Heteroskedastic Bayesian Regression With Ranked Data Points

Marvin Meng MLM821@NYU.EDU
New York University

Guido (Sid) Petri GP1655@NYU.EDU
New York University

Vicente Javier Gomez Herrera VGH225@NYU.EDU
New York University

Abstract

In frequentist circles, estimating parameters via datapoints that have different quality or accuracy is done via weighting the datapoints. This principle cannot directly be applied in a Bayesian setting, where datapoints are either accounted for or not. In our project, we have developed a Bayesian model that accounts for different trustworthiness or relevance of datapoints. This could be used, for instance, for election forecasts that account for different pollster quality ratings. Using synthetic data, we successfully reconstructed the noise parameters for non-parametric variance, albeit with imperfect models.

1. Introduction

For statistical learning, we require data of the highest possible quality. However, it is often the case that the data has a mix of "high-quality" and "low-quality" data points. In this work, we approach the problem of using a mixed-quality dataset for heteroskedastic Bayesian regression. Our solution is composed of a single model, which can be trained in a single step.

A lower quality data point can be defined as an approximation of a true value with a large addition of noise. This can be seen, for instance, in election and polling data: smaller samples, which are of lower quality, will have more noise in their vote share estimates compared to the largest sample of the election itself. In contrast, a higher quality data point will have a lower amount of noise, and thus we can state its value with a higher degree of certainty.

Bayesian Machine Learning, New York University, NY, USA, 2020.

In a classical frequentist framework, if we have a probabilistic model such that $y_i = \mu + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$, then the Gauss-Markov theorem tells us that $\hat{\mu}$ is the linear estimator of least variance, where $\hat{\mu}$ is given by:

$$\hat{\mu} = \frac{1}{\sum_i \sigma_i^{-2}} \sum_i \frac{y_i}{\sigma_i^2}$$

In classical statistics, this is called *weighted least squares* (WLS); the Bayesian equivalent is named *Bayesian WLS*[6]. However, in most cases, we do not have observations of the σ_i values. Instead, we might only observe a quantity c, which acts as a *confidence score* for each data point. The parameter of interest, variance in this case, relates to c by some function $g(c)$, which may not have a linear form. The confidence score is usually not observed directly, but can be scaled according to other parameters, such as Glicko rating deviation[10] or poll sample size. Le et al. [7] developed a regression model to adjust mean and variance locally in space, but they model the variance directly instead of via a confidence score. We have developed a method to do parameter estimation and regression over a dataset with varying quality, where the variance of the measurement is not directly known, instead using a confidence score.

2. Problem formulation

Given the dataset $\mathcal{D} = \{(y_1, x_1, c_1), \ldots, (y_n, x_n, c_n)\}$ such that $y(x, c) = f(x) + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma^2) = \mathcal{N}(0, g(c))$, we want to reconstruct f and g. $g(c)$ is some function that relates σ^2 to c: $\sigma^2 = g(c)$.

f can be modeled via the standard methodologies of linear regression or Gaussian Processes. g, however, must follow certain restrictions. First, it is necessary that $g(c) > 0 \; \forall \; c$, as a variance must be positive. Second, we enforce that $g(c)$ be monotonically increasing so that a higher score c translates to a higher variance. As a result, we must con-

struct g such that $\frac{\partial g(c)}{\partial c} \geq 0 \ \forall \ c$. This choice is arbitrary: it is also possible to construct g such that its derivative is non-positive, reversing the relation between c and variance.

3. Related Works

Frequentist approaches have been studied in applications for finance and sports, show in papers by Lenčiauskas et al (2013)[2] and Varin et al (2012)[3].

Shu et al (2019)[13] have developed complex methods of sample weighting for deep neural network input. This mainly consists of a single-hidden-layer MLP which learns the sample weights dynamically as the target network learns its representation of the data.

Wang et al (2019)[12] uses a more mathematical model of derivative manipulation in order to weight samples used as inputs to a deep neural network. The derivative manipulation replaces the loss function and directly modifies the gradient magnitude in different input dimensions, effectively achieving a form of per-dimension sample reweighting during the network's training procedure.

