\boldsymbol{\theta}_i \mid \mathbf{x}\right)\right)$ we will store both of these terms so that, in case of an acceptance, $\ln\left(p\left(\boldsymbol{\theta}^* \mid \mathbf{x}\right)\right)$ will be recycled, and in case of a rejection $\ln\left(p\left(\boldsymbol{\theta}_i \mid \mathbf{x}\right)\right)$ will.

6. **Burn-in**: The initial samples will be quite dependent on the initial choice of the value of $\boldsymbol{\theta}_0$. In order to obtain a more precise sample we can choose a number of *burn-in samples* that will be thrown away. In other words, if we want to generate N RWMH samples and we want a burn-in

of B samples, the algorithm will actually generate $B+N$ samples and then throw away the first B.

7. **Thinning**: Often we need to generate a very large number of samples in order to obtain accurate estimates. In such cases, storage can become an issue and we need to resort to a process called *thinning*. Suppose that we have generated 1 million samples but we only want to store 100 000. Our algorithm could be used with a thinning of 10 so that only every 10-th sample would be stored and the rest would be thrown away. Putting this in context with burn-in, if we wanted to obtain N RWMH samples with a burn-in of B and a thinning of T, the algorithm would actually run to generate $B+(N-1)\times T+1$ samples[2] and then return `samples[B::T]` in Python.

This algorithm was implemented in Python with the function `metropolis`, which can be found in the file `utility_functions.py`.

Due to the nature of MCMC algorithms we want to have a set of tools to assess whether RWMH has converged. In this thesis, I have used three main diagnostic plots [6]:

- **Trace plots**: These illustrate how quickly the chain mixes[3] with a time-series plot for each of the p parameters $\beta_1,\ldots,\beta_p$: if the samples seem to still be exploring Θ after a large number of iterations, then the chain is not mixing rapidly. Essentially we are looking for a time-series plot that shows no obvious pattern and whose behavior of the initial samples is similar to that of the last samples. This would provide evidence towards convergence of the chain.

- **Auto-Correlation plots**: It displays, for each parameter $\beta_1,\ldots,\beta_p$, the behavior of the Auto-Correlation Function (ACF). This function returns the correlation of each $\beta_1,\ldots,\beta_p$ with itself as the number of lags[4] increases. Successive samples are dependent by construction of the Markov Chain, so we aim to see the auto-correlation to decrease to zero as the number of lags increases. Ideally, we would like to see a decay that looks geometric.

- **Partial Auto-Correlation plots**: Similar to the Auto-Correlation plot, however when the lag is larger than 1, i.e. we are not looking at the preceding time step, the Partial Auto-Correlation Function (PACF) removes the correlation coming from the time-steps in between. In other words, it does not take into account "indirect" correlations due to the value of the parameter at shorter lags. We aim to have PAC plots with no "direct" correlation after a small number of iterations.

[2] Notice that this works in Python because it is zero-indexed and because of how slicing works. See the docs.

[3] Mixing means moving through the parameter space Θ. A Markov Chain that mixes rapidly gets close to the true distribution $p(\boldsymbol{\theta}\,|\,\mathbf{x})$ with a "small" number of samples.

[4] A lag is simply a time-step.

4.1.4 Laplace Implementation

In order to find the mean of the Laplace approximation I have used a pre-built optimization routine to minimize the negative log-posterior, which is equivalent to maximizing the posterior, i.e. finding the mode. The `minimize` function found in the `scipy.optimize` library also returns an approximation to the inverse hessian matrix, which is can then be used as the variance-covariance matrix.
Various functions and methods to obtain the laplace approximations can be found in the class `ExplanatoryVariables`.

4.1.5 Variational Implementation

The variational approximation is found by implementing the EM-Algorithm detailed in Algorithm 4, further details can be found in the appropriate methods of the class `ExplanatoryVariables`.

4.2 Case Studies Results

4.2.1 Models with Explanatory Variables

In this subsection we consider models where the design matrix $\mathbf{X}_D^{\text{obs}}$ has more than one column. Put differently, we have at least two parameters $\boldsymbol{\beta} = (\beta_1, \beta_2)^\top$, one for the intercept and the rest for the explanatory variables.
The simplest scenario that we can consider is when $p = 2$ so that we have two parameters $\boldsymbol{\beta} = (\beta_1, \beta_2)^\top$ and one explanatory variable $\mathbf{x}_i = (1, x_{i1})^\top$. Figure 4.1 shows the trace plot for a model with $n = 1\,000$ observations whose true parameters are $\boldsymbol{\beta} = (1, 0.5)^\top$ where RWMH was run with $N = 200\,000$ samples while burning-in the first $B = 1\,000$.

