How to build an automatic statistician

James Robert Lloyd\textsuperscript{1}, David Duvenaud\textsuperscript{1}, Roger Grosse\textsuperscript{2},

Joshua Tenenbaum\textsuperscript{2}, Zoubin Ghahramani\textsuperscript{1}

\textsuperscript{1}: Department of Engineering, University of Cambridge, UK
\textsuperscript{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?

Language of models

Data → Search

Model → Prediction

Report → Checking

Description

Evaluation

Search → Model

Model → Prediction
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 output
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 $\varepsilon$ are errors or noise.
Common approach: Estimate $m$ by least squares
Bayesian approach:
- Specify beliefs about $m$ using probability distributions
Bayesian approach:
- Specify prior beliefs about $m$ using probability distributions
- Follow rules of probability to update beliefs rationally after observing data
Bayesian Statistics

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

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

We represent our uncertainty about $\theta$ 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 $\theta$ to data $D$
  - $M$ defined by $p(D \mid \theta, M)$ - the likelihood

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

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

\[
p(\theta \mid D, M) = \frac{p(D \mid \theta, M) p(\theta \mid M)}{\int p(D \mid \theta, M) p(\theta \mid M) d\theta}
\]
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 \( \varepsilon \) are errors or noise.
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 $\varepsilon$ 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 \( \varepsilon \) are errors or noise.

- Represent prior beliefs about \( m \) using a probability distribution
  - e.g. \( m \sim \text{Normal}(\text{mean} = 0, \text{variance} = 1) \)
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 $\varepsilon$ are errors or noise.

Represent prior beliefs about $m$ using a probability distribution

- e.g. $m \sim \mathcal{N}(0, 1)$
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 $\varepsilon$ 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 $\varepsilon$ using a probability distribution:
- e.g. $\varepsilon_i \sim \mathcal{N}(0, \sigma^2_{\varepsilon})$ 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 \( \varepsilon \) 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 \( \varepsilon \) using a probability distribution
  - e.g. \( \varepsilon_i \sim \mathcal{N}(0, \sigma^2_\varepsilon) \) independently for all \( i \)

- Apply Bayes rule
Bayes rule applied to linear regression

\[ p(m \mid y, x) = \frac{p(y \mid m, x)p(m)}{\int p(y \mid m, x)p(m) \, dm} \]
Bayes rule applied to linear regression

\[ p(m \mid y, x) = \frac{p(y \mid m, x) p(m)}{\int p(y \mid m, x) p(m) \, dm} \]

\[ \propto p(y \mid m, x) p(m) \]
Bayes rule applied to linear regression

\[ p(m \mid y, x) = \frac{p(y \mid m, x) p(m)}{\int p(y \mid m, x)p(m) \, dm} \]
\[ \propto p(y \mid m, x) p(m) \]
\[ \propto \left( \prod_i \frac{1}{2\pi \sigma_\epsilon} e^{-\frac{(y_i-mx_i)^2}{2\sigma_\epsilon^2}} \right) \frac{1}{2\pi} e^{-m^2/2} \]
Bayes rule applied to linear regression

\[ p(m \mid y, x) = \frac{p(y \mid m, x) p(m)}{\int p(y \mid m, x)p(m) \, dm} \]

\[ \propto p(y \mid m, x) p(m) \]

\[ \propto \left( \prod_i \frac{1}{2\pi \sigma_\varepsilon} e^{-(y_i - mx_i)^2/(2\sigma_\varepsilon^2)} \right) \frac{1}{2\pi} e^{-m^2/2} \]

\[ \propto e^{-\sum_i (y_i - mx_i)^2/(2\sigma_\varepsilon^2) - m^2/2} \]
Bayes rule applied to linear regression

