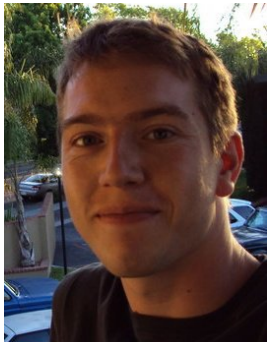
Automatic Model Construction



David Duvenaud, James Robert Lloyd, Roger Grosse,


Joshua Tenenbaum, Zoubin Ghahramani

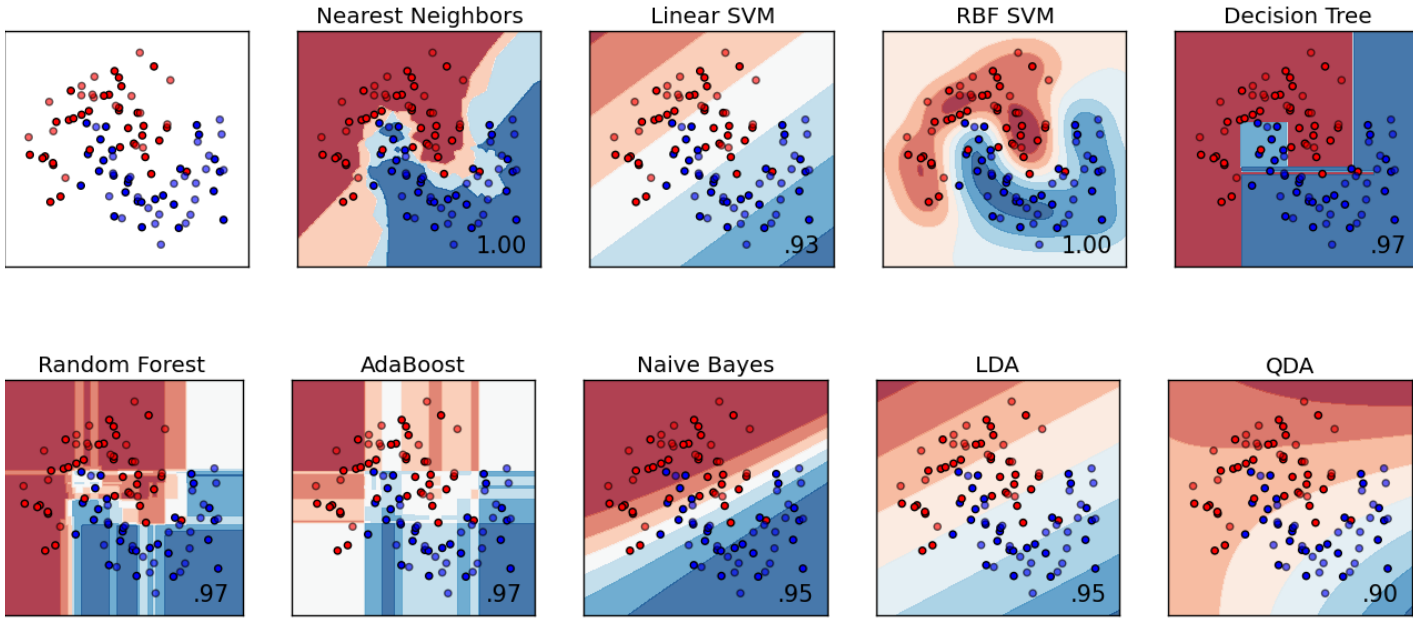
July 11, 2015

OUTLINE

- Structure discovery in regression
  - Gaussian processes
  - Structures expressible through covariance
  - Grammar & model search
  - Examples
- Structure discovery in matrix models
  - diverse models as matrix decompositions
  - Grammar & Special cases
  - Structure discovered in natural images
- Automating Statistics
  - model checking and marginal likelihood
  - espistemological questions

TYPICAL STATISTICAL MODELLING

- models typically built by hand, or chosen from a fixed set

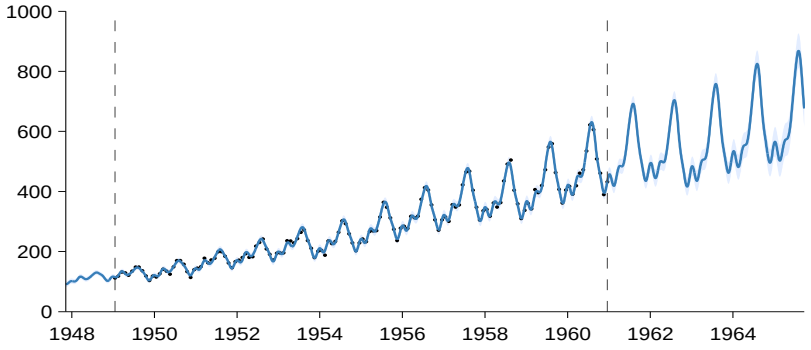


- Building by hand requires considerable expertise
- Just being nonparametric isn't good enough
  - Nonparametric does not mean assumption-free!
- Can silently fail
  - How to tell if none of the models fit the data well?

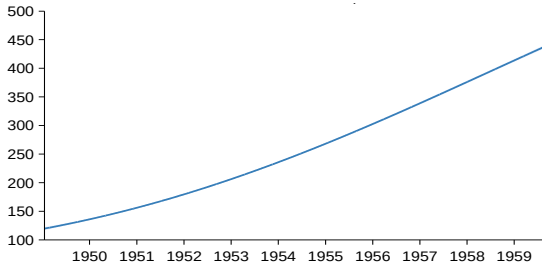
CAN WE DO BETTER?

- Andrew Gelman asks: How could an AI do statistics?
- An artificial statistician would need:
  - a language that could describe arbitrarily complicated models
  - a method of searching over those models
  - a procedure to check model fit
- We construct such a language over regression models, a procedure to search over it, and a method to describe in natural language the properties of the resulting models

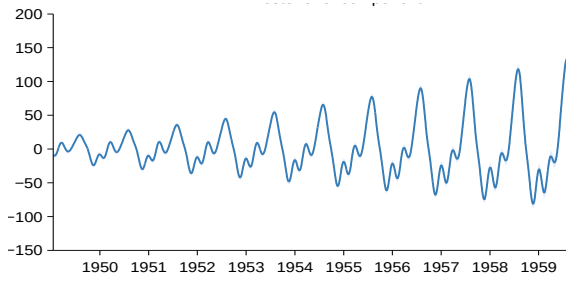
EXAMPLE: AN AUTOMATIC ANALYSIS



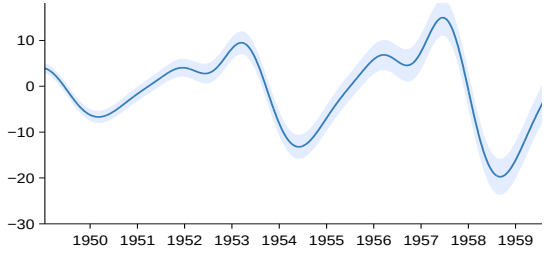
=



+



+



A very smooth, monotonically increasing function

An approximately periodic function with a period of 1.0 years and with approximately linearly increasing amplitude

An exactly periodic function with a period of 4.3 years but with linearly increasing amplitude

GAUSSIAN PROCESS REGRESSION

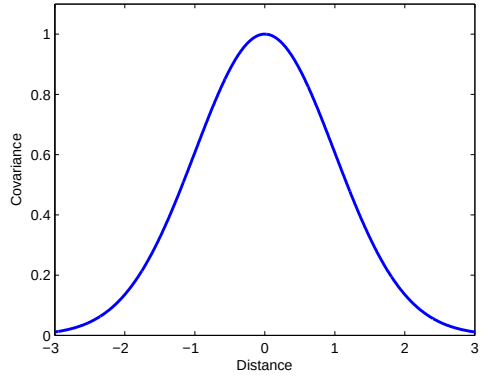
$f \sim \mathcal{GP}\left(\mu, k(\cdot,\cdot)\right)$ means that, for any points $x_1, x_2, \ldots, x_N$,

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(x_1) \\ \mu(x_2) \\ \vdots \\ \mu(x_N) \end{bmatrix}, \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \ldots & k(x_1,x_N) \\ k(x_2,x_1) & k(x_2,x_2) & \ldots & k(x_2,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N,x_1) & k(x_N,x_2) & \ldots & k(x_N,x_N) \end{bmatrix} \right)$$

covariance function or *kernel* gives $k(x,x') = \text{cov}\left[f(x), f(x')\right]$

Typically, if x_1, x_2 are close, then $f(x_1), f(x_2)$ are correlated:

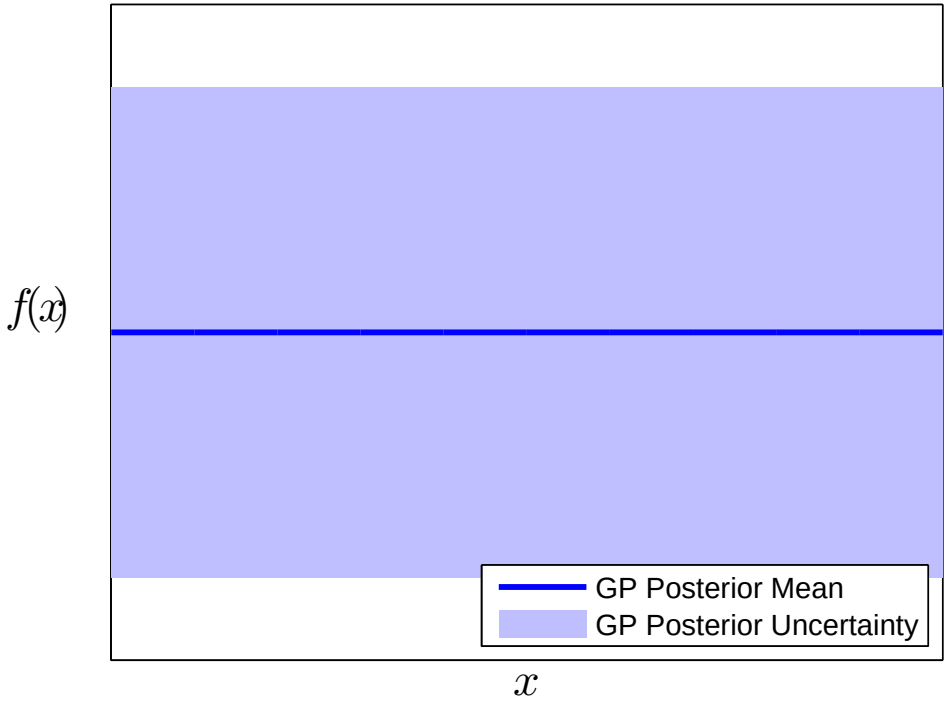
$$k_{\text{SE}}(x,x') = \exp\left(-\frac{1}{2\theta}|x-x'|_2^2\right)$$



CONDITIONAL POSTERIOR

$$\text{mean}(f(x^\star)) = k(x^*, \mathbf{X})K^{-1}\mathbf{y}$$
$$\text{variance}(f(x^\star)) = k(x^*, x^*) - k(x^*, \mathbf{X})K^{-1}k(\mathbf{X}, x^*)$$

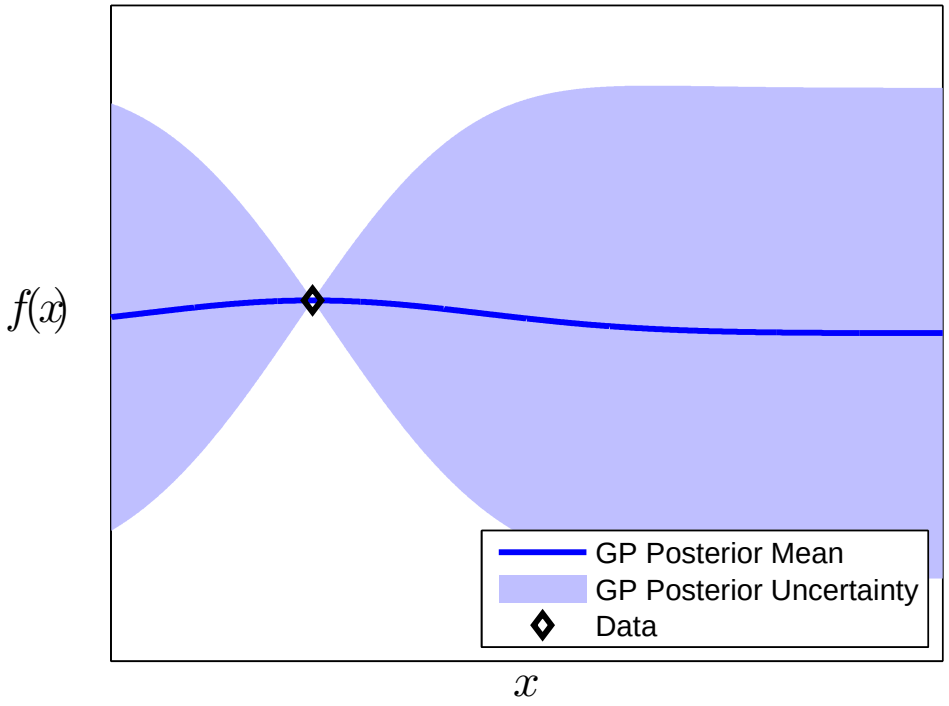
With SE kernel:



Conditional Posterior

$$\text{mean}(f(x^\star)) = k(x^*, \mathbf{X})K^{-1}\mathbf{y}$$
$$\text{variance}(f(x^\star)) = k(x^*, x^*) - k(x^*, \mathbf{X})K^{-1}k(\mathbf{X}, x^*)$$

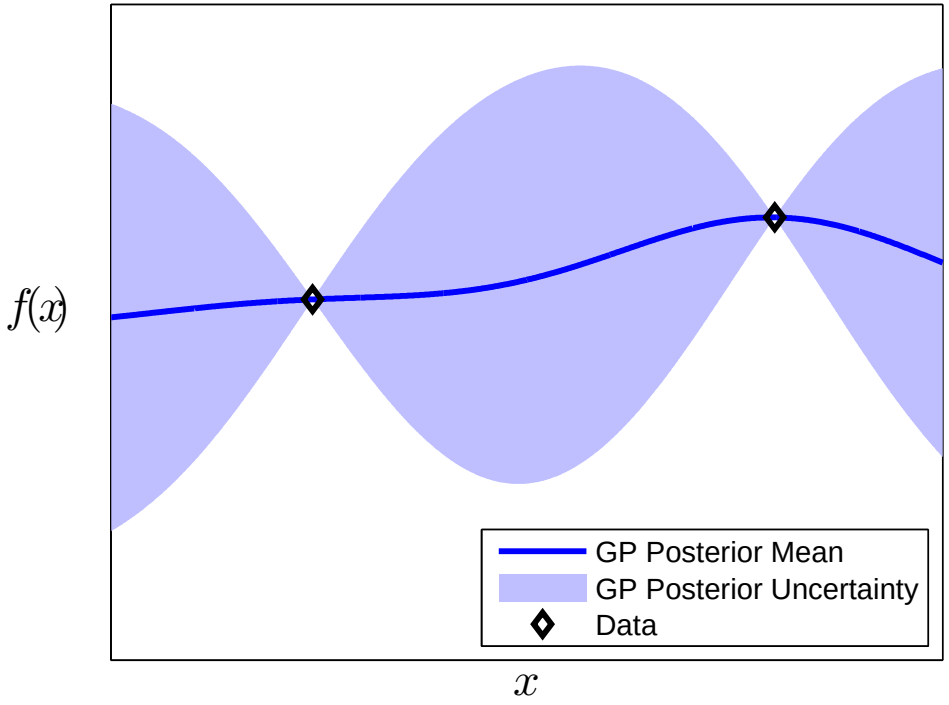
With SE kernel:



CONDITIONAL POSTERIOR

$$\text{mean}(f(x^\star)) = k(x^*, \mathbf{X})K^{-1}\mathbf{y}$$
$$\text{variance}(f(x^\star)) = k(x^*, x^*) - k(x^*, \mathbf{X})K^{-1}k(\mathbf{X}, x^*)$$

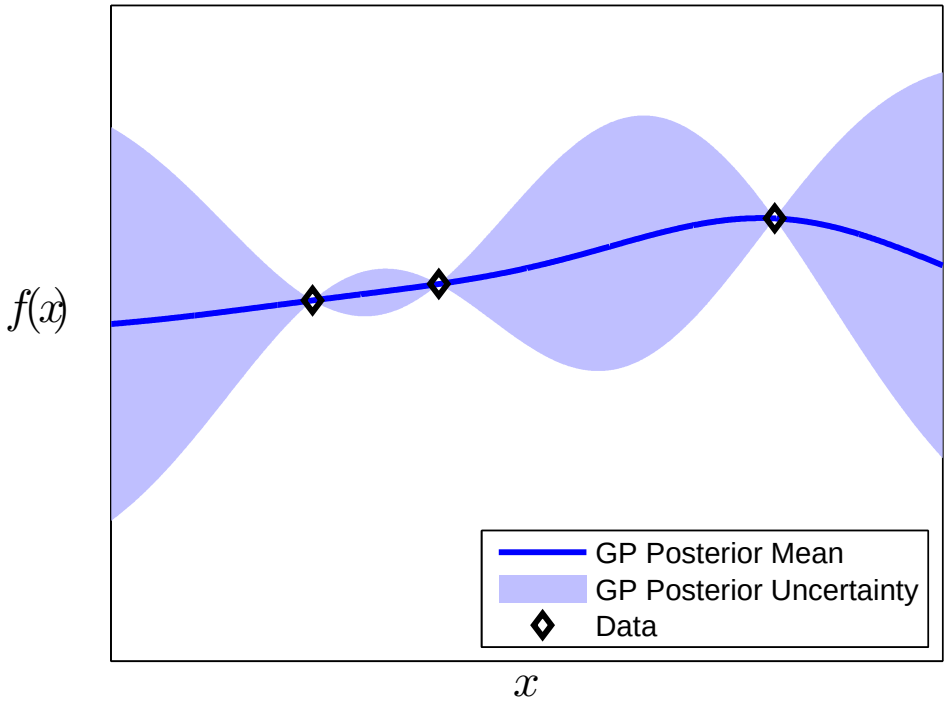
With SE kernel:



