(More) surrogate modelling

Mark Blyth

Week's goal

Keep working through different regression models:

- Function-space distribution over kernels
- Matern kernels
- Bayesian free-knot splines
- ~~Generalised spectral mixture kernels~~
- Switching kernel
- NARMAX
- Neural ODEs

Week's goal

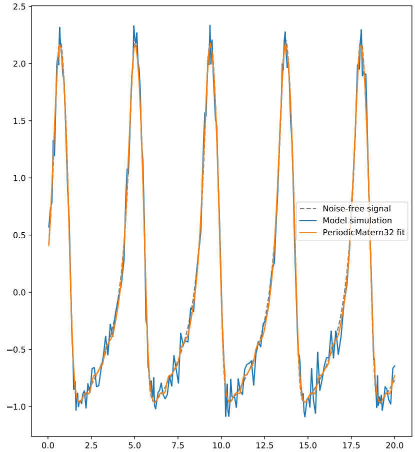
- Function-space distribution over kernels
  - Works, but no better than the other stationary kernels *[why not?]*
- Matern kernels
  - Works, but doesn't average noise out *[lengthscale issue]*
- Bayesian free-knot splines
  - Works well!
- ~~Generalised spectral mixture kernels~~
  - Couldn't get them to train

Matern kernel

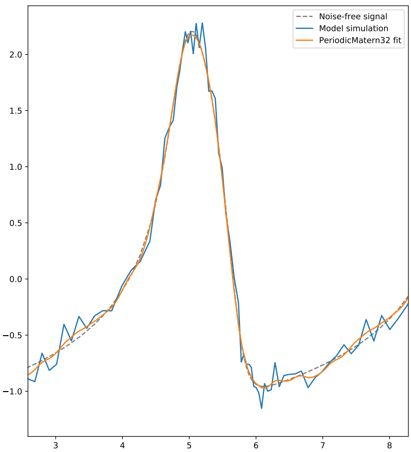
- SEKernel is C^∞ smooth
- Matern kernel generalises this to arbitrary degrees of smoothness
- Matern $\frac{3}{2}$ and $\frac{5}{2}$ are most commonly used
  - Once- and twice- differentiable posteriors
- Lack of smoothness adds more flexibility
- Quick and easy to test!

