Stat Papers (2011) 52:813–833
DOI 10.1007/s00362-009-0289-6

REGULAR ARTICLE

Bias-variance trade-off for prequential model list selection

Ernest Fokoue · Bertrand Clarke

Received: 9 December 2008 / Revised: 17 August 2009 / Published online: 13 December 2009

Abstract The prequential approach to statistics leads naturally to model list selection because the sequential reformulation of the problem is a guided search over model lists drawn from a model space. That is, continually updating the action space of a decision problem to achieve optimal prediction forces the collection of models under consideration to grow neither too fast nor too slow to avoid excess variance and excess bias, respectively. At the same time, the goal of good predictive performance forces the search over good predictors formed from a model list to close in on the data generator. Taken together, prequential model list re-selection favors model lists which provide an effective approximation to the data generator but do so by making the approximation match the unknown function on important regions as determined by empirical bias and variance.

Keywords Prequential · Online prediction · Bias-variance trade-off · Model selection · Bayes model averaging · Model list selection

E. Fokoue (✉)
Center for Quality and Applied Statistics, Rochester Institute of Technology,
98 Lomb Memorial Drive, Rochester, NY 14623, USA
e-mail: ernest.fokoue@rit.edu

B. Clarke
Department of Medicine, University of Miami, 1120 NW 14th Street CRB 611 (C-213),
Miami, FL 33136, USA
e-mail: bclarke2@med.miami.edu

B. Clarke
Department of Epidemiology and Public Health, University of Miami, Miami, USA

B. Clarke
Center for Computational Sciences, University of Miami, Miami, USA

1 Introduction

Model uncertainty in its various guises is the dominant source of uncertainty in a large class of problems, often from a Bayesian perspective, see Draper (1995), Gustafson and Clarke (2004) and Czado and Raftery (2006). In predictive contexts, it is well known that accounting for model uncertainty improves predictive performance, see for instance the moving block bootstrap, Alonso et al. (2006). This can also be done by Bayes model averaging, see Leamer (1978), Kass and Raftery (1995), as well as by Frequentist model averaging methods, see Wolpert (1992), Hjort and Claeskens (2003) and Wong (1995). Model averages provide improved prediction because predictions from single model ignores the fact that another plausible model could make equally good different predictions. Existing model averaging methods rest on using a fixed model list, so model list uncertainty is an open question.

Here we approach model list uncertainty in the context of averages of additive models. Our goal is to optimize the predictive error over choice of model list used to form the average. Since direct optimization is mathematically difficult, we search for an optimal model list by trying to evolve one as the data accumulate. We start with a simple initial model list and then add models to it greedily, so as to reduce the residual error. The predictor, here a Bayes model average, is fixed and the input model list to the Bayes model average is allowed to vary. Bayes model averages behave differently from model combinations, see Minka (2000) and differently from data pooling, cf. Toutenberg and Shalabh (2002)

An important feature of our technique is that the model list search is done prequentially in response to predictive errors in the sense of Dawid (1984), see also Dawid and Vovk (1999). One of the motivations for prequentialism is increased importance on validation of modeling strategies, here applied to model list selection. The (weak) prequential principle can be informally stated as follows: Methods of evaluation of a predictor should depend on the predictor only through the accuracy of its predictions. In particular, no other aspect of the modeling strategy should affect its assessment. Here, this principle is satisfied because the model list reselection depends only on the difference between the predictions and the observations. Thus, under the prequential principle two model lists leading to the same sequence of predictions would be indistinguishable. One of the implications of this is that all the comparisons of model lists can be done without favoring or disfavoring any particular sort; all model lists compete equally to to help predict the next outcome.

The main reason this approach is important for model list selection is that we will be searching for model lists rather than finding them through optimization. This means that the main pressure to find good lists comes from validation of predictions. So, it is important to validate after each batch of data to rule out poor lists quickly while constraining the growth of the number of models on the list. Note that the results we give are averaged over runs so that the specific effects of a sequence of observations are attenuated. The extensive validation is intended to isolate the effect of the list apart from the randomness in the data.

More formally, suppose we have a data generator, DG, producing pairs $(\mathbf{x}_i, y_i)$ one at a time, according to an underlying true function f^*. That is, each response is of the form $Y_i = f^*(\mathbf{x}_i) + \epsilon_i$, where the ϵ_i's are independently normally distributed

with mean 0 and variance σ^2. To approximate f^* consider additive models M of the form

$$Y = \beta_0 + \sum_j \beta_j B_j(\mathbf{x}) + \epsilon, \tag{1}$$

where each B_γ is a function of $\mathbf{x}$ from an ensemble

$$\mathbb{B} = \{B_1, B_2, \ldots, B_q, \ldots\}. \tag{2}$$

Typically, $\mathbb{B}$ is chosen to be a complete basis. In some settings it is worthwhile to choose $\mathbb{B}$ to be overcomplete in the sense that a linear combination of elements in $\mathbb{B}$ may equal or approximate another element in $\mathbb{B}$. In general, $\mathbb{M}$ denotes the model space derived from $\mathbb{B}$, and $\mathcal{M}$ denotes a subset of $\mathbb{M}$, which we call a model list. A model M as in (1) is an element M_j of $\mathcal{M} = \{M_1, \ldots, M_k\}$.

The goal here is to find the model list $\mathcal{M}^{\mathsf{opt}}$ that minimizes the Predictive Mean Squared Error (PMSE), i.e.,

$$\mathcal{M}^{\mathsf{opt}} = \arg \min_{\mathcal{M} \subset \mathbb{M}} \mathsf{PMSE}(\mathcal{M}), \tag{3}$$

where

$$\mathsf{PMSE}(\mathcal{M}) = \mathbb{E}_{\mathsf{P}_{\mathsf{true}}}\left[(Y^{\mathsf{new}} - \hat{Y}^{\mathsf{new}})^2\right], \tag{4}$$

in which the expectation is taken with respect to $\mathsf{P}_{\mathsf{true}}$, the density for Y^{new}, and the estimate $\hat{Y}^{\mathsf{new}}$ is the response predicted for $\mathbf{x}^{\mathsf{new}}$ using BMA on $\mathcal{M}$, i.e.,

$$\hat{Y}^{\mathsf{new}} \equiv \mathsf{BMA}(\mathbf{x}^{\mathsf{new}}; \mathcal{M}),$$

where BMA represents the Bayes model average. (The dependence of $\hat{Y}^{\mathsf{new}}$ on the data $(\mathbf{x}_i, y_i)$ is suppressed in the notation.) This criterion automatically gives sparsity because it devolves to a variance bias decomposition and too many terms will give excessive variance. We have chosen BMA because it is predictively optimal, see Berger and Barbieri (2004) for instance, and the references therein. However, our procedure can be applied to other model averaging strategies such as stacking, see Wolpert (1992).