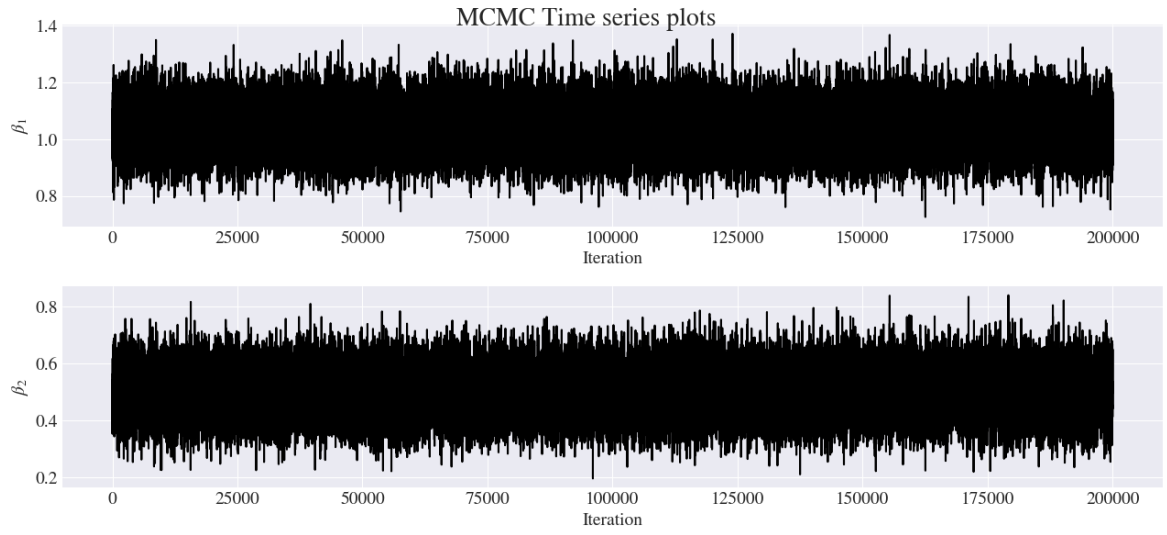

Figure 4.1: Trace of RWMH for $p = 2$.

No obvious pattern can be discerned from the two plots so we conclude that the chain is mixing rapidly. In addition, the AC and PAC plots shown in Figure 4.2 show a geometric decay and a negligible partial correlation at lags larger than 1, respectively, confirming the Markov property.

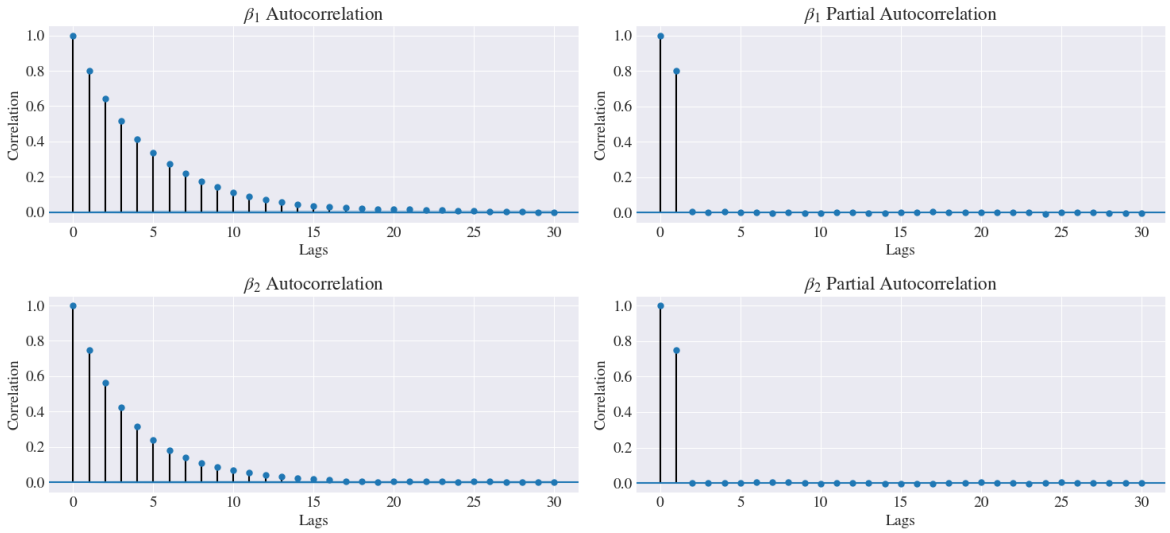

Figure 4.2: AC and PAC plots for $p = 2$.

In order to have an idea of what the posterior looks like we plot in Figure 4.3 a histogram of the samples and superimpose a line corresponding to the Kernel Density Estimation of such samples.

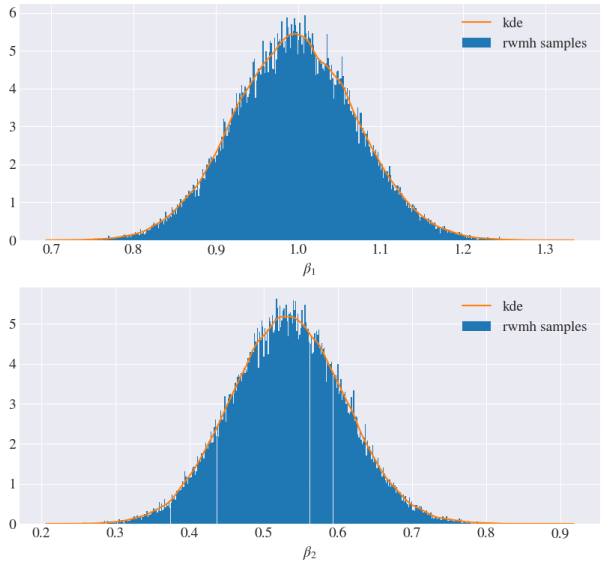

Figure 4.3: Samples Histogram and KDE for $p = 2$

Figure 4.4 compares the KDE for β_1 and β_2 of Figure 4.3 with the marginal distributions given by Laplace and Variational methods, while Figure 4.5 shows contour and surface plots.

