Automatically constructing models, and automatically explaining them, too



David Duvenaud[1], James Robert Lloyd[2], Roger Grosse[3],

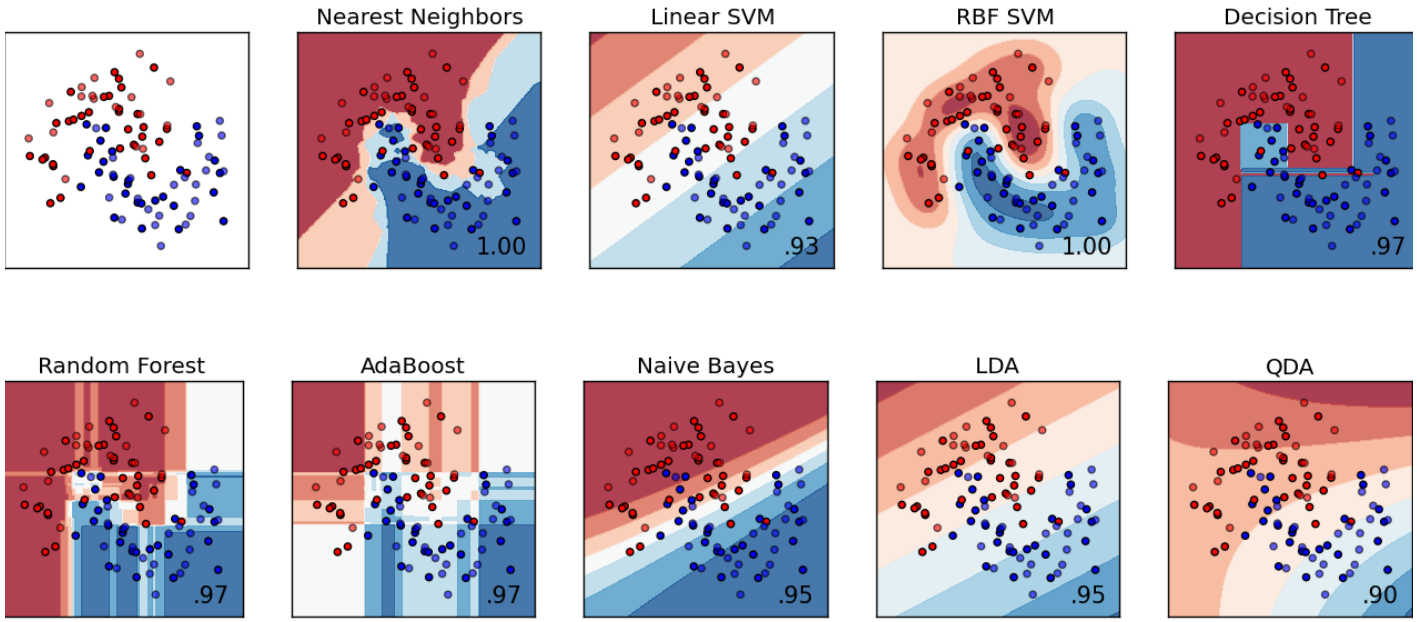

Joshua Tenenbaum[4], Zoubin Ghahramani[2]

2: University of Cambridge, 1: Harvard University
3: University of Toronto, 4: Massachusetts Institute of Technology

July 11, 2015

TYPICAL STATISTICAL MODELLING

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

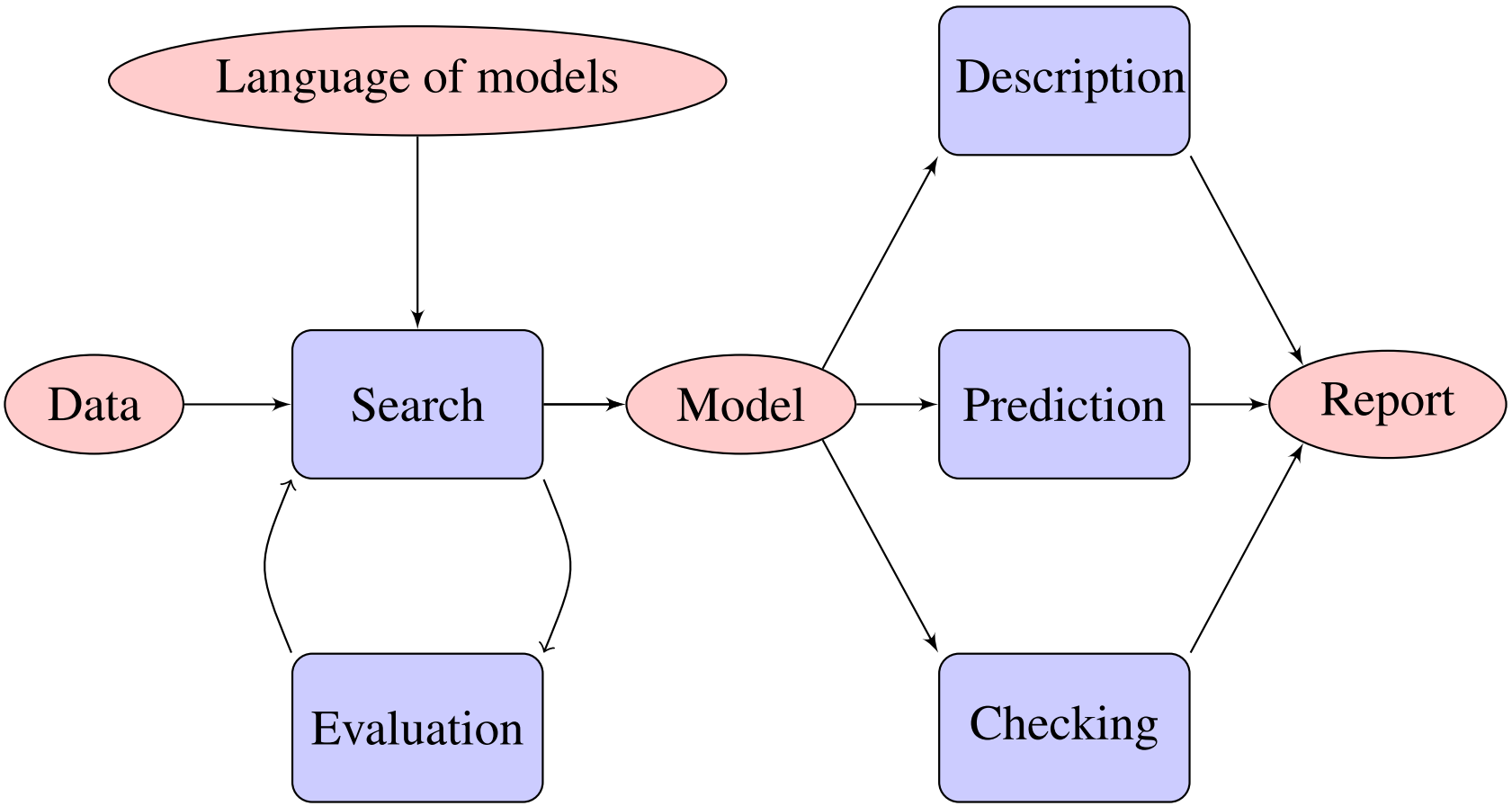


- Building by hand requires considerable expertise
- Just being nonparametric isn't good enough
  - Nonparametric does not mean assumption-free!
- Can silently fail
  - How to tell if none of the models fit the data well?

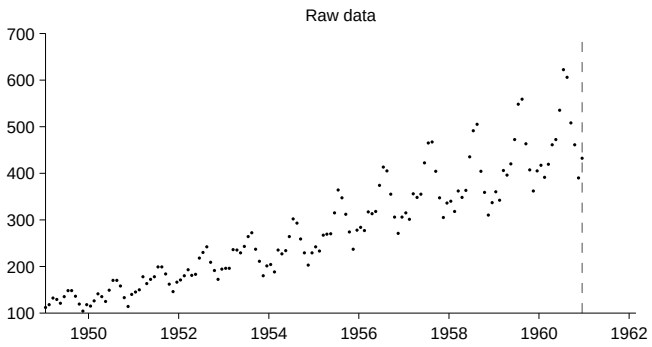
CAN WE DO BETTER?

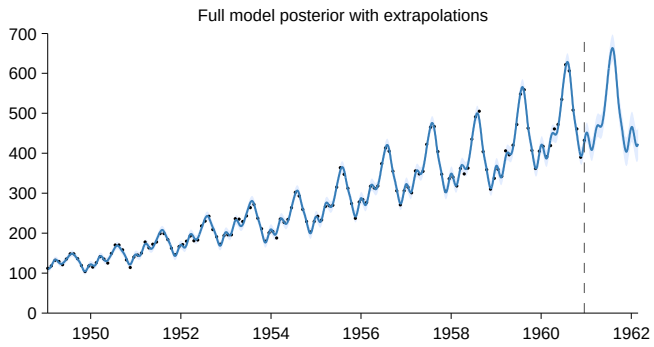
- How could an AI do modeling, forecasting, and statistics?
- An artificial statistician would need:
  - a language that could describe arbitrarily complicated models
  - a method of searching over those models
  - a procedure to check model fit
- We construct such a language over regression models,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

A SYSTEM FOR AUTOMATIC DATA ANALYSIS

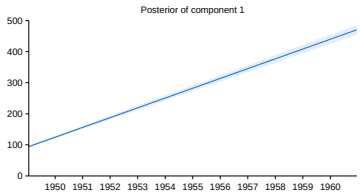


An entirely automatic analysis

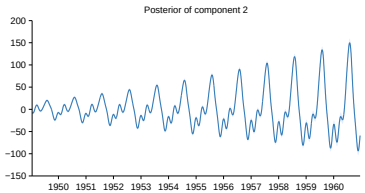




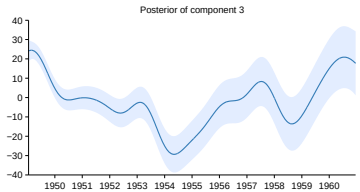
Four additive components have been identified in the data



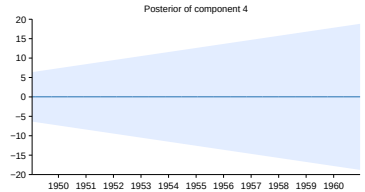
A linearly increasing function



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

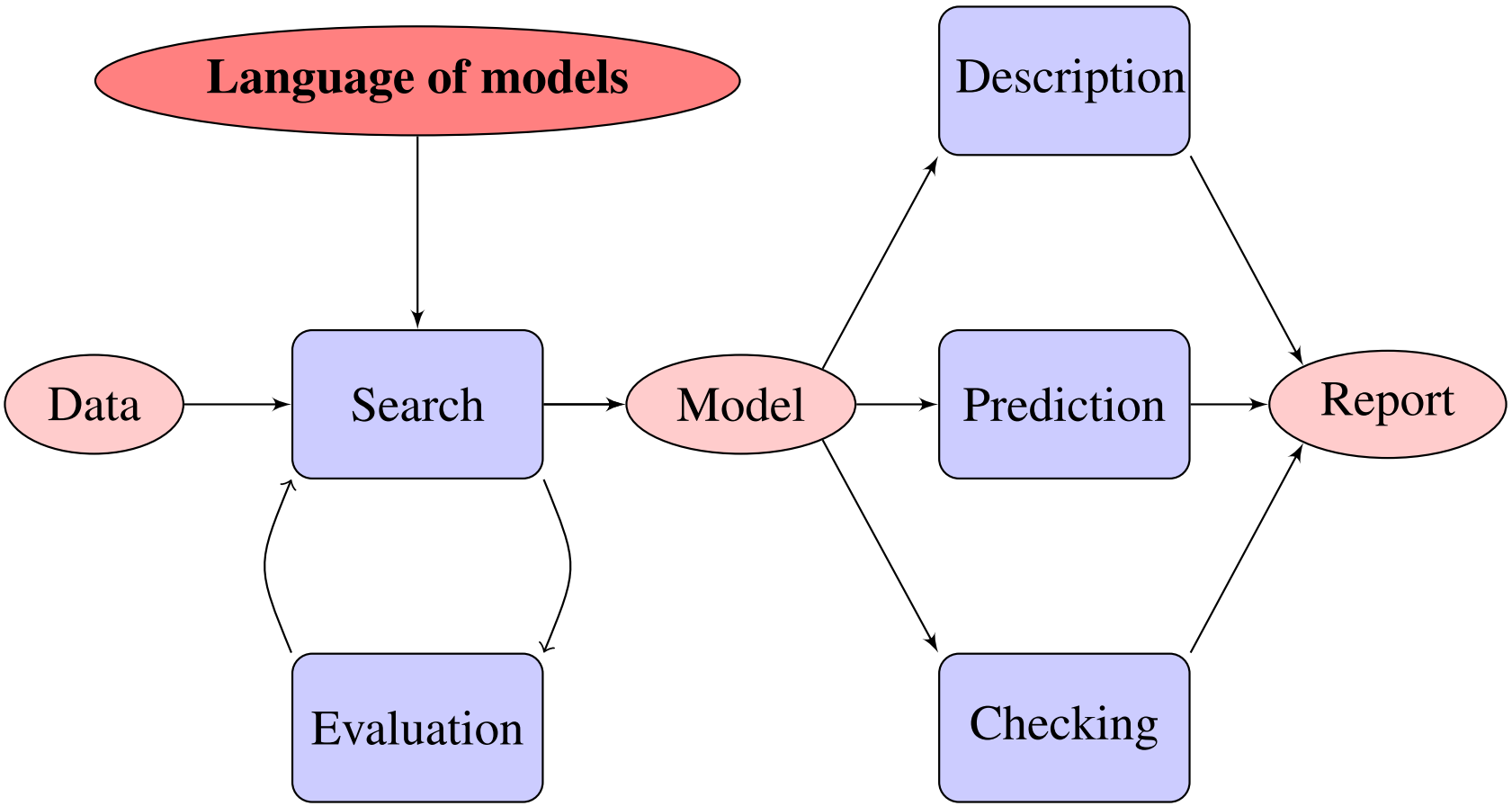


A smooth function



Uncorelated noise with linearly increasing standard deviation

Defining a language of models



A language of Gaussian process models

- A Gaussian process is collection of random variables, any finite number of which have a joint Gaussian distribution

A language of Gaussian process models

- A Gaussian process is collection of random variables, any finite number of which have a joint Gaussian distribution

- We can write this collection of random variables as $\{f(x) : x \in \mathcal{X}\}$ i.e. a function f evaluated at inputs x

A LANGUAGE OF GAUSSIAN PROCESS MODELS

- A Gaussian process is collection of random variables, any finite number of which have a joint Gaussian distribution

- We can write this collection of random variables as $\{f(x) : x \in \mathcal{X}\}$ i.e. a function f evaluated at inputs x

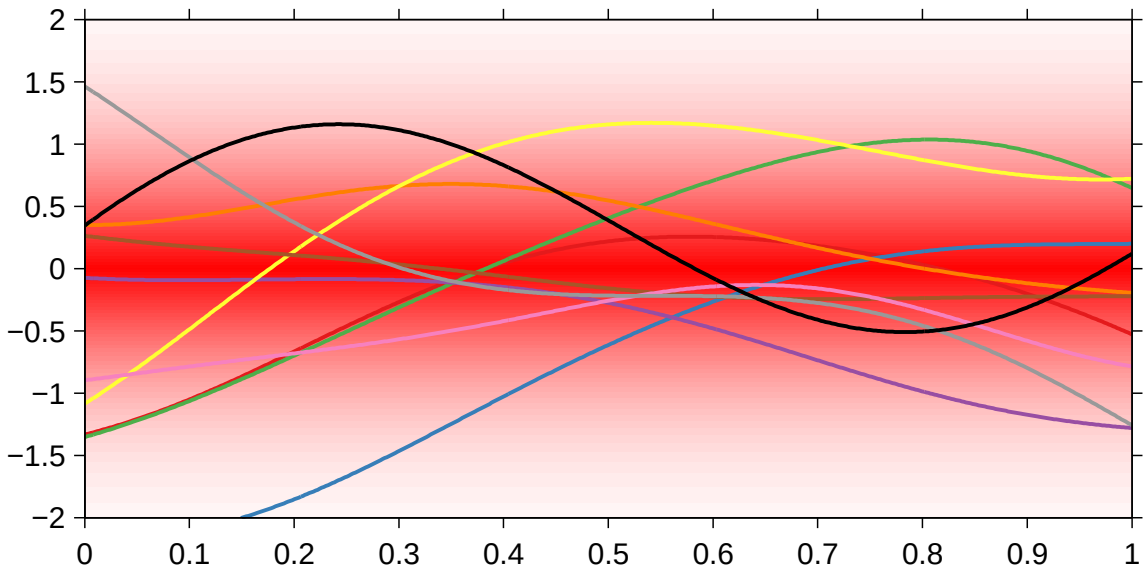
- A GP is completely specified by
  - Mean function, $\mu(x) = \mathbb{E}(f(x))$
  - Covariance / kernel function, $k(x, x') = \mathrm{Cov}(f(x), f(x'))$
  - Denoted $f \sim \mathrm{GP}(\mu, k)$