\[
p(m \mid y, x) = \frac{p(y \mid m, x) p(m)}{\int p(y \mid m, x) p(m) \, dm} \\
\propto p(y \mid m, x) p(m) \\
\propto \left( \prod_i \frac{1}{2\pi \sigma_\varepsilon} e^{-\frac{(y_i-mx_i)^2}{2\sigma_\varepsilon^2}} \right) \frac{1}{2\pi} e^{-m^2/2} \\
\propto e^{-\frac{\sum_i (y_i-mx_i)^2}{2\sigma_\varepsilon^2}-m^2/2} \\
\propto e^{-\frac{\sum_i x_i^2 + \sigma_\varepsilon^2}{2\sigma_\varepsilon^2} \left( m - \frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2} \right)^2}
\]
**Bayes rule applied to linear regression**

\[
p(m \mid y, x) = \frac{p(y \mid m, x) p(m)}{\int p(y \mid m, x) p(m) \, dm} \\
\propto p(y \mid m, x) p(m) \\
\propto \left( \prod_i \frac{1}{2\pi\sigma_\varepsilon} e^{-\frac{(y_i - mx_i)^2}{2\sigma_\varepsilon^2}} \right) \frac{1}{2\pi} e^{-m^2/2} \\
\propto e^{-\sum_i (y_i - mx_i)^2 / (2\sigma_\varepsilon^2) - m^2 / 2} \\
\propto e^{-\frac{\sum_i x_i^2 + \sigma_\varepsilon^2}{2\sigma_\varepsilon^2} \left( m - \frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2} \right)^2} \\
\propto e^{\frac{-\sigma_\varepsilon^2}{2\sigma_\varepsilon^2} \left( \frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2} \right)^2}
\]

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

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

\[ m \mid y, x \sim \mathcal{N} \left( \frac{\sum_i x_i y_i}{\sum_i x_i^2 + \sigma_\varepsilon^2}, \frac{\sigma_\varepsilon^2}{\sum_i x_i^2 + \sigma_\varepsilon^2} \right) \]
We defined a Bayesian linear regression model by specifying priors on $m$ and $\varepsilon_i$. 

\[
\begin{align*}
    m & \sim \mathcal{N}(0, 1) \\
    \varepsilon_i & \sim \text{iid } \mathcal{N}(0, \sigma^2) \\
    y_i | m, \varepsilon_i & = mx_i + \varepsilon_i
\end{align*}
\]

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

\[
y_i \sim \mathcal{N}(0, x_i^2 + \sigma^2) \quad \text{(sum of two normals)}
\]

\[
\operatorname{cov}(y_i, y_j) = x_i x_j \quad \forall i \neq j
\]

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

\[
y \sim \mathcal{N}(0, K)
\]

where $k_{ij} = x_i x_j + \delta_{ij} \sigma^2$. 

\[y \sim \mathcal{N}(0, 1)
\]

\[K = \begin{bmatrix} x_1^2 & \cdots & x_n^2 \end{bmatrix}
\]

\[
\begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}
\]
We defined a Bayesian linear regression model by specifying priors on $m$ and $\varepsilon_i$

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

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

- $m \sim N(0, 1)$, $\varepsilon_i \sim \text{iid } N(0, \sigma_\varepsilon^2)$
- $y_i \mid m, \varepsilon_i = mx_i + \varepsilon_i$

This implicitly defined a joint prior on $\{y_i : i = 1, \ldots, n\}$
We defined a Bayesian linear regression model by specifying priors on \( m \) and \( \varepsilon_i \):

\[
\begin{align*}
  m &\sim \mathcal{N}(0, 1), \\
  \varepsilon_i &\sim_{\text{iid}} \mathcal{N}(0, \sigma^2_\varepsilon)
\end{align*}
\]

\[
y_i \mid m, \varepsilon_i = mx_i + \varepsilon_i
\]

This implicitly defined a joint prior on \( \{y_i : i = 1, \ldots, n\} \):

\[
y_i \sim \mathcal{N}(0, x_i^2 + \sigma^2_\varepsilon) \quad \text{(sum of two normals)}
\]
We defined a Bayesian linear regression model by specifying priors on $m$ and $\varepsilon_i$

- $m \sim \mathcal{N}(0, 1), \varepsilon_i \sim_{iid} \mathcal{N}(0, \sigma^2_\varepsilon)$
- $y_i \mid m, \varepsilon_i = mx_i + \varepsilon_i$

