How to build an automatic statistician

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

Joshua Tenenbaum[2], Zoubin Ghahramani[1]

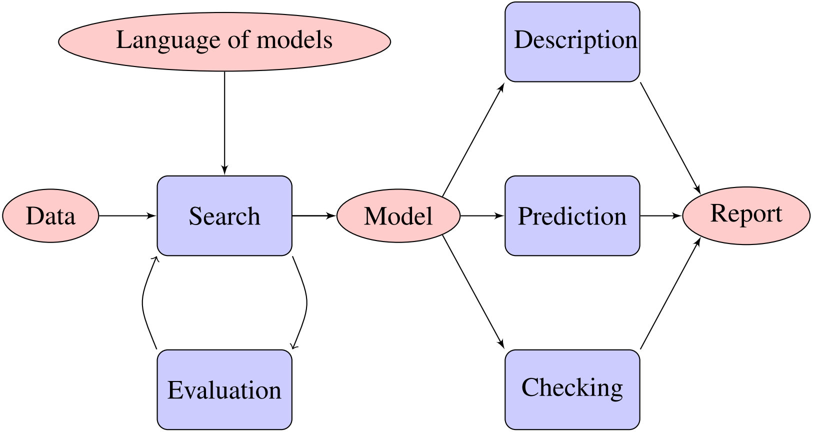
1: Department of Engineering, University of Cambridge, UK
2: Massachusetts Institute of Technology, USA

November 19, 2014

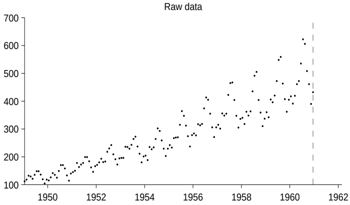
There is a growing need for data analysis

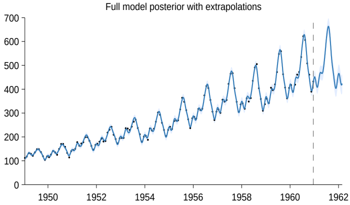
- We live in an era of abundant data
- The McKinsey Global Institute claim
  - *"The United States alone faces a shortage of 140,000 to 190,000 people with analytical expertise and 1.5 million managers and analysts with the skills to understand and make decisions based on the analysis of big data."*
- Diverse fields increasingly relying on expert statisticians, machine learning researchers and data scientists e.g.
  - Computational sciences (e.g. biology, astronomy, . . .)
  - Online advertising
  - Quantitative finance
  - . . .

What would an automatic statistician do?



PREVIEW: AN ENTIRELY AUTOMATIC ANALYSIS





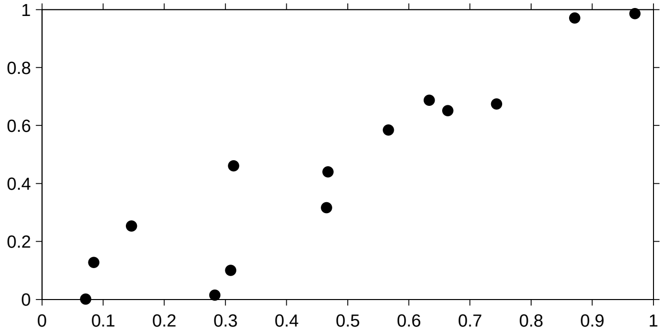
Four additive components have been identified in the data

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

AGENDA

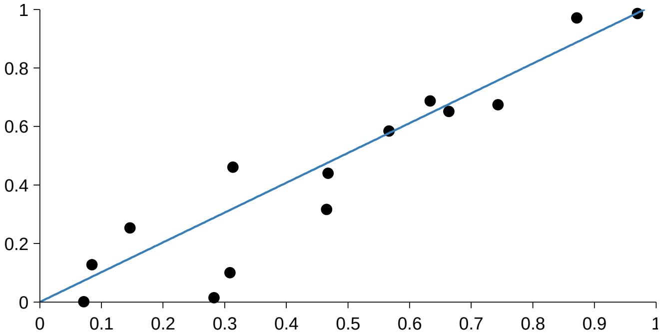
- Introduction to Bayesian statistics
- Introduction to Gaussian processes via linear regression
- Description of automatic statistician
- Examples of automatically generated ouput

How to do linear regression



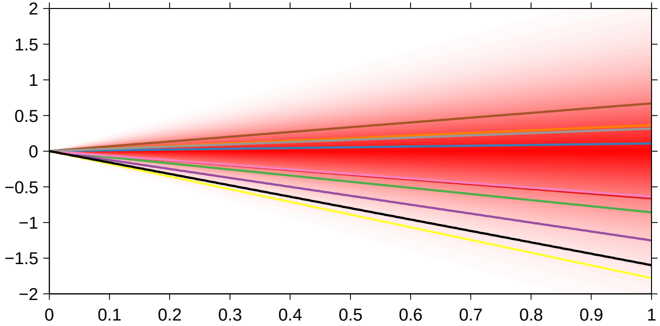
- Linear regression starts by assuming a model of the form $y_i = mx_i + \varepsilon_i$ where x and y are inputs and outputs respectively and ε are errors or noise

How to do linear regression



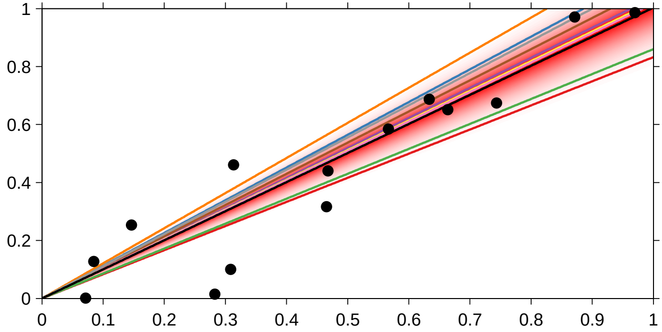
- Common approach: Estimate m by least squares

How to do Bayesian linear regression



- Bayesian approach:
  - Specify beliefs about *m* using probability distributions

How to do Bayesian linear regression



- Bayesian approach:
  - Specify prior beliefs about m using probability distributions
  - Follow rules of probability to update beliefs rationally after observing data

- Suppose we wish to fit a model M with parameters θ to data D
  - M defined by $p(D \mid \theta, M)$ - the likelihood

- Suppose we wish to fit a model M with parameters θ to data D
  - M defined by $p(D \mid \theta, M)$ - the likelihood
- We represent our uncertainty about θ with a probability distribution
  - $p(\theta \mid M)$ - the prior

- Suppose we wish to fit a model M with parameters θ to data D
  - M defined by $p(D \mid \theta, M)$ - the likelihood
- We represent our uncertainty about θ with a probability distribution
  - $p(\theta \mid M)$ - the prior
- We now derive $p(\theta \mid D, M)$ - the posterior

BAYES RULE

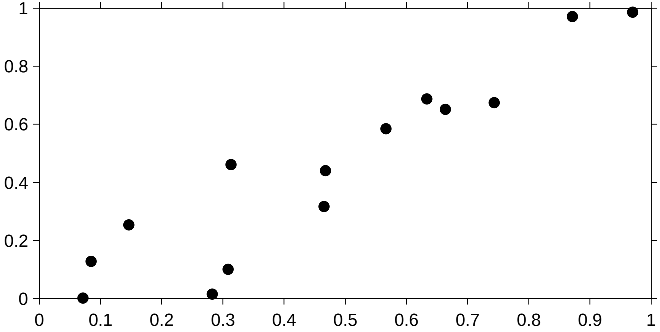
- Suppose we wish to fit a model M with parameters θ to data D
  - M defined by $p(D \mid \theta, M)$ - the likelihood

- We represent our uncertainty about θ with a probability distribution
  - $p(\theta \mid M)$ - the prior

- We then derive $p(\theta \mid D, M)$ - the posterior

$$\underbrace{p(\theta \mid D, M)}_{\text{posterior}} = \frac{\overbrace{p(D \mid \theta, M)}^{\text{likelihood}} \overbrace{p(\theta \mid M)}^{\text{prior}}}{\underbrace{\int p(D \mid \theta, M) p(\theta \mid M) \mathrm{d}\theta}_{\text{marginal likelihood}}}$$

BAYESIAN LINEAR REGRESSION



- Linear regression starts by assuming a model of the form $y_i = mx_i + \varepsilon_i$ where x and y are inputs and outputs respectively and ε are errors or noise

BAYESIAN LINEAR REGRESSION

- Linear regression starts by assuming a model of the form $y_i = mx_i + \varepsilon_i$ where x and y are inputs and outputs respectively and ε are errors or noise

- Represent prior beliefs about m using a probability distribution

Bayesian linear regression

- Linear regression starts by assuming a model of the form $y_i = mx_i + \varepsilon_i$ where x and y are inputs and outputs respectively and ε are errors or noise
- Represent prior beliefs about m using a probability distribution
  - e.g. $m \sim \text{Normal}(\text{mean} = 0, \text{variance} = 1)$

Bayesian linear regression

- Linear regression starts by assuming a model of the form $y_i = mx_i + \varepsilon_i$ where x and y are inputs and outputs respectively and ε are errors or noise
- Represent prior beliefs about m using a probability distribution
  - e.g. $m \sim \mathcal{N}(0, 1)$

