Dynamic Covariance Modelling Using Generalised Wishart Processes

Fredrik Nilsson

Master's thesis
2023:E68

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

Centrum Scientiarum Mathematicarum

Master's Theses in Mathematical Sciences 2023:E68
ISSN 1404-6342

LUTFMS-3490-2023

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lu.se/

Dynamic Covariance Modelling Using Generalised Wishart Processes

Fredrik Nilsson

Spring 2023

Abstract

Modern portfolio theory was pioneered by Markowitz who formulated the mean-variance problem, without which any discussion on quantitative approaches to portfolio selection would be incomplete. The framework boils down to finding the expected return μ and covariance Σ, after which the solution is proportional to $\Sigma^{-1}\mu$. Although the problem is simple at heart, finding estimates of the components constitutes an entire field of research. Common estimators are weighted sample means, and there exist various techniques designed to separate information from noise – the difficulty of which makes matters even worse when inverting the covariance.

In this project, we take a more probabilistic route to modelling the covariance and deploy a Markov chain Monte Carlo algorithm to perform Bayesian inference. We extend on existing frameworks by tailoring a Hamiltonian Monte Carlo algorithm to improve sampling efficiency. The model is validated on synthetic datasets and deployed on financial data in the form of future contract return series. Results are on par with benchmark models based on exponentially weighted moving averages, and we notice particular improvement by modelling the precision matrix Σ^{-1} directly, thus circumventing the otherwise problematic inversion.

Keywords: Covariance matrix, generalised Wishart process, Bayesian inference, Markov chain Monte Carlo, Hamiltonian Monte Carlo

Acknowlegdements

I would like to extend my utmost gratitude to Lynx Asset Management for the opportunity to write my Master's thesis here. The immeasurable knowledge and nurturing culture have been an absolute delight to experience. A special thanks to my supervisor Tianfang Zhang for his brilliant guidance, but also to Tobias Rydén and the rest of the research team for their valuable inputs. Furthermore I would like to thank Tim Jyrkäs, Kajsa Bjelle, and Pelle Nydahl for all the discussion and feedback throughout our time here. Last but certainly not least, this thesis would not have been possible without Magnus Wiktorsson, whom I owe gratitude not only for supervising my thesis on behalf of Lund Institute of Technology, but also for his extraordinary lecturing on Monte Carlo methods and derivative pricing. It has been a true pleasure.

Nomenclature

Mathematics

$\mathbb{R}$	The set of real numbers
$\mathbb{R}^+$	The set of non-negative real numbers, $\{x \in \mathbb{R} : x \geq 0\}$
$\mathcal{S}_D$	The set of symmetric $D \times D$ matrices
$\mathcal{S}_D^+$	The set of positive definite $D \times D$ matrices, $\{X \in \mathcal{S}_D : X \succ 0\}$
$\overline{\mathcal{S}_D^+}$	The set of positive semi-definite $D \times D$ matrices, $\{X \in \mathcal{S}_D : X \succeq 0\}$
I	Identity matrix, of appropriate size
∇	Gradient operator
$\|\cdot\|$	Vector norm, matrix norm
$\circ$	Entry-wise product
tr	Trace operator
vec	Vectorisation operator (suffix '-h' operates on lower triangular)
chol	Cholesky decomposition

Portfolio Theory

R_t	Arithmetic return on day t, normalised by σ_t
w_t	Portfolio weights on day t
σ_t	Volatility on day t

Statistics

$\mathcal{N}$	Normal distribution
$\mathcal{U}$	Uniform distribution
Γ	Gamma function
Σ	Covariance matrix, or just covariance
μ	Expectation, expected return
D	Dimensions, number of assets
ν	Degrees of freedom
N	Number of data points

Contents

1 Introduction

In his groundbreaking paper, Markowitz (1952) defines the mean-variance (MV) problem, which laid the foundation of modern portfolio theory. The idea is very simple; how do we choose a portfolio that has high expected returns with as low variance as possible? The trade-off can be defined as a quadratic program that incorporates the expected return μ and covariance matrix Σ of an asset universe. Slightly different formulations exist, but a common one is

$$\max_{w} \quad \mu^\top w - \lambda w^\top \Sigma w \tag{1}$$

where w are portfolio weights and $\lambda > 0$ is a risk aversion parameter. Oftentimes the formulation is constrained by having non-negative weights, and commonly that they should all sum to unity, meaning we allocate our entire capital. For the time being we will delay any constraints, and touch upon this later. The solution is proportional to $\Sigma^{-1}\mu$ and although elegant, problems quickly present themselves when we start to estimate μ and Σ in practice. We will focus on the latter, and arguably the most common approach is to use the sample covariance, possibly weighted in favour of more recent data points. The estimates often prove to be very noisy (Michaud 1989) and subsequently ill-conditioned. This makes inverting the covariance very problematic, and a common way to combat this is by shrinkage (Ledoit and Wolf 2003). We will revisit shrinkage very shortly, but in the most basic sense it means forming a linear combination of the sample covariance and some other, more structured matrix. The aim is for the resulting shrunk matrix to better adhere to some desirable properties. However, even with heavy regularisation, portfolios are not guaranteed to even outperform equally weighted ones. Furthermore, if we choose an exponential weighting on our sample covariance, we must choose a rate of decay that balances responsiveness and robustness, which is another dilemma of this traditional approach. Placing too much trust in new observations comes with a sensitivity to outliers or otherwise unusual events and results in unstable estimates. Contrarily, if the rate of decay is too small, the estimates will be very slow to react to actual changes in the underlying structure, which is why we want to adopt a dynamic framework in the first place.

In this project we aim to combat some of the aforementioned shortcomings akin to the standard procedure by choosing a Bayesian approach to modelling the covariance matrix. We will adapt a framework introduced by Wilson and Ghahramani (2010) that utilises the generalised Wishart process, and perform inference in order to find a posterior distribution of the covariance given the observed data. Advantages are manifold, and will be discussed throughout the paper. To mention a few, the first advantage is that we can choose to model the inverse covariance instead, circumventing the problematic matrix inversion mentioned above. Moreover, we do not have to choose between responsiveness and robustness, and we get a distribution of portfolio weights rather than just a single estimate, which can be useful for a number of reasons.

The project is conduced in collaboration with Lynx Asset management AB (Lynx), which is a managed futures/CTA fund based in Stockholm, Sweden.

1.1 Structure

This paper is structured as follows. Section 2 gives some background to the problem at hand and a brief rundown of common concepts. In section 3 we introduce the Wishart process and develop our model. Section 4 gives an introduction to Bayesian inference and some of its appeal, while section 5 accounts for Markov Chain Monte Carlo, as well as key algorithms utilised. Section 6 explains the data and describes the experiments and relevant metrics. The results are presented in section 7 and the paper is then rounded off with some final discussion in section 8.

2 Fundamentals

Assume any financial instrument and let $\{S_t\}$ be its price process. We need not define its dynamics, but we do restrict ourselves to processes with finite second moment, $\mathbb{E}\left[S^2\right] < \infty$. This of course implies the first moment is finite as well. Given a price series, we can form an arithmetic return series

$$\tilde{R}_t = S_t - S_{t-1}.$$

We can also note that financial time series typically come in OHLC format, condensing a full day of trading into just four numbers.[1] As such we would need to specify which of those points we use to form our returns, and throughout we will use the close price for this. As it turns out, return series are usually rather non-stationary. This is quite undesirable, and to improve the quality of life we wish to standardise the series by its volatility. For this we of course need a volatility proxy, variants of which exist in abundance. We will use the Garman-Klass-Yang-Zhang (GKYG) estimator (Yang and Zhang 2000),

$$\sigma^2_{\text{GKYZ}} = \frac{1}{N}\left[\sum_{i=1}^{N}\left(\ln\left(\frac{o_i}{c_{i-1}}\right)\right)^2 + \frac{1}{2}\sum_{i=1}^{N}\left(\ln\left(\frac{h_i}{l_i}\right)\right)^2 - \sum_{i=1}^{N}(2\ln(2)-1)\left(\ln\left(\frac{c_i}{o_i}\right)\right)^2\right]$$

which we alter into an exponentially weighted moving average estimate defined as

$$\sigma^2_t = (1-\alpha)\sigma^2_{t-1} + \alpha\left[\left(\ln\left(\frac{o_t}{c_{t-1}}\right)\right)^2 + \frac{1}{2}\left(\ln\left(\frac{h_t}{l_t}\right)\right)^2 - (2\ln(2)-1)\left(\ln\left(\frac{c_t}{o_t}\right)\right)^2\right] \tag{2}$$

where $\alpha \in (0,1)$ determines the weight decay – more on this shortly. We finally standardise the returns by

$$R_t = \sigma_t^{-1}\tilde{R}_t.$$

This renders the return series unitless, and rather than being nominal returns, R are the returns measured in standard deviations. By standardising the returns, constraints on the weights in the trade-off problem (1) are translated into constraints on risk allocation. The solution will be the amount of risk allocated in each asset, and this needs to be translated back into nominal positions by considering the risk for each asset. In this project we luckily need not concern ourselves with this, and we never leave the standardised domain. Working with unit variance means the covariance matrix loosely speaking becomes a correlation matrix, in the sense that the diagonal entries should be at least *close* to unity. Having standardised the returns, we are left with estimating the first and second moments

$$\begin{aligned}\mu_t &= \mathbb{E}\left[R_t\right]\\ \Sigma_t &= \mathbb{E}\left[(R_t-\mu_t)(R_t-\mu_t)^\top\right].\end{aligned}$$

2.1 EMA filter

The exponential moving average is a common tool in time series modelling and is typically used as a simple means of building trend followers or smoothing data. For a stochastic process X, we define it as

$$\begin{aligned}\text{EMA}(X_t) &= \alpha X_t + (1-\alpha)\text{EMA}(X_{t-1})\\ \text{EMA}(X_0) &= X_0\end{aligned}$$

[1] Short for open, high, low, and close price.

where the smoothing factor $\alpha \in (0, 1)$ determines how fast the weight placed on older data points decay. Choosing this is typically a trade-off between robustness and responsiveness. Small values of α will have a more significant smoothing effect as new observations have limited contribution to the output. This makes the output less sensitive to outliers but slower to react to changes in the underlying signal. Values close to one naturally give the opposite effect. The EMA filter can just as well be applied to the outer products of a signal to produce a dynamic estimate of the covariance matrix, in this case we replace X_t with $X_t X_t^\top$. The value of the smoothing factor can appear rather arbitrary, and to add some interpretability we define the half-life $\tau_{1/2}$ as the number of lags after which the weight has halved in size, given by

$$\tau_{1/2} = \ln(2) \ln(1 - \alpha).$$

Going forward, any subscript appended to an EMA filter will denote the half-life.

2.2 Shrinkage

Covariance estimates can oftentimes be very noisy in lack on enough data points, especially as dimensions increase. Poor estimations of Σ only become worse upon inversion, something Michaud (1989) coins *error maximisation*. To combat this, a common approach is to adapt the method presented by Ledoit and Wolf (2003). Their idea is to shrink the estimated covariance towards some matrix with more structure, referred to as the target matrix, T. The shrunk covariance estimate is defined as

$$\tilde{\Sigma}_t = (1 - \delta)\hat{\Sigma}_t + \delta T_t \tag{3}$$

where δ is the shrinkage parameter. What T and δ to use constitutes a research topic on its own,[2] and while significant shrinkage could potentially yield desirable qualities in Σ, we will primarily deploy it as a means of ensuring a *not too* ill-conditioned matrix. For this reason, we set the target to a diagonal matrix containing the diagonal elemenets of Σ. That is, $T = \Sigma \circ I$, where I is the identity matrix.

[2]See for instance Ledoit and Wolf (2003), Schäfer and Strimmer (2005), or Jyrkäs (2023).

3 The Wishart Process

We begin by defining the Wishart distribution and introducing the Wishart process. We then consider the Gaussian process and combine our findings in the generalised Wishart process. Finally we formulate the model and show how to make predictions.