Some sophisticated Bayesian methods have been explored by Dehghani et. al (2018) in their paper *Fidelity-Weighted Learning*[1]. Their method uses a "student" neural network and a "teacher" Gaussian Process (GP) to learn from weak, low confidence data, and strong, high confidence data, respectively.

Our method does not rely on neural networks and their relatively large dataset requirements. Rather, it can be used for datasets in the tens of datapoints instead of hundreds. As a result, our method is an attractive alternative when there is very little data available for a problem.

4. Methodology

We can separate the problem into two sub-problems. In the first case, assume a known parametric form of $g(c)$. In this case the objective is the discovery of optimal parameter values. In the second case, we do not know the functional form of $g(c)$, and thus use a non-parametric method to for its estimation.

The advantage of using a parametric approach to the variance function estimation is that it is easy to optimize the parameters, since we already know what form the function takes. However, if we assume an incorrect functional form, the parameter estimation will fail. On the other hand, using a non-parametric method allows us to not have to specify a functional form, but involves more computation and more uncertainty.

Each of the variance sub-problems can then be divided into two sub-problems as well: We can assume to know the functional form of the regression, $f(x)$, and use a parametric regression method, or assume no prior knowledge and approach the regression with a non-parametric method such as using Gaussian Processes (*GP*s).

In the following sections, we explore each of these possibilities.

4.1. Parametric variance

4.1.1. Parametric regression, parametric variance

Let $y_i = w \cdot \phi(x_i) + \epsilon(c_i)$, such that $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with $\sigma^2(c) = g(c, v)$ and $\phi(\cdot)$ a vector of linearly independent functions. In this example we use $g(c, v) = v_0^2 + v_1^2 c^2$. w, v are the parameters to be determined. The likelihood of this process is given by

$$\ell(w, v) = -\frac{1}{2} \sum_{i=1}^{N} \frac{(w \cdot f(x) - y_i)^2}{g(c_i, v)} - \log g(c_i, v) \quad (1)$$

We define the diagonal matrix G with the Kronecker delta, $\delta_{i,j}$:

$$[G]_{i,j} = g(c_i, v)\delta_{i,j}$$

We also define the design matrix Φ:

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \phi_1(x_2) & ... & \phi_1(x_n) \\ \phi_2(x_1) & \phi_2(x_2) & ... & \phi_2(x_n) \\ & ..... & & \\ \phi_m(x_1) & \phi_m(x_2) & ... & \phi_m(x_n) \end{bmatrix}$$

Y is the vector of all targets y_i. The gradient of the likelihood for w and v is then

$$\frac{\partial \ell}{\partial w} = \Phi^T G^{-1}(\Phi w - Y) \quad (2)$$

$$\begin{aligned} \frac{\partial \ell}{\partial v_j} =& \frac{1}{2} \sum_{i=1}^{N} \frac{(w \cdot f(x_i) - y_i)}{g^2(c_i, v)} \frac{\partial g(c_i, v_j)}{\partial v_j} \\ & - \frac{1}{2} \sum_{i=1}^{N} \frac{1}{g(c_i, v)} \frac{\partial g(c_i, v_j)}{\partial v_j} \end{aligned} \quad (3)$$

The above can be optimized directly via a minimization function, or, alternatively, via the matrix inversion method:

$$\frac{\partial \ell}{\partial w} = 0 \iff w = (\Phi^T G^{-1} \Phi)^{-1} \Phi^T G^{-1} Y.$$

With $H = \Phi(\Phi^T G^{-1}\Phi)\Phi^T G^{-1}$, then the maximum likelihood is given by

$$\tilde{\ell}(v) = -\frac{1}{2}||(G^{-1} - H)Y||^2 - \sum_{i=1}^{N} \log g(c_i, v). \quad (4)$$

4.1.2. Non-parametric regression, parametric variance

As a generalization, we can consider that $y = f(x) + \epsilon(c)$, where $f \sim \mathcal{GP}(0, k_\theta(x, x'))$, and $\epsilon \sim N(0, g(c, v))$. Here, θ are the parameters of the covariance kernel, and v are the parameters of $g(c, v)$ as in 4.1.1.