Can sometimes smooth data

Hindmarsh-Rose fast

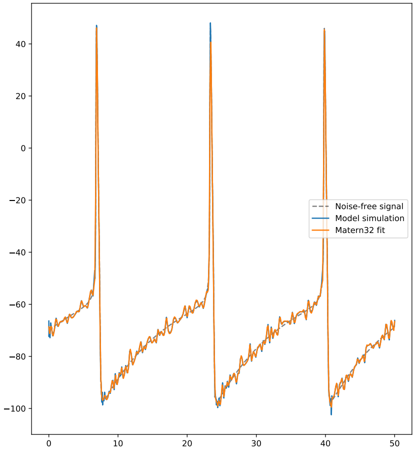


Hindmarsh-Rose fast

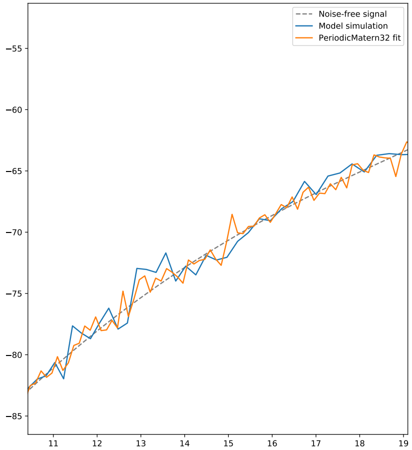


. . . but not always

Hodgkin-Huxley

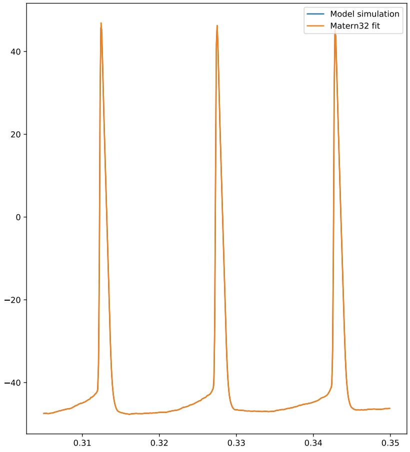


Hodgkin-Huxley

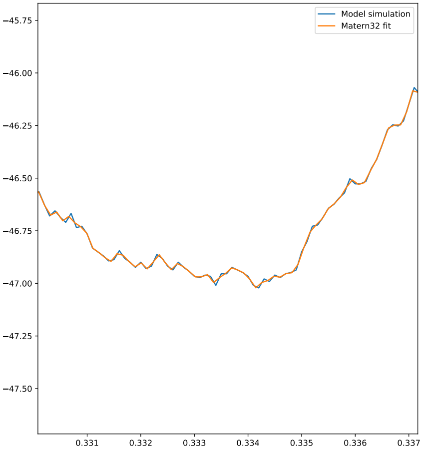


. . . but not always

Real data



Real data



MSPEs

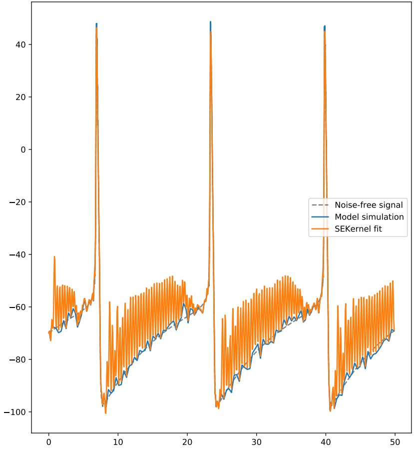
Exactly how good are any fits?

- Mean-square prediction error can quantify the goodness-of-fit with synthetic data
  - Split synthetic data into test and training
  - Fit model on training data
  - Find prediction error from test data

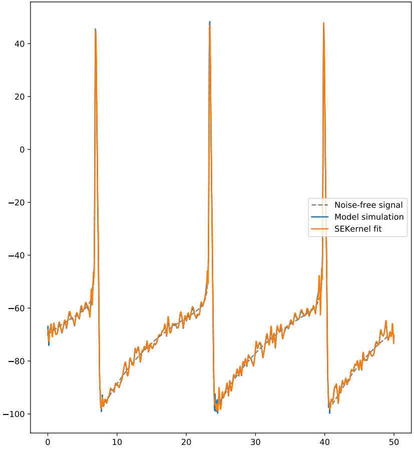
The downsampling step causes problems with a GPR fit

Model fits

Downsampled 666 to 333 datapoints



Simulated with 348 datapoints



MSPEs

Validation results can't always be trusted - MSPE values are often too high.

Possible hand-wavy explanation:

- More datapoints were generated by tightning the ODE solver tolerance

MSPEs

Validation results can't always be trusted - MSPE values are often too high.

Possible hand-wavy explanation:

- More datapoints were generated by tightning the ODE solver tolerance
- ODE solvers use an adaptive stepsize

MSPEs

Validation results can't always be trusted - MSPE values are often too high.

Possible hand-wavy explanation:

- More datapoints were generated by tightning the ODE solver tolerance
- ODE solvers use an adaptive stepsize
  - More datapoints where the system is locally stiff

MSPEs

Validation results can't always be trusted - MSPE values are often too high.

Possible hand-wavy explanation:

- More datapoints were generated by tightning the ODE solver tolerance
- ODE solvers use an adaptive stepsize
  - More datapoints where the system is locally stiff
  - Datapoints are therefore chosen to be as informative as possible

MSPEs

Validation results can't always be trusted - MSPE values are often too high.

Possible hand-wavy explanation:

- More datapoints were generated by tightning the ODE solver tolerance
- ODE solvers use an adaptive stepsize
  - More datapoints where the system is locally stiff
  - Datapoints are therefore chosen to be as informative as possible
  - Placed at points with the largest margin for error, to zero this error

MSPEs

Validation results can't always be trusted - MSPE values are often too high.

Possible hand-wavy explanation:

- More datapoints were generated by tightning the ODE solver tolerance
- ODE solvers use an adaptive stepsize
  - More datapoints where the system is locally stiff
  - Datapoints are therefore chosen to be as informative as possible
  - Placed at points with the largest margin for error, to zero this error
- Changing *rtol* always gives a maximally informative dataset, for the number of points

MSPEs

Validation results can't always be trusted - MSPE values are often too high.