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

- $y_i \sim \mathcal{N}(0, x_i^2 + \sigma^2_\varepsilon)$ (sum of two normals)
- $\text{cov}(y_i, y_j) = x_ix_j \quad \forall i \neq j$
We defined a Bayesian linear regression model by specifying priors on $m$ and $\varepsilon_i$

- $m \sim \mathcal{N}(0, 1)$, $\varepsilon_i \sim_{iid} \mathcal{N}(0, \sigma^2_\varepsilon)$
- $y_i \mid m, \varepsilon_i = mx_i + \varepsilon_i$

This implicitly defined a joint prior on \{\(y_i : i = 1, \ldots, n\)\}

- $y_i \sim \mathcal{N}(0, x_i^2 + \sigma^2_\varepsilon)$ (sum of two normals)
- $\text{cov}(y_i, y_j) = x_i x_j \ \forall \ i \neq j$

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

- $y \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$
- where $k_{ij} = x_i x_j + \delta_{ij}\sigma^2_\varepsilon$
\{y_i : i = 1,\ldots, n\} has a multivariate normal distribution

- \(y \sim \mathcal{N}(0, \mathbf{K})\)
- where \(k_{ij} = x_i x_j + \delta_{ij} \sigma^2_{\varepsilon}\)
\{y_i : i = 1, \ldots, n\} has a multivariate normal distribution

- \( y \sim \mathcal{N}(0, K) \)
- \( k_{ij} = x_i x_j + \delta_{ij} \sigma^2_\varepsilon \)

The covariance can be written as a function of pairs of \( x_i \)

\[ k_{ij} = k(x_i, x_j) = x_i x_j + \delta_{x_i=x_j} \sigma^2_\varepsilon \]
\{y_i : i = 1, \ldots, n\} has a multivariate normal distribution

\[ y \sim \mathcal{N}(\mathbf{0}, \mathbf{K}) \]

\[ k_{ij} = x_ix_j + \delta_{ij}\sigma^2_\varepsilon \]

The covariance can be written as a function of pairs of \(x_i\)

\[ k_{ij} = k(x_i, x_j) = x_ix_j + \delta_{x_i=x_j}\sigma^2_\varepsilon \]

Do we define a valid probability distribution if \(\{x_i\} = \mathbb{R}\)?
A Gaussian process is collection of random variables, any finite number of which have a joint Gaussian distribution.
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, \( k(x, x') = \text{Cov}(f(x), f(x')) \)
  - Denoted \( f \sim \text{GP}(\mu, k) \)
Gaussian Processes

- A Gaussian process is a 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, \( k(x, x') = \text{Cov}(f(x), f(x')) \)
  - Denoted \( f \sim \text{GP}(\mu, k) \)

- Can be thought of as a probability distribution on functions.
LINEAR KERNEL = LINEAR FUNCTIONS

\[ k(x, x') = xx' \]
What about other kernels?

$$k(x, x') = e^{-(x-x')^2}$$
BAYESIAN NON-LINEAR REGRESSION
BAYESIAN NON-LINEAR REGRESSION

James Robert Lloyd
BAYESIAN NON-LINEAR REGRESSION
BAYESIAN NON-LINEAR REGRESSION
BAYESIAN LINEAR REGRESSION GONE WRONG
BAYESIAN LINEAR REGRESSION GONE WRONG
BAYESIAN LINEAR REGRESSION GONE WRONG
Bayesian linear regression gone wrong
BAYESIAN LINEAR REGRESSION GONE WRONG
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
Which kernel function?