Various authors have examined the details of model list formation for BMA. For instance, Raftery et al. (1997) and Hoeting et al. (1999) used a reversible jump MCMC procedure to generate a list of models on which they could apply an Occam's window approach by thresholding the posterior model weights. In a moderate to large sample context, Clarke and Clarke (2009) used a mixture of random search and backwards elimination to generate model lists that would be useful for prediction from the complexity standpoint.

One of the difficulties with using BMA is that at each stage a prior must be selected. Here, we have defaulted to the uniform prior because we have ensured our model lists

only contain models of comparable size and hence roughly comparable explanatory power. Moreover, our lists do not grow too fast in size since only one B_j is added at a time. In narrow contexts such as adding terms in relatively small well-behaved additive models, the dependence on the prior seems to matter little, Berger and Barbieri (2004) provides a related example. More generally, prior selection is a major problem because as $\mathcal{W}^{(t)}$ increases the tendency to dilution increases. Dilution is the problem that if the prior is spread over too many models that are relatively adequate then the prior probabilities can be so small as to give predictions that are all zero or underweight the best models severely. This phenomenon was revealed in the work of George (2000) and George and McCulloch (1993) who proposed priors to overcome it. A uniform prior on a model list of reasonable size avoids dilution in our examples.

Note that in the strict Bayes sense the overall procedure here is incoherent. However, the incoherency results from the reselection of the model list at each time step. We defend this on the basis that the model list is rechosen based on fit and the concept of fit is not part of the formal Bayesian axiomatization. Once a model list is chosen, however, we do use the BMA which is the Bayes optimal solution under squared error loss.

There are two benefits to our overall approach. First, unlike other model averaging methods we are searching over model lists not just reweighting the models on a fixed list. When the model list is allowed to change, performance can improve substantially. Our results show that the predictive error attributable to the model list, when it is poorly chosen, may be larger than any other source of error, a major weakness in static model averaging schemes.

Second, the search over model lists leads to a useful variance-bias tradeoff on the level of the model list paralleling the variance bias tradeoff for model selection or estimation. This means we can, in principle, identify optimal model lists to use in model averages. Thus, for realism, we have focused here on the case that the true function is expressed most parsimoniously in one basis but the basis used to form the model list is different.

The structure of this paper is as follows. In Sect. 2, we review background material on Bayes model averaging and basis search schemes so that our method can be presented in Sect. 3. Then, in Sect. 4, we present the details of implementation and a series of examples. In Sect. 5 we discuss the significance and implications of our results.

2 BMA and basis search

Specification of a BMA setting requires a collection of models, a prior for each model, and a prior over the class of models. To start, consider a single model of the form (1) with a sample $D^{(t)} = \{(\mathbf{x}_i, y_i), i = 1, \ldots, n_t\}$ of IID observations, and a subset $\mathcal{W}^{(t)} = \{B_{t1}, B_{t2}, \ldots, B_{tk_t}\}$ of k_t basis functions from $\mathbb{B}$. With ϵ_t following a normal distribution with mean 0 and variance σ^2 as assumed earlier, the normal linear model with k_t explanatory variables is

$$\boldsymbol{Y} = \mathbf{X}\beta + \epsilon, \tag{5}$$

where $\boldsymbol{Y} = (Y_1, \ldots, Y_{n_t})^{\mathsf{T}}$ with outcome $\boldsymbol{y} = (y_1, y_2, \ldots, y_{n_t})^{\mathsf{T}}$, $\beta = (\beta_0, \beta_1, \beta_2, \ldots, \beta_{k_t})^{\mathsf{T}}$, $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_{n_t})^{\mathsf{T}}$ and

$$\mathbf{X} = \begin{bmatrix} 1 & B_{t1}(\mathbf{x}_1) & B_{t2}(\mathbf{x}_1) & \ldots & B_{tk_t}(\mathbf{x}_1) \\ 1 & B_{t1}(\mathbf{x}_2) & B_{t2}(\mathbf{x}_2) & \ldots & B_{tk_t}(\mathbf{x}_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & B_{t1}(\mathbf{x}_{n_t}) & B_{t2}(\mathbf{x}_{n_t}) & \ldots & B_{tk_t}(\mathbf{x}_{n_t}) \end{bmatrix}. \tag{6}$$

The likelihood comes from $\boldsymbol{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n)$. If we use a mean zero normal prior on β with identity covariance and precision δ, then $\beta \sim \mathcal{N}(\mathbf{0}, \delta^{-1}\mathbf{I}_{k_t+1})$ then (5) is fully specified.

To extend this to a class of models fix k_t and let $\mathcal{M}$ be a collection of submodels M_α as in (5) but indexed by α. Each M_α corresponds to a selection of the k_t basis elements in $\mathcal{W}^{(t)}$ giving a design matrix $\mathbf{X}_\alpha$, with corresponding β_j's from β, denoted β_α, equipped with the isotropic prior above. Now, suppose a prior probability on $\mathcal{M}$ has been assigned so that $P(M_\alpha)$ is well-defined. Then, the BMA prediction scheme based on $\mathcal{M}$ can be specified as follows.

Let Y^{new} be the response corresponding to a new design point $\mathbf{x}^{\text{new}}$. The BMA prediction for Y^{new} is the posterior predictive distribution for an outcome of Y^{new}, y^{new}, given by

$$\boldsymbol{p}(y^{\text{new}}|\boldsymbol{y}) = \sum_{M_\alpha \in \mathcal{M}} \boldsymbol{p}(y^{\text{new}}|\boldsymbol{y}, M_\alpha)\mathbf{Pr}(M_\alpha|\boldsymbol{y}) \tag{7}$$

where the dependence on the design points has been suppressed in the notation. Now, the marginal posterior predictive density given model M_α is

$$\boldsymbol{p}(y^{\text{new}}|\boldsymbol{y}, M_\alpha) = \int \boldsymbol{p}(y^{\text{new}}|M_\alpha, \beta_\alpha)\boldsymbol{p}(\beta_\alpha|\boldsymbol{y}, M_\alpha)d\beta_\alpha, \tag{8}$$

and $\mathbf{Pr}(M_\alpha|\boldsymbol{y})$ is the posterior probability of model M_α, i.e,