Conditional Posterior

$$\text{mean}(f(x^\star)) = k(x^*, \mathbf{X})K^{-1}\mathbf{y}$$
$$\text{variance}(f(x^\star)) = k(x^*, x^*) - k(x^*, \mathbf{X})K^{-1}k(\mathbf{X}, x^*)$$

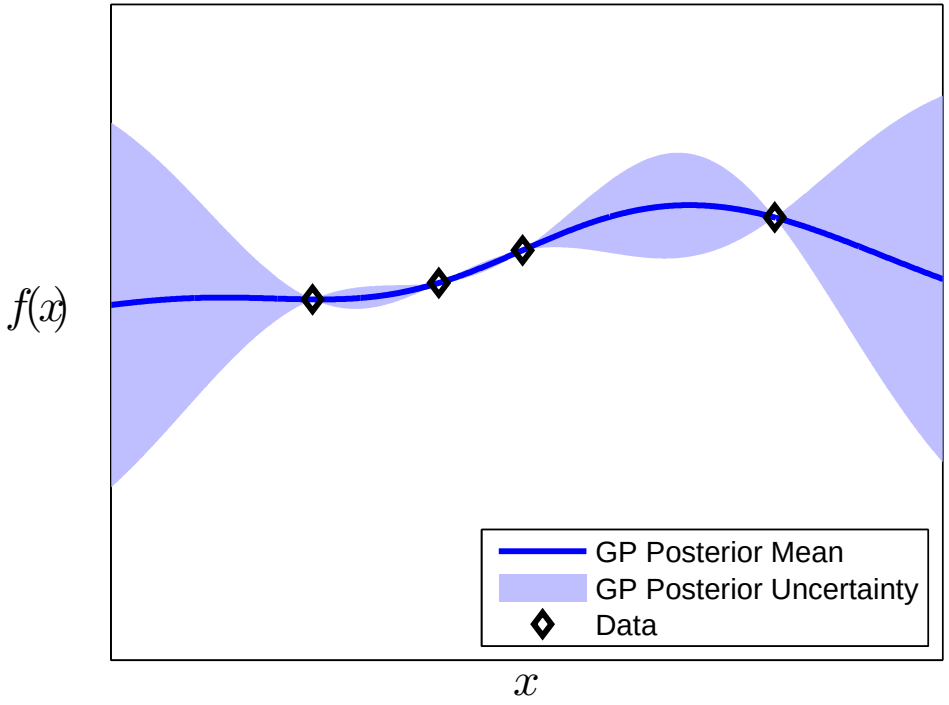
With SE kernel:



Conditional Posterior

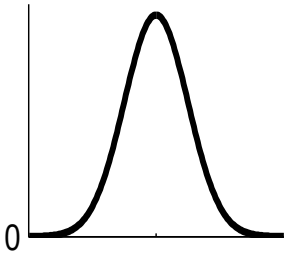
$$\text{mean}(f(x^\star)) = k(x^*, \mathbf{X})K^{-1}\mathbf{y}$$
$$\text{variance}(f(x^\star)) = k(x^*, x^*) - k(x^*, \mathbf{X})K^{-1}k(\mathbf{X}, x^*)$$

With SE kernel:

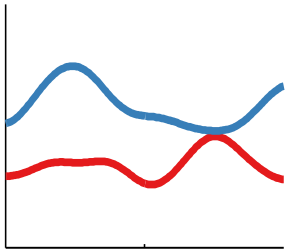


KERNELS DETERMINE STRUCTURE OF GPS

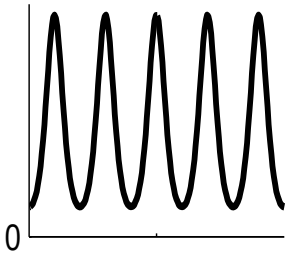
- Kernel determines almost all the properties of a GP prior
- Many different kinds, with very different properties:



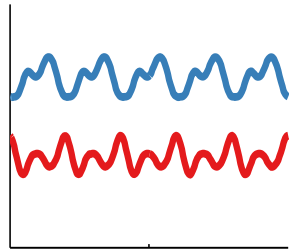
Squared-exp (SE)



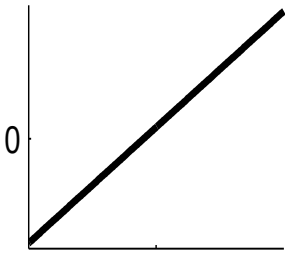
local variation



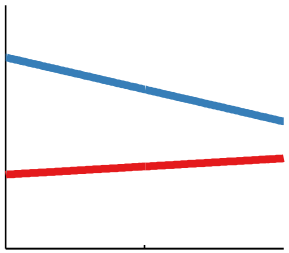
Periodic (PER)



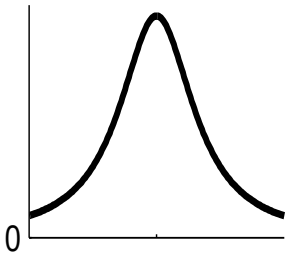
repeating structure



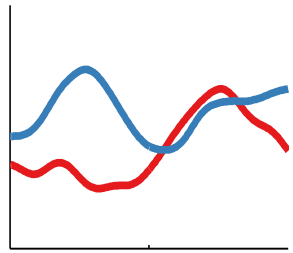
Linear (LIN)



linear functions



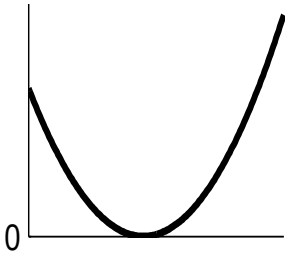
Rational-quadratic(RQ)



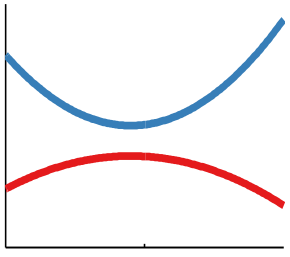
multi-scale variation

KERNELS CAN BE COMPOSED

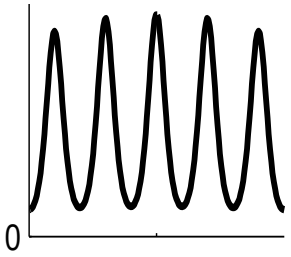
- Two main operations: addition, multiplication