Possible hand-wavy explanation:

- More datapoints were generated by tightning the ODE solver tolerance
- ODE solvers use an adaptive stepsize
  - More datapoints where the system is locally stiff
  - Datapoints are therefore chosen to be as informative as possible
  - Placed at points with the largest margin for error, to zero this error
- Changing *rtol* always gives a maximally informative dataset, for the number of points
- Downsampling doesn't always give maximally informative data

MSPEs

Validation results can't always be trusted - MSPE values are often too high.

Possible hand-wavy explanation:

- More datapoints were generated by tightning the ODE solver tolerance
- ODE solvers use an adaptive stepsize
  - More datapoints where the system is locally stiff
  - Datapoints are therefore chosen to be as informative as possible
  - Placed at points with the largest margin for error, to zero this error
- Changing *rtol* always gives a maximally informative dataset, for the number of points
- Downsampling doesn't always give maximally informative data
  - Removes datapoints based on their indices, rather than informativeness

MSPEs

Validation results can't always be trusted - MSPE values are often too high.

Possible hand-wavy explanation:

- More datapoints were generated by tightning the ODE solver tolerance
- ODE solvers use an adaptive stepsize
  - More datapoints where the system is locally stiff
  - Datapoints are therefore chosen to be as informative as possible
  - Placed at points with the largest margin for error, to zero this error
- Changing *rtol* always gives a maximally informative dataset, for the number of points
- Downsampling doesn't always give maximally informative data
  - Removes datapoints based on their indices, rather than informativeness
- Less informative dataset means worse GPR fit

Fixing MSPE

Alternative approaches to MSPE:

- Leave-one-out cross validation
  - Computationally expensive
- Visual inspection
  - Subjective, imprecise
- Run two solvers, one for test and one for training data
  - Need to make sure there's no shared datapoints for this to work
  - Bad test if test and training points are very close to each other

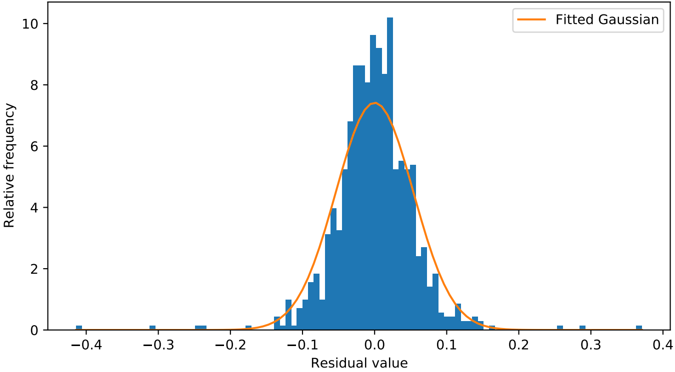
MSPE only seems to break on PeriodicKernels or Hodgkin Huxley dataset

- Chosen approach: use MSPE as-is, but do it carefully

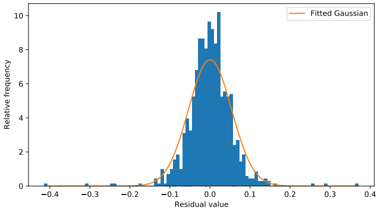
Real data

- Real data is the best test of a regression model
- Lack of ground-truth makes it harder to evaluate models on real data
- A heuristic method:
  - Fit model
  - Look at the model fit
  - Find residuals
  - Look at their distribution

Real data, splines model



Real data



- Nothing particularly alarming about the residuals
  - That's all we can really say
- ~~H_0: residuals are Gaussian~~ *[rejected]*
  - Shapiro-Wilk p-value: 3.0734860972720244e-21
  - D'Agostino's K^2 test: 4.3027710773715154e-37

Calculated MSPEs

Model	Matern32	Matern52	SEKernel	PeriodicKernel
Hodgkin Huxley	2.26(-2)	2.67 (-1)	5.57	9.25
Fitzhugh Nagumo	5.37 (-7)	2.34(-8)	2.97 (-4)	2.21 (-2)
HRFast	1.48 (-7)	4.30(-9)	2.37 (-6)	1.22 (-2)

- Calculated on noise-free models
- Matern kernels perform the best
- Can't compare MSPEs across neuron models, since it scales with the square of signal amplitude