BAYESIAN LINEAR REGRESSION

- Linear regression starts by assuming a model of the form $y_i = mx_i + \varepsilon_i$ where x and y are inputs and outputs respectively and ε are errors or noise

- Represent prior beliefs about m using a probability distribution
  - e.g. $m \sim \mathcal{N}(0, 1)$

- Define likelihood by representing prior beliefs about ε using a probability distribution
  - e.g. $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ independently for all i

Bayesian linear regression

- Linear regression starts by assuming a model of the form $y_i = mx_i + \varepsilon_i$ where x and y are inputs and outputs respectively and ε are errors or noise
- Represent prior beliefs about m using a probability distribution
  - e.g. $m \sim \mathcal{N}(0, 1)$
- Define likelihood by representing prior beliefs about ε using a probability distribution
  - e.g. $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ independently for all i
- Apply Bayes rule

Bayes rule applied to linear regression

$$p(m \mid y, x) \quad = \quad \frac{p(y \mid m, x)\, p(m)}{\int p(y \mid m, x) p(m)\, \mathrm{d}m}$$

Bayes rule applied to linear regression

$$
\begin{aligned}
p(m \mid y, x) &= \frac{p(y \mid m, x)\, p(m)}{\int p(y \mid m, x) p(m)\, \mathrm{d}m} \\
&\propto p(y \mid m, x)\, p(m)
\end{aligned}
$$

Bayes rule applied to linear regression

$$
\begin{aligned}
p(m \,|\, y, x) &= \frac{p(y \,|\, m, x)\, p(m)}{\int p(y \,|\, m, x) p(m)\, \mathrm{d}m} \\
&\propto p(y \,|\, m, x)\, p(m) \\
&\propto \left(\prod_i \frac{1}{2\pi\sigma_\varepsilon} e^{-(y_i - m x_i)^2/(2\sigma_\varepsilon^2)} \right) \frac{1}{2\pi} e^{-m^2/2}
\end{aligned}
$$

Bayes rule applied to linear regression

$$
\begin{aligned}
p(m \,|\, y, x) &= \frac{p(y \,|\, m, x)\, p(m)}{\int p(y \,|\, m, x) p(m)\, \mathrm{d}m} \\
&\propto p(y \,|\, m, x)\, p(m) \\
&\propto \left(\prod_i \frac{1}{2\pi\sigma_\varepsilon} e^{-(y_i - m x_i)^2/(2\sigma_\varepsilon^2)} \right) \frac{1}{2\pi} e^{-m^2/2} \\
&\propto e^{-\sum_i (y_i - m x_i)^2/(2\sigma_\varepsilon^2) - m^2/2}
\end{aligned}
$$

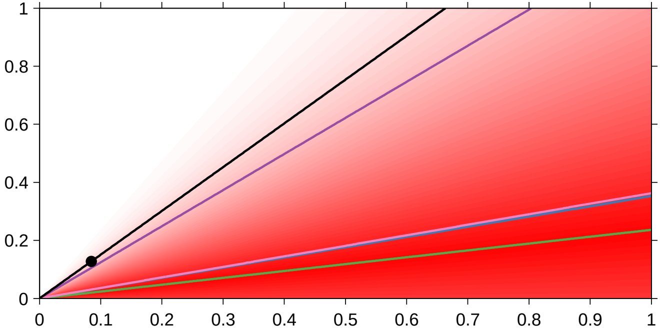
Bayes rule applied to linear regression

$$
\begin{aligned}
p(m \mid y, x) &= \frac{p(y \mid m, x)\, p(m)}{\int p(y \mid m, x) p(m)\, \mathrm{d}m} \\
&\propto p(y \mid m, x)\, p(m) \\
&\propto \left(\prod_i \frac{1}{2\pi\sigma_\varepsilon} e^{-(y_i - m x_i)^2/(2\sigma_\varepsilon^2)} \right) \frac{1}{2\pi} e^{-m^2/2} \\
&\propto e^{-\sum_i (y_i - m x_i)^2/(2\sigma_\varepsilon^2) - m^2/2} \\
&\propto e^{-\frac{\sum_i x_i^2 + \sigma_\varepsilon^2}{2\sigma_\varepsilon^2} \left(m - \frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2} \right)^2}
\end{aligned}
$$

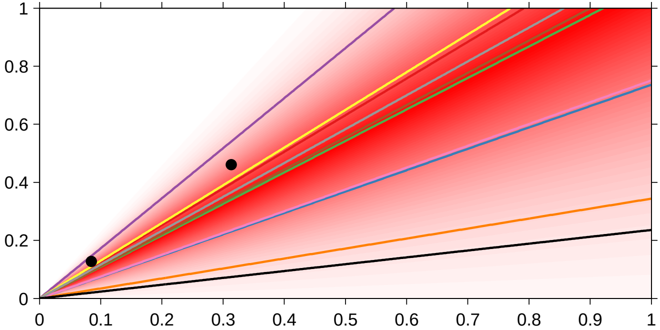
Bayes rule applied to linear regression

$$\begin{aligned}
p(m \mid y, x) &= \frac{p(y \mid m, x)\, p(m)}{\int p(y \mid m, x) p(m)\, \mathrm{d}m} \\
&\propto p(y \mid m, x)\, p(m) \\
&\propto \left(\prod_i \frac{1}{2\pi\sigma_\varepsilon} e^{-(y_i - m x_i)^2/(2\sigma_\varepsilon^2)} \right) \frac{1}{2\pi} e^{-m^2/2} \\
&\propto e^{-\sum_i (y_i - m x_i)^2/(2\sigma_\varepsilon^2) - m^2/2} \\
&\propto e^{-\frac{\sum_i x_i^2 + \sigma_\varepsilon^2}{2\sigma_\varepsilon^2}\left(m - \frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2}\right)^2}
\end{aligned}$$

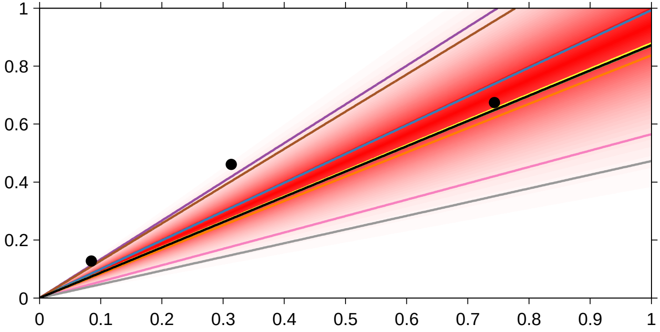
This is a normal distribution: $m \mid y, x \sim \mathcal{N}\left(\frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2}, \frac{\sigma_\varepsilon^2}{\sum_i x_i^2 + \sigma_\varepsilon^2}\right)$



$$m \mid y, x \sim \mathcal{N}\left(\frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2}, \frac{\sigma_\varepsilon^2}{\sum_i x_i^2 + \sigma_\varepsilon^2}\right)$$

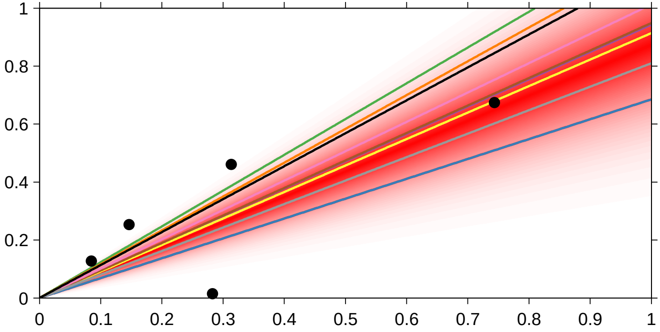


$$m \mid y, x \sim \mathcal{N}\left(\frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2}, \frac{\sigma_\varepsilon^2}{\sum_i x_i^2 + \sigma_\varepsilon^2}\right)$$



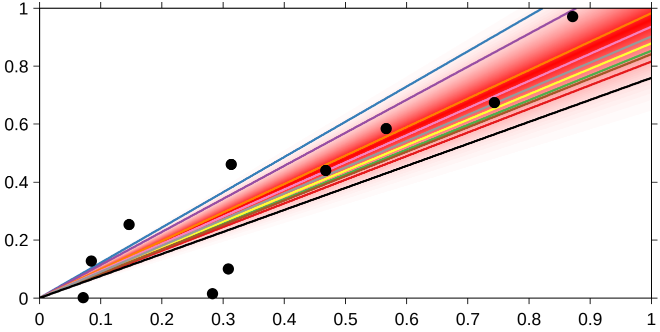
$$m \mid y, x \sim \mathcal{N}\left(\frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2}, \frac{\sigma_\varepsilon^2}{\sum_i x_i^2 + \sigma_\varepsilon^2}\right)$$

