Chapter 1

Additive Gaussian Processes

In this chapter, we introduce a Gaussian process model of functions which are *additive*. An additive function is one which decomposes into a sum of low-dimensional functions, each depending on only a subset of the input variables. Additive GPs generalize both Generalized Additive Models, and the standard GP models which use squared-exponential kernels. Hyperparameter learning in this model can be seen as Bayesian Hierarchical Kernel Learning (HKL). We introduce an expressive but tractable parameterization of the kernel function, which allows efficient evaluation of all input interaction terms, whose number is exponential in the input dimension. The additional structure discoverable by this model results in increased interpretability, as well as state-of-the-art predictive power in regression tasks.

1.1 Introduction

Most regression models in use today restrict the class of allowable functions to be of the form: $g(y) = f(x_1) + f(x_2) + \cdots + f(x_D)$. Popular special cases include logistic regression, linear regression, and Generalized Linear Models (Nelder & Wedderburn (1972)). Any function in this family defines a Generalized Additive Model (GAM) (Hastie & Tibshirani (1990)). Some extensions of this family, such as smoothing-splines ANOVA (Wahba (1990)), add terms depending on more than one variable. However, such models generally become intractable and difficult to fit as the number of terms increases.

At the other end of the spectrum are kernel-based models, which typically allow the response to depend on all input variables simultaneously. A popular example would be a Gaussian Process model using a squared-exponential (or Gaussian) kernel. We denote this model as SE-GP. This model is much more flexible than the GAM, but its flexibility poses a problem: To the extent that a set of variables is relevant to the output, this model must be highly uncertain about any novel combination of those variables.

In this paper, we introduce a Gaussian process model that generalizes both the GAM and SE-GP. This is achieved through a kernel which allows additive interactions of all orders, ranging from first order interactions (as in a GAM model) all the way to Dth-order interactions (as in a SE-GP). Although this kernel amounts to a sum over an exponential number of terms, we show how to compute this kernel efficiently, and introduce a parameterization which limits the number of hyperparameters to $O(D)$. A Gaussian process with this kernel function (an additive GP) constitutes a powerful model that allows one to automatically determine which orders of interactions are important. We show that this model can significantly improve modeling efficacy, and has major advantages for model interpretability. This model is also extremely simple to implement.

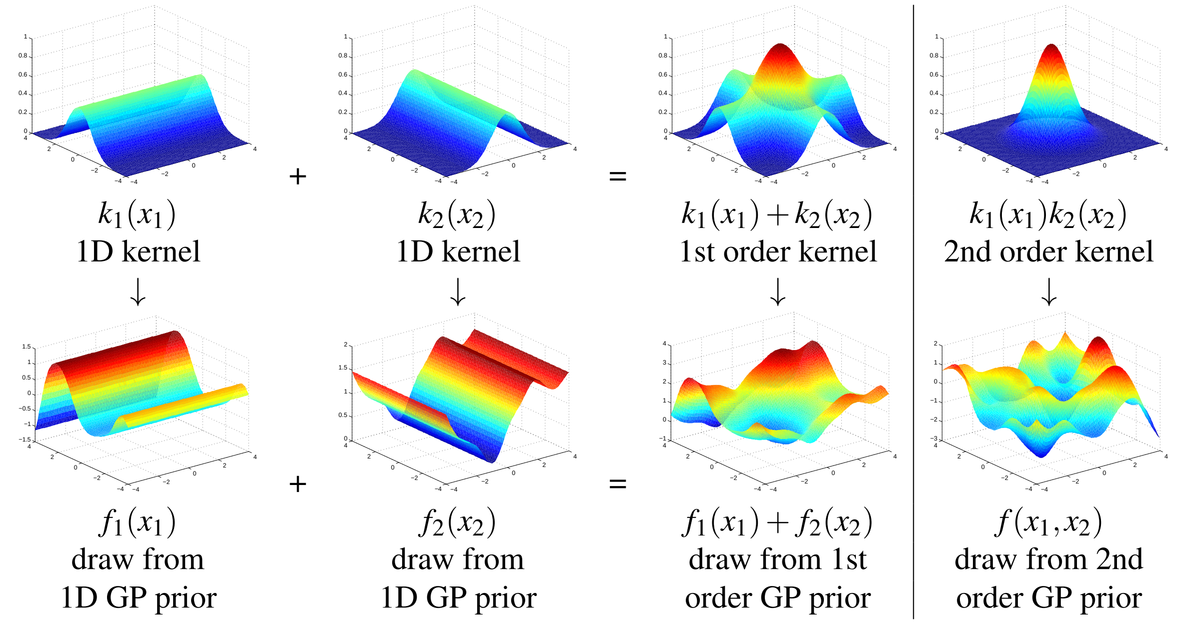
We note that a similar breakthrough has recently been made, called Hierarchical Kernel Learning (HKL) (Bach (2009)). HKL explores a similar class of models, and manages to deal with the possibly exponential number of interaction terms by cleverly selecting only a tractable subset. However, this method suffers considerably from the fact that cross-validation must be used to set hyperparameters. In addition, the machinery necessary to train these models is immense. Finally, on real datasets, HKL is outperformed by the standard SE-GP (Bach (2009)).

