Structure Discovery in Nonparametric Regression through Compositional Kernel Search

David Duvenaud[*][†]   DKD23@CAM.AC.UK
James Robert Lloyd[*][†]   JRL44@CAM.AC.UK
Roger Grosse[‡]   RGROSSE@MIT.EDU
Joshua B. Tenenbaum[‡]   JBT@MIT.EDU
Zoubin Ghahramani[†]   ZOUBIN@ENG.CAM.AC.UK

Abstract

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. 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 & Williams, 2006), linear combinations of fixed kernels (Bach, 2009), and mappings from the input space to an embedding space (Salakhutdinov & 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 & 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

---

[*]These authors contributed equally.
[†]University of Cambridge.
[‡]Massachusetts Institute of Technology.

the current and often opaque art of kernel engineering with a more transparent science of automated kernel discovery.

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 & Williams, 2006). Commonly used kernels families include the squared exponential (SE), periodic (PER), linear (LIN), and rational quadratic (RQ) (see Figure 1 and the appendix).
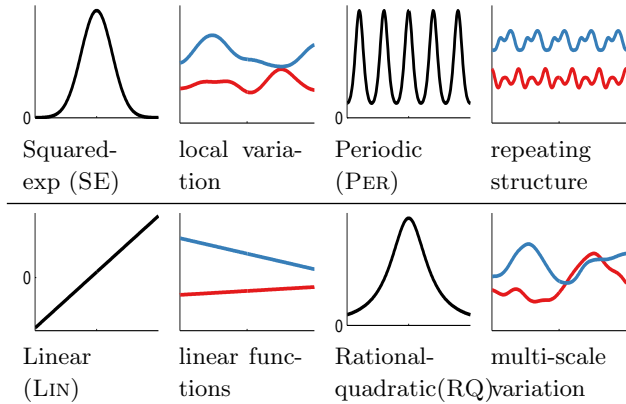
Figure 1. Left column and third columns: base kernels $k(\cdot, 0)$. Second and fourth columns: draws from a GP with each repective kernel. The x-axis has the same range on all plots.

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 kernel 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

[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.

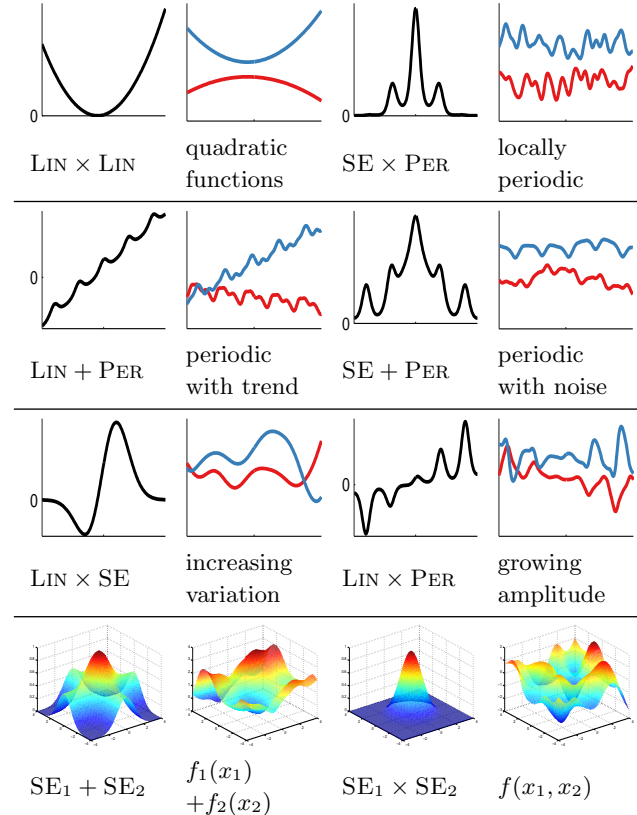kernel over the second dimension of x.

Figure 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.

Summation By summing kernels, we can model the data as a superposition of independent functions, possibly representing different structure. 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 it follows that $f := f_1 + f_2 \sim \mathcal{GP}(\mu_1 + \mu_2, k_1 + k_2)$.

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 & Tibshirani, 1990). These two kinds of structure are demonstrated in rows 2 and 4 of figure 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 $\text{SE}_1 \times \text{SE}_3$ 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 instance, PER corresponds to globally periodic structure, whereas PER $\times$ SE corresponds to locally periodic structure, as shown in row 1 of figure 2.