How far can we go with kernel functions?
DEFINING A LANGUAGE OF MODELS

Language of models

Data → Search → Model → Prediction → Report

Evaluation → Checking → Description
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
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 $\sigma$ is a sigmoid function between 0 and 1.
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 $\sigma$ is a sigmoid function between 0 and 1.
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 \( \sigma \) 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 \( k = \text{CP}(k_1, k_2) \).
## AN EXPRESSIVE LANGUAGE OF MODELS

<table>
<thead>
<tr>
<th>Regression model</th>
<th>Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP smoothing</td>
<td>SE + WN</td>
</tr>
<tr>
<td>Linear regression</td>
<td>C + LIN + WN</td>
</tr>
<tr>
<td>Multiple kernel learning</td>
<td>∑ SE + WN</td>
</tr>
<tr>
<td>Trend, cyclical, irregular</td>
<td>∑ SE + ∑ PER + WN</td>
</tr>
<tr>
<td>Fourier decomposition</td>
<td>C + ∑ cos + WN</td>
</tr>
<tr>
<td>Sparse spectrum GPs</td>
<td>∑ cos + WN</td>
</tr>
<tr>
<td>Spectral mixture</td>
<td>∑ SE × cos + WN</td>
</tr>
<tr>
<td>Changepoints</td>
<td>e.g. CP(SE, SE) + WN</td>
</tr>
<tr>
<td>Heteroscedasticity</td>
<td>e.g. SE + LIN × WN</td>
</tr>
</tbody>
</table>

Note: cos is a special case of our version of PER
DISCOVERING A GOOD MODEL VIA SEARCH

Language of models

Data → Search → Model → Prediction → Report

Evaluation → Search → Model

Description → Prediction

Checking → Report
Discovering a good model via search

- Language defined as the arbitrary composition of five base kernels (WN, C, LIN, SE, PER) via three operators (+, ×, 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

\[(\text{Per} + \text{RQ})\]

\[
\begin{array}{cccc}
\text{Start} & \text{SE} & \text{RQ} & \text{LIN} & \text{PER} \\
\text{SE + RQ} & \ldots & \text{PER + RQ} & \ldots & \text{PER} \times \text{RQ}
\end{array}
\]
EXAMPLE: MAUNA LOA KEELING CURVE

\[ SE \times ( \text{Per} + \text{RQ} ) \]

---

Start

SE

RQ

LIN

PER

SE \times ( \text{Per} + \text{RQ} )

SE + RQ

\ldots

\text{PER} + \text{RQ}

\ldots

\text{PER} \times \text{RQ}

SE + \text{PER} + \text{RQ}

\ldots

\text{SE} \times (\text{PER} + \text{RQ})

\ldots
EXAMPLE: MAUNA LOA KEELING CURVE

\[( SE + SE \times ( Per + RQ ) ) \]
Suppose we have a collection of models \( \{M_i\} \) and some data \( D \).
Suppose we have a collection of models \( \{M_i\} \) and some data \( D \)

Bayes rule tells us

\[
p(M_i \mid D) = \frac{p(D \mid M_i)p(M_i)}{p(D)}
\]

i.e. The most likely model has the highest marginal likelihood
Suppose we have a collection of models \( \{M_i\} \) and some data \( D \)

Bayes rule tells us

\[
p(M_i \mid D) = \frac{p(D \mid M_i)p(M_i)}{p(D)}
\]

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

\[
p(M_i \mid D) \propto p(D \mid M_i) = \int p(D \mid \theta_i, M_i)p(\theta_i \mid M_i)d\theta_i
\]
Suppose we have a collection of models \( \{M_i\} \) and some data \( D \).

Bayes rule tells us

\[
p(M_i \mid D) = \frac{p(D \mid M_i)p(M_i)}{p(D)}
\]

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

\[
p(M_i \mid D) \propto p(D \mid M_i) = \int p(D \mid \theta_i, M_i)p(\theta_i \mid M_i) \, d\theta_i
\]

i.e. The most likely model has the highest marginal likelihood.
AUTOMATIC TRANSLATION OF MODELS

Language of models

Data → Search → Model → Prediction → Report

Evaluation → Checking → Description

James Robert Lloyd
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
Suppose the search finds the following kernel

\[ SE \times (WN \times LIN + CP(C, PER)) \]
Sum of Products Normal Form

Suppose the search finds the following kernel

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

The changepoint can be converted into a sum of products

$$SE \times (WN \times LIN + C \times \sigma + PER \times \bar{\sigma})$$
Suppose the search finds the following kernel

\[ SE \times (WN \times LIN + CP(C, PER)) \]