1.2 Gaussian Process Models

Gaussian processes are a flexible and tractable prior over functions, useful for solving regression and classification tasks (Rasmussen & Williams (2006)). The kind of structure which can be captured by a GP model is mainly determined by its *kernel*: the covariance function. One of the main difficulties in specifying a Gaussian process model is in choosing a kernel which can represent the structure present in the data. For small to medium-sized datasets, the kernel has a large impact on modeling efficacy.

Figure 1.1 compares, for two-dimensional functions, a first-order additive kernel with a second-order kernel. We can see that a GP with a first-order additive kernel is an example of a GAM: Each function drawn from this model is a sum of orthogonal one-dimensional functions. Compared to functions drawn from the higher-order GP, draws from the first-order GP have more long-range structure.

We can expect many natural functions to depend only on sums of low-order interactions. For example, the price of a house or car will presumably be well approximated by a sum of prices of individual features, such as a sun-roof. Other parts of the price may depend jointly



$k_1(x_1)$ 1D kernel

$k_2(x_2)$ 1D kernel

$k_1(x_1) + k_2(x_2)$ 1st order kernel

$k_1(x_1)k_2(x_2)$ 2nd order kernel

$f_1(x_1)$ draw from 1D GP prior

$f_2(x_2)$ draw from 1D GP prior

$f_1(x_1) + f_2(x_2)$ draw from 1st order GP prior

$f(x_1, x_2)$ draw from 2nd order GP prior

Fig. 1.1 A first-order additive kernel, and a product kernel. Left: a draw from a first-order additive kernel corresponds to a sum of draws from one-dimensional kernels. Right: functions drawn from a product kernel prior have weaker long-range dependencies, and less long-range structure.

on a small set of features, such as the size and building materials of a house. Capturing these regularities will mean that a model can confidently extrapolate to unseen combinations of features.

1.3 Additive Kernels

We now give a precise definition of additive kernels. We first assign each dimension $i \in \{1 \dots D\}$ a one-dimensional *base kernel* $k_i(x_i, x_i')$. We then define the first order, second order and nth order additive kernel as:

$$k_{add_1}(\mathbf{x}, \mathbf{x'}) = \sigma_1^2 \sum_{i=1}^{D} k_i(x_i, x_i') \tag{1.1}$$

$$k_{add_2}(\mathbf{x}, \mathbf{x'}) = \sigma_2^2 \sum_{i=1}^{D} \sum_{j=i+1}^{D} k_i(x_i, x_i') k_j(x_j, x_j') \tag{1.2}$$

$$k_{add_n}(\mathbf{x}, \mathbf{x'}) = \sigma_n^2 \sum_{1 \le i_1 < i_2 < \dots < i_n \le D} \prod_{d=1}^{N} k_{i_d}(x_{i_d}, x_{i_d}') \tag{1.3}$$

where D is the dimension of our input space, and σ_n^2 is the variance assigned to all nth order interactions. The nth covariance function is a sum of $\binom{D}{n}$ terms. In particular, the Dth order additive covariance function has $\binom{D}{D} = 1$ term, a product of each dimension's covariance function:

$$k_{add_D}(\mathbf{x}, \mathbf{x}') = \sigma_D^2 \prod_{d=1}^{D} k_d(x_d, x_d') \tag{1.4}$$

In the case where each base kernel is a one-dimensional squared-exponential kernel, the Dth-order term corresponds to the multivariate squared-exponential kernel:

$$k_{add_D}(\mathbf{x}, \mathbf{x}') = \sigma_D^2 \prod_{d=1}^{D} k_d(x_d, x_d') = \sigma_D^2 \prod_{d=1}^{D} \exp\left(-\frac{(x_d - x_d')^2}{2l_d^2}\right) = \sigma_D^2 \exp\left(-\sum_{d=1}^{D} \frac{(x_d - x_d')^2}{2l_d^2}\right) \tag{1.5}$$

also commonly known as the Gaussian kernel. The full additive kernel is a sum of the additive kernels of all orders.