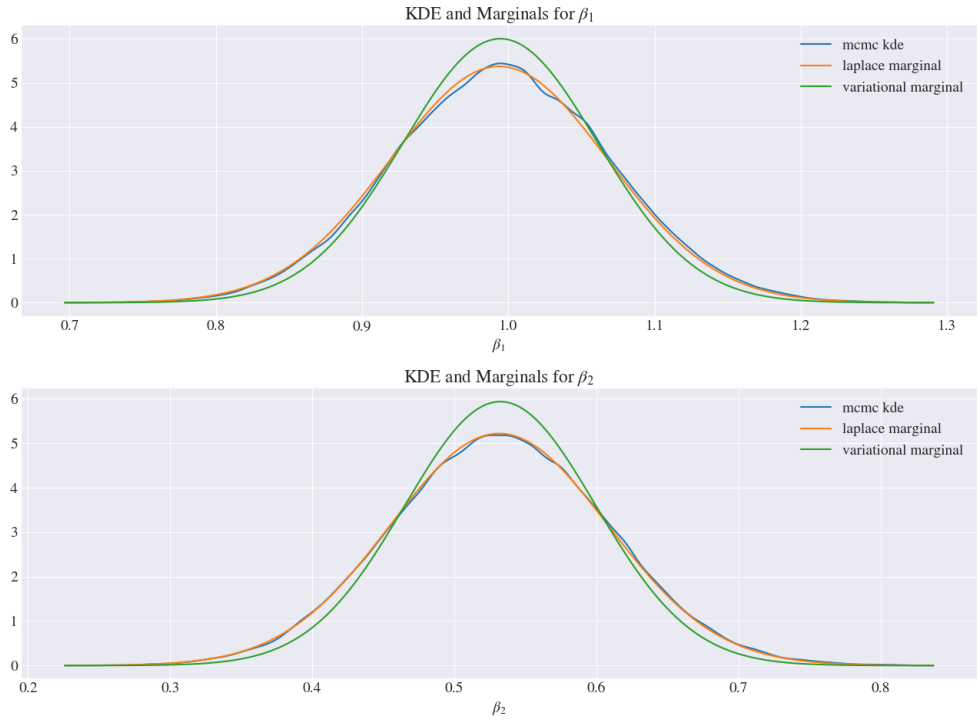

Figure 4.4: RWMH, Laplace and Variational marginals for $p = 2$.

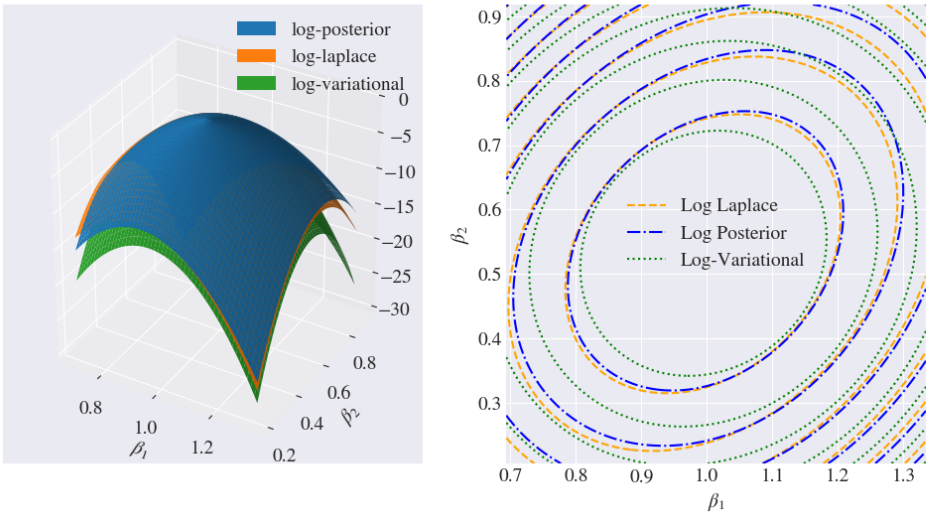

Figure 4.5: Surface plots for $p = 2$.

In the latter plot, I have subtracted the maximum value of each curve to every grid evaluation so that the mode of the curves all coincides to 0. This allows us to compare the curves even if we can only evaluate the log-posterior distribution up to a normalization constant.
In Figure 4.4 we use the KDE of the MCMC samples as our reference curve, because sampling methods like RWMH are considered the "gold standard" for intractable posterior distributions. We can notice that both Laplace and Variational marginals seem to agree on the mode of the distribution, however, the variance of the Variational marginal is smaller than the other two, leading to an underestimation of the tails. Laplace method therefore seems to outperform Variational and provide a similar approximation to the RWMH one.
A similar behavior can be seen for a larger number of parameters p both for small and large sample sizes n. Surprisingly, on rare occasions, and only for a handful of parameters, we see that the variational approximation and the KDE curves coincide almost perfectly, while Laplace approximation is off. This offers an alternative insight to that of Jaakkola and Jordan [13] where they showed that for only one Bernoulli observation Variational consistently outperformed Laplace. Some examples of the behavior described above can be found in the GitHub repository.