Bayesian linear regression

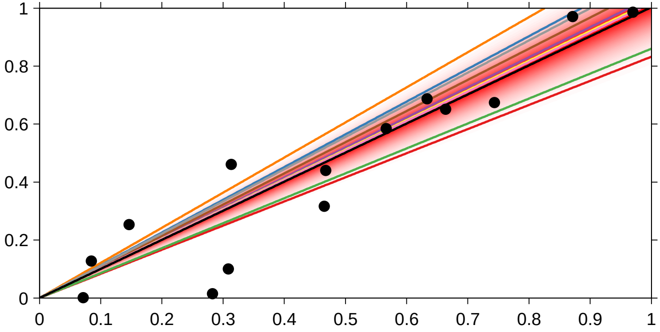


$$m \mid y, x \sim \mathcal{N}\left(\frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2}, \frac{\sigma_\varepsilon^2}{\sum_i x_i^2 + \sigma_\varepsilon^2}\right)$$

Bayesian linear regression



$$m \mid y, x \sim \mathcal{N}\left(\frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2}, \frac{\sigma_\varepsilon^2}{\sum_i x_i^2 + \sigma_\varepsilon^2}\right)$$



$$m \mid y, x \sim \mathcal{N}\left(\frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2}, \frac{\sigma_\varepsilon^2}{\sum_i x_i^2 + \sigma_\varepsilon^2}\right)$$

Bayesian linear regression rewritten

- We defined a Bayesian linear regression model by specifying priors on m and ε_i

Bayesian linear regression rewritten

- We defined a Bayesian linear regression model by specifying priors on m and ε_i
  - $m \sim \mathcal{N}(0, 1)$, $\varepsilon_i \sim_{\text{iid}} \mathcal{N}(0, \sigma_\varepsilon^2)$

Bayesian linear regression rewritten

- We defined a Bayesian linear regression model by specifying priors on m and ε_i
  - $m \sim \mathcal{N}(0, 1)$, $\varepsilon_i \sim_{\text{iid}} \mathcal{N}(0, \sigma_\varepsilon^2)$
  - $y_i \,|\, m, \varepsilon_i = mx_i + \varepsilon_i$

Bayesian linear regression rewritten

- We defined a Bayesian linear regression model by specifying priors on m and ε_i
  - $m \sim \mathcal{N}(0, 1)$, $\varepsilon_i \sim_{\text{iid}} \mathcal{N}(0, \sigma_\varepsilon^2)$
  - $y_i \,|\, m, \varepsilon_i = m x_i + \varepsilon_i$

- This implictly defined a joint prior on $\{y_i \,:\, i = 1, \ldots, n\}$

BAYESIAN LINEAR REGRESSION REWRITTEN

- We defined a Bayesian linear regression model by specifying priors on m and ε_i
  - $m \sim \mathcal{N}(0, 1)$, $\varepsilon_i \sim_{\text{iid}} \mathcal{N}(0, \sigma_\varepsilon^2)$
  - $y_i \,|\, m, \varepsilon_i = mx_i + \varepsilon_i$
- This implictly defined a joint prior on $\{y_i \,:\, i = 1, \ldots, n\}$
  - $y_i \sim \mathcal{N}(0, x_i^2 + \sigma_\varepsilon^2)$ (sum of two normals)

Bayesian linear regression rewritten

- We defined a Bayesian linear regression model by specifying priors on m and ε_i
  - $m \sim \mathcal{N}(0, 1)$, $\varepsilon_i \sim_{\text{iid}} \mathcal{N}(0, \sigma_\varepsilon^2)$
  - $y_i \,|\, m, \varepsilon_i = m x_i + \varepsilon_i$
- This implictly defined a joint prior on $\{y_i \,:\, i = 1, \ldots, n\}$
  - $y_i \sim \mathcal{N}(0, x_i^2 + \sigma_\varepsilon^2)$ (sum of two normals)
  - $\operatorname{cov}(y_i, y_j) = x_i x_j \quad \forall\, i \neq j$

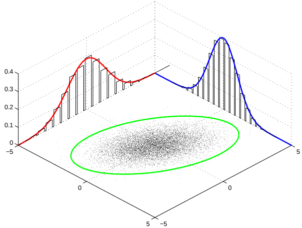
Bayesian linear regression rewritten

- We defined a Bayesian linear regression model by specifying priors on m and ε_i
  - $m \sim \mathcal{N}(0, 1)$, $\varepsilon_i \sim_{\text{iid}} \mathcal{N}(0, \sigma_\varepsilon^2)$
  - $y_i \,|\, m, \varepsilon_i = mx_i + \varepsilon_i$

- This implictly defined a joint prior on $\{y_i \,:\, i = 1, \ldots, n\}$
  - $y_i \sim \mathcal{N}(0, x_i^2 + \sigma_\varepsilon^2)$ (sum of two normals)
  - $\text{cov}(y_i, y_j) = x_i x_j \quad \forall\, i \neq j$

$\{y_i \,:\, i = 1, \ldots, n\}$ has a multivariate normal distribution

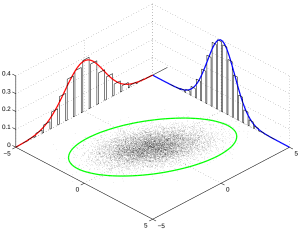
- $y \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$
- where $k_{ij} = x_i x_j + \delta_{ij}\sigma_\varepsilon^2$



BAYESIAN LINEAR REGRESSION REWRITTEN

$\{y_i \ : \ i = 1, \ldots, n\}$ has a multivariate normal distribution

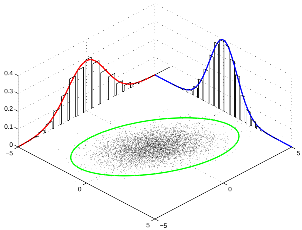
- $y \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$
- where $k_{ij} = x_i x_j + \delta_{ij}\sigma_\varepsilon^2$



BAYESIAN LINEAR REGRESSION REWRITTEN

$\{y_i \ : \ i = 1, \ldots, n\}$ has a multivariate normal distribution

- $y \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$
- where $k_{ij} = x_i x_j + \delta_{ij}\sigma_\varepsilon^2$



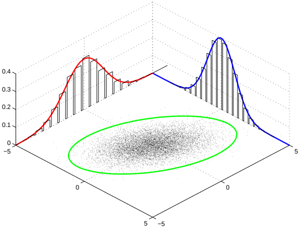
The covariance can be written as a function of pairs of x_i

$$k_{ij} = k(x_i, x_j) = x_i x_j + \delta_{x_i = x_j}\sigma_\varepsilon^2$$

Bayesian linear regression rewritten

$\{y_i \,:\, i = 1, \ldots, n\}$ has a multivariate normal distribution

- $y \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$
- where $k_{ij} = x_i x_j + \delta_{ij}\sigma_\varepsilon^2$



The covariance can be written as a function of pairs of x_i

$$k_{ij} = k(x_i, x_j) = x_i x_j + \delta_{x_i = x_j}\sigma_\varepsilon^2$$

Do we define a valid probability distribution if $\{x_i\} = \mathbb{R}$?

GAUSSIAN PROCESSES

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

GAUSSIAN PROCESSES

- 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

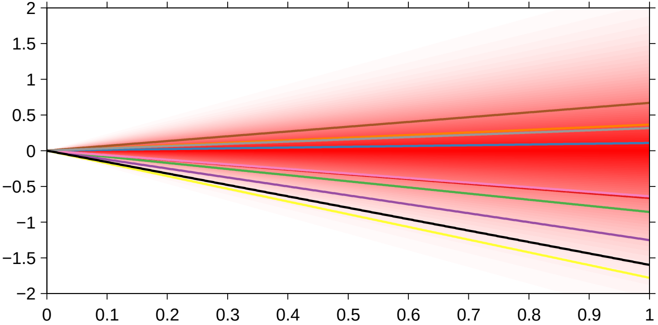
GAUSSIAN PROCESSES

- 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 $f \sim \mathrm{GP}(\mu, k)$

GAUSSIAN PROCESSES

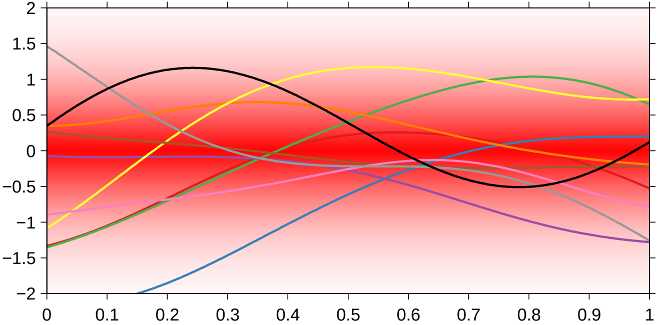
- A Gaussian process is collection of random variables, any finite number of which have a joint Gaussian distribution
- We can write this collection of random variables as $\{f(x) : x \in \mathcal{X}\}$ i.e. a function f evaluated at inputs x
- A GP is completely specified by
  - Mean function, $\mu(x) = \mathbb{E}(f(x))$
  - Covariance / kernel function, $k(x, x') = \mathrm{Cov}(f(x), f(x'))$
  - Denoted $f \sim \mathrm{GP}(\mu, k)$
- Can be thought of as a probability distribution on functions

LINEAR KERNEL = LINEAR FUNCTIONS

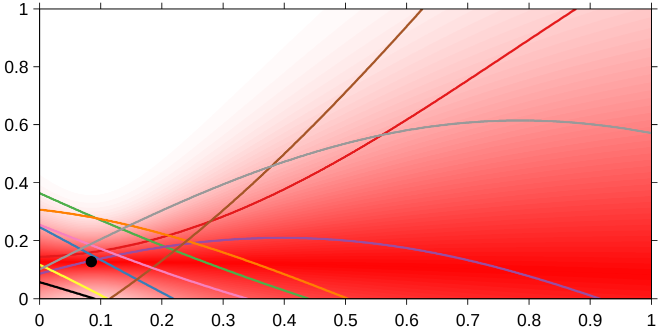


$$k(x, x') = xx'$$

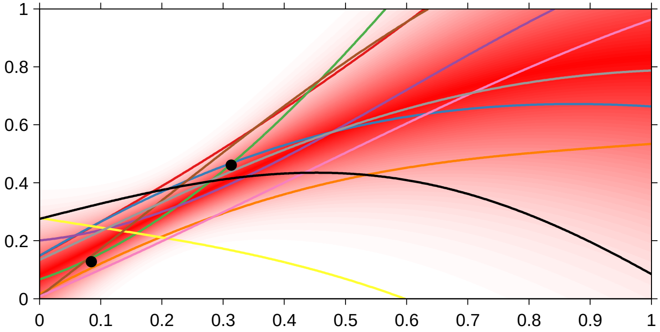
What about other kernels?