1.3.1 Parameterization

The only design choice necessary in specifying an additive kernel is the selection of a one-dimensional base kernel for each input dimension. Any parameters (such as length-scales) of the base kernels can be learned as per normal by maximizing the marginal likelihood of the training data.

In addition to the hyperparameters of each dimension-wise kernel, additive kernels are equipped with a set of D hyperparameters $\sigma_1^2 \dots \sigma_D^2$ controlling how much variance we assign to each order of interaction. These "order variance" hyperparameters have a useful interpretation: The dth order variance hyperparameter controls how much of the target function's variance comes from interactions of the dth order. Table 1.1 shows examples of normalized order variance hyperparameters.

On different datasets, the dominant order of interaction estimated by the additive model varies widely.An additive GP with all of its variance coming from the 1st order is equivalent to a GAM; an additive GP with all its variance coming from the Dth order is equivalent to a SE-GP.

Model Selection

Because the hyperparameters can specify which degrees of interaction are important, the additive GP is an extremely general model. By optimizing these hyperparameters, we are approximately performing weighted model selection over a set of D models. If the function we are

Table 1.1 Variance contribution of each order in the additive model, on different datasets. The maximum order of interaction is set to 10, or smaller if the input dimension less than 10. Values are normalized to sum to 100.

Order of interaction	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
pima	0.1	0.1	0.1	0.3	1.5	96.4	1.4	0.0		
liver	0.0	0.2	99.7	0.1	0.0	0.0				
heart	77.6	0.0	0.0	0.0	0.1	0.1	0.1	0.1	0.1	22.0
concrete	70.6	13.3	13.8	2.3	0.0	0.0	0.0	0.0		
pumadyn-8nh	0.0	0.1	0.1	0.1	0.1	0.1	0.1	99.5		
servo	58.7	27.4	0.0	13.9						
housing	0.1	0.6	80.6	1.4	1.8	0.8	0.7	0.8	0.6	12.7

modeling is, in fact, decomposable into a sum of low-dimensional functions, our model can discover this fact (see Figure 1.5) and exploit it. If this is not the case, the hyperparameters can specify a suitably flexible model.

Interpretability

As noted by Plate (1999), one of the chief advantages of additive models such as GAM is their interpretability, since a high-dimensional function is decomposed into a series of one- or two-dimensional functions. Plate also notes that by allowing high-order interactions as well as low-order interactions, one can trade off interpretability with predictive accuracy. In the case where the hyperparameters indicate that most of the variance in a function can be explained by low-order interactions, it is useful and easy to plot the corresponding low-order functions, as in Figure 1.2.

1.3.2 Efficient Evaluation of Additive Kernels

An additive kernel over D inputs with interactions up to order n has $O(2^n)$ terms. Computing the sum over these terms naïvely quickly becomes intractable. In this section, we show how one can evaluate the sum over all terms in $O(D^2)$.

The nth order additive kernel corresponds to the nth *elementary symmetric polynomial* (Macdonald (1998); Stanley (2001)), which we denote e_n. For example: if $\mathbf{x}$ has 4 input

