Chapter 1

Automatically Building Structured Covariance Functions

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.

1.2 Expressing structure through kernels

Gaussian process models use a kernel to define the covariance between any two function values: $\text{Cov}(y, y') = k(x, x')$. The kernel specifies which structures are likely under the gp prior, which in turn determines the generalization properties of the model. In this section, we review the ways in which kernel families[1]can be composed to express diverse priors over functions.

There has been significant work on constructing gp kernels and analyzing their properties, summarized in Chapter 4 of Rasmussen and Williams (2006). Commonly used kernels families include the squared exponential (SE), periodic (Per), linear (Lin), and rational quadratic (RQ) (see Figure 1.1 and the appendix).

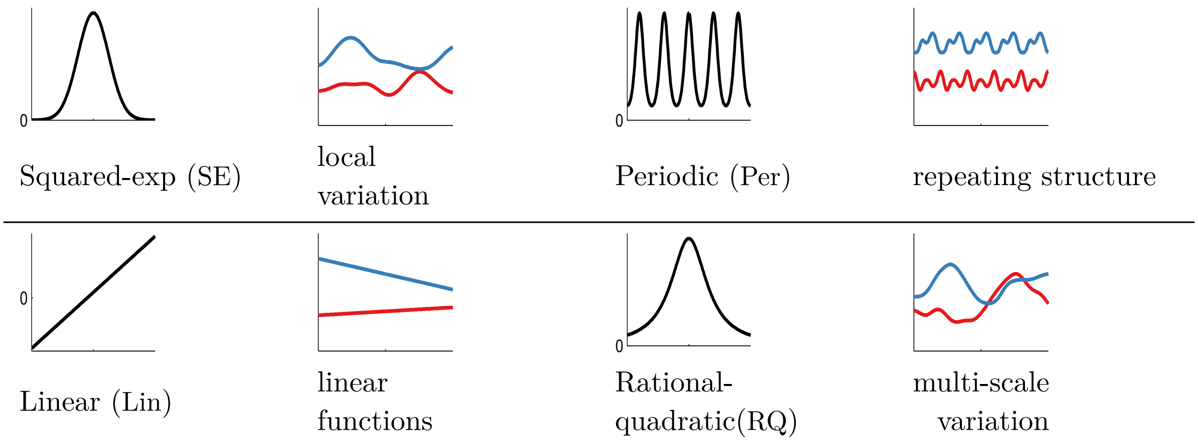

Fig. 1.1 Left and third columns: base kernels $k(\cdot, 0)$. Second and fourth columns: draws from a GP with each repective kernel. The x-axis has the same range on all plots.

Composing Kernels Positive semidefinite kernels (i.e. those which define valid covariance functions) are closed under addition and multiplication. This allows one to create richly structured and interpretable kernels from well understood base components.

All of the base kernels we use are one-dimensional; kernels over multidimensional inputs are constructed by adding and multiplying kernels over individual dimensions. These dimensions are represented using subscripts, e.g. SE_2 represents an SE kernel over the second dimension of x.

Summation By summing kernels, we can model the data as a superposition of independent functions, possibly representing different structures. Suppose functions f_1, f_2 are draw from independent gp priors, $f_1 \sim \mathcal{GP}(\mu_1, k_1)$, $f_2 \sim \mathcal{GP}(\mu_2, k_2)$. Then $f := f_1 + f_2 \sim \mathcal{GP}(\mu$

In time series models, sums of kernels can express superposition of different processes, possibly operating at different scales. In multiple dimensions, summing kernels gives additive structure over different dimensions, similar to generalized additive models (Hastie

[1]When unclear from context, we use 'kernel family' to refer to the parametric forms of the functions given in the appendix. A kernel is a kernel family with all of the parameters specified.

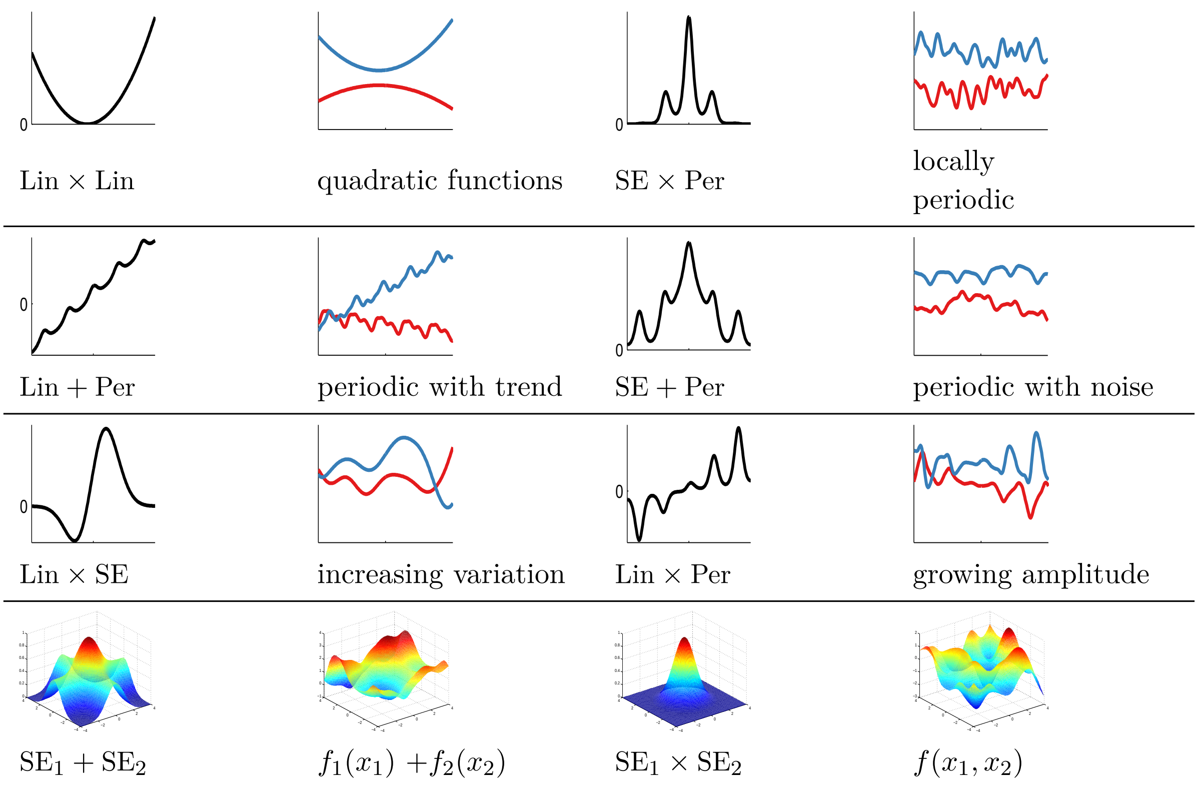

Fig. 1.2 Examples of structures expressible by composite kernels. Left column and third columns: composite kernels $k(\cdot, 0)$. Plots have same meaning as in Figure 1.1.

and Tibshirani, 1990). These two kinds of structure are demonstrated in rows 2 and 4 of figure 1.2, respectively.

Multiplication Multiplying kernels allows us to account for interactions between different input dimensions or different notions of similarity. For instance, in multidimensional data, the multiplicative kernel $SE_1 \times SE_3$ represents a smoothly varying function of dimensions 1 and 3 which is not constrained to be additive. In univariate data, multiplying a kernel by SE gives a way of converting global structure to local structure. For example, Per corresponds to globally periodic structure, whereas Per × SE corresponds to locally periodic structure, as shown in row 1 of figure 1.2.

Many architectures for learning complex functions, such as convolutional networks LeCun et al. (1989) and sum-product networks Poon and Domingos (2011), include units which compute AND-like and OR-like operations. Composite kernels can be viewed in this way too. A sum of kernels can be understood as an OR-like operation: two points are considered similar if either kernel has a high value. Similarly, multiplying kernels is an AND-like operation, since two points are considered similar only if both kernels have high values. Since we are applying these operations to the similarity functions rather than the regression functions themselves, compositions of even a few base kernels are able to capture complex relationships in data which do not have a simple parametric form.

Example expressions In addition to the examples given in Figure 1.2, 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	Lin $\times$ Lin $\times \ldots$
Generalized Fourier decomposition	Per $+$ 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	Lin $+$ SE
Linearly growing amplitude	Lin $\times$ 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.

1.3 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 1.2: 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 1.2.
- 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.

1.4 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 $\text{SE} \times \cos(x - x')$, forming a basis for stationary kernels. These kernels share similarities with $\text{SE} \times \text{Per}$ 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.

1.5 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 2, 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,

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

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 1.2.

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 1.4 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). 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.

1.6 Validation on synthetic data

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.
SE + RQ	1	SE	SE × Per	SE
Lin × Per	1	Lin × Per	Lin × Per	SE
$\text{SE}_1 + \text{RQ}_2$	2	$\text{SE}_1 + \text{SE}_2$	$\text{Lin}_1 + \text{SE}_2$	Lin
$\text{SE}_1 + \text{SE}_2 \times \text{Per}_1 + \text{SE}_3$	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
$\text{SE}_1 \times \text{SE}_2 + \text{SE}_2 \times \text{SE}_3$	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
$(\text{SE}_1 + \text{SE}_2) \times (\text{SE}_3 + \text{SE}_4)$	4	$(\text{SE}_1 + \text{SE}_2) \times (\text{SE}_3 \times \text{Lin}_3 \times \text{Lin}_1 + \text{SE}_4)$	$(\text{SE}_1 + \text{SE}_2) \times \text{SE}_3 \times \text{SE}_4$	-

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 1.1 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.

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 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
gp Additive	**0.045**	**0.089**	**0.316**	**0.110**	0.102	**−0.131**	**0.114**	**0.841**	**0.309**	0.194
Structure Search	**0.044**	**0.087**	**0.315**	**0.102**	**0.082**	**−0.141**	**0.065**	**0.840**	**0.265**	**0.059**

1.7 Quantitative evaluation

In addition to the qualitative evaluation in section 1.5, we investigated quantitatively how our method performs on both extrapolation and interpolation tasks.

1.7.1 Extrapolation

We compared the extrapolation capabilities of our model against standard baselines[2]. 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.

[2]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.

1.7.2 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 [3] 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[4]; Python code to perform all experiments is available on github[5].

1.8 Discussion

"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

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

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

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

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

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.

Acknowledgements

We thank Carl Rasmussen and Andrew G. Wilson for helpful discussions. This work was funded in part by NSERC, EPSRC grant EP/I036575/1, and Google.

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

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

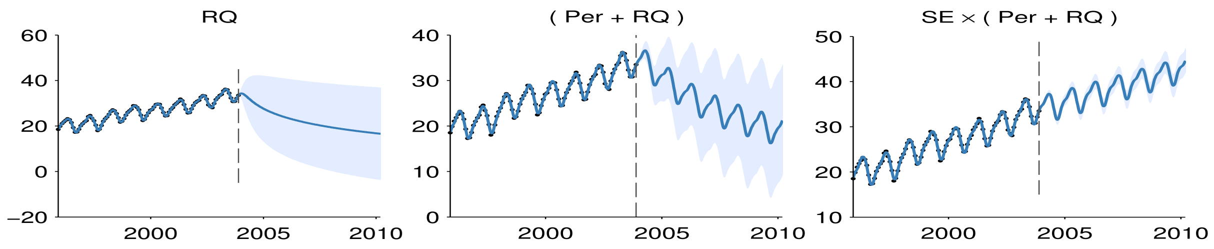

Fig. 1.3 Posterior mean and variance for different depths of kernel search. The dashed line marks the extent of the dataset. In the first column, the function is only modeled as a locally smooth function, and the extrapolation is poor. Next, a periodic component is added, and the extrapolation improves.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

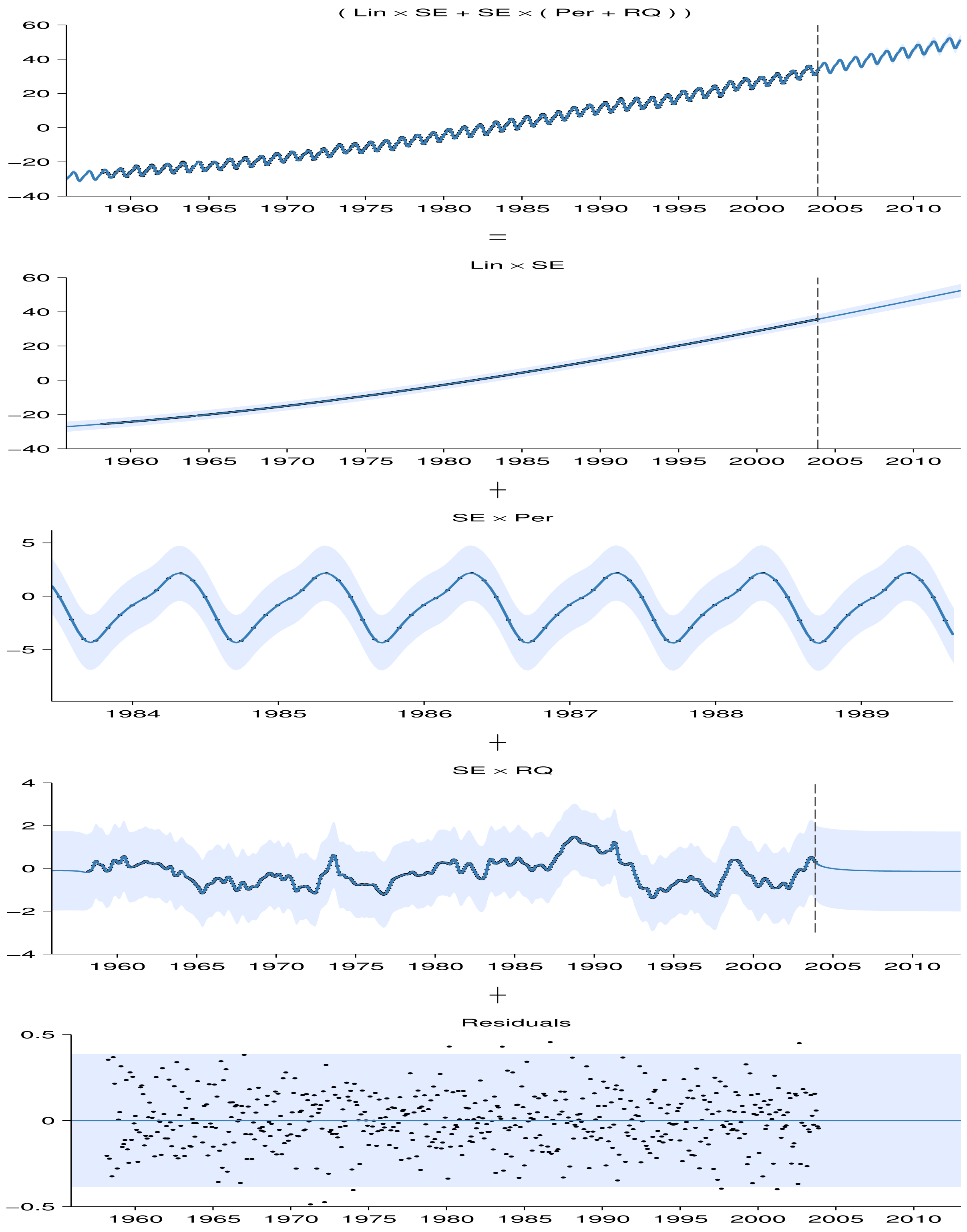

Fig. 1.4 First row: The posterior on the Mauna Loa dataset, after a search of depth 10. Subsequent rows show the automatic decomposition of the time series. The decompositions shows long-term, yearly periodic, medium-term anomaly components, and residuals, respectively. In the third row, the scale has been changed in order to clearly show the yearly periodic structure.

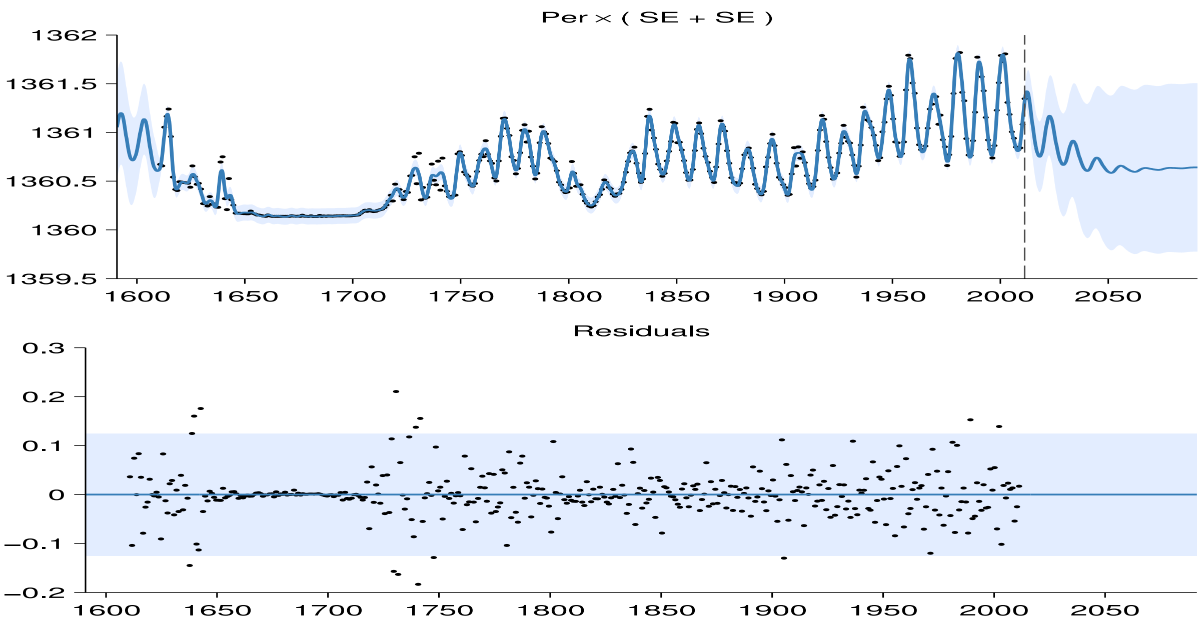

Fig. 1.5 Full posterior and residuals on the solar irradiance dataset.

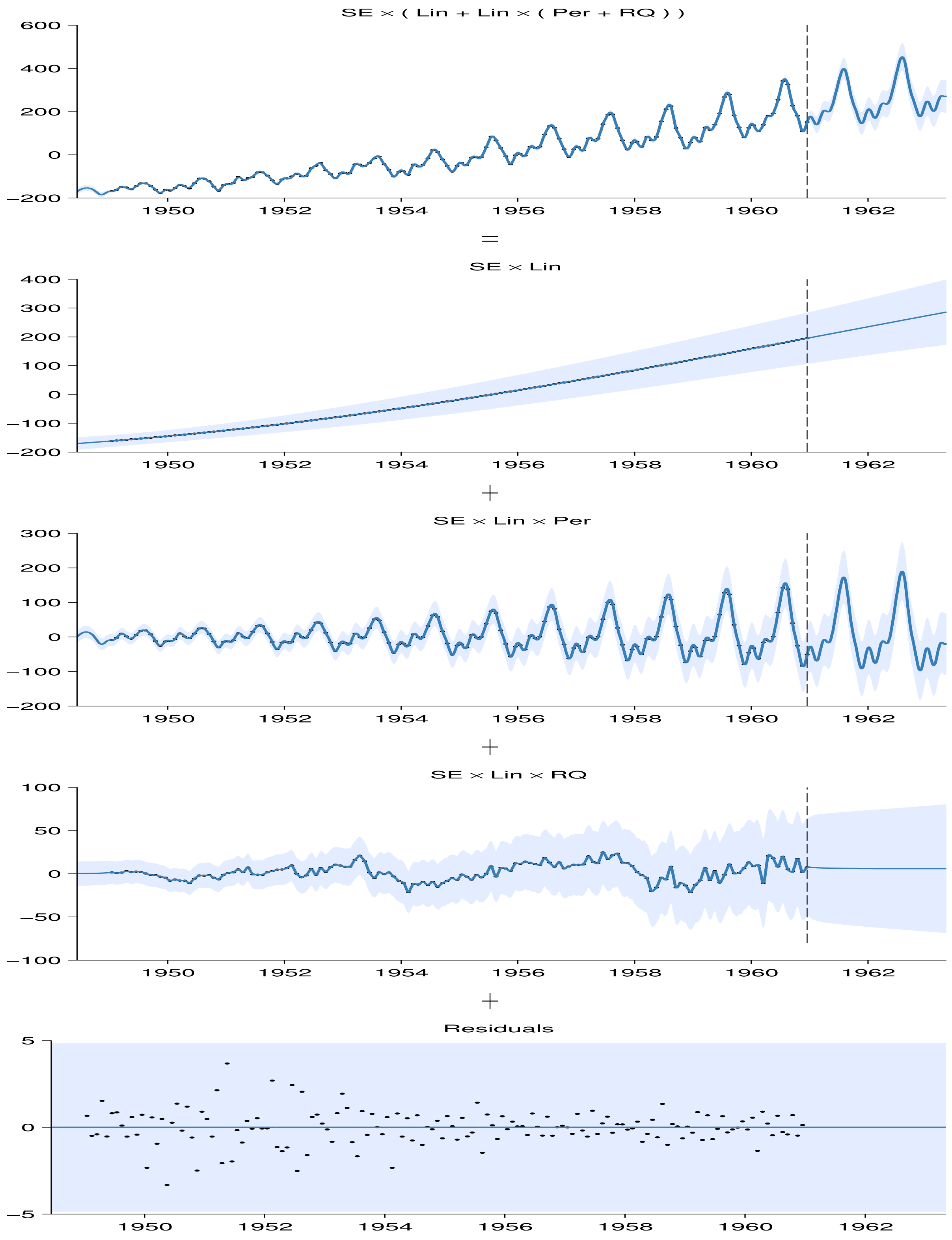

Fig. 1.6 First row: The airline dataset and posterior after a search of depth 10. Subsequent rows: Additive decomposition of posterior into long-term smooth trend, yearly variation, and short-term deviations. Due to the linear kernel, the marginal variance grows over time, making this a heteroskedastic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