Many architectures for learning complex functions, such as convolutional networks (LeCun et al., 1989) and sum-product networks (Poon & 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. Because 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 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	$\sum_{d=1}^{D}$ SE$_d$
Automatic relevance determination	$\prod_{d=1}^{D}$ SE$_d$
Linear trend with 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 sine and cosine.

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 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 2.

- One can add features incrementally, analgous to algorithms like boosting, backfitting, or forward selection. These correspond to applying rules (1) or (2) to new dimensions not yet included in the model.

Scoring kernel families Choosing kernel structures requires a criterion for evaluating the structures. We choose marginal likelihood as our criterion since it balances the fit and complexity of a structure (e.g. Rasmussen & Ghahramani, 2001). To avoid an expensive integration over kernel parameters, we used the Bayesian information criterion (Schwarz, 1978) as an approximation.

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 construct smart initializations: all of the parameters which were part of the previous kernel are initialized to their previous values. The 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.

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 (Wahba, 1990; Gu, 2002). 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. Christoudias et al., 2009; Bach, 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 points 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 new data points. Salakhutdinov & Hinton (2008) use a deep neural network to learn an embedding; this is a flexible approach to kernel learning but potentially less interpretable.

Wilson & Adams (2013) derive kernels of the form $SE \times \cos(x - x')$, forming a basis for stationary kernels through spectral density estimation. These kernels share similarities with $SE \times PER$ but can express negative prior correlation, and could usefully be added to 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 goes beyond this prior work by demonstrating the structure implied by composite kernels, employing a Bayesian search criterion, and allowing for the automatic discovery of interpretable structure from data.

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 & Lipson, 2009), (Todorovski & 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 & 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. $SE_1 \times SE_2$ and $SE_1 \times PER_2$ 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.

5. Structure discovery in time series

In order to investigate our method's ability to discover structure, we performed the search over kernels on several time series datasets. Our method discovers rich structure in these datasets, and produces plausible extrapolations.

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 e.g.

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

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 2.