A LANGUAGE OF GAUSSIAN PROCESS MODELS

- A Gaussian process is collection of random variables, any finite number of which have a joint Gaussian distribution
- We can write this collection of random variables as $\{f(x) : x \in \mathcal{X}\}$ i.e. a function f evaluated at inputs x
- A GP is completely specified by
  - Mean function, $\mu(x) = \mathbb{E}(f(x))$
  - Covariance / kernel function, $k(x, x') = \mathrm{Cov}(f(x), f(x'))$
  - Denoted $f \sim \mathrm{GP}(\mu, k)$
- Two important facts for this talk
  - Can be used to perform Bayesian (nonlinear) regression
  - High level properties determined by kernel

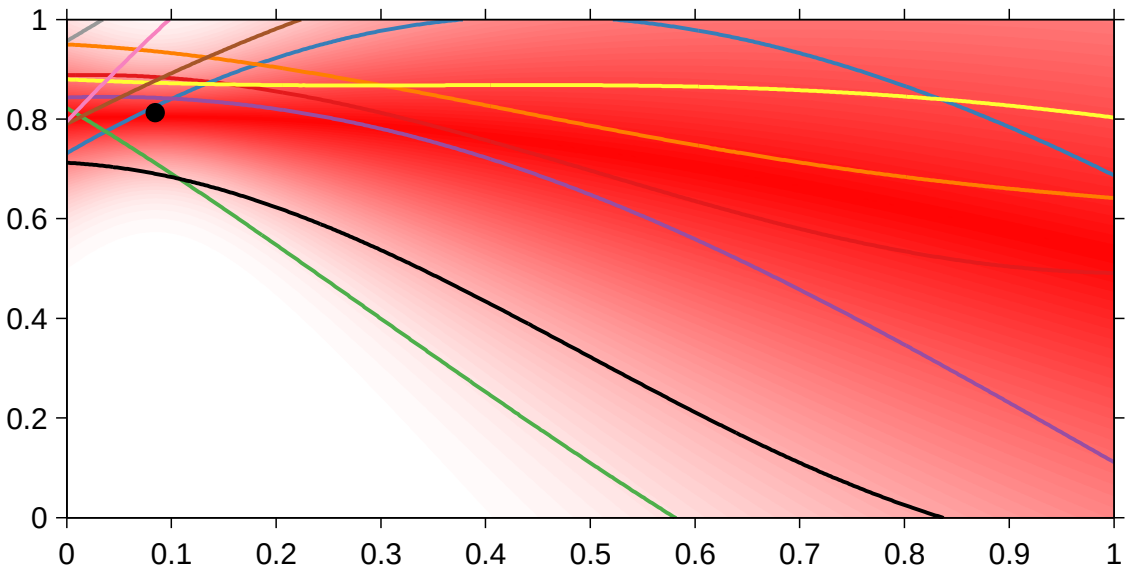
GAUSSIAN PROCESS REGRESSION IN PICTURES

- Define probability distributions on functions
- Used to perform Bayesian (nonlinear) regression
- Squared-exp kernel: $k(x, x') = \exp(-(x - x')^2/\ell)$



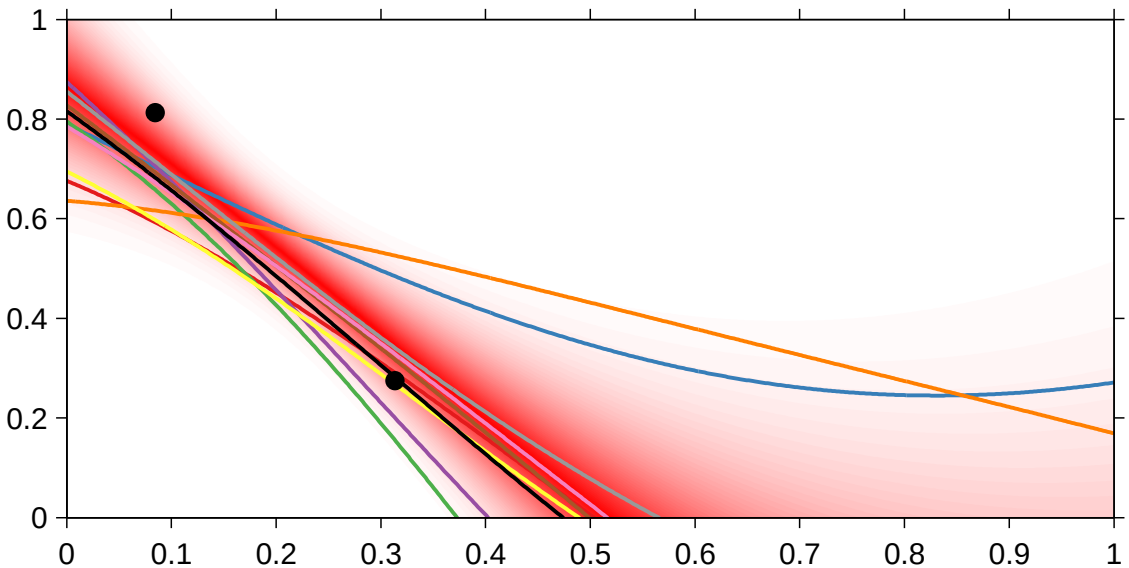
GAUSSIAN PROCESS REGRESSION IN PICTURES

- Define probability distributions on functions
- Used to perform Bayesian (nonlinear) regression
- Squared-exp kernel: $k(x, x') = \exp(-(x - x')^2/\ell)$



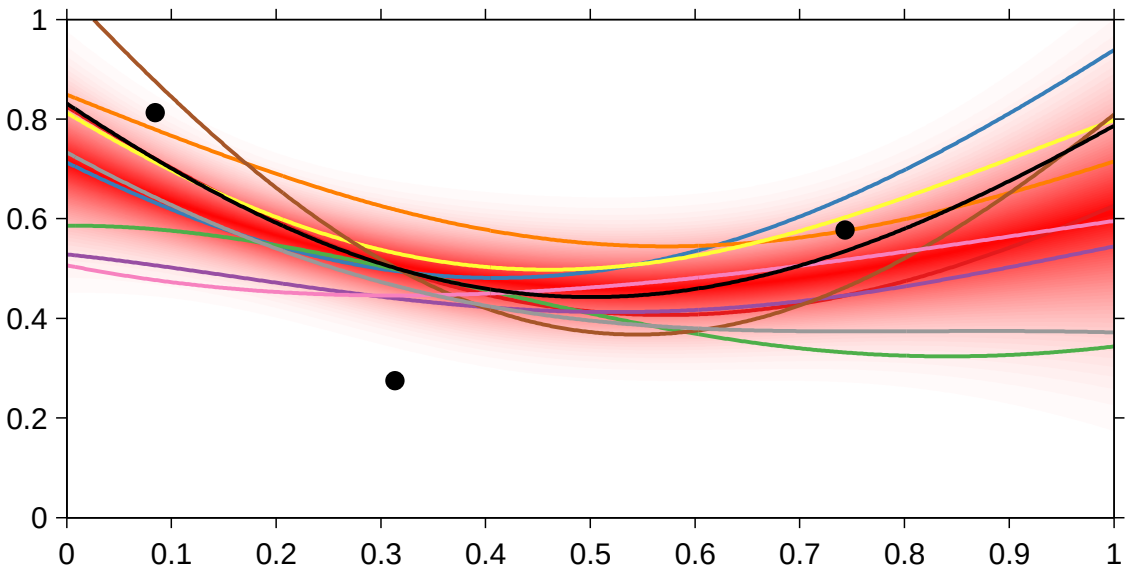
GAUSSIAN PROCESS REGRESSION IN PICTURES

- Define probability distributions on functions
- Used to perform Bayesian (nonlinear) regression
- Squared-exp kernel: $k(x, x') = \exp(-(x - x')^2/\ell)$



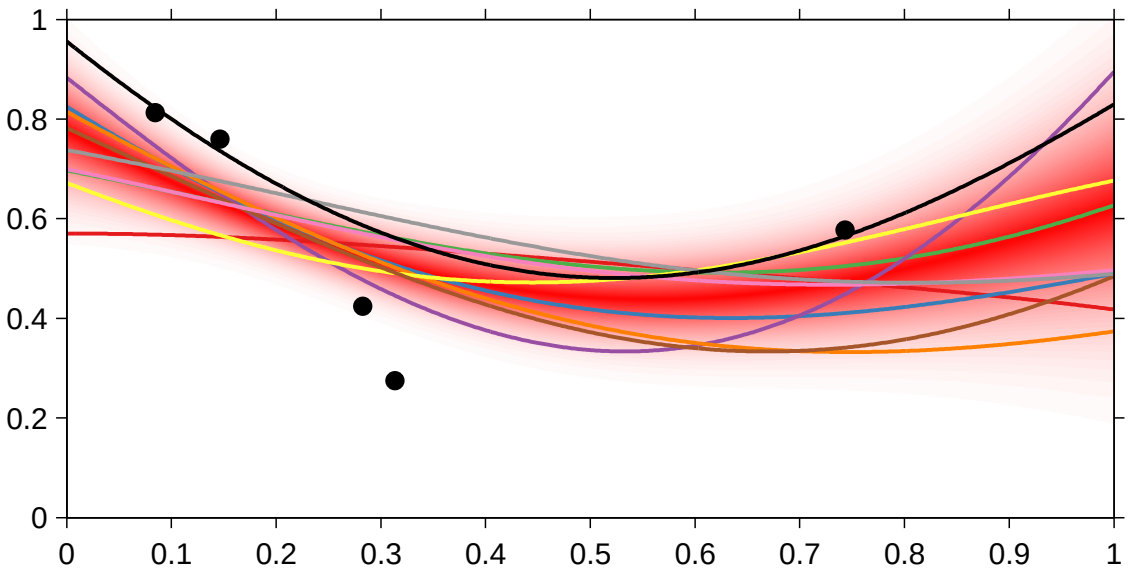
GAUSSIAN PROCESS REGRESSION IN PICTURES

- Define probability distributions on functions
- Used to perform Bayesian (nonlinear) regression
- Squared-exp kernel: $k(x, x') = \exp(-(x - x')^2/\ell)$



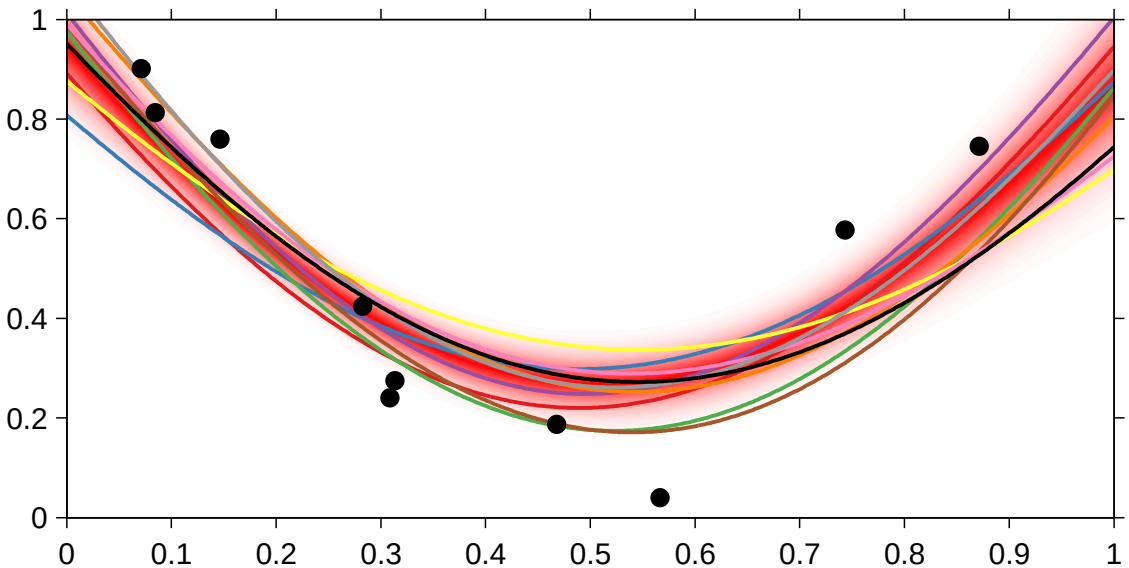
GAUSSIAN PROCESS REGRESSION IN PICTURES

- Define probability distributions on functions
- Used to perform Bayesian (nonlinear) regression
- Squared-exp kernel: $k(x, x') = \exp(-(x - x')^2/\ell)$



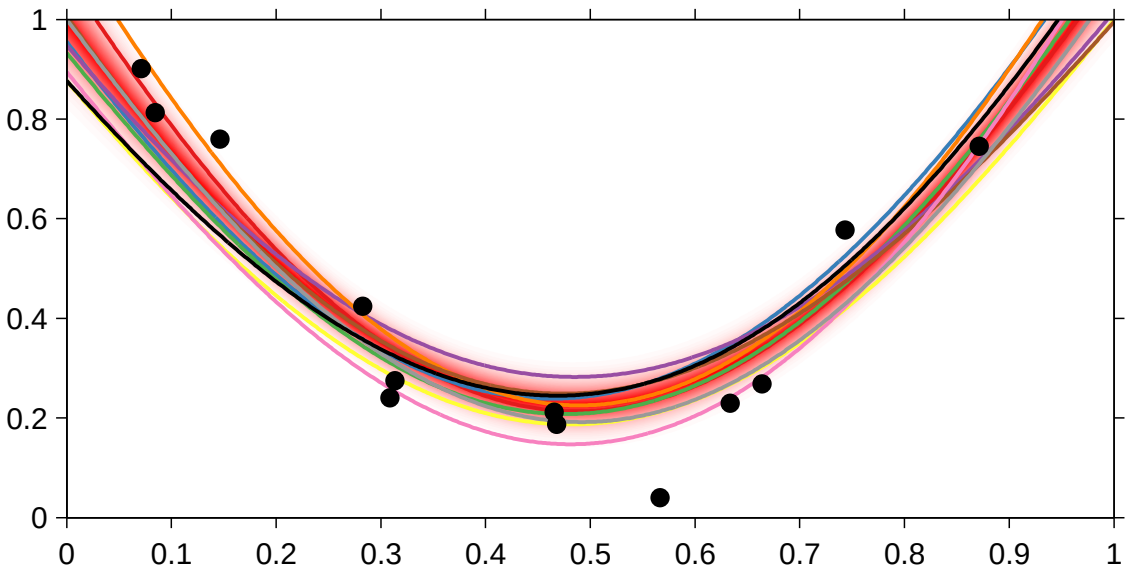
GAUSSIAN PROCESS REGRESSION IN PICTURES

- Define probability distributions on functions
- Used to perform Bayesian (nonlinear) regression
- Squared-exp kernel: $k(x, x') = \exp(-(x - x')^2/\ell)$



