APPLICATIONS OF GAUSSIAN PROCESSES IN FINANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Estimating covariances between financial assets plays an important role in risk management. In practice, when the sample size is small compared to the number of variables, the empirical estimate is known to be very unstable. Here, we propose a novel covariance estimator based on the Gaussian Process Latent Variable Model (GP-LVM). Our estimator can be considered as a non-linear extension of standard factor models with readily interpretable parameters reminiscent of market betas. Furthermore, our Bayesian treatment naturally shrinks the sample covariance matrix towards a more structured matrix given by the prior and thereby systematically reduces estimation errors. Finally, we discuss some financial applications of the GP-LVM model.

1 INTRODUCTION

Many financial problems require the estimation of covariance matrices between given assets. This may be useful to optimize one's portfolio, i.e.: maximize the portfolio returns $w^T r$ and/or minimize the volatility $\sqrt{w^T K w}$. Indeed, Markowitz received a Noble Price in economics for his treatment of modern portfolio theory (Markowitz, 1952). In practice, estimating historical returns and high-dimensional covariance matrices is challenging and often times equally weighted portfolio outperforms the portfolio constructed from sample estimates (Jobson & Korkie, 1981). The estimation of covariance matrices is especially hard, when the number of assets is large compared to the number of observations. Sample estimations in those cases are very unstable or can even become singular. To cope with this problem, a wide range of estimators, e.g. factor models such as the single-index model (Sharpe, 1991) or shrinkage estimators (Ledoit & Wolf, 2004), have been developed and employed in portfolio optimization.

With todays machine learning techniques we can even further improve those estimates. Machine learning has already arrived in finance. Nevmyvaka et al. (2006) trained an agent via reinforcement learning to optimally execute trades. Gately (1995) forecast asset prices with neural networks and Chapados & Bengio (2008) with Gaussian Processes. Recently, Heaton et al. (2016) made an ansatz to optimally allocate portfolios using deep autoencoders. Wu et al. (2014) used Gaussian Processes to build volatility models and Wilson & Ghahramani (2011) to estimate time varying covariance matrices. Bayesian machine learning methods are used more and more in this domain. The fact, that in Bayesian framework parameters are not treated as true values, but as random variables, accounts for estimation uncertainties and can even alleviate the unwanted impacts of outliers. Furthermore, one can easily incorporate additional information and/or personal views by selecting suitable priors.

In this paper, we propose a Bayesian covariance estimator based on the Gaussian Process Latent Variable Model (GP-LVM) (Lawrence, 2005), which can be considered as a non-linear extension of standard factor models with readily interpretable parameters reminiscent of market betas. Our Bayesian treatment naturally shrinks the sample covariance matrix (which maximizes the likelihood function) towards a more structured matrix given by the prior and thereby systematically reduces estimation errors. We evaluated our model on the stocks of S&P500 and found significant improvements in terms of model fit compared to classical linear models. Furthermore we suggest some financial applications, where Gaussian Processes can be used as well. That includes portfolio allocation, price prediction for less frequently traded stocks and non-linear clustering of stocks into their sub-sectors.

In section 2 we begin with an introduction to the Bayesian non-parametric Gaussian Processes and discuss the associated requirements for learning. Section 3 introduces the financial background

needed for portfolio optimization and how to relate it to Gaussian Processes. In section 4 we conduct experiments on covariance matrix estimations and discuss the results. We conclude in section 5.

2 BACKGROUND

In this paper, we utilize a Bayesian non-parametric machine learning approach based on Gaussian Processes (GPs). Combining those with latent variable models leads to Gaussian Process Latent Variable Models (GP-LVMs), that we use to estimate the covariance between different assets. These approaches have been described in detail in (Lawrence, 2005; Rasmussen, 2006). We provide a brief review here. Subsequently, we show, how to relate those machine learning approaches to the known models in finance, e.g. the single-index model (Sharpe, 1991).

2.1 GAUSSIAN PROCESSES

A Gaussian Process (GP) is a generalization of the Gaussian distribution. Using a GP, we can define a distribution over functions $f(\boldsymbol{x})$, where $\boldsymbol{x} \in \mathbb{R}^Q$ and $f \in \mathbb{R}$. Like a Gaussian distribution, the GP is specified by a mean and a covariance. In the GP case, however, the mean is a function of the input variable $m(\boldsymbol{x})$ and the covariance is a function of two variables $k(\boldsymbol{x}, \boldsymbol{x}')$, which contains information about how the GP evaluated at $\boldsymbol{x}$ and $\boldsymbol{x}'$ covary

$$m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})], \tag{1}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \text{cov}(f(\boldsymbol{x}), f(\boldsymbol{x}')). \tag{2}$$

We write $f \sim \text{GP}(m(\cdot), k(\cdot, \cdot))$. Any finite collection of function values, at $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, is jointly Gaussian distributed

$$p(f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), ..., f(\boldsymbol{x}_N)) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K}), \tag{3}$$

where $\boldsymbol{\mu} = (m(\boldsymbol{x}_1), ..., m(\boldsymbol{x}_N))^T$ is the mean vector and $\boldsymbol{K} \in \mathbb{R}^{N \times N}$ is the Gram matrix with entries $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. We refer to the covariance function as kernel function. The properties of the function f (i.e. smoothness, periodicity) are determined by the choice of this kernel function. For example, sampled functions from a GP with an exponentiated quadratic covariance function $k_{\text{SE}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp(-0.5||\boldsymbol{x} - \boldsymbol{x}'||_2^2/l^2)$ smoothly vary with lengthscale l and are infinitely often differentiable.