4.2.2 Model with no Explanatory Variables

In order to explore the unexpected behavior seen in the previous section, we want to see what happens in the limiting case of having no explanatory variables and using a flat prior. This allows us to compare the approximations to the likelihood rather than to the posterior.
This scenario is convenient because we can write down the posterior in a much nicer form.
Here we will assume that we have $y_1, \dots y_n \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(\pi)$ with $\mathbf{x}_i = 1$ essentially being a scalar and we will call the only parameter β.

$$\beta := \ln\left(\frac{\pi}{1-\pi}\right) \tag{4.1}$$

The likelihood for β can be found from equation (1.11) by setting $\boldsymbol{\beta} = (\beta)^\top$ and $\mathbf{X}_D^{\text{obs}} = (1, \dots, 1)^\top$ so that $\pi_i(\mathbf{x}_i^\top \boldsymbol{\beta}) = \pi(\beta) = \frac{e^\beta}{1+e^\beta}$

$$\mathscr{L}(\mathbf{y} \mid \beta) = \prod_{i=1}^{n} \left(\frac{e^\beta}{1+e^\beta}\right)^{y_i} \left(\frac{1}{1+e^\beta}\right)^{1-y_i} = \frac{e^{\beta n \bar{y}}}{\left(1+e^\beta\right)^n} \tag{4.2}$$

Next, we introduce a flat prior

$$p(\beta) \propto 1 \tag{4.3}$$

Putting together (4.2) and (4.3) we obtain an expression proportional to the posterior for β:

$$p(\beta \mid \mathbf{y}) \propto \frac{e^{\beta n \bar{y}}}{\left(1+e^\beta\right)^n}$$

We now find the posterior for π and by using the change of variable in equation (4.1) and multiplying by its derivative with respect to π. After a few lines of rearrangement we obtain:

$$p(\pi \mid \mathbf{y}) \propto \pi^{n\overline{y}-1}(1-\pi)^{n-n\overline{y}-1}$$

which we recognize to have the form of a Beta$(n\overline{y}, n-n\overline{y})$, which means that the proportionality constant is given by $B(n\overline{y}, n-n\overline{y})^{-1}$ where $B(\cdot,\cdot)$ is the beta function.
Therefore, we can write the posterior distribution for β in a closed form:

$$p(\beta \mid \mathbf{y}) = \frac{1}{B(n\overline{y}, n-n\overline{y})} \frac{e^{\beta n\overline{y}}}{\left(1+e^{\beta}\right)^{n}} \tag{4.4}$$

4.2.2.1 Laplace Approximation

We can find the mode by differentiating the log-posterior found in equation (4.4) and setting it to zero to find the mode:

$$\frac{d}{d\beta}\left(\beta n\overline{y} - n\ln\left(1+e^{\beta}\right)\right) = 0 \quad \Longrightarrow \quad \widehat{\beta} = \ln\left(\frac{\overline{y}}{1-\overline{y}}\right) \tag{4.5}$$

and we find the observed information matrix by taking the reciprocal of the negative second derivative of (4.4) with respect to β and evaluate it at $\beta = \widehat{\beta}$

$$\left[\frac{\partial^2 \ell(\beta \mid \mathbf{y})}{\partial \beta^2}\Big|_{\beta=\widehat{\beta}}\right]^{-1} = \frac{1}{n\overline{y}(1-\overline{y})} \tag{4.6}$$

Then the Laplace approximation is given by

$$q_l(\beta) = \mathcal{N}\left(\ln\left(\frac{\overline{y}}{1-\overline{y}}\right), \frac{1}{n\overline{y}(1-\overline{y})}\right) \tag{4.7}$$

4.2.2.2 Variational Approximation

Following the same steps as in Section 3.2.2, specifically of the derivation ending with equation (3.13), with $\boldsymbol{\beta} = (\beta)^{\top}$, $\mathbf{X}_D^{\text{obs}} = (1,\ldots,1)^{\top}$, and $\ln\left(p(\beta)\right) \propto 1$ being absorbed into the constant terms, we obtain

$$\left(\widehat{\sigma}_V^2\right)^{-1} = 2\sum_{i=1}^{n} \lambda(\xi_i) \qquad \text{and} \qquad \widehat{\mu}_V = \widehat{\sigma}_V^2 n\left(\overline{y} - \frac{1}{2}\right) \tag{4.8}$$

where the parameters are optimized with the recurrent formula below:

$$\xi_k^{(t+1)} := \sqrt{\widehat{\sigma}_V^2\left(1 + n\overline{y} - \frac{n}{2}\right)} \tag{4.9}$$

so that our Variational approximation to the posterior is

$$q_v(\beta) = \mathcal{N}\left(\widehat{\mu}_V, \widehat{\sigma}_V^2\right)$$

4.2.2.3 Results

The advantage of this scenario is that we don't need to use RWMH because we have the complete, normalized posterior distribution available. In addition, since this model has a flat prior on β and no explanatory variables, we expect to be able to appreciate the properties of the Laplace and Variational approximations better, because the influence of the arbitrary choice for $p(\beta)$ is reduced.
In Figure 4.6 we see Laplace and Variational approximations to the true posterior and to the true log-posterior for a sample size of $n = 300$, where $\beta = 1.0$ is the true value of β. It is clear that the Variational one is underestimating the tails of the distribution, and the Laplace normal distribution has a tighter fit.