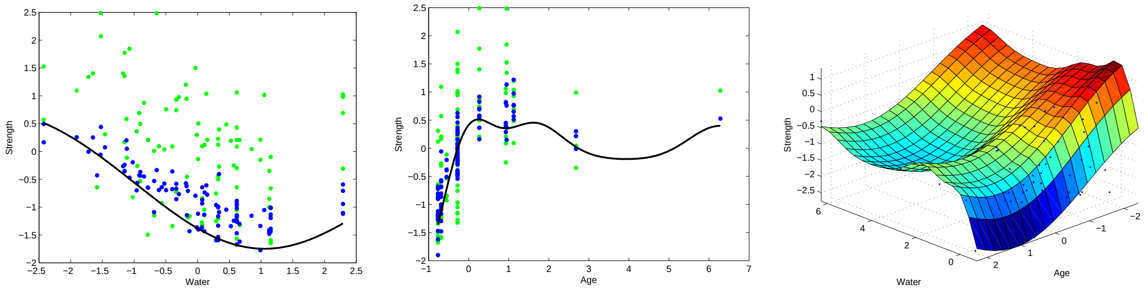


Fig. 1.2 Low-order functions on the concrete dataset. Left, Centre: By considering only first-order terms of the additive kernel, we recover a form of Generalized Additive Model, and can plot the corresponding 1-dimensional functions. Green points indicate the original data, blue points are data after the mean contribution from the other first-order terms has been subtracted. The black line is the posterior mean of a GP with only one term in its kernel. Right: The posterior mean of a GP with only one second-order term in its kernel.

dimensions ($D = 4$), and if we let $z_i = k_i(x_i, x_i')$, then

$$\begin{aligned}
k_{add_1}(\mathbf{x}, \mathbf{x'}) &= e_1(z_1, z_2, z_3, z_4) = z_1 + z_2 + z_3 + z_4 \\
k_{add_2}(\mathbf{x}, \mathbf{x'}) &= e_2(z_1, z_2, z_3, z_4) = z_1 z_2 + z_1 z_3 + z_1 z_4 + z_2 z_3 + z_2 z_4 + z_3 z_4 \\
k_{add_3}(\mathbf{x}, \mathbf{x'}) &= e_3(z_1, z_2, z_3, z_4) = z_1 z_2 z_3 + z_1 z_2 z_4 + z_1 z_3 z_4 + z_2 z_3 z_4 \\
k_{add_4}(\mathbf{x}, \mathbf{x'}) &= e_4(z_1, z_2, z_3, z_4) = z_1 z_2 z_3 z_4
\end{aligned}$$

The Newton-Girard formulae give an efficient recursive form for computing these polynomials:

$$k_{add_n}(\mathbf{x}, \mathbf{x'}) = e_n(z_1, \dots, z_n) = \frac{1}{n} \sum_{k=1}^{n} (-1)^{(k-1)} e_{n-k}(z_1, \dots, z_D) s_k(z_1, \dots, z_D) \qquad (1.6)$$

where s_k is the kth **power sum**: $s_k(z_1, z_2, \dots, z_n) = \sum_{i=1}^{D} z_i^k$. The Newton-Girard formulae have time complexity $O(D^2)$, while computing a sum over an exponential number of terms.

Conveniently, we can use the same trick to efficiently compute all of the necessary derivatives of the additive kernel with respect to the base kernels. We merely need to remove the kernel of interest from each term of the polynomials:

$$\frac{\partial k_{add_n}}{\partial z_j} = e_{n-1}(z_1, \dots, z_{j-1}, z_{j+1}, \dots z_D) \qquad (1.7)$$

This trick allows us to optimize the base kernel hyperparameters with respect to the marginal likelihood.

Computation

The computational cost of evaluating the Gram matrix of a product kernel (such as the SE kernel) is $O(N^2D)$, while the cost of evaluating the Gram matrix of the additive kernel is $O(N^2D+N^2R^2)$, where R is the maximum degree of interaction allowed (up to D). In higher dimensions, this can be a significant cost, even relative to the fixed $O(N^3)$ cost of inverting the Gram matrix. However, as our experiments show, typically only the first few orders of interaction are important for modeling a given function; hence if one is computationally limited, one can simply limit the maximum degree of interaction without losing much accuracy.

Additive Gaussian processes are particularly appealing in practice because their use requires only the specification of the base kernel. All other aspects of GP inference remain the same. All of the experiments in this paper were performed using the standard GPML toolbox [1]. Code to perform all experiments is available at the author's website.

1.4 Related Work

Plate (1999) constructs a form of additive GP, but using only the first-order and Dth order terms. This model is motivated by the desire to trade off the interpretability of first-order models, with the flexibility of full-order models. Our experiments show that often, the intermediate degrees of interaction contribute most of the variance.

A related functional ANOVA GP model by Kaufman & Sain (2010) decomposes the *mean* function into a weighted sum of GPs. However, the effect of a particular degree of interaction cannot be quantified by that approach. Also, computationally, the Gibbs sampling approach is disadvantageous.

Christoudias *et al.* (2009) previously showed how mixtures of kernels can be learnt by gradient descent in the Gaussian process framework. They call this *Bayesian localized multiple kernel learning*. However, their approach learns a mixture over a small, fixed set of kernels, while our method learns a mixture over all possible products of those kernels.

