Richard Creswell Meeting Notes

2020 January 28

1 Electrochemistry protocol optimization

1.1. I continued looking at the electrochemistry problem. The model parameter values are now more realistic and the initial protocol is a single sine wave with parameters taken from [1]. The latest results are shown in Figure 1 (single sine wave) and Figure 2 (sum of three sine waves).

1.2. In both cases, α and k_0 show large improvements in posterior uncertainty, while the other parameters are unaffected. The three sine wave protocol did not make much of a difference. A very low frequency component is visible in the optimized protocol, without much effect on any of the posteriors.

1.3. In [1], the objective function $\mathcal{L}_f$ was found to be independent of α. It seems strange to me that in these results, increasing the frequency and amplitude of the stimulus apparently can suddenly make α identifiable.

1.4. I added functionality to place weights on each parameter which then multiply the relevant portion of the objective function. However, even when ϵ_0 was weighted much more highly than any other parameter, the protocol optimization did not manage to improve its posterior uncertainty by any noticeable amount.

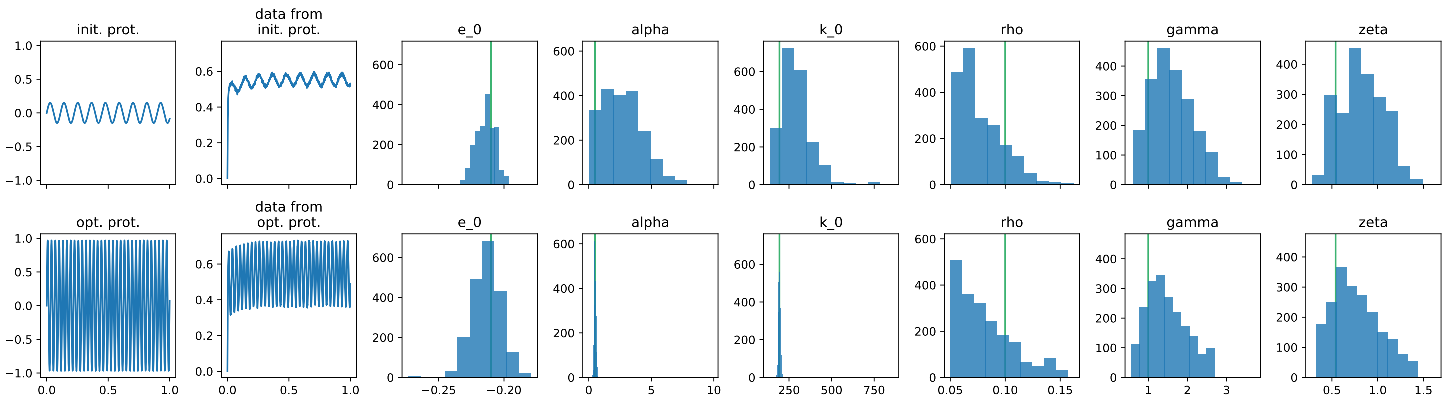


Figure 1: Protocol optimization in the electrochemistry problem. In this run, the parameters of a single sine wave protocol were optimized.

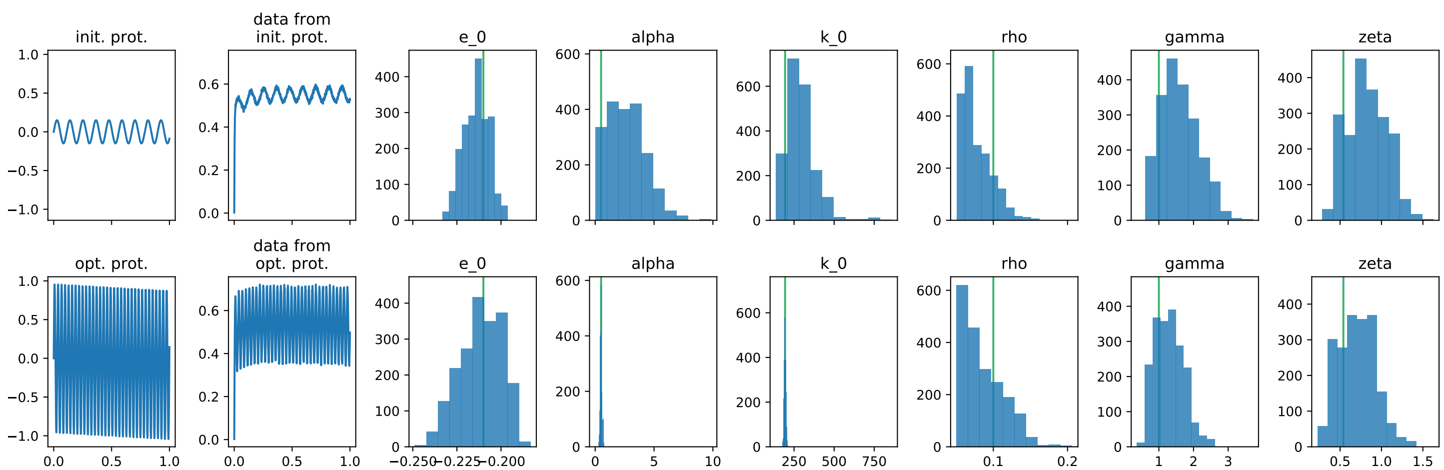


Figure 2: Protocol optimization in the electrochemistry problem. In this run, the parameters of a sum of three sine waves protocol were optimized.