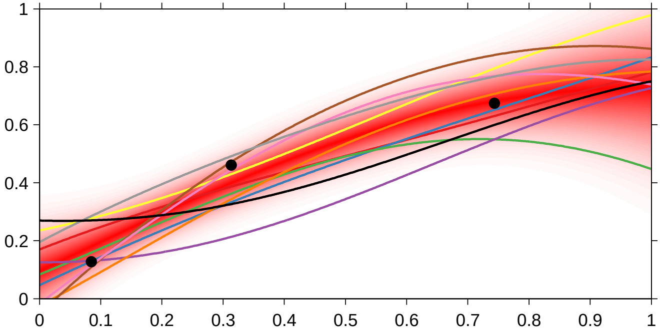
$$k(x, x') = e^{-(x-x')^2}$$


1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

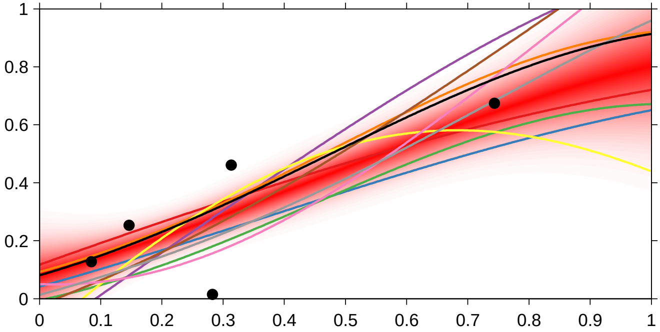
Bayesian non-linear regression



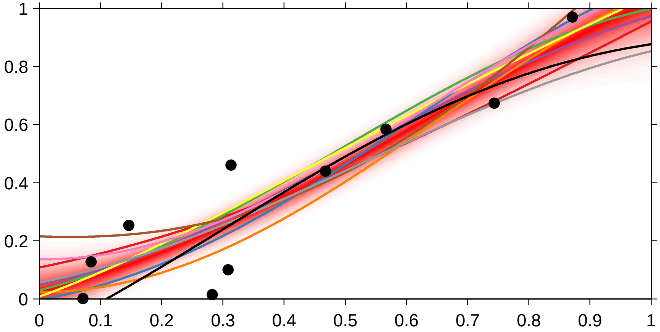
Bayesian non-linear regression

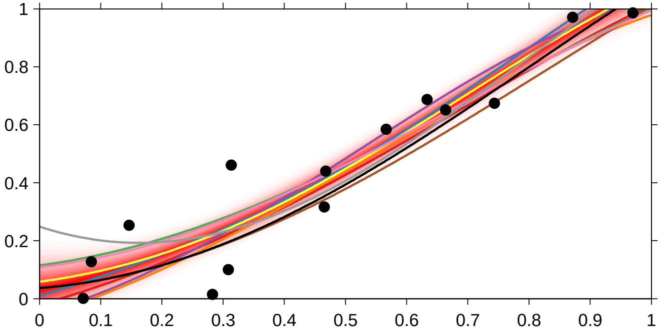


BAYESIAN NON-LINEAR REGRESSION

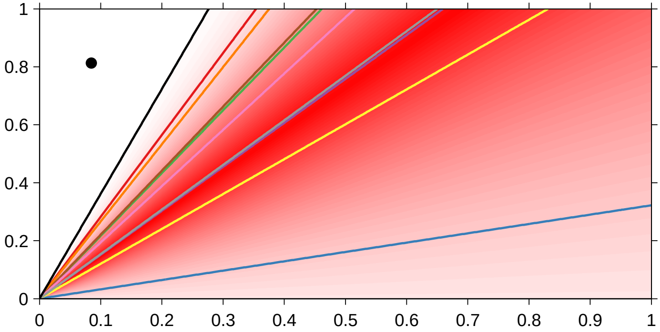


BAYESIAN NON-LINEAR REGRESSION

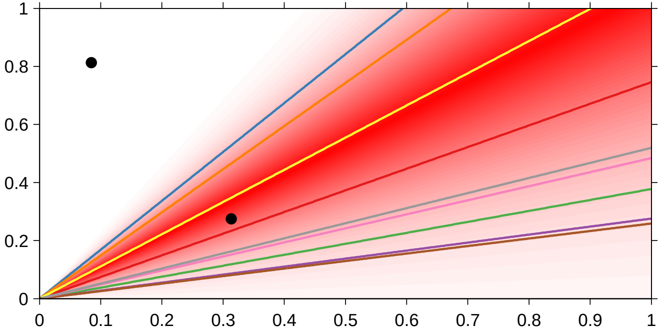



1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

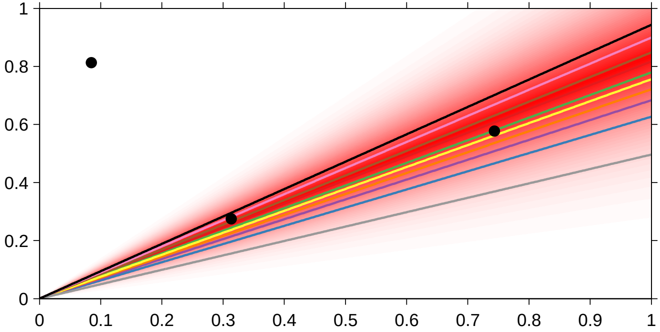
Bayesian linear regression gone wrong



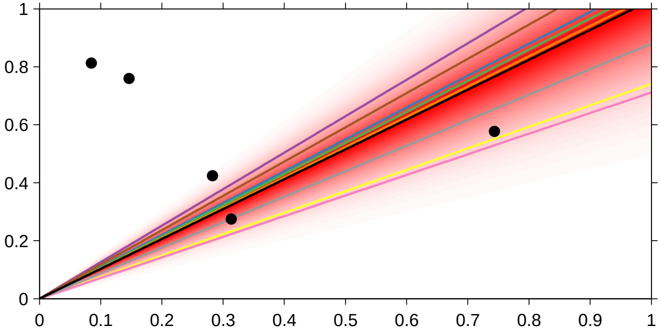
Bayesian linear regression gone wrong

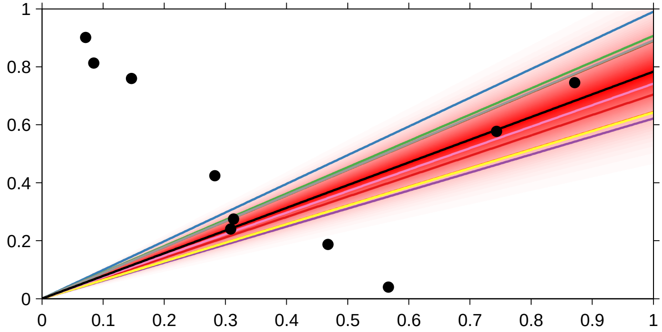


Bayesian linear regression gone wrong

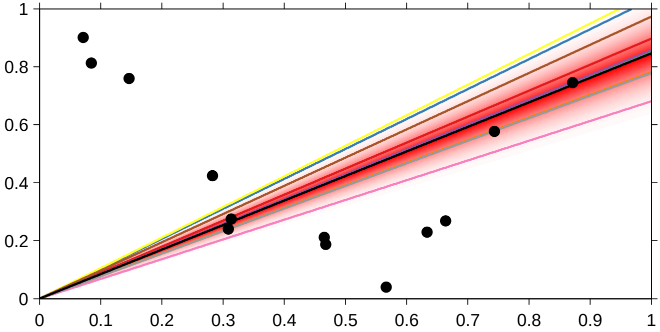


Bayesian linear regression gone wrong

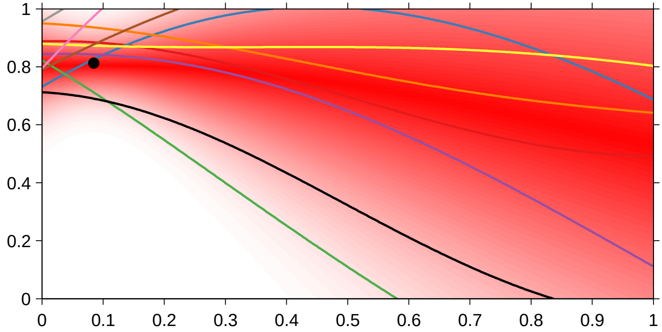



1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

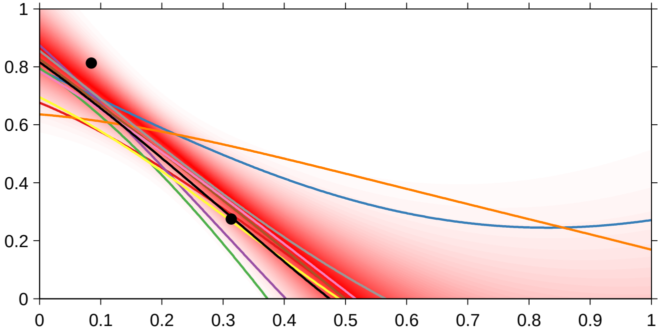
Bayesian linear regression gone wrong



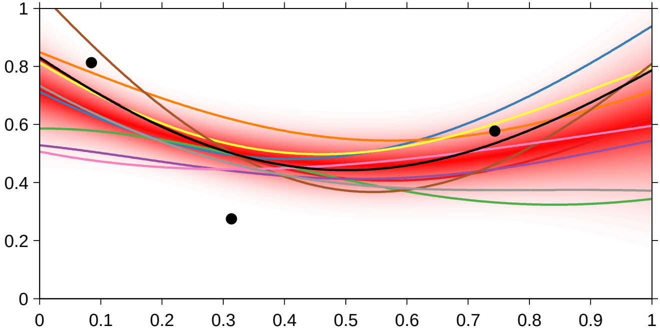
Non-linearity to the rescue



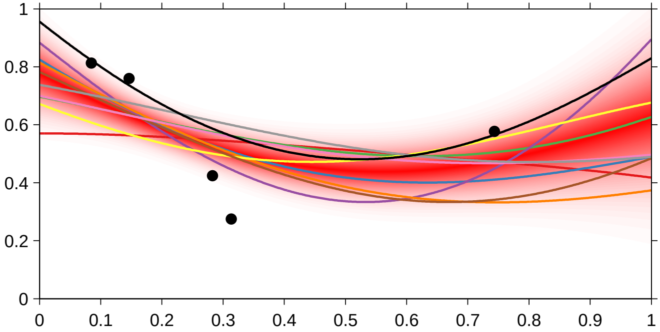
NON-LINEARITY TO THE RESCUE



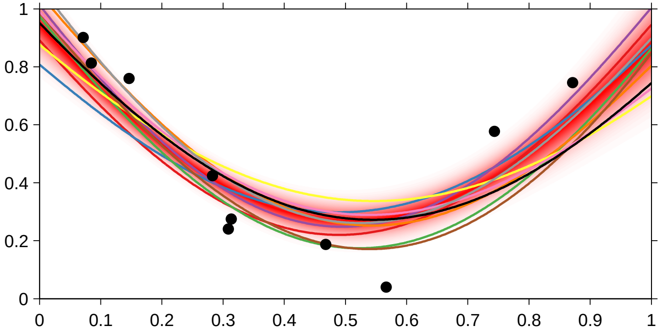
NON-LINEARITY TO THE RESCUE



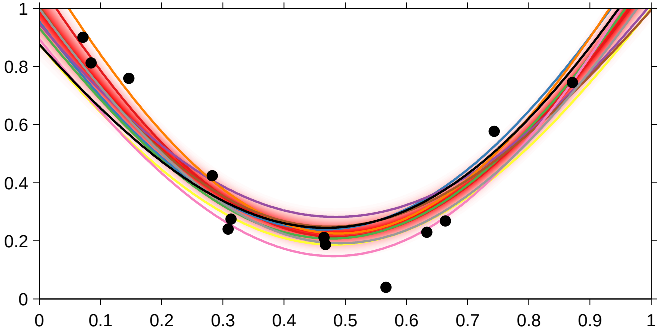
Non-linearity to the rescue



NON-LINEARITY TO THE RESCUE



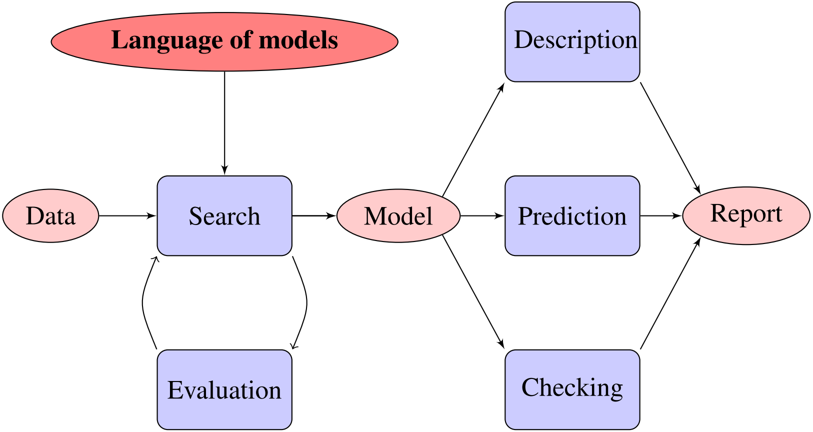
NON-LINEARITY TO THE RESCUE



WHICH KERNEL FUNCTION?

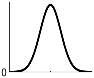
How far can we go with kernel functions?

DEFINING A LANGUAGE OF MODELS

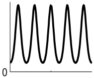


The atoms of our language

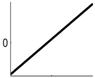
Five base kernels



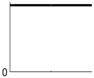
Squared exp. (SE)



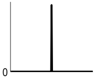
Periodic (Per)



Linear (Lin)

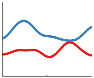


Constant (C)

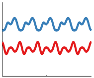