Note that if $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$, we can consider that y is a Gaussian process from $y : \mathbb{R}^{n+1} \to \mathbb{R}$ such that $y \sim \mathcal{GP}(0, k_\theta(x, x')) + \tilde{g}(c, c')$ and $\tilde{g}_v(c, c') = g(c, v)$ if and only if $c = c'$.

The marginal log likelihood for this model is given by

$$\begin{aligned}\log p(y|\theta, v) = &-\frac{1}{2} y^T (K_\theta + G)^{-1} y \\ &-\frac{1}{2}\log|K_\theta + G| - \frac{N}{2}\log(2\pi)\end{aligned} \quad (5)$$

Where K_θ is the kernel matrix of the GP. Let $\beta = (K_\theta + G)^{-1} y$. The gradients are then

$$\frac{\partial \ell}{\partial \theta_j} = \frac{1}{2}\text{Tr}(\beta\beta^T - (K_\theta + G)^{-1}\frac{\partial K_\theta}{\partial \theta_j}) \quad (6)$$

$$\frac{\partial \ell}{\partial v_j} = \frac{1}{2}\text{Tr}(\beta\beta^T - (K_\theta + G)^{-1}\frac{\partial G}{\partial v_j}) \quad (7)$$

4.2. Non-parametric variance

As a further generalization, we can consider the case of non-parametric variance. If we want to consider a non-parametric regression for g, we need to enforce the restrictions of monotonicity and positivity. This is not achievable through a classical GP, but it is possible through alternative models, such as from Andersen et al [8]. Let $\varphi(x) \sim GP(0, k^g_\omega)$ and $h : \mathbb{R} \to R_+$, with a GP φ, where ω is a set of hyperparameters. By construction, the random process

$$g(c) = g_0 + \int_0^c h(\varphi(c'))dc'$$

has the desired properties, with the condition $g_0 > 0$.

Let $h(x) = x^2$ and a stationary kernel for φ, it follows

$$\mathbb{E}[g(c)] = \mathbb{E}[g_0] + \eta c \quad (8)$$

$$\eta = \mathbb{V}\text{ar}[\varphi(c)]. \quad (9)$$

Note that since $\eta = \frac{\mathbb{E}[g(c)] - \mathbb{E}[g(0)]}{c - 0}$, it can be considered as the prior slope of the process.

If $S_\omega(\xi) = \int_R k^g_\omega(\tau) e^{-2\pi i \xi \tau} d\tau$, i.e. the spectral density[11] of the k_{θ_2}, then we can approximate ϕ in the domain $(-L, L)$[9] by considering the following eigenvalue problem:

$$\begin{cases} -\Delta\phi_j(x) = \lambda_j \phi_j(x), & x \in (-L, L) \\ \phi_j(x) = 0, & x \notin (-L, L) \end{cases} \quad (10)$$

which induces the approximation $\varphi(c) \approx \sum_{i=1}^N \alpha_i \phi_i(c)$, with $\alpha_i \sim \mathcal{N}(0, S_\omega(\lambda_i^{1/2}))$. In this case,

$$\phi_j(x) = \sqrt{\frac{1}{L}} \sin\left(\lambda_j^{1/2}(x + L)\right) \quad (11)$$

$$\lambda_j = \frac{j\pi}{2L} \quad (12)$$

In fact, $\varphi(c) = \lim_{L\to\infty} \lim_{N\to\infty} \sum_{i=1}^N \alpha_i \phi_i(c)$. We can see that large N and L will generate a good approximation of φ. Equivalently, this new Gaussian process can be reinterpreted as one that has basis functions ϕ_j with a decaying variance of $S_\omega(\lambda_j^{1/2}))$. Under that scope, L should be large enough so that the domain of interest φ is not affected by boundary effects, but small enough compared to N so that the number of functions is flexible enough to describe the function. For smooth functions, $N = 10$ and an L four times the domain of interest show good results.