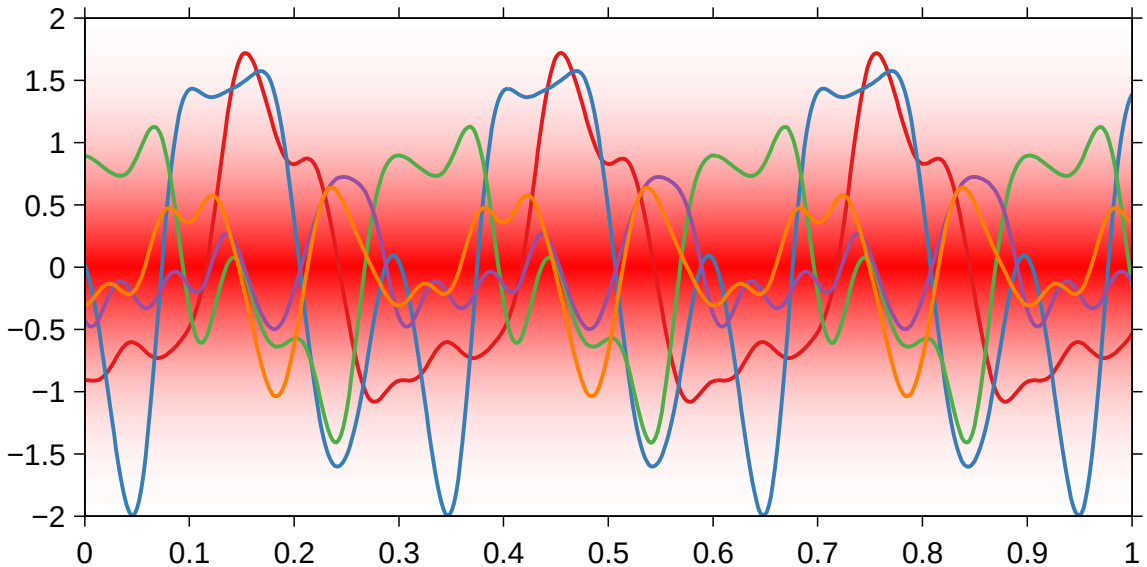
GAUSSIAN PROCESS REGRESSION IN PICTURES

- Define probability distributions on functions
- Used to perform Bayesian (nonlinear) regression
- Squared-exp kernel: $k(x, x') = \exp(-(x - x')^2/\ell)$



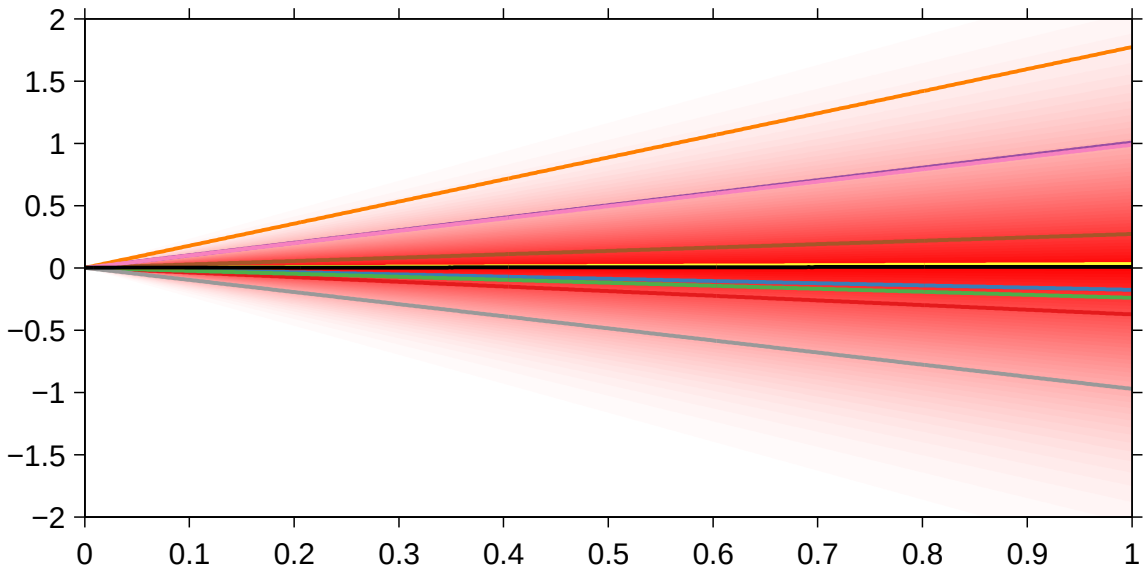
PROPERTIES SPECIFIED BY KERNEL

- The periodic kernel encodes for a probability distribution over periodic functions



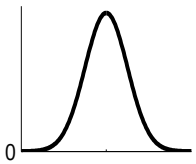
PROPERTIES SPECIFIED BY KERNEL

- The linear kernel results in a probability distribution over linear functions

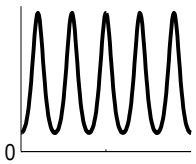


THE ATOMS OF OUR LANGUAGE

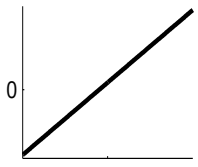
Five base kernels...



Squared exp. (SE)

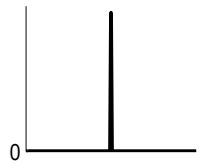


Periodic (PER)



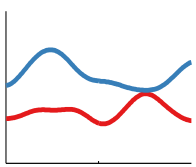
Linear (LIN)

Constant (C)

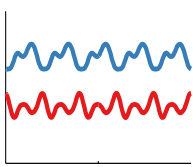


White noise (WN)

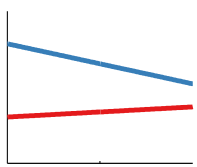
...encoding for the following types of functions



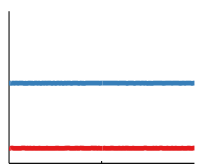
Smooth functions



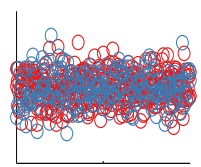
Periodic functions



Linear functions



Constant functions

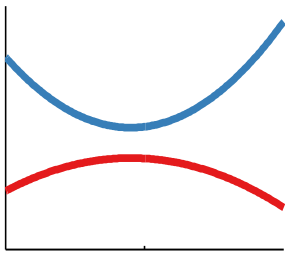


Gaussian noise

THE COMPOSITION RULES OF OUR LANGUAGE

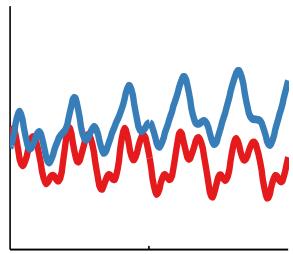
- Two main operations: addition, multiplication

LIN × LIN



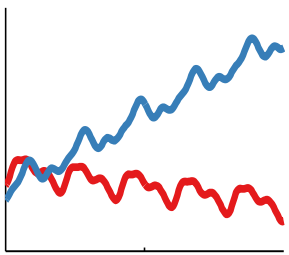
quadratic functions

SE × PER



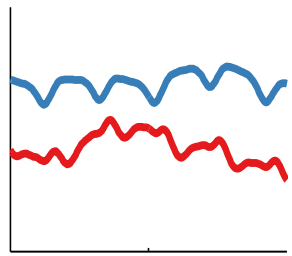
locally periodic

LIN + PER



periodic plus linear trend

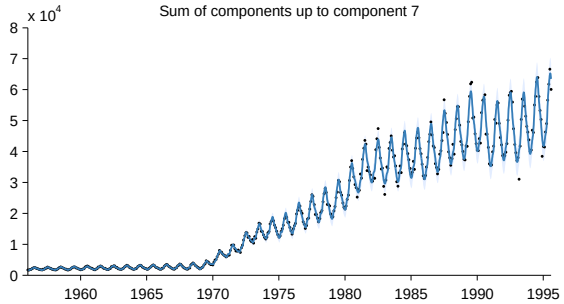
SE + PER

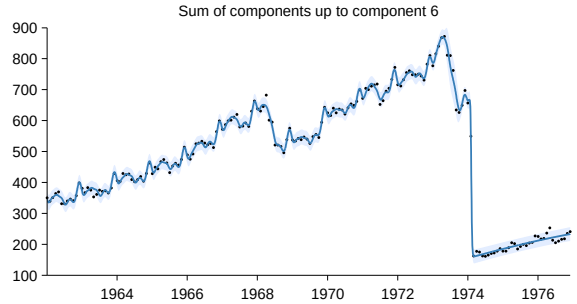


periodic plus smooth trend

MODELING CHANGEPOINTS

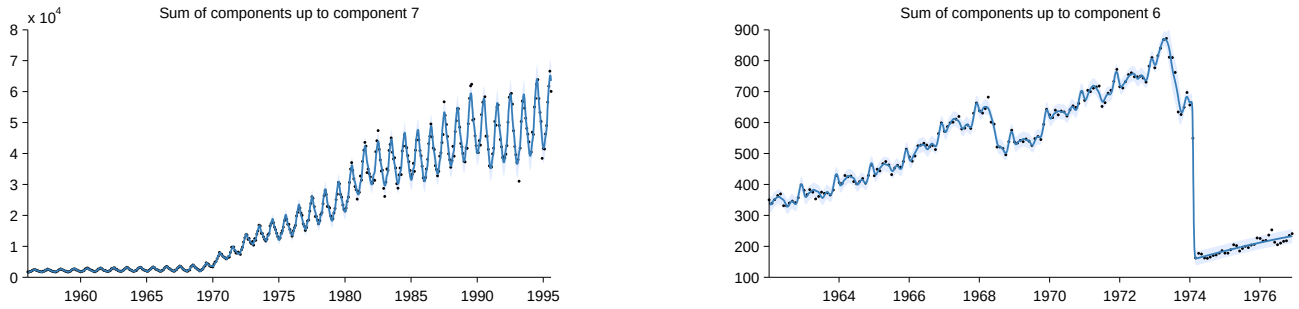
Time series data often exhibit changepoints:





MODELING CHANGEPOINTS

Time series data often exhibit changepoints:



We can model this by assuming $f_1(x) \sim \text{GP}(0, k_1)$ and $f_2(x) \sim \text{GP}(0, k_2)$ and then defining

$$f(x) = (1 - \sigma(x)) f_1(x) + \sigma(x) f_2(x)$$

where σ is a sigmoid function between 0 and 1.

Modeling changepoints

We can model this by assuming $f_1(x) \sim \mathrm{GP}(0, k_1)$ and $f_2(x) \sim \mathrm{GP}(0, k_2)$ and then defining

$$f(x) = (1 - \sigma(x)) f_1(x) + \sigma(x) f_2(x)$$

where σ is a sigmoid function between 0 and 1.

Then $f \sim \mathrm{GP}(0, k)$, where

$$k(x, x') = (1 - \sigma(x))\, k_1(x, x')\, (1 - \sigma(x')) + \sigma(x)\, k_2(x, x')\, \sigma(x')$$

We define the changepoint operator $k = \mathrm{CP}(k_1, k_2)$.

AN EXPRESSIVE LANGUAGE OF MODELS

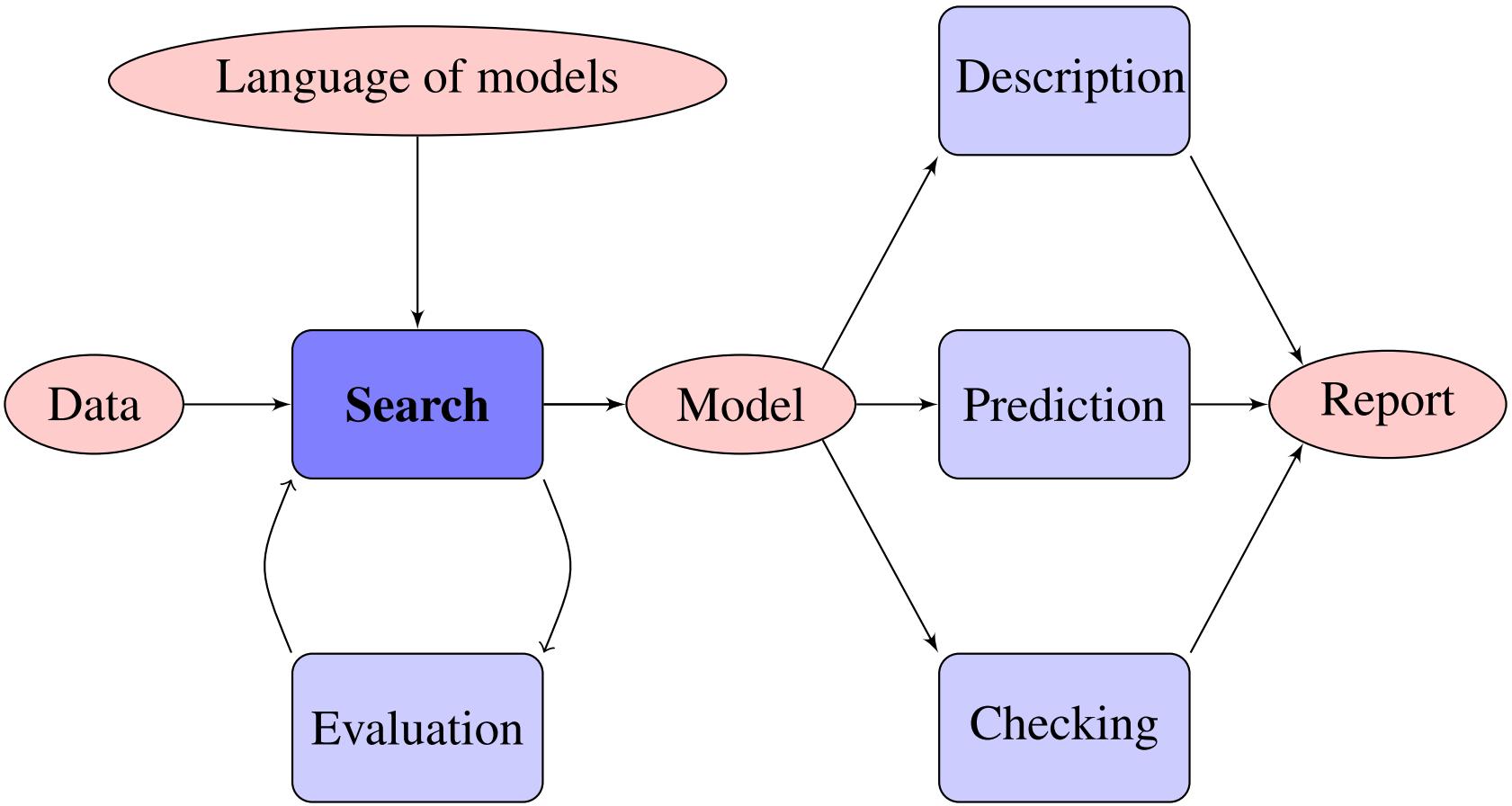
Regression model	Kernel
GP smoothing	$\mathrm{SE} + \mathrm{WN}$
Linear regression	$\mathrm{C} + \mathrm{LIN} + \mathrm{WN}$
Multiple kernel learning	$\sum \mathrm{SE} + \mathrm{WN}$
Trend, cyclical, irregular	$\sum \mathrm{SE} + \sum \mathrm{PER} + \mathrm{WN}$
Fourier decomposition	$\mathrm{C} + \sum \cos + \mathrm{WN}$
Sparse spectrum GPs	$\sum \cos + \mathrm{WN}$
Spectral mixture	$\sum \mathrm{SE} \times \cos + \mathrm{WN}$
Changepoints	e.g. $\mathrm{CP}(\mathrm{SE}, \mathrm{SE}) + \mathrm{WN}$
Heteroscedasticity	e.g. $\mathrm{SE} + \mathrm{LIN} \times \mathrm{WN}$