White noise (WN)

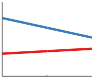
Encoding for the following types of functions



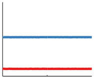
Smooth functions



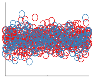
Periodic functions



Linear functions



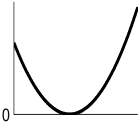
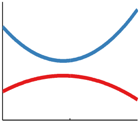
Constant functions



Gaussian noise

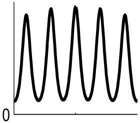
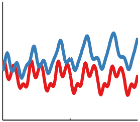
The composition rules of our language

- Two main operations: addition, multiplication

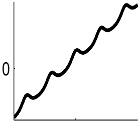
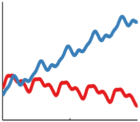
LIN $\times$ LIN

quadratic functions

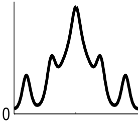
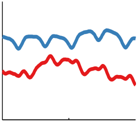
SE $\times$ PER

locally periodic

LIN $+$ PER

periodic plus linear trend

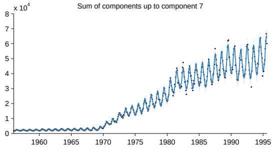



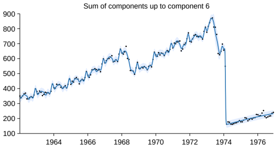
SE $+$ PER

periodic plus smooth trend

MODELING CHANGEPOINTS

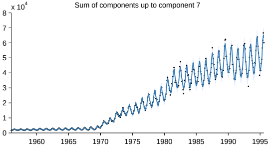
Time series data often exhibit changepoints:

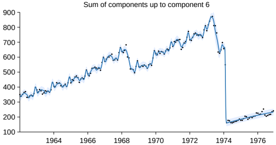




MODELING CHANGEPOINTS

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 $f_1(x) \sim \text{GP}(0, k_1)$ 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.

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

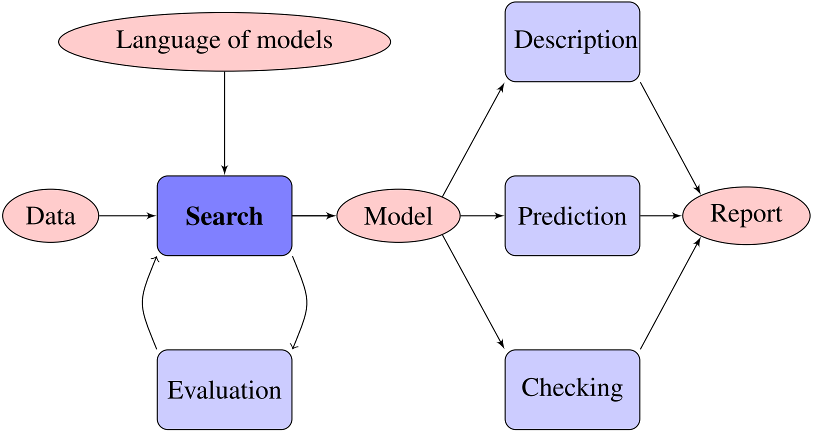
$$k(x, x') = (1 - \sigma(x))\, k_1(x, x')\, (1 - \sigma(x')) + \sigma(x)\, k_2(x, x')\, \sigma(x')$$

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

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

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

DISCOVERING A GOOD MODEL VIA SEARCH



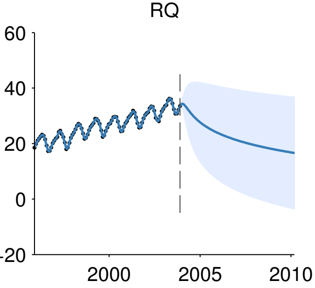
DISCOVERING A GOOD MODEL VIA SEARCH

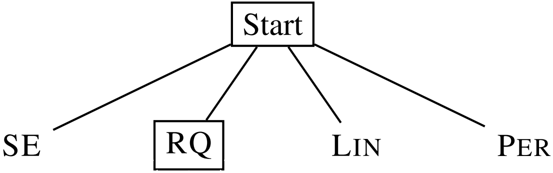
- Language defined as the arbitrary composition of five base kernels (WN, C, LIN, SE, PER) via three operators $(+, \times, \text{CP})$.