3.1 Wishart Distribution

Recall the χ^2 distribution is defined as the sum of squared independent standard Gaussians,

$$\sum_{i=1}^{\nu} z_i^2 \sim \chi^2(\nu).$$

The Wishart distribution is a multivariate generalisation of a gamma distribution, and its definition is a direct extension to that of the χ^2 distribution. Let V be a $D \times D$ positive definite matrix, and $x \sim \mathcal{N}(0, V)$. The sum of ν outer products of x_i is then said to be Wishart distributed with scale matrix V and ν degrees of freedom,

$$\sum_{i=1}^{\nu} x_i x_i^\top \sim \mathcal{W}_D(V, \nu). \tag{4}$$

It is the natural distribution from which sample covariances of multivariate normal distributions take values, and has the density

$$f(\Sigma) = \frac{|\Sigma|^{(\nu-D-1)/2} e^{-\frac{1}{2}\mathrm{tr}(V^{-1}\Sigma)}}{2^{\nu D/2}|V|^{\nu/2}\Gamma(\nu/2)}$$

where Γ is the gamma function. Although the above definition is defined for integer valued ν, this density holds for any real $\nu \geq D+1$, hence the notion of the Wishart distribution being a generalisation of the gamma distribution and not just the χ^2 distribution, which indeed is special case of the former, when ν is integer valued. An alternative but equivalent definition to (4) is to let $LL^\top = V$ be the Cholesky decomposition of the scale matrix V, and consider instead a vector of independent standard Gaussians $z \sim \mathcal{N}(0, I)$. Then,

$$\sum_{i=1}^{\nu} L z_i z_i^\top L^\top \sim \mathcal{W}_D(V, \nu). \tag{5}$$

A Wishart distributed random variable X has expectation $\mathbb{E}[X] = \nu V$ and it is the conjugate prior to the inverse covariance matrix in the multivariate normal distribution (Bishop 2006). This means that if the covariance has an inverse Wishart prior, $p(\Sigma) = \mathcal{W}^{-1}(V, \nu)$ and the data follows a Gaussian likelihood, $p(Y|\Sigma) = \mathcal{N}(0, \Sigma)$, then the posterior is also inverse Wishart distributed, $p(\Sigma|Y) = \mathcal{W}^{-1}(\tilde{V}, \tilde{\nu})$.

3.2 Wishart Process

Cox et al. (1985) introduce the univariate stochastic differential equation

$$dS_t = a(b - S_t)dt + \sigma\sqrt{S_t}dB_t, \;\; S_0 = s_0 \in \mathbb{R} \tag{6}$$

with positive parameters a, b, σ and one-dimensional Brownian motion B_t, and show that for $s_0 \in \mathbb{R}^+$ there always exists a solution, famously referred to as a Cox-Ingersoll-Ross (CIR) process. They also show that for $2ab \geq \sigma^2$ the solution remains positive, making the model a natural choice for interest

rates[3] and thus widely used in bond pricing (Lindström et al. 2015). Pfaffel (2012) considers a matrix variate extension of (6),

$$dS_t = (S_t K + K^\top S_t + \alpha Q^\top Q)dt + \sqrt{S_t} dB_t Q + Q^\top dB_t^\top \sqrt{S_t},\ S_0 = s_0 \in \mathbb{R}^{D\times D} \tag{7}$$

where Q and K are real $D \times D$ matrices, $\alpha \geq 0$ and B_t is a $D \times D$ Brownian motion, and proceeds to demonstrate that for $\alpha \geq D + 1$ and $s_0 \in \mathcal{S}_D^+$ there exists a solution to (7), which is then a Wishart process.[4] Similarly to the CIR process being positive, the Wishart process remains positive definite for all times.

3.3 Gaussian Process

Let $\{X_t : t \in T\}$ be a stochastic process defined on the index set T. If for all finite subsets of indices T_S the vector $\{X_t : t \in T_S\}$ is a multivariate Gaussian, the process is called a Gaussian process (GP). It may help to think of the index set as time, for then the covariance matrix essentially dictates how elements separated by time covary. As often is the case, we will let this time dependence be given by some kernel function $k(t_i, t_j; \theta) = \mathbb{C}\left[X_{t_i}, X_{t_j}\right]$ where θ denotes any set of parameters. The covariance function must of course be positive definite, but as long as this holds there will always exist an associated Gaussian process (Rasmussen and Williams 2006). The squared exponential kernel is common and makes for a good example as we will use it further on as well.[5] We define it as

$$k(t_i, t_j; \ell) = e^{-\frac{|t_i - t_j|^2}{2\ell^2}} \tag{8}$$

where the time scale ℓ is the only parameter. Beyond the necessary symmetry of a covariance, this kernel is also stationary, meaning it only depends on the difference in time, and not the time itself. It is very popular in a number of fields, and proven positive definite (Rasmussen and Williams 2006). We will denote our Gaussian processes $\mathbf{u} = \{u_t\}$. Those are generated by first constructing the covariance matrix $K = (k_{i,j})$, where $k_{i,j} = k(t_i, t_j; \ell)$, and then drawing $\mathbf{u} \sim \mathcal{N}(0, K)$.

3.4 Generalised Wishart Process

Having motivated that the Wishart process is a reasonable candidate for modelling dynamic covariance matrices, we adapt the framework presented in Wilson and Ghahramani (2010). They introduce the generalised Wishart process (GWP), and set up a Markov chain Monte Carlo procedure for Bayesian inference which is largely followed in this paper. The GWP introduces a time dependence in (4) by considering Gaussian processes indexed by some input variable, which for the entirety of this paper will be time. Any collection of variables indexed by t will therefore be referred to as a time series and denoted in bold. It is noteworthy however that this input is arbitrary as far as the framework goes, and so signals such as interest rates and other macroeconomic factors or could be used as well. Let $\mathbf{u}$ be a Gaussian process defined on some index set $\mathbf{t} = \{t_1, \ldots, t_N\}$ by some kernel function k with parameters θ. Then collect $D\nu$ independent GPs in a $D \times \nu \times N$ array. Furthermore, let

$$U_t = \begin{bmatrix} u_{1,1}(t) & \ldots & u_{1,\nu}(t) \\ \vdots & \ddots & \vdots \\ u_{D,1}(t) & \ldots & u_{D,\nu}(t) \end{bmatrix} = [\tilde{u}_1(t), \ldots, \tilde{u}_\nu(t)] \tag{9}$$

[3] Several extensions and variations exist to account for negative interest rates and time varying parameters, see for instance Orlando et al. (2020) or Chan et al. (1992).

[4] The solution is in fact strong, but we need not concern ourselves with the implications of this.

[5] Common aliases also include *radial basis function* kernel, or simply *Gaussian* kernel.

be a slice in time in the collection of GPs. All elements of U_t are independent a priori, and hence $\tilde{u}_i = u_{1:D,i}^\top$ is a vector of independent standard Gaussians. Therefore,

$$\Sigma_t = \sum_{i=1}^{\nu} L\tilde{u}_i(t)\tilde{u}_i(t)^\top L^\top = LU_tU_t^\top L^\top \sim \mathcal{W}_D(V, \nu) \tag{10}$$

where, again $LL^\top = V$. With this construction, $\mathbf{\Sigma} = \{\Sigma_t : t \in \mathbf{t}\}$ is a generalised Wishart process. The construction is identical to (5), and the notion of a process comes from the fact that each Σ_t will be correlated in time by the GPs used to construct them. For convenience, we let the index set be that on which the data is sampled, and the same for all GPs, each of which can have its own kernel function and unique set of parameters. The only condition put on the kernel functions is that $k(t, t) = 1$ in order for each element to have unit variance. This restriction comes with no loss of generality, as any scaling can be absorbed by the scale matrix.

3.5 Model Formulation

The model is finalised by assuming Gaussian data. While a common approach, this is not strictly necessary for the framework to hold, as we will see shortly. If all Gaussian processes have their own covariance function $K_{d,i}$, the model is summarised as

$$\begin{aligned} \mathbf{u}_{d,i} &\sim \mathcal{N}(0, K_{d,i}) \\ \mathbf{U} &\overset{(9)}{=} \{U_t : t \in \mathbf{t}\} \\ Y_t &\sim \mathcal{N}(\mu_t, LU_tU_t^\top L^\top). \end{aligned} \tag{11}$$

Notably, we could also choose to model the precision, Σ^{-1}, as this belongs to the same class of matrices as the covariance, that is, $\mathcal{S}_D^+$. With this parameterisation, (11) becomes

$$Y_t \sim \mathcal{N}(\mu_t, (LU_tU_t^\top L^\top)^{-1}).$$

In many situations the precision matrix is more useful than the covariance matrix, so we will run both variants throughout as part of the exploration. Before entering the details of our Monte Carlo procedure, we constrain the model somewhat.

3.5.1 Kernel Parameters

As previously mentioned, the kernels used have only minor restrictions on them. One might consider using a variety of different kernels in an attempt to better model different types of data. For instance, Wilson and Ghahramani (2010) experiment with a sinusoidal kernel to capture periodic behaviour. To constrict our focus, we settle on the squared exponential kernel defined in (8). Furthermore we let all kernels share the same scale parameter, which greatly reduces complexity as all Gaussian processes then share the same prior distribution, $\mathcal{N}(0, K)$. Even though the kernel we use has only one parameter ℓ, we will stick to the more general notation of θ going forward.

3.5.2 Gaussian Processes

In practice, it turns out the smallest eigenvalues of the floating point representation on K can become zero or even negative zero to working precision, as if $K \notin \mathcal{S}_D^+$. This is however a numerical issue, which Ranjan et al. (2010) find can occur when elements of K come close, and suggest adding a small diagonal matrix in order to lift the eigenvalues slightly. The small addition is often referred to as a *nugget*, and we use the regularised covariance matrix to generate Gaussian processes,

$$\mathbf{u} \sim \mathcal{N}(0, K + \delta I).$$

With the typical interpretation being that there is additive white noise with variance δ added to each element of $\mathbf{u}$. In our context this essentially means that the correlation between times is not exactly fixed, but rather has small random deviations from that of the kernel functions. Technically this regularisation method is a form of ridge shrinkage typically attributed to Hoerl and Kennard (1970).

3.5.3 Scale Matrix

We recall that a Wishart distributed variable $X \sim \mathcal{W}(V, \nu)$ has expectation $\nu V = \nu L L^\top$, and consider a sample covariance matrix estimated over a zero mean time series,

$$\bar{\Sigma} = \frac{1}{N} \sum_{n=1}^{N} Y_n Y_n^\top.$$

It makes intuitive sense for the framework to reproduce this a priori, suggesting

$$\mathbb{E}\left[L U_n U_n^\top L^\top\right] = \nu V = \bar{\Sigma}$$

which we can obtain by choosing

$$L = \nu^{-1/2} \text{chol}(\bar{\Sigma})$$

or, for the inverse case,

$$L = \nu^{-1/2} \text{chol}(\bar{\Sigma}^{-1}).$$

It is worth pointing out that the inversion in the latter case is one of the very things we aim to avoid by using the inverse parameterisation. However, since we expect $\mathbf{U}$ to capture the variations, the scale matrix is more of a rough target as opposed to a crucial component. As such we could most likely apply heavy regularisation to $\bar{\Sigma}$ before inverting it, or even setting it to a unit matrix. In the case where one wishes to sample L, we would replace the expression above by some matrix variate prior.

3.5.4 Degrees of Freedom

A parameterisation of the covariance according to (10) is by no means unique, meaning it could in theory be obtained for any valid choice of $\nu \geq D+1$. For simplicity we keep this fixed at $\nu = D+1$, as it would be structurally challenging to have this changing between samples. One might consider some experimentation on what values work best, but this is out of scope.

3.6 Making Predictions

Assume the Gaussian processes are evaluated on a time grid $\mathbf{t} = \{t_1, \ldots, t_q\}$. While we could easily perform interpolation, predictions would more commonly be made on time points proceeding the in-sample grid. Call the grid on which we want to make predictions $\mathbf{t}^* = \{t_1^*, \ldots, t_p^*\}$. As per definition the new points also follow a multivariate Gaussian distribution,

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{u}^* \end{bmatrix} \sim \mathcal{N}(0, \tilde{K}), \;\; \tilde{K} = \begin{bmatrix} K & A^\top \\ A & K^* \end{bmatrix}$$

where the covariance between points on the in-sample grid and prediction grid are contained in

$$A_i = k(\mathbf{t}, t_i^*), \;\; 1 \leq i \leq p$$

and covariance between the prediction points are contained in

$$K_i^* = k(\mathbf{t}^*, t_i^*), \;\; 1 \leq i \leq p.$$

The marginal distribution for $\mathbf{u}^*$ is found by conditioning on $\mathbf{u}$,

$$\mathbf{u}^*|\mathbf{u} \sim \mathcal{N}(AK^{-1}\mathbf{u}, K^* - AK^{-1}A^\top).$$

This is the density from which we draw the predictions for each GP. The implied predicted covariance matrix is then constructed just like before, $\hat{\Sigma}_t = L\hat{U}_t\hat{U}_t^\top L^\top$. Any regularisation added to K will also be added to K^*.