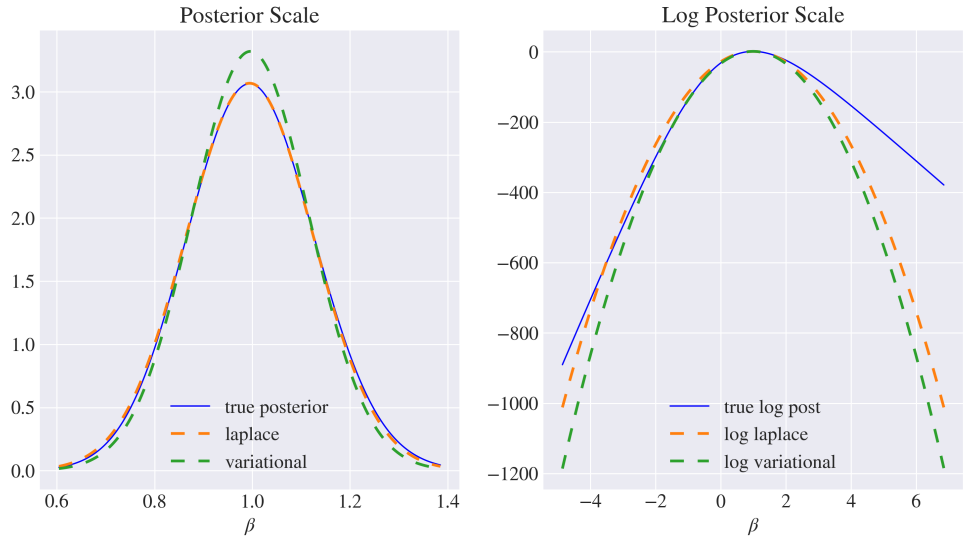

Figure 4.6: Posterior, Laplace and Variational for $p = 2$.

However, Figure 4.7 sheds further insight into the difference between the two normal distributions. Specifically, the mean of the Variational normal distribution is consistently closer to the population mode, i.e. the true value of β, than the mean of the Laplace one. In addition, Laplace normal has a consistently larger standard deviation, which results in a better fit. Overall, while the Variational distribution is more compact than the Laplace one, their mean both clearly converge to the true value of β and to a zero standard deviation, as the sample size increases.

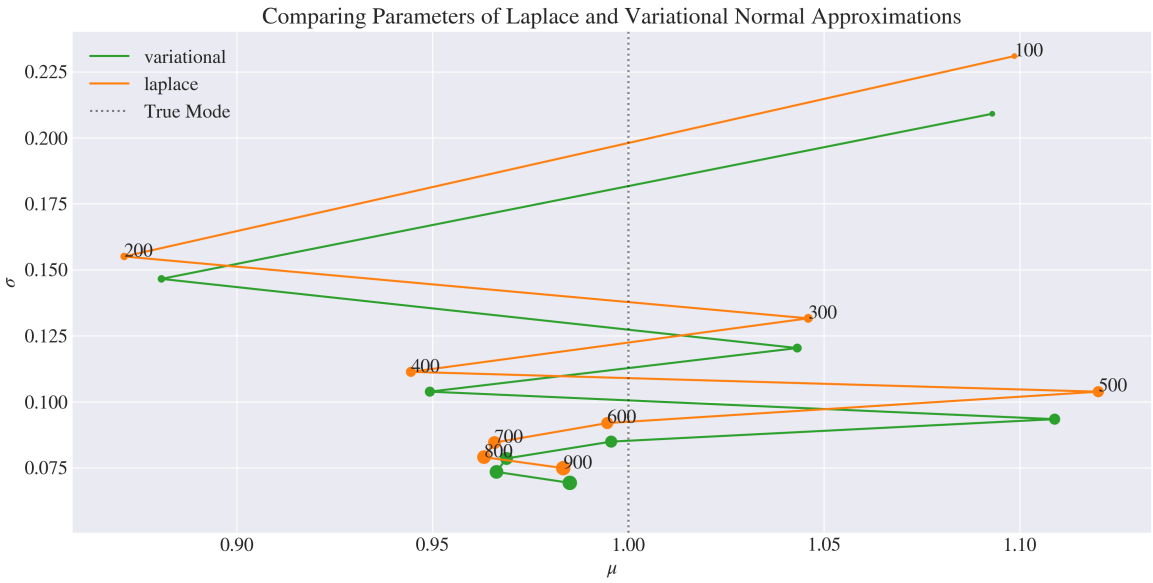

Figure 4.7: Convergence of Variational and Laplace μ and σ as sample size increases.

The tendency of the Variational distribution to have a mean closer to the population mode β is more evident for smaller data set sizes as the sample mode, which is also the mean of the Laplace normal, is not going to be a very good approximation of the population mode. This can be seen from Figure 4.7 where the numbers on the scatter points, and their relative size, represents the sample size.

4.3 Conclusion and Recommendations

The two case studies discussed in this chapter show that Laplace approximation has a closer fit to the posterior distribution than the Variational one, especially when the sample size is large enough for the sample mode to be very close to the population mode. The variational approximation underestimates the tails, however its mean is consistently closer to the true mode than Laplace, particularly for small sample sizes. This suggests that one might prefer to use Variational Inference when the data set size is small and we are interested in estimating the true value of β. As an example, we might want to use the Variational approximation when β represents some characteristic of a very rare disease, for which we have a limited number of observations.
It is important to note that in this thesis we have only worked with a uni-modal posterior distribution. Further work should explore the comparison of these techniques to multi-modal distributions. We expect the Variational approach to outperform the Laplace one in such scenario because it should be able to pick up on "global" features of the posterior, rather than focusing on one of the modes. Other lines of research could look into approximating the posterior using a mixture of Laplace normal approximations, where each approximation is fitted to a different mode. This could be compared with a similar application of Variational methods, as well as novel MCMC techniques specifically

aimed at multi-modal target distributions [17].