$$\mathbf{Pr}(M_\alpha|\boldsymbol{y}) = \frac{\boldsymbol{p}(\boldsymbol{y}|M_\alpha)P(M_\alpha)}{\sum_{\alpha'} \boldsymbol{p}(\boldsymbol{y}|M_{\alpha'})P(M_{\alpha'})}, \tag{9}$$

in which

$$\boldsymbol{p}(\boldsymbol{y}|M_\alpha) = \int \boldsymbol{p}(\boldsymbol{y}|M_\alpha, \beta_\alpha)\boldsymbol{p}(\beta_\alpha|M_\alpha)d\beta_\alpha \tag{10}$$

is the marginal probability of the data under M_α. If we further assume that σ^2 and δ are known, then for our normal linear model, it is easy to show that

$$\boldsymbol{p}(\cdot\,|M_\alpha) \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_{n_t} + \delta\mathbf{X}_\alpha\mathbf{X}_\alpha^{\mathsf{T}}), \tag{11}$$

where $\mathbf{X}_\alpha$ is the design matrix for model M_α. Since we do not anticipate the availability of any information on individual models in $\mathcal{M}$, we assume a noninformative prior that puts equal mass on each model in $\mathcal{M}$. Thus, the model probabilities in (9) cancel out.

Each model M_α corresponds to a selection of $B_{t,j}(\mathbf{x})$'s for $j = 1, \ldots, k_t$. Since a given $B_{t,j}(\cdot)$ may occur in more than one model, in (7), the coefficients for each $B_{t,j}(\cdot)$ as α varies can be summed over $\mathcal{M}$ to simplify (7). The result is that the prediction $\hat{y}^{\mathsf{new}}$ for $\mathbf{x}^{new}$ from BMA with model list $\mathcal{M}$ is given by

$$\mathsf{BMA}(\mathbf{x}^{\mathsf{new}}, \mathcal{M}) \equiv \tilde{\beta}_0 + \sum_{j=1}^{k_t} \tilde{\beta}_j B_j(\mathbf{x}^{\mathsf{new}}),$$

where

$$\tilde{\beta}_j = \sum_{M_\alpha \in \mathcal{M}} \mathbf{Pr}(M_\alpha \,|\, \boldsymbol{y}) \mathbb{I}(B_j \in \mathcal{W}^{(t)}) \hat{\beta}_{\alpha,j}$$

and $\hat{\beta}_{\alpha,j}$ is the j-th element in

$$\hat{\beta}_\alpha = \left[\mathbf{X}_\alpha^{\mathsf{T}} \mathbf{X}_\alpha + \sigma^2 \delta \mathbf{I}_{k_t}\right]^{-1} \mathbf{X}_\alpha^{\mathsf{T}} \boldsymbol{y}.$$

Although assuming both σ^2 and δ are known is unrealistic, in practice these two parameters can be estimated accurately.

It remains to search over the B_j's to form good models and good lists of models. The earliest techniques for basis search came from the signal processing literature in which the set of functions used to express a waveform is often more general than a basis and is called a dictionary or frame. The use of dictionaries for signal representation began with the method of frames, MOF, Daubechies (1988). Another basis search method is Basis Pursuit, BP, developed by Chen et al. (2001), Chen et al. (1998) and Chen (1995). At root, BP seeks a representation of the signal i.e., approximation to the function, with coefficients having minimal ℓ^1 norm and the MOF seeks the analogous representation by enforcing minimal ℓ^2 norm on the coefficients.

3 The predictive method

Our method is sequential and has two main components. The first is a search over elements in $\mathbb{B}$ at each t to find $\mathcal{W}^{(t)}$ so that model lists can be generated. The second component is a simple random sampling procedure over classes of models formed from the admitted basis elements.

3.1 Our basis search method

Our method rests on successive reduction of distance between the target function and an emerging approximation. Each search can therefore be summarized as follows:

$$\text{select one or many } B_j \in \mathbb{B} \text{ such that } d\left(B_j, \boldsymbol{r}\right) < \tau,$$

where $d(\cdot, \cdot)$ is any suitable distance or dissimilarity measure, $\boldsymbol{r}$ is the residual function from fitting a model at time t, and τ is a threshold parameter controlling how closely one requires the candidate B_j to match the residual function $\boldsymbol{y}$. In our work, we used the norm

$$d\left(B_j, \mathbf{r}\right) \equiv \left\| \frac{B_j(\mathbf{x})}{\|B_j(\mathbf{x})\|} - \frac{\boldsymbol{r}(\mathbf{x})}{\|\boldsymbol{r}(\mathbf{x})\|} \right\|_p \tag{12}$$

as our distance measure where

$$\|g(\mathbf{x})\|_p \equiv \left(\int |g(\mathbf{x})|^p \, d\mathbf{x} \right)^{1/p}, \tag{13}$$

but is interpreted empirically. Explicitly, $\boldsymbol{r}$ is the vector $\boldsymbol{r} = (r(\boldsymbol{x}_1), \ldots, r(\boldsymbol{x}_n))$ and B_j is the vector $B_j = (B_j(\boldsymbol{x}_1), \ldots, B_j(\boldsymbol{x}_n))$. Thus, as before, the role of the design points is suppressed in the notation. Note that by normalizing B_j and $\boldsymbol{r}$ before computing the norm of their difference, we have $\tau \in [0, 2]$.

3.2 Initializing and updating of the process

At each time point t, we have a subset $\mathcal{W}^{(t)} \subset \mathbb{B}$ that contains all the basis elements contributing to an ever more accurate predictive approximation. We refer to $\mathcal{W}^{(t)}$ as the working basis. We consider two ways of initializing $\mathcal{W}^{(t)}$: (a) random (b) non-random. The simpler is random initialization which consists in *randomly* drawing one element from $\mathbb{B}$ so that $B^{\text{init}} \equiv$ one random draw from $\mathbb{B}$. Non-random initialization chooses from $\mathbb{B}$ the one element closest to the response variable, i.e.,

$$B^{\text{init}} \equiv \arg\min_{B_\gamma \in \mathbb{B}} d\left(\frac{B_\gamma}{\|B_\gamma\|}, \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|} \right). \tag{14}$$

Below, we show some computations using each technique.

We update the working basis $\mathcal{W}^{(t)}$ by adding only the best candidate within distance τ , i.e., we set

$$\mathcal{W}^{(t)} := \mathcal{W}^{(t)} \cup \left\{ B^{\text{best}} \right\}, \tag{15}$$

where

$$B^{\text{best}} \equiv \arg\min_{B_\gamma \in \mathbb{B}} \left\{ d(B_\gamma, \boldsymbol{r}) : d(B_\gamma, \boldsymbol{y}) < \tau \right\}.$$

An alternative is adding all the candidates within distance τ. For orthogonal bases these two updating techniques are equivalent. However, we add $_\gamma$'s one at a time to avoid admitting too many similar terms when the set of B_γ's is overcomplete and may give collinearity. It is important to avoid this here, since our method does not include pruning.

