Automatic Construction and Natural-Language Description of Nonparametric Regression Models

James Robert Lloyd
Department of Engineering
University of Cambridge

David Duvenaud
Department of Engineering
University of Cambridge

Roger Grosse
Brain and Cognitive Sciences
Massachusetts Institute of Technology

Joshua B. Tenenbaum
Brain and Cognitive Sciences
Massachusetts Institute of Technology

Zoubin Ghahramani
Department of Engineering
University of Cambridge

Abstract

This paper presents the beginnings of an automatic statistician, focusing on regression problems. Our system explores an open-ended space of statistical models to discover a good explanation of a data set, and then produces a detailed report with figures and natural-language text.

Our approach treats unknown regression functions nonparametrically using Gaussian processes, which has two important consequences. First, Gaussian processes can model functions in terms of high-level properties (e.g. smoothness, trends, periodicity, changepoints). Taken together with the compositional structure of our language of models this allows us to automatically describe functions in simple terms. Second, the use of flexible nonparametric models and a rich language for composing them in an open-ended manner also results in state-of-the-art extrapolation performance evaluated over 13 real time series data sets from various domains.

1 Introduction

Automating the process of statistical modeling would have a tremendous impact on fields that currently rely on expert statisticians, machine learning researchers, and data scientists. While fitting simple models (such as linear regression) is largely automated by standard software packages, there has been little work on the automatic construction of flexible but interpretable models. What are the ingredients required for an artificial intelligence system to be able to perform statistical modeling automatically? In this paper we conjecture that the following ingredients may be useful for building an AI system for statistics, and we develop a working system which incorporates them:

- **An open-ended language of models** expressive enough to capture many of the modeling assumptions and model composition techniques applied by human statisticians to capture real-world phenomena
- **A search procedure** to efficiently explore the space of models spanned by the language
- **A principled method for evaluating models** in terms of their complexity and their degree of fit to the data
- **A procedure for automatically generating reports** which explain and visualize different factors underlying the data, make the chosen modeling assumptions explicit, and quantify how each component improves the predictive power of the model

In this paper we introduce a system for modeling time-series data containing the above ingredients which we call the Automatic Bayesian Covariance Discovery (ABCD) system. The system defines an open-ended language of Gaussian process models via a compositional grammar. The space is searched greedily, using marginal likelihood and the Bayesian Information Criterion (BIC) to evaluate models. The compositional structure of the language allows us to develop a method for automatically translating components of the model into natural-language descriptions of patterns in the data.

We show examples of automatically generated reports which highlight interpretable features discovered in a variety of data sets (e.g. figure 1). The supplementary material to this paper includes 13 complete reports automatically generated by ABCD.

Good statistical modeling requires not only interpretability but also predictive accuracy. We compare ABCD against existing model construction techniques in terms of predictive performance at extrapolation, and we find state-of-the-art performance on 13 time series.

2 A language of regression models

Regression consists of learning a function f mapping from some input space $\mathcal{X}$ to some output space $\mathcal{Y}$. We desire an expressive language which can represent both simple parametric forms of f such as linear or polynomial and also complex nonparametric functions specified in terms of properties such as smoothness or periodicity. Gaussian processes (GPs)

Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

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.

This component explains 71.5% of the residual variance; this increases the total variance explained from 72.8% to 92.3%. The addition of this component reduces the cross validated MAE by 16.82% from 0.18 to 0.15.

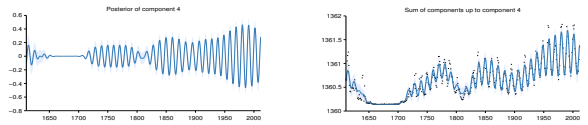

Figure 8: Pointwise posterior of component 4 (left) and the posterior of the cumulative sum of components with data (right)

Figure 1: Extract from an automatically-generated report describing the model components discovered by ABCD. This part of the report isolates and describes the approximately 11-year sunspot cycle, also noting its disappearance during the 16th century, a time known as the Maunder minimum (Lean, Beer, and Bradley, 1995).

provide a very general and analytically tractable way of capturing both simple and complex functions.

GPs are distributions over functions such that any finite set of function evaluations, $(f(x_1), f(x_2), \ldots f(x_N))$, have a jointly Gaussian distribution (Rasmussen and Williams, 2006). A GP is completely specified by its mean function, $\mu(x) = \mathbb{E}(f(x))$ and kernel (or covariance) function $k(x, x') = \mathrm{Cov}(f(x), f(x'))$. It is common practice to assume zero mean, since marginalizing over an unknown mean function can be equivalently expressed as a zero-mean GP with a new kernel. The structure of the kernel captures high-level properties of the unknown function, f, which in turn determines how the model generalizes or extrapolates to new data. We can therefore define a language of regression models by specifying a language of kernels.