Given a dataset $\mathcal{D}$ of N input points $\boldsymbol{X} = (\boldsymbol{x_1}, ..., \boldsymbol{x_N})^T$ and N corresponding targets $\boldsymbol{y} = (y_1, ..., y_N)^T$, the predictive distribution for a zero mean GP at N_* new locations $\boldsymbol{X}_*$ reads (Rasmussen, 2006)

$$\boldsymbol{y}_*|\boldsymbol{X}_*, \boldsymbol{y}, \boldsymbol{X} \sim \mathcal{N}(\boldsymbol{f}_*, \boldsymbol{K}_*), \tag{4}$$

where

$$\boldsymbol{f}_* = \boldsymbol{K}_{\boldsymbol{X}_*\boldsymbol{X}} \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}^{-1} \boldsymbol{y}, \tag{5}$$

$$\boldsymbol{K}_* = \boldsymbol{K}_{\boldsymbol{X}_*\boldsymbol{X}_*} - \boldsymbol{K}_{\boldsymbol{X}_*\boldsymbol{X}} \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}^{-1} \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}_*}. \tag{6}$$

$\boldsymbol{K}_{\boldsymbol{X}_*\boldsymbol{X}} \in \mathbb{R}^{N_* \times N}$ is the covariance matrix between the GP evaluated at $\boldsymbol{X}_*$ and $\boldsymbol{X}$, $\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} \in \mathbb{R}^{N \times N}$ is the covariance matrix of the GP evaluated at $\boldsymbol{X}$. As we can see in equations (5) and (6), the kernel function plays a very important role in the GP framework and will be important for our financial model as well.

2.2 GAUSSIAN PROCESS LATENT VARIABLE MODEL

Often times we are just given a data matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times D}$ and the goal is to find a lower dimensional representation $\boldsymbol{X} \in \mathbb{R}^{N \times Q}$, without losing too much information. Principal component analysis (PCA) is one of the most used techniques for reducing the dimensions of the data, which has also been motivated as the maximum likelihood solution to a particular form of Gaussian Latent Variable Model (Tipping & Bishop, 1999). PCA embeds $\boldsymbol{Y}$ via a linear mapping into the latent space $\boldsymbol{X}$. Lawrence (2005) introduced the Gaussian Process Latent Variable Model (GP-LVM) as a non-linear extension of probabilistic PCA. The generative procedure takes the form

$$\boldsymbol{Y}_{n,:} = \boldsymbol{f}(\boldsymbol{X}_{n,:}) + \boldsymbol{\epsilon}_n, \tag{7}$$

where $\boldsymbol{f} = (f_1, ..., f_D)^T$ is a group of D independent samples from a GP, i.e. $f_d \sim \text{GP}(0, k(\cdot, \cdot))$. By this, we assume the rows of $\boldsymbol{Y}$ to be jointly Gaussian distributed and the columns to be independent, i.e. each sample $\boldsymbol{Y}_{:,d} \sim \mathcal{N}(\boldsymbol{Y}_{:,d}|\boldsymbol{0}, \boldsymbol{K})$ where $\boldsymbol{K} = k(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I}$ and σ^2 denotes the variance of the random noise ϵ. The marginal likelihood of $\boldsymbol{Y}$ becomes (Lawrence, 2005)

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{Y}_{:,d}|\boldsymbol{0}, \boldsymbol{K}) = \frac{1}{(2\pi)^{ND/2}|\boldsymbol{K}|^{D/2}} \exp\left(-\frac{1}{2}\text{tr}(\boldsymbol{K}^{-1}\boldsymbol{Y}\boldsymbol{Y}^T)\right). \tag{8}$$

The dependency on the latent positions $\boldsymbol{X}$ and the kernel hyperparameters is given through the kernel matrix $\boldsymbol{K}$. As suggested by Lawrence (2005), we can optimize the log marginal likelihood $\log p(\boldsymbol{Y}|\boldsymbol{X})$ with respect to the latent positions and the hyperparameters.