APPENDIX A

APPENDIX A - STATISTICS BACKGROUND

We review some fundamental concepts in probability and statistics: the Exponential family, Generalized Linear Models, the multivariate normal distribution, the KL-divergence and Markov Chains.

A.1 Exponential Family of Distributions

Many of the most common probability distributions, such as Normal and Bernoulli, can be written, after an appropriate re-parametrization, in a common format which will allow us to construct a general theoretical framework for regression. In particular, we say that the probability (density) of a random variable Y belongs to the exponential family if we can rewrite it as

$$f_Y(y \mid \boldsymbol{\theta}) = h(y) \exp\left(\boldsymbol{\eta}(\boldsymbol{\theta})^\top \mathbf{T}(y) - A(\boldsymbol{\theta})\right) \tag{A.1}$$

where $\boldsymbol{\theta}$ is the vector of *natural parameters*, $\mathbf{T}(y)$ is a vector of *sufficient statistics*, and $A(\boldsymbol{\theta})$ is often called *log-partition function* and it ensures that the probability distribution function integrates to 1.

A.1.1 Bernoulli Distribution

A *binary* random variable Y that takes value 1 with probability π and value 0 with probability $1-\pi$ is said to be Bernoulli. It has probability mass function defined below

$$p(y;\pi) = \pi^y (1-\pi)^{1-y} \qquad \text{for } y \in \{0,1\} \tag{A.2}$$

and we denote this by $Y \sim \text{Bernoulli}(\pi)$. We can rearrange its probability mass function to show that the Bernoulli distribution belongs to the Exponential Family by noting that in this case we have

a scalar parameter $\boldsymbol{\theta} = (\pi)^\top \in \mathbb{R}$ rather than a vector of parameters, and by using the fact that the logarithm is the inverse of exponentiation:

$$\begin{aligned} p(y;\pi) &= \pi^y (1-\pi)^{1-y} \\ &= \exp\big(y\ln(\pi) + (1-y)\ln(1-\pi)\big) \\ &= \exp\big(y\ln(\pi) + \ln(1-\pi) - y\ln(1-\pi)\big) \\ &= \exp\left(y\ln\left(\frac{\pi}{1-\pi}\right) + \ln(1-\pi)\right) \end{aligned}$$

In expression (A.1) we can set

$$\begin{aligned} h(y) &= 1 \\ \boldsymbol{\eta}(\boldsymbol{\theta}) &= \eta(\pi) = \ln\left(\frac{\pi}{1-\pi}\right) \\ \mathbf{T}(y) &= T(y) = y \\ A(\boldsymbol{\theta}) &= A(\pi) = \ln(1-\pi) \end{aligned}$$

which shows that the Bernoulli distribution belongs to the Exponential Family.

A.2 Generalized Linear Models

Generalized Linear Models are powerful for modelling the relationship between a set of *independent* or *explanatory* variables, and a set of *dependent* or *response* variables.

A.2.1 Explanatory Variables

Suppose that we have $n \times p$ random variables $X_{11},\ldots,X_{1p},\ldots\ldots,X_{n1},\ldots,X_{np}$ which we arrange into a so-called design matrix for convenience

$$\mathbf{X}_D^{\text{RV}} := \begin{pmatrix} \mathbf{X}_1^\top \\ \vdots \\ \mathbf{X}_n^\top \end{pmatrix} = \begin{pmatrix} X_{11} & \cdots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{np} \end{pmatrix}$$

where the superscript of $\mathbf{X}_D^{\text{RV}}$ indicates that its elements are random variables, and $\mathbf{X}_i = (X_{i1},\ldots,X_{ip})^\top$. In addition, suppose that we have observed the realization of all these random variables, which we arrange in matrix form again

$$\mathbf{X}_D^{\text{obs}} := \begin{pmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix} \tag{A.3}$$

where each row of $\mathbf{X}_D^{\text{obs}}$ is an observation of p **explanatory variables**.
Throughout this thesis we will only consider the observed design matrix $\mathbf{X}_D^{\text{obs}}$, meaning that whenever exploratory variables will be involved, we will be considering realizations $X_{ij} = x_{ij}$. This is the standard approach in regression modelling.

A.2.2 Response Variables

In addition to recording n observations $\mathbf{x}_i$, we also consider n corresponding random variables $\mathbf{Y} = (Y_1, \ldots, Y_n)^\top$ and their realizations $\mathbf{y} = (y_1, \ldots, y_n)^\top$, where each Y_i follows the same member of the exponential family.

$$f_{Y_i}(y_i \mid \boldsymbol{\theta}) = h(y_i) \exp\left(\boldsymbol{\eta}(\boldsymbol{\theta})^\top \mathbf{T}(y_i) - A(\boldsymbol{\theta})\right)$$

To model the relationship between each pair of Y_i and $\mathbf{x}_i$, we choose a suitable function $g : \mathbb{R} \to \mathbb{R}$ that is *invertible* and *differentiable* and we assume

$$g(\mathbb{E}[Y_i \mid \mathbf{X}_i = \mathbf{x}_i]) = \mathbf{x}_i^\top \boldsymbol{\beta} \tag{A.4}$$

where $\boldsymbol{\beta} := (\beta_1, \ldots, \beta_p)^\top$ is a vector of parameters in $\mathbb{R}^p$.