1.4.1 Hierarchical Kernel Learning

Bach (2009) uses a regularized optimization framework to learn a weighted sum over an exponential number of kernels which can be computed in polynomial time. The subsets of kernels

[1] Available at `http://www.gaussianprocess.org/gpml/code/`

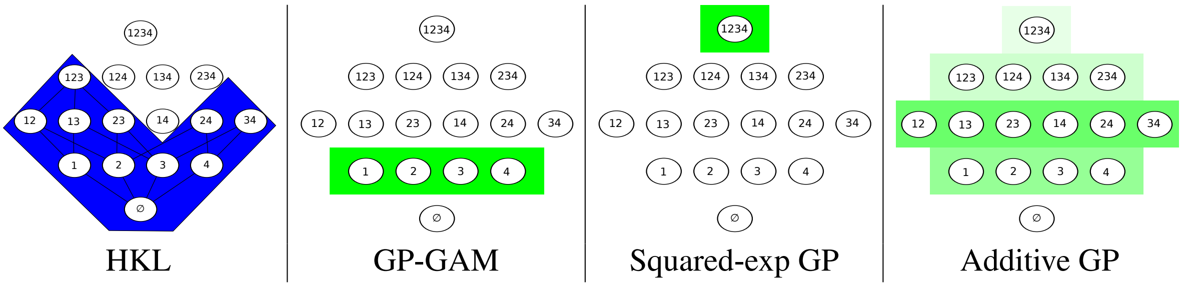


Fig. 1.3 Nodes represent different interaction terms, ranging from first-order to fourth-order interactions. Far left: HKL can select a hull of interaction terms, but must use a pre-determined weighting over those terms. Far right: the additive GP model can weight each order of interaction seperately. Neither the HKL nor the additive model dominate one another in terms of flexibility, however the GP-GAM and the SE-GP are special cases of additive GPs.

considered by this method are restricted to be a *hull* of kernels.[2] Given each dimension's kernel, and a pre-defined weighting over all terms, HKL performs model selection by searching over hulls of interaction terms. In Bach (2009), Bach also fixes the relative weighting between orders of interaction with a single term α, computing the sum over all orders by:

$$k_a(\mathbf{x},\mathbf{x}') = v_D^2 \prod_{d=1}^{D} \left(1 + \alpha k_d(x_d, x_d')\right) \tag{1.8}$$

which has computational complexity $O(D)$. However, this formulation forces the weight of all nth order terms to be weighted by α^n.

Figure 1.3 contrasts the HKL hull-selection method with the Additive GP hyperparameter-learning method. Neither method dominates the other in flexibility. The main difficulty with the approach of Bach (2009) is that hyperparameters are hard to set other than by cross-validation. In contrast, our method optimizes the hyperparameters of each dimension's base kernel, as well as the relative weighting of each order of interaction.

1.4.2 ANOVA Procedures

Vapnik (1998) introduces the support vector ANOVA decomposition, which has the same form as our additive kernel. However, they recommend approximating the sum over all D orders with only one term "of appropriate order", presumably because of the difficulty of setting the hyperparameters of a SVM. Stitson *et al.* (1999) performed experiments which favourably

[2] In the setting we are considering in this paper, a hull can be defined as a subset of all terms such that if term $\prod_{j\in J} k_j(\mathbf{x},\mathbf{x}')$ is included in the subset, then so are all terms $\prod_{j\in J/i} k_j(\mathbf{x},\mathbf{x}')$, for all $i \in J$. For details, see Bach (2009).

compared the support vector ANOVA decomposition to polynomial and spline kernels. They too allowed only one order to be active, and set hyperparameters by cross-validation.

A closely related procedure from the statistics literature is smoothing-splines ANOVA (SS-ANOVA) (Wahba (1990)). An SS-ANOVA model is estimated as a weighted sum of splines along each dimension, plus a sum of splines over all pairs of dimensions, all triplets, etc, with each individual interaction term having a separate weighting parameter. Because the number of terms to consider grows exponentially in the order, in practice, only terms of first and second order are usually considered. Learning in SS-ANOVA is usually done via penalized-maximum likelihood with a fixed sparsity hyperparameter.