2.3 VARIATIONAL INFERENCE

Optimization can easily lead to overfitting. Therefore, a fully Bayesian treatment of the model would be preferable but is intractable. Titsias & D. Lawrence (2010) introduced a variational inference framework, which not only handles the problem of overfitting but also allows to automatically select the dimensionality of the latent space. Instead of optimizing equation (8), we want to calculate the posterior using Bayes rule $p(\boldsymbol{X}|\boldsymbol{Y}) = p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})/p(\boldsymbol{Y})$, which is intractable. The idea behind variational Bayes is to approximate the true posterior $p(\boldsymbol{X}|\boldsymbol{Y})$ by another distribution $q(\boldsymbol{X})$, selected from a tractable family. The goal is to select the one distribution $q(\boldsymbol{X})$, that is closest to the true posterior $p(\boldsymbol{X}|\boldsymbol{Y})$ in some sense. A natural choice to quantify the closeness is given by the Kullback-Leibler divergence (Cover & Thomas, 1991)

$$\text{KL}[q(\boldsymbol{X})||p(\boldsymbol{X}|\boldsymbol{Y})] = \int q(\boldsymbol{X}) \log \frac{q(\boldsymbol{X})}{p(\boldsymbol{X}|\boldsymbol{Y})} d\boldsymbol{X}. \tag{9}$$

By defining $\tilde{p}(\boldsymbol{X}|\boldsymbol{Y}) = p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})$ as the unnormalized posterior, equation (9) becomes

$$\text{KL}[q(\boldsymbol{X})||p(\boldsymbol{X}|\boldsymbol{Y})] = \int q(\boldsymbol{X}) \log \frac{q(\boldsymbol{X})}{\tilde{p}(\boldsymbol{X}|\boldsymbol{Y})} d\boldsymbol{X} + \log p(\boldsymbol{Y}) = \underbrace{-\mathbb{E}_{q(\boldsymbol{X})}\left[\log \frac{\tilde{p}(\boldsymbol{X}|\boldsymbol{Y})}{q(\boldsymbol{X})}\right]}_{\text{ELBO}} + \log p(\boldsymbol{Y})$$

$$\tag{10}$$

with the first term on the right hand side being known as the evidence lower bound (ELBO). Equation (10) is the objective function we want to minimize with respect to $q(\boldsymbol{X})$ to get a good approximation to the true posterior. Note that on the left hand side only the ELBO is q dependent. So, in order to minimize (10), we can just as well maximize the ELBO. Because the Kullback-Leibler divergence is non-negative, the ELBO is a lower bound on the evidence $\log p(\boldsymbol{Y})$[1]. Therefore, this procedure not only gives the best approximation to the posterior, but it also bounds the evidence, which serves as a measure of the goodness of our fit. The number of latent dimensions Q can be chosen to be the one, which maximizes the ELBO.

So, GP-LVM is an algorithm, which reduces the dimensions of our data-matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times D}$ from D to Q in a non-linear way and at the same time estimates the covariance matrix between the N points. The estimated covariance matrix can then be used for further analysis.

3 FINANCE

Now we have a procedure to estimate the covariance matrix between different datapoints. This section discusses how we can relate this to financial models.

3.1 CAPM

The Capital Asset Pricing Model (CAPM) describes the relationship between the expected returns of an asset $\boldsymbol{r}_n \in \mathbb{R}^D$ for D days and its risk β_n

$$\mathbb{E}[\boldsymbol{r}_n] = \boldsymbol{r}_f + \beta_n \mathbb{E}[\boldsymbol{r}_m - \boldsymbol{r}_f], \tag{11}$$

---

[1]The evidence $\log p(\boldsymbol{Y})$ is also referred to as log marginal likelihood in the literature. The term marginal likelihood is already used for $p(\boldsymbol{Y}|\boldsymbol{X})$ in this paper. Therefore, we will refer to $\log p(\boldsymbol{Y})$ as the evidence.

where $\boldsymbol{r}_f \in \mathbb{R}^D$ is the risk free return on D different days and $\boldsymbol{r}_m$ is the market return on D different days. The main idea behind CAPM is, that an investor needs to be compensated for the risk of his holdings. For a risk free asset with $\beta_n = 0$, the expected return $\mathbb{E}[\boldsymbol{r}_n]$ is just the risk free rate $\boldsymbol{r}_f$. If an asset is risky with a risk $\beta_n \neq 0$, the expected return $\mathbb{E}[\boldsymbol{r}_n]$ is increased by $\beta_n \mathbb{E}[\tilde{\boldsymbol{r}}_m]$, where $\tilde{\boldsymbol{r}}_m$ is the excess return of the market $\tilde{\boldsymbol{r}}_m = \boldsymbol{r}_m - \boldsymbol{r}_f$.