3.3 Selecting basis elements and forming lists

The first of the three bases we considered was the set of Fourier sine waveforms defined on $[0, \pi]$:

$$B_j(\mathbf{x}) \equiv \sin(j\mathbf{x}) \text{ with } \mathbf{x} \in [0, \pi].$$

The second was the full Fourier basis set on $[-\pi, +\pi]$:

$$B_j(\mathbf{x}) \equiv \sin(j\mathbf{x}) \text{ or } B_j(\mathbf{x}) \equiv \cos(j\mathbf{x}).$$

The third was the set of Chebyshev polynomials on $[-1, +1]$:

$$B_j(\mathbf{x}) \equiv \cos(j \arccos(\mathbf{x})) \quad \text{where } \mathbf{x} \in [-1, 1].$$

Once the set $\mathbb{B}$ is chosen, the sequential process initializes the working basis $\mathcal{W}^{(t)} \subset \mathbb{B}$, and subsequent iterations update $\mathcal{W}^{(t)}$ using the residuals and the search method described earlier. Our process therefore implements a sort of automated residual analysis to improve the approximation sequentially.

With $\mathcal{W}^{(t)}$, one can generate a working model space with up to $2^{k_t} - 1$ models, where $k_t = |\mathcal{W}^{(t)}|$. For small values of τ, it is very likely that k_t will also be small, and all the relatively few models in the model space may contribute to the approximation. One might therefore be willing to retain the whole set of $2^{k_t} - 1$ models of the model list. In such cases, model list selection reduces to efficient search of the original basis set $\mathbb{B}$.

As τ gets larger, however, many of the B_γ's added are likely to contribute very little to bias correction while inflating the variance, thereby causing the prediction error to increase. When k_t gets really large as a result of a large τ, the explosive number of possible models $2^{k_t} - 1$ makes computation prohibitive and leads to inaccuracies from round-off errors. It therefore makes sense to find ways to select only a subset of $\mathbb{M}$. Considering the fact that the working basis $\mathcal{W}^{(t)}$ consists of screened basis functions that are deemed close enough to the true underlying target, we have used a random selection of a proportion μ of models from the explosive list of $2^{k_t} - 1$ models. Clearly, the first benefit here is the computational convenience although the logical justification seems solid as well. Heuristically, one could propose setting

$$\mu \equiv 0.95 - \tau/2,$$

when it is positive. This means that as the model space grows larger (as measured by τ), the proportion of models drawn from it to form the model list shrinks to guarantee

the model list remains computationally manageable. (We ignore problems when τ is close to 2 since such lists are formed using models with high errors.) Typically we chose values $\mu = 0.1, 0.25$. The exact value of μ seemed to make little difference qualitatively.

Since we are searching for optimal model lists, we define model lists with different sizes and complexities. Consider three model averaging strategies (AS), based on their model lists:

$$\mathsf{AS} \in \{\mathsf{small}, \mathsf{medium}, \mathsf{large}\}$$

- small: all models of size 1, 2 or 3.
- medium: all models of size $k_t/2$
- large: all models of size k_t, $k_t - 1$, or $k_t - 2$.

These model lists are ranked in order of increasing complexity, or size of their elements. It is seen that the first and third are the same size while the second is larger. The interplay between size of list and the complexity of models on it is seen in the computed results of the next section.

3.4 Term formation and overcompleteness

Clearly, sine and full Fourier are qualitatively similar as they are both trigonometric function sets. Also, as polynomials, Legendre and Chebyshev are qualitatively similar. We focused on these two classes, trigonometric and polynomial, and we explored the effect of combining basis sets from them. This allowed us to assess the gains derived from this type of overcompleteness in our context. We also considered a second type of overcompleteness, the formation of frames from complete sets (bases). We did this by extending the given basis with a few new elements formed as partial sums of its elements. Recall that frames may contain elements that, when taken together, are linearly dependent or in which a sum of elements may be a good approximation for another element, a sort of near overcompleteness. To form these frames, we defined three kinds of term formation (TF) strategy. They are:

1. $\mathsf{TF} = 1$: Use $\mathcal{W}^{(t)}$ exactly as it is, i.e, no addition (no assessment of overcompleteness)
2. $\mathsf{TF} = 2$: $\mathcal{W}^{(t)} := \mathcal{W}^{(t)} \cup \{\text{a fraction of sums of pairs from} \mathcal{W}^{(t)}\}$
3. $\mathsf{TF} = 3$: $\mathcal{W}^{(t)} := \mathcal{W}^{(t)} \cup \{\text{a fraction of sums of triples from} \mathcal{W}^{(t)}\}$

For $\mathsf{TF} \in \{2, 3\}$, the number of terms added to $\mathcal{W}^{(t)}$ can quickly became explosive. We therefore introduce an user defined extra parameter ν to control the proportion of terms added. We chose $\nu = 0.01, 0.05$. As with μ we do not comment further on ν because its value made little difference qualitatively; its main role was to make computations evaluating over-completeness regimes feasible.

3.5 The role of τ

The parameter τ indexes how easy it is to add new terms that could improve the approximation. Therefore τ controls both the quality, and the size of the model lists generated from $\mathcal{W}^{(t)}$. In fact, as τ increases, our model average involves more and more models because we have not imposed any parsimony. Using τ to characterize $\mathcal{M}$, we restate the goal of Eq. 3 as

$$\tau^{\mathsf{opt}} = \arg \min_{\tau \in [0,2]} \mathsf{PMSE}(\tau). \tag{16}$$

4 Details of implementation and numerical results

Our procedure has six overall inputs. First, a target function must be given. In practice, the investigator does not know this. Here we will choose four: a Hill function (in a sine basis) that looks like a rolling hill and should be easy, Valley function (in a Fourier basis) that has a pronounced minimum, a Mexican Hat function (polynomial and exponential) which has three regions of high curvature, and a Tooth function (linear plus exponential) which has a strong single mode on an incline. These are in increasing difficulty.

Second, data to predict must be generated. In all our examples below, we have used a noise variance $\sigma^2 = (0.2)^2$. We chose this value because it ensured a good trade-off between identifying the target function and retaining enough randomness. Thus, in the limit of many runs of large length the lower bound for the average prediction error is $\sigma^2 = .04$. Also, our focus in the small sample case, so we took outcomes in batches of 5 per time step and limited our computations to 10 time steps at most, a total of 50 data points.