The changepoint can be converted into a sum of products

\[ SE \times (WN \times LIN + C \times \sigma + PER \times \bar{\sigma}) \]

Multiplication can be distributed over addition

\[ SE \times WN \times LIN + SE \times C \times \sigma + SE \times PER \times \bar{\sigma} \]
Suppose the search finds the following kernel

\[ SE \times (WN \times LIN + CP(C, PER)) \]

The changepoint can be converted into a sum of products

\[ SE \times (WN \times LIN + C \times \sigma + PER \times \bar{\sigma}) \]

Multiplication can be distributed over addition

\[ SE \times WN \times LIN + SE \times C \times \sigma + SE \times PER \times \bar{\sigma} \]

Simplification rules are applied

\[ WN \times LIN + SE \times \sigma + SE \times PER \times \bar{\sigma} \]
Sums of kernels are sums of functions

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

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

e.g.

We can therefore describe each component separately
On their own, each kernel is described by a standard noun phrase.
**PRODUCTS OF KERNELS - SE**

\[
\text{SE} \times \text{PER}
\]

approximately periodic function

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

![Graphs of SE and PER functions](image)
**PRODUCTS OF KERNELS - LIN**

\[
\text{SE} \times \text{PER} \times \text{LIN}
\]

- approximately periodic function 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 - CHANGEPOLNTS

\[ \text{SE} \times \text{PER} \times \text{LIN} \times \sigma \]

approximately periodic function with linearly growing amplitude until 1700

Multiplication by \( \sigma \) is equivalent to multiplying the function being modeled by a sigmoid.
AUTOMATICALLY GENERATED REPORTS

Language of models

Data → Search → Model → Prediction → Report

Description

Evaluation → Checking
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.
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.
This component models uncorrelated noise. The standard deviation of the noise increases linearly.
EXAMPLE: SOLAR IRRADIANCE

This component is constant.
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
**SUM AND PRODUCT RULES OF PROBABILITY**

**Sum rule:**
\[ P(X = x) = \sum_y P(X = x, Y = y) \]

**Product rule:**
\[ P(X = x, Y = y) = P(X = x) P(Y = y | X = x) \]
**Sum and Product Rules of Probability**

**Sum Rule:**
\[ p(x) = \int p(x, y) \, dy \]

**Product Rule:**
\[ p(x, y) = p(x) \, p(y | x) \]
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

\[ p(\theta, D \mid M) = p(\theta, D \mid M) \]

\[ p(\theta \mid D, M)p(D \mid M) = p(D \mid \theta, M)p(\theta \mid M) \]  \hspace{1cm} \text{(product rule)}

\[ p(\theta \mid D, M) = \frac{p(D \mid \theta, M)p(\theta \mid M)}{p(D \mid M)} \]  \hspace{1cm} \text{(divide)}

\[ p(\theta \mid D, M) = \frac{p(D \mid \theta, M)p(\theta \mid M)}{\int p(D, \theta \mid M)d\theta} \]  \hspace{1cm} \text{(sum rule)}
DERIVING BAYES RULE

\[ p(\theta, D | M) = p(\theta, D | M) \]

\[ p(\theta | D, M)p(D | M) = p(D | \theta, M)p(\theta | M) \] (product rule)

\[ p(\theta | D, M) = \frac{p(D | \theta, M)p(\theta | M)}{p(D | M)} \] (divide)

\[ p(\theta | D, M) = \frac{p(D | \theta, M)p(\theta | M)}{\int p(D, \theta | M)d\theta} \] (sum rule)

\[ p(\theta | D, M) = \frac{p(D | \theta, M)p(\theta | M)}{\int p(D | \theta, M)p(\theta | M)d\theta} \] (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**.
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):

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

where $p$ is the number of kernel parameters, $D$ represents the data, and $n$ is the number of data points.
GOOD PREDICTIVE PERFORMANCE AS WELL

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
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 search over the model space in an automated statistician