The elements of this language are a set of base kernels capturing different function properties, and a set of composition rules which combine kernels to yield other valid kernels. Our base kernels are white noise (WN), constant (C), linear (LIN), squared exponential (SE) and periodic (PER), which on their own encode for uncorrelated noise, constant functions, linear functions, smooth functions and periodic functions respectively[1]. The composition rules are addition and multiplication:

$$(k_1 + k_2)(x, x') = k_1(x, x') + k_2(x, x') \tag{2.1}$$

$$(k_1 \times k_2)(x, x') = k_1(x, x') \times k_2(x, x') \tag{2.2}$$

Combining kernels using these operations can yield kernels encoding for richer structures such as approximate periodicity (SE $\times$ PER) or smooth functions with linear trends (SE $+$ LIN).

This kernel composition framework (with different base kernels) was described by Duvenaud et al. (2013). We extend and adapt this framework in several ways. In particular, we have found that incorporating changepoints into the language is essential for realistic models of time series (e.g. figure 1). We define changepoints through addition and multiplication with sigmoidal functions:

$$\mathrm{CP}(k_1, k_2) = k_1 \times \boldsymbol{\sigma} + k_2 \times \bar{\boldsymbol{\sigma}} \tag{2.3}$$

where $\boldsymbol{\sigma} = \sigma(x)\sigma(x')$ and $\bar{\boldsymbol{\sigma}} = (1 - \sigma(x))(1 - \sigma(x'))$. We define changewindows $\mathrm{CW}(\cdot, \cdot)$ similarly by replacing $\sigma(x)$ with a product of two sigmoids.

We also expanded and reparametrised the set of base kernels so that they were more amenable to automatic description (see section 6 for details) and to extend the number of common regression models included in the language. Table 1 lists common regression models that can be expressed by our language.

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

Table 1: Common regression models expressible in our language. cos is a special case of our reparametrised PER. * indicates a model that could not be expressed by the language used in Duvenaud et al. (2013).

3 Model Search and Evaluation

As in Duvenaud et al. (2013) we explore the space of regression models using a greedy search. We use the same search operators, but also include additional operators to incorporate changepoints; a complete list is contained in the supplementary material.

After each model is proposed its kernel parameters are optimised by conjugate gradient descent. We evaluate each optimized model, M, using the Bayesian Information Criterion (BIC) (Schwarz, 1978):

$$\mathrm{BIC}(M) = -2 \log p(D \mid M) + |M| \log n \tag{3.1}$$

where $|M|$ is the number of kernel parameters, $p(D|M)$ is the marginal likelihood of the data, D, and n is the number of data points. BIC trades off model fit and complexity and implements what is known as "Bayesian Occam's Razor" (e.g. Rasmussen and Ghahramani, 2001; MacKay, 2003).

4 Automatic description of regression models

Overview In this section, we describe how ABCD generates natural-language descriptions of the models found by the search procedure. There are two main features of our language of GP models that allow description to be performed automatically.

First, the sometimes complicated kernel expressions can be simplified into a sum of products. A sum of kernels corresponds to a sum of functions so each product can be described separately. Second, each kernel in a product modifies

[1]Definitions of kernels are in the supplementary material.

the resulting model in a consistent way. Therefore, we can choose one kernel to be described as a noun, with all others described using adjectives or modifiers.

Sum of products normal form We convert each kernel expression into a standard, simplified form. We do this by first distributing all products of sums into a sum of products. Next, we apply several simplifications to the kernel expression: The product of two SE kernels is another SE with different parameters. Multiplying WN by any stationary kernel (C, WN, SE, or PER) gives another WN kernel. Multiplying any kernel by C only changes the parameters of the original kernel.

After applying these rules, the kernel can as be written as a sum of terms of the form:

$$K \prod_m \text{LIN}^{(m)} \prod_n \boldsymbol{\sigma}^{(n)},$$

where K, if present, is one of WN, C, SE, $\prod_k \text{PER}^{(k)}$ or $\text{SE} \prod_k \text{PER}^{(k)}$ and $\prod_i k^{(i)}$ denotes a product of kernels, each with different parameters.

Sums of kernels are sums of functions Formally, if $f_1(x) \sim \text{GP}(0, k_1)$ and independently $f_2(x) \sim \text{GP}(0, k_2)$ then $f_1(x) + f_2(x) \sim \text{GP}(0, k_1 + k_2)$. This lets us describe each product of kernels separately.