We can write down equation (11) in terms of the excess return $\tilde{\boldsymbol{r}} = \boldsymbol{r} - \boldsymbol{r}_f$ and get

$$\mathbb{E}[\tilde{\boldsymbol{r}}_n] = \beta_n \mathbb{E}[\tilde{\boldsymbol{r}}_m], \tag{12}$$

where $\tilde{\boldsymbol{r}}_n$ is the excess return of a given asset and $\tilde{\boldsymbol{r}}_m$ is the excess return of the market (also called a risk factor). Arbitrage pricing theory (Roll & Ross, 1995) generalizes the above model by allowing multiple risk factors $\boldsymbol{F}$ beside the market $\tilde{\boldsymbol{r}}_m$. In particular, it assumes that asset returns follow a factor structure

$$\boldsymbol{r}_n = \alpha_n + \boldsymbol{F}\boldsymbol{\beta}_n + \boldsymbol{\epsilon}_n, \tag{13}$$

with $\boldsymbol{\epsilon}_n$ denoting some independent zero mean noise with variance σ_n^2. Here, $\boldsymbol{F} \in \mathbb{R}^{D \times Q}$ is the matrix of Q factor returns on D days and $\boldsymbol{\beta}_n \in \mathbb{R}^Q$ is the loading of stock n to the Q factors. Arbitrage pricing theory (Ross, 1976) then shows that the expected excess returns adhere to

$$\mathbb{E}[\tilde{\boldsymbol{r}}_n] = \mathbb{E}[\tilde{\boldsymbol{F}}]\boldsymbol{\beta}_n, \tag{14}$$

i.e. the CAPM is derived as special case when assuming a single risk factor (single-index model).

To match the form of the GP-LVM (see equation (7)), we rewrite equation (13) as

$$\boldsymbol{r}_{n,:} = \boldsymbol{f}(\boldsymbol{B}_{n,:}) + \boldsymbol{\epsilon}_n, \tag{15}$$

where $\boldsymbol{r} \in \mathbb{R}^{N \times D}$ is the return matrix[2]. Note that assuming latent factors distributed as $\boldsymbol{F} \sim \mathcal{N}(0, 1)$ and marginalizing over them, equation (13) is a special case of equation (15) with $\boldsymbol{f}$ drawn from a GP mapping $\boldsymbol{\beta}_n$ to $\boldsymbol{r}_n$ with a linear kernel. Interestingly, this provides an exact correspondence with factor models by considering the matrix of couplings $\boldsymbol{B} = (\boldsymbol{\beta_1}, ..., \boldsymbol{\beta_N})^T$ as the latent space positions[3]. In this perspective the factor model can be seen as a linear dimensionality reduction, where we reduce the $N \times D$ matrix $\boldsymbol{r}$ to a low rank matrix $\boldsymbol{B}$ of size $N \times Q$. By chosing a non-linear kernel $k(\cdot, \cdot)$ the GP-LVM formulation readily allows for non-linear dimensionality reductions. Since, it is generally known, that different assets have different volatilities, we further generalize the model. In particular, we assume the noise $\boldsymbol{\epsilon}$ to be a zero mean Gaussian, but allow for different variances σ_n^2 for different stocks. For this reason, we also have to parameterize the kernel (covariance) matrix in a different way than usual. Section 4.1 explains how to deal with that. The model is then approximated using variational inference as described in section 2.3. After inferring $\boldsymbol{B}$ and the hyperparameters of the kernel, we can calculate the covariance matrix $\boldsymbol{K}$ and use it for further analysis.

3.2 MODERN PORTFOLIO THEORY

Markowitz (1952) provided the foundation for modern portfolio theory, for which he received a Nobel Price in economics. The method analyses how good a given portfolio is, based on the mean and the variance of the returns of the assets contained in the portfolio. This can also be formulated as an optimization problem for selecting an optimal portfolio, given the covariance between the assets and the risk tolerance q of the investor.

Given the covariance matrix $\boldsymbol{K} \in \mathbb{R}^{N \times N}$, we can calculate the optimal portfolio weights $\boldsymbol{w}$ by

$$\boldsymbol{w}_{\text{opt}} = \min_{\boldsymbol{w}}(\boldsymbol{w}^T \boldsymbol{K} \boldsymbol{w} - q\bar{\boldsymbol{r}}^T \boldsymbol{w}), \tag{16}$$

where $\bar{\boldsymbol{r}}$ is the mean return vector. Risk friendly investors have a higher q than risk averse investors. The model is constrained by $\sum_{\boldsymbol{w}} = 1$. Since $\bar{\boldsymbol{r}}$ is very hard to estimate in general and we are primarily interested in the estimation of the covariance matrix $\boldsymbol{K}$, we set q to zero and get

$$\boldsymbol{w}_{\text{opt}} = \min_{\boldsymbol{w}}(\boldsymbol{w}^T \boldsymbol{K} \boldsymbol{w}). \tag{17}$$

This portfolio is called the minimal risk portfolio, i.e. the solution to equation (17) provides the weights for the portfolio, which minimizes the overall risk, assuming the estimated $\boldsymbol{K}$ is the true covariance matrix.

---

[2]To stay consistent with the financial literature, we denote the return matrix with lower case $\boldsymbol{r}$.

[3]Because of the context, from now on we will use $\boldsymbol{\beta}$ for the latent space instead of $\boldsymbol{x}$.

