Chapter 1

Automatically Building Structured Covariance Functions

> "It would be very nice to have a formal apparatus that gives us some 'optimal' way of recognizing unusual phenomena and inventing new classes of hypotheses that are most likely to contain the true one; but this remains an art for the creative human mind."
>
> E. T. Jaynes, 1985

Joint work with James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, Zoubin Ghahramani

Despite its importance, choosing the structural form of the kernel in nonparametric regression remains a black art. We define a space of kernel structures which are built compositionally by adding and multiplying a small number of base kernels. We present a method for searching over this space of structures which mirrors the scientific discovery process. The learned structures can often decompose functions into interpretable components and enable long-range extrapolation on time-series datasets. Our structure search method outperforms many widely used kernels and kernel combination methods on a variety of prediction tasks.

1.1 Introduction

Kernel-based nonparametric models, such as support vector machines and Gaussian processes (GPs), have been one of the dominant paradigms for supervised machine learning over the last 20 years. These methods depend on defining a kernel function, $k(x, x')$,

which specifies how similar or correlated outputs y and y' are expected to be at two inputs x and x'. By defining the measure of similarity between inputs, the kernel determines the pattern of inductive generalization.

Most existing techniques pose kernel learning as a (possibly high-dimensional) parameter estimation problem. Examples include learning hyperparameters (Rasmussen and Williams, 2006), linear combinations of fixed kernels Bach (2009), and mappings from the input space to an embedding space Salakhutdinov and Hinton (2008).

However, to apply existing kernel learning algorithms, the user must specify the parametric form of the kernel, and this can require considerable expertise, as well as trial and error.

To make kernel learning more generally applicable, we reframe the kernel learning problem as one of structure discovery, and automate the choice of kernel form. In particular, we formulate a space of kernel structures defined compositionally in terms of sums and products of a small number of base kernel structures. This provides an expressive modeling language which concisely captures many widely used techniques for constructing kernels. We focus on Gaussian process regression, where the kernel specifies a covariance function, because the Bayesian framework is a convenient way to formalize structure discovery. Borrowing discrete search techniques which have proved successful in equation discovery Todorovski and Dzeroski (1997) and unsupervised learning Grosse et al. (2012), we automatically search over this space of kernel structures using marginal likelihood as the search criterion.

We found that our structure discovery algorithm is able to automatically recover known structures from synthetic data as well as plausible structures for a variety of real-world datasets. On a variety of time series datasets, the learned kernels yield decompositions of the unknown function into interpretable components that enable accurate extrapolation beyond the range of the observations. Furthermore, the automatically discovered kernels outperform a variety of widely used kernel classes and kernel combination methods on supervised prediction tasks.

While we focus on Gaussian process regression, we believe our kernel search method can be extended to other supervised learning frameworks such as classification or ordinal regression, or to other kinds of kernel architectures such as kernel SVMs. We hope that the algorithm developed in this paper will help replace the current and often opaque art of kernel engineering with a more transparent science of automated kernel construction.

Example expressions In addition to the examples given in Figure **??**, many common motifs of supervised learning can be captured using sums and products of one-dimensional base kernels:

Bayesian linear regression	Lin
Bayesian polynomial regression	$\text{Lin} \times \text{Lin} \times \ldots$
Generalized Fourier decomposition	$\text{Per} + \text{Per} + \ldots$
Generalized additive models	$\sum_{d=1}^{D} \text{SE}_d$
Automatic relevance determination	$\prod_{d=1}^{D} \text{SE}_d$
Linear trend with local deviations	$\text{Lin} + \text{SE}$
Linearly growing amplitude	$\text{Lin} \times \text{SE}$

We use the term 'generalized Fourier decomposition' to express that the periodic functions expressible by a GP with a periodic kernel are not limited to sinusoids.

1.2 Searching over structures

As discussed above, we can construct a wide variety of kernel structures compositionally by adding and multiplying a small number of base kernels. In particular, we consider the four base kernel families discussed in Section **??**: SE, Per, Lin, and RQ. Any algebraic expression combining these kernels using the operations $+$ and $\times$ defines a kernel family, whose parameters are the concatenation of the parameters for the base kernel families.

Our search procedure begins by proposing all base kernel families applied to all input dimensions. We allow the following search operators over our set of expressions:

(1) Any subexpression $\mathcal{S}$ can be replaced with $\mathcal{S}+\mathcal{B}$, where $\mathcal{B}$ is any base kernel family.

(2) Any subexpression $\mathcal{S}$ can be replaced with $\mathcal{S}\times\mathcal{B}$, where $\mathcal{B}$ is any base kernel family.

(3) Any base kernel $\mathcal{B}$ may be replaced with any other base kernel family $\mathcal{B}'$.

These operators can generate all possible algebraic expressions. To see this, observe that if we restricted the $+$ and $\times$ rules only to apply to base kernel families, we would obtain a context-free grammar (CFG) which generates the set of algebraic expressions. However, the more general versions of these rules allow more flexibility in the search procedure, which is useful because the CFG derivation may not be the most straightforward way to arrive at a kernel family.

Our algorithm searches over this space using a greedy search: at each stage, we choose the highest scoring kernel and expand it by applying all possible operators.

Our search operators are motivated by strategies researchers often use to construct kernels. In particular,

- One can look for structure, e.g. periodicity, in the residuals of a model, and then extend the model to capture that structure. This corresponds to applying rule (1).
- One can start with structure, e.g. linearity, which is assumed to hold globally, but find that it only holds locally. This corresponds to applying rule (2) to obtain the structure shown in rows 1 and 3 of figure **??**.
- One can add features incrementally, analogous to algorithms like boosting, backfitting, or forward selection. This corresponds to applying rules (1) or (2) to dimensions not yet included in the model.

Scoring kernel families Choosing kernel structures requires a criterion for evaluating structures. We choose marginal likelihood as our criterion, since it balances the fit and complexity of a model (Rasmussen and Ghahramani, 2001). Conditioned on kernel parameters, the marginal likelihood of a GP can be computed analytically. However, to evaluate a kernel family we must integrate over kernel parameters. We approximate this intractable integral with the Bayesian information criterion (Schwarz, 1978) after first optimizing to find the maximum-likelihood kernel parameters.

Unfortunately, optimizing over parameters is not a convex optimization problem, and the space can have many local optima. For example, in data with periodic structure, integer multiples of the true period (i.e. harmonics) are often local optima. To alleviate this difficulty, we take advantage of our search procedure to provide reasonable initializations: all of the parameters which were part of the previous kernel are initialized to their previous values. All parameters are then optimized using conjugate gradients, randomly restarting the newly introduced parameters. This procedure is not guaranteed to find the global optimum, but it implements the commonly used heuristic of iteratively modeling residuals.

1.3 Related Work

Nonparametric regression in high dimensions Nonparametric regression methods such as splines, locally weighted regression, and GP regression are popular because they are capable of learning arbitrary smooth functions of the data. Unfortunately, they suffer from the curse of dimensionality: it is very difficult for the basic versions of these

methods to generalize well in more than a few dimensions. Applying nonparametric methods in high-dimensional spaces can require imposing additional structure on the model.

One such structure is additivity. Generalized additive models (GAM) assume the regression function is a transformed sum of functions defined on the individual dimensions: $\mathbb{E}[f(\mathbf{x})] = g^{-1}(\sum_{d=1}^{D} f_d(x_d))$. These models have a limited compositional form, but one which is interpretable and often generalizes well. In our grammar, we can capture analogous structure through sums of base kernels along different dimensions.

It is possible to add more flexibility to additive models by considering higher-order interactions between different dimensions. Additive Gaussian processes Duvenaud et al. (2011) are a GP model whose kernel implicitly sums over all possible products of one-dimensional base kernels. Plate (1999) constructs a GP with a composite kernel, summing an SE kernel along each dimension, with an SE-ARD kernel (i.e. a product of SE over all dimensions). Both of these models can be expressed in our grammar.

A closely related procedure is smoothing-splines ANOVA Gu (2002); Wahba (1990). This model is a linear combinations of splines along each dimension, all pairs of dimensions, and possibly higher-order combinations. Because the number of terms to consider grows exponentially in the order, in practice, only terms of first and second order are usually considered.

Semiparametric regression (e.g. Ruppert et al., 2003) attempts to combine interpretability with flexibility by building a composite model out of an interpretable, parametric part (such as linear regression) and a 'catch-all' nonparametric part (such as a GP with an SE kernel). In our approach, this can be represented as a sum of SE and Lin.

Kernel learning There is a large body of work attempting to construct a rich kernel through a weighted sum of base kernels (e.g. Bach, 2009; Christoudias et al., 2009). While these approaches find the optimal solution in polynomial time, speed comes at a cost: the component kernels, as well as their hyperparameters, must be specified in advance.

Another approach to kernel learning is to learn an embedding of the data points. Lawrence (2005) learns an embedding of the data into a low-dimensional space, and constructs a fixed kernel structure over that space. This model is typically used in unsupervised tasks and requires an expensive integration or optimisation over potential embeddings when generalizing to test points. Salakhutdinov and Hinton (2008) use a

deep neural network to learn an embedding; this is a flexible approach to kernel learning but relies upon finding structure in the input density, p(x). Instead we focus on domains where most of the interesting structure is in f(x).

Wilson and Adams (2013) derive kernels of the form SE $\times \cos(x - x')$, forming a basis for stationary kernels. These kernels share similarities with SE $\times$ Per but can express negative prior correlation, and could usefully be included in our grammar.

Diosan et al. (2007) and Bing et al. (2010) learn composite kernels for support vector machines and relevance vector machines, using genetic search algorithms. Our work employs a Bayesian search criterion, and goes beyond this prior work by demonstrating the interpretability of the structure implied by composite kernels, and how such structure allows for extrapolation.

Structure discovery There have been several attempts to uncover the structural form of a dataset by searching over a grammar of structures. For example, Schmidt and Lipson (2009), Todorovski and Dzeroski (1997) and Washio et al. (1999) attempt to learn parametric forms of equations to describe time series, or relations between quantities. Because we learn expressions describing the covariance structure rather than the functions themselves, we are able to capture structure which does not have a simple parametric form.

Kemp and Tenenbaum (2008) learned the structural form of a graph used to model human similarity judgments. Examples of graphs included planes, trees, and cylinders. Some of their discrete graph structures have continous analogues in our own space; e.g. $\text{SE}_1 \times \text{SE}_2$ and $\text{SE}_1 \times \text{Per}_2$ can be seen as mapping the data to a plane and a cylinder, respectively.

Grosse et al. (2012) performed a greedy search over a compositional model class for unsupervised learning, using a grammar and a search procedure which parallel our own. This model class contained a large number of existing unsupervised models as special cases and was able to discover such structure automatically from data. Our work is tackling a similar problem, but in a supervised setting.

1.4 Structure discovery in time series

To investigate our method's ability to discover structure, we ran the kernel search on several time-series.

As discussed in section 2, a GP whose kernel is a sum of kernels can be viewed

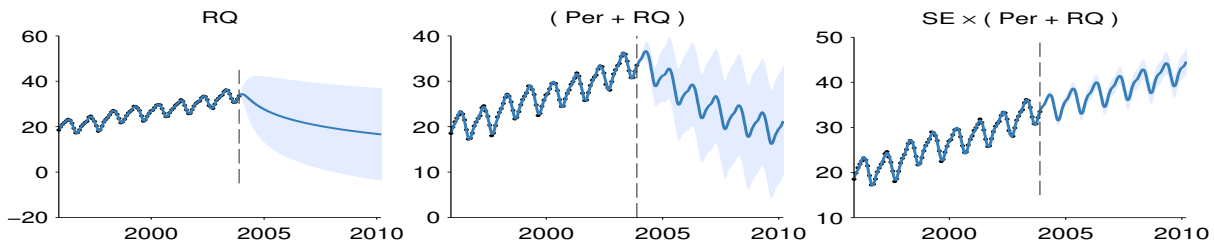

Fig. 1.1 Posterior mean and variance for different depths of kernel search. The dashed line marks the extent of the dataset. In the first column, the function is only modeled as a locally smooth function, and the extrapolation is poor. Next, a periodic component is added, and the extrapolation improves. At depth 3, the kernel can capture most of the relevant structure, and is able to extrapolate reasonably.

as a sum of functions drawn from component GPs. This provides another method of visualizing the learned structures. In particular, all kernels in our search space can be equivalently written as sums of products of base kernels by applying distributivity. For example,

$$\mathrm{SE} \times (\mathrm{RQ} + \mathrm{Lin}) = \mathrm{SE} \times \mathrm{RQ} + \mathrm{SE} \times \mathrm{Lin}.$$

We visualize the decompositions into sums of components using the formulae given in the appendix. The search was run to depth 10, using the base kernels from Section **??**.

Mauna Loa atmospheric CO_2 Using our method, we analyzed records of carbon dioxide levels recorded at the Mauna Loa observatory. Since this dataset was analyzed in detail by Rasmussen and Williams (2006), we can compare the kernel chosen by our method to a kernel constructed by human experts.

Figure 1.4 shows the posterior mean and variance on this dataset as the search depth increases. While the data can be smoothly interpolated by a single base kernel model, the extrapolations improve dramatically as the increased search depth allows more structure to be included.

Figure 1.2 shows the final model chosen by our method, together with its decomposition into additive components. The final model exhibits both plausible extrapolation and interpretable components: a long-term trend, annual periodicity and medium-term deviations; the same components chosen by Rasmussen and Williams (2006). We also plot the residuals, observing that there is little obvious structure left in the data.

Airline passenger data Figure 1.6 shows the decomposition produced by applying our method to monthly totals of international airline passengers (Box et al., 1976).

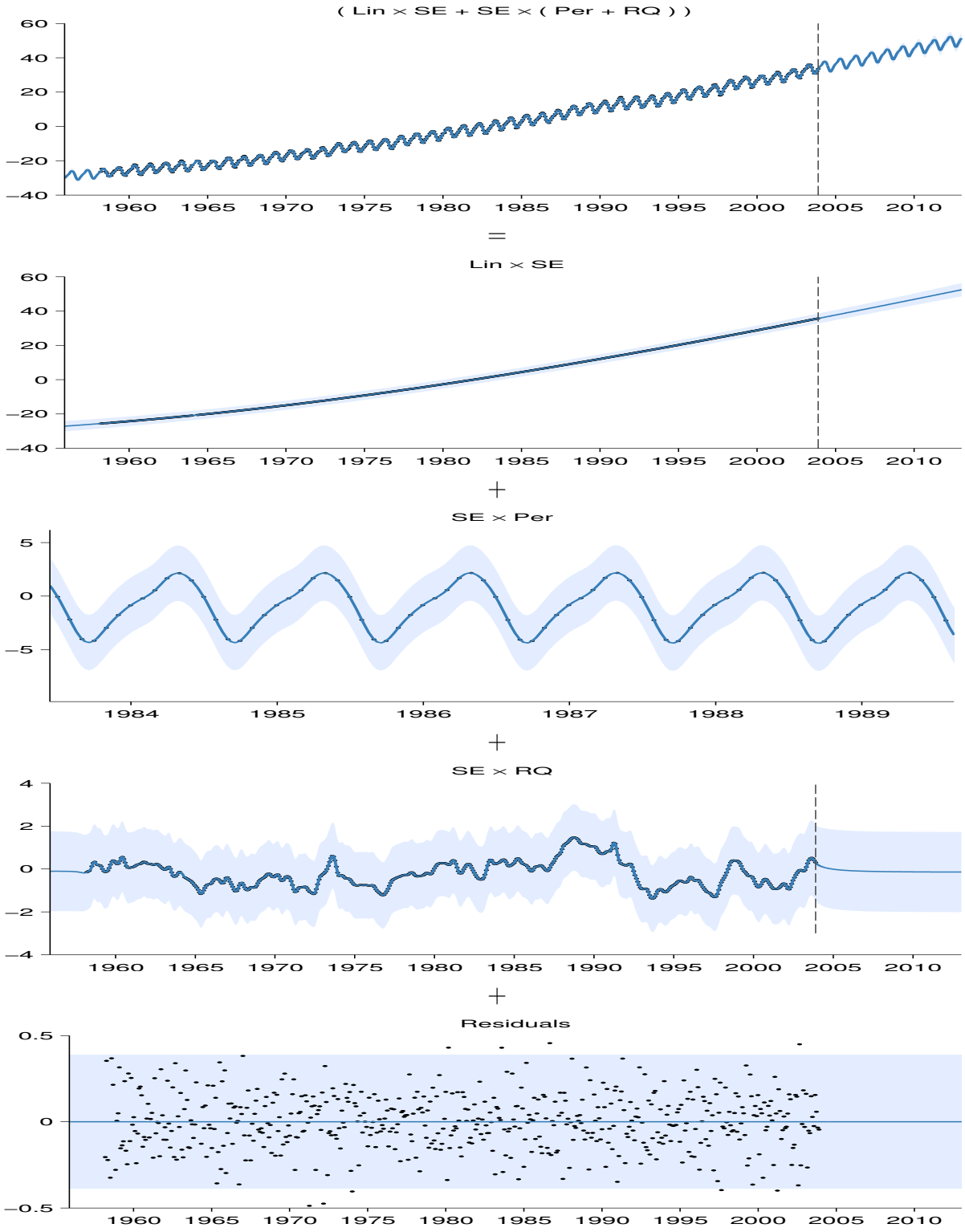

Fig. 1.2 First row: The posterior on the Mauna Loa dataset, after a search of depth 10. Subsequent rows show the automatic decomposition of the time series. The decompositions shows long-term, yearly periodic, medium-term anomaly components, and residuals, respectively. In the third row, the scale has been changed in order to clearly show the yearly periodic structure.

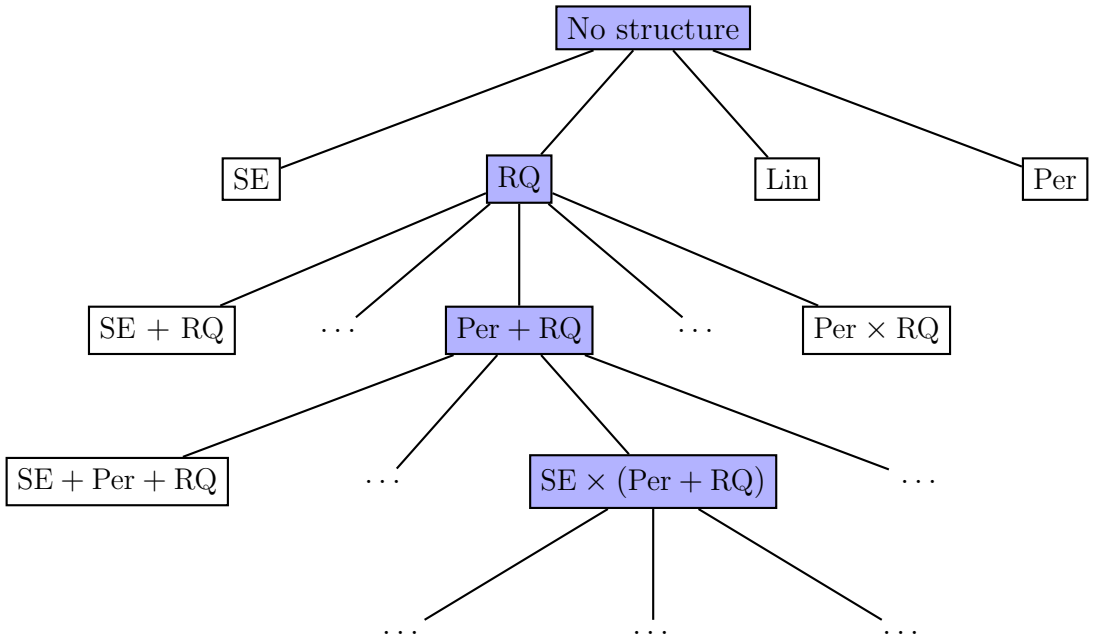

Fig. 1.3 An example of a search tree over kernel expressions. Figure 1.4 shows the model increasing in sophistication as the kernel expression grows.

We observe similar components to the previous dataset: a long term trend, annual periodicity and medium-term deviations. In addition, the composite kernel captures the near-linearity of the long-term trend, and the linearly growing amplitude of the annual oscillations.

Solar irradiance Data Finally, we analyzed annual solar irradiation data from 1610 to 2011 (Lean et al., 1995). The posterior and residuals of the learned kernel are shown in figure 1.5. None of the models in our search space are capable of parsimoniously representing the lack of variation from 1645 to 1715. Despite this, our approach fails gracefully: the learned kernel still captures the periodic structure, and the quickly growing posterior variance demonstrates that the model is uncertain about long term structure.

1.5 Validation on synthetic data

We validated our method's ability to recover known structure on a set of synthetic datasets. For several composite kernel expressions, we constructed synthetic data by first sampling 300 points uniformly at random, then sampling function values at those points from a GP prior. We then added i.i.d. Gaussian noise to the functions, at various

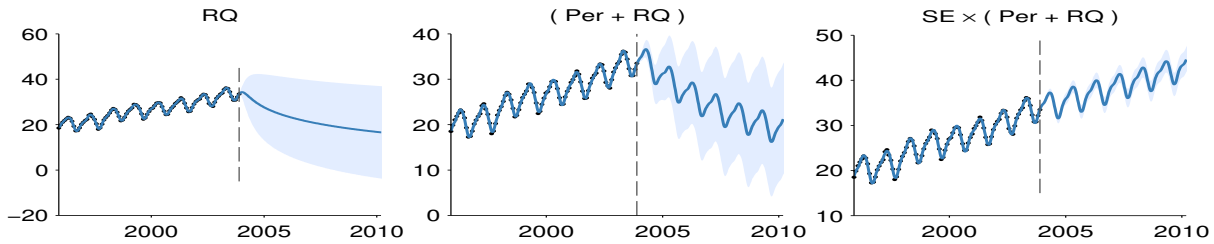

Fig. 1.4 Posterior mean and variance for different depths of kernel search. The dashed line marks the extent of the dataset. In the first column, the function is only modeled as a locally smooth function, and the extrapolation is poor. Next, a periodic component is added, and the extrapolation improves. At depth 3, the kernel can capture most of the relevant structure, and is able to extrapolate reasonably.

Table 1.1 Kernels chosen by our method on synthetic data generated using known kernel structures. D denotes the dimension of the functions being modeled. SNR indicates the signal-to-noise ratio. Dashes - indicate no structure.

True Kernel	D	SNR = 10	SNR = 1	SNR = 0.1
$\mathrm{SE} + \mathrm{RQ}$	1	SE	$\mathrm{SE} \times \mathrm{Per}$	SE
$\mathrm{Lin} \times \mathrm{Per}$	1	$\mathrm{Lin} \times \mathrm{Per}$	$\mathrm{Lin} \times \mathrm{Per}$	SE
$\mathrm{SE}_1 + \mathrm{RQ}_2$	2	$\mathrm{SE}_1 + \mathrm{SE}_2$	$\mathrm{Lin}_1 + \mathrm{SE}_2$	Lin_1
$\mathrm{SE}_1 + \mathrm{SE}_2 \times \mathrm{Per}_1 + \mathrm{SE}_3$	3	$\mathrm{SE}_1 + \mathrm{SE}_2 \times \mathrm{Per}_1 + \mathrm{SE}_3$	$\mathrm{SE}_2 \times \mathrm{Per}_1 + \mathrm{SE}_3$	-
$\mathrm{SE}_1 \times \mathrm{SE}_2$	4	$\mathrm{SE}_1 \times \mathrm{SE}_2$	$\mathrm{Lin}_1 \times \mathrm{SE}_2$	Lin_2
$\mathrm{SE}_1 \times \mathrm{SE}_2 + \mathrm{SE}_2 \times \mathrm{SE}_3$	4	$\mathrm{SE}_1 \times \mathrm{SE}_2 + \mathrm{SE}_2 \times \mathrm{SE}_3$	$\mathrm{SE}_1 + \mathrm{SE}_2 \times \mathrm{SE}_3$	SE_1
$(\mathrm{SE}_1 + \mathrm{SE}_2) \times (\mathrm{SE}_3 + \mathrm{SE}_4)$	4	$(\mathrm{SE}_1 + \mathrm{SE}_2) \times \ldots$ $(\mathrm{SE}_3 \times \mathrm{Lin}_3 \times \mathrm{Lin}_1 + \mathrm{SE}_4)$	$(\mathrm{SE}_1 + \mathrm{SE}_2) \times \ldots$ $\mathrm{SE}_3 \times \mathrm{SE}_4$	-

signal-to-noise ratios (SNR).

Table 1.1 lists the true kernels we used to generate the data. Subscripts indicate which dimension each kernel was applied to. Subsequent columns show the dimensionality D of the input space, and the kernels chosen by our search for different SNRs. Dashes - indicate that no kernel had a higher marginal likelihood than modeling the data as i.i.d. Gaussian noise.

For the highest SNR, the method finds all relevant structure in all but one test. The reported additional linear structure is explainable by the fact that functions sampled from SE kernels with long length scales occasionally have near-linear trends. As the noise increases, our method generally backs off to simpler structures.

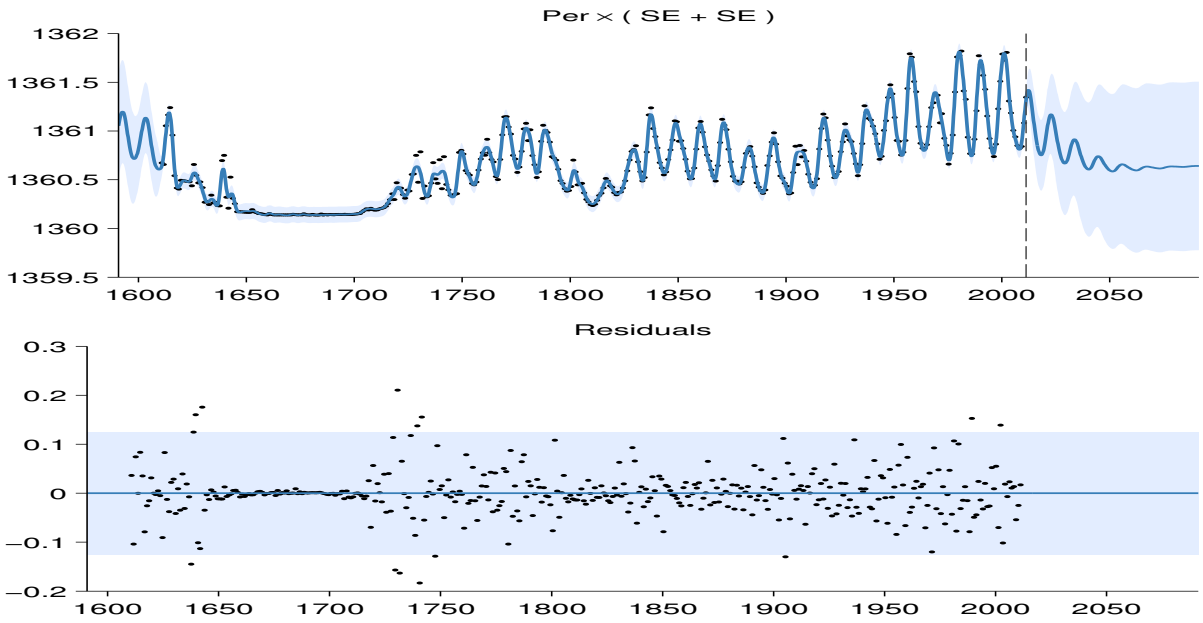

Fig. 1.5 Full posterior and residuals on the solar irradiance dataset.

Table 1.2 Comparison of multidimensional regression performance. Bold results are not significantly different from the best-performing method in each experiment, in a paired t-test with a p-value of 5%.

	Mean Squared Error (MSE)					Negative Log-Likelihood				
Method	bach	concret	puma	servo	housing	bach	concret	puma	servo	housing
Linear Regression	1.031	0.404	0.641	0.523	0.289	2.430	1.403	1.881	1.678	1.052
GAM	1.259	0.149	0.598	0.281	0.161	1.708	0.467	1.195	0.800	0.457
HKL	**0.199**	0.147	0.346	0.199	0.151	-	-	-	-	-
GP SE-ARD	**0.045**	0.157	**0.317**	**0.126**	**0.092**	**−0.131**	0.398	**0.843**	0.429	0.207
Additive GP	**0.045**	**0.089**	**0.316**	**0.110**	0.102	**−0.131**	**0.114**	**0.841**	0.309	0.194
Search - SE, RQ	**0.044**	**0.087**	**0.315**	**0.102**	**0.082**	**−0.141**	**0.065**	**0.840**	0.265	**0.059**
Search - All kernels	**0.509**	**0.079**	**0.321**	**0.094**	**0.112**	**0.357**	**0.114**	**0.837**	**−0.427**	**0.151**

1.6 Quantitative evaluation

In addition to the qualitative evaluation in section 1.4, we investigated quantitatively how our method performs on both extrapolation and interpolation tasks.

1.6.1 Extrapolation

We compared the extrapolation capabilities of our model against standard baselines[1]. Dividing the airline dataset into contiguous training and test sets, we computed the predictive mean-squared-error (MSE) of each method. We varied the size of the training set from the first 10% to the first 90% of the data.

Figure 1.7 shows the learning curves of linear regression, a variety of fixed kernel family GP models, and our method. GP models with only SE and Per kernels did not capture the long-term trends, since the best parameter values in terms of GP marginal likelihood only capture short term structure. Linear regression approximately captured the long-term trend, but quickly plateaued in predictive performance. The more richly structured GP models (SE + Per and SE × Per) eventually captured more structure and performed better, but the full structures discovered by our search outperformed the other approaches in terms of predictive performance for all data amounts.

1.6.2 High-dimensional prediction

To evaluate the predictive accuracy of our method in a high-dimensional setting, we extended the comparison of Duvenaud et al. (2011) to include our method. We performed 10 fold cross validation on 5 datasets [2] comparing 5 methods in terms of MSE and predictive likelihood. Our structure search was run up to depth 10, using the SE and RQ base kernel families.

The comparison included three methods with fixed kernel families: Additive GPs, Generalized Additive Models (GAM), and a GP with a standard SE kernel using Automatic Relevance Determination (GP SE-ARD). Also included was the related kernel-search method of Hierarchical Kernel Learning (HKL).

Results are presented in table 1.2. Our method outperformed the next-best method in each test, although not substantially.

All GP hyperparameter tuning was performed by automated calls to the GPML toolbox[3]; Python code to perform all experiments is available on github[4].

[1] In one dimension, the predictive means of all baseline methods in table 1.2 are identical to that of a GP with an SE kernel.

[2] The data sets had dimensionalities ranging from 4 to 13, and the number of data points ranged from 150 to 450.

[3] Available at `www.gaussianprocess.org/gpml/code/`

[4] `github.com/jamesrobertlloyd/gp-structure-search`

1.7 Discussion

Towards the goal of automating the choice of kernel family, we introduced a space of composite kernels defined compositionally as sums and products of a small number of base kernels. The set of models included in this space includes many standard regression models. We proposed a search procedure for this space of kernels which parallels the process of scientific discovery.

We found that the learned structures are often capable of accurate extrapolation in complex time-series datasets, and are competitive with widely used kernel classes and kernel combination methods on a variety of prediction tasks. The learned kernels often yield decompositions of a signal into diverse and interpretable components, enabling model-checking by humans. We believe that a data-driven approach to choosing kernel structures automatically can help make nonparametric regression and classification methods accessible to non-experts.

1.8 Appendix

Kernel definitions For scalar-valued inputs, the squared exponential (SE), periodic (Per), linear (Lin), and rational quadratic (RQ) kernels are defined as follows:

$$\begin{aligned}
k_{\mathrm{SE}}(x, x') &= \sigma^2 \exp\left(-\tfrac{(x-x')^2}{2\ell^2}\right) \\
k_{\mathrm{Per}}(x, x') &= \sigma^2 \exp\left(-\tfrac{2\sin^2(\pi(x-x')/p)}{\ell^2}\right) \\
k_{\mathrm{Lin}}(x, x') &= \sigma_b^2 + \sigma_v^2(x-\ell)(x'-\ell) \\
k_{\mathrm{RQ}}(x, x') &= \sigma^2 \left(1 + \tfrac{(x-x')^2}{2\alpha\ell^2}\right)^{-\alpha}
\end{aligned}$$

Posterior decomposition We can analytically decompose a GP posterior distribution over additive components using the following identity: The conditional distribution of a Gaussian vector $\mathbf{f}_1$ conditioned on its sum with another Gaussian vector $\mathbf{f} = \mathbf{f}_1 + \mathbf{f}_2$ where $\mathbf{f}_1 \sim \mathcal{N}(\mu_1, \mathbf{K}_1)$ and $\mathbf{f}_2 \sim \mathcal{N}(\mu_2, \mathbf{K}_2)$ is given by

$$\begin{aligned}
\mathbf{f}_1 | \mathbf{f} \sim \mathcal{N}\Big(&\mu_1 + {\mathbf{K}_1}^{\mathsf{T}}(\mathbf{K}_1 + \mathbf{K}_2)^{-1}\left(\mathbf{f} - \mu_1 - \mu_2\right), \\
&\mathbf{K}_1 - {\mathbf{K}_1}^{\mathsf{T}}(\mathbf{K}_1 + \mathbf{K}_2)^{-1}\mathbf{K}_1\Big).
\end{aligned}$$

1.9 abstract

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

Our approach treats unknown functions nonparametrically using Gaussian processes, which has two important consequences. First, Gaussian processes 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 through a decomposition into additive parts. 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.

1.10 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 1.8). The supplementary material to this paper includes 13 complete reports automatically generated by ABCD.

Good statistical modeling requires not only interpretability but 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. In the remainder of this paper we describe the components of ABCD in detail.

1.11 A language of regression models

The general problem of regression consists of learning a function f mapping from some input space $\mathcal{X}$ to some output space $\mathcal{Y}$. We would like an expressive language which can represent both simple parametric forms of f such as linear, polynomial, etc. and also complex nonparametric functions specified in terms of properties such as smoothness, periodicity, etc. Fortunately, Gaussian processes (GPs) provide a very general and analytically tractable way of capturing both simple and complex functions.

Gaussian processes are distributions over functions such that any finite subset of function evaluations, $(f(x_1), f(x_2), \ldots f(x_N))$, have a joint 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[5]. The composition rules are addition and multiplication:

$$k_1 + k_2 = k_1(x, x') + k_2(x, x') \tag{1.1}$$

$$k_1 \times k_2 = k_1(x, x') \times k_2(x, x') \tag{1.2}$$

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 1.8). Changepoints can be defined through addition and multiplication with sigmoidal functions:

$$\mathrm{CP}(k_1, k_2) = k_1 \times \boldsymbol{\sigma} + k_2 \times \bar{\boldsymbol{\sigma}} \tag{1.3}$$

where $\boldsymbol{\sigma} = \sigma(x)\sigma(x')$ and $\bar{\boldsymbol{\sigma}} = (1 - \sigma(x))(1 - \sigma(x'))$. Changewindows $\mathrm{CW}(\cdot, \cdot)$ can be defined 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 and to extend the number of common regression models included in the language. Table 1.3 lists common regression models that can be expressed by our language.

1.12 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 gra-

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

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

Table 1.3 Common regression models expressible in our language. cos is a special case of our reparametrised Per.

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

$$\mathrm{BIC}(M) = -2 \log p(D \mid M) + p \log n \tag{1.4}$$

where p is the number of kernel parameters, $\log p(D|M)$ 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. MacKay, 2003; Rasmussen and Ghahramani, 2001).

1.13 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 found 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.

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:

$$K \prod_m \mathrm{Lin}^{(m)} \prod_n \boldsymbol{\sigma}^{(n)}, \tag{1.5}$$

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

Sums of kernels are sums of functions Formally, if $f_1(x) \sim \mathrm{GP}(0, k_1)$ and independently $f_2(x) \sim \mathrm{GP}(0, k_2)$ then $f_1(x) + f_2(x) \sim \mathrm{GP}(0, k_1 + k_2)$. 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 in a product as an adjective, or more generally as a modifier of a noun. We now describe how each kernel modifies a model and how this can be described in natural language:

- **Multiplication by SE** removes long range correlations from a model since $\mathrm{SE}(x, x')$ decreases monotonically to 0 as $|x - x'|$ increases. This can be described as making an existing model's correlation structure 'local' or 'approximate'.
- **Multiplication by Lin** is equivalent to multiplying the function being modeled by a linear function. If $f(x) \sim \mathrm{GP}(0, k)$, then $xf(x) \sim \mathrm{GP}(0, k \times \mathrm{Lin})$. This causes the standard deviation of the model to vary linearly without affecting the correlation and can be described as e.g. 'with linearly increasing standard deviation'.
- **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. This can be described as e.g. 'from [time]' or 'until [time]'.
- **Multiplication by Per** modifies the correlation structure in the same way as multiplying the function by an independent periodic function. Formally, if $f_1(x) \sim \mathrm{GP}(0, k_1)$

and $f_2(x) \sim \mathrm{GP}(0, k_2)$ then

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

This can be loosely described as e.g. 'modulated by a periodic function with a period of [period] [units]'.

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 e.g. 'smooth function' for SE. Modifiers corresponding to the other kernels in the product are then appended to this description, forming a noun phrase of the form:

$$\text{Determiner} + \text{Premodifiers} + \text{Noun} + \text{Postmodifiers}$$

As an example, a kernel of the form $\mathrm{SE} \times \mathrm{Per} \times \mathrm{Lin} \times \boldsymbol{\sigma}$ could be described as an

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

where Per has been selected as the head noun.

In principle, any assignment of kernels in a product to these different phrasal roles is possible, but in practice we found certain assignments to produce more interpretable phrases than others. The head noun is chosen according to the following ordering:

$$\mathrm{Per} > \mathrm{WN}, \mathrm{SE}, \mathrm{C} > \prod_m \mathrm{Lin}^{(m)} > \prod_n \boldsymbol{\sigma}^{(n)}$$

i.e. Per is always chosen as the head noun when present.

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.

1.13.1 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 standard deviation'. 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]'.

1.14 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.

1.14.1 Summarizing 400 Years of Solar Activity

We show excerpts from the report automatically generated on annual solar irradiation data from 1610 to 2011 (figure 1.9). 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 et al., 1995). ABCD clearly identifies these two features, as discussed below.

Figure 1.10 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 1.11 and 1.8, respectively. The third component corresponds to long-term trends, as visualized in figure 1.12.

1.14.2 Finding heteroscedasticity in air traffic data

Next, we present the analysis generated by our procedure on international airline passenger data (figure 1.13). The model constructed by ABCD has four components: Lin + SE × Per × Lin + SE + WN × Lin, with descriptions given in figure 1.14.

The second component (figure 1.15) is accurately described as approximately (SE) periodic (Per) with linearly growing amplitude (Lin). By multiplying a white noise kernel by a linear kernel, the model is able to express heteroscedasticity (figure 1.16).

1.14.3 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.

1.15 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 (2012) and Lloyd (2013).

Bing et al. (2010); Diosan et al. (2007) 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 which 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 $\sum \mathrm{SE} \times \cos$. 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 1.3).

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 et al., 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 1.16 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 et al., 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).

1.16 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 et al., 2004), change point modeling (e.g. Fox and Dunson, 2013; Garnett et al., 2010; Saatçi et al., 2010), spectral mixture kernels (Wilson and Adams, 2013) (spectral kernels) and trend-cyclical-irregular models (e.g. Lind et al., 2006).

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 1.3) so we perform inference using the same search methodology and selection criterion[6] 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 et al., 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 1.17 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).

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

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.

1.17 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.

Source Code Source code to perform all experiments is available on github[7].

1.18 Kernels

1.18.1 Base kernels

For scalar-valued inputs, the white noise (WN), constant (C), linear (Lin), squared exponential (SE), and periodic kernels (Per) are defined as follows:

$$\mathrm{WN}(x, x') = \sigma^2 \delta_{x,x'} \tag{1.6}$$

$$\mathrm{C}(x, x') = \sigma^2 \tag{1.7}$$

$$\mathrm{Lin}(x, x') = \sigma^2 (x - \ell)(x' - \ell) \tag{1.8}$$

$$\mathrm{SE}(x, x') = \sigma^2 \exp\left(-\tfrac{(x-x')^2}{2\ell^2}\right) \tag{1.9}$$

$$\mathrm{Per}(x, x') = \sigma^2 \frac{\exp\left(\frac{\cos\frac{2\pi(x-x')}{p}}{\ell^2}\right) - I_0\left(\frac{1}{\ell^2}\right)}{\exp\left(\frac{1}{\ell^2}\right) - I_0\left(\frac{1}{\ell^2}\right)} \tag{1.10}$$

[7] 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/.

where $\delta_{x,x'}$ is the Kronecker delta function, I_0 is the modified Bessel function of the first kind of order zero and other symbols are parameters of the kernel functions.

1.18.2 Changepoints and changewindows

The changepoint, $\mathrm{CP}(\cdot,\cdot)$ operator is defined as follows:

$$\begin{aligned}\mathrm{CP}(k_1,k_2)(x,x') = \quad & \sigma(x)k_1(x,x')\sigma(x') \\ & +(1-\sigma(x))k_2(x,x')(1-\sigma(x'))\end{aligned} \tag{1.11}$$

where $\sigma(x) = 0.5 \times (1 + \tanh(\frac{\ell - x}{s}))$. This can also be written as

$$\mathrm{CP}(k_1,k_2) = \boldsymbol{\sigma} k_1 + \bar{\boldsymbol{\sigma}} k_2 \tag{1.12}$$

where $\boldsymbol{\sigma}(x,x') = \sigma(x)\sigma(x')$ and $\bar{\boldsymbol{\sigma}}(x,x') = (1-\sigma(x))(1-\sigma(x'))$.

Changewindow, $\mathrm{CW}(\cdot,\cdot)$, operators are defined similarly by replacing the sigmoid, $\sigma(x)$, with a product of two sigmoids.

1.18.3 Properties of the periodic kernel

A simple application of l'Hôpital's rule shows that

$$\mathrm{Per}(x,x') \to \sigma^2 \cos\left(\frac{2\pi(x-x')}{p}\right) \quad \text{as} \quad \ell \to \infty. \tag{1.13}$$

This limiting form is written as the cosine kernel (cos).

1.19 Model construction / search

1.19.1 Overview

The model construction phase of ABCD starts with the kernel equal to the noise kernel, WN. New kernel expressions are generated by applying search operators to the current kernel. When new base kernels are proposed by the search operators, their parameters are randomly initialised with several restarts. Parameters are then optimized by conjugate gradients to maximise the likelihood of the data conditioned on the kernel parameters. The kernels are then scored by the Bayesian information criterion and the

top scoring kernel is selected as the new kernel. The search then proceeds by applying the search operators to the new kernel i.e. this is a greedy search algorithm.

In all experiments, 10 random restarts were used for parameter initialisation and the search was run to a depth of 10.

1.19.2 Search operators

ABCD is based on a search algorithm which used the following search operators

$$\mathcal{S} \rightarrow \mathcal{S} + \mathcal{B} \tag{1.14}$$
$$\mathcal{S} \rightarrow \mathcal{S} \times \mathcal{B} \tag{1.15}$$
$$\mathcal{B} \rightarrow \mathcal{B}' \tag{1.16}$$

where $\mathcal{S}$ represents any kernel subexpression and $\mathcal{B}$ is any base kernel within a kernel expression i.e. the search operators represent addition, multiplication and replacement.

To accommodate changepoint/window operators we introduce the following additional operators

$$\mathcal{S} \rightarrow \mathrm{CP}(\mathcal{S}, \mathcal{S}) \tag{1.17}$$
$$\mathcal{S} \rightarrow \mathrm{CW}(\mathcal{S}, \mathcal{S}) \tag{1.18}$$
$$\mathcal{S} \rightarrow \mathrm{CW}(\mathcal{S}, \mathrm{C}) \tag{1.19}$$
$$\mathcal{S} \rightarrow \mathrm{CW}(\mathrm{C}, \mathcal{S}) \tag{1.20}$$

where C is the constant kernel. The last two operators result in a kernel only applying outside or within a certain region.

Based on experience with typical paths followed by the search algorithm we introduced the following operators

$$\mathcal{S} \rightarrow \mathcal{S} \times (\mathcal{B} + \mathrm{C}) \tag{1.21}$$
$$\mathcal{S} \rightarrow \mathcal{B} \tag{1.22}$$
$$\mathcal{S} + \mathcal{S}' \rightarrow \mathcal{S} \tag{1.23}$$
$$\mathcal{S} \times \mathcal{S}' \rightarrow \mathcal{S} \tag{1.24}$$

where $\mathcal{S}'$ represents any other kernel expression. Their introduction is currently not rigorously justified.

1.20 Predictive accuracy

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. We trained each algorithm on the training half of the data, produced predictions for the remaining half and then computed the root mean squared error (RMSE). The values of the RMSEs are then standardised by dividing by the smallest RMSE for each data set i.e. the best performance on each data set will have a value of 1.

Figure 1.18 shows the standardised RMSEs for the different algorithms. The box plots show that all quartiles of the distribution of standardised RMSEs are lower for both versions of ABCD. The median for ABCD-accuracy is 1; it is the best performing algorithm on 7 datasets. The largest outliers of ABCD and spectral kernels are similar in value.

Changepoints performs slightly worse than MKL despite being strictly more general than Changepoints. The introduction of changepoints allows for more structured models, but it introduces parametric forms into the regression models (i.e. the sigmoids expressing the changepoints). This results in worse interpolations at the locations of the change points, suggesting that a more robust modeling language would require a more flexible class of changepoint shapes or improved inference (e.g. fully Bayesian inference over the location and shape of the changepoint).

Eureqa is not suited to this task and performs poorly. The models learned by Eureqa tend to capture only broad trends of the data since the fine details are not well explained by parametric forms.

1.20.1 Tabels of standardised RMSEs

See table 1.4 for raw interpolation results and table 1.5 for raw extrapolation results. The rows follow the order of the datasets in the rest of the supplementary material. The following abbreviations are used: ABCD-accuracy (ABCD-acc), ABCD-interpretability ((ABCD-int), Spectral kernels (SP), Trend-cyclical-irregular (TCI), Bayesian MKL (MKL), Eureqa (EL), Changepoints (CP), Squared exponential (SE) and Linear regression (Lin).

1.21 Guide to the automatically generated reports

Additional supplementary material to this paper is 13 reports automatically generated by ABCD. A link to these reports will be maintained at `http://mlg.eng.cam.ac.uk/lloyd/`.

ABCD-acc	ABCD-int	SP	TCI	MKL	EL	CP	SE	Lin
1.04	1.00	2.09	1.32	3.20	5.30	3.25	4.87	5.01
1.00	1.27	1.09	1.50	1.50	3.22	1.75	2.75	3.26
1.00	1.00	1.09	1.00	2.69	26.20	2.69	7.93	10.74
1.09	1.04	1.00	1.00	1.00	1.59	1.37	1.33	1.55
1.00	1.06	1.08	1.06	1.01	1.49	1.01	1.07	1.58
1.50	1.00	2.19	1.37	2.09	7.88	2.23	6.19	7.36
1.55	1.50	1.02	1.00	1.00	2.40	1.52	1.22	6.28
1.00	1.30	1.26	1.24	1.49	2.43	1.49	2.30	3.20
1.00	1.09	1.08	1.06	1.30	2.84	1.29	2.81	3.79
1.08	1.00	1.15	1.19	1.23	42.56	1.38	1.45	2.70
1.13	1.00	1.42	1.05	2.44	3.29	2.96	2.97	3.40
1.00	1.15	1.76	1.20	1.79	1.93	1.79	1.81	1.87
1.00	1.10	1.03	1.03	1.03	2.24	1.02	1.77	9.97

Table 1.4 Interpolation standardised RMSEs

We recommend that you read the report for '01-airline' first and review the reports that follow afterwards more briefly. '02-solar' is discussed in the main text. '03-mauna' analyses a dataset mentioned in the related work. '04-wheat' demonstrates changepoints being used to capture heteroscedasticity. '05-temperature' extracts an exactly periodic pattern from noisy data. '07-call-centre' demonstrates a large discontinuity being modeled by a changepoint. '10-sulphuric' combines many changepoints to create a highly structured model of the data. '12-births' discovers multiple periodic components.

1.22 Ingredients of an automatic statistician

Gelman (2013) asks "How can an artificial intelligence do statistics? ... It needs not just an inference engine, but also a way to construct new models and a way to check models. Currently, those steps are performed by humans, but the AI would have to do it itself."

In this section, we discuss in more detail the elements we believe are required to build an artificial intelligence that can do statistics.

1. An open-ended language of models Many statistical procedures consider all models in a class of fixed size - for example, graphical model construction algorithms [(1)] search over connectivity graphs for a given set of nodes. While these methods can be powerful, human statisticians are capable of deriving novel model classes when required by the modelling task. An automatic search through an open-ended class of models can

(1) Cite

ABCD-acc	ABCD-int	SP	TCI	MKL	EL	CP	SE	Lin
1.14	2.10	1.00	1.44	4.73	3.24	4.80	32.21	4.94
1.00	1.26	1.21	1.03	1.00	2.64	1.03	1.61	1.07
1.40	1.00	1.32	1.29	1.74	2.54	1.74	1.85	3.19
1.07	1.18	3.00	3.00	3.00	1.31	1.00	3.03	1.02
1.00	1.00	1.03	1.00	1.35	1.28	1.35	2.72	1.51
1.00	2.03	3.38	2.14	4.09	6.26	4.17	4.13	4.93
2.98	1.00	11.04	1.80	1.80	493.30	3.54	22.63	28.76
3.10	1.88	1.00	2.31	3.13	1.41	3.13	8.46	4.31
1.00	2.05	1.61	1.52	2.90	2.73	3.14	2.85	2.64
1.00	1.45	1.43	1.80	1.61	1.97	2.25	1.08	3.52
2.16	2.03	3.57	2.23	1.71	2.23	1.66	1.89	1.00
1.06	1.00	1.54	1.56	1.85	1.93	1.84	1.66	1.96
3.03	4.00	3.63	3.12	3.16	1.00	5.83	5.35	4.25

Table 1.5 Extrapolation standardised RMSEs

achieve some of this flexibility, growing the complexity of the model to fit the task at hand, and possibly combining existing structures in novel ways.

2. Searching through model space An open-ended space of models cannot be searched exhaustively. Just as human researchers iteratively refine their models, search procedures can propose new search directions based on the results of previous model fits. Because any search in an open-ended space must start with relatively simple models before moving on to more complex ones, any model search in an open-ended space will likely resemble a model-building procedure.

L: Two sections per- - or are they com- y intertwined?

3. Model comparison and checking model fit (2) An automatic statistician should be able to question the models it has constructed, and formal procedures from model checking provide a way for it to do this. Gelman and Shalizi (2012) review the literature on model checking. (!) In this work, we use approximate marginal likelihood to compare models, penalizing complexity using the Bayesian Information Criterion as a heuristic.

4. Describing models Part of the value of statistical models comes from enabling humans to understand a dataset or a phenomenon. Furthermore, a clear description of the statistical structure found in a dataset helps a user to notice when the dataset has errors, the wrong question was asked, the model-building procedure failed to capture known structure, a relevant piece of data or constraint is missing, or when a novel

statistical structure has been found.

In this work, we demonstrate that the properties of Gaussian processes allow for a modular description generation procedure. Whether or not such modularity and interpretability is present in other open-ended model classes is an open question.

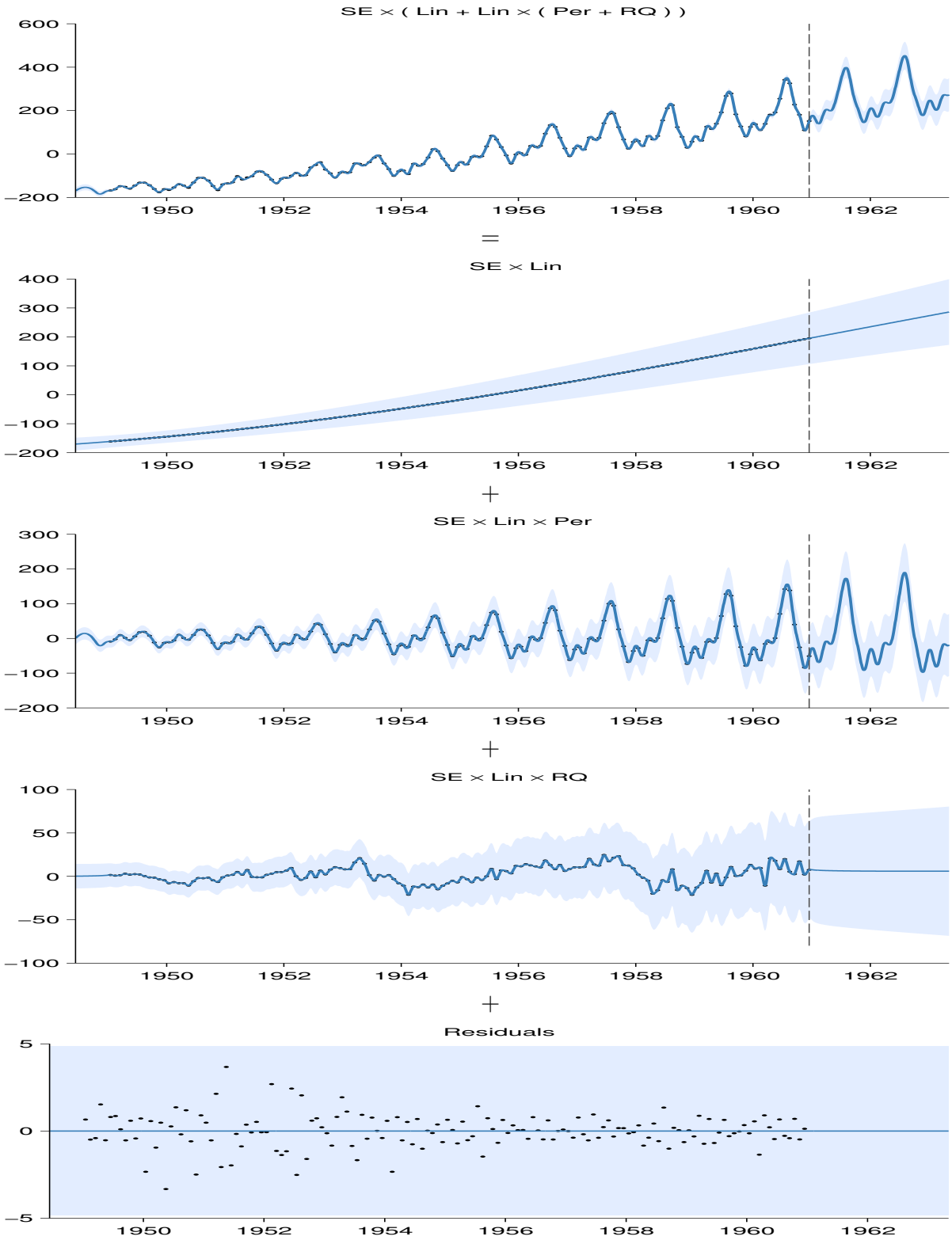

Fig. 1.6 First row: The airline dataset and posterior after a search of depth 10. Subsequent rows: Additive decomposition of posterior into long-term smooth trend, yearly variation, and short-term deviations. Due to the linear kernel, the marginal variance grows over time, making this a heteroskedastic model.

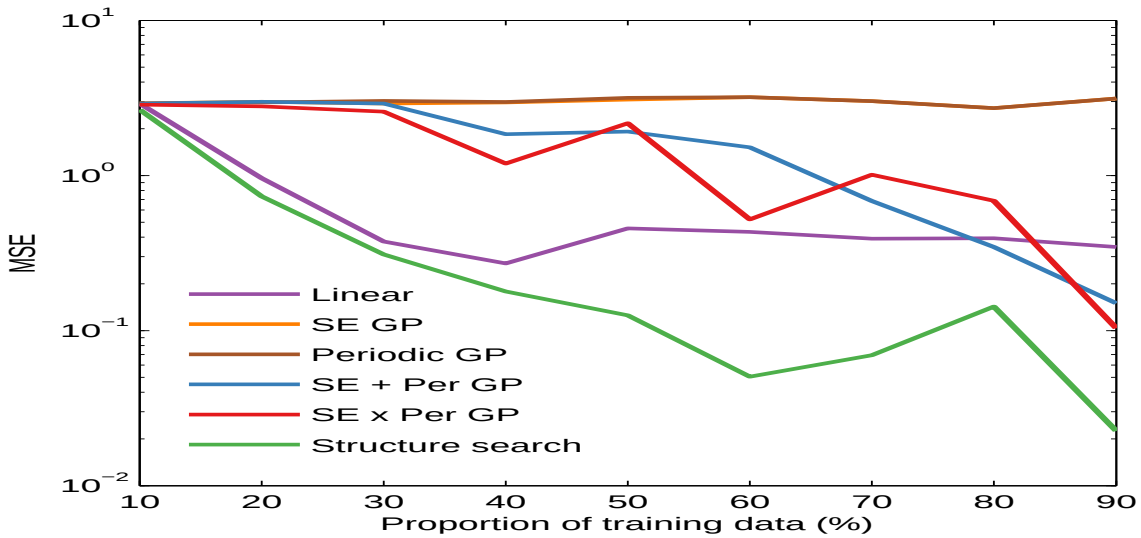

Fig. 1.7 Extrapolation performance on the airline dataset. We plot test-set MSE as a function of the fraction of the dataset used for training.

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.

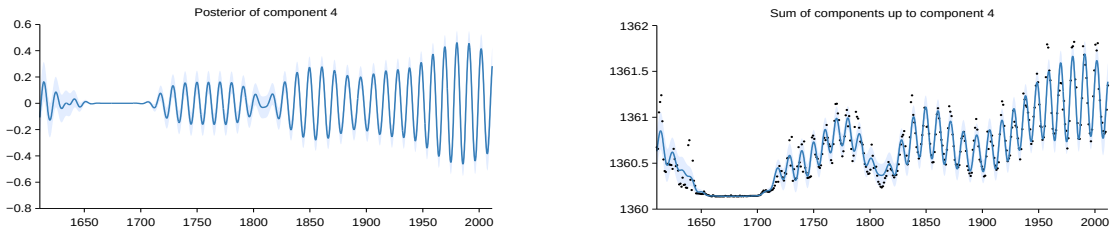

Figure 8: Pointwise posterior of component 4 (left) and the posterior of the cumulative sum of components with data (right)

Fig. 1.8 Extract from an automatically-generated report describing the model components discovered by automatic model search. 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 et al., 1995).

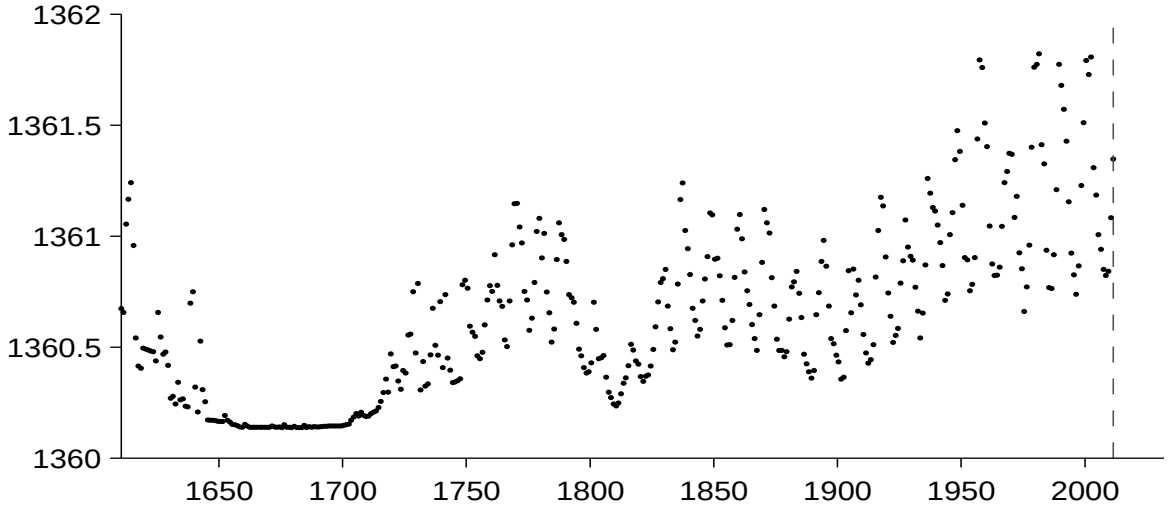

Fig. 1.9 Solar irradiance data.

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.

Fig. 1.10 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.

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.

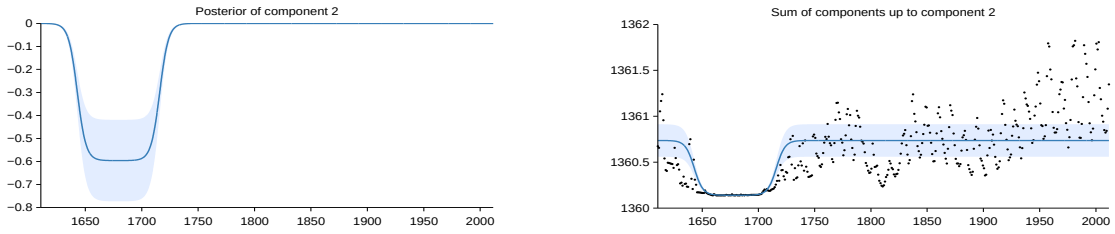

Figure 4: Pointwise posterior of component 2 (left) and the posterior of the cumulative sum of components with data (right)

Fig. 1.11 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.

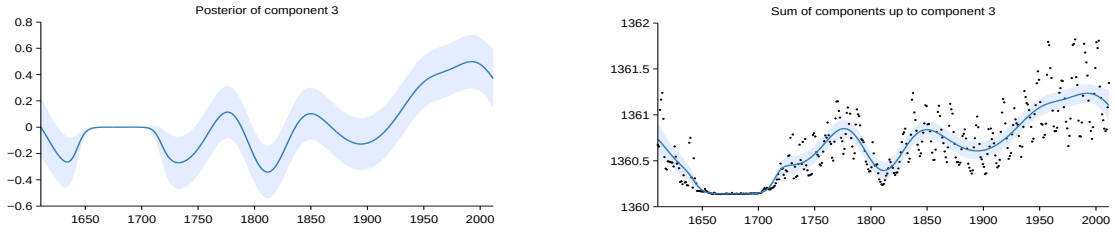

Figure 6: Pointwise posterior of component 3 (left) and the posterior of the cumulative sum of components with data (right)

Fig. 1.12 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.

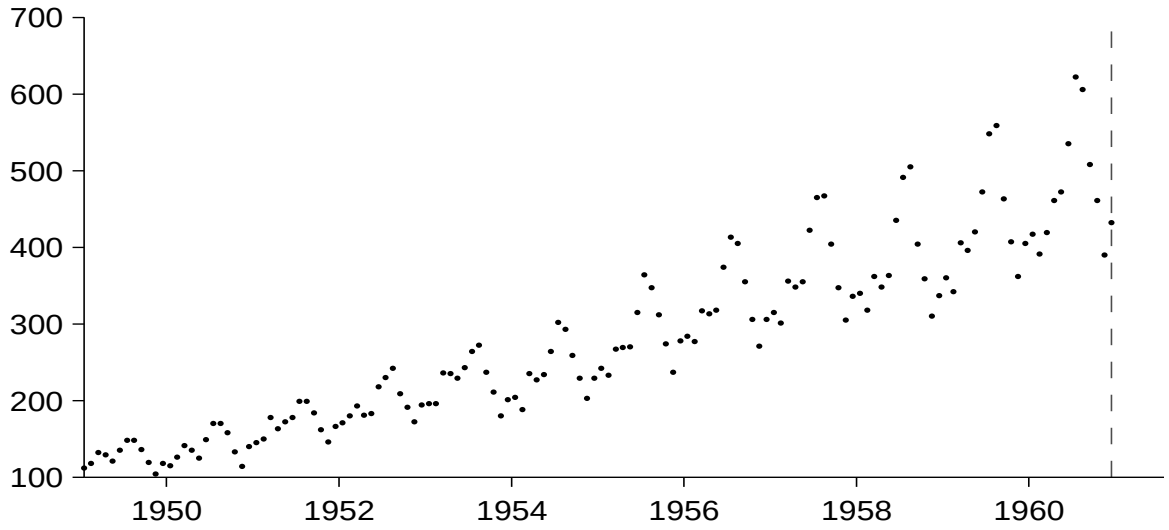

Fig. 1.13 International airline passenger monthly volume (e.g. Box et al., 2013).

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

Fig. 1.14 Short descriptions and summary statistics for the four components of the airline model.

2.2 Component 2 : An approximately periodic function with a period of 1.0 years and with linearly increasing amplitude

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.

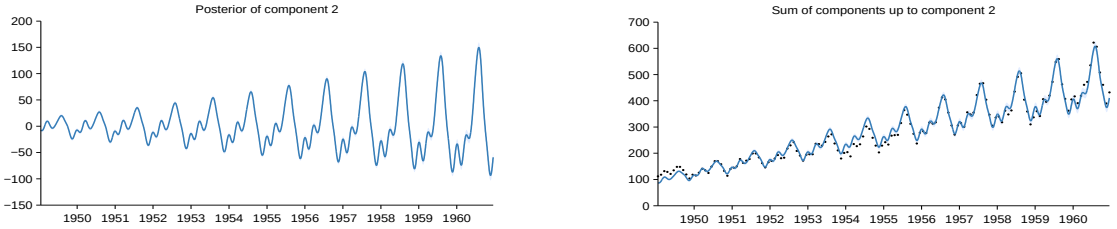

Figure 4: Pointwise posterior of component 2 (left) and the posterior of the cumulative sum of components with data (right)

Fig. 1.15 Capturing non-stationary periodicity in the airline data

2.4 Component 4 : Uncorrelated noise with linearly increasing standard deviation

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.

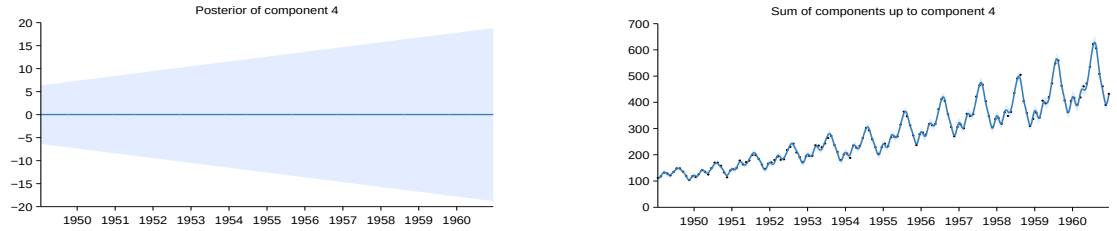

Figure 8: Pointwise posterior of component 4 (left) and the posterior of the cumulative sum of components with data (right)

Fig. 1.16 Modeling heteroscedasticity in the airline dataset.

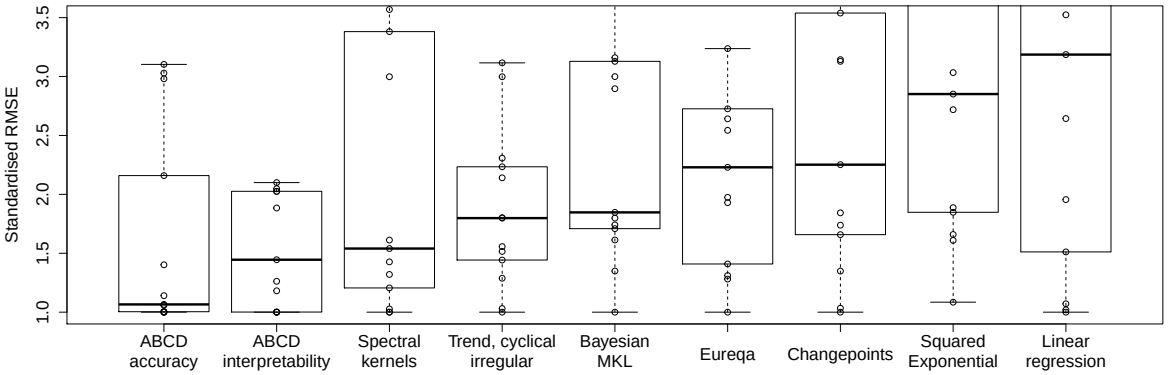

Fig. 1.17 Raw data, and box plot (showing median and quartiles) of standardised extrapolation RMSE (best performance = 1) on 13 time-series. The methods are ordered by median.

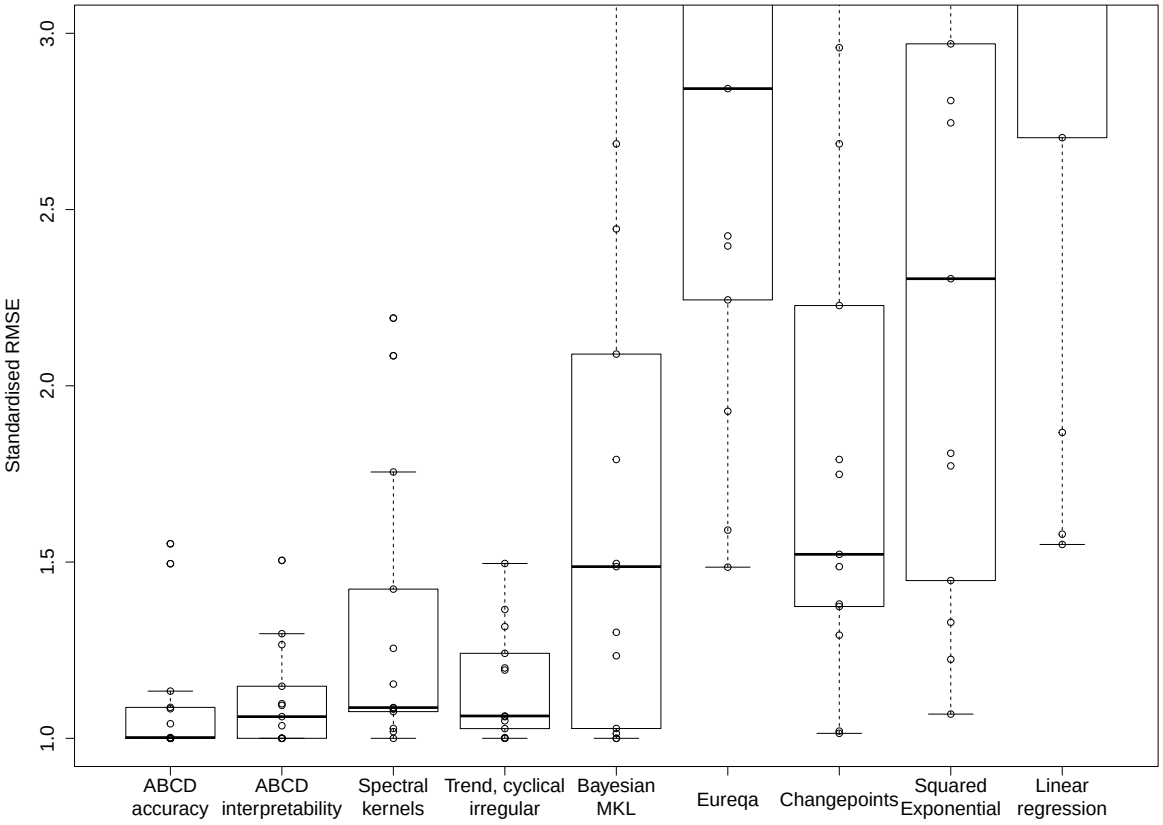

Fig. 1.18 Box plot of standardised RMSE (best performance = 1) on 13 interpolation tasks.

References

F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Advances in Neural Information Processing Systems*, pages 105–112. 2009.

Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM, 2004.

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

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

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

George EP Box, Gwilym M Jenkins, and Gregory C Reinsel. *Time series analysis: forecasting and control*. Wiley. com, 2013.

G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. *Time series analysis: forecasting and control*. 1976.

M. Christoudias, R. Urtasun, and T. Darrell. Bayesian localized multiple kernel learning. *Technical report, EECS Department, University of California, Berkeley*, 2009.

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

David Duvenaud, Hannes Nickisch, and Carl Edward Rasmussen. Additive Gaussian processes. In *Advances in Neural Information Processing Systems 24*, pages 226–234, Granada, Spain, 2011.

David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Structure discovery in nonparametric regression through compositional kernel search. In *Proceedings of the 30th International Conference on Machine Learning*, June 2013.

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

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

Roman Garnett, Michael A Osborne, Steven Reece, Alex Rogers, and Stephen J Roberts. Sequential bayesian prediction in the presence of changepoints and faults. *The Computer Journal*, 53(9):1430–1446, 2010.

Andrew Gelman. Why waste time philosophizing?, 2013. URL `http://andrewgelman.com/2013/02/11/why-waste-time-philosophizing/`.

Andrew Gelman and Cosma Rohilla Shalizi. Philosophy and the practice of bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 2012.

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

C. Gu. *Smoothing spline ANOVA models*. Springer Verlag, 2002. ISBN 0387953531.

Rob J. Hyndman. Time series data library, Accessed summer 2013. URL `http://data.is/TSDLdemo`.

E. T. Jaynes. Highly informative priors. In *Proceedings of the Second International Meeting on Bayesian Statistics*, 1985.

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

Edgar Klenske. *Nonparametric System Identification and Control for Periodic Error Correction in Telescopes*. PhD thesis, University of Stuttgart, 2012.

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

N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, 2005.

Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum gaussian process regression. *The Journal of Machine Learning Research*, 99:1865–1881, 2010.

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

Douglas A Lind, William G Marchal, Samuel Adam Wathen, and Business Week Magazine. *Basic statistics for business and economics*. McGraw-Hill/Irwin Boston, 2006.

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

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Nutonian. Eureqa, 2011. URL http://www.nutonian.com/.

T.A. Plate. Accuracy versus interpretability in flexible modeling: Implementing a tradeoff using Gaussian process models. *Behaviormetrika*, 26:29–50, 1999. ISSN 0385-7417.

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

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

D. Ruppert, M.P. Wand, and R.J. Carroll. *Semiparametric regression*, volume 12. Cambridge University Press, 2003.

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

R. Salakhutdinov and G. Hinton. Using deep belief nets to learn covariance kernels for Gaussian processes. *Advances in Neural information processing systems*, 20:1249–1256, 2008.

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

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

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

G. Wahba. *Spline models for observational data*. Society for Industrial Mathematics, 1990. ISBN 0898712440.

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

Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian process covariance kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning*, June 2013.