Note: cos is a special case of our version of PER

LANGUAGE EXTENDS TO MULTIPLE DIMENSIONS

Regression model	Kernel
Multiple linear regression	$\mathrm{C} + \sum_d \mathrm{LIN}_d + \mathrm{WN}$
Polynomial regression	$\mathrm{C} + \sum_d \prod \mathrm{SE}_d + \mathrm{WN}$
Additive smoothing	$\sum_d \mathrm{SE}_d + \mathrm{WN}$
Nonparametric ANOVA	$\sum_d \mathrm{SE}_d + \sum_{d_1,d_2} \mathrm{SE}_{d_1} \mathrm{SE}_{d_2} + \ldots$
Automatic relevance determination	$\prod_d \mathrm{SE}_d + \mathrm{WN}$

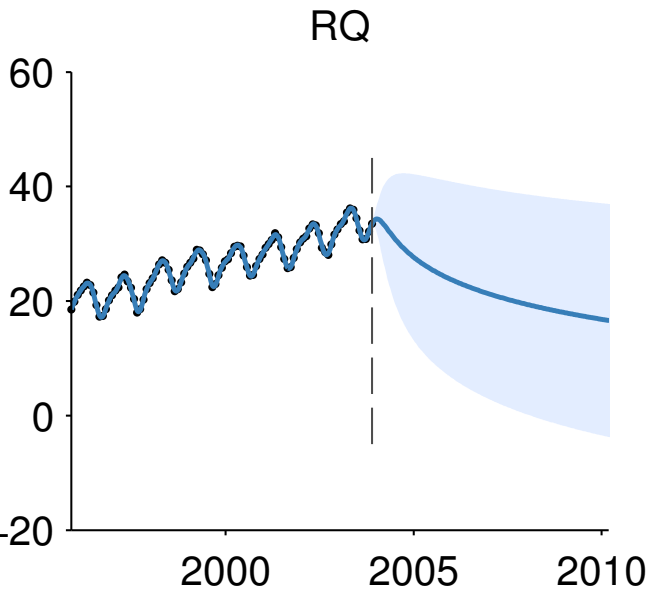
DISCOVERING A GOOD MODEL VIA SEARCH

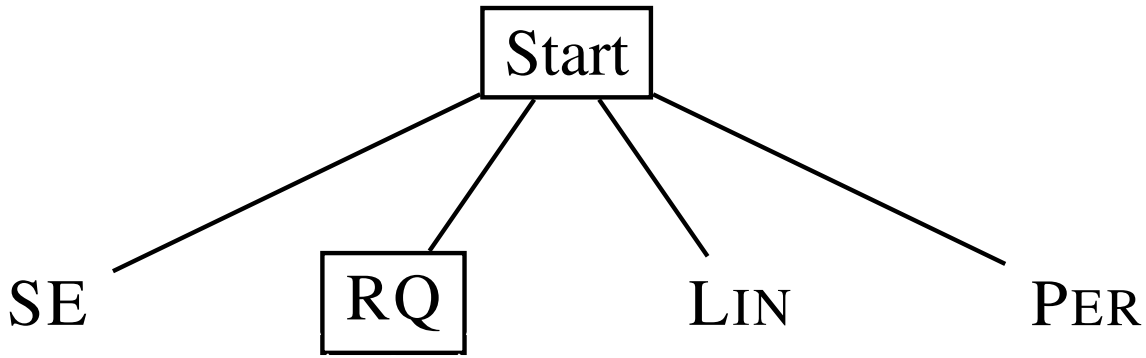


DISCOVERING A GOOD MODEL VIA SEARCH

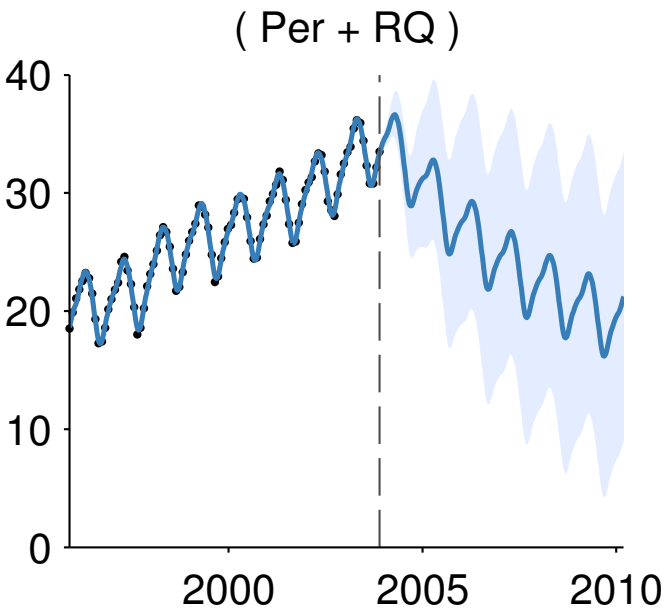
- Language defined as the arbitrary composition of five base kernels (WN, C, LIN, SE, PER) via three operators ($+$, $\times$, CP).
- The space spanned by this language is open-ended!
- We propose a greedy search for its simplicity and similarity to human model-building