4 Bayesian Inference

Bayesian statistics is a statistical paradigm revolving around Bayes theorem

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$$

where the ingredients have clear interpretations. The *likelihood* $p(x|\theta)$ is parameterised by whatever model is assumed to relate the parameters θ to the data. When θ has finite dimensionality, we have a parametric model. If however θ has an infinite number of dimensions, the model is called non-parametric. The *prior* $p(\theta)$ reflects beliefs of θ without having observed any data. Having then observed data, $p(\theta|x)$ is the *posterior* density. The denominator $p(x)$ is typically referred to as the *evidence* or *marginal likelihood*, and accounts for the total likelihood of the observed data, typically calculated using the law of total probability,

$$p(x) = \int p(x|\theta)p(\theta)d\theta.$$

This usually acts as a normalising constant, and we commonly express the posterior as being proportional to the likelihood times the prior,

$$p(\theta|x) \propto p(x|\theta)p(\theta). \tag{12}$$

The *posterior predictive* is the density of unobserved data x^* marginalised over the posterior

$$p(x^*|x, \theta) = \int p(x^*|\theta)p(\theta|x)d\theta$$

and this is typically used to approximate predictions. The Bayesian philosophy is often contrasted by the *frequentist* counterpart, which instead interprets the idea of probability as the limit of relative occurrences after many trials, that is, the frequency of an event after arbitrary many samples. While arguably two sides of the same coin, both carrying intuitive reason, the idea of making a guess and then updating it as data presents itself is often a Bayesian inducement. Further reading on Bayesian inference as well as comparisons to the frequentist counterpart can be found in Zhang (2021) among others.

Since their introduction, Monte Carlo methods have been a staple tool for Bayesian inference, and one we will deploy as well. Alternative approaches include variational inference, which very loosely speaking revolves around finding a density that approximates the true posterior, and then finding an optimum to this, usually in a maximum likelihood meaning. Heaukulani and van der Wilk (2019) utilises variational inference in a setting very similar to ours.

5 Markov Chain Monte Carlo

Markov chain Monte Carlo was introduced by Metropolis et al. (1953) and is undoubtedly one of the greatest feats in numerical statistics. We first account for the Markov chain and a small but important selection of associated properties, from which we motivate the method. Then we define the customary Metropolis-Hastings algorithm and Gibbs sampler before describing the more intricate algorithms we deploy.

5.1 Markov Chain

A discrete-time Markov chain (MC) is a stochastic process for which evolution between states only depends on the closest previous one. Formally we define it as any stochastic process $\{X_0, X_1, \dots\}$ taking values in $\mathcal{X}$ that satisfies the Markovian property

$$p(X_{t+1}|X_t, \dots, X_1) = p(X_{t+1}|X_t).$$

Here, we call $p(X_{t+1}|X_t)$ the transition density, supported on $\mathcal{X}$. A distribution π is said to be *stationary* if it satisfies the *global balance* equation

$$\int p(x|z)\pi(z)dz = \pi(x) \ \forall \ x, z \in \mathcal{X}$$

The concept of stationarity is central to Markov chains, and the name is motivated by the fact that any MC that starts in its stationary distribution will never leave it. Furthermore, we also consider the *local balance* equation

$$\pi(x)p(z|x) = \pi(z)p(x|z) \ \forall \ x, z \in \mathcal{X}. \tag{13}$$

which intuitively describes a balance in *flow* between two states x and z. Any distribution satisfying local balance also satisfies global balance and is hence also stationary. Integrating (13) with respect to z quickly proves this. The last property of Markov chains we need to present before moving on is *ergodicity*. Trying not to fall down the MC rabbit hole, a Markov chain that will converge to its stationary distribution regardless of the initial distribution is said to be ergodic.[6] It will become clear shortly why this property is crucial, and formally it holds that for a *geometrically ergodic* Markov chain, there exists a $\rho < 1$ such that for any initial distribution p_0,

$$\sup_{A \subseteq \mathcal{X}} |\mathbb{P}(X_n \in A) - \pi(A)| \leq \rho^n$$

where π is the stationary distribution.

5.2 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) was pioneered by the ground breaking paper of Metropolis et al. (1953). The idea is that we can emulate sampling from a distribution by creating a Markov chain with the same stationary distribution as that which we want to sample from. This allows us to sample from distributions with no closed form, high dimensions, or otherwise impractical in conjunction with other sampling methods. For a geometrically ergodic Markov chain with stationary distribution π, there exists a law of large numbers that states convergence in probability. Let $\{X_n\}$ be a geometrically ergodic Markov chain with stationary distribution π. Then, for all $\varepsilon > 0$ and arbitrary functions f,

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{k=1}^{N} f(X_k) - \int f(x)\pi(x)dx\right| \geq \varepsilon\right) \to 0 \text{ as } N \to \infty.$$

[6] In short, any irreducible, positively recurrent and aperiodic Markov chain is ergodic. See for example Robert and Casella (1999) for further reading.

We omit the proof, which is otherwise readily available in literature on MCMC, see for instance Robert and Casella (1999). This result enables us to, for instance, estimate expectations by simulating the Markov chain and evaluating the function at each state,

$$\frac{1}{N}\sum_{k=1}^{N} f(X_k) \to \mathbb{E}\left[f(x)\right] \text{ as } N \to \infty. \tag{14}$$

5.3 Metropolis Hastings

What was long known as just the Metropolis algorithm presented by Metropolis et al. (1953) was extended by Hastings (1970), and later became known as the Metropolis-Hastings (MH) algorithm. It describes how to create a Markov chain with the same stationary distribution as that which we want to sample from, which we will call $p(x)$. We define a proposal density $q(x^*|x)$ from which we can easily sample, called the *proposal kernel*. Assuming the current state is X_k, we then draw a proposal from q, and assign this a probability of acceptance which we define as

$$\alpha = \min\left\{1, \frac{p(x^*)}{p(x)}\frac{q(x|x^*)}{q(x^*|x)}\right\}$$

where the second term is known as the Hastings ratio. Lastly, we accept the proposal with probability α and set $X_{k+1} = x^*$. Should α not be accepted, we let $X_{k+1} = X_k$. The terms give a balance between likelihood and exploration, and increases a proposals probability of acceptance if it has higher likelihood, but decreases it according to how likely that proposal was compared to how likely the opposite transition would be. The Metropolis-Hastings algorithm has a stationary distribution that coincides with our target distribution $p(x)$, which is fundamental. Furthermore it can be shown to be geometrically ergodic, which is critical in order to ensure convergence according to (14) (Robert and Casella 1999). The proposal kernel can be chosen in a few different ways. A kernel is said to be symmetric if it satisfies $q(x|z) = q(z|x)$, and the acceptance probability subsequently reduces to

$$\alpha = \min\left\{1, \frac{p(x^*)}{p(x)}\right\}.$$

We note that because we only evaluate the ratio $p(x^*)/p(x)$, we only need to know p up to a normalising constant, which is exactly what we assume in Bayesian inference (12). The algorithm is given in pseudo code in algorithm 1.

5.4 Gibbs Sampler

Assume we have a multidimensional space $\mathcal{X}$ which we can divide in blocks, and let $x = \{x^1, \ldots, x^m\} \in \mathcal{X}$. If we want to sample from a multivariate density $p(x)$ on $\mathcal{X}$, we can utilise the Gibbs sampler. Denote by $p(x^i|x^{-i})$ the marginal density for x^i given all the other components $x^{-i} = \{x^k\}_{k\neq i}$. The Gibbs sampler updates each block by drawing $x^i \sim p_i(x^i|x^{-i})$ in cycles, thereby simulating a Markov chain on $\mathcal{X}$. The cycle is outlined very briefly in algorithm 2. Much like the Metropolis-Hastings algorithm, the Gibbs sampler has a stationary distribution and can be proven geometrically ergodic (Robert and Casella 1999). With the reduced problem at hand, we are left to sample in cycles from

$$p(\theta|\mathbf{Y}, \mathbf{U}) \propto p(\mathbf{U}|\theta)p(\theta)$$
$$p(\mathbf{U}|\mathbf{Y}, \theta) \propto p(\mathbf{Y}|\mathbf{U})p(\mathbf{U}|\theta).$$

We sample from the former density simply using Metropolis-Hastings with a uniform proposal kernel, meaning

$$q(\theta_{t+1}|\theta_t) = \theta_t + \mathcal{U}(-\rho, \rho).$$

The latter density is more intricate, and to sample the Gaussian processes we will run two separate methods in parallel. We deploy the elliptical slice sampler used by Wilson and Ghahramani (2010), as well as a Hamiltonian Monte Carlo algorithm which we manually tailor to our model.

5.5 Elliptical Slice Sampling

Wilson and Ghahramani (2010) make use of the elliptical slice sampler (ESS) introduced by Murray et al. (2010) and designed specifically to sample from a distribution that is proportional to a Gaussian prior and some likelihood that ties the parameters to the data. This is exactly what we want to sample from,

$$p(\mathbf{U}|\mathbf{Y},\theta) \propto p(\mathbf{U}|\theta)p(\mathbf{Y}|\mathbf{U}) = \prod_{d=1}^{D}\prod_{i=1}^{\nu}\mathcal{N}(U_{d,i};0,K_{d,i})p(\mathbf{Y}|\mathbf{U}).$$

The likelihood function

$$p(\mathbf{Y}|\mathbf{U}) = \prod_{n=1}^{N}\mathcal{N}(Y_n;0,LU_nU_n^\top L^\top),$$

while not central to the algorithm, is used in validating the proposals in light of the observed data. In the algorithm, the Gaussian prior is used to sample an auxiliary variable which defines a high-dimensional ellipse on the density of $\mathbf{U}$, hence the name. A random proposal is drawn from this ellipse, parameterised by an angle φ. Should the proposal not be accepted, the angle is adjusted until so. This means the algorithm has an internal rejection process, but always outputs a new state. The ESS algorithm is briefly presented in algorithm 3 below, and the interested reader is referred to the original paper (Murray et al. 2010) for further details.

Algorithm 1 Metropolis-Hastings

Input: $\theta_0, p(U|\theta), q(\theta'|\theta)$
Output: θ_1
  $\theta^* \sim q(\theta|\theta_0)$
  $\alpha \leftarrow \min\left\{1, \frac{p(U|\theta^*)}{p(U|\theta_0)} \cdot \frac{q(\theta_0|\theta^*)}{q(\theta^*|\theta_0))}\right\}$
  $u \sim \mathcal{U}(0,1)$
  if $u \leq \alpha$ **then**
    $\theta_1 \leftarrow \theta^*$
  else
    $\theta_1 \leftarrow \theta_0$
  end if

Algorithm 2 Gibbs Sampler

Input: $\{X_k\}$
  $X_{k+1}^1 \sim p_1(x^1|X_k^2,\ldots,X_k^m)$
  $X_{k+1}^2 \sim p_2(x^2|X_{k+1}^1,X_k^3,\ldots,X_k^m)$
  $\vdots$
  $X_{k+1}^m \sim p_m(x^m|X_{k+1}^1,\ldots,X_k^{m-1})$

Algorithm 3 Elliptical Slice Sampling

Input: $\mathbf{U}_0$, $\{K_{d,i}\}$, $p(\mathbf{Y}|\mathbf{U})$
Output: $\mathbf{U}_1$
  $V_{d,i} \sim \mathcal{N}(0,K_{d,i})$
  $\mathbf{V} \leftarrow \{V_{d,i}\}$
  $u \sim \mathcal{U}(0,1)$
  $\log y \leftarrow \log p(\mathbf{Y}|\mathbf{U}_0) + \log(u)$
  $\varphi \sim \mathcal{U}(0,2\pi)$
  $\varphi_{min} \leftarrow \varphi - 2\pi$
  $\varphi_{max} \leftarrow \varphi$
  $\mathbf{U}^* \leftarrow \mathbf{U}\cos(\varphi) + \mathbf{V}\sin(\varphi)$
  while $\log p(\mathbf{Y}|\mathbf{U}^*) < \log y$ **do**
    if $\varphi < 0$ **then**
      $\varphi_{min} \leftarrow \varphi$
    else
      $\varphi_{max} \leftarrow \varphi$
    end if
    $\varphi \sim \mathcal{U}(\varphi_{min},\varphi_{max})$
    $\mathbf{U}^* \leftarrow \mathbf{U}\cos(\varphi) + \mathbf{V}\sin(\varphi)$
  end while

5.6 Hamiltonian Monte Carlo

While the ESS algorithm is fast, it is also somewhat inefficient in the sense that proposals are produced somewhat randomly. To extend the work done by Wilson and Ghahramani (2010), we set up an alternative algorithm used to sample the GPs as an attempt to improve sampling efficiency. Acknowledging a few prior works, Neal (2011) presents a Monte Carlo algorithm designed to accelerate state exploration when compared to methods using random walk proposals. The method utilises an auxiliary *momentum* variable, and computes a proposal by letting the system evolve under Hamiltonian dynamics,[7] motivating the name. This model will carry the abbreviation HMC and work as a drop-in alternative to ESS. The auxiliary variable $\mathbf{\Omega}$ has the same exact form as $\mathbf{U}$, and the Hamiltonian is defined as

$$\begin{aligned} H(\mathbf{\Omega}, \mathbf{U}) &= -\log p(\mathbf{\Omega}, \mathbf{U}) \\ &= -\log p(\mathbf{\Omega}|\mathbf{U}) - \log p(\mathbf{U}) \\ &= T(\mathbf{\Omega}) + V(\mathbf{U}) \end{aligned}$$

where $T(\mathbf{\Omega})$ and $V(\mathbf{U})$ are referred to as *kinetic energy* and *potential energy* respectively. Furthermore, the Hamiltonian dynamics are

$$\begin{aligned} \frac{d\mathbf{U}}{dt} &= \nabla T \\ \frac{d\mathbf{\Omega}}{dt} &= -\nabla V. \end{aligned}$$

This system is typically integrated by the leapfrog method, which we will use as well and introduce shortly. After letting the system evolve over some time period, the state reached will act as the proposal. This then undergoes a Metropolis acceptance step, concluding the method. We outline this in algorithm 4. With a large enough integration time, the proposal can be vastly different from the initial state while still having a significant probability of acceptance. As mentioned, this should bypass the slow exploration associated with random walks and, hopefully, give an improvement in efficiency over ESS. The momentum is typically chosen as independent univariate Gaussians,

$$\begin{aligned} \Omega_{d,i} &\sim \mathcal{N}(0, M) \\ M &= mI. \end{aligned}$$

Probably the most critical step in the HMC algorithm is finding the gradients of the momentum and potential. With Gaussian momentum this gradient is trivial,

$$\nabla T = m^{-1}\mathbf{\Omega},$$

but for the potential we have to derive it ourselves.