Third, an ensemble of functions must be chosen. Here we consider four cases. Three are bases: Fourier, Chebyshev, sine, representing trigonometric functions and polynomials. The fourth ensemble is the union of Fourier and Chebyshev. Fourth, we must choose one of the three TF's to decide which functions are weighted by parameters. Fifth, we must choose one of the three AS's to decide which terms get combined. Finally, we must choose a value of $\tau \in [0, 2]$.

We have the choice of starting randomly or by initializing the procedure by selecting a certain number of ensemble elements based on the initial data that are closer than randomly chosen elements would be on average. We prefer to choose the best element from the frame to initialize the predictive process because this is consistent with how later elements are added.

Given the ensemble and initialization, our predictive procedure applies the chosen TF and AS strategies. This gives our predictor for the data points from the next time step. The five residuals from the next prediction stage get used in the iterative procedure to choose the model list for generating the next predictor. Thus, by repeating the procedure numerous times and averaging the predictive performance from the runs of length 50, we can track how model uncertainty affects the average prediction error.

Specifically, we generate two kinds of performance graphs. The fist kind of performance graph is for a fixed τ to see how the APE decreases from time step to time step

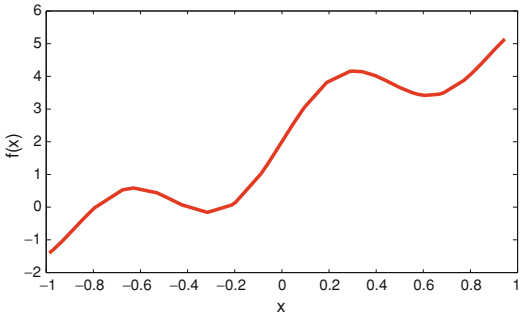
to give the terminal APE. The second kind of performance graph gives the approximate final Average Prediction Error, APE, for a given value of τ. After obtaining the second kind of performance graphs, the first kind of performance graphs for the τ's that gave the maximum and minimum terminal APE could be found and examined.

The running times for the graphs given below depend heavily on the system used to do the computing. On a Pentium 4 with 1.7 Ghz of CPU speed and 1GB of RAM and 80 GB HD, the simplest case ($m = 5$, and $\mathsf{AS} = \mathsf{TF} = 1$) took 17 min to complete (with μ, ν in the range of 0.03—0.15). The same case but with $m = 10$ required 30 min However, on a 1.8 GHz (Intel Xeon) Dual Processor with 512 MB RAM and 80 GB HD, the longest running cases ($m = 10$, $\mathsf{AS} = \mathsf{TF} = 2$) took under five minutes. On currently standard equipment (e.g., Intel Core-2-quad, cost USD $1500) we conjecture the longest running times would be even less.

4.1 The Hill function

As a first example, let f^* be the Hill function given in the sine basis by

$$f^*(\mathbf{x}) = 2 + 2\sin(\mathbf{x}) + 1.25\sin(2\mathbf{x}) + \sin(7\mathbf{x}) \text{ on } [-1, 1].$$



First, we use the correct, sine, basis for prediction. It is usually unrealistic to assume knowledge of the correct basis for uncovering a function, however, we start with this case for comparison purposes. For each of the nine (TF, AS) pairs, we give the performance graphs of the second kind and then the performance graphs of the first kind for the best and worst τ's. Then we give the corresponding graphs using model averages formed from the polynomial Chebyshev basis.

4.1.1 The Hill function sine *basis*

In Fig. 1, the columns correspond to TF = 1, 2, 3 for AS = 1. The first row shows the performance graph of the second kind; the second row shows the performance graphs of the first kind for the τ achieving the knee APE. The first row shows that the graphs do not depend on the TF strategy. This suggests that term formation does not affect the variance bias tradeoff in this predictive setting. It is seen that the APE over time decreases for the best τ; if the corresponding performance graphs of the first kind were plotted for the worst τ they are more erratic, sometimes decreasing, sometimes

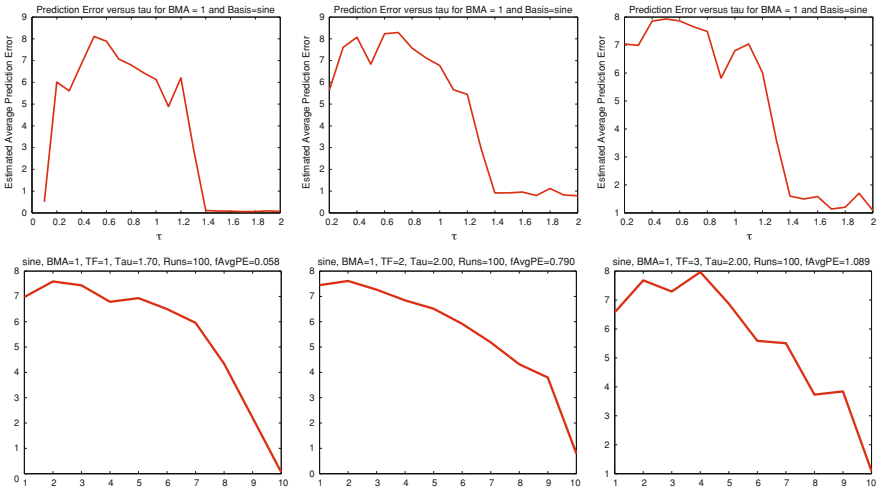


Fig. 1 Hill function in the sine basis with AS = 1; the knee value of τ was around 1.4 in all three cases

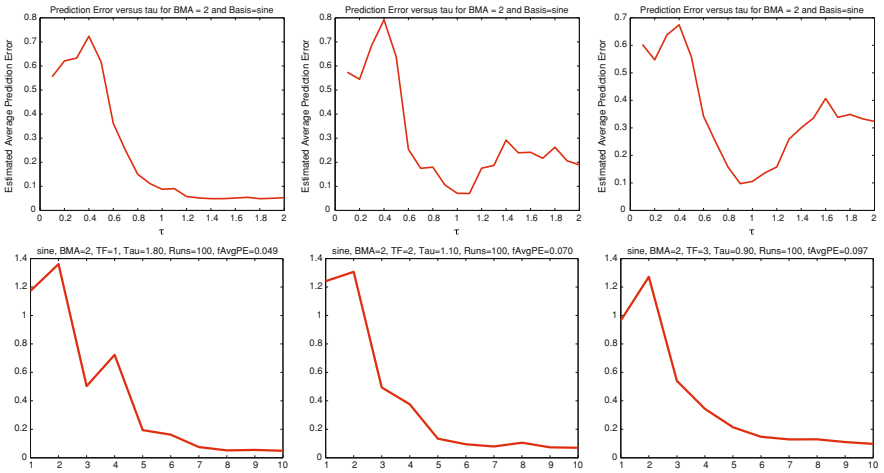


Fig. 2 Hill function in the sine basis with AS = 2; the knee for the first panel was 1.2

increasing, and sometimes just unstable. This may mean that bad τ's permit good models and bad models indiscriminately.