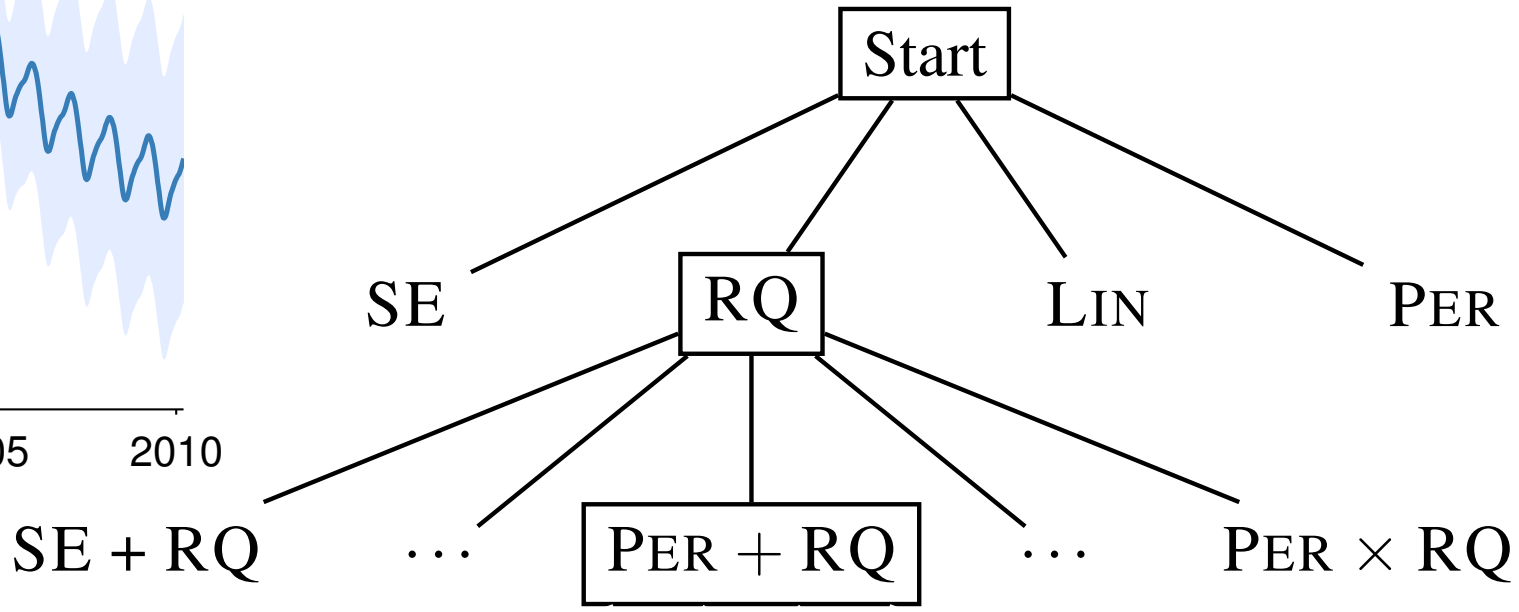
EXAMPLE: MAUNA LOA KEELING CURVE



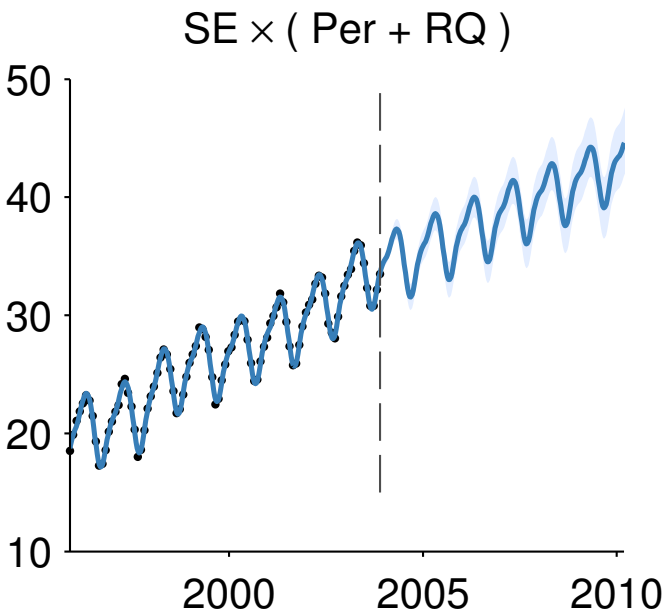


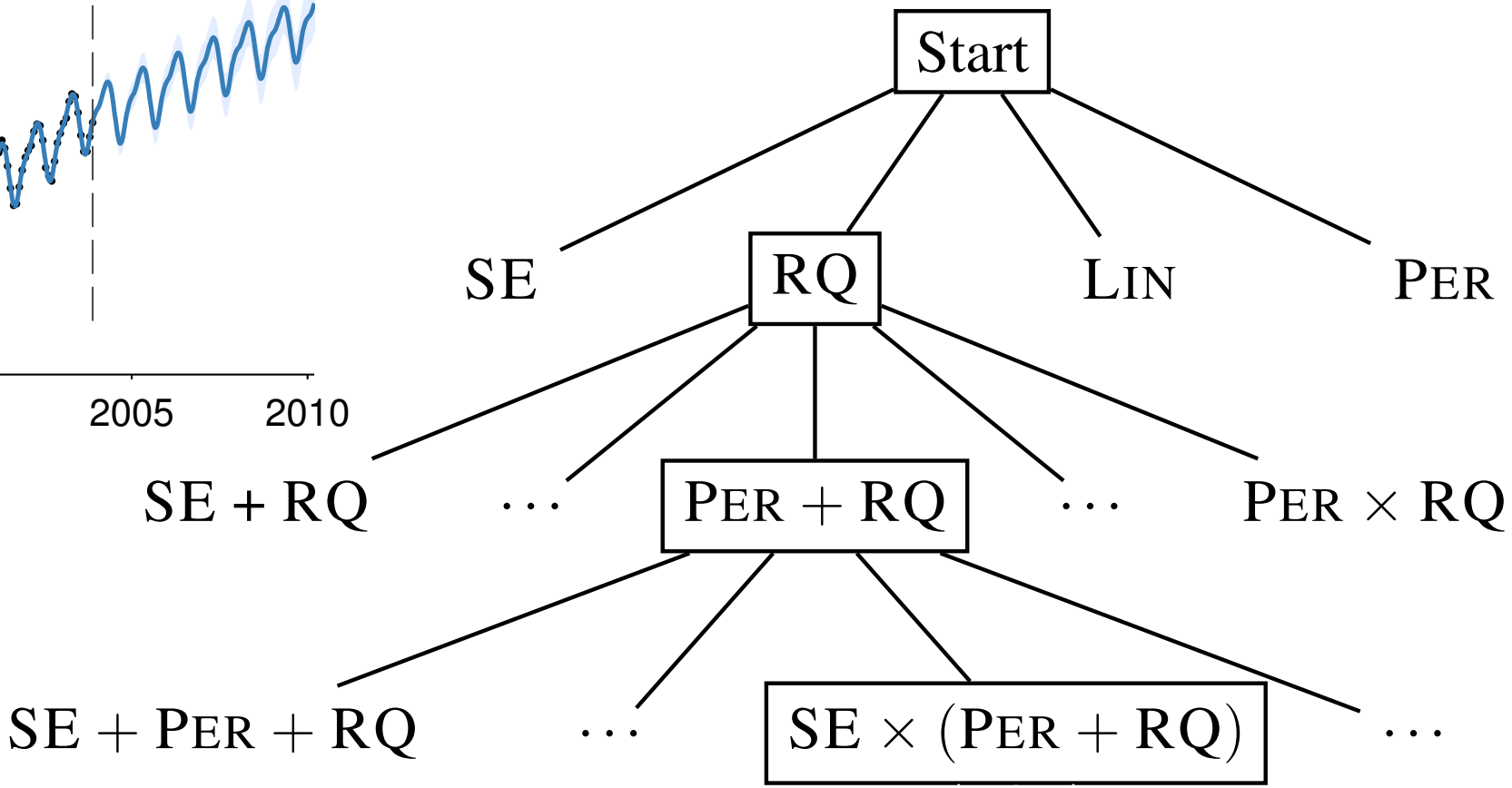
EXAMPLE: MAUNA LOA KEELING CURVE



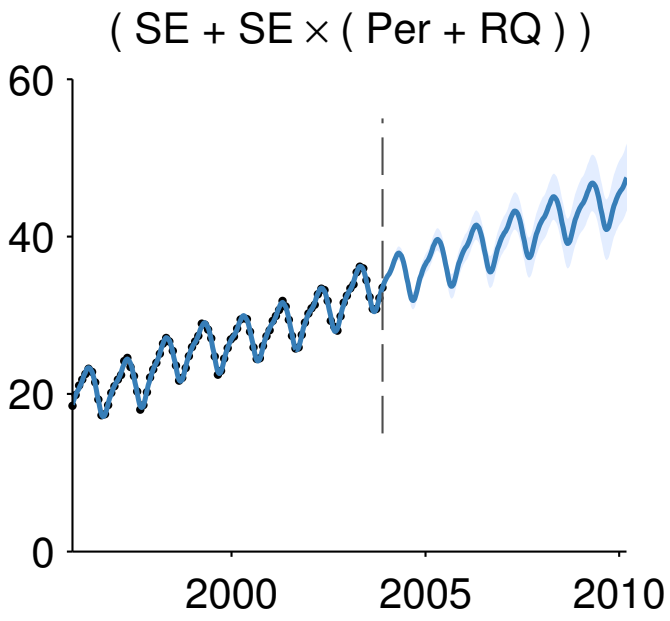


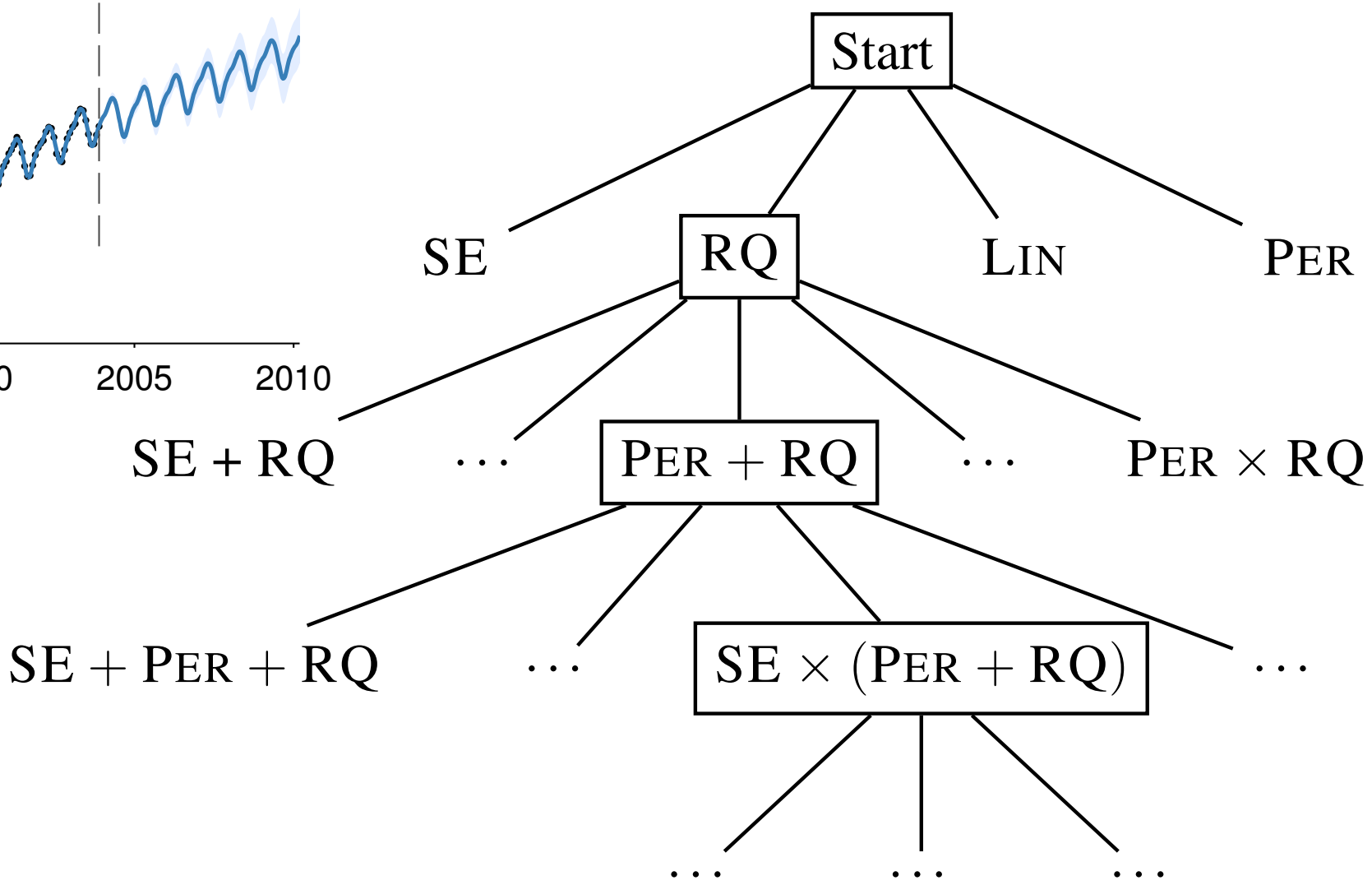
EXAMPLE: MAUNA LOA KEELING CURVE



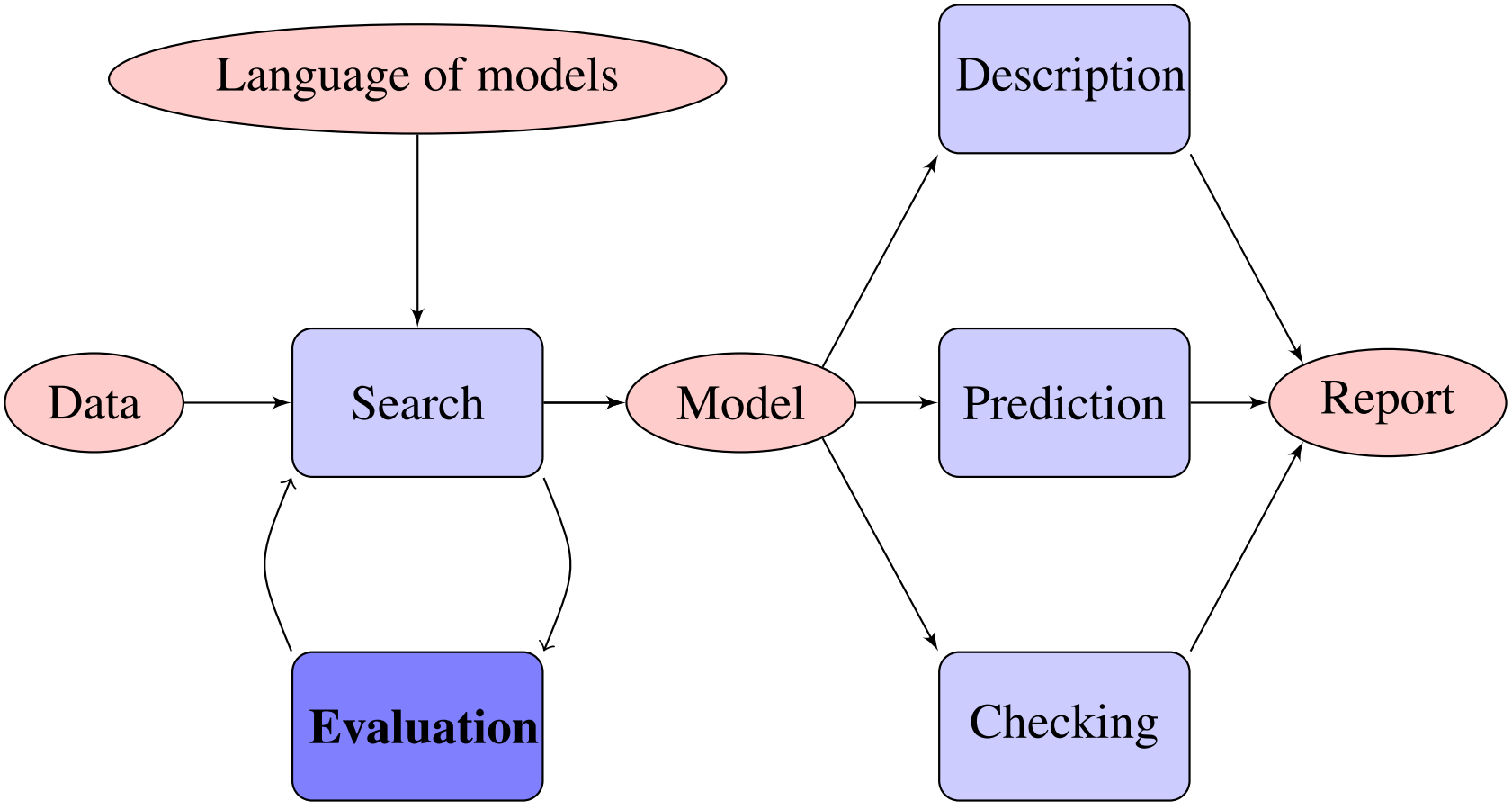


EXAMPLE: MAUNA LOA KEELING CURVE





Model evaluation



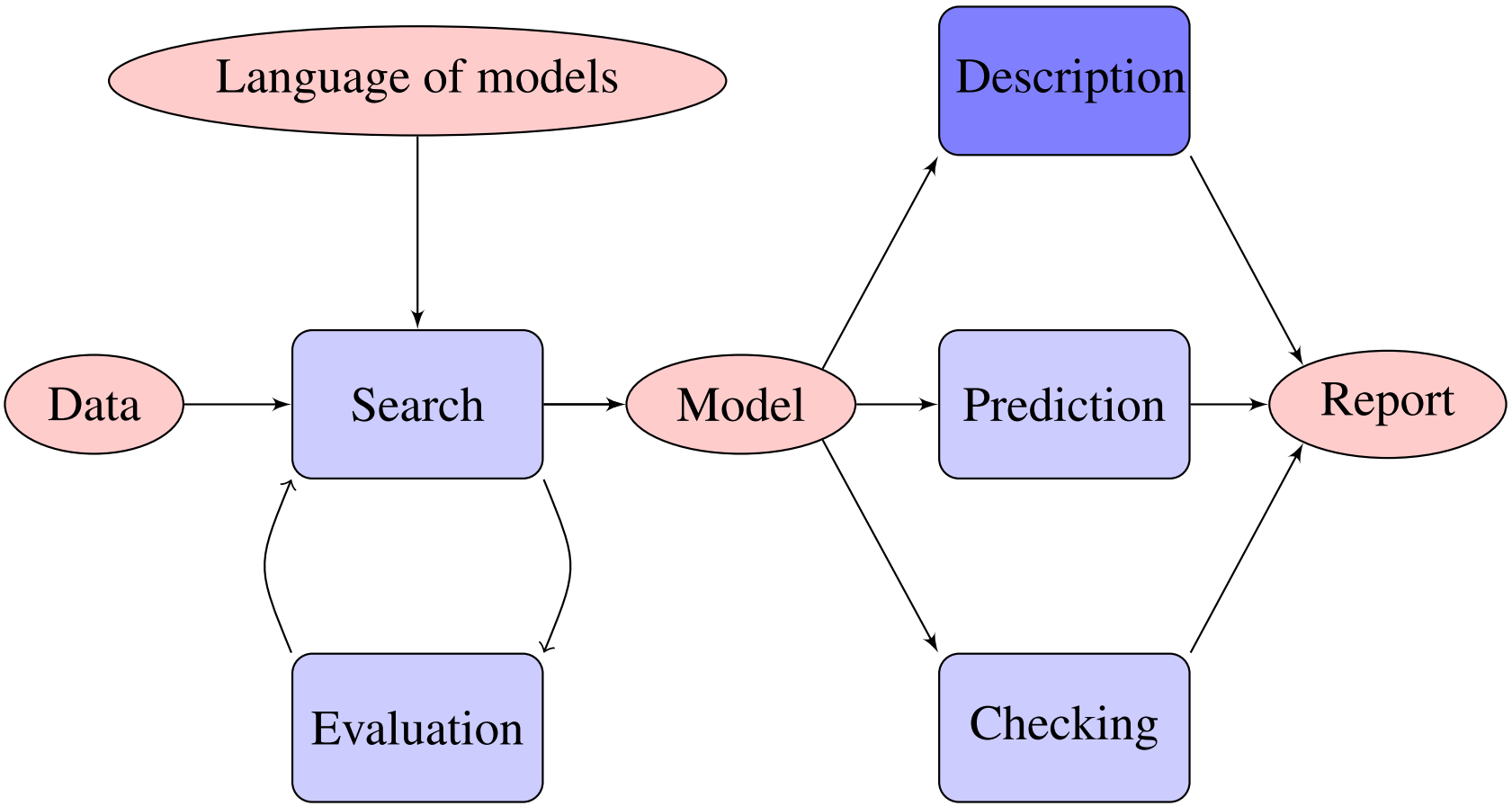
Model evaluation

- After proposing a new model its kernel parameters are optimised by conjugate gradients
- We evaluate each optimised model, M, using the marginal likelihood which can be computed analytically for GPs
- We penalise the marginal likelihood for the optimised kernel parameters using the Bayesian Information Criterion (BIC):

$$-0.5 \times \mathrm{BIC}(M) = \log p(D \mid M) - \frac{p}{2} \log n$$

where p is the number of kernel parameters, D represents the data, and n is the number of data points.

AUTOMATIC TRANSLATION OF MODELS



AUTOMATIC TRANSLATION OF MODELS

- Search can produce **arbitrarily complicated models** from open-ended language but two main properties allow description to be automated
- Kernels can be **decomposed** into a **sum of products**
  - A sum of kernels corresponds to a sum of functions
  - Therefore, we can describe each product of kernels separately
- Each kernel in a product modifies a model in a **consistent** way
  - Each kernel roughly corresponds to an adjective

SUM OF PRODUCTS NORMAL FORM

Suppose the search finds the following kernel

$$\text{SE} \times (\text{WN} \times \text{LIN} + \text{CP}(\text{C}, \text{PER}))$$

The changepoint can be converted into a sum of products

$$\text{SE} \times (\text{WN} \times \text{LIN} + \text{C} \times \boldsymbol{\sigma} + \text{PER} \times \bar{\boldsymbol{\sigma}})$$

Multiplication can be distributed over addition

$$\text{SE} \times \text{WN} \times \text{LIN} + \text{SE} \times \text{C} \times \boldsymbol{\sigma} + \text{SE} \times \text{PER} \times \bar{\boldsymbol{\sigma}}$$

Simplification rules are applied

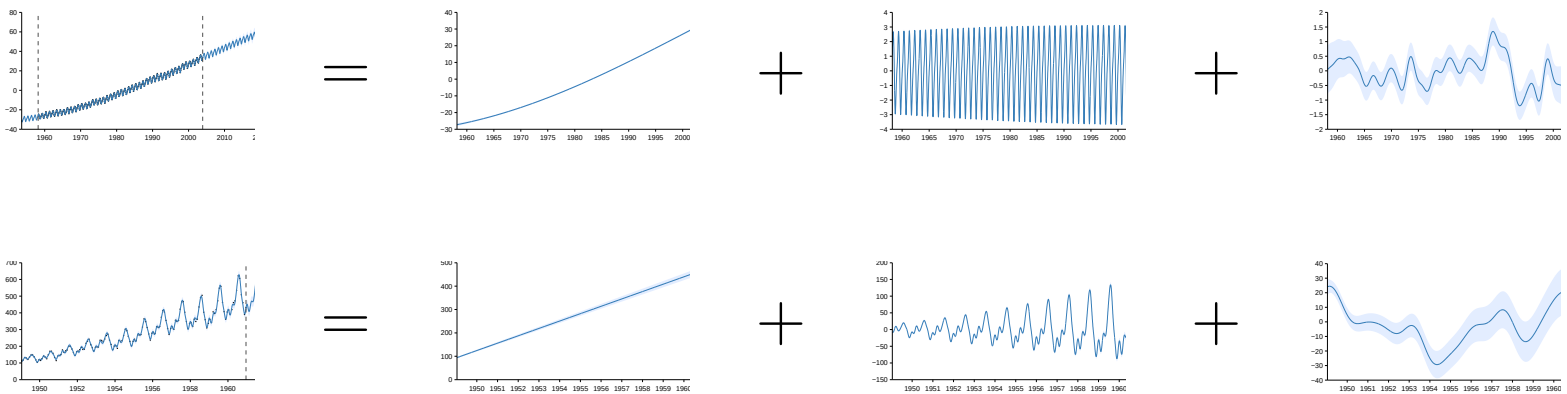
$$\text{WN} \times \text{LIN} + \text{SE} \times \boldsymbol{\sigma} + \text{SE} \times \text{PER} \times \bar{\boldsymbol{\sigma}}$$

SUMS OF KERNELS ARE SUMS OF FUNCTIONS

If $f_1 \sim \mathrm{GP}(0, k_1)$ and independently $f_2 \sim \mathrm{GP}(0, k_2)$ then

$$f_1 + f_2 \sim \mathrm{GP}(0, k_1 + k_2)$$

e.g.



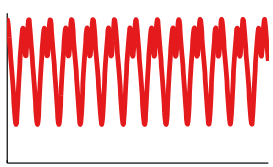
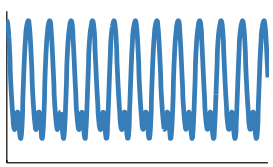
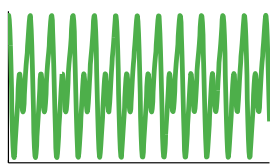
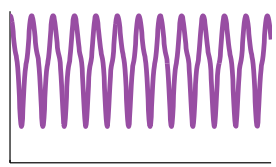
We can therefore describe each component separately

PRODUCTS OF KERNELS

PER

a periodic function

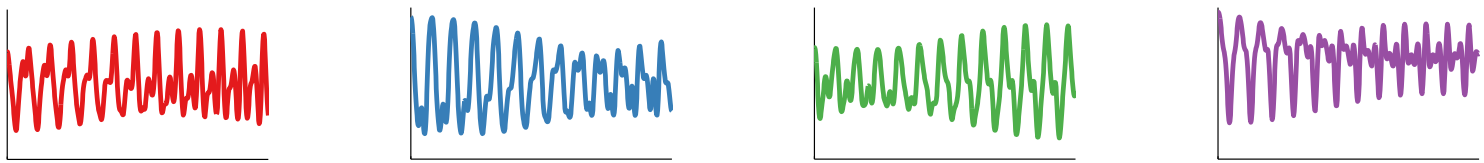
On their own, each kernel is described by a standard noun phrase

PRODUCTS OF KERNELS - SE

$$\underbrace{\text{PER}}_{\text{a periodic function}} \times \underbrace{\text{SE}}_{\text{whose shape changes smoothly}}$$

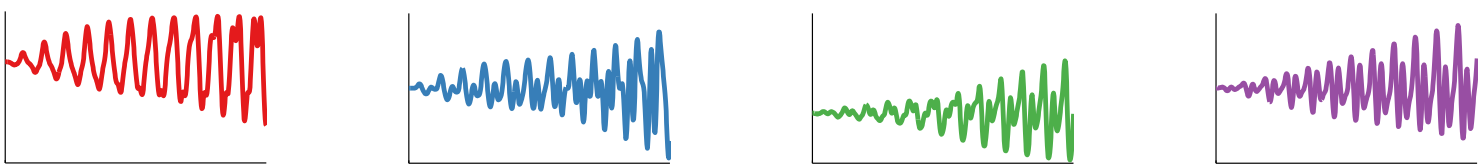
Multiplication by SE removes long range correlations from a model since $\text{SE}(x, x')$ decreases monotonically to 0 as $|x - x'|$ increases.