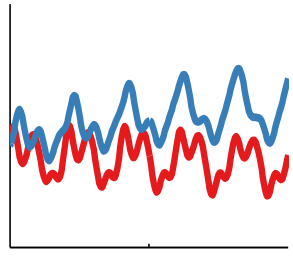
LIN × LIN



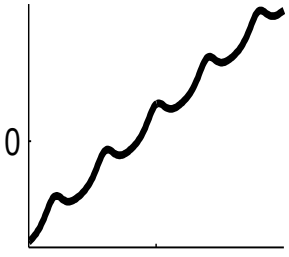
quadratic functions



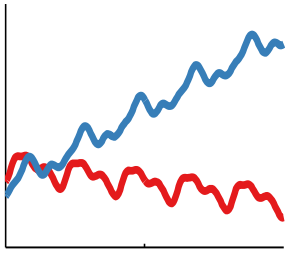
SE × PER



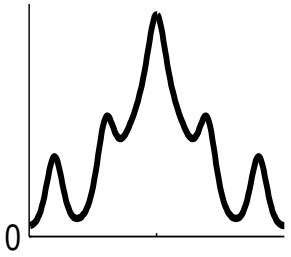
locally periodic



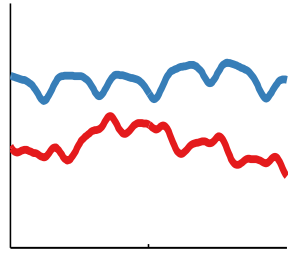
LIN + PER



periodic with trend



SE + PER



periodic with noise

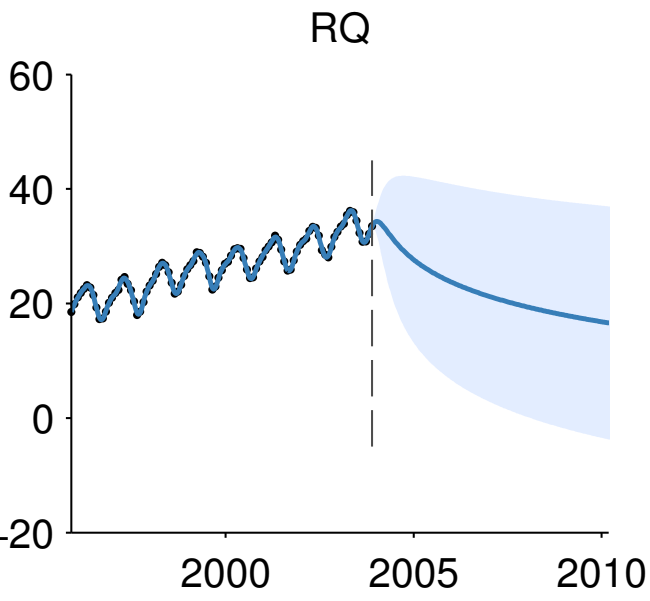
A LANGUAGE OF REGRESSION MODELS

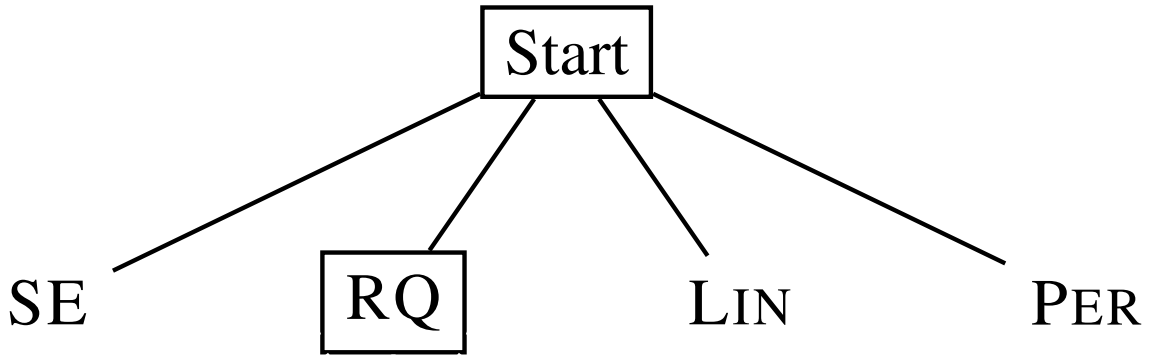
- We define a language of regression models by defining a language over kernel functions
- We start with a small set of base kernels and define a grammar
  - $K \rightarrow K + K$
  - $K \rightarrow K \times K$
  - $K \rightarrow \{\text{SE}, \text{LIN}, \text{PER}\}$
  - $K \rightarrow CP(K, K)$
- The language is open-ended, every model in it is tractable

SPECIAL CASES IN THIS LANGUAGE

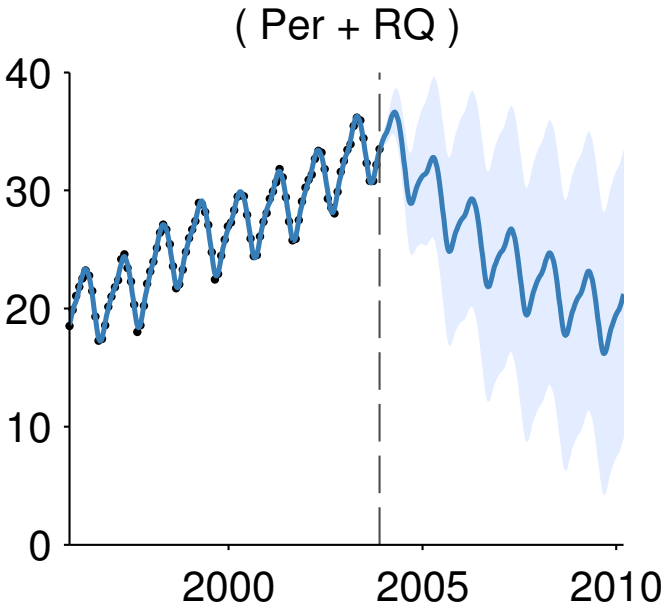
Regression motif	Example kernel
Bayesian Linear regression	LIN
Bayesian polynomial regression	$\text{LIN} \times \text{LIN} \times \ldots$
Generalized Fourier decomposition	$\text{PER} + \text{PER} + \ldots$
Changepoints	$\text{CP}(K_1, K_2)$
Generalized additive models	$\sum_{d=1}^{D} \text{SE}_d$
Automatic relevance determination	$\prod_{d=1}^{D} \text{SE}_d$