In contrast to these procedures, our method can easily include all D orders of interaction, each weighted by a separate hyperparameter. As well, we can learn kernel hyperparameters individually per input dimension, allowing automatic relevance determination to operate.

1.4.3 Non-local Interactions

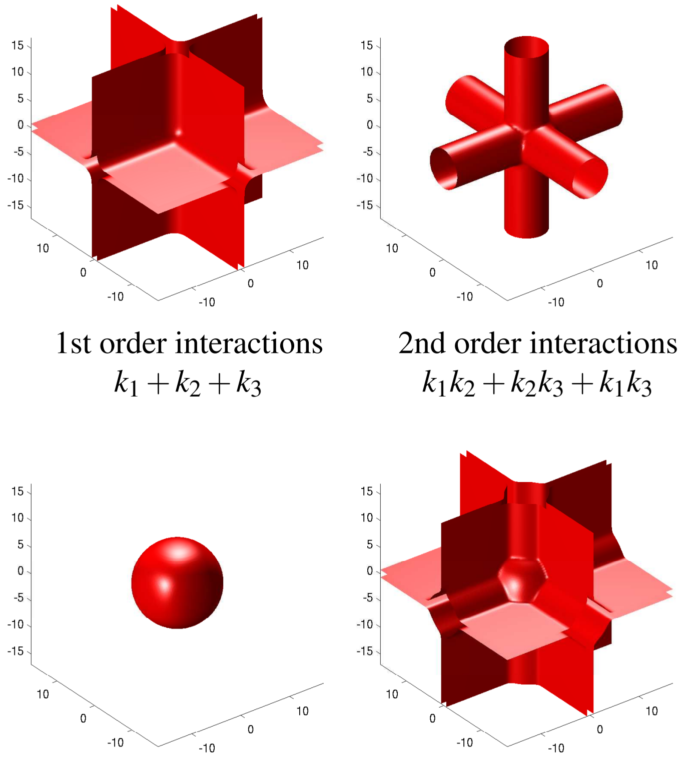
By far the most popular kernels for regression and classification tasks on continuous data are the squared exponential (Gaussian) kernel, and the Matérn kernels. These kernels depend only on the scaled Euclidean distance between two points, both having the form: $k(\mathbf{x}, \mathbf{x}') = f(\sum_{d=1}^{D} \left(x_d - x'_d\right)^2 / l_d^2)$. Bengio *et al.* (2006) argue that models based on squared-exponential kernels are particularily susceptible to the *curse of dimensionality*. They emphasize that the locality of the kernels means that these models cannot capture non-local structure. They argue that many functions that we care about have such structure. Methods based solely on local kernels will require training examples at all combinations of relevant inputs.

Additive kernels have a much more complex structure, and allow extrapolation based on distant parts of the input space, without spreading the mass of the kernel over the whole space. For example, additive kernels of the first order allow strong non-local interactions along every axis. Figure 1.4 compares a squared-exponential kernel with an additive kernel in 3 dimensions.

1.5 Experiments

1.5.1 Synthetic Data

Because additive kernels can discover non-local structure in data, they are exceptionally well-suited to problems where local interpolation fails. We constructed a dataset to demonstrate this ability in additive GPs, consisting of data drawn from a sum of two axis-aligned sine functions.



1st order interactions
$k_1 + k_2 + k_3$

2nd order interactions
$k_1 k_2 + k_2 k_3 + k_1 k_3$

3rd order interactions
$k_1 k_2 k_3$
(Squared-exp kernel)

All interactions

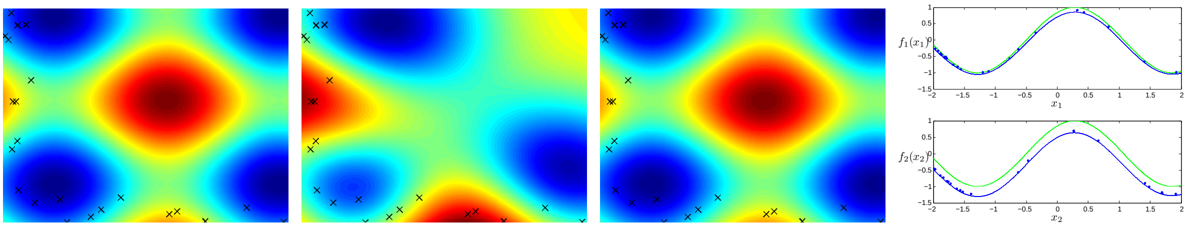
(Additive kernel)

Fig. 1.4 Isocontours of additive kernels in 3 dimensions. The third-order kernel only considers nearby points relevant, while the lower-order kernels allow the output to depend on distant points, as long as they share one or more input value.