PRODUCTS OF KERNELS - LIN

$$\underbrace{\text{PER}}_{\text{a periodic function}} \times \underbrace{\text{SE}}_{\text{whose shape changes smoothly}} \times \underbrace{\text{LIN}}_{\text{with linearly growing amplitude}}$$

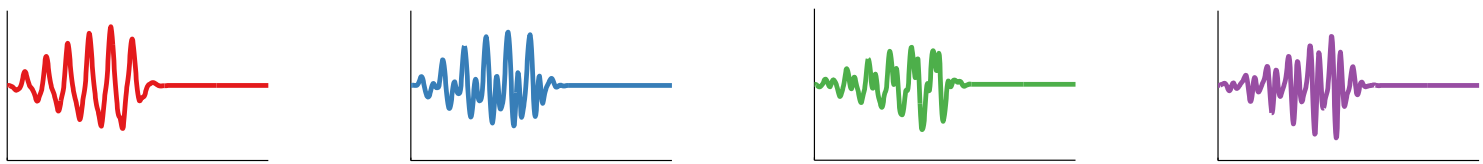
Multiplication by LIN is equivalent to multiplying the function being modeled by a linear function. If $f(x) \sim \text{GP}(0, k)$, then $xf(x) \sim \text{GP}(0, k \times \text{LIN})$. This causes the standard deviation of the model to vary linearly without affecting the correlation.



PRODUCTS OF KERNELS - CHANGEPOINTS

$$\underbrace{\text{PER}}_{\text{a periodic function}} \times \underbrace{\text{SE}}_{\text{whose shape changes smoothly}} \times \underbrace{\text{LIN}}_{\text{with linearly growing amplitude}} \times \underbrace{\sigma}_{\text{until 1700}}$$

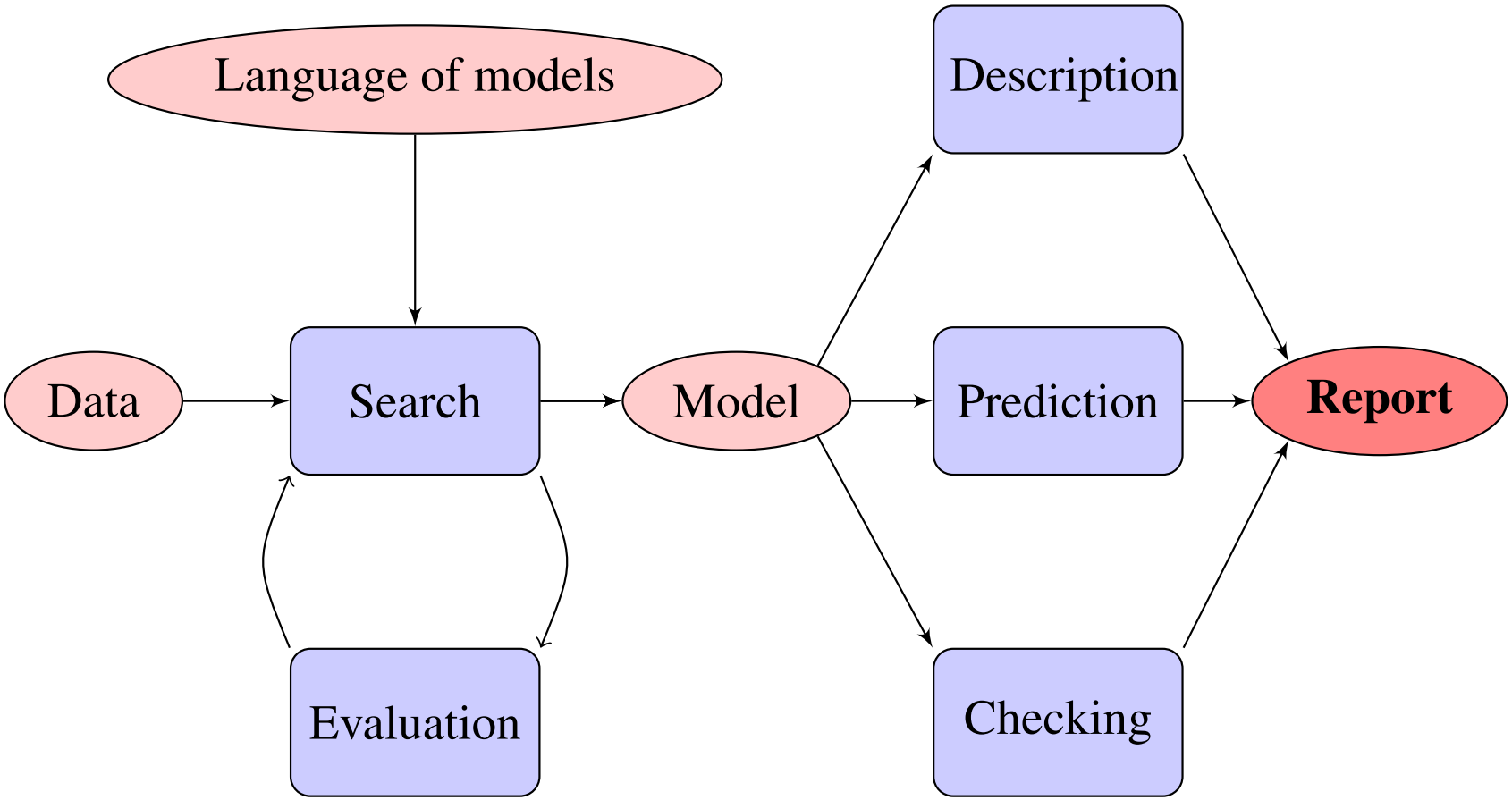
Multiplication by σ is equivalent to multiplying the function being modeled by a sigmoid.



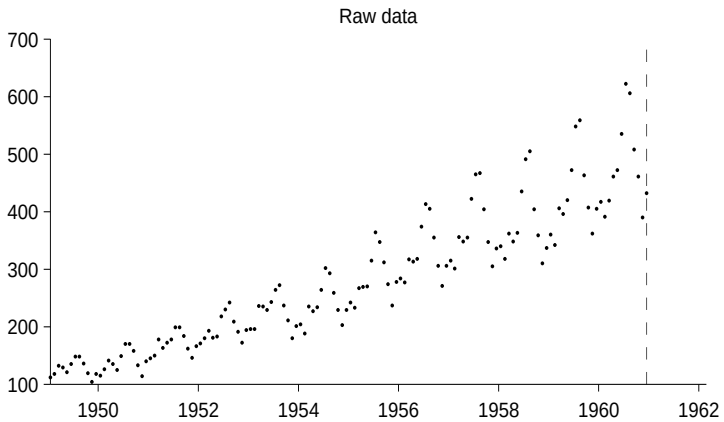
NOUN PHRASE AND POSTMODIFIER FORMS

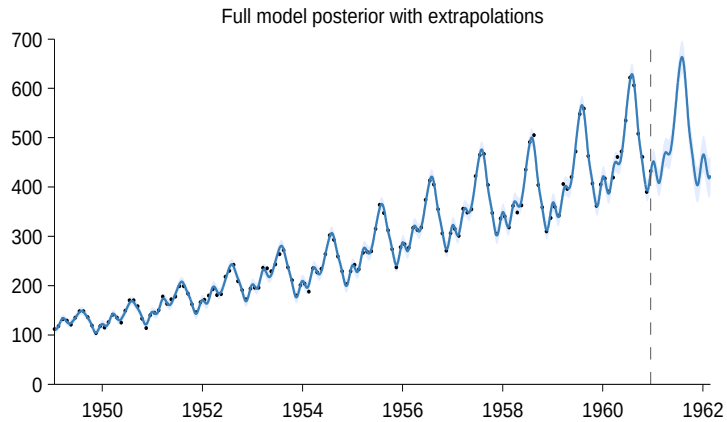
Kernel	Noun phrase	Postmodifier phrase
WN	uncorrelated noise	n/a
C	constant	n/a
SE	smooth function	whose shape changes smoothly
PER	periodic function	modulated by a periodic function
LIN	linear function	with linearly varying amplitude
$\prod_k \text{LIN}^{(k)}$	polynomial	with polynomially varying amplitude
$\prod_k \boldsymbol{\sigma}^{(k)}$	n/a	which applies until / from [changepoint]

AUTOMATICALLY GENERATED REPORTS



EXAMPLE: AIRLINE PASSENGER VOLUME



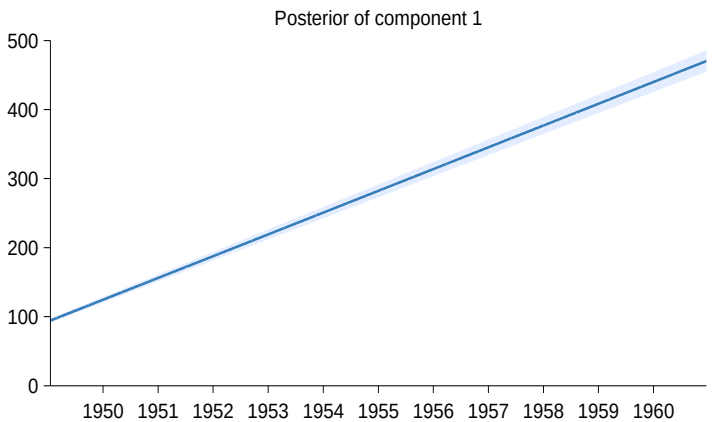


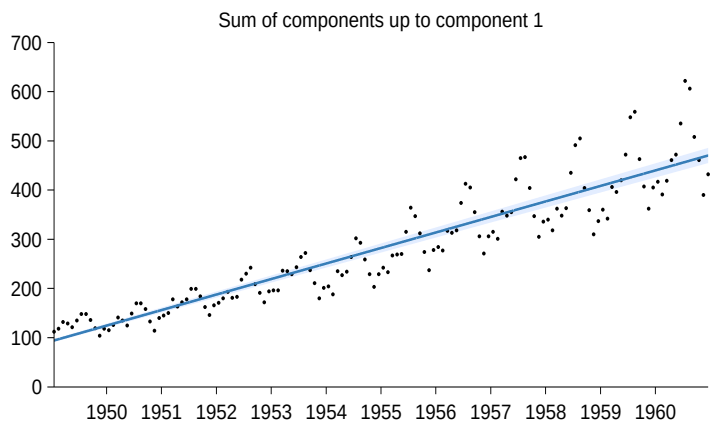
Four additive components have been identified in the data

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

Example: Airline passenger volume

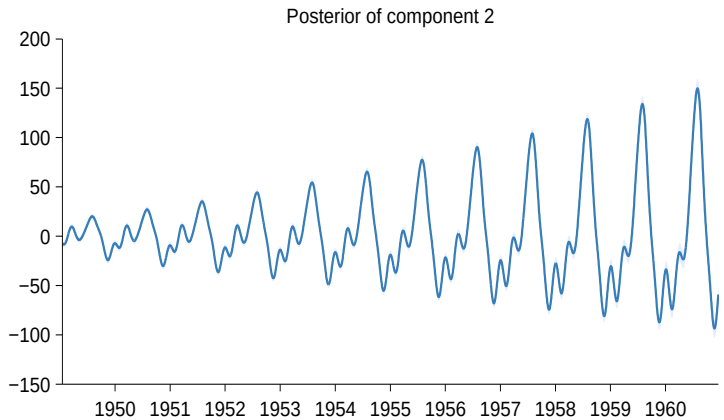
This component is linearly increasing.

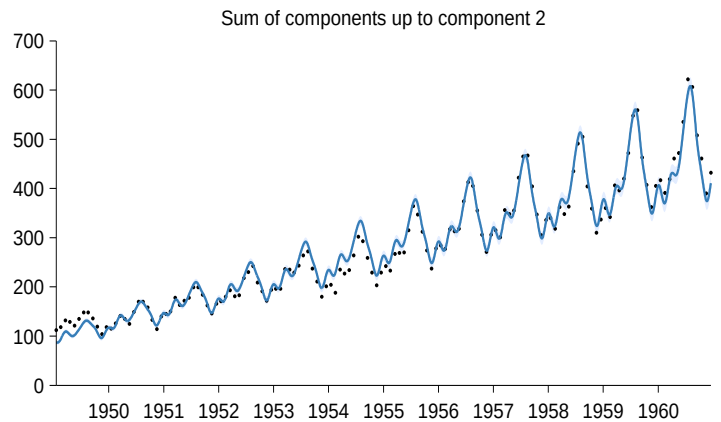




Example: Airline passenger volume

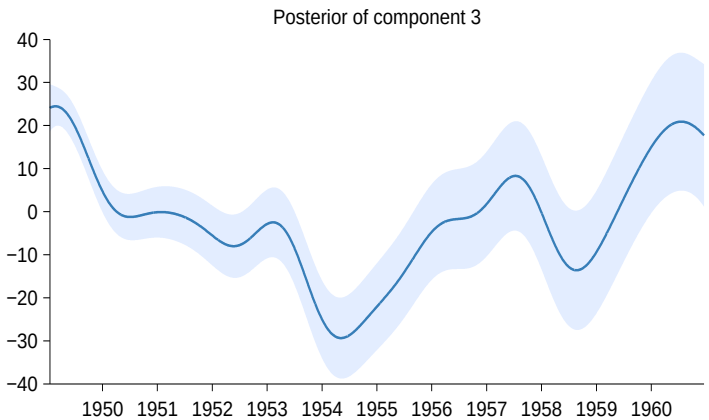
This component is approximately periodic with a period of 1.0 years and varying amplitude. Across periods the shape of this function varies very smoothly. The amplitude of the function increases linearly. The shape of this function within each period has a typical lengthscale of 6.0 weeks.

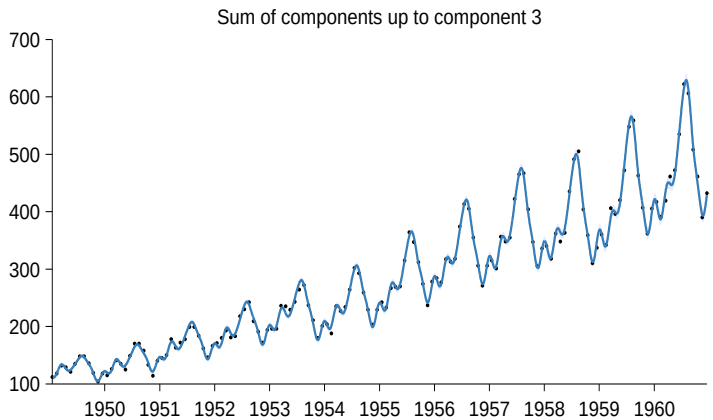




Example: Airline passenger volume

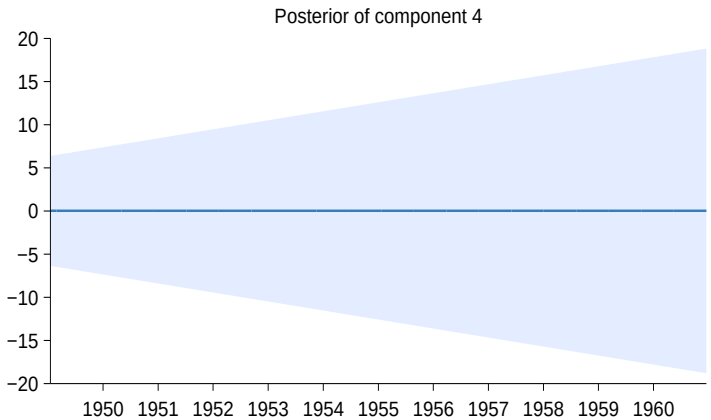
This component is a smooth function with a typical lengthscale of 8.1 months.