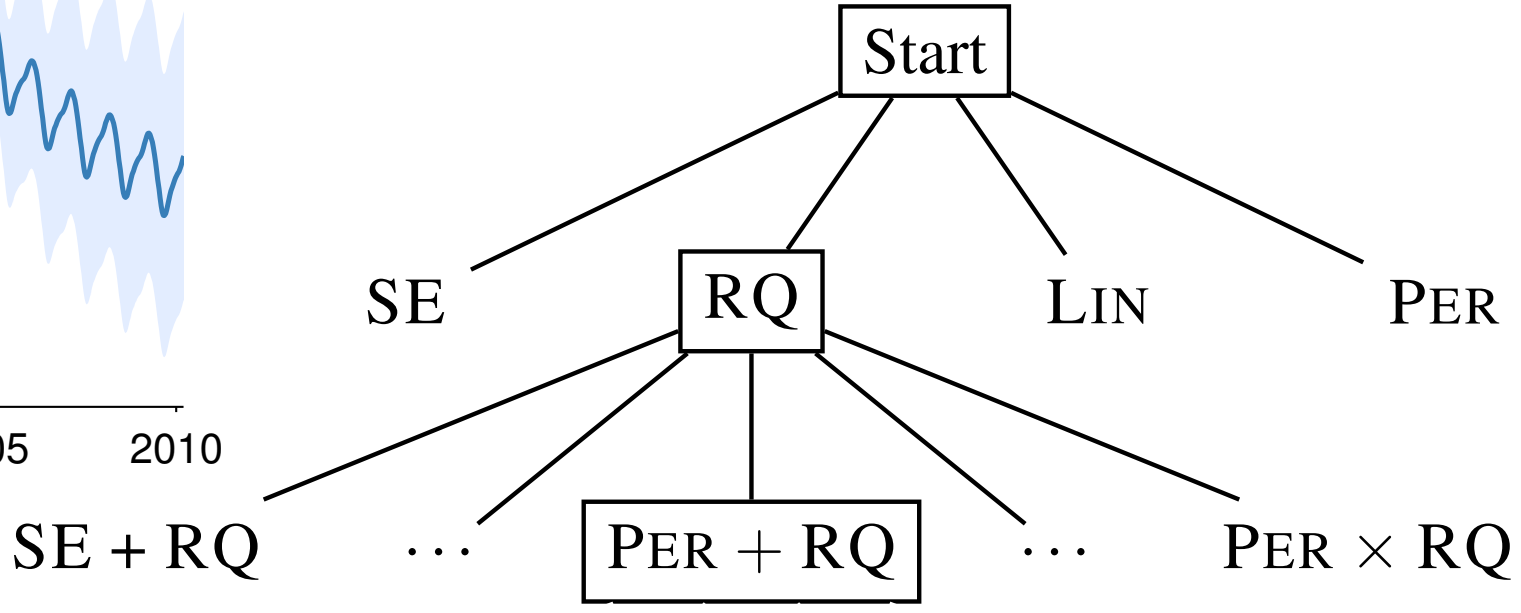
COMPOSITIONAL STRUCTURE SEARCH



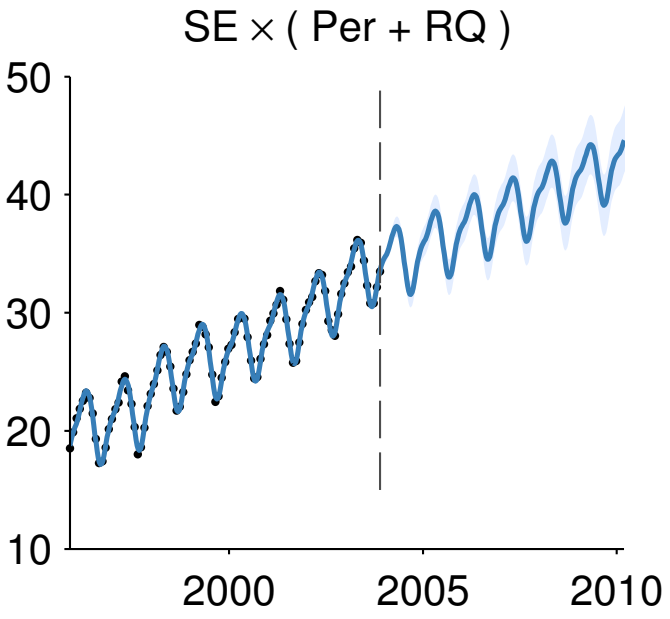


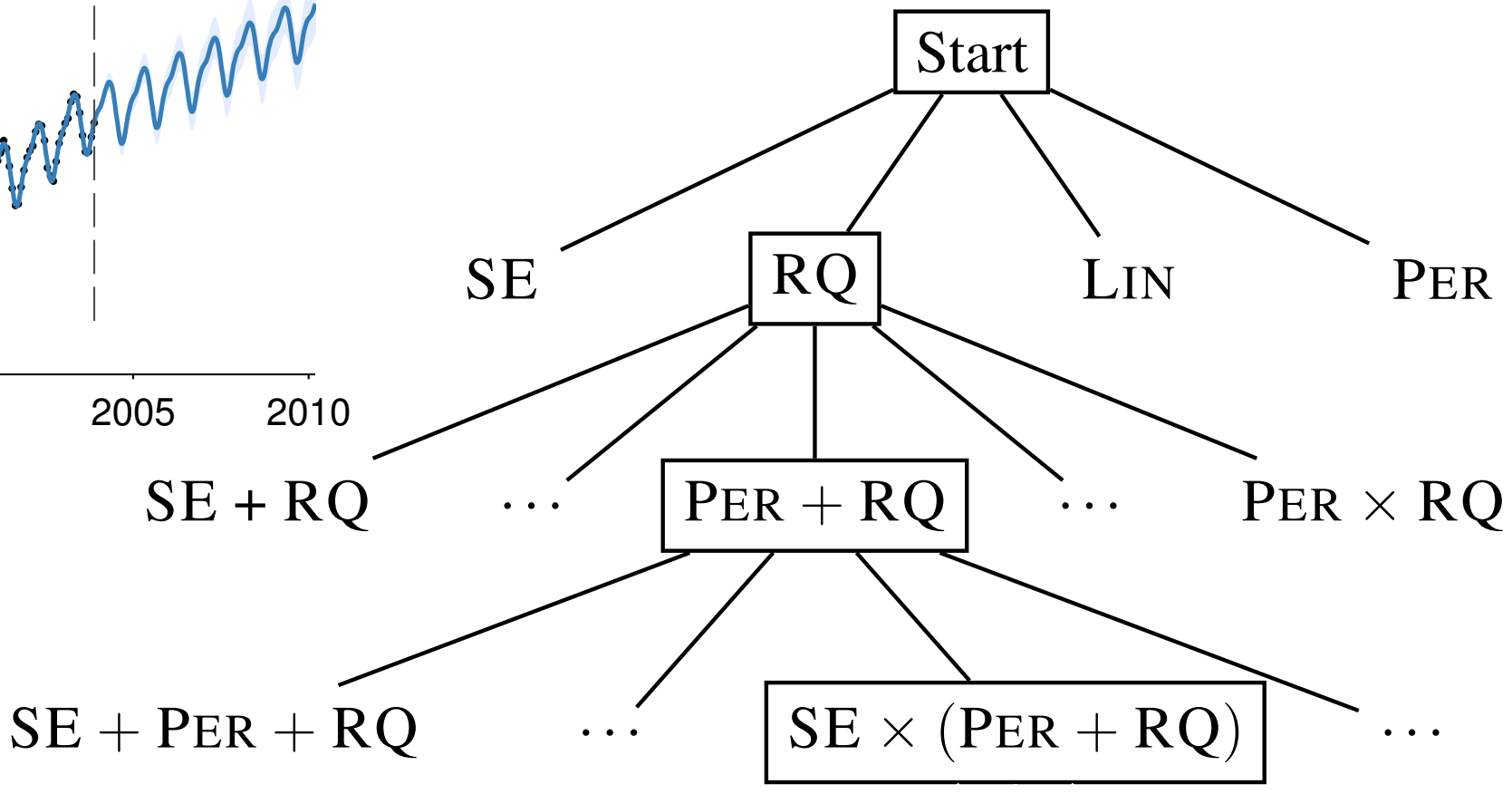
COMPOSITIONAL STRUCTURE SEARCH



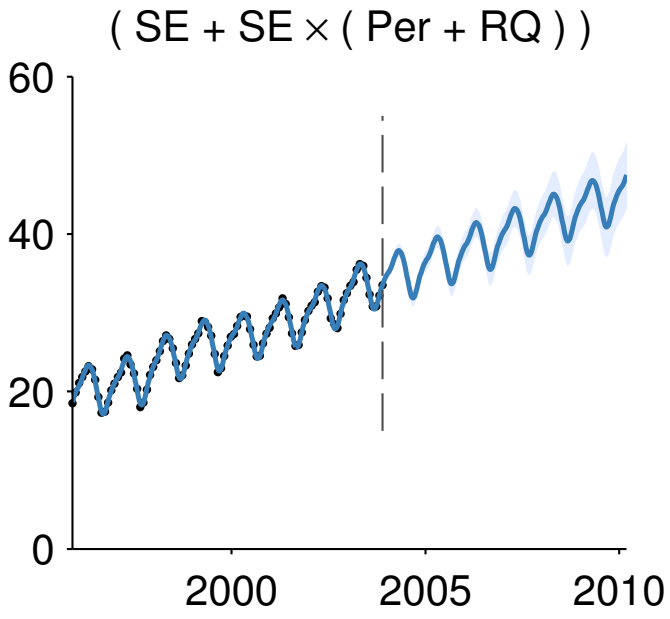


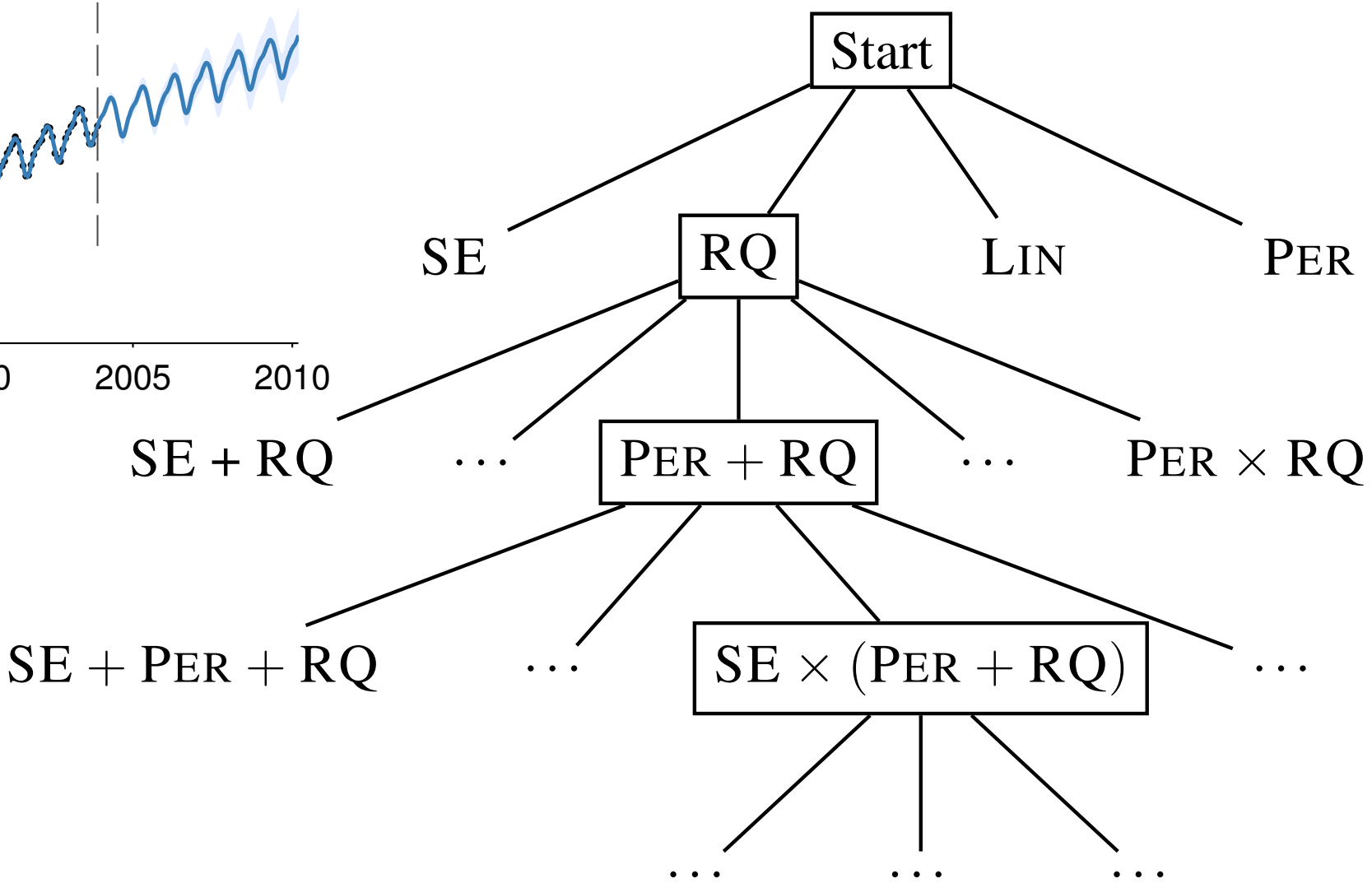
COMPOSITIONAL STRUCTURE SEARCH





COMPOSITIONAL STRUCTURE SEARCH





DISTRIBUTIVITY HELPS INTERPRETABILITY

We can write all kernels as sums of products of base kernels:

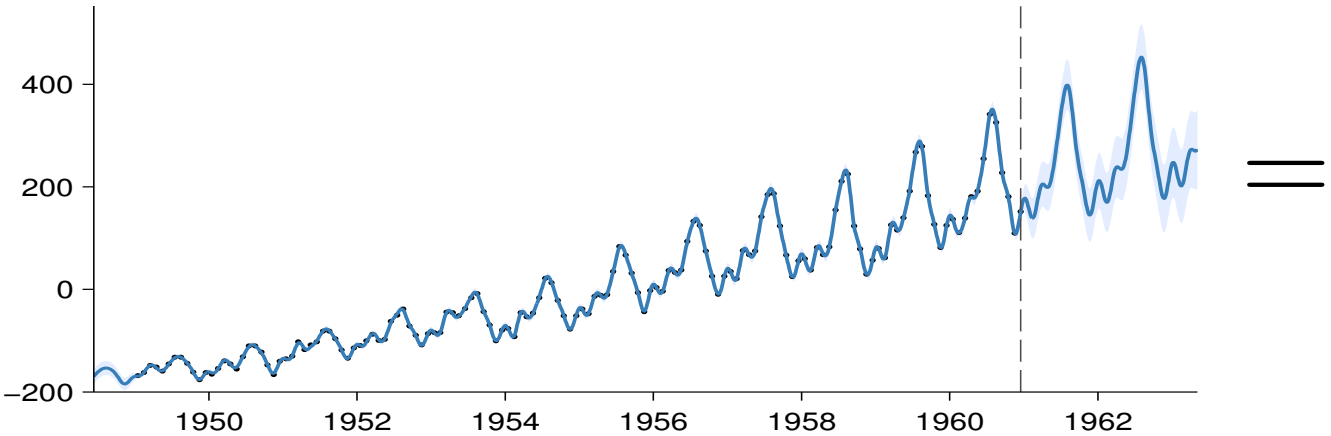
$$\text{SE} \times (\text{RQ} + \text{LIN}) = (\text{SE} \times \text{RQ}) + (\text{SE} \times \text{LIN}).$$

Sums of kernels are equivalent to sums of functions.
If f_1, f_2 are independent, and $f_1 \sim \mathcal{GP}(\mu_1, k_1), f_2 \sim \mathcal{GP}(\mu_2, k_2)$ then

$$(f_1 + f_2) \sim \mathcal{GP}(\mu_1 + \mu_2, k_1 + k_2)$$

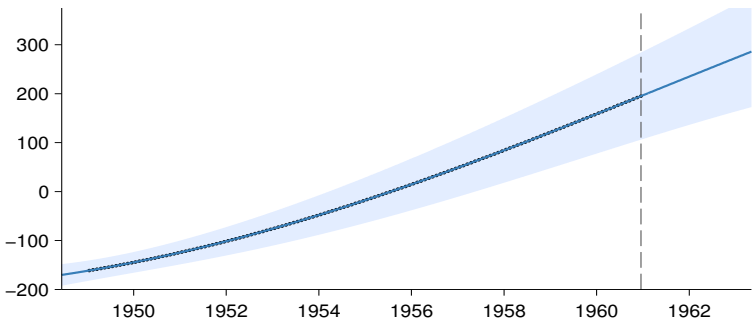
Example Decomposition: Airline

SE × (Lin + Lin × (Per + RQ))

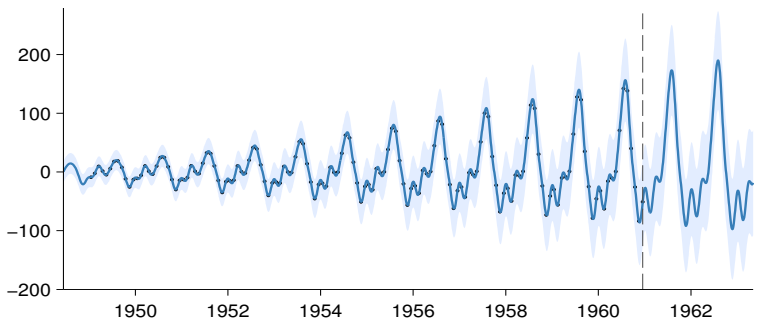


=

SE × Lin

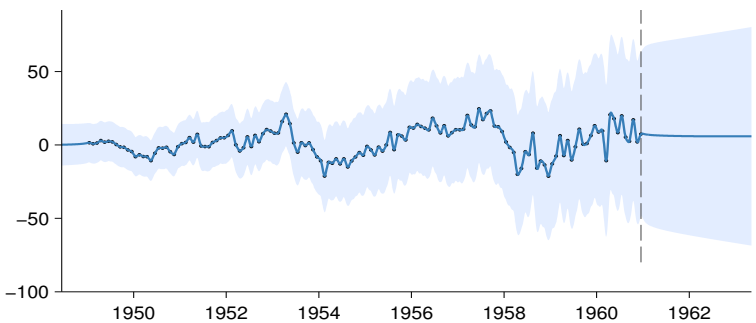


SE × Lin × Per

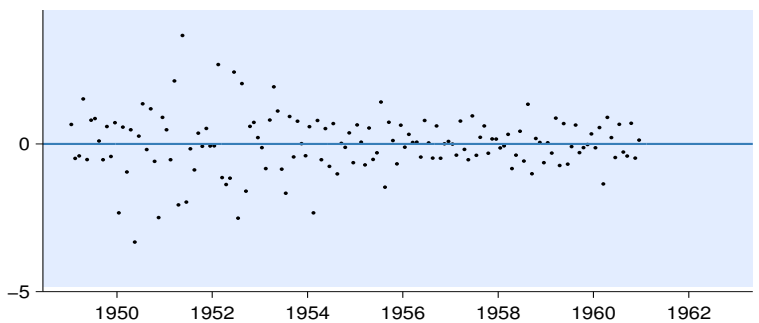


+

SE × Lin × RQ



Residuals

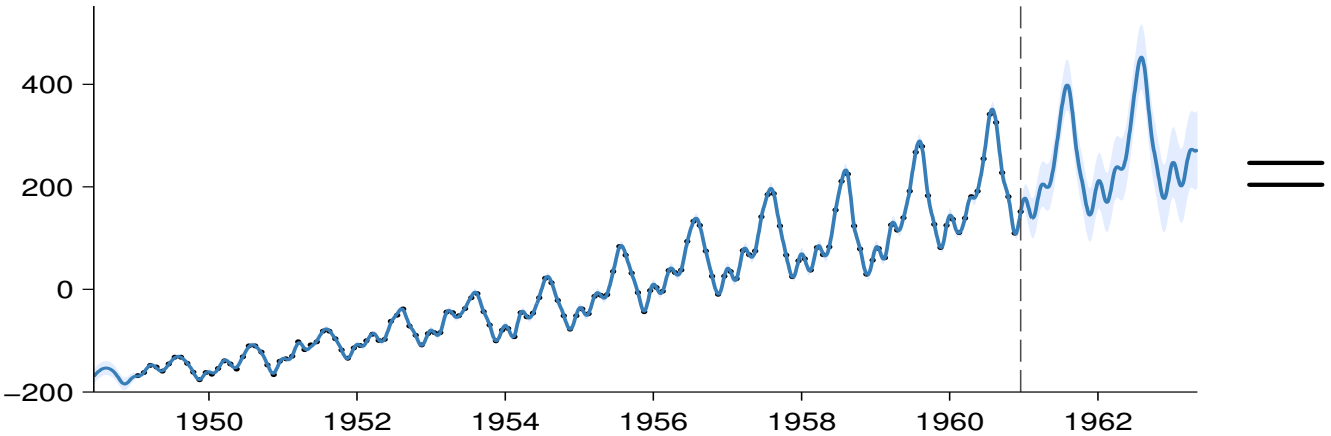


SUMMARY

- Choosing form of kernel is currently done by hand.
- Compositions of kernels lead to more interesting priors on functions than typically considered.
- A simple grammar specifies all such compositions, and can be searched over automatically.
- Composite kernels lead to interpretable decompositions . . . automatically?

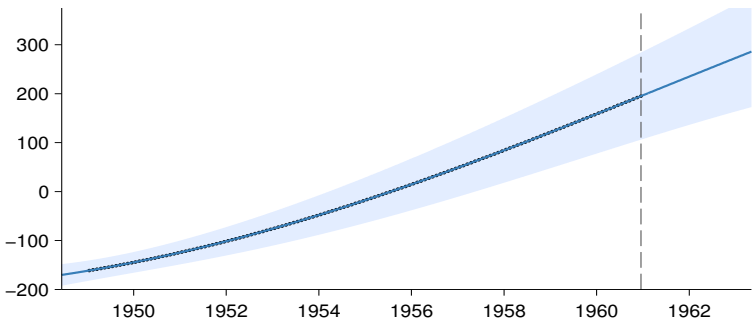
Example Decomposition: Airline

SE × (Lin + Lin × (Per + RQ))