The training set is restricted to a small, L-shaped area; the test set contains a peak far from the training set locations. The additive GP recovered both of the original sine functions (shown in green), and inferred correctly that most of the variance in the function comes from first-order interactions. Figure 1.5 shows the results. The ability of additive GPs to discover long-range structure suggests that this model may be well-suited to deal with covariate-shift problems.

1.5.2 Experimental Setup

On a diverse collection of datasets, we compared five different models. In the results tables below, GP Additive refers to a GP using the additive kernel with squared-exp base kernels. For speed, we limited the maximum order of interaction in the additive kernels to 10. GP-GAM denotes an additive GP model with only first-order interactions. GP Squared-Exp is



True Function & data locations

Squared-exp GP posterior mean

Additive GP posterior mean

Additive GP 1st-order functions

Fig. 1.5 Long-range inference in functions with additive structure.

a GP model with a squared-exponential ARD kernel. HKL[3] was run using the all-subsets kernel, which corresponds to the same set of kernels as considered by the additive GP with a squared-exp base kernel.

For all GP models, we fit hyperparameters by the standard method of maximizing training-set marginal likelihood, using L-BFGS (Nocedal (1980)) for 500 iterations, allowing five random restarts. In addition to learning kernel hyperparameters, we fit a constant mean function to the data. In the classification experiments, GP inference was done using expectation propagation (Minka (2001)).

1.5.3 Results

Tables 1.2, 1.3, 1.4 and 1.5 show mean performance across 10 train-test splits. Because HKL does not specify a noise model, it could not be included in the likelihood comparisons.

Table 1.2 Regression Mean Squared Error

Method	bach	concrete	pumadyn-8nh	servo	housing
Linear Regression	1.031	0.404	0.641	0.523	0.289
GP GAM	1.302	0.142	0.602	0.281	0.179
HKL	**0.199**	0.147	0.346	0.199	0.151
GP Squared-exp	**0.045**	0.159	**0.317**	**0.124**	**0.092**
GP Additive	**0.045**	**0.097**	**0.317**	**0.110**	**0.102**

The model with best performance on each dataset is in bold, along with all other models that were not significantly different under a paired t-test. The additive model never performs significantly worse than any other model, and sometimes performs significantly better than all

[3] Code for HKL available at `http://www.di.ens.fr/~fbach/hkl/`

Table 1.3 Regression Negative Log Likelihood

Method	bach	concrete	pumadyn-8nh	servo	housing
Linear Regression	2.430	1.403	1.881	1.678	1.052
GP GAM	1.746	0.433	1.167	0.800	0.563
GP Squared-exp	−**0.131**	0.412	**0.843**	0.425	**0.208**
GP Additive	−**0.131**	**0.181**	**0.843**	**0.309**	**0.161**

Table 1.4 Classification Percent Error

Method	breast	pima	sonar	ionosphere	liver	heart
Logistic Regression	7.611	24.392	26.786	16.810	45.060	**16.082**
GP GAM	**5.189**	**22.419**	**15.786**	**8.524**	**29.842**	**16.839**
HKL	**5.377**	24.261	**21.000**	9.119	**27.270**	**18.975**
GP Squared-exp	**4.734**	**23.722**	**16.357**	**6.833**	**31.237**	**20.642**
GP Additive	**5.566**	**23.076**	**15.714**	**7.976**	**30.060**	**18.496**

Table 1.5 Classification Negative Log Likelihood

Method	breast	pima	sonar	ionosphere	liver	heart
Logistic Regression	0.247	0.560	4.609	0.878	0.864	0.575
GP GAM	**0.163**	**0.461**	**0.377**	**0.312**	**0.569**	**0.393**
GP Squared-exp	**0.146**	0.478	**0.425**	**0.236**	**0.601**	0.480
GP Additive	**0.150**	**0.466**	**0.409**	**0.295**	**0.588**	**0.415**

other models. The difference between all methods is larger in the case of regression experiments. The performance of HKL is consistent with the results in Bach (2009), performing competitively but slightly worse than SE-GP.