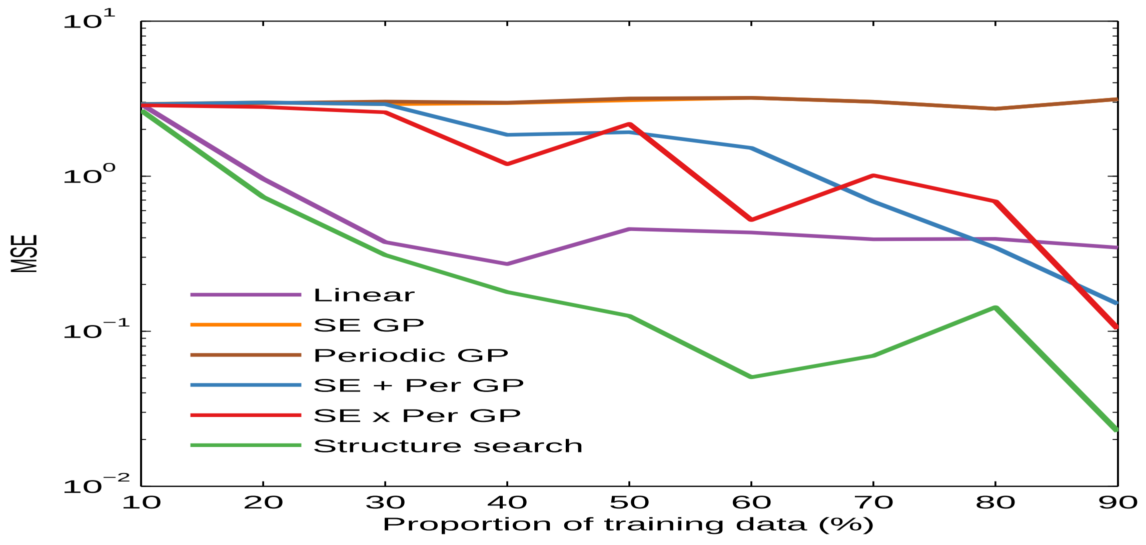

Fig. 1.7 Extrapolation performance on the airline dataset. We plot test-set MSE as a function of the fraction of the dataset used for training.

References

F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Advances in Neural Information Processing Systems*, pages 105–112. 2009. (pages 1 and 7)

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. (page 8)

G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. *Time series analysis: forecasting and control*. 1976. (page 9)

M. Christoudias, R. Urtasun, and T. Darrell. Bayesian localized multiple kernel learning. *Technical report, EECS Department, University of California, Berkeley*, 2009. (page 7)

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. (page 8)

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 7 and 12)

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. (pages 2 and 8)

C. Gu. *Smoothing spline ANOVA models*. Springer Verlag, 2002. ISBN 0387953531. (page 7)

T.J. Hastie and R.J. Tibshirani. *Generalized additive models*. Chapman & Hall/CRC, 1990. (page 3)

E. T. Jaynes. Highly informative priors. In *Proceedings of the Second International Meeting on Bayesian Statistics*, 1985. (page 12)

C. Kemp and J.B. Tenenbaum. The discovery of structural form. *Proceedings of the National Academy of Sciences*, 105(31):10687–10692, 2008. (page 8)

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 8)

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. (page 10)

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989. (page 4)

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 7)

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 689–690. IEEE, 2011. (page 4)

C.E. Rasmussen and Z. Ghahramani. Occam's razor. In *Advances in Neural Information Processing Systems*, 2001. (page 6)

C.E. Rasmussen and CKI Williams. Gaussian Processes for Machine Learning. *The MIT Press, Cambridge, MA, USA*, 2006. (pages 1, 3, and 9)

D. Ruppert, M.P. Wand, and R.J. Carroll. *Semiparametric regression*, volume 12. Cambridge University Press, 2003. (page 7)

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. (pages 2 and 8)

M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009. (page 8)

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

L. Todorovski and S. Dzeroski. Declarative bias in equation discovery. In *International Conference on Machine Learning*, pages 376–384, 1997. (pages 2 and 8)

G. Wahba. *Spline models for observational data.* Society for Industrial Mathematics, 1990. ISBN 0898712440. (page 7)

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. (page 8)

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. (page 8)