2 General noise model

2.1. Using the general noise method discussed last week, I have been working on non-stationary kernels in order for the covariance matrix to be highly adaptive. [2] suggests the following non-stationary kernel, which they found useful for Gaussian process regression

$$k(x_i, x_j) = \sigma(x_i)\sigma(x_j)\sqrt{\frac{2l(x_i)l(x_j)}{l(x_i)^2 + l(x_j)^2}} \exp\left(-\frac{(x_i - x_j)^2}{l(x_i)^2 + l(x_j)^2}\right) \tag{1}$$

where $\sigma(x)$ and $l(x)$ are modelled with Gaussian processes.

2.2. I have implemented this kernel as a parametrization of the covariance matrix for use in the general noise model for ODEs and started testing it. The implementation uses Pints, with a new LogLikelihood class and the kernel parameter GP priors built up using pints.ComposedLogPrior.

2.3. The results for multiplicative noise are shown in Figure 3. I am happy with the way it scales the noise variance as needed over a wide range of magnitudes and avoids using any correlation, but there is an undesirable section of overfitting around $t = 80$. I am working on handling this better with the length scale prior on the $\sigma(x)$ GP. However, the question arises how much to penalize these "sharp" changes in the noise properties. If they were penalized too heavily, the method would not work well in cases where the noise form switches suddenly.

2.4. The results for AR1 noise are shown in Figure 4. As seen in the covariance matrix (b), this kernel is learning that the noise is correlated, but it does not seem to be adding enough correlation to fully capture AR1. Thus, the parameter inference results fall in between the correctly specified case and the incorrect iid assumption. It is possible that this kernel cannot satisfactorily express an AR1 covariance matrix given that the only possible dependence on distance $x_i - x_j$ is via the squared exponential (the AR1 covariance matrix was captured using a power form $ab^{|x_i - x_j|}$ previously). I am thinking of adjusting the kernel to replace the squared exponential with a power law of this type, adding another GP term as a sort of length scale. Although it does not quite get the correct parameter uncertainty yet, I was pleased to see that it captured the correlated noise with smooth, steady values of $\sigma(x)$ and $l(x)$ without overfitting to noisy segments.

2.5. The main challenge that I am still working on is the inference. Following the recommendation in the paper [2], I applied a transform ("posterior whitening") to the $l(x)$ and $\sigma(x)$, so that MCMC samples are actually taken from

$$\hat{l} = L_l^{-1} \log(l) \tag{2}$$

$$\hat{\sigma} = L_\sigma^{-1} \log(\sigma) \tag{3}$$

where L_l and L_σ are the Cholesky factors of the prior covariance matrices for the Gaussian processes specifying l and σ. This transform is helping but clearly does not make inference trivial.

2.6. Using adaptive covariance MCMC, the longest run I did was 60,000 iterations, and independent chains were not even close to converging to a common posterior. I have tried most of the MCMC samplers available in Pints, of which Hamiltonian MCMC seems to be the best but even Hamiltonian is slow to converge. The results below are from these Hamiltonian MCMC runs with 6000 iterations, which is sufficient for the model parameters to converge, but I still have some concerns about how robust the inference is for the kernel parameters.

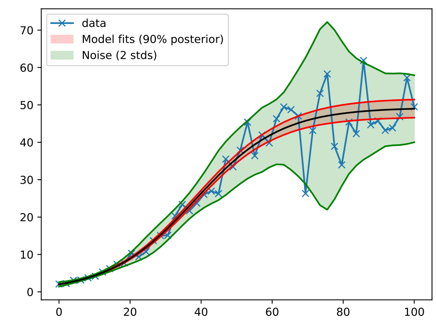
2.7. Meanwhile, [2] reports using just 1000 MCMC samples (although their problem is different, since they are doing GP regression and do not have an ODE model). They used a Hamiltonian-NUTS sampler—when NUTS is ready in Pints I will be keen to try it out and see if that helps.

2.8. I haven't started on the blocked covariance matrix idea yet. I did consider that the two approaches (blocks and non-stationary kernel) could actually be combined, so that each block was specified by a nonstationary kernel. This would allow sharp changes in the GPs for $l(x)$ and $\sigma(x)$ to be heavily penalized (2.3), while any true sharp changes in the noise format would be handled by a new block.

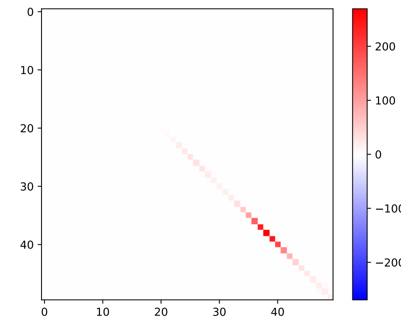
References

[1] Adamson, H. et al, "Analysis of HypD Disulfide Redox Chemistry via Optimization of Fourier Transformed ac Voltammetric Data" *Anal. Chem.* 2017, 89, 1565–1573.