Each kernel in a product modifies a model in a consistent way This allows us to describe the contribution of each kernel as a modifier of a noun phrase. These descriptions are summarised in table 2 and justified below:

- **Multiplication by SE** removes long range correlations from a model since $\text{SE}(x, x')$ decreases monotonically to 0 as $|x - x'|$ increases. This will convert any global correlation structure into local correlation only.
- **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.
- **Multiplication by $\boldsymbol{\sigma}$** is equivalent to multiplying the function being modeled by a sigmoid which means that the function goes to zero before or after some point.
- **Multiplication by PER** modifies the correlation structure in the same way as multiplying the function by an independent periodic function. Formally, if $f_1(x) \sim \text{GP}(0, k_1)$ and $f_2(x) \sim \text{GP}(0, k_2)$ then

$$\text{Cov}\left[f_1(x)f_2(x), f_1(x')f_2(x')\right] = k_1(x, x')k_2(x, x').$$

Constructing a complete description of a product of kernels We choose one kernel to act as a noun which is then described by the functions it encodes for when unmodified (see table 3). Modifiers corresponding to the other kernels in the product are then appended to this description, forming a noun phrase of the form:

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

Table 2: Postmodifier descriptions of each kernel

Determiner + Premodifiers + Noun + Postmodifiers

As an example, a kernel of the form $\text{PER} \times \text{LIN} \times \boldsymbol{\sigma}$ could be described as a

$$\underbrace{\text{PER}}_{\text{periodic function}} \times \underbrace{\text{LIN}}_{\text{with linearly varying amplitude}} \times \underbrace{\boldsymbol{\sigma}}_{\text{which applies until 1700.}}$$

where PER has been selected as the head noun.

Kernel	Noun phrase
WN	uncorrelated noise
C	constant
SE	smooth function
PER	periodic function
LIN	linear function
$\prod_k \text{LIN}^{(k)}$	polynomial

Table 3: Noun phrase descriptions of each kernel

Refinements to the descriptions There are a number of ways in which the descriptions of the kernels can be made more interpretable and informative:

- Which kernel is chosen as the head noun can change the interpretability of a description.
- Descriptions can change qualitatively according to kernel parameters e.g. 'a rapidly varying smooth function'.
- Descriptions can include kernel parameters e.g. 'modulated by a periodic function with a period of [period]'.
- Descriptions can include extra information calculated from data e.g. 'a linearly increasing function'.
- Some kernels can be described as premodifiers e.g. 'an approximately periodic function'.

The reports in the supplementary material and in section 5 include some of these refinements. For example, the head noun is chosen according to the following ordering:

$$\text{PER} > \text{WN}, \text{SE}, \text{C} > \prod_m \text{LIN}^{(m)} > \prod_n \boldsymbol{\sigma}^{(n)}$$

i.e. PER is always chosen as the head noun when present. The parameters and design choices of these refinements have been chosen by our best judgement, but learning these parameters objectively from expert statisticians would be an interesting area for future study.

Ordering additive components The reports generated by ABCD attempt to present the most interesting or important features of a data set first. As a heuristic, we order components by always adding next the component which most reduces the 10-fold cross-validated mean absolute error.

4.1 Worked example

Suppose we start with a kernel of the form

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

This is converted to a sum of products:

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

which is simplified to

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

To describe the first component, the head noun description for WN, 'uncorrelated noise', is concatenated with a modifier for LIN, 'with linearly increasing amplitude'. The second component is described as 'A smooth function with a lengthscale of [lengthscale] [units]', corresponding to the SE, 'which applies until [changepoint]', which corresponds to the $\boldsymbol{\sigma}$. Finally, the third component is described as 'An approximately periodic function with a period of [period] [units] which applies from [changepoint]'.

5 Example descriptions of time series

We demonstrate the ability of our procedure to discover and describe a variety of patterns on two time series. Full automatically-generated reports for 13 data sets are provided as supplementary material.

5.1 Summarizing 400 Years of Solar Activity

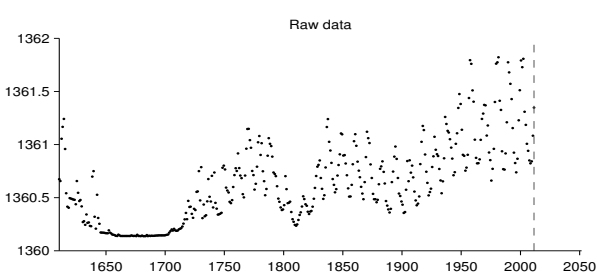

Figure 2: Solar irradiance data.

We show excerpts from the report automatically generated on annual solar irradiation data from 1610 to 2011 (figure 2). This time series has two pertinent features: a roughly 11-year cycle of solar activity, and a period lasting from 1645 to 1715 with much smaller variance than the rest of the dataset. This flat region corresponds to the Maunder minimum, a period in which sunspots were extremely rare (Lean, Beer, and Bradley, 1995). ABCD clearly identifies these two features, as discussed below.

The structure search algorithm has identified eight additive components in the data. The first 4 additive components explain 92.3% of the variation in the data as shown by the coefficient of determination (R^2) values in table 1. The first 6 additive components explain 99.7% of the variation in the data. After the first 5 components the cross validated mean absolute error (MAE) does not decrease by more than 0.1%. This suggests that subsequent terms are modelling very short term trends, uncorrelated noise or are artefacts of the model or search procedure. Short summaries of the additive components are as follows:

- A constant.
- A constant. This function applies from 1643 until 1716.
- A smooth function. This function applies until 1643 and from 1716 onwards.
- An approximately periodic function with a period of 10.8 years. This function applies until 1643 and from 1716 onwards.

Figure 3: Automatically generated descriptions of the components discovered by ABCD on the solar irradiance data set. The dataset has been decomposed into diverse structures with simple descriptions.

Figure 3 shows the natural-language summaries of the top four components chosen by ABCD. From these short summaries, we can see that our system has identified the Maunder minimum (second component) and 11-year solar cycle (fourth component). These components are visualized in figures 4 and 1, respectively. The third component corresponds to long-term trends, as visualized in figure 5.

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

This component explains 37.4% of the residual variance; this increases the total variance explained from 0.0% to 37.4%. The addition of this component reduces the cross validated MAE by 31.97% from 0.33 to 0.23.

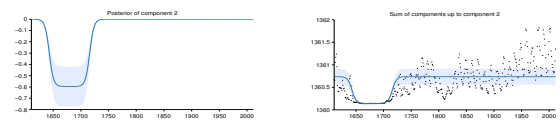

Figure 4: Pointwise posterior of component 2 (left) and the posterior of the cumulative sum of components with data (right)

Figure 4: One of the learned components corresponds to the Maunder minimum.

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

This component explains 56.6% of the residual variance; this increases the total variance explained from 37.4% to 72.8%. The addition of this component reduces the cross validated MAE by 21.08% from 0.23 to 0.18.

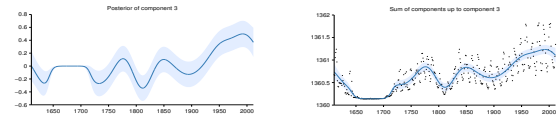

Figure 6: Pointwise posterior of component 3 (left) and the posterior of the cumulative sum of components with data (right)

Figure 5: Characterizing the medium-term smoothness of solar activity levels. By allowing other components to explain the periodicity, noise, and the Maunder minimum, ABCD can isolate the part of the signal best explained by a slowly-varying trend.

5.2 Finding heteroscedasticity in air traffic data

Next, we present the analysis generated by our procedure on international airline passenger data (figure 6). The model

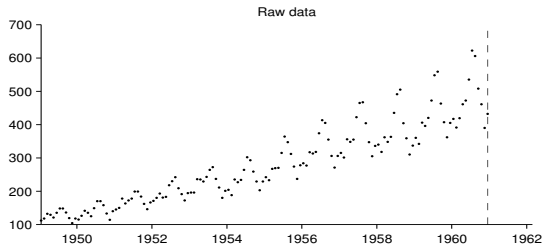

Figure 6: International airline passenger monthly volume (e.g. Box, Jenkins, and Reinsel, 2013).

constructed by ABCD has four components: LIN + SE × PER × LIN + SE + WN × LIN, with descriptions given in figure 7.

The structure search algorithm has identified four additive components in the data. The first 2 additive components explain 98.5% of the variation in the data as shown by the coefficient of determination (R^2) values in table 1. The first 3 additive components explain 99.8% of the variation in the data. After the first 3 components the cross validated mean absolute error (MAE) does not decrease by more than 0.1%. This suggests that subsequent terms are modelling very short term trends, uncorrelated noise or are artefacts of the model or search procedure. Short summaries of the additive components are as follows:

- 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.

#	R^2 (%)	ΔR^2 (%)	Residual R^2 (%)	Cross validated MAE	Reduction in MAE (%)
-	-	-	-	280.30	-
1	85.4	85.4	85.4	34.03	87.9
2	98.5	13.2	89.9	12.44	63.4
3	99.8	1.3	85.1	9.10	26.8
4	100.0	0.2	100.0	9.10	0.0

Figure 7: Short descriptions and summary statistics for the four components of the airline model.

The second component (figure 8) is accurately described as approximately (SE) periodic (PER) with linearly increasing amplitude (LIN). By multiplying a white noise kernel by

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.

This component explains 89.9% of the residual variance; this increases the total variance explained from 85.4% to 98.5%. The addition of this component reduces the cross validated MAE by 63.45% from 34.03 to 12.44.

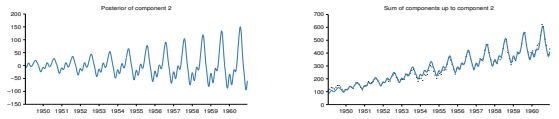

Figure 4: Pointwise posterior of component 2 (left) and the posterior of the cumulative sum of components with data (right)

Figure 8: Capturing non-stationary periodicity in the airline data

a linear kernel, the model is able to express heteroscedasticity (figure 9).

5.3 Comparison to equation learning

We now compare the descriptions generated by ABCD to parametric functions produced by an equation learning system. We show equations produced by Eureqa (Nutonian, 2011) for the data sets shown above, using the default mean absolute error performance metric.

The learned function for the solar irradiance data is

$$\text{Irradiance}(t) = 1361 + \alpha \sin(\beta + \gamma t) \sin(\delta + \epsilon t^2 - \zeta t)$$

where t is time and constants are replaced with symbols for brevity. This equation captures the constant offset of the data, and models the long-term trend with a product of sinusoids, but fails to capture the solar cycle or the Maunder minimum.

The learned function for the airline passenger data is

$$\text{Passengers}(t) = \alpha t + \beta \cos(\gamma - \delta t) \text{logistic}(\epsilon t - \zeta) - \eta$$

which captures the approximately linear trend, and the periodic component with approximately linearly (logistic) increasing amplitude. However, the annual cycle is heavily approximated by a sinusoid and the model does not capture heteroscedasticity.

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

This component explains 100.0% of the residual variance; this increases the total variance explained from 99.8% to 100.0%. The addition of this component reduces the cross validated MAE by 0.00% from 9.10 to 9.10. This component explains residual variance but does not improve MAE which suggests that this component describes very short term patterns, uncorrelated noise or is an artefact of the model or search procedure.

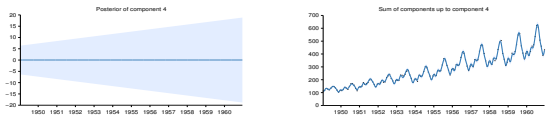

Figure 8: Pointwise posterior of component 4 (left) and the posterior of the cumulative sum of components with data (right)

Figure 9: Modeling heteroscedasticity

6 Designing kernels for interpretability

The span of the language of kernels used by ABCD is similar to those explored by Duvenaud et al. (2013) and Kronberger and Kommenda (2013). However, ABCD uses a different set of base kernels which are chosen to significantly improve the interpretability of the models produced by our method which we now discuss.

Removal of rational quadratic kernel The rational quadratic kernel (e.g. Rasmussen and Williams, 2006) can be expressed as a mixture of infinitely many SE kernels. This can have the unattractive property of capturing both long term trends and short term variation in one component.

The left of figure 10 shows the posterior of a component involving a rational quadratic kernel produced by the procedure of Duvenaud et al. (2013) on the Mauna Loa data set (see supplementary material). This component has captured both a medium term trend and short term variation. This is both visually unappealing and difficult to describe simply. In contrast, the right of figure 10 shows two of the components produced by ABCD on the same data set which clearly separate the medium term trend and short term deviations.

We do not include the Matérn kernel (e.g. Rasmussen and Williams, 2006) used by Kronberger and Kommenda (2013) for similar reasons.

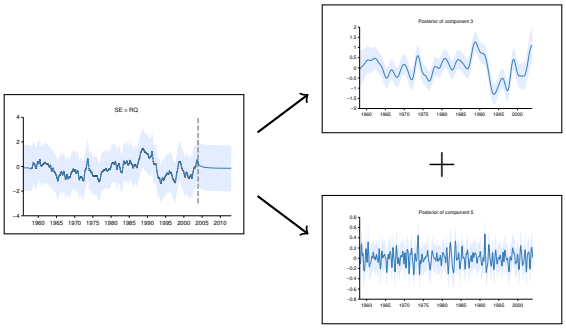

Figure 10: Left: Posterior of rational quadratic component of model for Mauna Loa data from Duvenaud et al. (2013). Right: Posterior of two components found by ABCD - the different lenthscales have been separated.

Subtraction of unnecessary constants The typical definition of the periodic kernel (e.g. Rasmussen and Williams, 2006) used by Duvenaud et al. (2013) and Kronberger and Kommenda (2013) is always greater than zero. This is not necessary for the kernel to be positive semidefinite; we can subtract a constant from this kernel. Similarly, the linear kernel used by Duvenaud et al. (2013) contained a constant term that can be subtracted.

If we had not subtracted these constant, we would have observed two main problems. First, descriptions of products would become convoluted e.g. $(\text{PER} + \text{C}) \times (\text{LIN} + \text{C}) = \text{C} + \text{PER} + \text{LIN} + \text{PER} \times \text{LIN}$ is a sum of four qualitatively different functions. Second, the constant functions can result in anti-correlation between components in the posterior, resulting in inflated credible intervals for each component which is shown in figure 11.

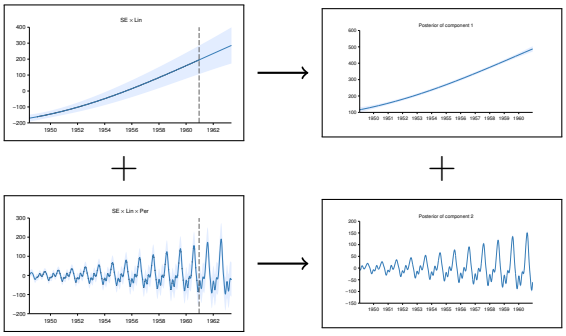

Figure 11: Left: Posterior of first two components for the airline passenger data from Duvenaud et al. (2013). Right: Posterior of first two components found by ABCD - removing the constants from LIN and PER has removed the inflated credible intervals due to anti-correlation in the posterior.

7 Related work

Building Kernel Functions Rasmussen and Williams (2006) devote 4 pages to manually constructing a composite kernel to model a time series of carbon dioxode concentrations. In the supplementary material, we include a report automatically generated by ABCD for this dataset; our procedure chose a model similar to the one they constructed by hand. Other examples of papers whose main contribution is to manually construct and fit a composite GP kernel are Klenske et al. (2013) and Lloyd (2013).

Diosan, Rogozan, and Pecuchet (2007); Bing et al. (2010) and Kronberger and Kommenda (2013) search over a similar space of models as ABCD using genetic algorithms but do not interpret the resulting models. Our procedure is based on the model construction method of Duvenaud et al. (2013) which automatically decomposed models but components were interpreted manually and the space of models searched over was smaller than that in this work.

Kernel Learning Sparse spectrum GPs (Lázaro-Gredilla et al., 2010) approximate the spectral density of a stationary kernel function using delta functions; this corresponds to kernels of the form $\sum \cos$. Similarly, Wilson and Adams (2013) introduce spectral mixture kernels which approximate the spectral density using a scale-location mixture of Gaussian distributions corresponding to kernels of the form $\sum \text{SE} \times \cos$. Both demonstrate, using Bochner's theorem (Bochner, 1959), that these kernels can approximate any stationary covariance function. Our language of kernels includes both of these kernel classes (see table 1).

There is a large body of work attempting to construct rich kernels through a weighted sum of base kernels called multiple kernel learning (MKL) (e.g. Bach, Lanckriet, and Jordan, 2004). These approaches find the optimal solution in polynomial time but only if the component kernels and parameters are pre-specified. We compare to a Bayesian variant of MKL in section 8 which is expressed as a restriction of our language of kernels.

Equation learning Todorovski and Dzeroski (1997), Washio et al. (1999) and Schmidt and Lipson (2009) learn parametric forms of functions specifying time series, or relations between quantities. In contrast, ABCD learns a parametric form for the covariance, allowing it to model functions without a simple parametric form.

Searching over open-ended model spaces This work was inspired by previous successes at searching over open-ended model spaces: matrix decompositions (Grosse, Salakhutdinov, and Tenenbaum, 2012) and graph structures (Kemp and Tenenbaum, 2008). In both cases, the model spaces were defined compositionally through a handful of components and operators, and models were selected using criteria which trade off model complexity and goodness of fit. Our work differs in that our procedure automatically interprets the chosen model, making the results accessible to non-experts.

Natural-language output To the best of our knowledge, our procedure is the first example of automatic description of nonparametric statistical models. However, systems with natural language output have been built in the areas of video

interpretation (Barbu et al., 2012) and automated theorem proving (Ganesalingam and Gowers, 2013).

8 Predictive Accuracy

In addition to our demonstration of the interpretability of ABCD, we compared the predictive accuracy of various model-building algorithms at interpolating and extrapolating time-series. ABCD outperforms the other methods on average.

Data sets We evaluate the performance of the algorithms listed below on 13 real time-series from various domains from the time series data library (Hyndman, Accessed summer 2013); plots of the data can be found at the beginning of the reports in the supplementary material.

Algorithms We compare ABCD to equation learning using Eureqa (Nutonian, 2011) and six other regression algorithms: linear regression, GP regression with a single SE kernel (squared exponential), a Bayesian variant of multiple kernel learning (MKL) (e.g. Bach, Lanckriet, and Jordan, 2004), change point modeling (e.g. Garnett et al., 2010; Saatçi, Turner, and Rasmussen, 2010; Fox and Dunson, 2013), spectral mixture kernels (Wilson and Adams, 2013) (spectral kernels) and trend-cyclical-irregular models (e.g. Lind et al., 2006).

ABCD is based on the work of Duvenaud et al. (2013), but with a focus on producing interpretable models. As noted in section 6, the spans of the languages of kernels of these two methods are very similar. Consequently their predictive accuracy is nearly identical so we only include ABCD in the results for brevity.

We use the default mean absolute error criterion when using Eureqa. All other algorithms can be expressed as restrictions of our modeling language (see table 1) so we perform inference using the same search methodology and selection criterion[2] with appropriate restrictions to the language. For MKL, trend-cyclical-irregular and spectral kernels, the greedy search procedure of ABCD corresponds to a forward-selection algorithm. For squared exponential and linear regression the procedure corresponds to marginal likelihood optimisation. More advanced inference methods are typically used for changepoint modeling but we use the same inference method for all algorithms for comparability.