It is seen that the performance curves of the second kind decrease from a peak to a minimizing value and stay there as τ increases. This is a degenerate V in which the right arm increase does not happen because the models are small. Such models do not tend to overfit.

We comment that the rapid increase in APE for small values of τ seen in some of the performance graphs of the second kind, here and in Figs. 2 and 3, is the consequence of random initialization. That is, when we used a random initialization, sometimes we got a rapid increase in APE as the procedure locked onto models that gave improved

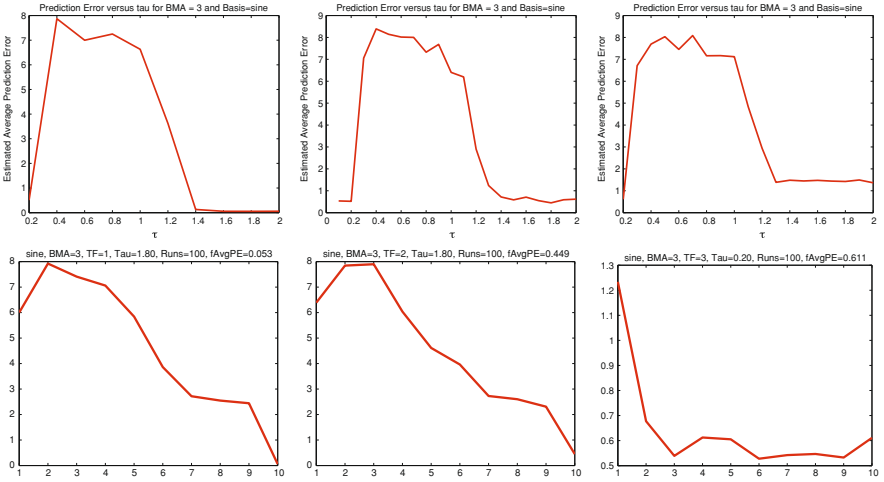


Fig. 3 Hill function in the sine basis with AS = 3; knee values around 1.4, 1.4 and 1.3

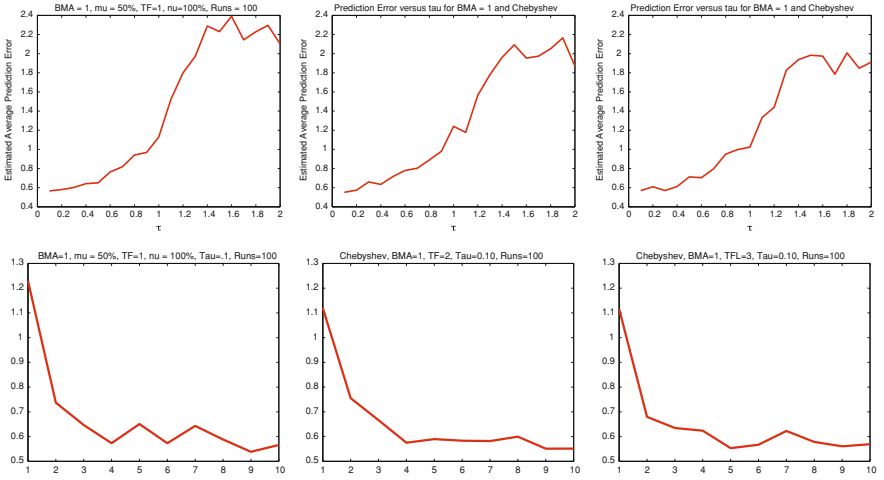


Fig. 4 Hill function in the Chebyshev basis with AS = 1

predictions. In Figs. 4 and 5 where we used non-random initialization, this increase does not appear.

In Fig. 2, the columns correspond to TF = 1, 2, 3, as before, but for AS = 2, medium sized models. Again, the first row shows the performance graph of the second kind; the second row shows the performance graphs of the first kind for the τ achieving the knee or least final APE.

The first row shows a dependence on TF. As TF increases, the strength of the V-formation increases. This is the only case we found where the value of TF affected the results. We suspect this is not random variability because the model class for AS = 2 has mid-sized models: They have $k/2$ terms midway between the fewest terms, k=1, 2, 3 and the maximal numbers of terms k, $k-1$, $k-2$. Since there are more mid-

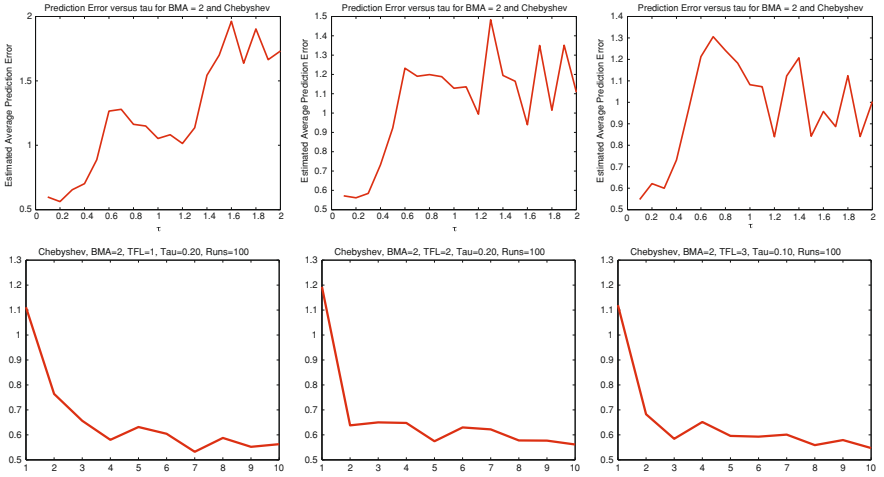


Fig. 5 Hill function in the Chebyshev basis with AS $= 2$

sized models than large or small ones, and the basis is the same as that of the target function, it may be that term formation permits a faster search of the models, as expected from using overcompleteness. In this case, the APE over time decreases for the worst τ's (not shown) as well as the best ones.