5.6.1 Potential Gradients

The potential can be split into likelihood and prior,

$$V(\mathbf{U}) = -\log p(\mathbf{Y}|\mathbf{U}) - \log p(\mathbf{U}) = V_1 + V_2$$

and the gradient can thus be expressed as

$$\nabla V_{d,i,n} = \frac{\partial V_1}{\partial U_n}\frac{\partial U_n}{\partial u_{d,i,n}} + \frac{\partial V_2}{\partial U_{d,i}}\frac{\partial U_{d,i}}{\partial u_{d,i,n}}.$$

[7] Introduced by Hamilton (1833), oftentimes interpreted as a connection between classical and quantum mechanics.

Here it is enough to find the first part of each term on matrix form, as the second parts are essentially just indicator functions to single out individual elements. The matrices are then stacked appropriately to construct a matrix of the same dimensions as $\mathbf{U}$ containing the element wise derivatives. We present the full derivations in appendix B.2-B.3 and skip straight to the results here. Denoting by U^+ the pseudo-inverse of U, for the regular parameterisation we get

$$\frac{\partial V_1}{\partial U_n} = (U_n^+)^\top - \left(U_n^+ A (U_n U_n^\top)^{-1}\right)^\top$$

where $A = L^{-1} Y Y^\top L^{-\top}$, and

$$\frac{\partial V_2}{\partial U_{d,i}} = K_{d,i}^{-1} U_{d,i}.$$

For the inverse parameterisation the first term of is slightly different, and becomes

$$\frac{\partial V_1}{\partial U_n} = -(U_n^+)^\top + B U_n$$

having used $B = L Y Y^\top L^\top$. The contribution from the second term V_2 is unchanged, and the entries are inserted into the full gradient separately just like before.

5.6.2 Leapfrog Integrator

The leapfrog integrator is a second order numerical integration method used to solve dynamical systems on particular forms. It frequently accompanies Hamiltonian Monte Carlo methods, and we will account for the necessary basics which can also be found in Neal (2011). The method can be used for a wider range of problems, and further details are readily available in literature on numerical integration. Probably the most common approach to numerically solving differential equations would be to discretise time on a grid $\{t_0, t_0 + \varepsilon, \dots, t_0 + T\varepsilon\}$ and then making steps at those points according to whatever numerical approximation is used. The leapfrog instead updates one of its variables on the aforementioned grid, and the other variables on a grid shifted by half a time step, $\{t_0 + \varepsilon/2, t_0 + 3\varepsilon/2, \dots\}$. After making an initial half-step for one of the variables, steps are iterated on the two grids such that they *leapfrog* over each other. Compared to, say, an Euler integration scheme, this achieves much better approximations with the same number of evaluations per step, and can handle periodic problems.

The choice of ε usually comes down to the tolerance of the gradient. The Hamiltonian needs to be stable, and the integration step has to be small enough to allow this. Finding this limit requires some pre-testing and tuning, and after finding it T is set to $b\varepsilon^{-1}$ where $b \approx 2$ is a popular ballpark. Increasing this value will let the system evolve over a longer time, producing less correlated samples. This is of course desirable, but a computationally heavy gradient stipulates a trade-off between sample efficiency and sampling time. Finally, we can rid any possible periodicity by randomising the exact number of integration steps performed. The algorithm is presented in algorithm 5 below.

Algorithm 4 Hamiltonian Monte Carlo

Input: $\mathbf{U}_0$, $H(\mathbf{\Omega}, \mathbf{U})$
Output: $\mathbf{U}_1$
  $\Omega_{d,i} \sim \mathcal{N}(0, mI)$
  $\mathbf{\Omega}_0 \leftarrow \{\Omega_{d,i}\}$
  $\mathbf{U}^*, \mathbf{\Omega}^* \leftarrow \text{leapfrog}(\mathbf{U}_0, \mathbf{\Omega}_0, \varepsilon, T)$
  $u \sim \mathcal{U}(0, 1)$
  if $u < \min\{1, \exp\{H(\mathbf{\Omega}_0, \mathbf{U}_0) - H(\mathbf{\Omega}^*, \mathbf{U}^*)\}\}$
  then
    $\mathbf{U}_1 \leftarrow \mathbf{U}^*$
  else
    $\mathbf{U}_1 \leftarrow \mathbf{U}_0$
  end if

Algorithm 5 Leapfrog Integrator

Input: $\mathbf{U}$, $\mathbf{\Omega}$, ε, T
Output: $\mathbf{U}^*, \mathbf{\Omega}^*$
  $T \leftarrow T + \text{floor}(\mathcal{N}(0, kT))$
  $\mathbf{\Omega} \leftarrow \mathbf{\Omega} - \frac{\varepsilon}{2}\text{dVdU}(\mathbf{U})$
  for t = 1:T-1 **do**
    $\mathbf{U} \leftarrow \mathbf{U} + \frac{\varepsilon}{m}\mathbf{\Omega}$
    $\mathbf{\Omega} \leftarrow \mathbf{\Omega} - \varepsilon\text{dVdU}(\mathbf{U})$
  end for
  $\mathbf{U}^* \leftarrow \mathbf{U} + \frac{\varepsilon}{m}\mathbf{\Omega}$
  $\mathbf{\Omega}^* \leftarrow \mathbf{\Omega} - \frac{\varepsilon}{2}\text{dVdU}(\mathbf{U})$

6 Experiments

Before applying a model to real data sets, it is standard practice to first test it on synthetic ones. Recalling the assumption of Gaussian data, it is important to remember this is merely a model intended to describe the data well enough. In reality we only observe Y_t without the access to a *true* covariance. This adds extra incentive to consider simulated data, as this allows us to actually have a ground truth to benchmark against. We do this by constructing an artificial series of covariance matrices $\mathbf{\Sigma}$, which we will sometimes refer to as the *generator*, and then drawing data points $\mathbf{Y} = \{Y_t\}$, $Y_t \sim \mathcal{N}(0, \Sigma_t)$ from this. As the objective is to model the covariance, we the let expectation be zero for all simulated data, effectively simulating the expectation being subtracted from each point in time.

We first define our synthetic datasets and suggest a means of assessing the sampling. For the real data we introduce the futures contract and define the portfolio selection before developing a set of evaluation metrics. Lastly, we discuss sampling efficiency and how we measure this.

6.1 Synthetic Datasets

We draw two sets from time-invariant correlations, one of which have zero cross-correlation. Those act as basic but important tests of the framework as a covariance estimator in the most general sense. Abbreviations $\mathbf{\Sigma}^{\text{ID}}$ and $\mathbf{\Sigma}^{\text{CC}}$ will refer to those sets. Joining are then two shock-like structures. The first is a piece-wise constant correlation structure like the previous ones, but with a discontinuous *shift* between different correlation terms $\rho_{t<t_0}$ and $\rho_{t\geq t_0}$. Secondly, we have a Gaussian *shock* where correlation becomes large and positive, with the volatility simultaneously increasing. This is a commonly observed phenomena during times of financial crisis (Sandoval and Franca 2012). Those sets of underlying covariance structure will be denoted $\mathbf{\Sigma}^{\text{RS}}$ and $\mathbf{\Sigma}^{\text{CR}}$ respectively. We also construct a dataset with sinusoidal covariance $\rho_t = a\sin(kt)$, to investigate capturing periodic behavior. As discussed before one would consider adding a periodic kernel if periodicity is expected, but given we have restricted ourselves to the Gaussian kernel it serves as a good assessment on the implications of this choice. Lastly, we generate a covariance series using our GWP framework. This probably bears some, at least qualitatively speaking, resemblance to the return series data we later consider. We denote this $\mathbf{\Sigma}^{\text{GWP}}$. The covariance structures used to generate synthetic data will be present in all related plots.

6.2 Assessment on Synthetic Data

Evaluating posterior sampling is somewhat of a constant quandary in Bayesian inference when there is no closed form solution at hand. With access to the ground truth, we first introduce a selection of matrix norms that serve as metrics for the sampled covariance. The Frobenius norm summarises the squared error on each element

$$||A||_F^2 = \sum_i \sum_j |a_{ij}|^2 = \text{tr}(A^\top A).$$

As the covariances are symmetric, this will essentially put twice the emphasis on the off-diagonal elements. We can account for this by defining a *lower* Frobenius norm,

$$||A||_{LF}^2 = ||\text{vech}(A)||_2^2 = \sum_i \sum_{j\leq i} |a_{ij}|^2$$

which only counts each unique element once by the half-vectorisation. The operator norm describes the maximum stretch which a projection performs on a vector of unit length, and in the context of a covariance matrix it could be loosely interpreted as the largest standard deviation that a covariance matrix produces. It is defined as

$$\|A\|_{OP} = \sigma_{max}(A)$$

where $\sigma(A)$ are the singular values of A. Both the Frobenius and operator norm are accounted for in most literature on matrix theory, including (Holst and Ufnarovski 2014). To compare the sampled covariances with the underlying, we insert in all the definitions above $A = \Sigma_t^j - \Sigma_t$ as the difference between any sampled covariance and that from which we generate data, Σ_t.

6.3 Real Data Sets

We restrict ourselves to a very limited set of assets that hopefully give a rough representation of their respective asset class and how they are correlated.[8] The return series are derived from prices of future contracts on S&P 500, the American 10 year note, the Yen/Dollar pair, and corn, all standardised by GKYZ volatility (2).

6.3.1 Futures Contract

The futures contract is an agreement between two parties to exchange an underlying asset, at an agreed price. It it similar to the forward contracts, which is possibly more well known, but differs primarily on two points. Firstly, it is traded on an exchange as opposed to OTC, and the liquidity is typically good. Secondly it settles daily, meaning every day money is transferred between parties according to the difference between quote and settle price. Entering a futures contract requires posting a margin account from which the daily settlements are drawn. This margin is typically very small relative to the settle price, and we can think of the contract as effectively free to enter.

Let $F(t, T; S)$ denote the price of a non-dividend paying futures contract on day t, with delivery on day T and written on the underlying S, and r the continuously compounded risk free rate. Then, $F(t, T; S) = S_t e^{r(T-t)}$. Although showing this is no more than a quick exercises in arbitrage pricing, the general case is somewhat more complicated, and we purposely avoid this detour. Arbitrage theory as a whole lies just outside the perimeter of this project, but a critically acclaimed introduction is given by Björk (2019). Because the risk-free rate is accounted for in the futures price, we get the excess returns by default when forming our return series. As such we need not adjust for this manually, as we might for other instruments.

6.3.2 Portfolio Construction

Say we observe data up until and including day t. On this day we can make any predictions of the future returns and covariance, however far ahead into the future as we see fit. On day $t + 1$ we would enter those positions with whatever order strategies we have to minimise market impact and other potential effects. Then we would get returns on day $t + 2$. If we make predictions and take positions on the same day, we would effectively use at-the-time unobserved information and get a look-ahead bias. To avoid this, we use predictions of μ and Σ made two days prior. In other words, we use

$$\hat{\mu}_t = \hat{\mu}_{t|t-2}$$
$$\hat{\Sigma}_t = \hat{\Sigma}_{t|t-2}.$$

Whereas predictions for our framework are described in 3.6, the EMA filters use what is commonly referred to as the naive predictor

$$\hat{\Sigma}_{t|t-k} = \hat{\Sigma}_{t-k}^{\text{EMA}}$$

[8] Asset classes being equities, fixed income, FX, and commodities.

with $k = 2$ in our case. This is the same as saying the current estimate is the best prediction for the covariance k days is the future. With a unit target volatility, we choose our weights as

$$w_t = \frac{\hat{\Sigma}_t^{-1}\hat{\mu}}{\sqrt{\hat{\mu}_t^\top \hat{\Sigma}_t^{-1}\hat{\mu}_t}}.$$

6.4 Assessment on Real Data

There exists a vast number of metrics used to evaluate the performance of a portfolio. Although motivated by the difficulty of constructing portfolios, this project focuses on the covariance modelling in a slightly more general sense. As such, the portfolio metrics are motivated under the loose assumption that a better estimate of Σ should yield better performing portfolios. We therefore emphasise that modelling a covariance and creating a desirable trading strategy are two related but not synonymous problems. We have also omitted transaction costs and other possible frictions, which is important to keep in mind.