Returning to the random process of interest, we have that

$$g(c) = g_0 + \boldsymbol{\alpha}^T \Psi(c) \boldsymbol{\alpha}$$

where

$$\Psi_{ij}(c) = \begin{cases} \frac{1}{2L}c - \frac{\sin(\gamma^+_{jj}(c'+L))}{2L\gamma^+_{jj}}\Big|_{c'=0}^{c}, & i = j \\ \frac{\sin(\gamma^-_{ij}(c'+L))}{2L\gamma^-_{ij}}\Big|_{c'=0}^{c} - \frac{\sin(\gamma^+_{ij}(c'+L))}{2L\gamma^+_{ij}}\Big|_{c'=0}^{c}, & i \neq j \end{cases}$$

for $\gamma^-_{ij} = \sqrt{\lambda_i} - \sqrt{\lambda_j}, \gamma^+_{ij} = \sqrt{\lambda_i} + \sqrt{\lambda_j}$. We can then place a prior over σ_0.

One final remark is that if $\Sigma_{|D} = \text{Cov}(\boldsymbol{\alpha}, \boldsymbol{\alpha}|D)$, then

$$\mathbb{E}[g|D] = \mathbb{E}[g_0|D] + \mathbb{E}[\boldsymbol{\alpha}^T|D]\Psi(c)\mathbb{E}[\boldsymbol{\alpha}^T|D] + \Sigma_{|D} : \Psi(c),$$

With ":" representing the Frobenius inner product.

4.2.1. Parametric regression, non-parametric variance

Given the same model $g(c)$ as in 4.2 and $y = \mu + \epsilon$, where $\epsilon|c, \sigma_0, \boldsymbol{\alpha} \sim \mathcal{N}(0, g(c))$, the marginal likelihood is given by

$$p(Y|\mu, \omega) = \int p(Y|\mu, \omega, g_0, \boldsymbol{\alpha}) p(g_0, \boldsymbol{\alpha}|\omega) d\boldsymbol{\alpha} dg_0 \quad (13)$$

In this case, the mean needs to be treated in the same way as a hyperparameter. The procedure to choose μ would be to approximate equation (10) and optimize over that approximation.

Nevertheless, this method for approximating the optimum of the evidence might not be reliable, since accurately approximating (10) is not an easy task, which is why usually this procedure is more targeted to tuning the model, rather than for doing inference, that is why we introduce a prior over μ. In that way we have that

$$p(Y|\omega) = \int p(Y|\mu, \omega, g_0, \boldsymbol{\alpha}) p(\mu) p(g_0, \boldsymbol{\alpha}|\omega) d\mu d\boldsymbol{\alpha} dg_0 \quad (14)$$

and a posterior

$$p(\mu, g_0, \boldsymbol{\alpha}) \propto p(Y|\mu, \omega, g_0, \boldsymbol{\alpha}) p(\mu) p(g_0, \boldsymbol{\alpha}|\omega)$$

4.2.2. Non-parametric regression, non-parametric variance

Given the model $y = f(x) + \epsilon(c)$, where f is the Gaussian Process $GP(0, k_\theta(x, x'))$, ϵ is the same as section 4.2, and the data is the set $\mathcal{D} = \{(y_1, x_1, c_1), \ldots, (y_n, x_n, c_n)\}$ the marginal likelihood is

$$p(Y|\theta, \omega) = \int p(Y|g_0, \boldsymbol{\alpha}, \theta, D) p(g_0, \boldsymbol{\alpha}|\omega) d\boldsymbol{\alpha} dg_0 \quad (15)$$