- The space spanned by this language is open-ended and can have a high branching factor requiring a judicious search

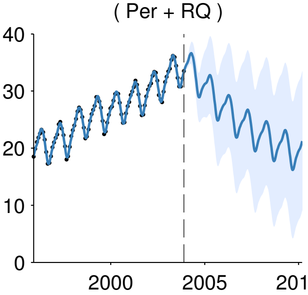
- We propose a greedy search for its simplicity and similarity to human model-building

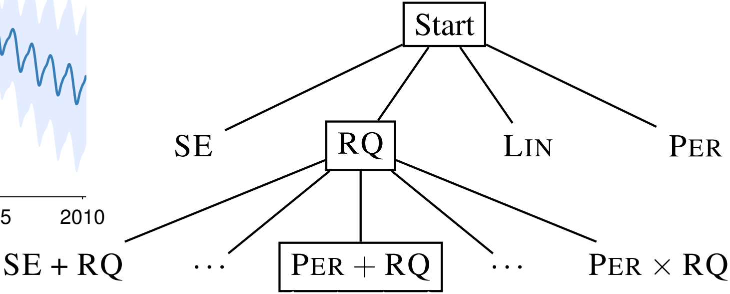
Example: Mauna Loa Keeling Curve



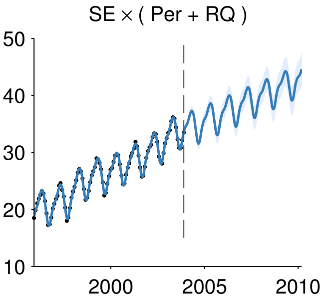


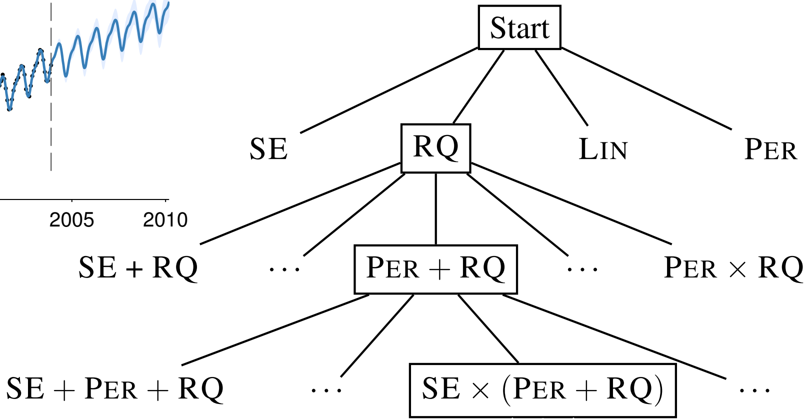
EXAMPLE: MAUNA LOA KEELING CURVE



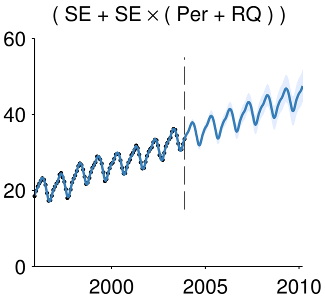


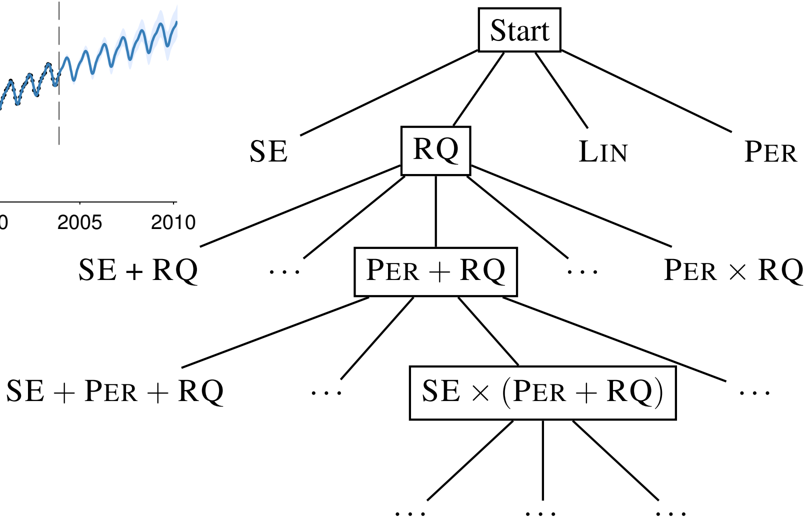
EXAMPLE: MAUNA LOA KEELING CURVE



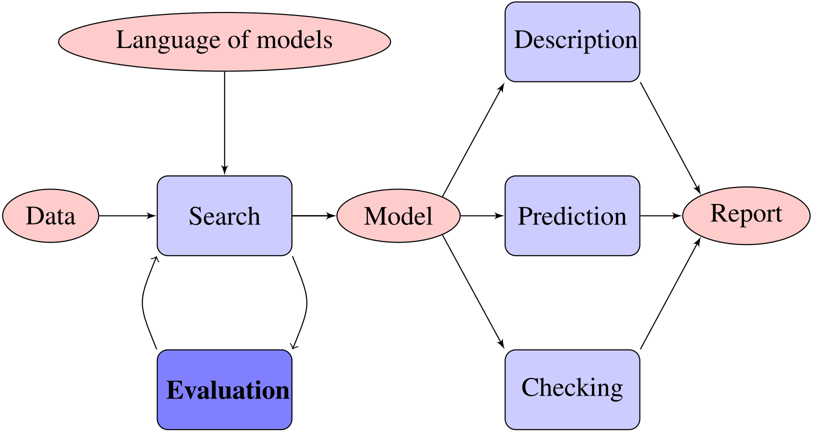


Example: Mauna Loa Keeling Curve





Model evaluation



Bayesian model selection

Suppose we have a collection of models $\{M_i\}$ and some data D

Bayesian model selection

Suppose we have a collection of models $\{M_i\}$ and some data D

Bayes rule tells us

$$p(M_i \mid D) = \frac{p(D \mid M_i)p(M_i)}{p(D)}$$

Bayesian model selection

Suppose we have a collection of models $\{M_i\}$ and some data D

Bayes rule tells us

$$p(M_i \,|\, D) = \frac{p(D \,|\, M_i) p(M_i)}{p(D)}$$

If $p(M_i)$ is equal for all i (prior ignorance) then

$$p(M_i \,|\, D) \propto p(D \,|\, M_i) = \int p(D \,|\, \theta_i, M_i) p(\theta_i \,|\, M_i) \mathrm{d}\theta_i$$

Bayesian model selection

Suppose we have a collection of models $\{M_i\}$ and some data D

Bayes rule tells us

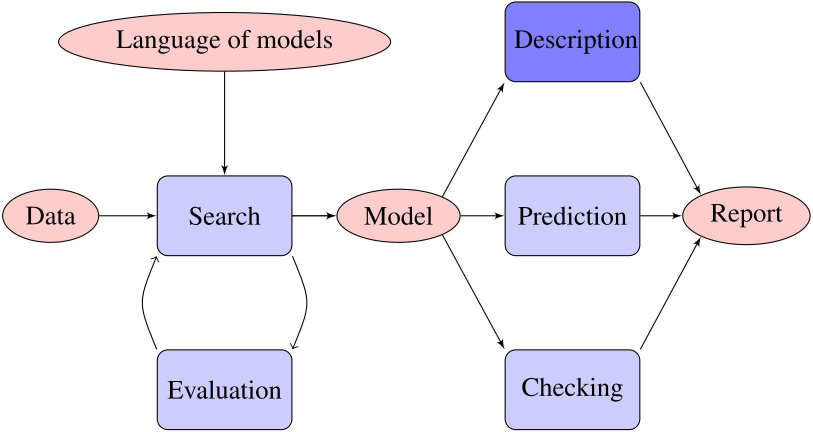
$$p(M_i \,|\, D) = \frac{p(D \,|\, M_i)p(M_i)}{p(D)}$$

If $p(M_i)$ is equal for all i (prior ignorance) then

$$p(M_i \,|\, D) \propto p(D \,|\, M_i) = \int p(D \,|\, \theta_i, M_i)p(\theta_i \,|\, M_i)\mathrm{d}\theta_i$$

i.e. The most likely model has the highest marginal likelihood

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

$$\mathrm{SE} \times (\mathrm{WN} \times \textsc{Lin} + \mathrm{CP}(\mathrm{C}, \textsc{Per}))$$

