

Systematic testing of GPR kernels

Mark Blyth

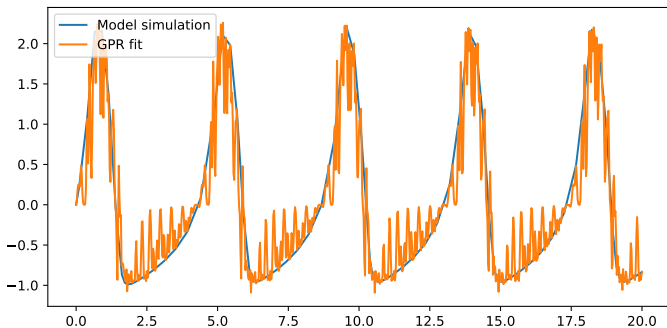
Week's goals

- ✶ Redraft paper
 - ▶ *[done]*
- ✶ Teaching stuff
 - ▶ *[done]*
- ✶ Do some more systematic testing of the GPR kernels
 - ▶ *[in progress]*

Kernel fitting

- ✎ Tried the kernels I coded, on several models
- ✎ Tried with and without noise
- ✎ Found some interesting results...

Good LL, bad fit

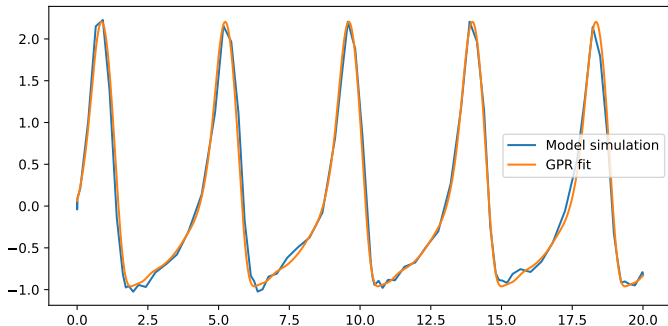


Log-likelihood = -132

$$\sigma_n^2 = 0$$

periodic kernel, HRFast model

Good LL, good fit

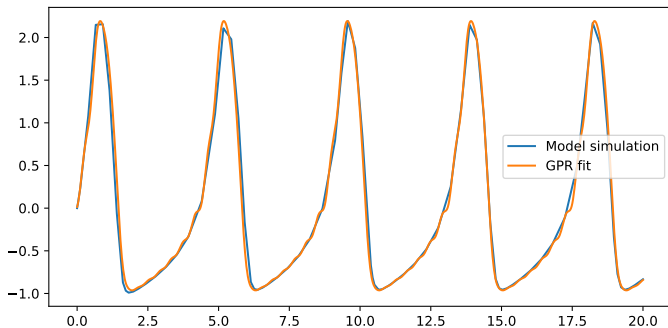


Log-likelihood = -100

$\sigma_n^2 = 0.05$

periodic kernel, HRFast model

Bad LL, good fit

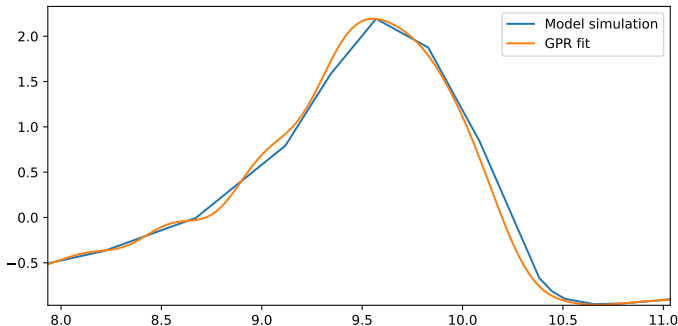


Log-likelihood = -377,038

$\sigma_n^2 = 0$

periodic kernel, HRFast model

How good is good enough?



Even the best-achievable fit isn't perfect; gets worse with more data

Log-likelihoods

	My SEKernel	Modulo kernel	Periodic kernel	σ_n^2
FitzhughNagumo	-289	-283	-290	0.1
HR fast	-277	-287	-284	0.05
Hodgkin Huxley	-6170	-2690	-2657	2

	My SEKernel	Modulo kernel	Periodic kernel
FitzhughNagumo	-135	-178	-170
HR fast	-114	-152	-132
Hodgkin Huxley	-8547	-9952	-9878

Good LL, bad fits

Log-likelihoods

	My SEKernel	Modulo kernel	Periodic kernel
FitzhughNagumo	-135	-457,451	-456,778
HR fast	-1,120,469	-145,659	-377,038
Hodgkin Huxley	-175,468,920	-39,092,086	-39,117,518

Good fits, bad LL

Log-likelihoods

- ✿ LL is meant to prevent overfitting, but it doesn't
- ✿ Can have a terrible fit and a good LL
- ✿ Can have a terrible LL and a good fit
- ✿ LL is not always a good measure of success

Causes of weirdness

- ✂ No noise means small deviations give large LL penalties
 - ▶ Noise-free models are fitted with $\sigma_n^2 = 0$
 - ▶ Overfitting happens: model tries too hard to match provided datapoints
 - ▶ Loses ability to generalise (becomes dippy)
- ✂ Noisy fits help
 - ▶ No exact datapoints, so model is forced to average out noise
 - ▶ Reduces the maximum goodness-of-fit, and forces the model to more cleverly estimate posterior
 - ▶ Less overfitting, so more generalisable, reducing dipping

Jitter

Overfitting can sometimes be cured by adding some noise

- ✂ Jitter is manually added noise
- ✂ Adding jitter has the potential to offset overfitting, by pretending there's noise when there isn't
- ✂ Can give a more sensible LL value
 - ▶ LL depends on the size of the jitter, impossible to choose a 'right' value [*not strictly true*]

Adding jitter isn't a useful method, as it degenerates into regular noisy data-fitting.