where we have summarized the dependencies on X, C. Note that $\log p(Y|X, \sigma_0, \boldsymbol{\alpha}, \theta, C)$ is the same as in (5), i.e. the marginal likelihood of a standard GP. Even further, note that $p(y|x, c, \mathcal{D}, \theta, \omega) = p(y|x, c, \mathcal{D}, g_0, \boldsymbol{\alpha}) p(g_0, \boldsymbol{\alpha}|\mathcal{D}, \theta, \omega)$ where $p(y|x, c, \mathcal{D}, g_0, \boldsymbol{\alpha}, \omega, \theta)$ is a known Gaussian (the posterior of a Gaussian process) and

$$p(g_0, \boldsymbol{\alpha}|\mathcal{D}, \theta, \omega) = \frac{p(Y|g_0, \boldsymbol{\alpha}, \theta, D) p(g_0, \boldsymbol{\alpha}|\omega)}{p(Y|\theta, \omega)}$$

As a result, we only need to approximate $p(g_0, \boldsymbol{\alpha}|\mathcal{D}, \theta, \omega)$.

4.2.3. Evaluating the evidence

Here we present some brief pseudo-code on how we approximate the evidence function for the posterior of a distribution $p(\beta|D, \theta)$, with likelihood $p(D|\beta, \theta)$ and prior $p(\beta|\theta)$:

Algorithm 1 Evaluating the evidence function

f(θ) :
$\beta^* = \text{argmax}\{\log(p(D|\beta, \theta)) + \log(p(\beta|\theta))\}$
$A = -\nabla \log(p(D|\beta, \theta)) + \log(p(\beta|\theta))\}|_{\beta^*}$
return $\log(p(D|\beta^*, \theta)) + \log(p(\beta^*|\theta)) - \frac{1}{2}\log(\det(2\pi A))$

5. Experiments

The experiments we ran are proofs of concept. We generated random data from a distribution with a known functional form between σ and c. Using this data, we verify whether we can recover the original function and its parameters. Given functions $f(x)$ and $g(c)$, for different combinations of x, c we can simulate the random process $f(x) + \epsilon(c)$.

A hyperlink to all of the following experiments can be found under https://github.com/MARVINMENG0/bayes-ml

5.1. Parametric variance

Assuming a parametric form for the variance, we try to recover the noise from the optimization problem discussed in 4.1.

5.1.1. Parametric Regression, Parametric Variance

We first attempt this with a parametric form of the regression f(x). The generated data has the form

$$Y = f(x) + \epsilon(c); \quad \epsilon \sim N(0, g(c)); \quad f(x) = \mu = 3$$

The function for the noise is

$$\sigma^2 = g(c) = v_0^2 + v_1^2 c + v_2^2 c^2$$

with values $v_0 = 0.7101423, v_1 = 0.79067559, v_2 =$

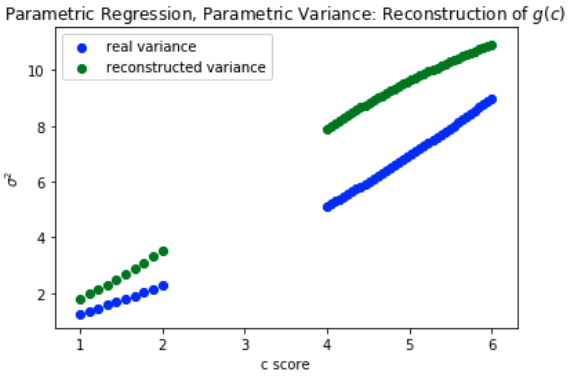

Figure 1. True variance values (blue) and reconstructed variance values (green).

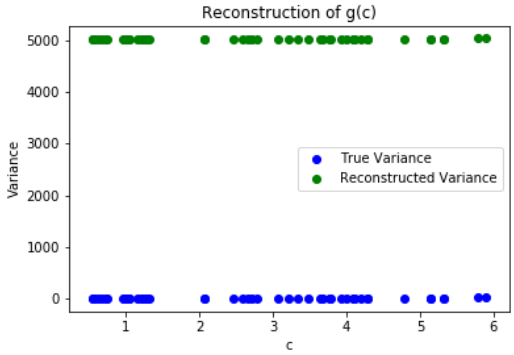

Figure 2. True variance values (blue) and reconstructed variance values (green).