Calculated MSPEs

Model	Matern32	Matern52	SEKernel	PeriodicKernel
Hodgkin Huxley	6.64	8.57	24.6	150
Fitzhugh Nagumo	3.95 (-3)	3.80e-3	4.85 (-3)	2.85 (-2)
HRFast	1.17 (-2)	1.16 (-2)	1.11e-2	2.18 (-2)

- Calculated on noise-perturbed models
- Matern kernels are generally good
- Representative values only; should really be ran lots of times to get an average

Splines

- 'Tie' together pieces of polynomials at knot-points

Splines

- 'Tie' together pieces of polynomials at knot-points
- Lower degree-of-freedom than GPR, so they forcibly remove noise *[see later slide]*

Splines

- ‘Tie’ together pieces of polynomials at knot-points
- Lower degree-of-freedom than GPR, so they forcibly remove noise *[see later slide]*
- No stationarity assumptions

Splines

- ‘Tie’ together pieces of polynomials at knot-points
- Lower degree-of-freedom than GPR, so they forcibly remove noise *[see later slide]*
- No stationarity assumptions
  - Can account for varying lengthscales by placing more knots at fast-changing points

Splines

- ‘Tie’ together pieces of polynomials at knot-points
- Lower degree-of-freedom than GPR, so they forcibly remove noise *[see later slide]*
- No stationarity assumptions
  - Can account for varying lengthscales by placing more knots at fast-changing points
- Successful splining needs good choices of knots

Splines

- ‘Tie’ together pieces of polynomials at knot-points
- Lower degree-of-freedom than GPR, so they forcibly remove noise *[see later slide]*
- No stationarity assumptions
  - Can account for varying lengthscales by placing more knots at fast-changing points
- Successful splining needs good choices of knots
  - Too many or too few knots will give bad results

Splines

- 'Tie' together pieces of polynomials at knot-points
- Lower degree-of-freedom than GPR, so they forcibly remove noise *[see later slide]*
- No stationarity assumptions
  - Can account for varying lengthscales by placing more knots at fast-changing points
- Successful splining needs good choices of knots
  - Too many or too few knots will give bad results
  - Poorly placed knots will mean splines can't capture signal

Splines

- ‘Tie’ together pieces of polynomials at knot-points
- Lower degree-of-freedom than GPR, so they forcibly remove noise *[see later slide]*
- No stationarity assumptions
  - Can account for varying lengthscales by placing more knots at fast-changing points
- Successful splining needs good choices of knots
  - Too many or too few knots will give bad results
  - Poorly placed knots will mean splines can’t capture signal
- Can choose degree of smoothness, for smoothing splines

Splines

- ‘Tie’ together pieces of polynomials at knot-points
- Lower degree-of-freedom than GPR, so they forcibly remove noise *[see later slide]*
- No stationarity assumptions
  - Can account for varying lengthscales by placing more knots at fast-changing points
- Successful splining needs good choices of knots
  - Too many or too few knots will give bad results
  - Poorly placed knots will mean splines can’t capture signal
- Can choose degree of smoothness, for smoothing splines
  - Downside: no good way to choose this!

Free-knot splines

A clever approach: free-knot splines

- Automatically choose both location and number of knots

Free-knot splines

A clever approach: free-knot splines

- Automatically choose both location and number of knots
- A GPR paper said free-knot splines work well

Free-knot splines

A clever approach: free-knot splines

- Automatically choose both location and number of knots
- A GPR paper said free-knot splines work well
- Current splines method: Bayesian adaptive regression splines

Free-knot splines

A clever approach: free-knot splines

- Automatically choose both location and number of knots
- A GPR paper said free-knot splines work well
- Current splines method: Bayesian adaptive regression splines
  - Also called BARS, Bayesian free-knot splines

Free-knot splines

A clever approach: free-knot splines

- Automatically choose both location and number of knots
- A GPR paper said free-knot splines work well
- Current splines method: Bayesian adaptive regression splines
  - Also called BARS, Bayesian free-knot splines
- There's a few free-knot splines methods out there

