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)
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 \( k(x, x') = \text{Cov}(f(x), f(x')) \). 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:

\[
\begin{align*}
(k_1 + k_2)(x, x') &= k_1(x, x') + k_2(x, x') \\
(k_1 \times k_2)(x, x') &= k_1(x, x') \times k_2(x, x')
\end{align*}
\]

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.

\[\text{Table 1: Common regression models expressible in our language.}\]

<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>( \sum ) SE ( + ) WN</td>
</tr>
<tr>
<td>Trend, cyclical, irregular</td>
<td>( \sum ) SE ( + \sum ) PER ( + ) WN</td>
</tr>
<tr>
<td>Fourier decomposition*</td>
<td>( C ) ( + \sum ) COS ( + ) WN</td>
</tr>
<tr>
<td>Sparse spectrum GPs*</td>
<td>( \sum ) COS ( + ) WN</td>
</tr>
<tr>
<td>Spectral mixture*</td>
<td>e.g. CP(SE, SE) ( + ) WN</td>
</tr>
<tr>
<td>Changepoints*</td>
<td>e.g. SE ( + ) LIN ( \times ) WN</td>
</tr>
</tbody>
</table>

\[\text{Table 1: Common regression models expressible in our language.}\]

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 optimized by conjugate gradient descent. We evaluate each optimized model, \( M \), using the Bayesian Information Criterion (BIC) (Schwarz, 1978):

\[
\text{BIC}(M) = -2 \log p(D \mid M) + |M| \log n
\]

where \(|M|\) is the number of kernel parameters, \( p(D \mid M) \) 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
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:

\[ K \prod_{m} \text{LIN}^{(m)} \prod_{n} \sigma^{(n)}, \]

where \( K \), if present, is one of WN, C, SE, \( \prod_k \text{PER}^{(k)} \) or SE \( \prod_k \text{PER}^{(k)} \) and \( \prod_k \text{PER}^{(k)} \) 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 \( \sigma \)** 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 \( \text{Cov} [f_1(x)f_2(x), f_1(x')f_2(x')] = k_1(x,x')k_2(x,x') \).

**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

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Postmodifier phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE</td>
<td>whose shape changes smoothly</td>
</tr>
<tr>
<td>PER</td>
<td>modulated by a periodic function</td>
</tr>
<tr>
<td>LIN</td>
<td>with linearly varying amplitude</td>
</tr>
<tr>
<td>( \prod_k \text{LIN}^{(k)} )</td>
<td>with polynomially varying amplitude</td>
</tr>
<tr>
<td>( \prod_k \sigma^{(k)} )</td>
<td>which applies until / from [changepoint]</td>
</tr>
</tbody>
</table>

Table 2: Postmodifier descriptions of each kernel

The product are then appended to this description, forming a noun phrase of the form:

**Determiner + Premodifiers + Noun + Postmodifiers**

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

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

where \( \text{PER} \) has been selected as the head noun.

**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:

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

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

\[ \text{SE} \times (\text{WN} \times \text{LIN} + \text{CP}(C, \text{PER})). \]

This is converted to a sum of products:

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

which is simplified to

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

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 \( \sigma \). 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

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.
constructed by ABCD has four components: \( \text{LIN} + \text{SE} \times \text{PER} \times \text{LIN} + \text{WN} \times \text{LIN} \), 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 is an artefact 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.

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 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 sys-
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. \((\text{PER} + C) \times (\text{LIN} + C) = C + \text{PER} + \text{LIN} + \text{PER} \times \text{LIN}\) 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 \text{LIN} and \text{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 compos-
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; Saatç, 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 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.
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.

References


3http://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/.