0.36303851. In the experiment we use N = 50 generated data points.

In Figure 1, we show the true variance values in blue, and our reconstructed variance values in green. We see that even though we know the parametric form of the noise, the results are not optimal.

5.1.2. Non-Parametric Regression, Parametric Variance

We attempt another experiment with parametric variance, this time with a non-parametric regression. For this experiment, we now assume the data is generated by a zero-mean Gaussian Process: $y = f(x) + \epsilon(c)$ where $f \sim GP(0, k(x, x'))$. We chose k to be an RBF kernel of the form:

$$k(x, x') = a^2 \exp(-\frac{||x - x'||^2}{2L^2})$$

As before, we used the function $g(c, v) = v_0^2 + v_1^2 + v_2^2 c^2$ to relate the score with the variance. With the same values for v_0, v_1, v_2 as above, we generated 50 random samples from the uniform distribution for c. We set the lengthscale $L = 7.3$ and the kernel's output scale $a = 12.4$. The results are shown in Figure 2.

Interestingly, using a more flexible GP model sees the performance decrease.

5.2. Fully non-parametric inference

For this experiment, we take

$$g_0 \sim \text{Gamma}(\theta_1 = 3.0, \theta_2 = 3.0)$$
$$\mu \sim N(0, s^2 = 1)$$
$$S_\omega(\tau) = \omega_1^2 \exp(-\frac{\tau^2}{\omega_2^2})$$
$$L = 20$$
$$N = 10,$$

where $\omega = (\omega_1, \omega_2)$ and θ_1, θ_2 are hyperparameters. Algorithm 1 tells us how to evaluate the evidence function and we optimize it over the mentioned hyper parameters with a non-gradient optimization procedure. In our example, we used Powell optimization.

In this experiment we could also optimize over the parameters s^2, L, N. The parameter N is not easy to optimize over; however, we also expect that larger N will not contribute much new information, since the diagonal of the covariance matrix tends to 0. The L parameter is highly dependant on N. Finally, optimizing over s^2 could generate an overconfidence phenomenon. Additionally, evaluation of the evidence is computationally expensive, and so we want to minimize the hyperparameter space we optimize over.

Having optimized the hyperparameters, we find the MAP for $\mu, g_0, \boldsymbol{\alpha}$ and use that as the starting position for a Hamiltonian Monte Carlo (HMC) procedure. A total of 500 samples were taken, shown in Figure 4.

The reconstruction of the variance is shown in Figure 3.

5.3. Fully non-parametric inference, regression

For this experiment we generated a data sample from the random process

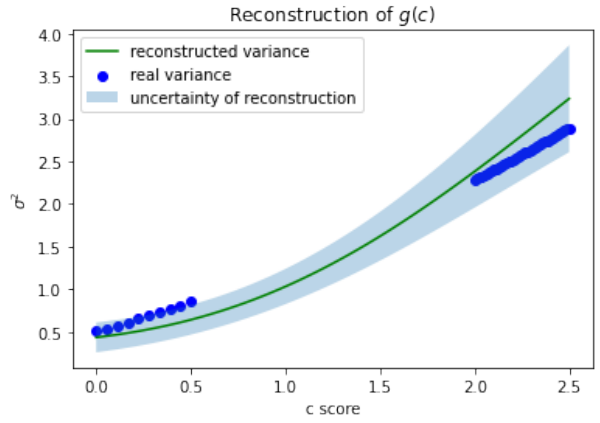

Figure 3. Reconstruction of the variance function of experiment 5.2. The real function values are in the uncertainty interval.