Free-knot splines

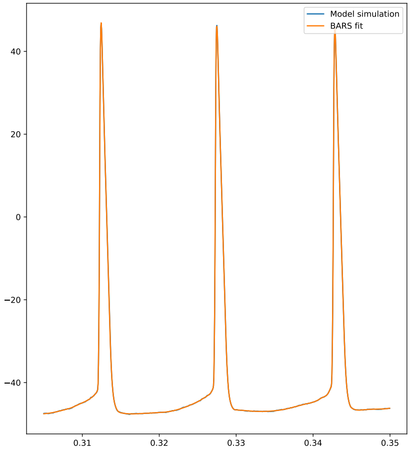
A clever approach: free-knot splines

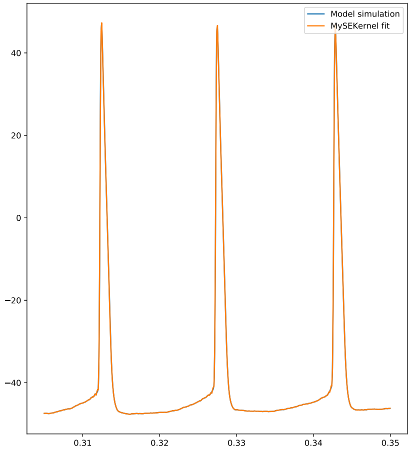
- Automatically choose both location and number of knots
- A GPR paper said free-knot splines work well
- Current splines method: Bayesian adaptive regression splines
  - Also called BARS, Bayesian free-knot splines
- There's a few free-knot splines methods out there
- I don't know how they work…

Splines vs GPR

Splines

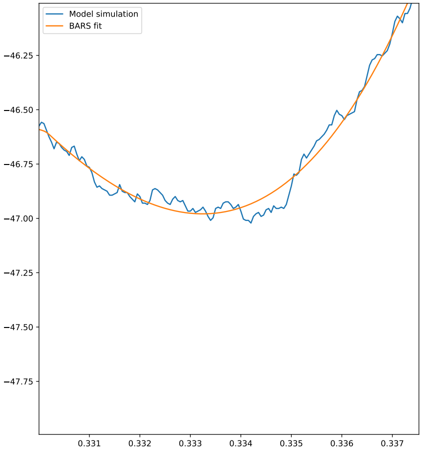
GPR (SEKernel)



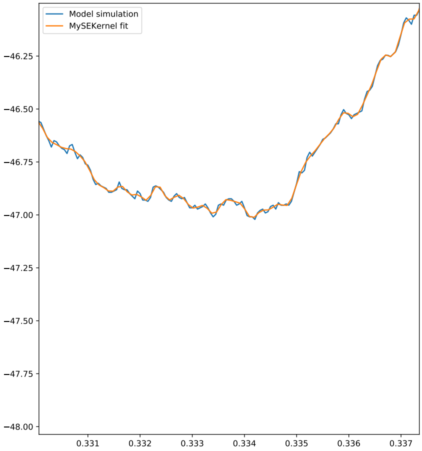


Splines vs GPR

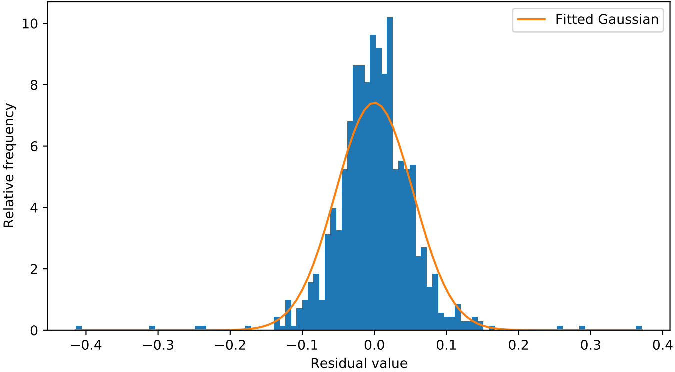
Splines



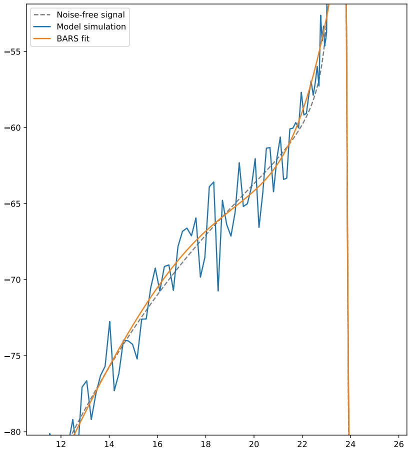
GPR (SEKernel)