6.4.1 Sharpe Ratio

The Sharpe ratio (SR) is one of the most common metrics for portfolios. It was introduced by Sharpe (1966) and relates a portfolios excess returns to its volatility. Let X denote the excess returns of a portfolio. Then the Sharpe ratio is defined as

$$\mathrm{SR} = \frac{\mathbb{E}[X]}{\sqrt{\mathbb{V}[X]}}$$

where we estimate the variance by

$$\hat{\mathbf{V}}[X] = \frac{1}{N-1}\sum_{t=1}^{N}(X_t - \bar{X})^2. \tag{15}$$

As mentioned in the introduction of the futures contract, we can treat the risk free rate as being zero, as the daily settlement effectively includes this in the price. This means that excess returns and just returns are the same. The Sharpe ratio is oftentimes annualised to extrapolate it to a yearly basis, which we do by multiplying by a factor of $\sqrt{252}$.[9]

6.4.2 Value at Risk

Value at Risk (VaR) is a common tool for measuring risk during some time period, specifically how large losses can be expected with a given probability. It is is a nonconstructive metric and as such there exist several alternative definitions. We define it as

$$\mathrm{VaR}_\alpha = \inf\{x : F_X(x) \geq \alpha\} = F_X^{-1}(\alpha)$$

where F_X is the cumulative distribution of portfolio returns. In other words, the VaR for a given level α, is the loss which is matched or exceeded with probability α. There are a few different approaches to estimating this distribution, and we choose to look at the empirical distribution as opposed to, for example, fitting a normal distribution to the historical returns.

[9] A year has on average 252 trading days.

6.4.3 Expected Shortfall

As a complement to VaR, expected shortfall (ES) is the expected loss given a tail event. Formally we define it as

$$\mathrm{ES}_\alpha = \mathbb{E}\left[X | X \leq \mathrm{VaR}_\alpha\right] = \frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_s ds$$

For an empirical distribution this becomes the average over all returns such that $X \leq \mathrm{VaR}_\alpha$.

6.4.4 Skewness and Kurtosis

Skewness and kurtosis are the third and fourth standardised moments for a distribution and indicate primarily how the tails of a distribution behave. A negative skewness means the distribution has a longer left tail, and vice versa. We define and approximate it by

$$\mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] \approx \frac{1}{N}\sum_i \left(\frac{X_i - \bar{X}}{\sigma}\right)^3$$

where $\hat{\sigma}^2 = \hat{\mathbb{V}}[X]$ as defined in (15). Similarly for the kurtosis,

$$\mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] \approx \frac{1}{N}\sum_i \left(\frac{X_i - \bar{X}}{\sigma}\right)^4.$$

For a financial return series, negative skewness indicates a heavier negative tail, which is why larger values are desirable. As short tails means less risk, we conversely want kurtosis to be small.

6.5 Assessing Sampling Efficiency

To different degrees, MCMC algorithms typically suffer from samples being correlated as a result of the current state being used to propose the subsequent one. This can influence how well the sample distribution resembles the desired density, especially for smaller sample sizes. A common solution is to create a sub-sample of every n_0:th sample which will then be a factor n_0^{-1} as large. The reasoning is that autocorrelation drops as lags between two samples increase, and for a large enough n_0 the samples can be regarded as effectively independent. Thus for a desired number of efficient samples, the *raw* sample size might need to be significantly larger. It is possible to produce even better sub-samples by randomising the indices, as this also rids any periodicity of the MC.

We introduced the HMC sampling as a means to increase efficiency and improve convergence compared to the ESS, and so naturally we are interested in to what degree this was successful. As we are sampling a matrix rather than one dimensional parameters, we need a proxy for reducing the sample matrices to a single value which we can use to assess the sampling efficiency. We already discussed a selection of matrix norms, and those we can re-purpose here. Assume we have a set of samples $\mathbf{\Sigma}^j$ and fixate an arbitrary time point t. We choose the Frobenius norm and define

$$f^j = ||\Sigma_t^j - \Sigma_t||_F$$

where Σ_t is the generator. The idea is that similar matrices likely have similar norms, and so if the samples are efficient, f^j should have low autocorrelation between lags. Importantly, the implication between similar matrices and similar norms only goes one way meaning this approach, while probably sufficient, is not completely rigorous.

7 Results

Through the variants of the framework we have collected a total of four models. We name them by the abbreviation akin to the method used to sample the Gaussian processes, and add a suffix '-I' when the inverse parameterisation is used. Hence we have ESS, ESSI, HMC and HMCI. Our benchmark is the EMA filtered sample covariance, which we will simply denote EMA. Henceforth we will not distinguish between the sampling algorithms and models using them, but the context should make it clear which is meant. In the interest of brevity we typically only show a selection of models and results that sufficiently highlight the focus points. Supplementary plots are found in appendix A.

7.1 Sampling Efficiency

To denote an equidistant sub-sample, we let $\mathbf{f}^{n_0} = \{f^{n_0}, f^{2n_0}, \dots\}$. Figure 1 shows the autocorrelation for subsequent samples using both ESS and HMC. It is clear that the latter produces less correlated samples, which is what we expect. Not only does HMC display lower autocorrelation already, it can be adjusted manually to give even less correlated samples by simply increasing the integration time. This adds some freedom in adjusting the model, whereas the ESS is essentially uncontrollable. In practice we randomise the indices from which we draw our sub-samples, but the analysis above acts as a demonstration of the differences in efficiency, as well as providing a basis for how large the raw sample size needs to be. It should be noted that $\mathbf{f}^{n_0}$ of course varies between different time points t and choice of norm, which prohibits us from making exact statements of how much improvement HMC brings even at fixed integration times. Some investigation shows that the samples are at least somewhere between 5-10 times as efficient in this sense, as $\mathbf{f}^5_{\text{HMC}}$ has similar ACF to $\mathbf{f}^{40}_{\text{ESS}}$. We provide an example in figure 2 below.

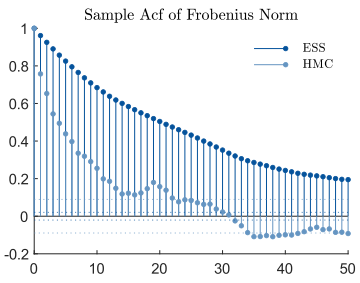

Figure 1: Autocorrelation function for Frobenius norm of subsequent samples.

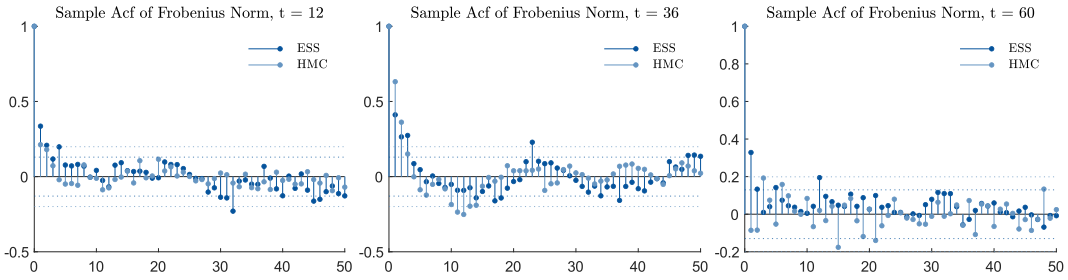

Figure 2: Autocorrelation function of $\mathbf{f}^{40}_{\text{ESS}}$ and $\mathbf{f}^5_{\text{HMC}}$, with $t = 12, 36, 60$. While varying somewhat with t, the two could be deemed somewhat similar overall.

7.2 Computational Time

Although HMC produces much less correlated samples, it requires more computational time to do so. As it turns out, the leapfrog integrator needs to make fairly small steps in order for the high-dimensional log-likelihood to remain stable, and as such many integration steps are needed. This in turns means the likelihood needs to be evaluated many times, which quickly adds up and makes the algorithm sluggish. Although no extensive investigation is made on exactly how the time scales with

different dimensions, at the scales of this project the extra computational time unfortunately outweighs the increase in efficiency. We could however speculate that as the number of assets increases, the already low acceptance rate within the elliptical sampler will likely diminish even further due to the stochastic nature of the proposals. This could mean HMC does have a place in the framework in certain conditions.

In a live scenario one data point would be added per day and our Gibbs algorithm could be given plenty of time to converge and produce a desirable sample size. The choice of sampling could come down to how independent we like our samples to be, or how the algorithms seem to perform with the dimensions of a given use case. Regardless, simulating this procedure being executed on years of data means we must limit the number of iterations we allow for each additional data point.

7.3 In-Sample Performance on Synthetic Data

Before moving on to making predictions, we want to make sure the framework captures in-sample covariance in a satisfactory manner. For constant covariance we expect the scale parameter to grow, giving essentially straight lines for U. This is exactly what happens, as can be see in figure 3 which shows a trace plot of θ for the data with constant correlation. The marginal posterior distribution for θ flattens, and samples cover a large region of values. In other words, as long as a proposed scale parameter is large enough to generate a virtually flat GP, the exact value is insignificant. The same plot also indicates a difference in exploration between the two sampling algorithms. The Hamiltonian sampler seems to allow for more rapid changes in θ, whereas the elliptical counterpart moves slower. Differences in exploration and computation aside, both models successfully find a distribution resembling that from which the data is generated. Figure 4 below depicts samples on $\mathbf{\Sigma}^{\text{CC}}$ using the elliptical model, and as the corresponding plots for $\mathbf{\Sigma}^{\text{ID}}$ look essentially identical, we leave them in appendix A.1.

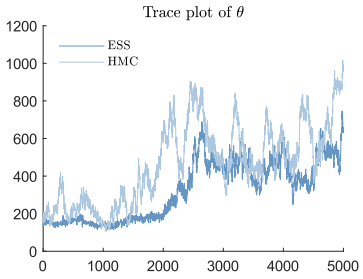

Figure 3: Trace plot of θ.

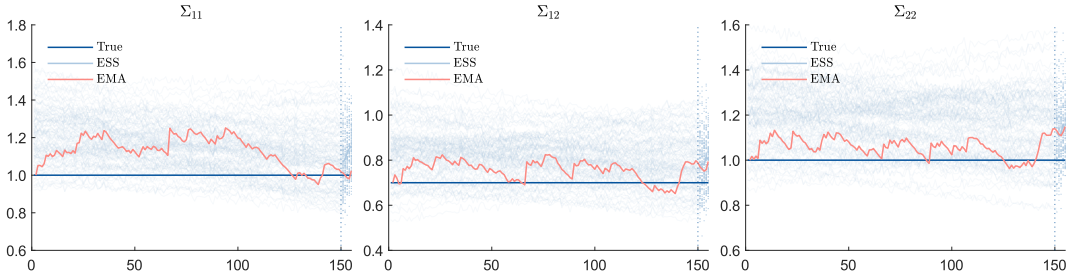

Figure 4: Samples of $||\Sigma_t^j - \Sigma_t^{\text{CC}}||_F$ using ESS, accompanied by generator (dark blue) and EMA (red).

The shock-like datasets challenges the framework in that the parameters quickly changes. The continuity of the Gaussian processes prohibits the model from capturing the sudden jump of the regime shift, and it responds by something reminiscent of a sinusoidal approximation of a box function. The reaction is similar for the Gaussian shock, where the peak is not fully captured, and the flat sections gets overcompensated. This is the model trying to balance the scale parameter, which otherwise would be large for the edges but small for the peak. As the covariance structure for the latter set is perhaps not obvious, we show the samples for $\mathbf{\Sigma}^{\text{CR}}$ using ESS in figure 5 below.

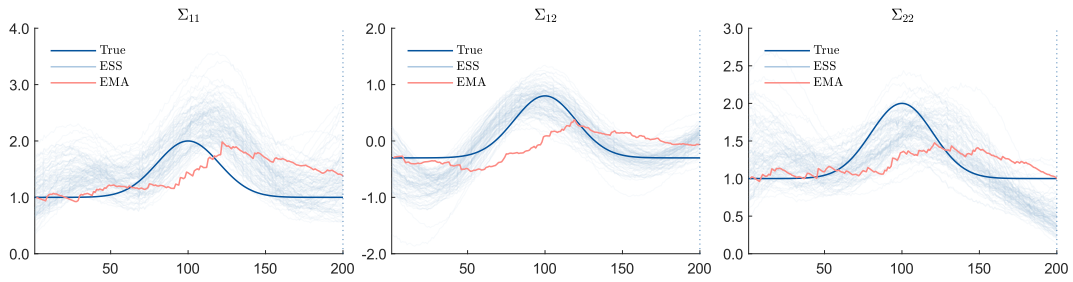

Figure 5: Samples of $||\Sigma_t^j - \Sigma_t^{\text{CR}}||_F$ using ESS, accompanied by the generator (dark blue) and EMA filter (red).