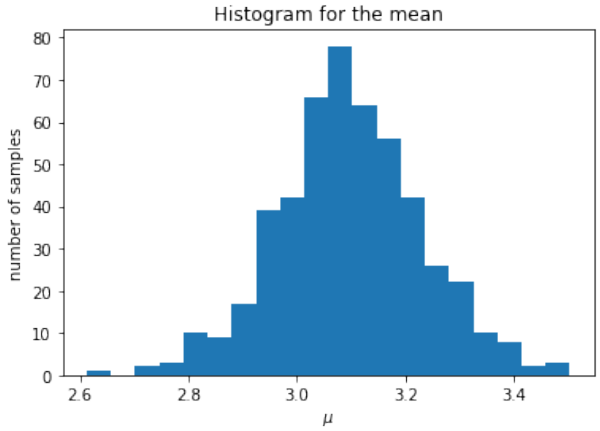

Figure 4. Samples of the HMC sampler of experiment 5.2. The x-axis shows the values of the μ parameter described in 5.2. The relative error of mean of this samples with respect to the real value is 0.032. A simple average has a relative error of 0.137.

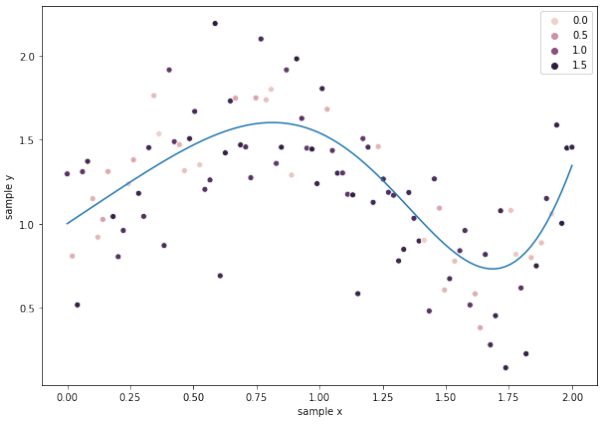

Figure 5. The blue line represents the real function. The datapoints' hue represents the sampled confidence score.

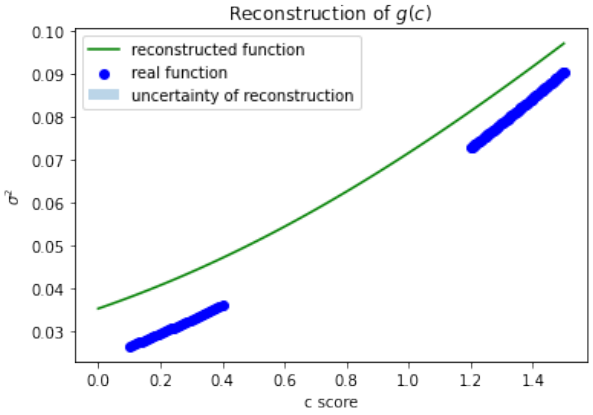

Figure 6. Reconstruction of the noise. Even though this reconstruction is in the same order of magnitude, it is overconfident in its prediction.

$$y = \exp(\cos(x^2)) + x + g_r(c)\epsilon$$

where $g_r(c)$ is the $g(c)/10$ of previous examples.

We took 100 uniform samples in $[0, 2]$ for x data, and 30 uniform samples from $[0.1, 0.4]$ and 70 from $[1.2, 1.5]$ for c. These 2 data sets where randomly matched, then we sampled from y. The samples and the real function can be seen in Figure 5.

For g, we kept the same prior and hyperparameters. For modelling $f(x)$ we used the model $f(x) = \sum_{n=0}^{7} A_n \cos(nx)$, such that $A_n \sim N(0, \frac{8}{(n+1)^2})$.

Note that for regression, $\text{Var}[y|D] = \text{Var}[f|D] + \mathbb{E}[g^2|D]$, so for plotting purposes we take $c = 0$. We took 1000 samples via the HMC process.

We can see from Figure 6 that we can reconstruct the noise up to some extent. However, the model is overconfident in the predictions of g, since the epistemic uncertainty is practically 0. Additionally, we reconstructed f, which can be seen in Figure 7.

6. Discussion

In our experiments we have shown that it is possible to reconstruct a heteroskedastic variance through mean estimation and parameter estimation. For the estimation of the mean, it is clear that this procedure gives better results than the standard case.