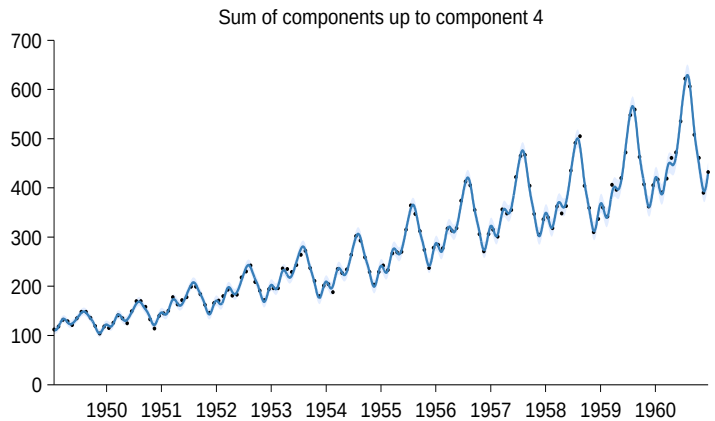




Example: Airline passenger volume

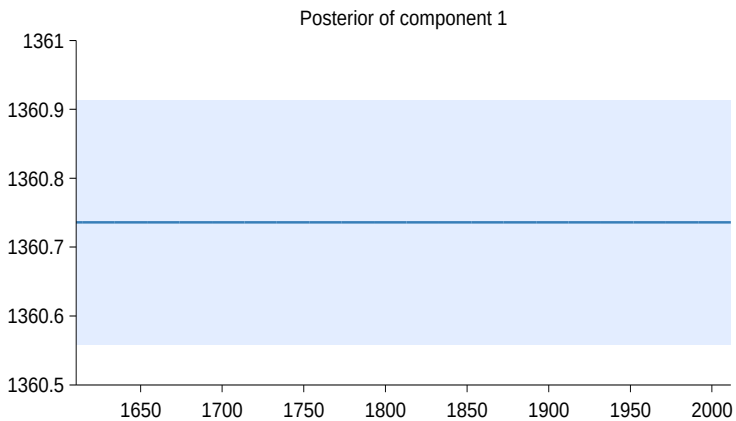
This component models uncorrelated noise. The standard deviation of the noise increases linearly.

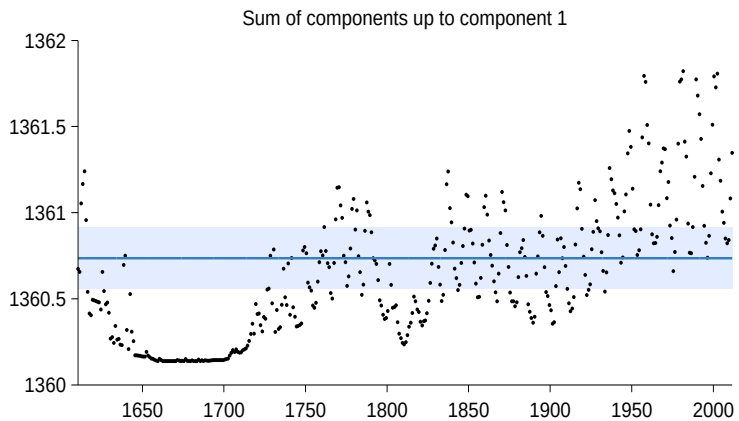




EXAMPLE: SOLAR IRRADIANCE

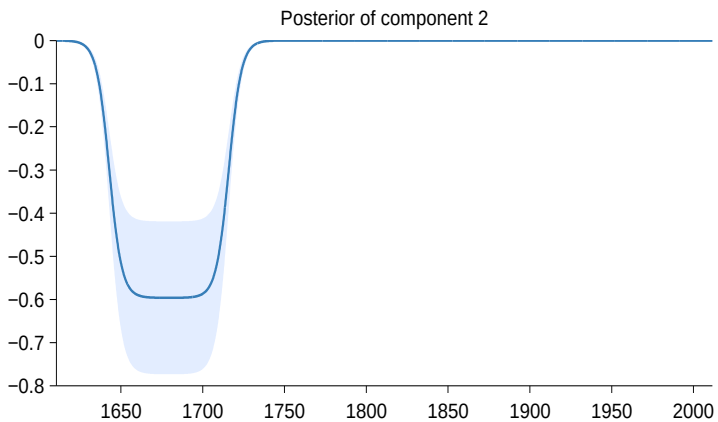
This component is constant.

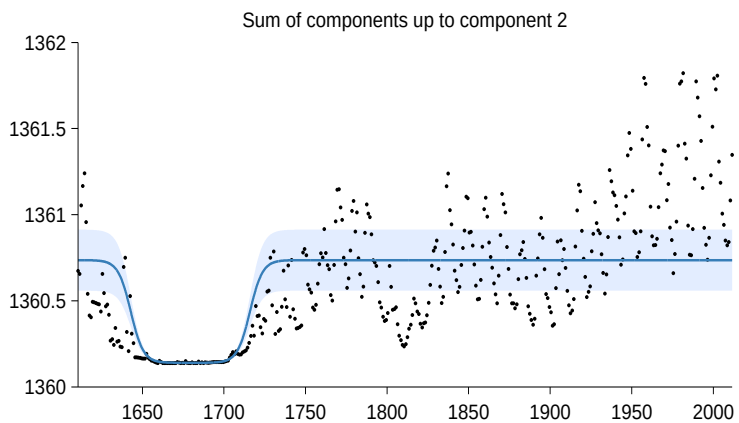




EXAMPLE: SOLAR IRRADIANCE

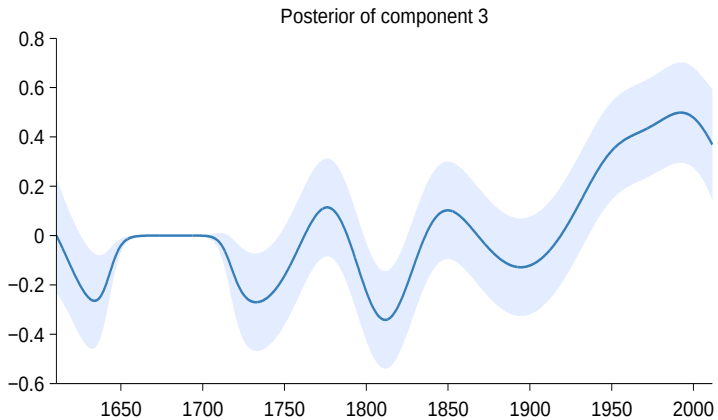
This component is constant. This component applies from 1643 until 1716.

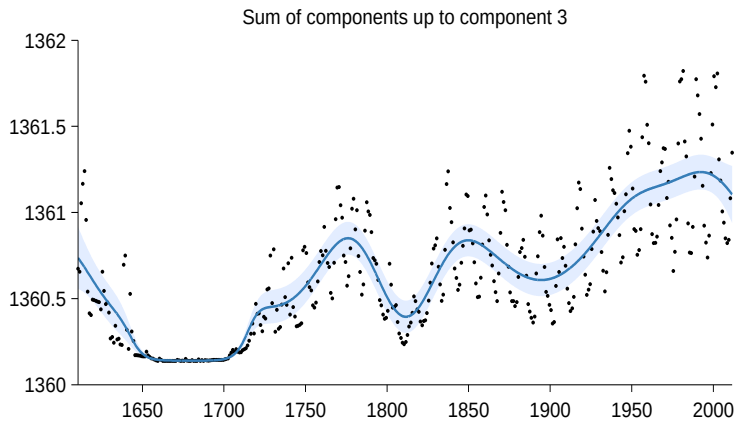




Example: Solar irradiance

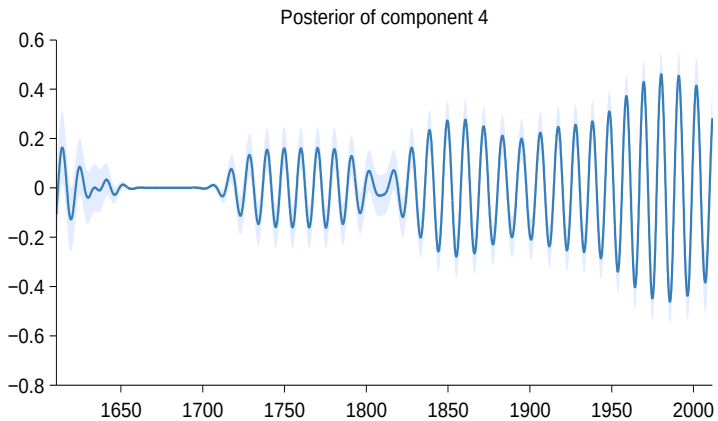
This component is a smooth function with a typical lengthscale of 23.1 years. This component applies until 1643 and from 1716 onwards.

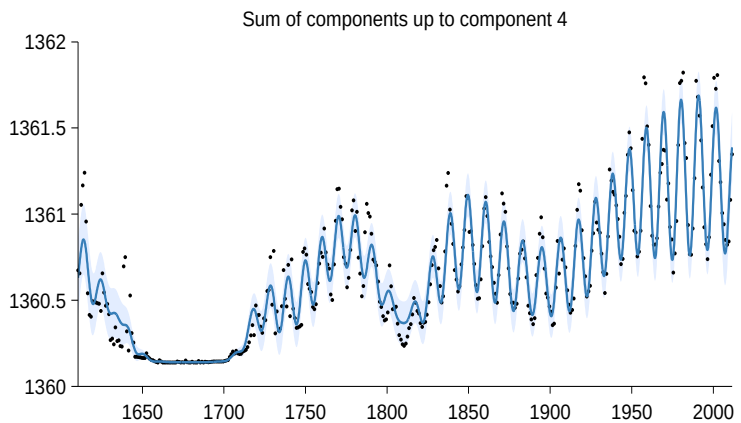




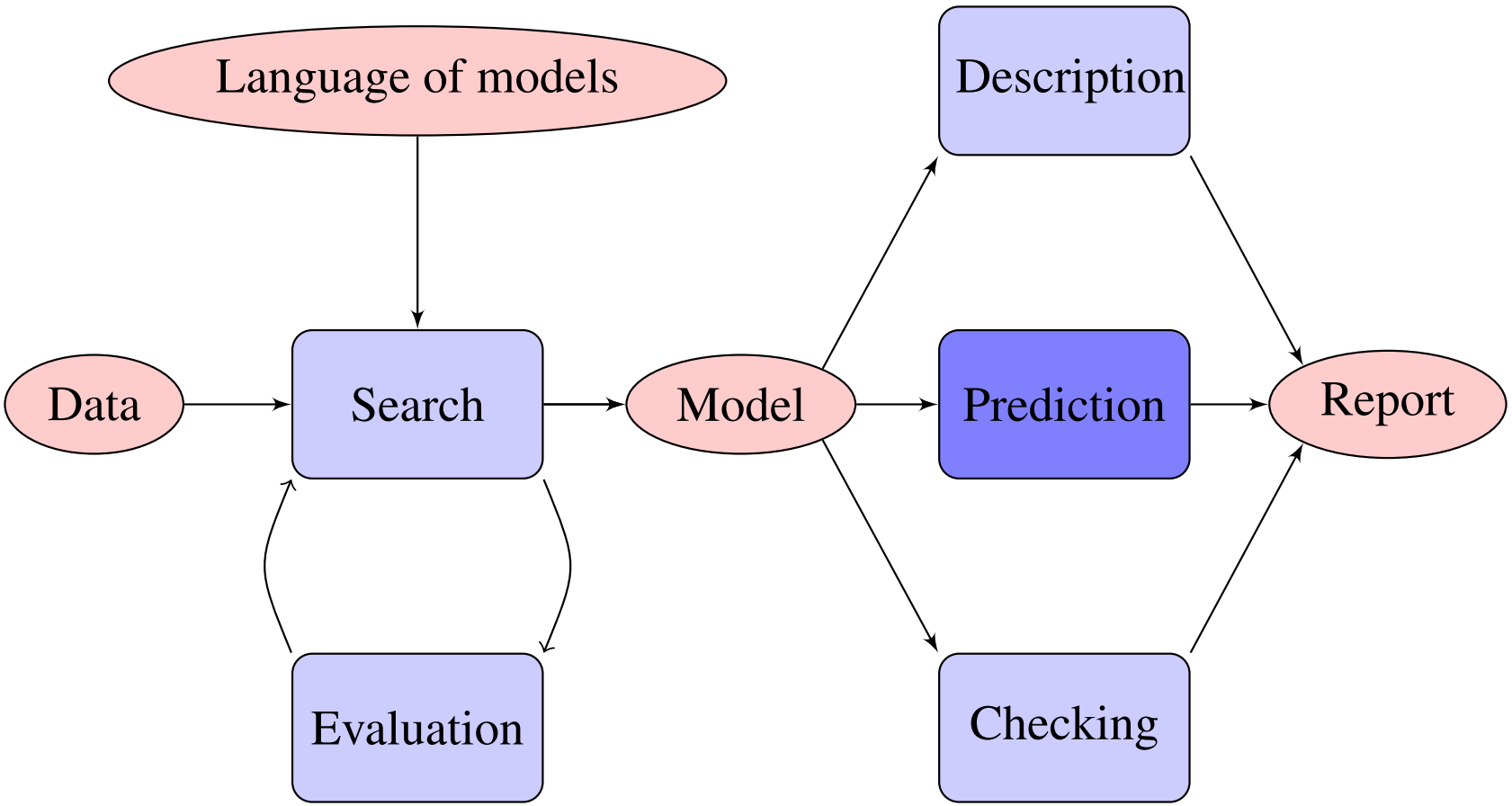
Example: Solar irradiance

This component is approximately periodic with a period of 10.8 years. Across periods the shape of this function varies smoothly with a typical lengthscale of 36.9 years. The shape of this function within each period is very smooth and resembles a sinusoid. This component applies until 1643 and from 1716 onwards.