A.3 Multivariate Normal Distribution

Recall that a multivariate normal distribution with mean $\boldsymbol{\mu}$ and with $D \times D$ covariance matrix $\boldsymbol{\Sigma}$ can be written as

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{\det(\boldsymbol{\Sigma})^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\} \tag{A.5}$$

Very often one works with quadratic functions of $\mathbf{x}$ and wants to "complete the square" to rewrite such expressions as multivariate Gaussian distributions. By taking the natural logarithm of expression (A.5) we can then rearrange the terms in powers of $\mathbf{x}$

$$\begin{aligned}
\ln(\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})) &= -\frac{D}{2}\ln(2\pi) - \frac{1}{2}\det(\Sigma) - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \\
&= -\frac{D}{2}\ln(2\pi) - \frac{1}{2}\det(\Sigma) - \frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} + \frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{1}{2}\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\
&= -\frac{D}{2}\ln(2\pi) - \frac{1}{2}\det(\Sigma) - \frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} + \mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \frac{1}{2}\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\
&= -\frac{1}{2}\left(D\ln(2\pi) + \det(\Sigma) + \boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right) - \frac{1}{2}\left(-2\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right) - \frac{1}{2}\left(\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x}\right) \\
&= -\frac{1}{2}\left(\mathscr{A}_0 + \mathbf{x}^\top \mathscr{A}_1 + \mathbf{x}^\top \mathscr{A}_3 \mathbf{x}\right)
\end{aligned}$$

where

$$\mathscr{A}_0 := D\ln(2\pi) + \det(\Sigma) + \boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \tag{A.6}$$

$$\mathscr{A}_1 := -2\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \tag{A.7}$$

$$\mathscr{A}_2 := \boldsymbol{\Sigma}^{-1} \tag{A.8}$$

One can then use equations (A.6)-(A.8) in completing the square.

A.4 Information Theory

The Kullback-Leibler divergence of a continuous distribution $p(\boldsymbol{\theta} \mid \mathbf{x})$ with respect to a continuous distribution $q(\boldsymbol{\theta})$, denoted by $\mathrm{KL}\big(p(\boldsymbol{\theta} \mid \mathbf{x}) \mid\mid q(\boldsymbol{\theta})\big)$, can be interpreted as the amount of information lost when $q(\boldsymbol{\theta})$ is used to approximate $p(\boldsymbol{\theta} \mid \mathbf{x})$ [5]. It is defined as

$$\mathrm{KL}\big(p(\boldsymbol{\theta} \mid \mathbf{x}) \mid\mid q(\boldsymbol{\theta})\big) := \mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{x})}\left[\ln\left(\frac{p(\boldsymbol{\theta} \mid \mathbf{x})}{q(\boldsymbol{\theta})}\right)\right] = \int_{\Theta} p(\boldsymbol{\theta} \mid \mathbf{x}) \ln\left(\frac{p(\boldsymbol{\theta} \mid \mathbf{x})}{q(\boldsymbol{\theta})}\right) d\boldsymbol{\theta} \tag{A.9}$$

Notice that if $p(\boldsymbol{\theta} \mid \mathbf{x})$ is our true posterior distribution, it will generally be unknown, which means that the integration above is intractable. Deterministic methods for approximating $p(\boldsymbol{\theta} \mid \mathbf{x})$ by minimizing (A.9) are generally called Expectation Propagation methods, and will not be covered in this thesis. A much simpler problem to solve, is to minimize the reversed KL-divergence $\mathrm{KL}\big(q(\boldsymbol{\theta}) \mid\mid p(\boldsymbol{\theta} \mid \mathbf{x})\big)$. In what follows we show that the KL-divergence is non-negative.

A.4.1 Non-negativity Property

One can use the power series definition of e^x to show

$$\begin{aligned}
e^{x-1} &:= \sum_{k=1}^{\infty} \frac{(x-1)^k}{k!} \\
&= 1 + (x-1) + \frac{(x-1)^2}{2} + \dots \\
&= x + \frac{(x-1)^2}{2} + \dots \\
&\geq x \qquad\qquad \text{for } x > 0
\end{aligned}$$

where equality $e^{x-1} = x$ holds for $x = 1$. Since also the logarithm is a strictly increasing function, one has

$$x - 1 \geq \ln(x) \qquad \forall\, x > 0 \tag{A.10}$$

Now consider the negative KL-divergence

$$\begin{aligned}
-\mathrm{KL}\big(q(\boldsymbol{\theta}) \mid\mid p(\boldsymbol{\theta} \mid \mathbf{x})\big) &= -\int_{\Theta} q(\boldsymbol{\theta}) \ln\left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \mathbf{x})}\right) d\boldsymbol{\theta} \\
&= \int_{\Theta} q(\boldsymbol{\theta}) \ln\left(\frac{p(\boldsymbol{\theta} \mid \mathbf{x})}{q(\boldsymbol{\theta})}\right) d\boldsymbol{\theta} \\
&\leq \int_{\Theta} q(\boldsymbol{\theta}) \left(\frac{p(\boldsymbol{\theta} \mid \mathbf{x})}{q(\boldsymbol{\theta})} - 1\right) d\boldsymbol{\theta} \qquad \text{using (A.10)} \\
&= \int_{\Theta} p(\boldsymbol{\theta} \mid \mathbf{x}) - q(\boldsymbol{\theta})\, d\boldsymbol{\theta} \\
&= \int_{\Theta} p(\boldsymbol{\theta} \mid \mathbf{x})\, d\boldsymbol{\theta} - \int_{\Theta} q(\boldsymbol{\theta})\, d\boldsymbol{\theta} \\
&= 1 - 1 \\
&= 0
\end{aligned}$$

so that $\mathrm{KL}\big(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x})\big) \geq 0$ with equality achieved only when $\frac{p(\boldsymbol{\theta}|\mathbf{x})}{q(\boldsymbol{\theta})} = 1$ or equivalently $p(\boldsymbol{\theta} \mid \mathbf{x}) = q(\boldsymbol{\theta})$ for every $\boldsymbol{\theta} \in \Theta$.

APPENDIX B

APPENDIX B - COMPUTING

Here the reader will find the link to the GitHub repository storing the code and images used in this thesis. In addition, I will provide a brief description of how to navigate the repository and what can be found in it.

B.1 Code Structure

All the code for this dissertation can be found here.

- **images**: Contains all the images shown in this thesis plus additional ones that were created for testing purposes. In particular, images for different runs of the algorithms are saved there.
- `explanatory_variables.py`: Code to run all the functions and case studies relating to Section 4.2.1.
- `no_explanatory_variables.py`: Code to run all the functions and case studies relating to Section 4.2.2.
- `gaussian_bound_on_sigmoid.py`: Code to reproduce figure 3.1.
- `sampling_methods.py`: Code to reproduce Figure 2.1 and Figure 2.3.
- `uniform_monte_carlo.py`: Code to reproduce Figure 2.2.
- `variational_bound_symmetric.py`: Code to reproduce Figure 3.2.
- `utility_functions.py`: Common functions used by several scripts. These include `sigmoid()`, `metropolis()` and various helper plotting functions.

BIBLIOGRAPHY

[1] C. ANDRIEU, N. D. FREITAS, AND ET AL., *An introduction to mcmc for machine learning*, 2003.

[2] J. E. ANGUS, *The probability integral transform and related results*, SIAM Review, 36 (1994), pp. 652–654.

[3] C. M. BISHOP, *Pattern Recognition and Machine Learning*, Springer, 2006.

[4] D. M. BLEI, A. KUCUKELBIR, AND J. D. MCAULIFFE, *Variational inference: A review for statisticians*, Journal of the American Statistical Association, 112 (2017), pp. 859–877.

[5] A. D. R. BURNHAM, K. P., *Model Selection and Multimodel Inference*, Springer, 2nd ed., 2002, p. 51.

[6] P. S. P. COWPERTWAIT AND A. V. METCALFE, *Introductory Time Series with R*, Springer Publishing Company, Incorporated, 1st ed., 2009.

[7] B. D. FLURY, *Acceptance–rejection sampling made easy*, Siam Review - SIAM REV, 32 (1990).

[8] M. DEGROOT AND M. SCHERVISH, *Probability and statistics*, Pearson custom library, Pearson Education, 2013.

[9] L. DEVROYE, *Non-uniform random variate generation*, 1986.

[10] D. GAMERMAN AND H. LOPES, *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, Second Edition*, Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis, 2006.

[11] W. GILKS, S. RICHARDSON, AND D. SPIEGELHALTER, *Markov Chain Monte Carlo in Practice*, Chapman & Hall/CRC Interdisciplinary Statistics, Taylor & Francis, 1995.

[12] W. K. HASTINGS, *Monte carlo sampling methods using markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.

[13] T. S. JAAKKOLA AND M. I. JORDAN, *A variational approach to bayesian logistic regression models and their extensions*, 1996.

[14] D. J. C. MACKAY, *Information Theory, Inference & Learning Algorithms*, Cambridge University Press, New York, NY, USA, 2002.

[15] N. METROPOLIS, A. ROSENBLUTH, M. ROSENBLUTH, A. TELLER, AND E. TELLER, *Equations of state calculations by fast computing machines*, Journal of Chemical Physics, 21 (1953), pp. 1087–1091.

[16] A. O'HAGAN AND J. J. FORSTER, *Kendall's Advanced Theory of Statistics, volume 2B: Bayesian Inference, second edition*, vol. 2B, Arnold, 2004.

[17] E. POMPE, C. HOLMES, AND K. ŁATUSZYŃSKI, *A Framework for Adaptive MCMC Targeting Multimodal Distributions*, arXiv e-prints, (2018), p. arXiv:1812.02609.

[18] C. P. ROBERT, *The Metropolis–Hastings Algorithm*, American Cancer Society, 2015, pp. 1–15.

[19] C. P. ROBERT AND G. CASELLA, *Monte Carlo Statistical Methods (Springer Texts in Statistics)*, Springer-Verlag, Berlin, Heidelberg, 2005.

[20] G. O. ROBERTS, A. GELMAN, AND W. R. GILKS, *Weak convergence and optimal scaling of random walk metropolis algorithms*, Ann. Appl. Probab., 7 (1997), pp. 110–120.