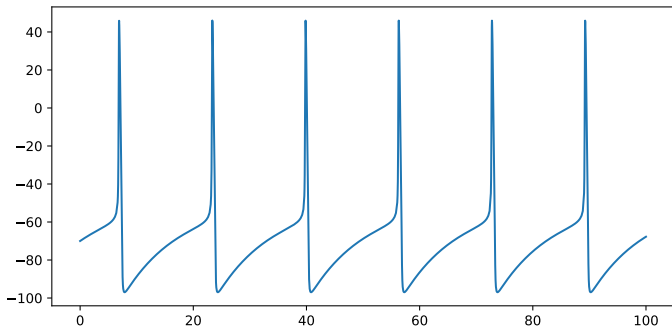
Best approach

- ✎ Add noise into data
- ✎ Fit other hyperparameters, with signal noise fixed at whatever noise variance we added in
 - ▶ Maximise log-likelihood
- ✎ Reuse these results on the noise-free case
- ✎ We can do this, as the fitted hyperparameters will take the same value, independent of clean or noisy signals

Fitting to noisy data gives the best possible results, regardless of what the LL says

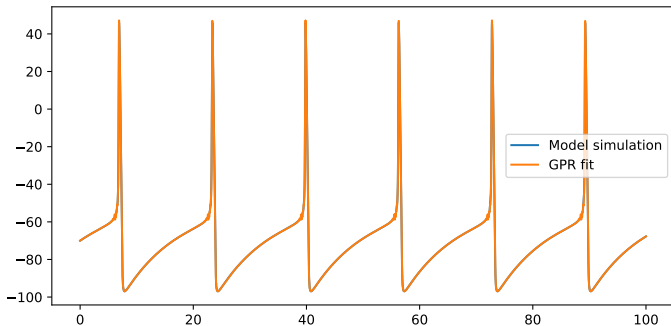
- ✎ It's still not always a good enough fit . .

Nonstationarity



Hodgkin-Huxley is a good test of the models – nonstationary, and realistic

Nonstationarity

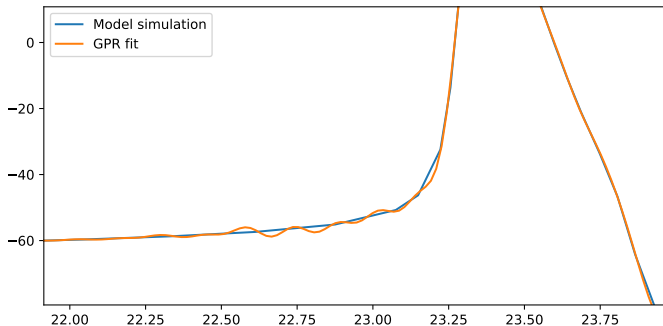


Looks good, right?

Fitted on $\sigma_n^2 = 2$

Tested on $\sigma_n^2 = 0$

Nonstationarity

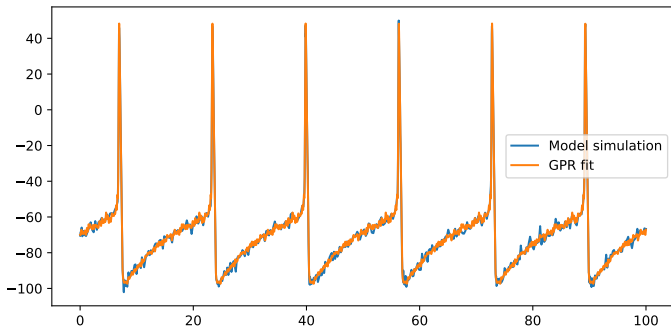


Still looks good

Fitted on $\sigma_n^2 = 2$

Tested on $\sigma_n^2 = 0$

Nonstationarity

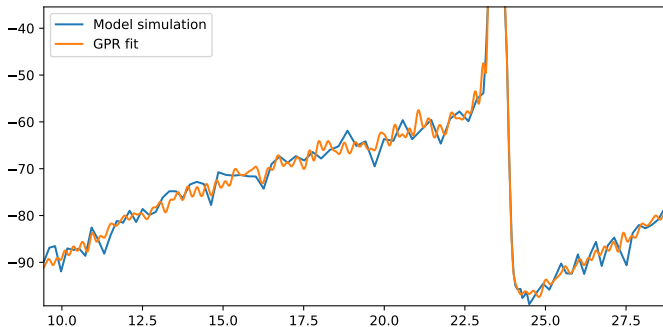


Uh oh...

Fitted on $\sigma_n^2 = 2$

Tested on $\sigma_n^2 = 2$

Nonstationarity



Uh oh...

Fitted on $\sigma_n^2 = 2$

Tested on $\sigma_n^2 = 2$

Nonstationarity

Hodgkin-Huxley fit loses the ability to average out noise

- ✦ Requires very small lengthscales to model spikes
- ✦ Small lengthscales overfit the noise
- ✦ We end up with *more* noise in the model than in the original signal!
 - ▶ Worse results by fitting a model

Nonstationarity would fix this!

Next steps: part 1

Other GPR methods:

- ✿ Function-space distribution over kernels
- ✿ Generalised spectral mixture kernels
- ✿ Switching kernel

Implement and test these on the shown datasets

Next steps: part 2

Lit-review of some other techniques I've spotted *[useful for a paper]*

Implement and test some other methods:

- ✿ GOAL: black-box data model *[parametric or non-parametric]*
- ✿ NARMAX *[simple nonlinear timeseries models]*
 - ▶ Popular in systems identification community
- ✿ Neural ODEs *[often combines NNs with GPs for latent ODE models]*
- ✿ Bayesian free-knot splines
 - ▶ Powerful for scalar signals!