Sum of products normal form

Suppose the search finds the following kernel

$$\text{SE} \times (\text{WN} \times \textsc{Lin} + \text{CP}(\text{C}, \textsc{Per}))$$

The changepoint can be converted into a sum of products

$$\text{SE} \times (\text{WN} \times \textsc{Lin} + \text{C} \times \boldsymbol{\sigma} + \textsc{Per} \times \bar{\boldsymbol{\sigma}})$$

SUM OF PRODUCTS NORMAL FORM

Suppose the search finds the following kernel

$$\text{SE} \times (\text{WN} \times \textsc{Lin} + \text{CP}(\text{C}, \textsc{Per}))$$

The changepoint can be converted into a sum of products

$$\text{SE} \times (\text{WN} \times \textsc{Lin} + \text{C} \times \boldsymbol{\sigma} + \textsc{Per} \times \bar{\boldsymbol{\sigma}})$$

Multiplication can be distributed over addition

$$\text{SE} \times \text{WN} \times \textsc{Lin} + \text{SE} \times \text{C} \times \boldsymbol{\sigma} + \text{SE} \times \textsc{Per} \times \bar{\boldsymbol{\sigma}}$$

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

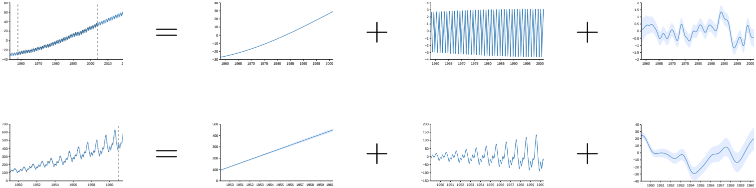
$$\text{WN} \times \text{Lin} + \text{SE} \times \boldsymbol{\sigma} + \text{SE} \times \text{Per} \times \bar{\boldsymbol{\sigma}}$$

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

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

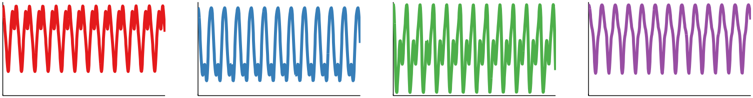
e.g.



We can therefore describe each component separately

$$\underbrace{\text{PER}}_{\text{periodic function}}$$

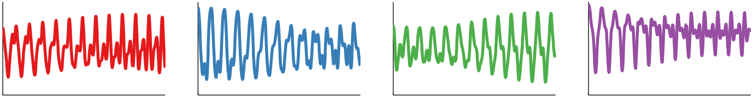
On their own, each kernel is described by a standard noun phrase



PRODUCTS OF KERNELS - SE

$$\underbrace{\text{SE}}_{\text{approximately}} \times \underbrace{\text{PER}}_{\text{periodic function}}$$

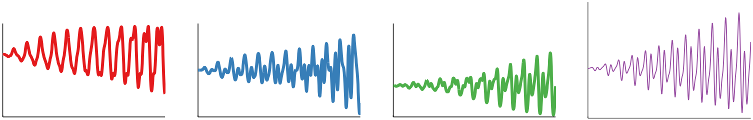
Multiplication by SE removes long range correlations from a model since $\text{SE}(x, x')$ decreases monotonically to 0 as $|x - x'|$ increases.



PRODUCTS OF KERNELS - LIN

$$\underbrace{\text{SE}}_{\text{approximately}} \times \underbrace{\text{PER}}_{\text{periodic function}} \times \underbrace{\text{LIN}}_{\text{with linearly growing amplitude}}$$

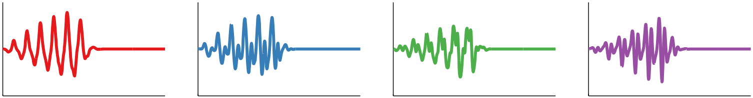
Multiplication by LIN is equivalent to multiplying the function being modeled by a linear function. If $f(x) \sim \mathrm{GP}(0, k)$, then $xf(x) \sim \mathrm{GP}(0, k \times \mathrm{LIN})$. This causes the standard deviation of the model to vary linearly without affecting the correlation.



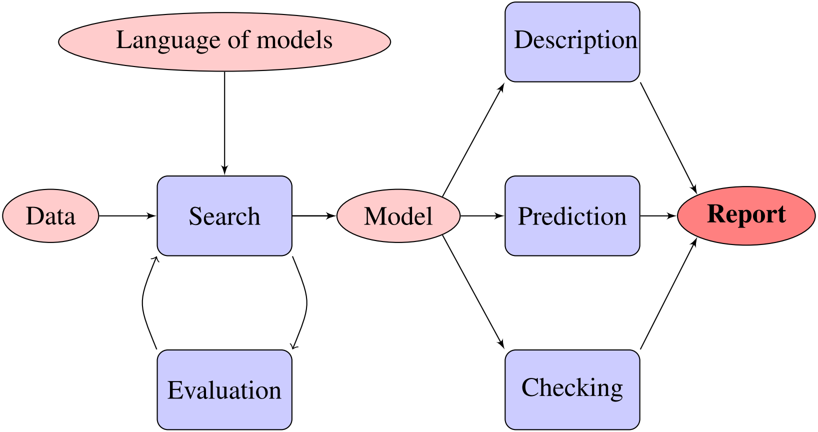
Products of kernels - changepoints

$$\underbrace{\text{SE}}_{\text{approximately}} \times \underbrace{\text{Per}}_{\text{periodic function}} \times \underbrace{\text{Lin}}_{\text{with linearly growing amplitude}} \times \underbrace{\boldsymbol{\sigma}}_{\text{until 1700}}$$

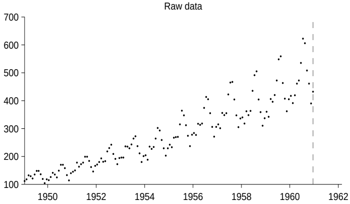
Multiplication by $\boldsymbol{\sigma}$ is equivalent to multiplying the function being modeled by a sigmoid.

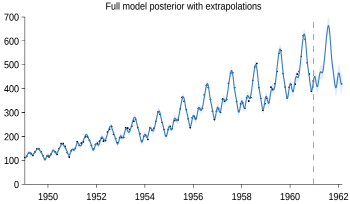


AUTOMATICALLY GENERATED REPORTS



EXAMPLE: AIRLINE PASSENGER VOLUME



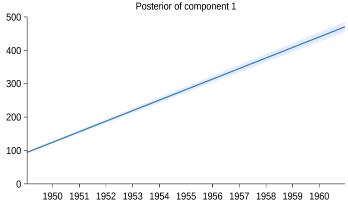


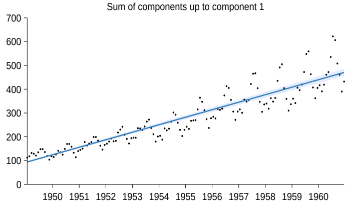
Four additive components have been identified in the data

- 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.

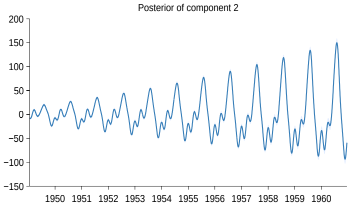
Example: Airline passenger volume

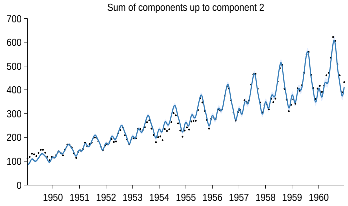
This component is linearly increasing.





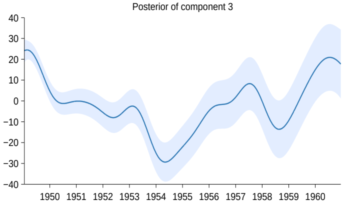
This component is approximately periodic with a period of 1.0 years and varying amplitude. Across periods the shape of this function varies very smoothly. The amplitude of the function increases linearly. The shape of this function within each period has a typical lengthscale of 6.0 weeks.

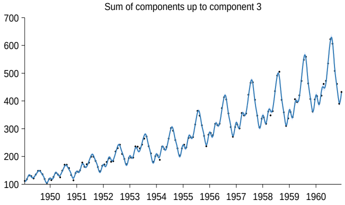




Example: Airline passenger volume

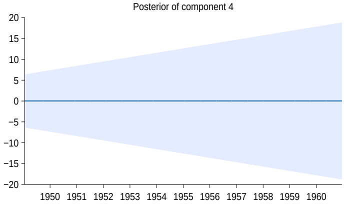
This component is a smooth function with a typical lengthscale of 8.1 months.

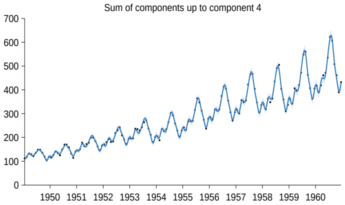




Example: Airline passenger volume

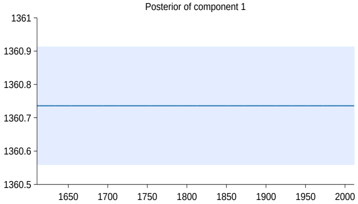
This component models uncorrelated noise. The standard deviation of the noise increases linearly.