Mauna Loa atmospheric CO_2   First, we analyzed records of carbon dioxide levels recorded at the Mauna Loa observatory. Since this dataset was analyzed in detail by Rasmussen & Williams (2006), it serves as a test of our algorithm's ability to recover known structure.
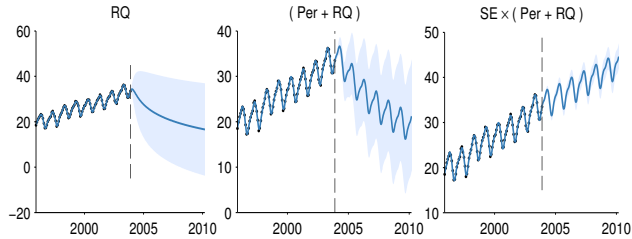
Figure 3. Posterior mean and variance for different depths of kernel search. The dashed line marks the extent of the dataset. In the first column, the function is only modeled as a locally smooth function, and the extrapolation is poor. Next, a periodic component is added, and the extrapolation improves. At depth 3, the kernel can capture most of the relevant structure, and is able to extrapolate reasonably.
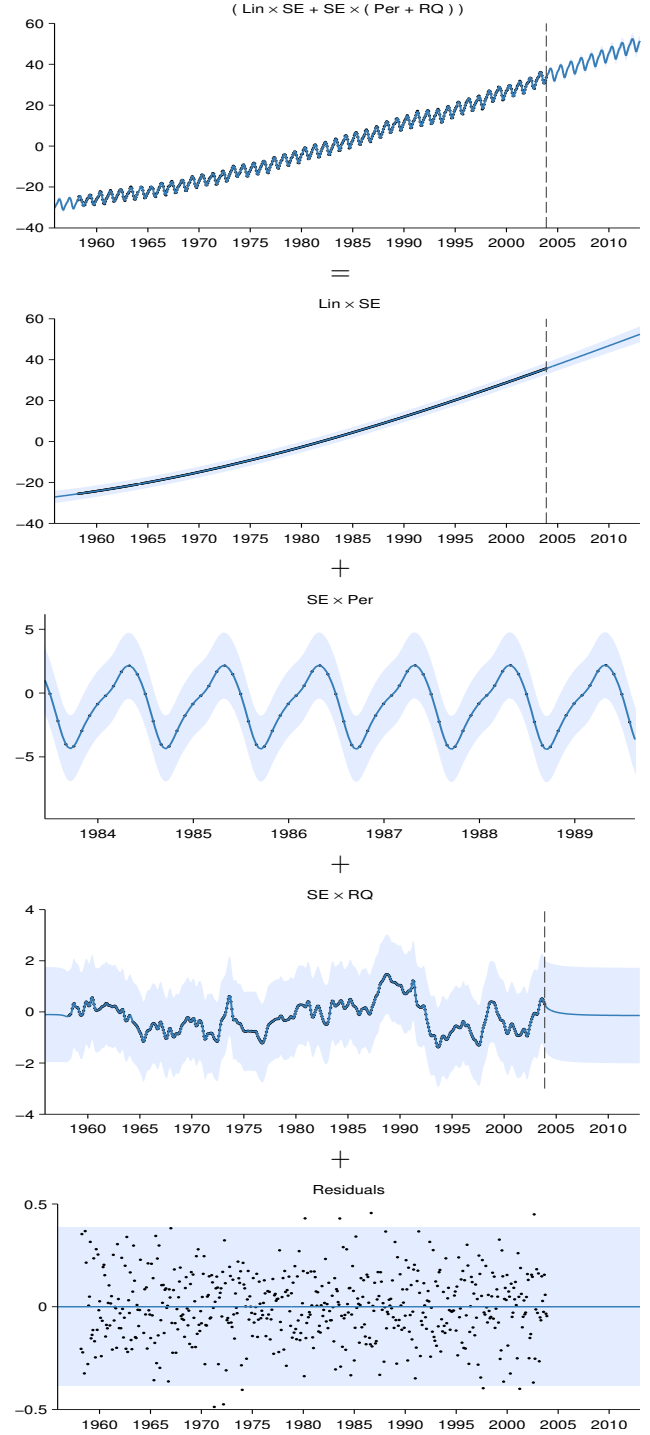
Figure 4. First row: The posterior on the Mauna Loa dataset, after a search of depth 10. Subsequent rows show the automatic decomposition of the time series. The decompositions shows long-term, yearly periodic, medium-term anomaly components and residuals, respectively. In the third row, the scale has been changed in order to clearly show the yearly periodic structure.
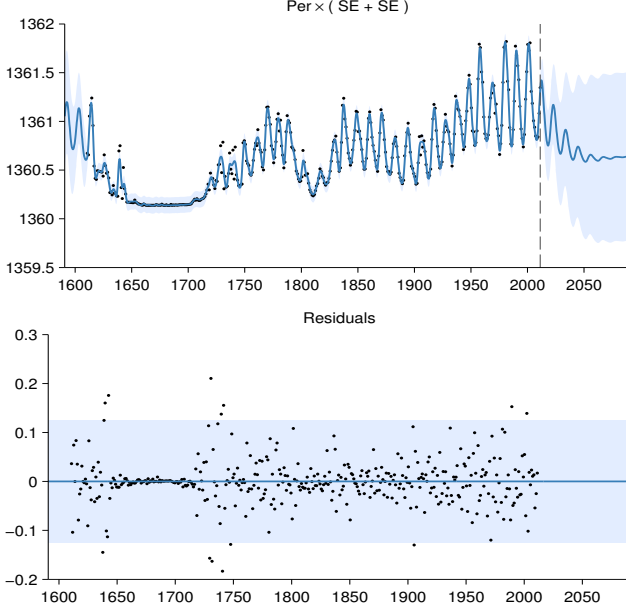
Figure 5. Full posterior and residuals on the solar irradiance dataset.

Figure 3 shows the posterior function estimates under increasingly structured models, as measured by approximate marginal likelihood, as the search depth increases on this dataset. Note that, while the data can be smoothly interpolated by a single base kernel model, the extrapolations improve dramatically as the search depth increases.

Figure 4 shows the complete posterior of the final model chosen by our method, together with its decomposition into additive components. The final model exhibits both a plausible extrapolation and interpretable components: a long-term trend, annual periodicity and medium-term deviations from the trend. We also plot the residuals, observing that there is little obvious structure left in the data.