4 EXPERIMENTS

In this section, we discuss the performance of the GP-LVM on financial data. After describing the data collection and modeling procedure, we evaluate the model on the daily return series of the S&P500 stocks. Subsequently, we discuss further financial applications. In particular, we show how to build a minimal risk portfolio (this can easily be extended to maximizing returns as well), how to fill-in prices for assets which are not traded frequently and how to visualize sector relations between different stocks (latent space embedding).

4.1 DATA COLLECTION AND MODELING

For a given time period, we take all the stocks from the S&P500, whose daily close prices were available for the whole period[4]. The data were downloaded from Yahoo Finance. After having the close prices in a matrix $p \in \mathbb{R}^{N \times (D+1)}$, we calculate the return matrix $r \in \mathbb{R}^{N \times D}$, where $r_{nd} = (p_{n,d} - p_{n,d-1})/p_{n,d-1}$. r builds the basis of our analysis.

We can feed r into the GP-LVM. The GP-LVM procedure, as described in section 2, assumes the likelihood to be Gaussian with the covariance given by the kernel function for each day and assumes independency over different days. We use the following kernel functions

$$k_{\text{noise}}(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j) = \sigma_{\text{noise},i}^2 \delta_{i,j},$$
$$k_{\text{linear}}(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j) = \sigma^2 \boldsymbol{\beta}_i^T \boldsymbol{\beta}_j, \tag{18}$$

and the stationary kernels

$$k_{\text{se}}(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j) = k_{\text{se}}(d_{ij}) = \exp\left(-\frac{1}{2l^2}d_{ij}^2\right),$$
$$k_{\text{exp}}(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j) = k_{\text{exp}}(d_{ij}) = \exp\left(-\frac{1}{2l}d_{ij}\right),$$
$$k_{\text{m32}}(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j) = k_{\text{m32}}(d_{ij}) = \left(1 + \frac{\sqrt{3}d_{ij}}{l}\right)\exp\left(-\frac{\sqrt{3}}{2l}d_{ij}\right), \tag{19}$$

where $d_{ij} = ||\boldsymbol{\beta}_i - \boldsymbol{\beta}_j||_2$ is the Euclidean distance between $\boldsymbol{\beta}_i$ and $\boldsymbol{\beta}_j$. σ^2 is the kernel variance and l kernel lengthscale. Note that since the diagonal elements of stationary kernels are the same, they are not well suited for an estimation of a covariance matrix between different financial assets. Therefore, in the case of stationary kernel we decompose our covariance matrix $\boldsymbol{K}_{\text{cov}}$ into a vector of coefficient scales $\boldsymbol{\sigma}$ and a correlation matrix $\boldsymbol{K}_{\text{corr}}$, such that $\boldsymbol{K}_{\text{cov}} = \boldsymbol{\Sigma}\boldsymbol{K}_{\text{corr}}\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma}$ is a diagonal matrix with $\boldsymbol{\sigma}$ on the diagonal. The full kernel function $k(\cdot, \cdot)$ at the end is the sum of the noise kernel k_{noise} and one of the other kernels. In matrix form we get

$$\boldsymbol{K}_{\text{linear}} = k_{\text{linear}}(\boldsymbol{B}, \boldsymbol{B}) + k_{\text{noise}}(\boldsymbol{B}, \boldsymbol{B}),$$
$$\boldsymbol{K}_{\text{se}} = \boldsymbol{\Sigma}\,k_{\text{se}}(\boldsymbol{B}, \boldsymbol{B})\,\boldsymbol{\Sigma} + k_{\text{noise}}(\boldsymbol{B}, \boldsymbol{B}),$$
$$\boldsymbol{K}_{\text{exp}} = \boldsymbol{\Sigma}\,k_{\text{exp}}(\boldsymbol{B}, \boldsymbol{B})\,\boldsymbol{\Sigma} + k_{\text{noise}}(\boldsymbol{B}, \boldsymbol{B}),$$
$$\boldsymbol{K}_{\text{m32}} = \boldsymbol{\Sigma}\,k_{\text{m32}}(\boldsymbol{B}, \boldsymbol{B})\,\boldsymbol{\Sigma} + k_{\text{noise}}(\boldsymbol{B}, \boldsymbol{B}), \tag{20}$$

where $\boldsymbol{B} = (\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_N)^T$. We chose the following priors

$$\boldsymbol{B} \sim \mathcal{N}(0, 1) \qquad l, \sigma \sim \text{InvGamma}(3, 1) \qquad \sigma, \sigma_{noise} \sim \mathcal{N}(0, 0.5). \tag{21}$$

The prior on $\boldsymbol{B}$ determines how much space is allocated to the points in the latent space. The volume of this space expands with higher latent space dimension, which make the model prone to overfitting. To cope with that, we assign an inverse gamma prior to the lengthscale l and σ (σ in the linear kernel has a similar functionality as l in the stationary kernels). It allows larger values for higher latent space dimension, thereby shrinking the effective latent space volume and exponentially suppresses very small values, which deters overfitting as well. The parameters of the priors are chosen such that they allow for enough volume in the latent space for roughly 100-150 datapoints, which we use