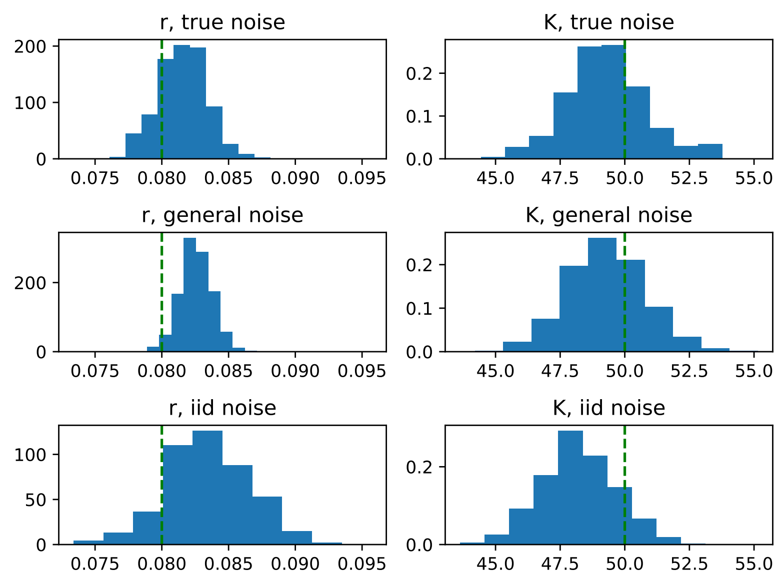
[2] Heinonen, Markus, et al. "Non-stationary gaussian process regression with hamiltonian monte carlo." Artificial Intelligence and Statistics. 2016.



(a)

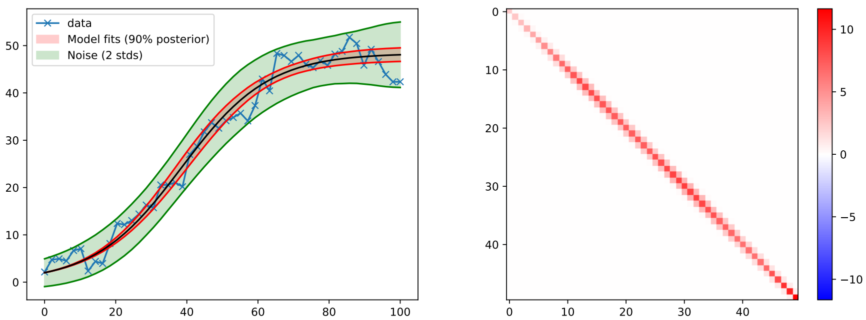


(b)

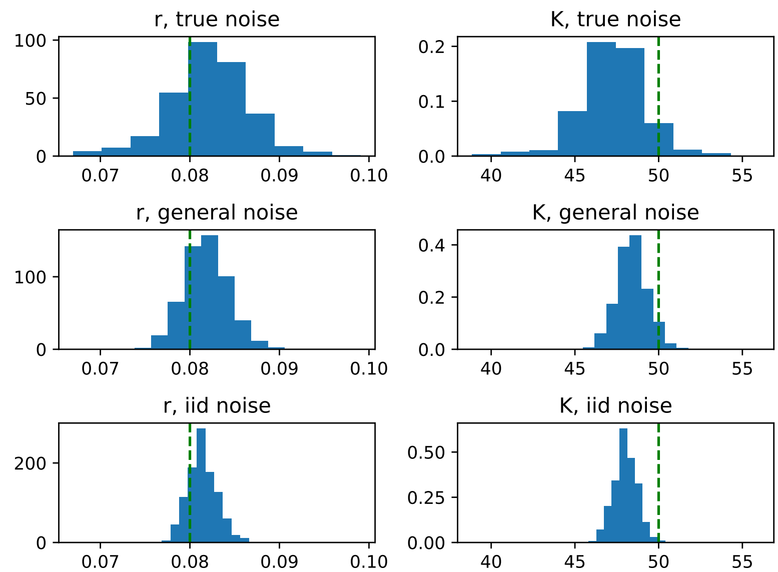


(c)

Figure 3: General noise model with multiplicative noise and logistic growth. (a) The data and general noise model fits, with the red lines showing the trajectories corresponding to the 90% central posterior interval of model parameters, and the green lines showing the $\pm$ 2 standard deviations of the learned noise parameters. (b) A representative covariance matrix learned by the general noise method. (c) The results for parameter inference with the correctly specified model, the general model, and the incorrect iid assumption.



(a) (b)



(c)

Figure 4: General noise model with AR1 noise and logistic growth. (a) The data and general noise model fits, with the red lines showing the trajectories corresponding to the 90% central posterior interval of model parameters, and the green lines showing the $\pm$ 2 standard deviations of the learned noise parameters. (b) A representative covariance matrix learned by the general noise method. (c) The results for parameter inference with the correctly specified model, the general model, and the incorrect iid assumption.