For the regression model, the reconstruction of the original function f was achieved to some extent. The mean processes do not perfectly match, but our reconstruction is still consistent with the data.

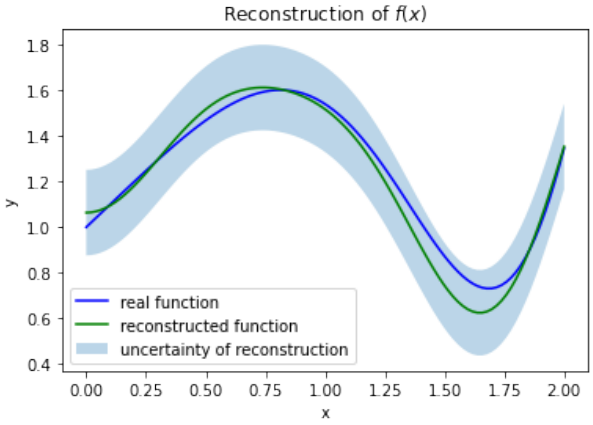

Figure 7. Reconstruction of f. Note that the uncertainty is mostly uniform, which means that most of the uncertainty in the prediction comes from the variance, and not an epistemic uncertainty.

In the same example, the noise was also reconstructed, but with less satisfying results than for the previous example, since the estimation is overconfident.

These problems can be due to the way that the optimization procedure is interpreting the data. This indicates that the problem might have local maxima in the posterior that are not being sampled, but rather the optimization procedure is sampling from only a single local maximum.

For future work, we believe it would be interesting to redo these experiments with different samplers such as stochastic Hamiltonian gradient methods. Another direction to investigate is to model the precision instead of the variance.

7. References

1: Dehghani, M., Mehrjou, A., Gouws, S., Kamps, J., Scholkopf, B. (2018) ICLR. *Fidelity-Weighted Learning.*

2. Raudys, A., Lenčiauskas, V., Malčius, E. (2013) ICIST. *Moving Averages for Financial Data Smoothing.*

3: Cattelan, M., Varin, C., Firth, D. (2012) Journal of the Royal Statistical Society: Series C (Applied Statistics), vol. 62, no. 1, pp. 135-150. *Dynamic Bradley–Terry modelling of sports tournaments.*

4: Nakanoa, M., Takahashi, A., Takahashi, S. (2017) Expert Systems with Applications, Vol. 73 (pp. 187-200). *Generalized exponential moving average (EMA) model with particle filtering and anomaly detection.*

5: Siruddin, M., Herdiani, E. T., Thamrin, S. A. (2019) IOP Conference Series: Earth and Environmental Science, 279:012012. *Estimation Mean by the Bayesian Approach on the Exponentially Weighted Moving Average Control Chart.*

6: Ting, J. A., D'Souza, A., Schaal, S. (2007) 2007 IEEE International Conference on Robotics and Automation (pp. 2489-2494). *Automatic outlier detection: A Bayesian approach.*

7: Le, Q. V., Smola, A. J., Canu, S. (2005) Proceedings of the 22nd international conference on Machine learning. *Heteroscedastic Gaussian process regression.*

8: Andersen, M. R., Siivola, E., Riutort-Mayol, G., Vehtari, A. (2018) 32nd Conference on Neural Information Processing Systems. *A non-parametric probabilistic model for monotonic functions.*

9: Solin, A., Särkkä, S. (2019) Journal of Statistics and Computing. *Hilbert Space Methods for Reduced-Rank Gaussian Process Regression.*

10: Glickman, M. Unknown year. *The Glicko System.*

11: Wilson, A. G., Adams, R. P. (2013) ICML. *Gaussian Process Kernels for Pattern Discovery and Extrapolation.*

12: Wang, X., Kodirov, E., Hua, Y., Robertson, N. M. (2019) Arxiv. *Derivative Manipulation for General Example Weighting.*

13: Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., Meng, D. (2019) NeurIPS 2019. *Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting*