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

In section 1.10, we saw that the choice of kernel determines the type of structure that can be learnt by a GP model, and that a wide variety of models could be constructed through simply adding and multiplying a few base kernels together. We didn't answer the question, however, of how to tell which kernel to use for a given problem. Even for experts, choosing the kernel in nonparametric regression remains a black art.

In this chapter, we'll automate the process of building kernels for GP models. To do so, all we need to do is define an open-ended space of kernels, which we can do by simply adding and multiplying together simple kernels from a fixed set. We can then simply search over this space to find a kernel which captures as much structure in the data as possible.

Searching over such a large, structured model class has two benefits. First, this method has very good predictive accuracy, since it effectively tries out a huge number of different regression models. Second, we end up discovering interpretable structure in our dataset. Because GP posteriors can easily be decomposed (as in **??**), we can examine the resulting structures graphically. In section 1.10, we'll even show how to automatically generate english-language descriptions of the resulting models.

1.1 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 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.

3. Model comparison and checking model fit 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 chapter, we introduce a system containing all the above ingredients. We call this system the Automatic Bayesian Covariance Discovery (ABCD) system.

1.2 A Language of Regression Models

As shown in Chapter 1.10, 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 × defines a kernel family, whose parameters are the concatenation of the parameters for the base kernel families.

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.

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. The composition rules are addition and multiplication:

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

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

We have found that incorporating changepoints into the language is essential for realistic models of time series (e.g. figure **??**).

Table 1.1 lists common regression models that can be expressed by our language.

Regression model	Kernel
Linear regression	$\mathrm{C} + \mathrm{Lin} + \mathrm{WN}$
Kernel ridge regression	$\mathrm{SE} + \mathrm{WN}$
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 1.1 Common regression models expressible in our language. cos is a special case of our reparametrised Per.

1.3 Model Search

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.

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 $\mathcal{B}'$.

$$\begin{array}{rccc} \text{Replacement} & k_i & \to & k_i' \\ \text{Addition} & k_i & \to & k_i + k_j' \\ \text{Multiplication} & k_i & \to & k_i \times k_j' \end{array}$$

These operators can generate all possible algebraic expressions. To see this, observe that if we restricted the + and × rules to only apply to base kernel families, we would obtain a context-free grammar 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 context-free grammar 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.

Hyperparameter initialization

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 (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.

1.4 Model Evaluation

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.

In a fully Bayesian approach, we would put priors over the parameters and compute the marginal likelihood of the models with all the parameters integrated out. However,

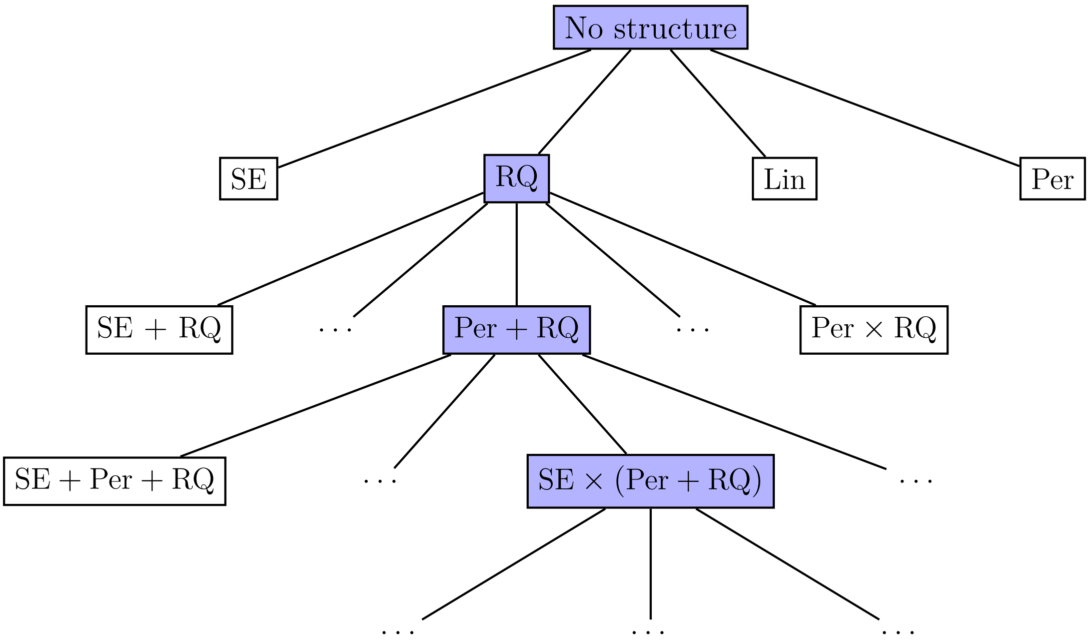


Fig. 1.1 An example of a search tree over kernel expressions. Figure 1.3 shows the model increasing in sophistication as the kernel expression grows.

as this would be difficult to do across our space of models, we approximate this integral by choosing the parameters to optimize the marginal likelihood, and then apply the Bayesian information criterion (BIC) to penalize model complexity.

Of course, other model selection criteria could be used with our search procedure. For instance, cross-validation could be used when the goal is interpolation.

We optimized kernel parameters at each step using conjugate gradients, randomly restarting any newly introduced kernel parameters. To approximate the marginal likelihood of a kernel without integrating over parameters, used the Bayesian Information Criterion (BIC) (Schwarz, 1978). We also experimented with using the Laplace approximation to the marginal likelihood, but this was found to be less numerically stable and was not meaningful in cases when the optimiser failed to reach an optimum. Because BIC is a function of the number of parameters in a model, we adjusted for cases where two parameters were only serving the role of one. e.g. when two kernels are multiplied, one of the variance parameters becomes redundant.

After each model is proposed its kernel parameters are optimised by conjugate gradient descent. We evaluate each optimized model, M, using the Bayesian Information

Criterion (BIC) (Schwarz, 1978):

$$\text{BIC}(M) = -2 \log p(D \,|\, M) + p \log n \tag{1.3}$$

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 against model complexity and implements what is known as "Bayesian Occam's Razor" (MacKay, 2003; Rasmussen and Ghahramani, 2001).

1.5 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 1.10, a GP whose kernel is a sum of kernels can be viewed 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,

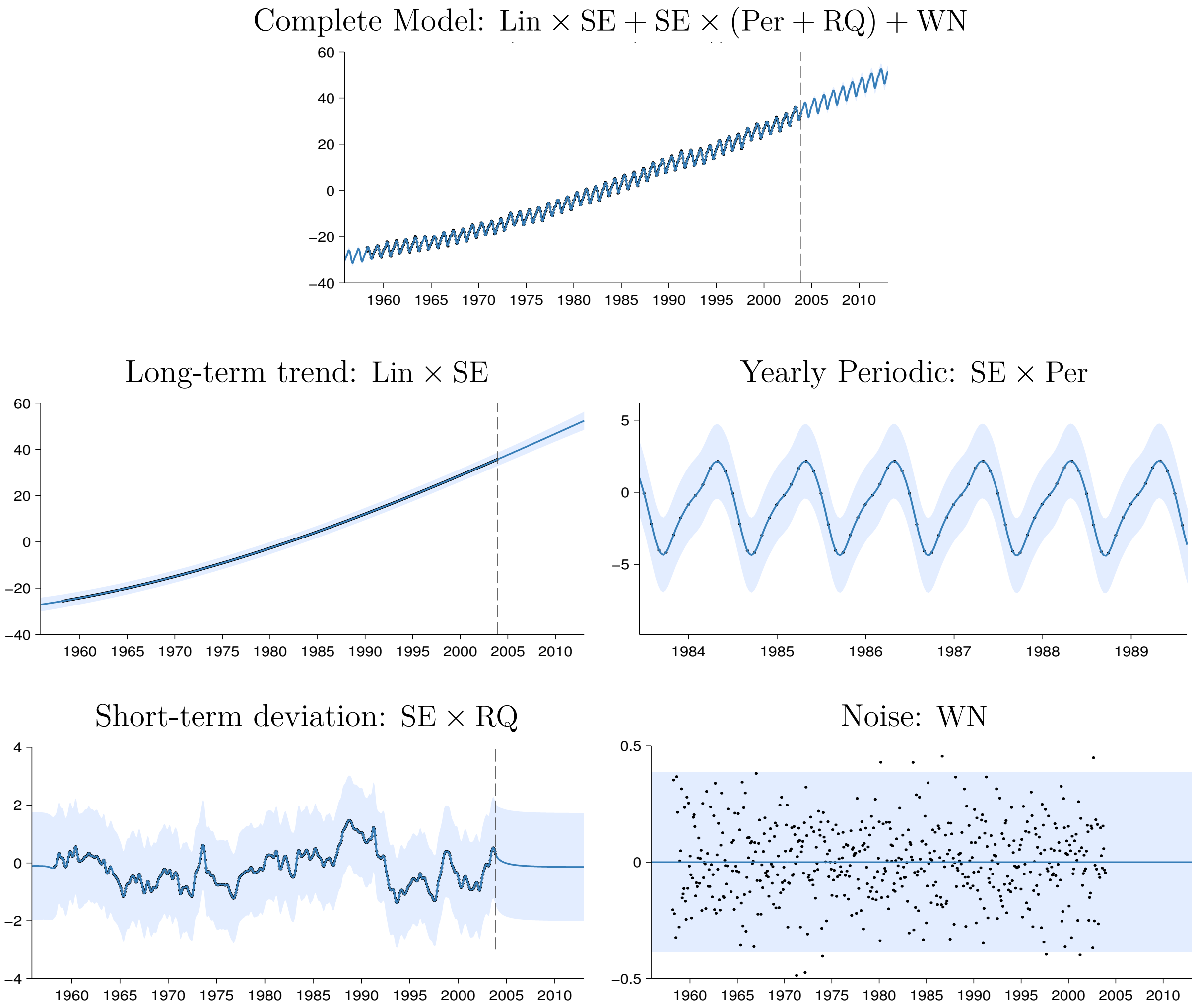
$$\text{SE} \times (\text{RQ} + \text{Lin}) = \text{SE} \times \text{RQ} + \text{SE} \times \text{Lin} \tag{1.4}$$

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 $\mathbf{CO_2}$ 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 1.3 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


Complete Model: Lin × SE + SE × (Per + RQ) + WN

Long-term trend: Lin × SE

Yearly Periodic: SE × Per

Short-term deviation: SE × RQ

Noise: WN

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.

plot the residuals, showing that there is little obvious structure left in the data.

Airline passenger data Figure 1.4 shows the decomposition produced by applying our method to monthly totals of international airline passengers (Box et al., 1976). 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.

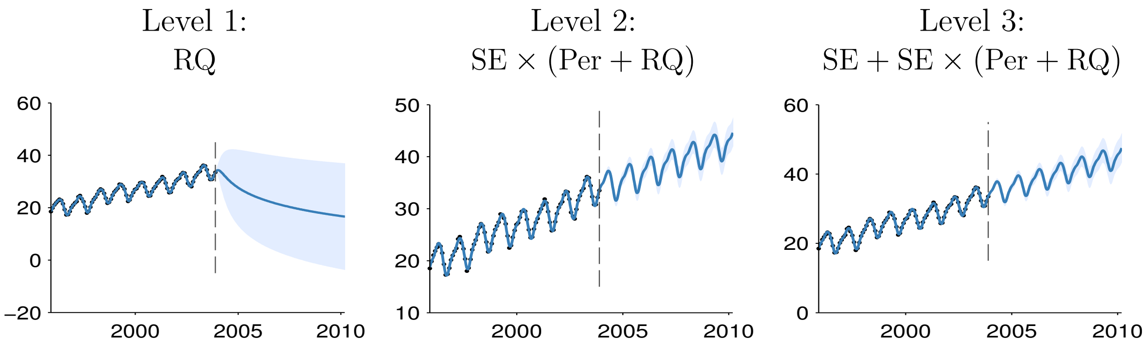


Fig. 1.3 Posterior mean and variance for different depths of kernel search. The dashed line marks the extent of the dataset. In the first column, the function is only modeled as a locally smooth function, and the extrapolation is poor. Next, a periodic component is added, and the extrapolation improves. At depth 3, the kernel can capture most of the relevant structure, and is able to extrapolate reasonably.

1.6 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).

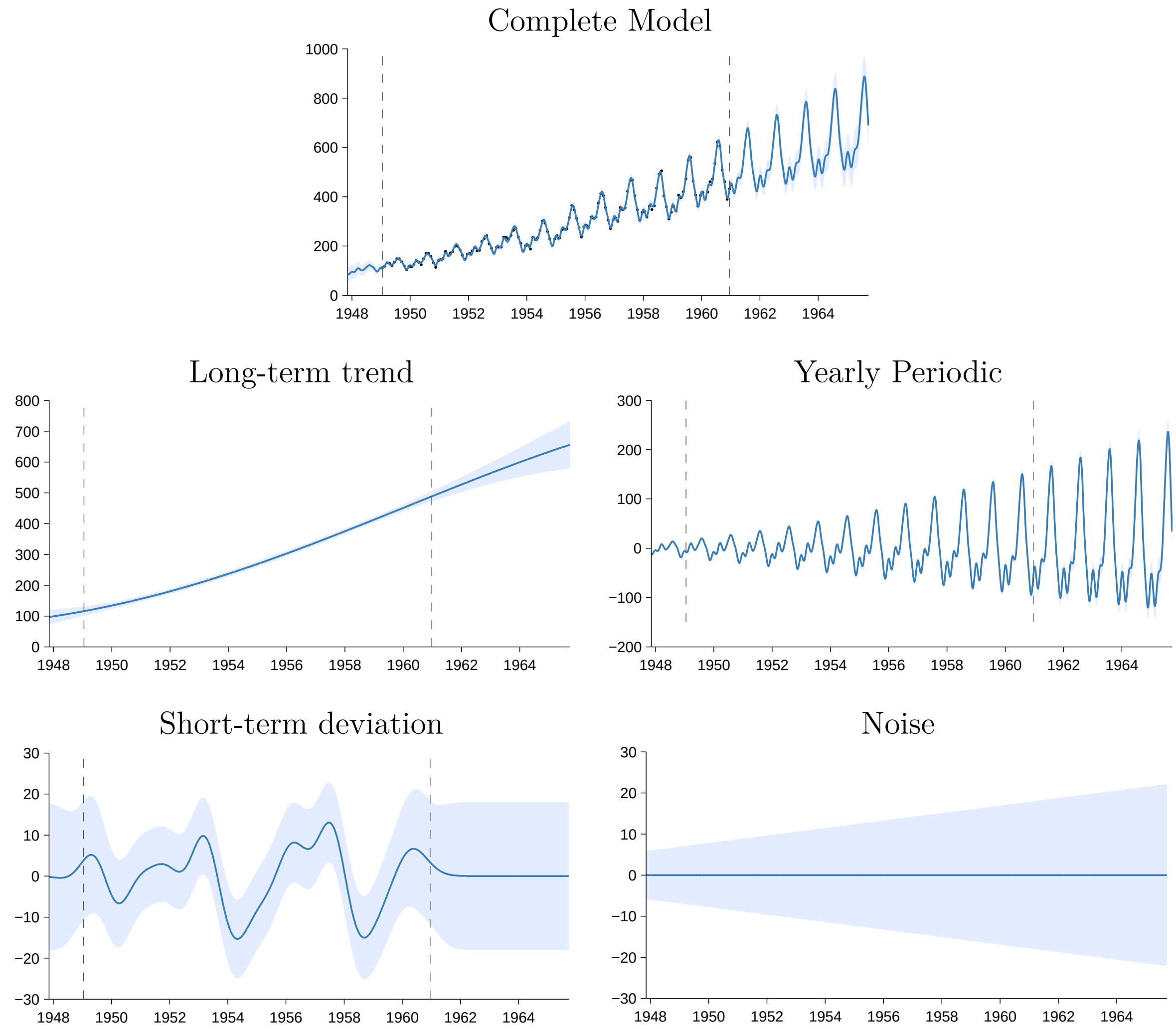


Fig. 1.4 First row: The airline dataset and posterior after a search of depth 10. Subsequent rows: Additive decomposition of posterior into long-term smooth trend, yearly variation, and short-term deviations. Due to the linear kernel, the marginal variance grows over time, making this a heteroskedastic model.

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).

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.

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.

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 $\sum \text{SE} \times \cos$. Both demonstrate, using Bochner's theorem (Bochner, 1959), that these kernels can approximate any stationary covariance function. Our language of kernels includes both of these kernel classes (see table 1.1).

There is a large body of work attempting to construct rich kernels through a weighted sum of base kernels called multiple kernel learning (MKL) (e.g. Bach 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 1.7 which is expressed as a restriction of our language of kernels.

Equation learning

Todorovski and Dzeroski (1997), Washio et al. (1999) and **?** 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.

nonparametric nonstationary covariance learning

TODO: Add a lit review of nonparametric nonstationary covariance learning.

Discovering interpretable structure

Besides allowing faster learning and extrapolation, learning a more structured kernel sometimes has the added benefit of making the resulting model more intepretable. This is a similar motivation as for the use of sparsity-inducing methods: on many real datasets, the signal can be well-predicted by some small subset of the inputs. Identifying this subset allows both better generalization, and a more interpretable model.

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).

1.6.1 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.

1.7 Predictive Accuracy

1.7.1 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.

1.7.2 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.

1.7.3 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 1.1) so we perform inference using the same search methodology and selection criterion[1] 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.

1.7.4 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.

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

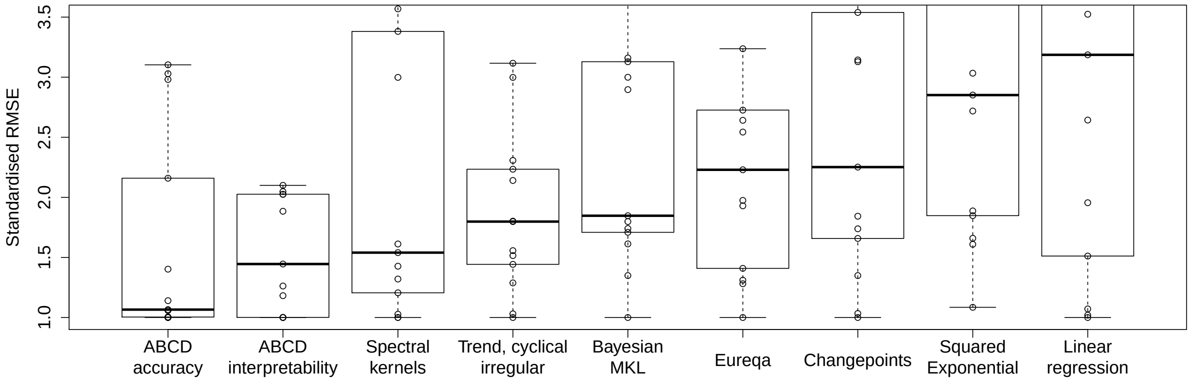


Fig. 1.5 Box plot (showing median and quartiles) of standardised extrapolation RMSE (best performance = 1) on 13 time-series. The methods are ordered by median.

Figure 1.5 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 appendix.

1.8 High-dimensional Prediction

ABCD can also be applied to multidimensional regression problems. 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, comparing 5 methods in terms of MSE and predictive likelihood.

The data sets had dimensionalities ranging from 4 to 13, and the number of data points ranged from 150 to 450. Our structure search was run up to depth 10, using the SE and RQ base kernel families. All GP parameter optimisation was performed by automated calls to the GPML toolbox available at `http://www.gaussianprocess.org/gpml/code/`.

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 with all base kernels was always the best performing method, or the difference in performance was not statiscally significant at the 5% level.

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	concrete	puma	servo	housing	bach	concrete	puma	servo	housing
Linear reg.	1.031	0.404	0.641	0.523	0.289	3.430	1.403	1.881	2.678	1.052
GAM	1.259	0.149	0.598	0.281	0.161	2.708	0.467	1.195	1.800	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**	**0.869**	0.398	**0.843**	1.429	0.207
Additive GP	**0.045**	**0.089**	**0.316**	**0.110**	0.102	**0.869**	**0.114**	**0.841**	1.309	0.194
SE, RQ Search	**0.044**	**0.087**	**0.315**	**0.102**	**0.082**	**0.859**	**0.065**	**0.840**	1.265	**0.059**
SE, RQ, Lin, Per	**0.509**	**0.079**	**0.321**	**0.094**	**0.112**	**1.357**	**0.114**	**0.837**	**0.573**	**0.151**

1.8.1 Validation on Synthetic Data

Because it is difficult to visualize the structures discovered in multiple dimensions, it is difficult to tell from predictive accuracy alone if the search procedure is finding all structure present. To address this questions, 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 signal-to-noise ratios (SNR).

Table 1.3 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 was found.

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

Table 1.3 shows the results. The first column 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 case. 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, rather than overfitting.

1.9 Discussion

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.

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.

We hope that the ABCD algorithm will help replace the current and often opaque art of kernel engineering with a more transparent science of automated kernel construction.

We demonstrate that the properties of Gaussian processes allow for an automatic, modular description generation procedure, through graphs illustrating interpretable decomposition of the posterior.

Whether or not such modularity and interpretability is present in other open-ended model classes is an open question.

1.10 Future work

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.

References

F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Advances in Neural Information Processing Systems*, pages 105–112. 2009. (page 11)

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. (pages 12 and 15)

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. (page 13)

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. (pages 11 and 12)

Salomon Bochner. *Lectures on Fourier integrals*, volume 42. Princeton University Press, 1959. (page 12)

George EP Box, Gwilym M Jenkins, and Gregory C Reinsel. *Time series analysis: forecasting and control*. Wiley. com, 2013. (page 15)

G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. *Time series analysis: forecasting and control*. 1976. (page 8)

M. Christoudias, R. Urtasun, and T. Darrell. Bayesian localized multiple kernel learning. *Technical report, EECS Department, University of California, Berkeley*, 2009. (page 11)

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. (pages 11 and 12)

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. (pages 9 and 16)

E.B. Fox and D.B. Dunson. Multiresolution Gaussian Processes. In *Neural Information Processing Systems 25*. MIT Press, 2013. (page 15)

M. Ganesalingam and W. T. Gowers. A fully automatic problem solver with human-style output. *CoRR*, abs/1309.4501, 2013. (page 13)

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. (page 15)

Andrew Gelman. Why waste time philosophizing?, 2013. URL `http://andrewgelman.com/2013/02/11/why-waste-time-philosophizing/`. (page 2)

Andrew Gelman and Cosma Rohilla Shalizi. Philosophy and the practice of bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 2012. (page 2)

Roger B. Grosse, Ruslan Salakhutdinov, William T. Freeman, and Joshua B. Tenenbaum. Exploiting compositionality to explore a large space of model structures. In *Uncertainty in Artificial Intelligence*, 2012. (pages 11 and 13)

C. Gu. *Smoothing spline ANOVA models*. Springer Verlag, 2002. ISBN 0387953531. (page 9)

Rob J. Hyndman. Time series data library, Accessed summer 2013. URL `http://data.is/TSDLdemo`. (page 14)

E. T. Jaynes. Highly informative priors. In *Proceedings of the Second International Meeting on Bayesian Statistics*, 1985. (page 1)

C. Kemp and J.B. Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Sciences*, 105(31):10687–10692, 2008. (pages 11 and 13)

Edgar Klenske. *Nonparametric System Identification and Control for Periodic Error Correction in Telescopes*. PhD thesis, University of Stuttgart, 2012. (page 12)

Gabriel Kronberger and Michael Kommenda. Evolution of covariance functions for gaussian process regression using genetic programming. *arXiv preprint arXiv:1305.3794*, 2013. (page 12)

N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, 2005. (page 11)

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. (page 12)

Douglas A Lind, William G Marchal, Samuel Adam Wathen, and Business Week Magazine. *Basic statistics for business and economics*. McGraw-Hill/Irwin Boston, 2006. (page 15)

James Robert Lloyd. GEFCom2012 hierarchical load forecasting: Gradient boosting machines and gaussian processes. *International Journal of Forecasting*, 2013. (page 12)

David JC MacKay. *Information theory, inference, and learning algorithms*. Cambridge university press, 2003. (page 7)

Nutonian. Eureqa, 2011. URL `http://www.nutonian.com/`. (pages 13 and 15)

T.A. Plate. Accuracy versus interpretability in flexible modeling: Implementing a tradeoff using Gaussian process models. *Behaviormetrika*, 26:29–50, 1999. ISSN 0385-7417. (page 9)

Carl Edward Rasmussen and Zoubin Ghahramani. Occam's razor. *Advances in neural information processing systems*, pages 294–300, 2001. (pages 5 and 7)

C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*, volume 38. The MIT Press, Cambridge, MA, USA, 2006. (pages 7 and 12)

D. Ruppert, M.P. Wand, and R.J. Carroll. *Semiparametric regression*, volume 12. Cambridge University Press, 2003. (page 10)

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. (page 15)

Ruslan Salakhutdinov and Geoffrey Hinton. Using deep belief nets to learn covariance kernels for Gaussian processes. *Advances in Neural information processing systems*, 20:1249–1256, 2008. (page 11)

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, April 2009. ISSN 1095-9203. doi: 10.1126/science.1165893. (page 11)

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978. (pages 5, 6, and 7)

L. Todorovski and S. Dzeroski. Declarative bias in equation discovery. In *International Conference on Machine Learning*, pages 376–384, 1997. (pages 11 and 12)

G. Wahba. *Spline models for observational data.* Society for Industrial Mathematics, 1990. ISBN 0898712440. (page 9)

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. (pages 11 and 12)

Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian process covariance kernels for pattern discovery and extrapolation. *arXiv: 1302.4245*, June 2013. (pages 12 and 15)