7.4 Predictive Performance on Synthetic Data

Progressing from constant to dynamic covariance, we redirect our focus to the data generated from a GWP, as well that with periodic correlation structure. The first half of each time series serves as modelling data on which we run inference until convergence. Predictions are made and new data points are added sequentially from the validation data, being the second half of each time series. We display a selection of sample plots below, that highlight the advantage of a Bayesian approach. Rather than a single estimate, we get a cloud of samples which illustrates the posterior distribution of covariances having observed the data. We can again compare this cloud to the covariance that the data is sampled from, and in large the results are reassuring. We used the same settings for both datasets, letting the models infer for 10^5 and 10^4 iterations respectively at each new data point. From the second half of those samples, we randomly pick 100 ones constituting our sub-samples.

Figures 6 and 7 show the one step predictions made on $\mathbf{\Sigma}^{\text{GWP}}$ using ESS and HMC, both accompanied by EMA using a half life of 50 days. In practice this is a fairly quick filter, but for such a rapidly changing time series it is still evidently too slow.[10] In the synthetic cases we need not concern ourselves with this too much however, as the focus lies on validating the model. Each plot shows how each unique element of Σ evolves over time, and the dashed line indicates where the predictions start. For each sample we calculate the different norms $||\Sigma^j - \Sigma||_{(\cdot)}$, and show the averages in figure 8.

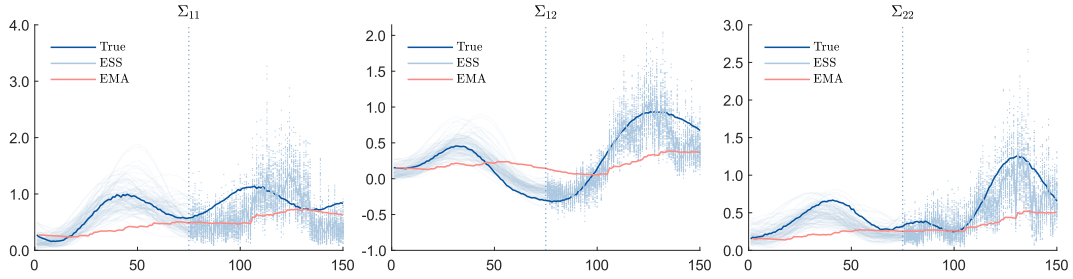

Figure 6: 1-step predictions for $\mathbf{\Sigma}^{\text{GWP}}$ using ESS, accompanied by generator (dark blue) and EMA (red). Dotted line indicates prediction start.

[10] For a sample covariance it is not unrealistic to have upwards of 500 days (2 years of trading) in half-life.

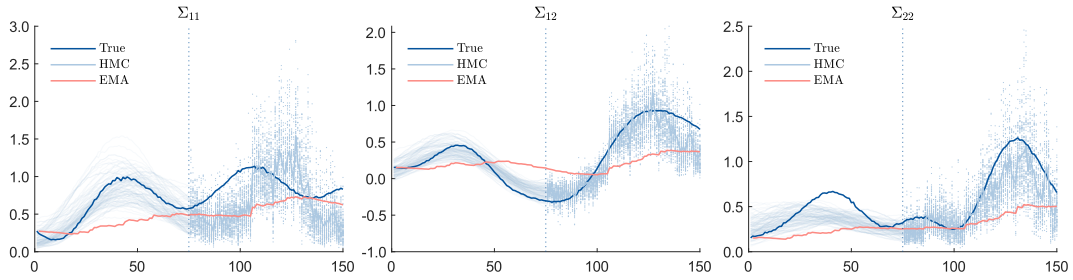

Figure 7: 1-step predictions for $\mathbf{\Sigma}^{\text{GWP}}$ using HMC, accompanied by generator (dark blue) and EMA (red). Dotted line indicates prediction start.

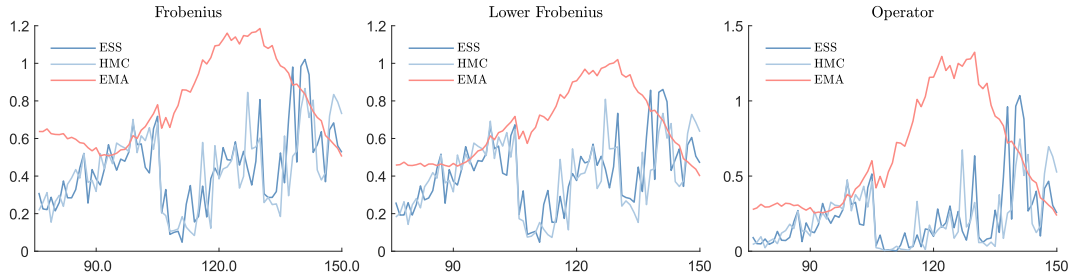

Figure 8: Average over $||\Sigma_t^j - \Sigma_t^{\text{GWP}}||_F$ for 1-step prediction samples.

Initially we notice striking similarities between both sampling algorithms. Indeed they are constructed to sample from the exact same posterior, so this is expected. The sampling captures the generator and responds fairly well to changes. One of the suppositions about the framework was escaping the robustness-responsiveness trade-off akin to the filter, and the Bayesian approach appears successful in that. Figure 8 further highlight the similarity of the two algorithms. The result on periodic covariance tell effectively the same story, and for this reason we leave those plots in appendix A.5.

7.5 Predictive Performance on Real Data

Due to aforementioned computational burden of HMC we only run the two elliptical models beside two benchmarks on the futures data sets, both EMA filters with half-lives of 50 and 500 respectively. Portfolio construction with the filter is straight forward as only one covariance estimate exists for each time point. For the samplers we transform the sample distribution of matrices Σ_t^j to a distribution of weights w_t^j, and the sample mean forms the portfolio for which the returns are calculated. The transformation from matrix to portfolio is always harmless for the inverse model, but for the other models we regularise the covariance by ridge shrinkage according to (3) using $\delta = 0.05$ before inverting it, to ensure conditioning somewhat. For each additional data point the inference produces 3000 samples, from the second half of which we keep 100 randomised sub-samples.

As an experiment we try two different models for the expected returns. The first and most intuitive one is using a trend follower based on an EMA_{60} filter where the half-life is chosen as around three trading months, without any optimisation or further motivations. The intuition behind such a trend follower is that assets that are performing well are expected to continue doing so, and vice versa. Secondly, we employ what we refer to as a *passive* trend, which puts the expected return to unity for all assets except the FX pair, which get assigned expectation 0. Omitting friction, an FX pair should

have a drift term proportional to the difference in short rates connected to the respective currencies (Björk 2019). So while setting this to 0 for our toy example implies we believe the Japanese and American short rates remain the same throughout the period, this is probably no more strange than having a constant expected return of 1 for the other assets. It does however add slight value as we can see how the models invest in the FX pair as a means of reducing risk, despite having no expected return. Cumulative returns are presented in figure 9 and portfolio metrics in tables 1 and 2.

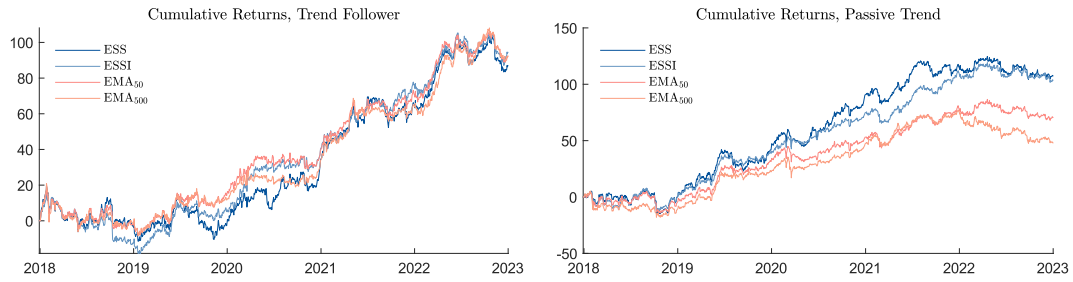

Figure 9: Cumulative portfolio returns, using a trend follower (left) and passive trend (right).

The normal trend follower is perhaps the sensible option in any real scenario, and so we start by focusing on those results. In a Sharpe sense, ESSI and EMA_{50} perform virtually the same, while EMA_{500} is slightly behind and ESS lags significantly. Considering VaR, the inverse model comes out a clear winner, as both filters are closer together. With expected shortfall, the outcome is qualitatively the same as for Sharpe, and it becomes clear that ESS is under performing so far. On tail events the slow filter suffers worst, and while the skewness is very close for the remaining models, the fast filter has the best kurtosis. Probably the most notable insight is that the inverse parameterisation offers a substantial improvement, and performs in parity with the fast filter.

Table 1: Portfolio metrics using a trend follower with $\tau_{1/2} = 60$. Sharpe is annualised and $\alpha = 0.05$ for VaR and ES.

	SR	VaR	ES	Skewness	Kurtosis
ESS	0.776	-2.14	-3.38	-0.698	6.87
ESSI	1.01	-1.67	-2.75	-0.689	6.32
EMA50	1.00	-1.81	-2.76	-0.723	5.95
EMA500	0.938	-1.84	-2.86	-0.844	10.1

When we instead consider the passive trend, results are possibly surprising. Both filters experience a clear drop in Sharpe, with the fast still outperforming the slow. On VaR they perform virtually the same, and essentially the same as they did for the trend follower. The striking difference is that ESS see substantial improvement on all metrics apart from a slight increase in kurtosis. For the inverse model metrics move in different directions. The tail measures suggest we see an improvement in shape, in the sense that the return distribution becomes more normal. This coincides with the risk measures increasing, if our distribution straightens up towards the left. The mean return possibly decreases slightly with this, but seeing as the Sharpe remains roughly the same one could speculate that the variance decreases in a way that compensates for this.

Table 2: Portfolio metrics using a passive trend. Sharpe is annualised and $\alpha = 0.05$ for VaR and ES.

	SR	VaR	ES	Skewness	Kurtosis
ESS	0.947	-2.06	-3.24	-0.357	7.19
ESSI	1.07	-1.84	-2.74	-0.318	5.59
EMA50	0.766	-1.82	-2.64	-0.494	7.12
EMA500	0.527	-1.82	-2.65	-0.499	5.81

8 Discussion

All in all the framework is successful in sampling from the posterior distribution of Σ given the data. It captures both fast and slow changes, but struggles on the shock-like datasets where long periods of constant correlation briefly undergoes rapid changes, as this challenges the time-scale parameter of the Gaussian processes. Although those cases are perhaps not the most crucial, one could continue to try the same generators but with a much more dense grid of data points to see if any improvement could be found.

Introducing Hamiltonian Monte Carlo was successful in the sense that sampling efficiency improved drastically compared to elliptical slice sampling. The acceptance rate increases vastly and the samples have a much lower autocorrelation. The algorithm is adjustable as we can let the leapfrog integrator run for longer, producing even less dependence between states. As such it is a more elegant approach and could potentially prove useful as dimensions increase and the elliptical model suffers further. In practice it does come with a price in computational time, and for scenarios similar to those investigated in this paper it is arguably a more simplistic approach to use the ESS and let it run for longer. However one must remember that the testing scenario is limited by time and computational power, whereas a live implementation can run much longer and with added computational resources, possibly facilitating the otherwise desirable Hamiltonian Monte Carlo sampler.

The predictive performance on synthetic data highlights the advantage of having a distribution to work with as opposed to a single estimate, and we see that both the elliptical and Hamiltonian sampler react well to changes in the underlying covariance without being very sensitive to outliers. Although the benchmark model was not particularly optimised in this setting, the results indicate that our framework models the covariance closer than the EMA filter while circumventing the previously discussed trade off between robustness and responsiveness.

Although the performance on real data is mixed, the overall results might appear somewhat underwhelming. By no means does our framework dominate the benchmarks, however this would arguably be a bit much to ask given the limited tests performed. With only 4 assets the filter estimates are likely not super noisy, at least not to the point where it immediately shows in the tests. The passive trend does suggest the slightly more rapid changes akin to the framework are beneficial, but it is difficult to make firm conclusions based on that test alone. It definitely appears being able to model the precision matrix directly has tangible advantages, which is in line with our presumptions. Its performance is on par or even slightly above that of the filter based models, but the differences are small. As such we refrain from making any firm claims, especially given the experiments are very limited in order to avoid overfitting. The objective is to produce a good portfolio given the same expected return, and optimising over different settings in order to find a high Sharpe ratio would arguably stray somewhat from the focus of the project. While not an unreasonable investigation to conduct, for the sake of covariance modelling the conclusions of such experiments might not carry over to other applications. The same reasoning is applied to the shrinkage performed ahead of inversion, as well as the half-lives of the filter models. The passive trend does indicate that the covariance is modelled well by the framework, which is supported by the synthetic experiments. In lack of a true covariance, it is very difficult to accurately assess how well we model this. After all, the metrics were motivated by the assumption that a better covariance estimate should yield better portfolios, but this need not always be the case.