Airline passenger data Figure 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 i.e. a long term trend, annual periodicity and medium term deviations. In addition, the composite kernel captures the near-linearity of the trend as well as 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 5.
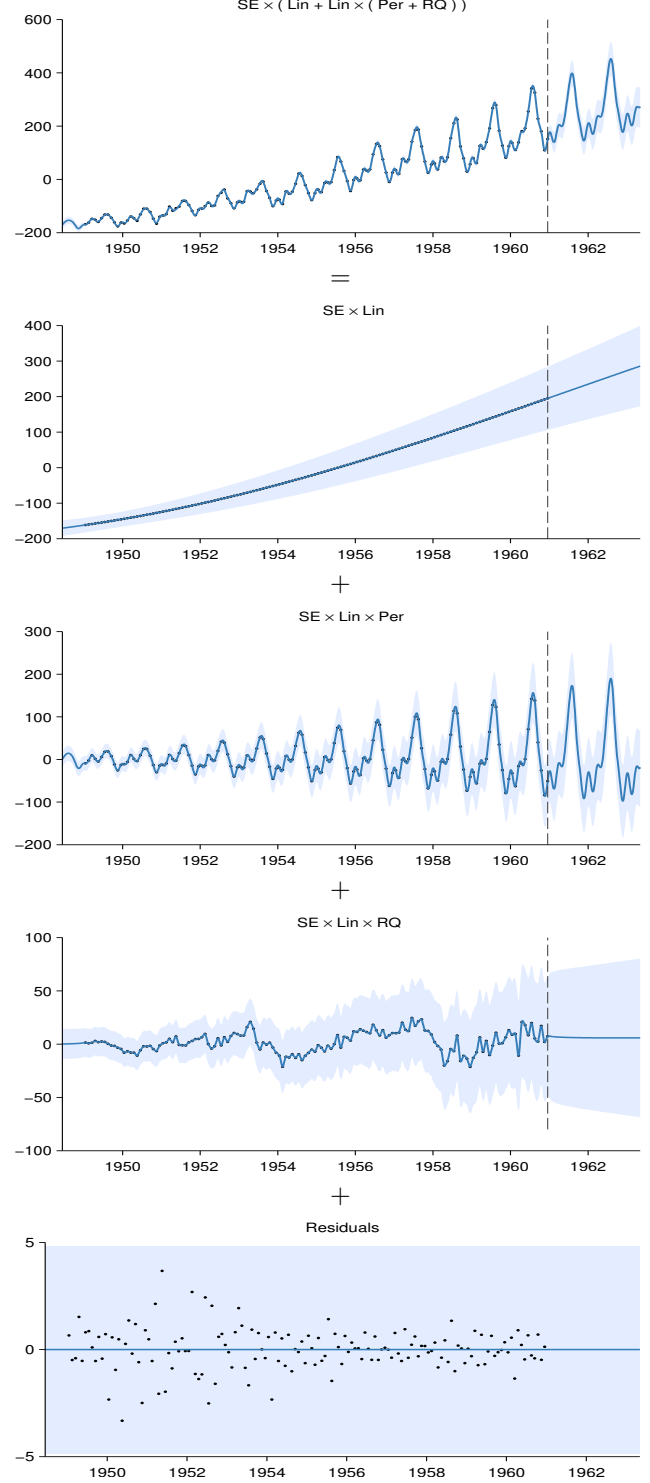

Figure 6. First row: The airline dataset and posterior after a search of depth 10. Subsequent rows: Additive decomposition of posterior into long-term smooth trend, yearly variation, and short-term deviations. Due to the linear kernel, the marginal variance grows over time, making this a heteroskedastic model.

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.

6. 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 function values, with variance chosen such that the standard deviation of the noise σ_n relative to the sample variance of the function was 0.1. Table 1 lists the composite kernels we used to generate data (subscripts indicate which dimension each kernel was applied to), the dimensionality D of the input space, the kernel chosen by our search, and the estimated noise variance.

Our method finds all of the relevant structure in all but one test. In fact, it also discovers unintentionally introduced linear structure: functions sampled from SE kernels with long length scales occasionally resulted in near-linear trends in the data.

The estimated noise level σ_n numerically assesses the quality of the fit to the data; large positive or negative deviations from 0.1 would indicate under-fitting and over-fitting, respectively. The values reported show that the model is not severely over- or under-fitting.

7. Quantitative evaluation

In addition to the qualitative evaluation in section 5, we investigated quantitatively how our method performs on both extrapolation and interpolation tasks.