Because the additive GP is a superset of both the GP-GAM model and the SE-GP model, instances where the additive GP model performs significantly worse are presumably due to over-fitting, or due to the hyperparameter optimization becoming stuck in a local maximum. Additive GP performance can be expected to benefit significantly from integrating out the kernel hyperparameters.

1.6 Future Work

The kernel used by Bach (2009) is less flexible than ours, but has time complexity $O(N^2 D)$ as opposed to the additive kernel as defined in this chapter. An experimental evaluation of Gaussian process regression with this kernel would reveal if the all-subsets kernel is a faster alternative to the fully-parametrized GP model.

One difficulty with the current formulation of additive GPs is that the length-scale parameters of the base kernels are use as distance metrics in spaces of different dimension. Upon examination of the lengthscales of sq-exp GPs and additive GPs, it was found that in the cases where the lower orders contributed the most variance, the length-scale parameters were consistently much smaller than in the cases where the high-order terms dominated. This is strong evidence that different length-scales are appropriate for different orders of interaction.

An naive solution to this problem would be to multiply the lengthscales of the interaction terms in the dth order of interaction by a $\sqrt(d)$.

One tantalizing takeaway from the experimental results presented here is that, for Gaussian process classification (GPC), an additive model using only the first-order interactions is as good or better that full-interaction additive GPs as well as sq-exp GPs. This result is encouraging because recently, (Yunus et al) demonstrated that inference in first-order additive GPs is possible in $O(ND)$. In the case of GPC, enforcing only first-order interactions may give a massive gain in speed for essentially no cost in accuracy when using maximum-likelihood estimates for the hyperparameters.

1.7 Conclusion

We present additive Gaussian processes: a simple family of models which generalizes two widely-used classes of models. Additive GPs also introduce a new type of structure into the GP framework. Our experiments indicate that such additive structure is present in real datasets, allowing our model to perform better than standard GP models. In the case where no such structure exists, our model can recover arbitrarily flexible models, as well.

In addition to improving modeling efficacy, the additive GP also improves model interpretability: the order variance hyperparameters indicate which sorts of structure are present in our model.

Compared to HKL, which is the only other tractable procedure able to capture the same types of structure, our method benefits from being able to learn individual kernel hyperparameters, as well as the weightings of different orders of interaction. Our experiments show that additive GPs are a state-of-the-art regression model.

end of document 1

References

BACH, F. (2009). High-dimensional non-linear variable selection through hierarchical kernel learning. *CoRR*, **abs/0909.0844**.

BENGIO, Y., DELALLEAU, O. & LE ROUX, N. (2006). The curse of highly variable functions for local kernel machines. *Advances in neural information processing systems*, **18**, 107.

CHRISTOUDIAS, M., URTASUN, R. & DARRELL, T. (2009). Bayesian localized multiple kernel learning.

HASTIE, T. & TIBSHIRANI, R. (1990). *Generalized additive models*. Chapman & Hall/CRC.

KAUFMAN, C. & SAIN, S. (2010). Bayesian functional anova modeling using gaussian process prior distributions. *Bayesian Analysis*, **5**, 123–150.

MACDONALD, I. (1998). *Symmetric functions and Hall polynomials*. Oxford University Press, USA.

MINKA, T. (2001). Expectation propagation for approximate bayesian inference. In *Uncertainty in Artificial Intelligence*, vol. 17, 362–369, Citeseer.

NELDER, J. & WEDDERBURN, R. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, **135**, 370–384.

NOCEDAL, J. (1980). Updating quasi-newton matrices with limited storage. *Mathematics of computation*, **35**, 773–782.

PLATE, T. (1999). Accuracy versus interpretability in flexible modeling: Implementing a tradeoff using gaussian process models. *Behaviormetrika*, **26**, 29–50.

RASMUSSEN, C. & WILLIAMS, C. (2006). Gaussian Processes for Machine Learning. *The MIT Press, Cambridge, MA, USA*, **38**, 715–719.

STANLEY, R. (2001). *Enumerative combinatorics*. Cambridge University Press.

STITSON, M., GAMMERMAN, A., VAPNIK, V., VOVK, V., WATKINS, C. & WESTON, J. (1999). Support vector regression with ANOVA decomposition kernels. *Advances in kernel methods—Support vector learning*, 285–292.

VAPNIK, V. (1998). *Statistical learning theory*, vol. 2. Wiley New York.

WAHBA, G. (1990). *Spline models for observational data*. Society for Industrial Mathematics.