Shifting focus from the different tests and metrics, we must also consider the Monte Carlo setup. First, it is worth mentioning sample sizes. Retaining only 100 sub-samples is an extremely small number in most MCMC settings, but as the matrices are used to project the expected returns down to just 4 portfolio weights, it appears this number gives a sufficient representation, as tests with even fewer samples yielded similar results. That being said, it is important to remember there is a significant burn-in period followed by 1500 samples, where the sub-samples where randomised to have very low autocorrelation. So while the sample should be sufficient, it is not impossible that longer runs could

yield slight improvements. Especially as dimensions increase and more parameters are to be sampled it becomes crucial to keep an eye on the sample size to make sure it is sufficient. Furthermore we have only used a very small asset universe, and a mere five years of data. Without the computational burden, experiments could have been reproduced over a number of different combinations of assets to get a more reliable and representative result.

We have kept the scale matrix fixed, and restricted the model to using just one kernel for all Gaussian processes. Extending our sampling to include the former, and experimenting with different kernels or at least different scale parameters could very well improve results. While extending the model would require greater sample sizes, it would be perfectly reasonable for future work.

In conclusion, the framework shows promise, and has a number of clear advantages. A limited test environment restricts us from making confident claims regarding exact performance, but we have indicated a successful dynamic covariance modelling procedure. We introduced a Hamiltonian Monte Carlo based sampling algorithm which greatly improved sampling efficiency over the elliptical counterpart, but unfortunately at a much greater computational cost. While not exhaustive, synthetic tests display desirable behaviour, and tests on financial data further indicate potential in the model. Future work could include extending the model by loosening the parameter restrictions, and conducting more extensive testing to further explore this otherwise neat framework.

References

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer New York, NY.

Björk, T. (2019). *Arbitrage Theory in Continuous Time*. Oxford University Press.

Chan, K. C., G. A. Karolyi, F. A. Longstaff, and A. B. Sanders (1992). An empirical comparison of alternative models of the short-term interest rate. *The Journal of Finance 47*(3), 1209–1227.

Cox, J. C., J. E. Ingersoll, and S. A. Ross (1985). A theory of the term structure of interest rates. *Econometrica 53*(2), 385–407.

Hamilton, W. R. (1833). On a general method of expressing the paths of light, & of the planets, by the coefficients of a characteristic function.

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

Heaukulani, C. and M. van der Wilk (2019). Scalable Bayesian dynamic covariance modeling with variational Wishart and inverse Wishart processes. In *Advances in Neural Information Processing Systems*, Volume 32, pp. 4582–4592. Curran Associates, Inc.

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 12*(1), 55–67.

Holst, A. and V. Ufnarovski (2014). *Matrix Theory*. Studentlitteratur AB.

Jyrkäs, T. (2023). An optimisation approach to portfolio selection. Master's thesis, Royal Institute of Technology.

Ledoit, O. and M. Wolf (2003). Honey, I shrunk the sample covariance matrix. *The Journal of Portfolio Management 30*, 110–119.

Lindström, E., H. Madsen, and J. Nygaard Nielsen (2015). *Statistics for Finance*. Chapman & Hall/CRC Texts in Statistical Science. Chapman and Hall.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance 7*(1), 77–91.

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

Michaud, R. (1989). The Markowitz optimization enigma: Is 'optimized' optimal? *Financial Analysts Journal 45*, 31–42.

Murray, I., R. Adams, and D. MacKay (2010). Elliptical slice sampling. *Journal of Machine Learning Research 9*, 541–548.

Neal, R. M. (2011). *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC.

Orlando, G., R. M. Mininni, and M. Bufalo (2020). A new approach to forecast market interest rates through the cir model. *Studies in Economics and Finance 37*(2), 267–292.

Petersen, K. B. and M. S. Pedersen (2012). The matrix cookbook.

Pfaffel, O. (2012). Wishart processes. Master's thesis, Technische Universität München.

Ranjan, P., R. D. Haynes, and R. Karsten (2010). A computationally stable approach to Gaussian process interpolation of deterministic computer simulation data. *Technometrics 53*, 366 – 378.

Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

Robert, C. P. and G. Casella (1999). *Monte Carlo Statistical Methods*. Springer New York, NY.

Sandoval, L. and I. D. P. Franca (2012). Correlation of financial markets in times of crisis. *Physica A: Statistical Mechanics and its Applications 391*(1-2), 187–208.

Schäfer, J. and K. Strimmer (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology 4*(1), 1–32.

Sharpe, W. F. (1966). Mutual fund performance. *The Journal of Business 39*(1), 119–138.

Wilson, A. and Z. Ghahramani (2010). Generalised Wishart processes. *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, UAI 2011*, 736–744.

Yang, D. and Q. Zhang (2000). Drift-independent volatility estimation based on high, low, open, and close prices. *The Journal of Business 73*(3), 477–492.

Zhang, T. (2021). *Probabilistic machine learning methods for automated radiation therapy treatment planning*. Ph. D. thesis, KTH Royal Institute of Technology.

A Supplementary Plots and Tables

In the interest of brevity we tried to only present plots adding significant value. The remaining plots are presented below for full disclosure.

A.1 Independent Standard Gaussians

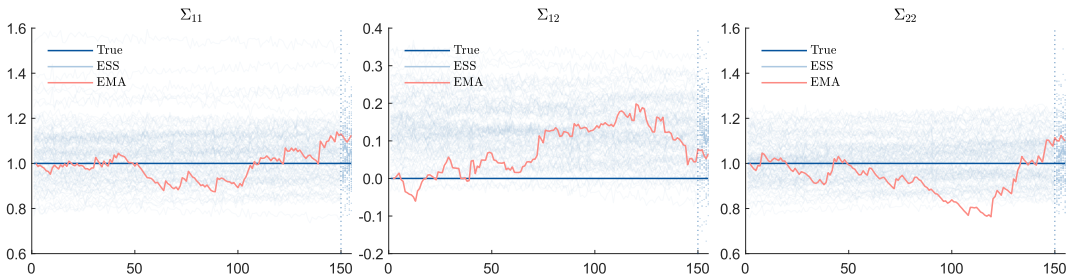

Figure 10: Samples of $\mathbf{\Sigma}^{\mathrm{ID}}$ using ESS, accompanied by the generator (dark blue) and EMA filter (red).

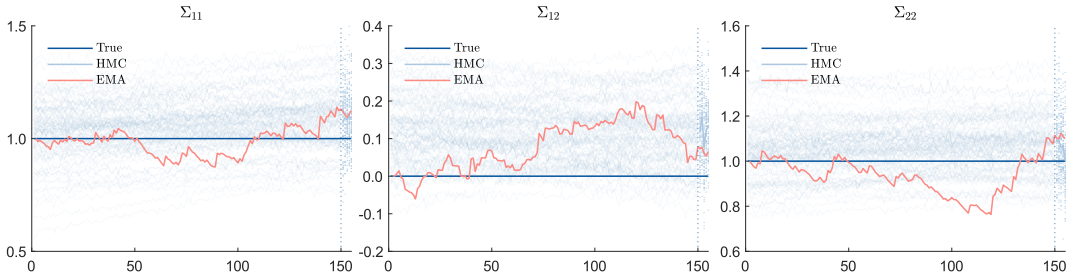

Figure 11: Samples of $\mathbf{\Sigma}^{\mathrm{ID}}$ using HMC, accompanied by the generator (dark blue) and EMA filter (red).

A.2 Constant Correlation Gaussians

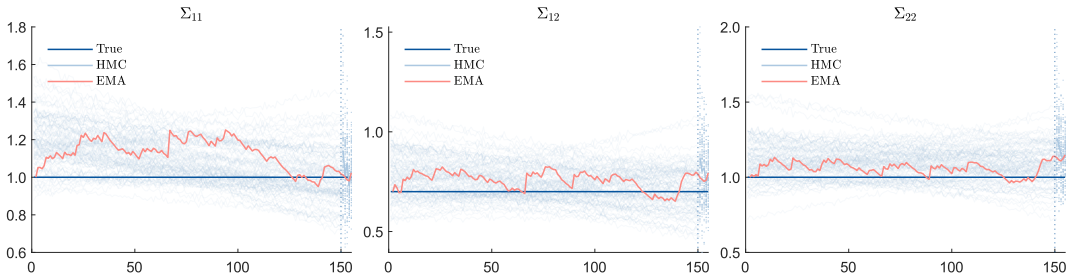

Figure 12: Samples of $\mathbf{\Sigma}^{\mathrm{CC}}$ using HMC, accompanied by the generator (dark blue) and EMA filter (red).

A.3 Regime Shift

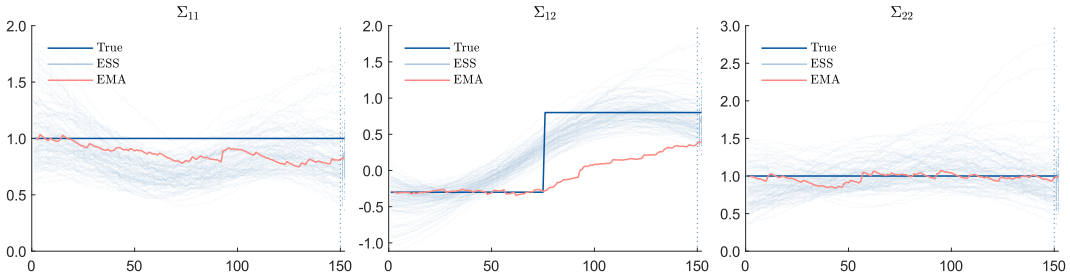

Figure 13: Samples of $\mathbf{\Sigma}^{\mathrm{RS}}$ using ESS, accompanied by the generator (dark blue) and EMA filter(red).

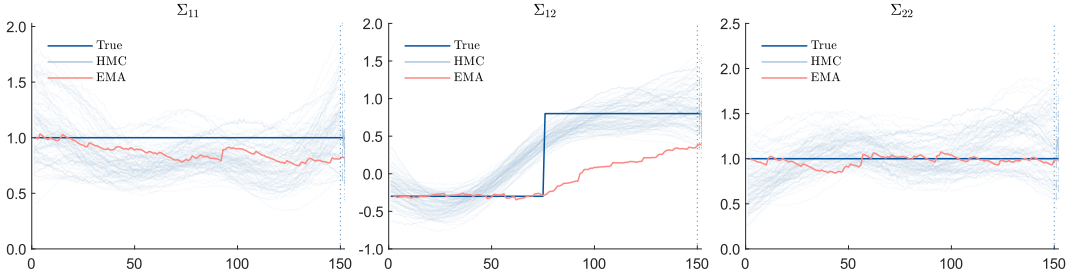

Figure 14: Samples of $\mathbf{\Sigma}^{\mathrm{RS}}$ using HMC, accompanied by the generator (dark blue) and EMA filter (red).

A.4 Crisis

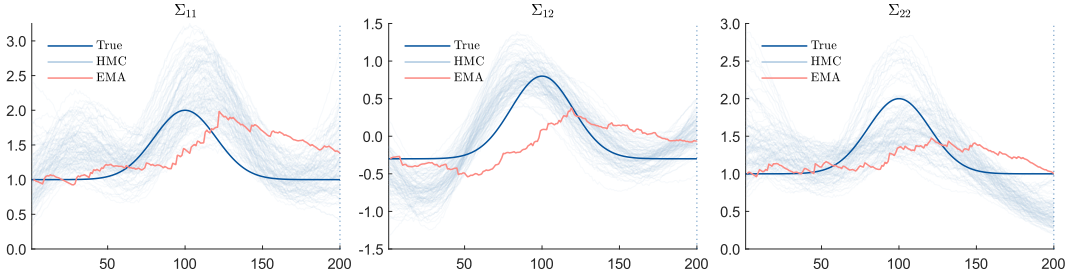

Figure 15: Samples of $\mathbf{\Sigma}^{\mathrm{CR}}$ using HMC, accompanied by the generator (dark blue) and EMA filter (red).