We restricted to regression algorithms for comparability; this excludes models which regress on previous values of times series, such as autoregressive or moving-average models (e.g. Box, Jenkins, and Reinsel, 2013). Constructing a language for this class of time-series model would be an interesting area for future research.

Interpretability versus accuracy BIC trades off model fit and complexity by penalizing the number of parameters in a kernel expression. This can result in ABCD favoring kernel expressions with nested products of sums, producing descriptions involving many additive components. While these models have good predictive performance the large number of components can make them less interpretable. We experimented with distributing all products over addition during the search, causing models with many additive components to be more heavily penalized by BIC. We call this procedure ABCD-interpretability, in contrast to the unrestricted version of the search, ABCD-accuracy.

Extrapolation To test extrapolation we trained all algorithms on the first 90% of the data, predicted the remaining 10% and then computed the root mean squared error (RMSE). The RMSEs are then standardised by dividing by the smallest RMSE for each data set so that the best performance on each data set will have a value of 1.

Figure 12 shows the standardised RMSEs across algorithms. ABCD-accuracy outperforms ABCD-interpretability but both versions have lower quartiles than all other methods.

Overall, the model construction methods with greater capacity perform better: ABCD outperforms trend-cyclical-irregular, which outperforms Bayesian MKL, which outperforms squared exponential. Despite searching over a rich model class, Eureqa performs relatively poorly, since very few datasets are parsimoniously explained by a parametric equation.

Not shown on the plot are large outliers for spectral kernels, Eureqa, squared exponential and linear regression with values of 11, 493, 22 and 29 respectively. All of these outliers occurred on a data set with a large discontinuity (see the call centre data in the supplementary material).

Interpolation To test the ability of the methods to interpolate, we randomly divided each data set into equal amounts of training data and testing data. The results are similar to those for extrapolation and are included in the supplementary material.

9 Conclusion

Towards the goal of automating statistical modeling we have presented a system which constructs an appropriate model from an open-ended language and automatically generates detailed reports that describe patterns in the data captured by the model. We have demonstrated that our procedure can discover and describe a variety of patterns on several time series. Our procedure's extrapolation and interpolation performance on time-series are state-of-the-art compared to existing model construction techniques. We believe this procedure has the potential to make powerful statistical model-building techniques accessible to non-experts.

10 Acknowledgements

We thank Colorado Reed, Yarin Gal and Christian Steinruecken for helpful discussions. This work was funded in part by NSERC and Google.

[2] We experimented with using unpenalised marginal likelihood as the search criterion but observed overfitting, as is to be expected.

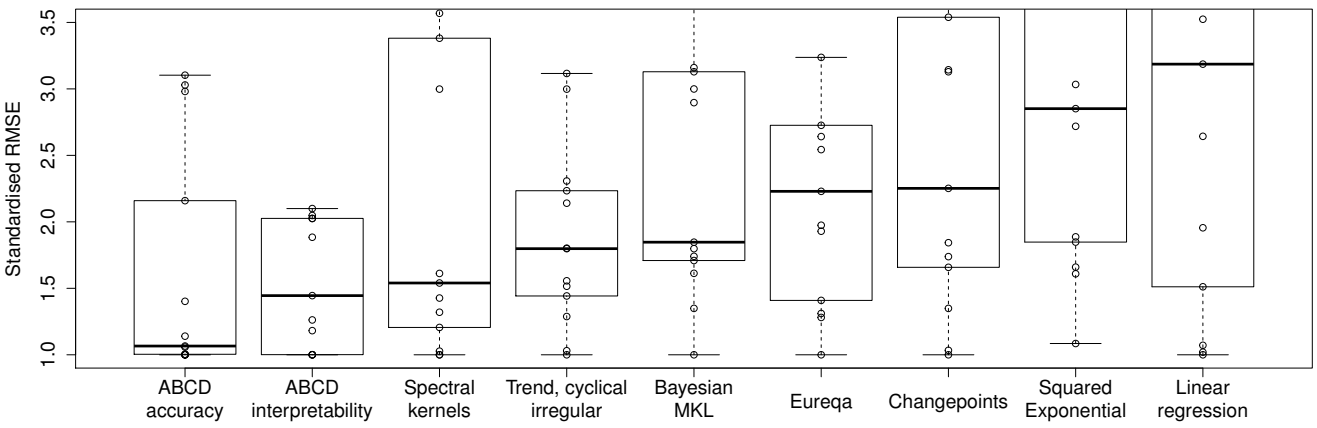

Figure 12: Raw data, and box plot (showing median and quartiles) of standardised extrapolation RMSE (best performance = 1) on 13 time-series. The methods are ordered by median.

Source Code Source code to perform all experiments is available on github[3].

References

Bach, F. R.; Lanckriet, G. R.; and Jordan, M. I. 2004. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, 6. ACM.