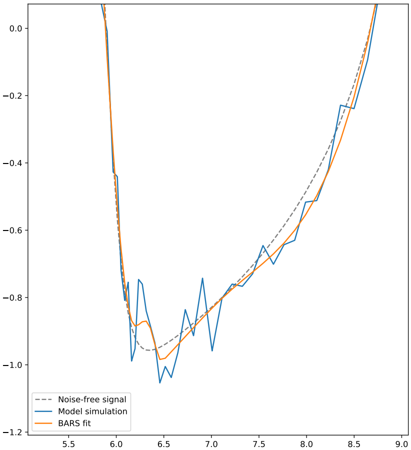


Real data, splines model



Not perfect, but good enough





Splines caveats

- BARS works well, without any hyperparameter tuning

Splines caveats

- BARS works well, without any hyperparameter tuning
- ISSUE: I don’t know how or why it works

Splines caveats

- BARS works well, without any hyperparameter tuning
- ISSUE: I don't know how or why it works
  - Can't rigorously justify why it's a good method

Splines caveats

- BARS works well, without any hyperparameter tuning
- ISSUE: I don’t know how or why it works
  - Can’t rigorously justify why it’s a good method
  - Can’t predict when it would work and when it would fail

Splines caveats

- BARS works well, without any hyperparameter tuning
- ISSUE: I don't know how or why it works
  - Can't rigorously justify why it's a good method
  - Can't predict when it would work and when it would fail
  - Can't determine good hyperparameter values

Splines caveats

- BARS works well, without any hyperparameter tuning
- ISSUE: I don't know how or why it works
  - Can't rigorously justify why it's a good method
  - Can't predict when it would work and when it would fail
  - Can't determine good hyperparameter values
- ISSUE: haven't implemented it

Splines caveats

- BARS works well, without any hyperparameter tuning
- ISSUE: I don't know how or why it works
  - Can't rigorously justify why it's a good method
  - Can't predict when it would work and when it would fail
  - Can't determine good hyperparameter values
- ISSUE: haven't implemented it
  - Relying on some old *C* code to make it run

Splines caveats

- BARS works well, without any hyperparameter tuning
- ISSUE: I don't know how or why it works
  - Can't rigorously justify why it's a good method
  - Can't predict when it would work and when it would fail
  - Can't determine good hyperparameter values
- ISSUE: haven't implemented it
  - Relying on some old *C* code to make it run
  - *C* implementation evaluates the splines model at the training points, and returns them

Splines caveats

- BARS works well, without any hyperparameter tuning
- ISSUE: I don't know how or why it works
  - Can't rigorously justify why it's a good method
  - Can't predict when it would work and when it would fail
  - Can't determine good hyperparameter values
- ISSUE: haven't implemented it
  - Relying on some old *C* code to make it run
  - *C* implementation evaluates the splines model at the training points, and returns them
  - Can't evaluate model at non-training points; doesn't give a continuous (interpolating) model, *can't be validated!*

Splines caveats

- BARS works well, without any hyperparameter tuning
- ISSUE: I don't know how or why it works
  - Can't rigorously justify why it's a good method
  - Can't predict when it would work and when it would fail
  - Can't determine good hyperparameter values
- ISSUE: haven't implemented it
  - Relying on some old *C* code to make it run
  - *C* implementation evaluates the splines model at the training points, and returns them
  - Can't evaluate model at non-training points; doesn't give a continuous (interpolating) model, *can't be validated!*
- ISSUE: harder to encode periodicity

Splines caveats

- BARS works well, without any hyperparameter tuning
- ISSUE: I don't know how or why it works
  - Can't rigorously justify why it's a good method
  - Can't predict when it would work and when it would fail
  - Can't determine good hyperparameter values
- ISSUE: haven't implemented it
  - Relying on some old *C* code to make it run
  - *C* implementation evaluates the splines model at the training points, and returns them
  - Can't evaluate model at non-training points; doesn't give a continuous (interpolating) model, *can't be validated!*
- ISSUE: harder to encode periodicity
  - Periodic kernels almost surely (probability 1) give periodic posteriors

Splines caveats

- BARS works well, without any hyperparameter tuning
- ISSUE: I don't know how or why it works
  - Can't rigorously justify why it's a good method
  - Can't predict when it would work and when it would fail
  - Can't determine good hyperparameter values
- ISSUE: haven't implemented it
  - Relying on some old *C* code to make it run
  - *C* implementation evaluates the splines model at the training points, and returns them
  - Can't evaluate model at non-training points; doesn't give a continuous (interpolating) model, *can't be validated!*
- ISSUE: harder to encode periodicity
  - Periodic kernels almost surely (probability 1) give periodic posteriors
  - Periodic splines are a thing, maybe try periodic BARS?