7.1. Extrapolation

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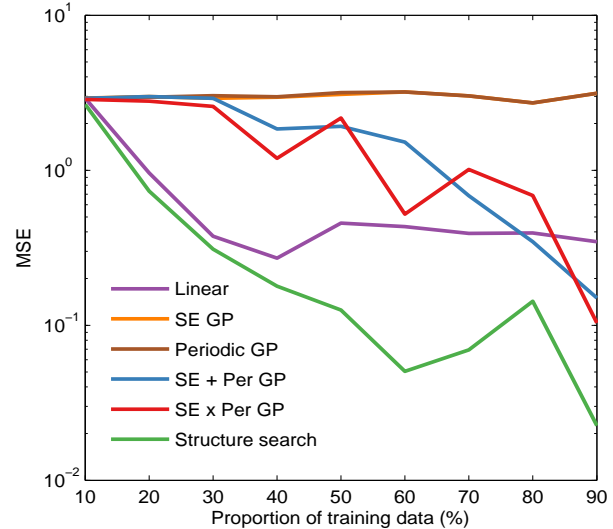

Figure 7. Extrapolation performance on the airline dataset. We plot test-set MSE as a function of the fraction of the dataset used for training.

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.

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[2] comparing 5 methods in terms of MSE and predictive likelihood. Our structure search was run up to depth 10, using SE and RQ as the 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 Hierarichical Kernel Learning (HKL).

Results are presented in table 2; our method outperformed all other methods in all tests.

All GP hyperparameter tuning was performed by au-

---

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

Table 1. Kernels used to generate synthetic data, dimensionality D of the input space, inferred kernels, and estimated noise level.

True kernel	D	Inferred kernels	Estimated σ_n (Truth = 0.1)
SE + RQ	1	RQ	0.090
LIN × PER	1	LIN × PER	0.098
$SE_1 + RQ_2$	2	$SE_1 + RQ_2$	0.094
$SE_1 + SE_2 \times PER_1 + SE_3$	3	$SE_1 + SE_2 \times PER_1 + SE_3 \times LIN_2$	0.082
$SE_1 \times SE_2$	4	$SE_1 \times SE_2 \times LIN_1$	0.103
$SE_1 \times SE_2 + SE_2 \times SE_3$	4	$(SE_1 \times SE_2 + SE_2 \times SE_3) \times LIN_3$	0.089
$(SE_1 + SE_2) \times (SE_3 + SE_4)$	4	$(SE_1 + SE_2) \times (SE_3 + SE_4 \times LIN_4)$	0.084

Table 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%.

Method	Mean Squared Error (MSE)					Negative Log-Likelihood				
	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**

tomated calls to the GPML toolbox[3]; Python code to perform all experiments is available on github[4].

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. 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, which provides an additional degree of reassurance that the learned structure reflects the world. We believe that a data-driven approach to choosing

kernel structures automatically can help make nonparametric regression and classification methods accessible to non-experts.

A. Appendix

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

$$k_{SE}(x,x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$$

$$k_{PER}(x,x') = \sigma^2 \exp\left(-\frac{2\sin^2(\pi|x-x'|/p)}{\ell^2}\right)$$

$$k_{LIN}(x,x') = \sigma_b^2 + \sigma_v^2(x-\ell)(x'-\ell)$$

$$k_{RQ}(x,x') = \sigma^2 \left(1 + \frac{(x-x')^2}{2\alpha\ell^2}\right)^{-\alpha}$$

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

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

---

[3] Available at www.gaussianprocess.org/gpml/code/
[4] github.com/jamesrobertlloyd/gp-structure-search

References

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

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

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

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

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

Duvenaud, D., Nickisch, H., and Rasmussen, C.E. Additive Gaussian processes. In *Advances in Neural Information Processing Systems*, 2011.

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

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

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

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

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

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

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

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

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

Poon, H. and Domingos, P. Sum-product networks: a new deep architecture. In *Conference on Uncertainty in AI*, 2011.

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

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

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

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

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

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

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

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

Washio, T., Motoda, H., Niwa, Y., 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, pp. 772–779, 1999.

Wilson, Andrew Gordon and Adams, Ryan Prescott. Gaussian process covariance kernels for pattern discovery and extrapolation. Technical Report arXiv:1302.4245 [stat.ML], February 2013.