---

[4]We are aware that this introduces survivorship bias. Thus, in some applications our results might be overly optimistic. Nevertheless, we expect relative comparisons between different models to be meaningful.
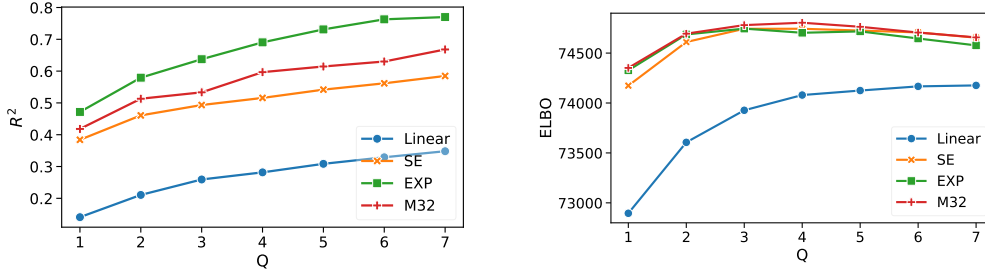
Figure 1: Left: R^2-score as a function of the latent dimension Q for different kernel functions. Right: ELBO as a function of the latent dimension Q. We randomly chose 120 stocks from the S&P500 and made the analysis on returns from Jan 2017 to Dec 2017.

in our analysis. If the number of points is drastically different, one should adjust the parameters accordingly. The kernel standard deviations σ_{noise} and $\boldsymbol{\sigma}$ are assigned a half Gaussian prior with variance 0.5, which is essentially a flat prior, since the returns are rarely above 0.1 for a day.

Model inference under these specifications (GP-LVM likelihood for the data and prior for $\boldsymbol{B}$ and all kernel hyperparameters $\sigma, \sigma_{\text{noise}}, l$ and $\boldsymbol{\sigma}$, which we denote by $\boldsymbol{\theta}$) was carried out by variational inference as described in section 2.3. To this end, we implemented all models in the probabilistic programming language Stan (Carpenter et al., 2017), which supports variational inference out of the box. The source code is available on Github (link anonymized for review). We tested different initializations for the parameter (random, PCA solution and Isomap solution), but there were no significant differences in the ELBO. So, we started the inference 50 times with random initializations and took the result with highest ELBO for each kernel function and Q.

4.2 Model Comparison

The GP-LVM can be evaluated in many different ways. Since, it projects the data from a D-dimensional space to a Q-dimensional latent space, we can look at the reconstruction error. A suitable measure of the reconstruction error is the R-squared (R^2) score, which is equal to one if there is no reconstruction error and decreases if the reconstruction error increases. It is defined by

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}, \tag{22}$$

where $y = (y_1, ..., y_N)^T$ are the true values, $f = (f_1, ... f_N)^T$ are the predicted values and $\bar{y}$ is the mean of y. In the following, we look at the R^2 as a function of the latent dimension Q for different kernels. Figure 1 (left plot) shows the results for three non-linear kernels and the linear kernel. Only a single dimension in the non-linear case can already capture almost 50% of the structure, whereas the linear kernel is at 15%. As one would expect, the higher Q, the more structure can be learned. But at some point the model will also start learning the noise and overfit.

As introduced in section 2.3, the ELBO is a good measure to evaluate different models and already incorporates models complexity (overfitting). Figure 1 (right plot) shows the ELBO as a function of the latent dimension Q. Here we see, that the model selects just a few latent dimensions. Depending on the used kernel, latent dimensions from three to five are already enough to capture the structure. If we increase the dimensions further, the ELBO starts dropping, which is a sign of overfitting. As can be seen from Figure 1, we do not need to go to higher dimensions. Q between two and five is already good enough and captures the structure that can be captured by the model.

4.3 Applications

The GP-LVM provides us the covariance matrix $\boldsymbol{K}$ and the latent space representation $\boldsymbol{B}$ of the data. We can build a lot of nice applications upon that, some of which are discussed in this section.

4.3.1 Portfolio Allocation

After inferring $\boldsymbol{B}$ and $\boldsymbol{\theta}$, we can reconstruct the covariance matrix $\boldsymbol{K}$ using equation (20). Thereafter, we only need to minimize equation (17), which provides the weights $\boldsymbol{w}$ for our portfolio in the

future. Minimization of (17) is done under the constraints: $\sum_n \boldsymbol{w}_n = 1$ and $0 < \boldsymbol{w}_n < 0.1, \forall n$. These constraints are commonly employed in practice and ensure that we are fully invested, take on long positions only and prohibit too much weight for a single asset.