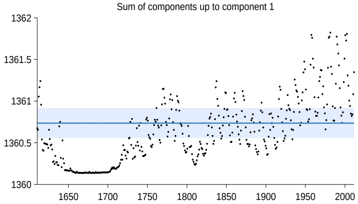




Example: Solar irradiance

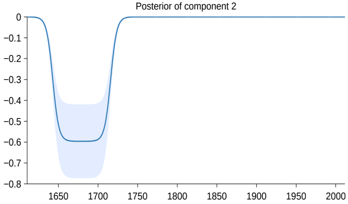
This component is constant.

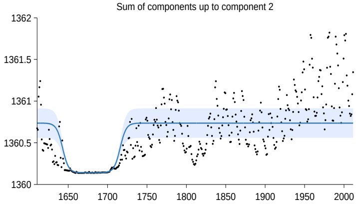




Example: Solar irradiance

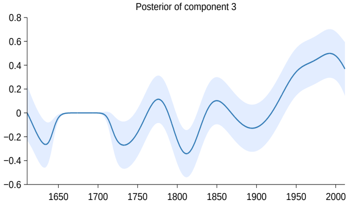
This component is constant. This component applies from 1643 until 1716.

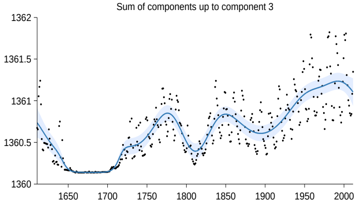




EXAMPLE: SOLAR IRRADIANCE

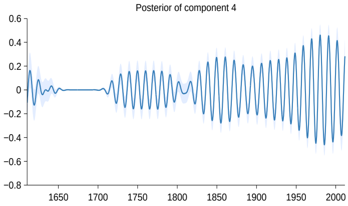
This component is a smooth function with a typical lengthscale of 23.1 years. This component applies until 1643 and from 1716 onwards.

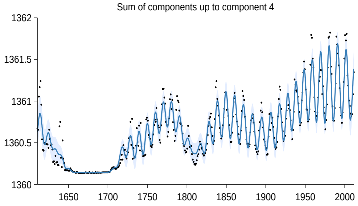




EXAMPLE: SOLAR IRRADIANCE

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.





Example: Full report

See pdf

SUMMARY

- We have briefly introduced Bayesian statistics and Gaussian processes
- Described the beginnings of an automatic statistician for regression
  - Defines an open-ended language of models
  - Searches greedily through this space
  - Produces detailed reports describing patterns in data
- We believe this line of research has the potential to make powerful statistical model-building techniques accessible to non-experts

Thanks

APPENDIX

Sum and product rules of probability

$$\text{Sum rule :} \qquad \mathbb{P}(X = x) = \sum_{y} \mathbb{P}(X = x, Y = y)$$

$$\text{Product rule :} \qquad \mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x)\,\mathbb{P}(Y = y \,|\, X = x)$$

SUM AND PRODUCT RULES OF PROBABILITY

$$\text{Sum rule :} \qquad p(x) = \int p(x, y)\, \mathrm{d}y$$

$$\text{Product rule :} \qquad p(x, y) = p(x)\, p(y \,|\, x)$$

Deriving Bayes rule

$$p(\theta, D \mid M) = p(\theta, D \mid M)$$

DERIVING BAYES RULE

$$p(\theta, D \mid M) = p(\theta, D \mid M)$$

$$p(\theta \mid D, M) p(D \mid M) = p(D \mid \theta, M) p(\theta \mid M) \quad \text{(product rule)}$$

Deriving Bayes rule

$$p(\theta, D \mid M) = p(\theta, D \mid M)$$

$$p(\theta \mid D, M) p(D \mid M) = p(D \mid \theta, M) p(\theta \mid M) \quad \text{(product rule)}$$

$$p(\theta \mid D, M) = \frac{p(D \mid \theta, M) p(\theta \mid M)}{p(D \mid M)} \quad \text{(divide)}$$

Deriving Bayes rule

$$p(\theta, D \mid M) = p(\theta, D \mid M)$$

$$p(\theta \mid D, M)p(D \mid M) = p(D \mid \theta, M)p(\theta \mid M) \quad \text{(product rule)}$$

$$p(\theta \mid D, M) = \frac{p(D \mid \theta, M)p(\theta \mid M)}{p(D \mid M)} \quad \text{(divide)}$$

$$p(\theta \mid D, M) = \frac{p(D \mid \theta, M)p(\theta \mid M)}{\int p(D, \theta \mid M)\mathrm{d}\theta} \quad \text{(sum rule)}$$

Deriving Bayes rule

$$p(\theta, D \,|\, M) = p(\theta, D \,|\, M)$$

$$p(\theta \,|\, D, M)p(D \,|\, M) = p(D \,|\, \theta, M)p(\theta \,|\, M) \quad \text{(product rule)}$$

$$p(\theta \,|\, D, M) = \frac{p(D \,|\, \theta, M)p(\theta \,|\, M)}{p(D \,|\, M)} \quad \text{(divide)}$$

$$p(\theta \,|\, D, M) = \frac{p(D \,|\, \theta, M)p(\theta \,|\, M)}{\int p(D, \theta \,|\, M)\mathrm{d}\theta} \quad \text{(sum rule)}$$

$$p(\theta \,|\, D, M) = \frac{p(D \,|\, \theta, M)p(\theta \,|\, M)}{\int p(D \,|\, \theta, M)p(\theta \,|\, M)\mathrm{d}\theta} \quad \text{(product rule)}$$

DEFINING A LANGUAGE OF REGRESSION MODELS

- Regression consists of **learning a function** $f : \mathcal{X} \to \mathcal{Y}$ from inputs to outputs from example input / output pairs

- Language should include **simple parametric forms**...
  - e.g. Linear functions, Polynomials, Exponential functions

- ...as well as functions specified by **high level properties**
  - e.g. Smoothness, Periodicity

- Inference should be **tractable for all models** in language

- It is common practice to use a zero mean function since the mean can be marginalised out
  - Suppose, $f(x) \mid a \sim \text{GP}(a \times \mu(x), k(x, x'))$ where $a \sim \mathcal{N}(0, 1)$
  - Then equivalently, $f(x) \sim \text{GP}(0, \mu(x)\mu(x') + k(x, x'))$
- We therefore define a language of GP regression models by specifying a **language of kernels**

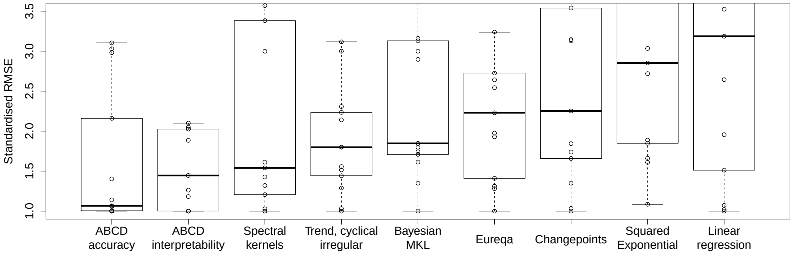
Model evaluation

- 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.

Standardised RMSE over 13 data sets



- Tweaks can be made to the algorithm to improve accuracy or interpretability of models produced. . .
- . . . but both methods are highly competitive at extrapolation (shown above) and interpolation

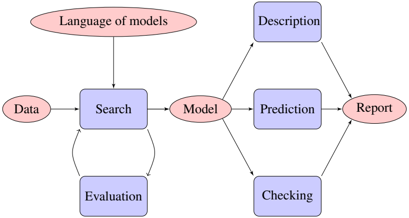
GOALS OF THE AUTOMATIC STATISTICIAN PROJECT

- Provide a set of tools for understanding data that require minimal expert input

- Uncover challenging research problems in e.g.
  - Automated inference
  - Model construction and comparison
  - Data visualisation and interpretation

- Advance the field of machine learning in general

INGREDIENTS OF AN AUTOMATIC STATISTICIAN



- **An open-ended language of models**
  - Expressive enough to capture real-world phenomena…
  - …and the techniques used by human statisticians
- **A search procedure**
  - To efficiently explore the language of models
- **A principled method of evaluating models**
  - Trading off complexity and fit to data
- **A procedure to automatically explain the models**
  - Making the assumptions of the models explicit…
  - …in a way that is intelligible to non-experts

CHALLENGES

- Interpretability / accuracy
- Increasing the expressivity of language
  - e.g. Monotonocity, positive functions, symmetries
- Computational complexity of searching through a huge space of models
- Extending the automatic reports to multidimensional datasets
  - Search and descriptions naturally extend to multiple dimensions, but automatically generating relevant visual summaries harder

CURRENT AND FUTURE DIRECTIONS

- Automatic statistician for:
  - Multivariate nonlinear regression
  - Classification
  - Completing and interpreting tables and databases

- Probabilistic programming
  - Probabilistic models are expressed in a general (Turing complete) programming language (e.g. Church/Venture/Anglican)
  - A universal inference engine can then be used to infer unobserved variables given observed data
  - This can be used to implement seach over the model space in an automated statistician