In Fig. 3, the columns correspond to TF $= 1, 2, 3$, as before, but for AS $= 3$. The first row shows the performance graph of the second kind; the second row shows the performance graphs of the first kind for the τ achieving the knee or least final APE.

Like Fig. 1, the first row shows no dependence on TF and the performance curves of the first kind decrease from a peak to a minimizing value and stay there as τ increases. As before, the right arm of this degenerate V does not increase. This may be due to fitting large models with complicated terms: These models fit data readily and may not overfit because the basis is correct. Averaging over models that do not overfit will not give errors typical of overfitting. Also as before, we see a smooth decrease in the row for the best τ's.

Aside from the dependence on TF in Fig. 2 in the performance curves of the second kind that is not seen in the rows of Figs. 1, and 3, the other striking finding here is that the curves in Fig. 2 are qualitatively different from those in Figs. 1, and 3. Specifically, as TF increases in Fig. 2, the full V-formation becomes apparent. This is strongest when the model list is large AS $= 2$ as opposed to AS $= 1, 3$ and the terms are most complex.

4.1.2 The Hill function in Chebyshev *basis*

Here, we have redone the computations from the previous subsection using the Chebyshev basis in place of the sine basis. We have also used non-random initialization on the working basis. The format of the figures is otherwise the same.

In Fig. 4 (AS $= 1$), TF does not appear to make a difference. The common appearance of the performance graphs of the first kind is increasing. This means that many small models in the wrong basis may approximate the target function well but that so

many are required the variance is large; a small number of small models actually does best. This is a degenerate V in which the left arm is flat because the bias is not high enough for small τ's. As before, the best τ's show a decrease in APE. Although the graphs are not shown, the performance curves of the first kind for the worst τ's are smoothly increasing meaning that good models are essentially never found.

In Fig. 5 ($\mathsf{AS} = 2$), TF does not appear to make a difference. The common appearance of the performance graphs of the second kind is increasing, qualitatively similar to Fig. 4 probably with the same interpretation. As τ gets larger, note that there is instability in the curve: We attribute this to the richness of the model space making the search for models as τ increases very rough. It is unclear how reliable these values are; this issue is taken up at the end of this section. As before, the best τ's show a decrease in APE and are quite small. The performance graphs of the first kind for the worst τ's (not given) show a decrease and then an increase in APE over time, indicating an early decrease in bias and a later increase in variance. Since it happens in all three cases, we do not attribute this to the TF strategy. We regard this V-formation purely as a result of cumulative model overfitting from the list.

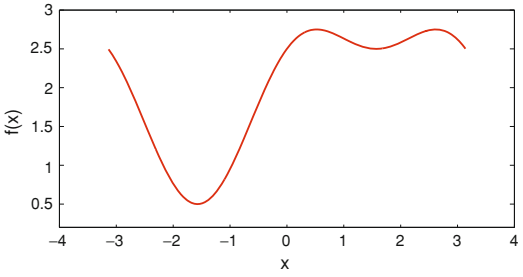
When the Chebyshev basis is used with $\mathsf{AS} = 3$, the results are qualitatively the same as for the case $\mathsf{AS} = 2$, but surprisingly cleaner. A stronger V-formation is observed (with instability for large τ) in the performance graphs of the second kind. It may be that because the models for $\mathsf{AS} = 3$ are large, though not as numerous as for $\mathsf{AS} = 2$, each model on the list may be big enough to provide good approximation. Thus, as more models are used, APE decreases in the performance graphs of the second kind until problems with overfit emerge: Overfit from individual models is leading to overfitting of the model list. The performance graphs of the first kind admit interpretations as before.

4.2 The Valley function

The function

$$f^*(\mathbf{x}) = 2 + \sin(\mathbf{x}) + 0.5\cos(\mathbf{x}) \quad \mathbf{x} \in [-\pi, \pi]$$

shown below, is expressible with finitely many terms from the Fourier basis. We have included it for contrast with the Hill function expressible in the sine basis.



The results for the Valley function in the Fourier basis are shown in the first five panels of Fig. 6. (We have only shown five of the nine possible graphs for brevity.)

Somewhat like the Hill function case, the V-formation strengthens as either the model list size or the term complexity increases. That is, for $\mathsf{AS} = 1, 3$ and $\mathsf{TF} = 1$ we get a degenerate V that does not increase as τ gets close to 2. But as the model list increases, $\mathsf{AS} = 2$, or term complexity increases, $\mathsf{TF} = 3$ the V-formation strengthens. This is borne out in the sixth panel using $\mathsf{AS} = 3$ and $\mathsf{TF} = 1$ which essentially only decreases. (Any increase for large τ is slight.)

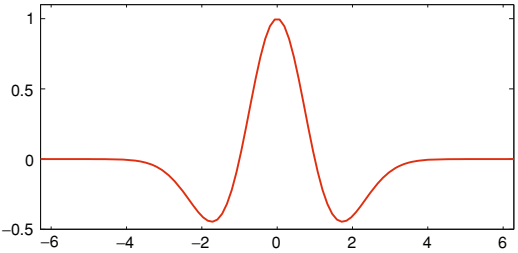
So, far the examples suggest that full V-formation only occur when the models or model list is sufficiently complex. This tends to be the case in the examples below, where we focus more on the mismatch between the modeling basis and the target function.

4.3 The Mexican Hat function

The symmetric function

$$f^*(\mathbf{x}) = (1 - \mathbf{x}^2)\exp(-0.5\mathbf{x}^2) \qquad \mathbf{x} \in [-2\pi, 2\pi]$$

shown below, is not expressible with finitely many elements from the bases here.



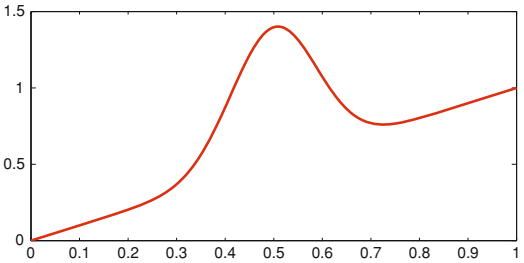
The results for the Mexican Hat are shown in Fig. 7. Rows correspond to the bases Chebyshev and Fourier; columns correspond to $\mathsf{AS} = \mathsf{TF} \in \{1, 2, 3\}$. We have set these two equal for this case because the TF value either strengthened the V-formation (when it was large) or did did not appear to make much difference.

Only performance curves of of the second kind for APE for Chebyshev over $\tau \in [0, 2]$ are shown. It is seen that these performance graphs exhibit, to varying degrees, the expected V-formation. Indeed, the strength of the V-formation increases from left to right as expected. Although not shown, when Fourier is used, the results are qualitatively similar, and when the two bases are combined small and large models evidence a weak V, while mid-size models do not appear able to discriminate over model richness as summarized by τ.