Abandoned ideas

- Generalised spectral mixture kernels
  - Couldn't get them to train
- Support vector regression
  - Couldn't find any justification to use this over GPR
- Latent ODEs / neural ODEs / physics-informed NNs
  - Would require state-space reconstruction, doesn't seem like a beneficial use of time

Other models for a paper

- NARMAX
- Wavelets
- Warping GPs
  - Either learn a warp. . .
  - . . . or apply a simple transformation to the data (log, exp, logistic, . . .)
- Deep GPs
- Hybrid methods
- Other nonparametric methods
  - RKHS, KNN, etc.

Next steps

- Dig into free-knot splines literature

Then. . .

Next steps

- Dig into free-knot splines literature
  - BARS and other free-knot methods

Then…

Next steps

- Dig into free-knot splines literature
  - BARS and other free-knot methods
- Understand how and why it works

Then…

Next steps

- Dig into free-knot splines literature
  - BARS and other free-knot methods
- Understand how and why it works
  - Useful for justifying why it's a good model, and when it will and won't work

Then. . .

Next steps

- Dig into free-knot splines literature
  - BARS and other free-knot methods
- Understand how and why it works
  - Useful for justifying why it's a good model, and when it will and won't work
  - Will help decide whether periodic BARS is possible

Then...

Next steps

- Dig into free-knot splines literature
  - BARS and other free-knot methods
- Understand how and why it works
  - Useful for justifying why it's a good model, and when it will and won't work
  - Will help decide whether periodic BARS is possible
- Write up my own implementation

Then...

Next steps

- Dig into free-knot splines literature
  - BARS and other free-knot methods
- Understand how and why it works
  - Useful for justifying why it's a good model, and when it will and won't work
  - Will help decide whether periodic BARS is possible
- Write up my own implementation
  - Allows for validation, interpolation

Then…

Next steps

- Dig into free-knot splines literature
  - BARS and other free-knot methods
- Understand how and why it works
  - Useful for justifying why it's a good model, and when it will and won't work
  - Will help decide whether periodic BARS is possible
- Write up my own implementation
  - Allows for validation, interpolation

Then...

- Other data sources

Next steps

- Dig into free-knot splines literature
  - BARS and other free-knot methods
- Understand how and why it works
  - Useful for justifying why it's a good model, and when it will and won't work
  - Will help decide whether periodic BARS is possible
- Write up my own implementation
  - Allows for validation, interpolation

Then...

- Other data sources
- Warping GPs, deep GPs, NARMAX, etc.

Next steps

- Dig into free-knot splines literature
  - BARS and other free-knot methods
- Understand how and why it works
  - Useful for justifying why it's a good model, and when it will and won't work
  - Will help decide whether periodic BARS is possible
- Write up my own implementation
  - Allows for validation, interpolation

Then...

- Other data sources
- Warping GPs, deep GPs, NARMAX, etc.
- MATLAB wrapper?

Next steps

- Dig into free-knot splines literature
  - BARS and other free-knot methods
- Understand how and why it works
  - Useful for justifying why it's a good model, and when it will and won't work
  - Will help decide whether periodic BARS is possible
- Write up my own implementation
  - Allows for validation, interpolation

Then...

- Other data sources
- Warping GPs, deep GPs, NARMAX, etc.
- MATLAB wrapper?
  - Having ready-to-go codes might make a paper more popular

```
./model_tester.py -d HRFast -m MySEKernel

./model_tester.py -d HindmarshRose -m Matern32 -p CleanFitted
  -r 1e-6 -V

./model_tester.py -d FitzhughNagumo -m ModuloKernel -r 1e-6
  -n 0.1

./model_tester.py -d HodgkinHuxley -m PeriodicKernel -o

./model_tester.py -d 08o28004_channel_0_sweep_8.np -m
  MySEKernel -f 6.23e2 -l 5.71e-8 -n 0.1
```