For our tests, we proceed as follows: First, we get data for 60 randomly selected stocks from the S&P500 from Jan 2008 to Jan 2018. Then, we learn $\boldsymbol{w}$ from the past year and buy accordingly for the next six months. Starting from Jan 2008, this procedure is repeated every six months. We calculate the average return, standard deviation and the Sharpe ratio. Sharpe (1966) suggested the Sharpe ratio as a measure for a portfolio's performance, which is the average return earned per unit of volatility and can be calculated by dividing the mean return of a series by its standard deviation. Table 1 shows the performance of the portfolio for different kernels for $Q = 3$. For the GP-LVM we chose the linear, SE, EXP and M32 kernels and included the performance given by the sample covariance matrix, i.e. $\boldsymbol{K} = \frac{1}{D}(\boldsymbol{r} - \hat{\boldsymbol{\mu}})(\boldsymbol{r} - \hat{\boldsymbol{\mu}})^T$, where $\hat{\boldsymbol{\mu}}_n = \frac{1}{D}\sum_{d=1}^{D} \boldsymbol{r}_{nd}$, the shrunk Ledoit-Wolf covariance matrix[5] (Ledoit & Wolf, 2004) and the equally weighted portfolio, where $\boldsymbol{w} = (1, 1, ..., 1)/N$.

Table 1: Mean returns, standard deviation and the Sharpe ratio of different models on a yearly basis.

Model	Linear	SE	EXP	M32	Sample Cov	Ledoit Wolf	Eq. Weighted
Mean	0.142	0.151	0.155	0.158	0.149	0.148	0.182
Std	0.158	0.156	0.154	0.153	0.159	0.159	0.232
Sharpe ratio	0.901	0.969	1.008	1.029	0.934	0.931	0.786

Non-linear kernels have the minimal variance and at the same time the best Sharpe ratio values. Note that we are building a minimal variance portfolio and therefore not maximizing the mean returns as explained in section 3.2. For finite q in equation (16) one can also build portfolios, which not only minimize risk but also maximize the returns. Another requirement for that would be to have a good estimator for the expected return as well.

4.3.2 FILL IN MISSING VALUES

Regulation requires fair value assessment of all assets (Financial Accounting Standards Board, 2006), including illiquid and infrequently traded ones. Here, we demonstrate how the GP-LVM could be used to fill-in missing prices, e.g. if no trading took place. For illustration purposes, we continue working with our daily close prices of stocks from the S&P500, but the procedure can be applied to any other asset class and time resolution.

First, we split our data $\boldsymbol{r}$ into training and test set. The test set contains days where the returns of assets are missing and our goal is to accurately predict the return. In our case, we use stock data from Jan 2017 to Oct 2017 for training and Nov 2017 to Dez 2017 to test. The latent space $\boldsymbol{B}$ and the hyperparameters $\boldsymbol{\theta}$ are learned from the training set. Given $\boldsymbol{B}$ and $\boldsymbol{\theta}$, we can use the standard GP equations (eq. (5) and (6)) to get the posterior return distribution. A suggestion for the value of the return at a particular day can be made by the mean of this distribution. Given N stocks for a particular day d, we fit a GP to $N - 1$ stocks and predict the value of the remaining stock. We repeat the process N times, each time leaving out a different stock (Leave-one-out cross-validation). Figure 2 shows the R^2-score and the average absolute deviation of the suggested return to the real return. The average is build over all stocks and all days in the test set.

The predictions with just the historical mean have a negative R^2-score. The linear model is better than that. But if we switch to non-linear kernels, we can even further increase the prediction score. For Q between 2 and 4 we obtain the best results. Note that to make a decent suggestion for the return of an asset, there must be some correlated assets in the data as well. Otherwise, the model has no information at all about the asset, we want to predict the returns for.

4.3.3 INTERPRETATION OF THE LATENT SPACE

The 2-dimensional latent space can be visualized as a scatter plot. For a stationary kernel function like the SE, the distance between the stocks is directly related to their correlation. In this case, the

---

[5]Here, we have used the implementation in the Python toolbox scikit-learn (Pedregosa et al., 2011).
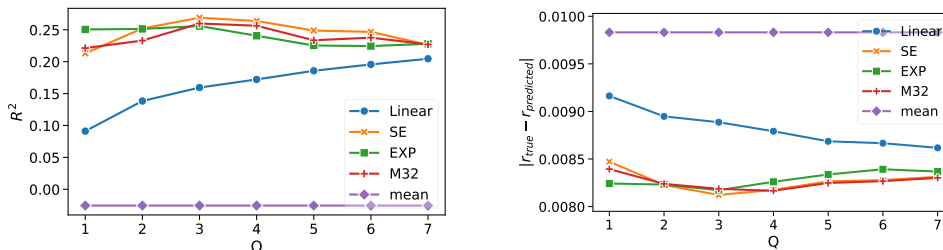
Figure 2: Left: R^2-score of the predicted values as a function of the latent dimension Q. Right: The average absolute deviation of suggested return to the real return evaluated by Leaving-one-out cross-validation. Historical mean is indicated by "mean" and is Q-independent.
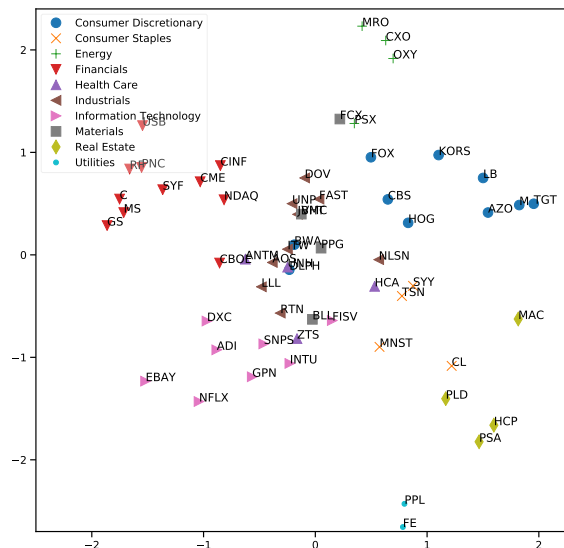