A.5 Sinusoidal Correlation

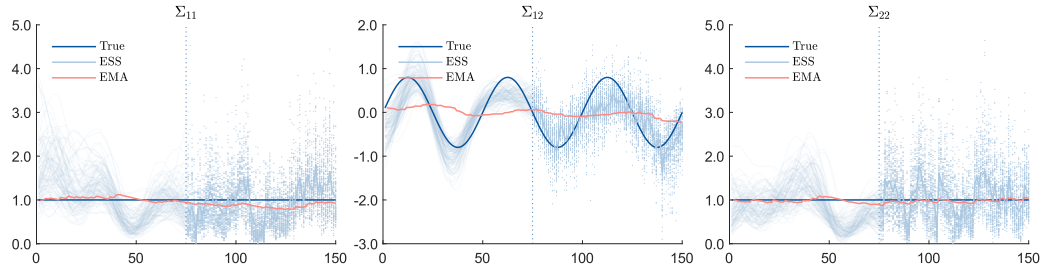

Figure 16: 1-step prediction samples for $\mathbf{\Sigma}^{\text{SIN}}$ using ESS, accompanied by the generator (dark blue) and EMA (red). Dotted line indicates prediction start.

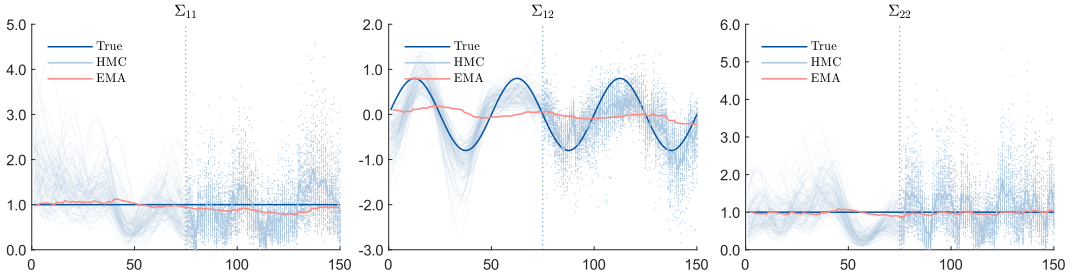

Figure 17: 1-step prediction samples for $\mathbf{\Sigma}^{\text{SIN}}$ using HMC, accompanied by generator (dark blue) and EMA (red). Dotted line indicates prediction start.

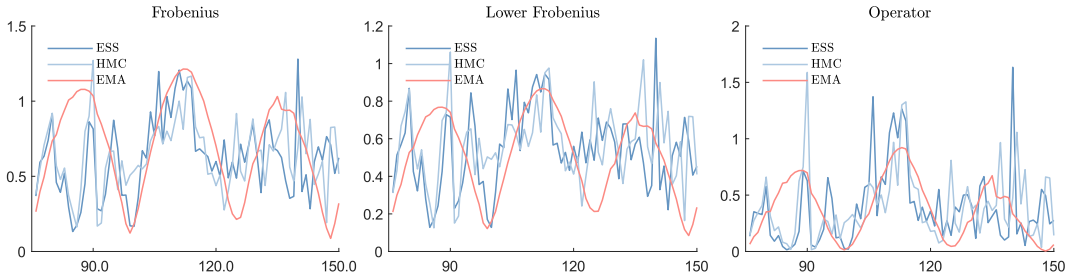

Figure 18: Average norms for 1-step prediction samples of $\mathbf{\Sigma}^{\text{SIN}}$.

B Derivations

B.1 Matrix Identities

Aside from a few trivial results, we collect and utilise a few identities from the renowned *Matrix Cookbook* (Petersen and Pedersen (2012)). Those are primarily used in deriving the gradients for the HMC algorithm, which done in appendix B.2-B.3. Each identity is presented by its number in the cookbook, and for ease of reference we label them locally.

Eq. 44

$$\partial X^\top = (\partial X)^\top \tag{16}$$

Eq. 55

$$\frac{\partial \log(\det(X^\top X))}{\partial X} = 2(X^+)^\top \tag{17}$$

Eq. 85 using s = 0 and S symmetric

$$\frac{\partial}{\partial x} x^\top S x = 2Sx \tag{18}$$

Eq. 110

$$\frac{\partial}{\partial X} \mathrm{tr}\left(XX^\top B\right) = BX + B^\top X \tag{19}$$

Eq. 125 using $C = I$

$$\frac{\partial}{\partial X} \mathrm{tr}((X^\top X)^{-1} A) = -(X(X^\top X)^{-1}(A + A^\top)(X^\top X)^{-1}) \tag{20}$$

B.2 Potential Gradient

For the potential we have

$$\begin{aligned}
V(\mathbf{U}) =& -\log p(\mathbf{Y}|\mathbf{U}) - \log p(\mathbf{U}) \\
=& -\log\left(\prod_{n=1}^{N} (2\pi|\Sigma_n|)^{-1/2} \exp\left\{-\frac{1}{2} Y_n^\top (L U_n U_n^\top L^\top)^{-1} Y_n\right\}\right) \\
& -\log\left(\prod_{d=1}^{D}\prod_{i=1}^{\nu} (2\pi|K_{d,i}|)^{-1/2} \exp\left\{-\frac{1}{2} U_{d,i}^\top K_{d,i} U_{d,i}\right\}\right) \\
=& \sum_{n=1}^{N}\left(\frac{D}{2}\log(2\pi) + \frac{1}{2}\log(\det(\Sigma_n)) + \frac{1}{2} Y_n^\top (L U_n U_n^\top L^\top)^{-1} Y_n\right) \\
& + \sum_{d=1}^{D}\sum_{i=1}^{\nu}\left(\frac{N}{2}\log(2\pi) + \frac{1}{2}\log(\det(K_{d,i})) + \frac{1}{2} U_{d,i}^\top K_{d,i} U_{d,i}\right) \\
=& V_1 + V_2
\end{aligned}$$

The derivative of V with respect to each element can then be written as

$$\nabla V_{d,i,n} = \frac{\partial V_1}{\partial U_n}\frac{\partial U_n}{\partial u_{d,i,n}} + \frac{\partial V_2}{\partial U_{d,i}}\frac{\partial U_{d,i}}{\partial u_{d,i,n}}.$$

Here it is enough to find the first part of each term on matrix form, as the second parts are essentially just indicator functions to single out individual elements. The matrices are then stacked appropriately to construct a matrix of the same size as $\mathbf{U}$ containing the element wise derivatives. We treat the terms

separately, starting with the slices in time. We use a selection of non-trivial identities collected from Petersen and Pedersen (2012) and presented in above in appendix B.1, which when used are denoted by an overset equality sign. Introducing $\tilde{Y} = L^{-1}Y$, we have

$$\begin{aligned}
\frac{\partial V_1}{\partial U_n} &= \frac{\partial}{\partial U_n}\sum_{n=1}^{N}\left(\frac{D}{2}\log(2\pi) + \frac{1}{2}\log(\det(LU_nU_n^\top L^\top)) + \frac{1}{2}Y_n^\top(LU_nU_n^\top L^\top)^{-1}Y_n\right) \\
&= \frac{1}{2}\frac{\partial}{\partial U_n}\log(2\det(L)) + \log(\det(U_nU_n^\top))) + \frac{1}{2}\frac{\partial}{\partial u_n}Y_n^\top L^{-\top}(U_nU_n^\top)^{-1}L^{-1}Y_n \\
&= \frac{1}{2}\frac{\partial}{\partial U_n}\log(\det(U_nU_n^\top)) + \frac{1}{2}\frac{\partial}{\partial U_n}\tilde{Y}_n^\top(U_nU_n^\top)^{-1}\tilde{Y}_n. \qquad (21)
\end{aligned}$$

To ease things even further, we treat these terms individually too.

$$\begin{aligned}
\frac{1}{2}\frac{\partial}{\partial U_n}\log(\det(U_nU_n^\top)) &= \left[W^\top = U_n\right] \\
&= \frac{1}{2}\frac{\partial}{\partial W^\top}\log(\det(W^\top W)) \\
&\overset{(16)}{=} \frac{1}{2}\left(\frac{\partial}{\partial W}(W^\top W)\right)^\top \\
&\overset{(17)}{=} \frac{1}{2}\left(2(W^+)\right)^\top \\
&= W^+ \\
&= (U_n^+)^\top = (U_nU_n^\top)^{-\top}U_n.
\end{aligned}$$

In the last step, we use the construction of the pseudo-inverse of a broad ($n \times m$, rank n) matrix. Moving on,

$$\begin{aligned}
\frac{1}{2}\frac{\partial}{\partial U_n}\tilde{Y}_n^\top(U_nU_n^\top)^{-1}\tilde{Y}_n &= \frac{1}{2}\frac{\partial}{\partial U_n}\mathrm{tr}\left((U_nU_n^\top)^{-1}\tilde{Y}_n\tilde{Y}_n^\top\right) \\
&= \left[W^\top = U_n, A = \tilde{Y}_n\tilde{Y}_n^\top\right] \\
&= \frac{1}{2}\frac{\partial}{\partial W^\top}\mathrm{tr}\left((W^\top W)^{-1}A\right) \\
&\overset{(16)}{=} \frac{1}{2}\left(\frac{\partial}{\partial W}\mathrm{tr}\left((W^\top W)^{-1}A\right)\right)^\top \\
&\overset{(20)}{=} \left(-W(W^\top W)^{-1}A(W^\top W)^{-1}\right)^\top \\
&= \left(-U_n^\top(U_nU_n^\top)^{-1}A(U_nU_n^\top)^{-1}\right)^\top
\end{aligned}$$

where we used the symmetry of A to cancel the factor 2. Inserted into (21), we get

$$\frac{\partial V_1}{\partial U_n} = (U_n^+)^\top - \left(U_n^+A(U_nU_n^\top)^{-1}\right)^\top \qquad (22)$$

where $A = L^{-1}YY^\top L^{-\top}$. Considering the second part of the potential,

$$\begin{aligned}
\frac{\partial V_2}{\partial U_{d,i}} &= \frac{\partial}{\partial U_{d,i}}\sum_{d=1}^{D}\sum_{i=1}^{\nu}\left(\frac{N}{2}\log(2\pi) + \frac{1}{2}\log(\det(K_{d,i})) + \frac{1}{2}U_{d,i}^\top K_{d,i}U_{d,i}\right) \\
&= \frac{1}{2}\frac{\partial}{\partial U_{d,i}}U_{d,i}^\top K_{d,i}U_{d,i} \overset{(18)}{=} K_{d,i}^{-1}U_{d,i}. \qquad (23)
\end{aligned}$$

We remind ourselves that (22) and (23) have different dimensions, and have to be inserted into the matrix separately.

B.3 Potential Gradient, Inverse Parameterisation

The derivation for the inverse case is near identical, but has two minute differences which we treat explicitly for full clarity. The potential is now

$$\begin{aligned}
V(\mathbf{U}) =& -\log p(\mathbf{Y}|\mathbf{U}) - \log p(\mathbf{U}) \\
=& -\log\left(\prod_{n=1}^{N}(2\pi|\Sigma_n^{-1}|)^{-1/2}\exp\left\{-\frac{1}{2}Y_n^\top LU_nU_n^\top L^\top)Y_n\right\}\right) \\
& -\log\left(\prod_{d=1}^{D}\prod_{i=1}^{\nu}(2\pi|K_{d,i}|)^{-1/2}\exp\left\{-\frac{1}{2}U_{d,i}^\top K_{d,i}U_{d,i}\right\}\right) \\
=& \sum_{n=1}^{N}\left(\frac{D}{2}\log(2\pi) - \frac{1}{2}\log(\det(\Sigma_n)) + \frac{1}{2}Y_n^\top LU_nU_n^\top L^\top Y_n\right) \\
& + \sum_{d=1}^{D}\sum_{i=1}^{\nu}\left(\frac{N}{2}\log(2\pi) + \frac{1}{2}\log(\det(K_{d,i})) + \frac{1}{2}U_{d,i}^\top K_{d,i}U_{d,i}\right) \\
=& V_1 + V_2
\end{aligned} \tag{24}$$

where the only two differences are the sign change in front of the determinant, and the missing inverse on the last term of (24). Skipping straight to the last term, and using a slightly different change of variables, $\tilde{Y} = LY$, this becomes

$$\begin{aligned}
\frac{1}{2}\frac{\partial}{\partial U_n}\tilde{Y}_n^\top U_nU_n^\top\tilde{Y}_n &= \frac{1}{2}\frac{\partial}{\partial U_n}\mathrm{tr}\left((U_nU_n^\top)\tilde{Y}_n\tilde{Y}_n^\top\right) \\
&= \left[B = \tilde{Y}_n\tilde{Y}_n^\top\right] \\
&= \frac{1}{2}\frac{\partial}{\partial U}\mathrm{tr}\left((UU^\top)B\right) \\
&\overset{(19)}{=} \frac{1}{2}(BU + B^\top U) = BU
\end{aligned}$$

where, again, the symmetry of B cancels the factor 2. For the inverse parameterisation, we get in total

$$\frac{\partial V_1}{\partial U_n} = -(U_n^+)^\top + BU_n$$

having used $B = LYY^\top L^\top$. The contribution from the second term V_2 is unchanged, and the entries are inserted into the full gradient separately just like before.