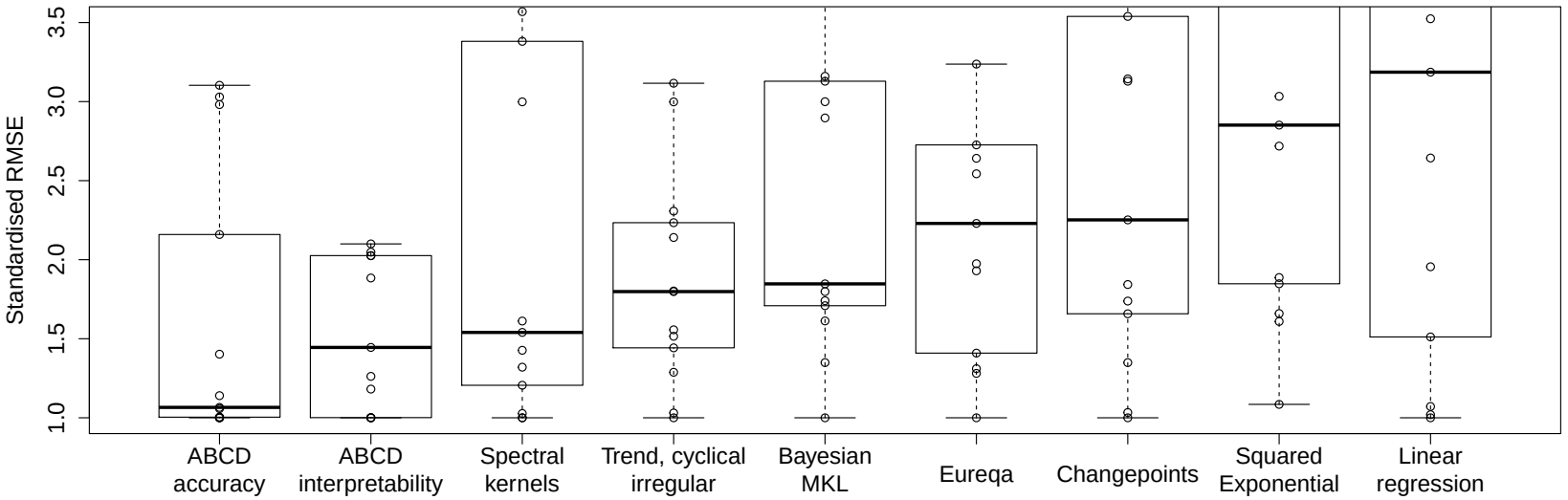


PREDICTIVE PERFORMANCE



GOOD PREDICTIVE PERFORMANCE AS WELL

Standardised RMSE over 13 data sets



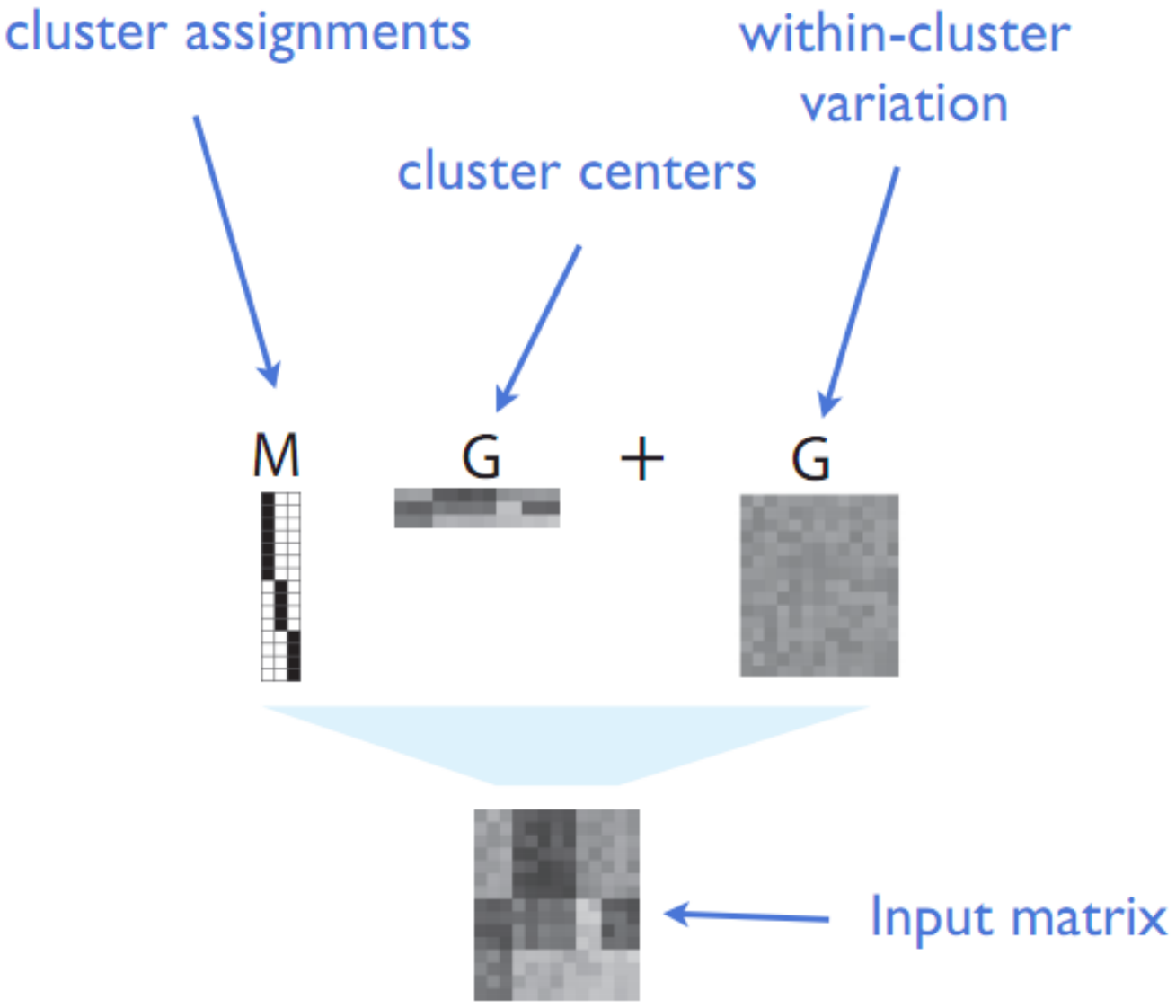
- This method is slow but contains most common methods as a special case
- many extensions possible

GRAMMARS FOR MATRIX DECOMPOSITIONS

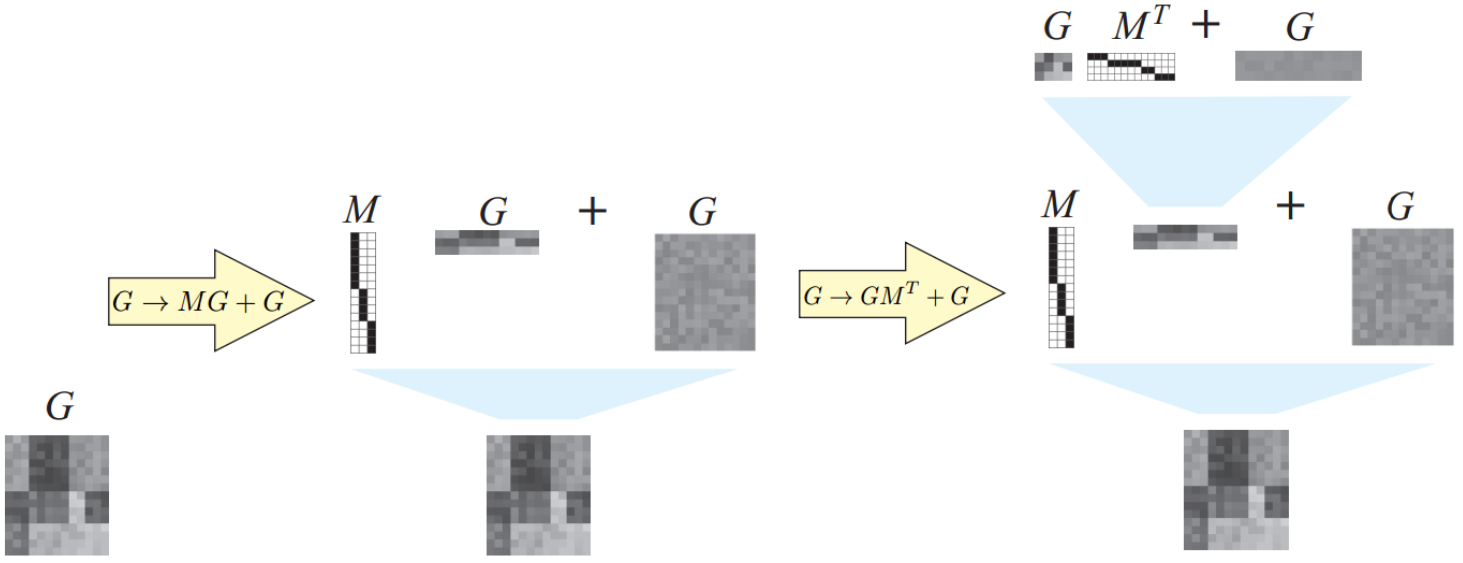
Previous work introduced idea of grammar of compositions:

- Exploiting compositionality to explore a large space of model structures [UAI 2012]
  Roger B. Grosse, Ruslan Salakhutdinov,
  William T. Freeman, Joshua B. Tenenbaum
- Slides that follow are from Roger Grosse

MATRIX DECOMPOSITION



RECURSIVE MATRIX DECOMPOSITION



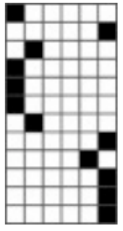
- Main idea: Matrices can be recursively decomposed
- Example: Co-clustering by clustering cluster assignments.

BUILDING BLOCKS

Gaussian
(G)

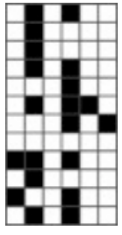
$$\lambda_i \sim \text{Gamma}(a, b)$$
$$\nu_j \sim \text{Gamma}(a, b)$$
$$u_{ij} \sim \text{Normal}(0, \lambda_i^{-1}\nu_j^{-1})*$$

* variance parameters shared between input rows/columns



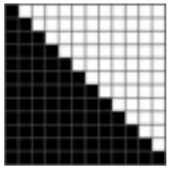
Multinomial
(M)

$$\pi \sim \text{Dirichlet}(\alpha)$$
$$u_i \sim \text{Multinomial}(\pi)$$



Bernoulli
(B)

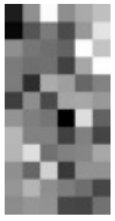
$$p_j \sim \text{Beta}(\alpha, \beta)$$
$$u_{ij} \sim \text{Bernoulli}(p_j)$$



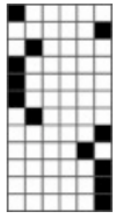
Integration
(C)

$$u_{ij} = \begin{cases} 1 & \text{if } i \geq j \\ 0 & \text{otherwise} \end{cases}$$

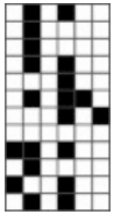
Matrix decomposition: Grammar



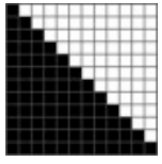
Gaussian
(G)



Multinomial
(M)



Bernoulli
(B)



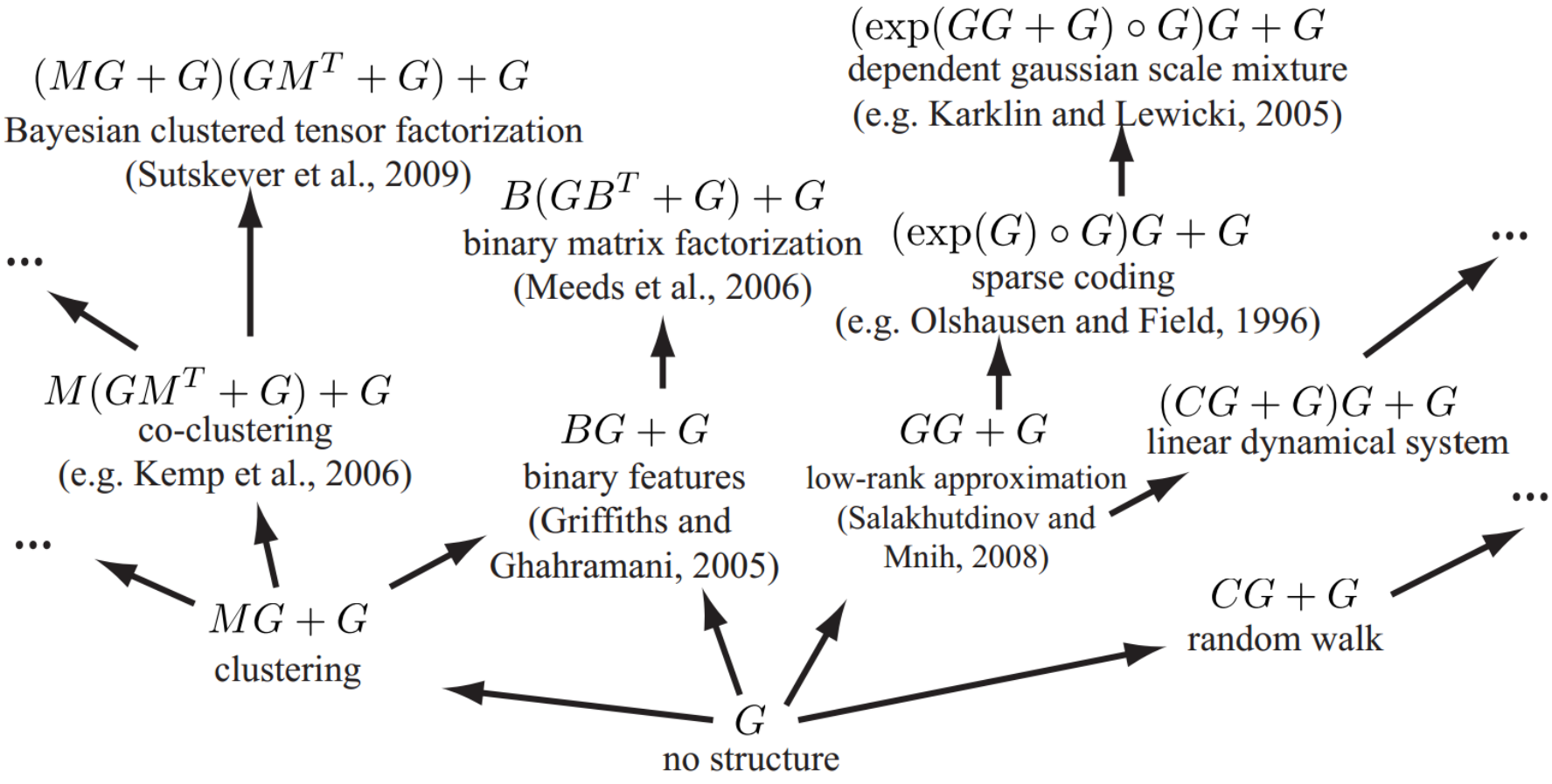
Integration
(C)

Starting symbol: G

Production rules:

clustering	$G \to MG + G \mid GM^T + G$
	$M \to MG + G$
low rank	$G \to GG + G$
binary features	$G \to BG + G \mid GB^T + G$
	$B \to BG + G$
	$M \to B$
linear dynamics	$G \to CG + G \mid GC^T + G$
sparsity	$G \to \exp(G) \circ G$

MATRIX DECOMPOSITION: SPECIAL CASES



Evolution of Image Models


Bossomaier and Snyder, 1987

Modeling images as linear combinations of uncorrelated basis functions gives a Fourier representation.

Modeling the sparse distribution of the linear reconstruction coefficients gives oriented edges.


Olshausen and Field, 1996


Karklin and Lewicki, 2005

Modeling the dependencies in the sparsity pattern gives a high-level texture model.

Application to Natural Image Patches

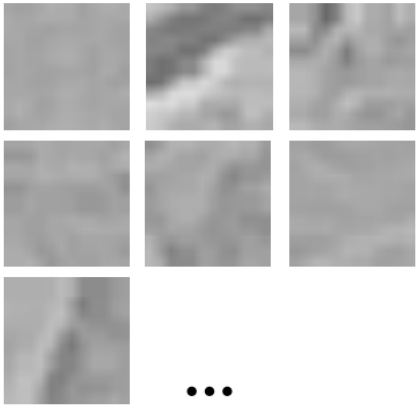
Image patches $GG + G$ $(\exp(G) \circ G)G + G$ $(\exp(GG + G) \circ G)G + G$

Data: 1,000 12x12 patches from 10 blurred and whitened images.

Model 1: Low-rank approximation (PCA).

Model 2: Sparsify coefficients to get sparse coding

Model 3: Model dependencies between scale variables



SUMMARY

How could an AI do statistics?

- Grammars over composite structures are a simple way to specify open-ended model classes.
- Composite structures sometimes give interpretable decompositions.
- Searching over these model classes is a step towards automating statistical analysis.

SUMMARY

How could an AI do statistics?

- Grammars over composite structures are a simple way to specify open-ended model classes.
- Composite structures sometimes give interpretable decompositions.
- Searching over these model classes is a step towards automating statistical analysis.

Thanks!