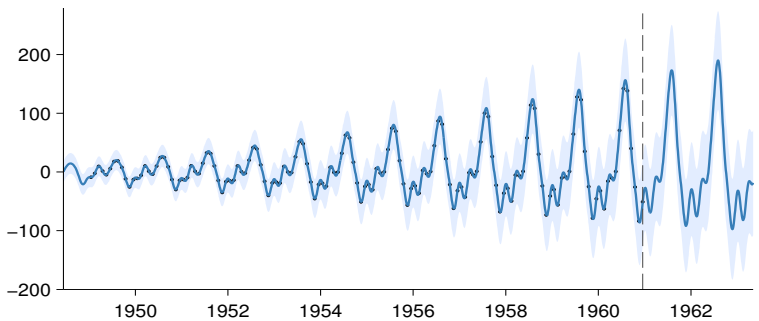


=

SE × Lin

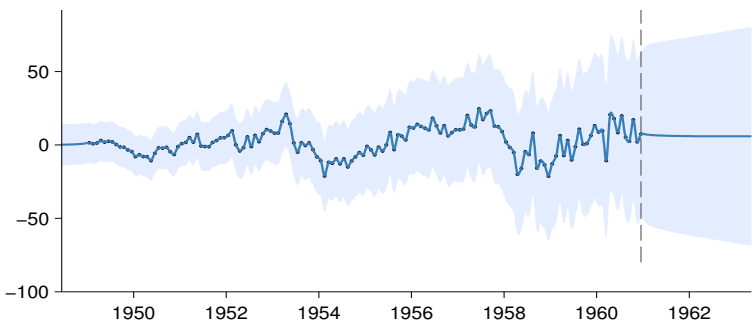


SE × Lin × Per

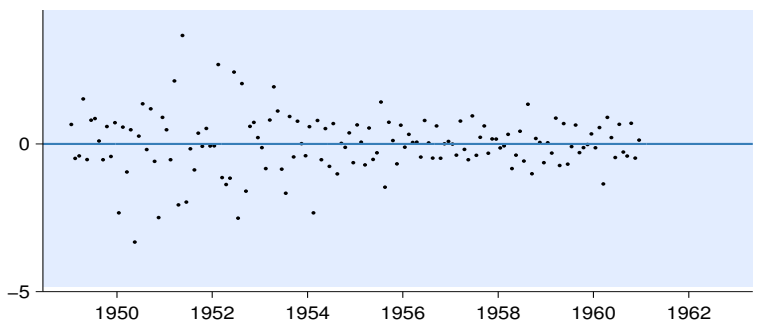


+

SE × Lin × RQ



Residuals



DESCRIBING KERNELS

Each kernel acts as a modifier in a standard way: an “adjective”.

Kernel	Becomes
$K \times \mathrm{SE}$	’locally’ or ’approximately’
$K \times \mathrm{LIN}$	’with linearly growing amplitude’
$K \times \mathrm{PER}$	’periodic’
$\mathrm{CP}(K1, K2)$	’. . . changing at x to . . . ’

- Special cases for when they’re on their own
- Extra adjectives depending on hyperparameters

EXAMPLE KERNEL DESCRIPTIONS

Product of Kernels	Description
PER	An exactly periodic function
PER $\times$ SE	An approximately periodic function
PER $\times$ SE $\times$ LIN	An approximately periodic function with linearly varying amplitude

THIS ANALYSIS WAS AUTOMATICALLY GENERATED

The raw data and full model posterior with extrapolations are shown in figure 1.

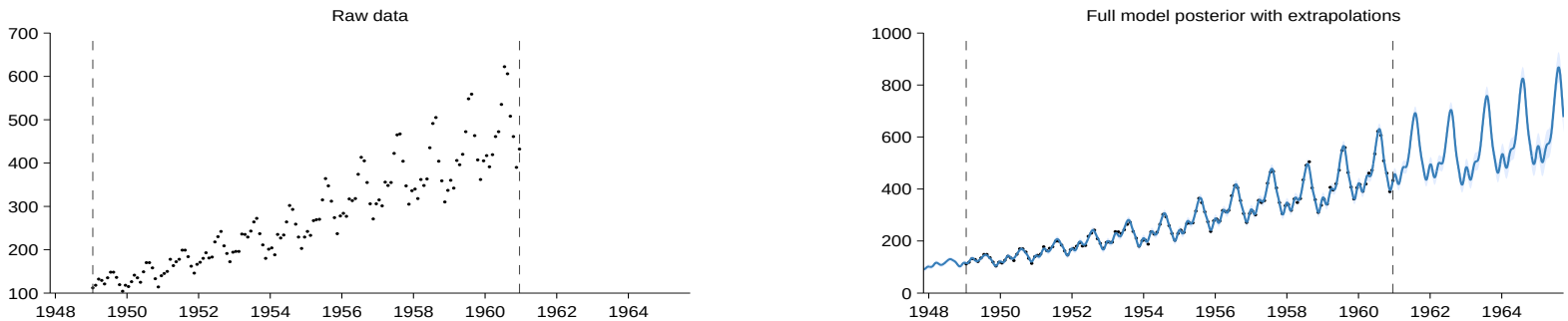


Figure 1: Raw data (left) and model posterior with extrapolation (right)

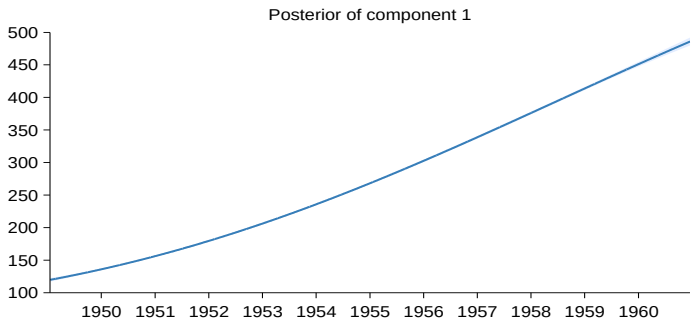
The structure search algorithm has identified four additive components in the data:

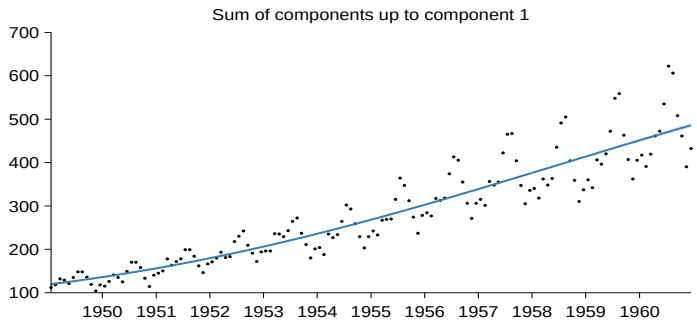
- A very smooth monotonically increasing function.
- An approximately periodic function with a period of 1.0 years and with approximately linearly increasing amplitude.
- An exactly periodic function with a period of 4.3 years but with linearly increasing amplitude.
- Uncorrelated noise with linearly increasing standard deviation.

THIS ANALYSIS WAS AUTOMATICALLY GENERATED

2.1 Component 1

This component is a very smooth and monotonically increasing function.

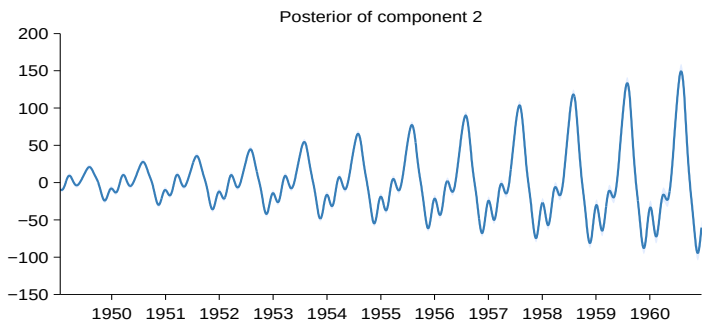


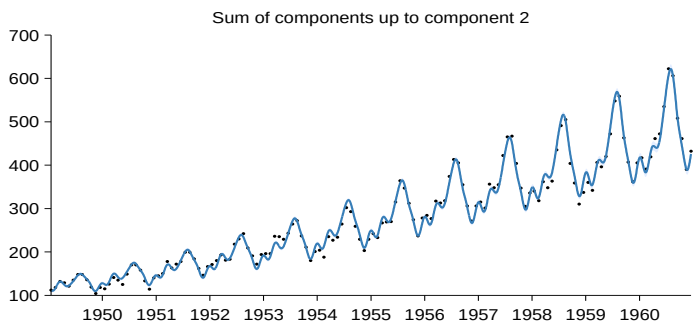


THIS ANALYSIS WAS AUTOMATICALLY GENERATED

2.2 Component 2

This component is approximately periodic with a period of 1.0 years and varying amplitude. Across periods the shape of this function varies very smoothly. The amplitude of the function increases approximately linearly. The shape of this function within each period has a typical lengthscale of 6.0 weeks.





SUMMARY

- Constructed a language of regression models through kernel composition
- Searched over this language greedily
- Kernel sums and products modify prior in predictable ways, allowing automatic natural-language description of models

SUMMARY

How could an AI do statistics?

- Grammars over composite structures are a simple way to specify open-ended model classes.
- Composite structures often imply interpretable decompositions of the data.
- Searching over these model classes is a step towards automating statistical analysis.

A Puzzle

- How could an AI do statistics?
- An artificial statistician would need:
  - a language that could describe arbitrarily complicated models
  - a method of searching over those models
  - a procedure to check model fit

Model selection

- Looking only at tractable models
  - "Searching for the keys where the light is"
  - Selecting/weighting based on marginal likelihood only optimal when truth is in model class
  - Maybe don't trust marginal likelihood, need model checking
- What does automatic model-checking look like?
  - searching for patterns in residuals
    - need to average over free parameters to prevent overfitting...
- Is there a difference between sufficiently thorough model-checking, and marginal likelihood comparison in a larger model class?

Model selection

- Looking only at tractable models
  - "Searching for the keys where the light is"
  - Selecting/weighting based on marginal likelihood only optimal when truth is in model class
  - Maybe don't trust marginal likelihood, need model checking
- What does automatic model-checking look like?
  - searching for patterns in residuals
    - need to average over free parameters to prevent overfitting...
- Is there a difference between sufficiently thorough model-checking, and marginal likelihood comparison in a larger model class?