Barbu, A.; Bridge, A.; Burchill, Z.; Coroian, D.; Dickinson, S.; Fidler, S.; Michaux, A.; Mussman, S.; Narayanaswamy, S.; Salvi, D.; Schmidt, L.; Shangguan, J.; Siskind, J.; Waggoner, J.; Wang, S.; Wei, J.; Yin, Y.; and Zhang, Z. 2012. Video in sentences out. In *Conference on Uncertainty in Artificial Intelligence*.

Bing, W.; Wen-qiong, Z.; Ling, C.; and Jia-hong, L. 2010. A GP-based kernel construction and optimization method for RVM. In *International Conference on Computer and Automation Engineering (ICCAE)*, volume 4, 419–423.

Bochner, S. 1959. *Lectures on Fourier integrals*, volume 42. Princeton University Press.

Box, G. E.; Jenkins, G. M.; and Reinsel, G. C. 2013. *Time series analysis: forecasting and control*. Wiley. com.

Diosan, L.; Rogozan, A.; and Pecuchet, J. 2007. Evolving kernel functions for SVMs by genetic programming. In *Machine Learning and Applications, 2007*, 19–24. IEEE.

Duvenaud, D.; Lloyd, J. R.; Grosse, R.; Tenenbaum, J. B.; and Ghahramani, Z. 2013. Structure discovery in nonparametric regression through compositional kernel search. In *Proceedings of the 30th International Conference on Machine Learning*.

[3] `http://www.github.com/jamesrobertlloyd/gpss-research`. All GP parameter optimisation was performed by automated calls to the GPML toolbox available at `http://www.gaussianprocess.org/gpml/code/`.

Fox, E., and Dunson, D. 2013. Multiresolution Gaussian Processes. In *Neural Information Processing Systems 25*. MIT Press.

Ganesalingam, M., and Gowers, W. T. 2013. A fully automatic problem solver with human-style output. *CoRR* abs/1309.4501.

Garnett, R.; Osborne, M. A.; Reece, S.; Rogers, A.; and Roberts, S. J. 2010. Sequential bayesian prediction in the presence of changepoints and faults. *The Computer Journal* 53(9):1430–1446.

Grosse, R.; Salakhutdinov, R.; and Tenenbaum, J. 2012. Exploiting compositionality to explore a large space of model structures. In *Uncertainty in Artificial Intelligence*.

Hyndman, R. J. Accessed summer 2013. Time series data library.

Kemp, C., and Tenenbaum, J. 2008. The discovery of structural form. *Proceedings of the National Academy of Sciences* 105(31):10687–10692.

Klenske, E.; Zeilinger, M.; Scholkopf, B.; and Hennig, P. 2013. Nonparametric dynamics estimation for time periodic systems. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, 486–493.

Kronberger, G., and Kommenda, M. 2013. Evolution of covariance functions for gaussian process regression using genetic programming. *arXiv preprint arXiv:1305.3794*.

Lázaro-Gredilla, M.; Quiñonero-Candela, J.; Rasmussen, C. E.; and Figueiras-Vidal, A. R. 2010. Sparse spectrum gaussian process regression. *The Journal of Machine Learning Research* 99:1865–1881.

Lean, J.; Beer, J.; and Bradley, R. 1995. Reconstruction of solar irradiance since 1610: Implications for climate change. *Geophysical Research Letters* 22(23):3195–3198.

Lind, D. A.; Marchal, W. G.; Wathen, S. A.; and Magazine, B. W. 2006. *Basic statistics for business and economics*. McGraw-Hill/Irwin Boston.

Lloyd, J. R. 2013. GEFCom2012 hierarchical load forecasting: Gradient boosting machines and gaussian processes. *International Journal of Forecasting*.

MacKay, D. J. 2003. *Information theory, inference and learning algorithms*. Cambridge university press.

Nutonian. 2011. Eureqa.

Rasmussen, C., and Ghahramani, Z. 2001. Occam's razor. In *Advances in Neural Information Processing Systems*.

Rasmussen, C., and Williams, C. 2006. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, USA.

Saatçi, Y.; Turner, R. D.; and Rasmussen, C. E. 2010. Gaussian process change point models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 927–934.

Schmidt, M., and Lipson, H. 2009. Distilling free-form natural laws from experimental data. *Science* 324(5923):81–85.

Schwarz, G. 1978. Estimating the dimension of a model. *The Annals of Statistics* 6(2):461–464.

Todorovski, L., and Dzeroski, S. 1997. Declarative bias in equation discovery. In *International Conference on Machine Learning*, 376–384.

Washio, T.; Motoda, H.; Niwa, Y.; et al. 1999. Discovering admissible model equations from observed data based on scale-types and identity constraints. In *International Joint Conference On Artifical Intelligence*, volume 16, 772–779.

Wilson, A. G., and Adams, R. P. 2013. Gaussian process covariance kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning*.