Figure 3: Stocks visualised in the 2-D latent space for the exponential kernel.

latent positions are even easier to interpret than market betas. As an example, Figure 3 shows the 2-D latent space from 60 randomly selected stocks from the S&P500 from Jan 2017 to Dec 2017. Visually stocks from the same sector tend to cluster together and we consider our method as an alternative to other methods for detecting structure in financial data (Tumminello et al., 2010).

5 CONCLUSION

We applied the Gaussian Process Latent Variable Model (GP-LVM) to estimate the covariance matrix between different assets, given their time series. We then showed how the GP-LVM can be seen as a non-linear extension to the CAPM with latent factors. Based on the R^2-score and the ELBO, we concluded, that for fixed latent space dimension Q, every non-linear kernel can capture more structure than the linear one.

The estimated covariance matrix helps us to build a minimal risk portfolio according to Markowitz Portfolio theory. We evaluated the performance of different models on the S&P500 from year 2008 to 2018. Again, non-linear kernels had lower risk in the suggested portfolio and higher Sharpe ratios than the linear kernel and the baseline measures. Furthermore, we showed how to use the GP-LVM to fill in missing prices of less frequently traded assets and we discussed the role of the latent positions of the assets. In the future, one could also put a Gaussian Process on the latent positions and allow them to vary in time, which would lead to a time-dependent covariance matrix.

REFERENCES

Bob Carpenter, Andrew Gelman, Matthew Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software, Articles*, 76(1):1–32, 2017. ISSN 1548-7660.

Nicolas Chapados and Yoshua Bengio. Augmented functional time series representation and forecasting with gaussian processes. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis (eds.), *Advances in Neural Information Processing Systems 20*, pp. 265–272. Curran Associates, Inc., 2008.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA, 1991. ISBN 0-471-06259-6.

Financial Accounting Standards Board. Statement of financial accounting standards no. 157 – fair value measurements, 2006. Norwalk, CT.

Edward Gately. *Neural Networks for Financial Forecasting*. John Wiley & Sons, Inc., New York, NY, USA, 1995. ISBN 0471112127.

J. B. Heaton, N. G. Polson, and J. H. Witte. Deep Portfolio Theory. Papers 1605.07230, arXiv.org, May 2016.

J. D. Jobson and Robert M Korkie. Putting markowitz theory to work. *The Journal of Portfolio Management*, 7(4):70–74, 1981. ISSN 0095-4918.

Neil Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *J. Mach. Learn. Res.*, 6:1783–1816, December 2005. ISSN 1532-4435.

Olivier Ledoit and Michael Wolf. Honey, I shrunk the sample covariance matrix. *The Journal of Portfolio Management*, 30(4):110–119, 2004. ISSN 0095-4918.

Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. ISSN 00221082, 15406261.

Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 673–680, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT Press, 2006.

Richard Roll and Stephen Ross. The arbitrage pricing theory approach to strategic portfolio planning. *Financial Analysts Journal*, 51:122–131, Jan 1995.

Stephen A Ross. The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13(3): 341 – 360, 1976. ISSN 0022-0531.

William F. Sharpe. Mutual fund performance. *The Journal of Business*, 39(1):119–138, 1966. ISSN 00219398, 15375374.

William F. Sharpe. Capital asset prices with and without negative holdings. *The Journal of Finance*, 46(2):489–509, 1991.

Michael E. Tipping and Chris M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

Michalis Titsias and Neil D. Lawrence. Bayesian gaussian process latent variable model. *Journal of Machine Learning Research*, 9:844–851, Jan 2010.

Michele Tumminello, Fabrizio Lillo, and Rosario N. Mantegna. Correlation, hierarchies, and networks in financial markets. *Journal of Economic Behavior & Organization*, 75:40–58, 2010.

Andrew Gordon Wilson and Zoubin Ghahramani. Generalised wishart processes. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, pp. 736–744, Arlington, Virginia, United States, 2011. AUAI Press. ISBN 978-0-9749039-7-2.

Yue Wu, José Miguel Hernández-Lobato, and Zoubin Ghahramani. Gaussian process volatility model. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 1044–1052. Curran Associates, Inc., 2014.