Thanks!

BIBLIOGRAPHY

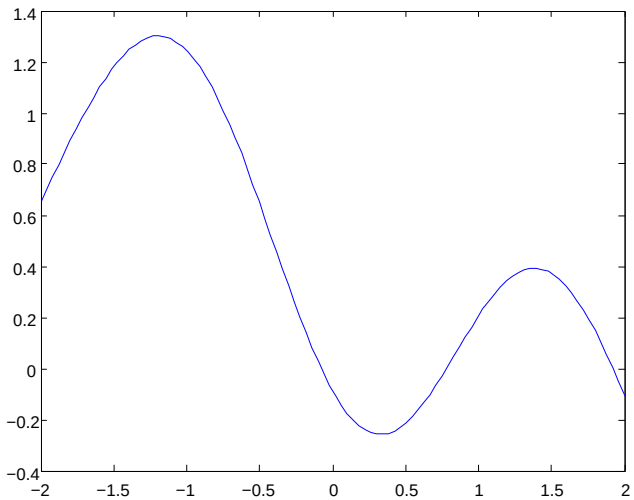
Talk based on two papers:

- Structure Discovery in Nonparametric Regression through Compositional Kernel Search [ICML 2013]
  David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, Zoubin Ghahramani
- Exploiting compositionality to explore a large space of model structures [UAI 2012]
  Roger B. Grosse, Ruslan Salakhutdinov, William T. Freeman, Joshua B. Tenenbaum

Extra Slides

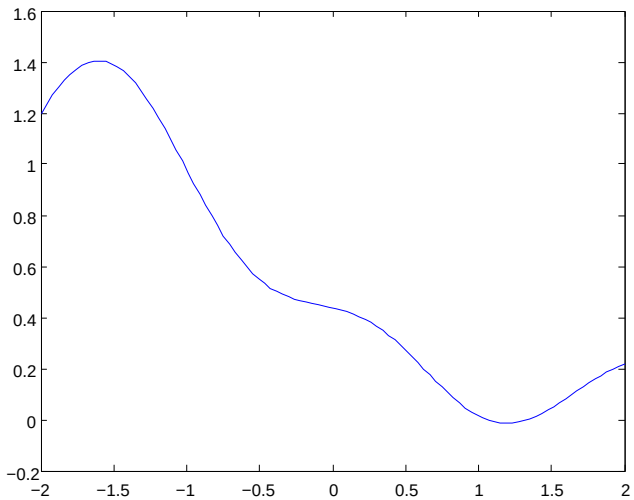
SAMPLING FROM A GP

```
function simple_gp_sample

   % Choose a set of x locations.
   N = 100;
   x = linspace( -2, 2, N);

   % Specify covariance between all f(x)
   for j = 1:N
      for k = 1:N
         sigma(j,k) = covariance( x(j), x(k) );
      end
   end

   % Specify that prior mean of f is zero.
   mu = zeros(N, 1);

   % Sample from a multivariate Gaussian.
   f = mvnrnd( mu, sigma );

   plot(x, f);
end

% Squared-exp covariance function.
function k = covariance(x, y)
   k = exp( -0.5*( x - y )^2 );
end
```



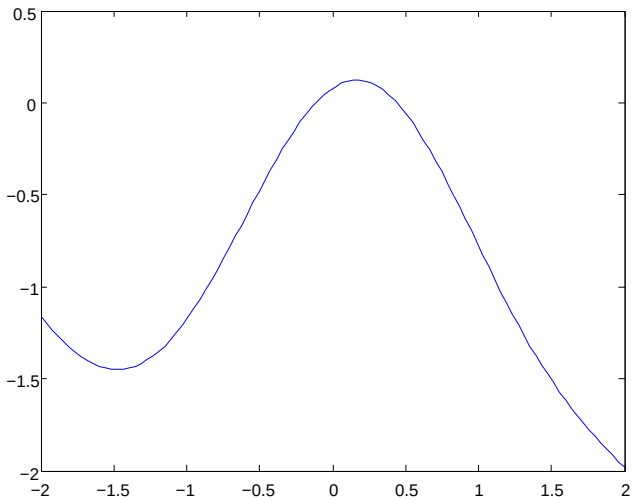
SAMPLING FROM A GP

```
function simple_gp_sample

   % Choose a set of x locations.
   N = 100;
   x = linspace( -2, 2, N);

   % Specify covariance between all f(x)
   for j = 1:N
      for k = 1:N
         sigma(j,k) = covariance( x(j), x(k) );
      end
   end

   % Specify that prior mean of f is zero.
   mu = zeros(N, 1);

   % Sample from a multivariate Gaussian.
   f = mvnrnd( mu, sigma );

   plot(x, f);
end

% Squared-exp covariance function.
function k = covariance(x, y)
   k = exp( -0.5*( x - y )^2 );
end
```



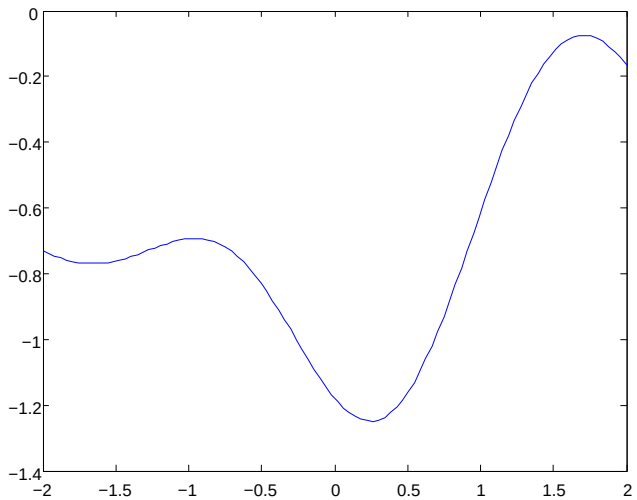
SAMPLING FROM A GP

```
function simple_gp_sample

   % Choose a set of x locations.
   N = 100;
   x = linspace( -2, 2, N);

   % Specify covariance between all f(x)
   for j = 1:N
      for k = 1:N
         sigma(j,k) = covariance( x(j), x(k) );
      end
   end

   % Specify that prior mean of f is zero.
   mu = zeros(N, 1);

   % Sample from a multivariate Gaussian.
   f = mvnrnd( mu, sigma );

   plot(x, f);
end

% Squared-exp covariance function.
function k = covariance(x, y)
   k = exp( -0.5*( x - y )^2 );
end
```



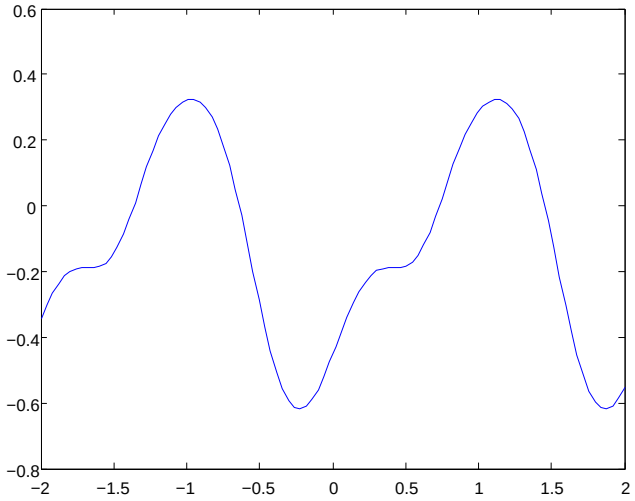
SAMPLING FROM A GP

```
function simple_gp_sample

   % Choose a set of x locations.
   N = 100;
   x = linspace( -2, 2, N);

   % Specify covariance between all f(x)
   for j = 1:N
      for k = 1:N
         sigma(j,k) = covariance( x(j), x(k) );
      end
   end

   % Specify that prior mean of f is zero.
   mu = zeros(N, 1);

   % Sample from a multivariate Gaussian.
   f = mvnrnd( mu, sigma );

   plot(x, f);
end

% Squared-exp covariance function.
function k = covariance(x, y)
   k = exp( -0.5*( x - y )^2 );
end
```



SAMPLING FROM A GP

```
function simple_gp_sample

   % Choose a set of x locations.
   N = 100;
   x = linspace( -2, 2, N);

   % Specify the covariance between function
   % values, depending on their location.
   for j = 1:N
      for k = 1:N
         sigma(j,k) = covariance( x(j), x(k) );
      end
   end

   % Specify that the prior mean of f is zero.
   mu = zeros(N, 1);

   % Sample from a multivariate Gaussian.
   f = mvnrnd( mu, sigma );

   plot(x, f);
end

% Periodic covariance function.
function c = covariance(x, y)
   c = exp( -0.5*( sin(( x - y )*1.5).^2 ));
end
```



DESCRIBING KERNELS

Products of same type of kernel collapse.

Product of Kernels	Becomes
SE $\times$ SE $\times$ SE ...	SE
LIN $\times$ LIN $\times$ LIN ...	A polynomial
PER $\times$ PER $\times$ PER	Same covariance as product of periodic functions