4.4 The Tooth Function

The function

$$f^*(\mathbf{x}) = \mathbf{x} + \frac{9}{4\sqrt{2\pi}} \exp\left[-4^2(2\mathbf{x} - 1)^2\right] \qquad \mathbf{x} \in [0, 1]$$



shown below, is not expressible with finitely many elements from the bases we have consid ered here. In addition, it is asymmetric. However, unlike the earlier examples it is localized.

The results for the tooth are shown in Fig. 8. The row corresponds to the Chebyshev basis; columns correspond to $\mathsf{AS} = \mathsf{TF} \in \{1, 2, 3\}$. All are performance curves of

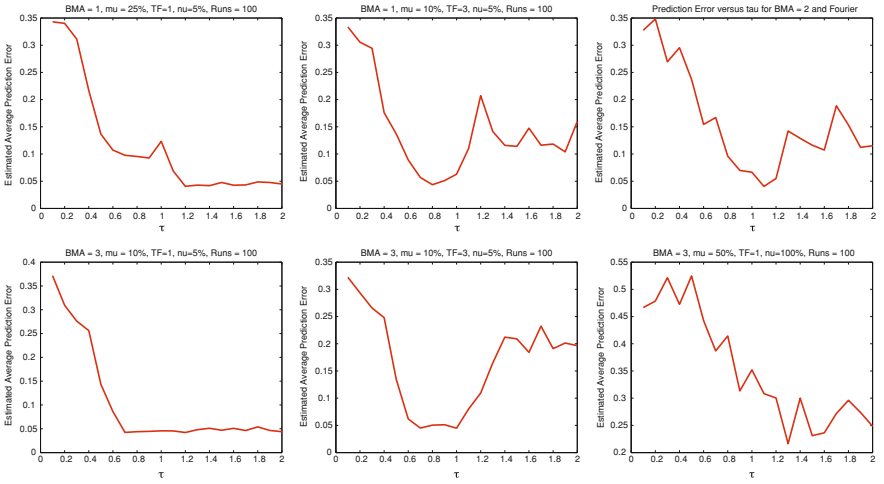
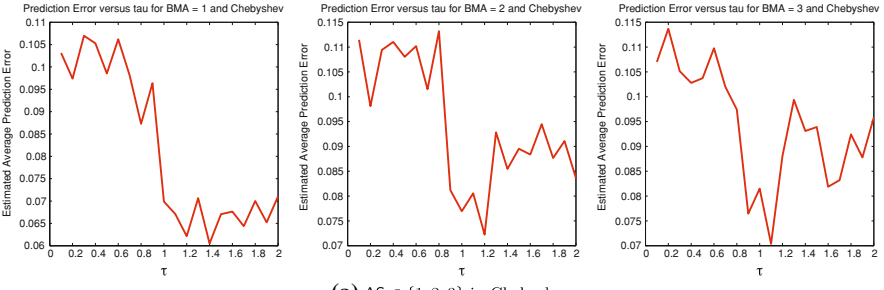
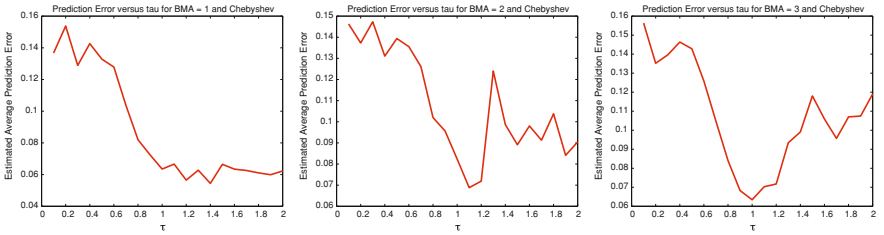


Fig. 6 Second performance curves for the Valley function. The first two plots have $\mathsf{AS} = 1$ and $\mathsf{TF} = 1.3$. The third has $\mathsf{AS} = 2 = \mathsf{TF}$. On the second row, the first two have $\mathsf{AS} = 3$ and $\mathsf{TF} = 1.3$, all in the Fourier basis. The last plot has $\mathsf{AS} = 3$ and $\mathsf{TF} = 1$ using Chebyshev



(a) $\mathsf{AS} \in \{1, 2, 3\}$ in Chebyshev

Fig. 7 Mexican Hat function in the Chebyshev and Fourier bases



(a) AS $\in \{1, 2, 3\}$ in Chebyshev

Fig. 8 Tooth function in the Chebyshev basis

the second kind for APE over $\tau \in [0, 2]$. The results for the Fourier basis and the combined Chebyshev–Fourier basis are qualitatively similar to the Chebyshev basis.

It is seen that these performance graphs exhibit, to varying degrees, the expected V-formation. Again, the V-formation strengthens as AS increases; the value of TF either strengthened the V-formation (when it was large) or did did not appear to make much difference.

For all bases, the AS = 1 column showed a degenerate V that does not increase as τ gets close to 2. This may indicate that for highly localized functions such as the tooth, small models only fit well when it is easy to add terms. However, moderate sized models and large lists, or large models even if fewer (AS = 3) lead to clear V-formations. In this example, the combined basis did not lead to instability and higher error, but the best predictive error of the combined bases in (c) did not out perform the individual bases in (a) and (b).

4.4.1 Summary

Overall, the graphs presented here look somewhat rough or choppy. There are two reasons for this. First, the results from some test cases show that many of the rough parts of the graphs would smooth out if the number of iterations were increased enough. We have not done this because we wanted to see the effect of the running time in the accuracy. Second, the matrix manipulations to obtain BMA predictions involve large matrices. So, small approximation errors can become self-reinforcing. Even with very high precision computing, model averages can be unstable because slight changes to models can give big differences in predictive performance.

An intriguing possibility is that, in some cases, the roughness of the curves may in fact be real. Consider the Valley function with $AS = TF = 1$. The top two panels of Fig. 9 show the same computation as in the top left panel of Fig. 6, but done on different grids for τ. It is seen that the spike near 1 shifts to about 1.3, but the spike remains. There genuinely seems to be a secondary peak, that may indicate some unexpected property of the approximation process. For instance, the search for a good model list, as a function of τ, generally finds improvement as τ increases except that near $\tau = 1.3$ the search has unfortunately climbed a hill in model space before resuming its descent.

Thus, apart from accidental hill climbing on the least final APE surface (as a function of the model lists), every performance graph of the second kind we have shown

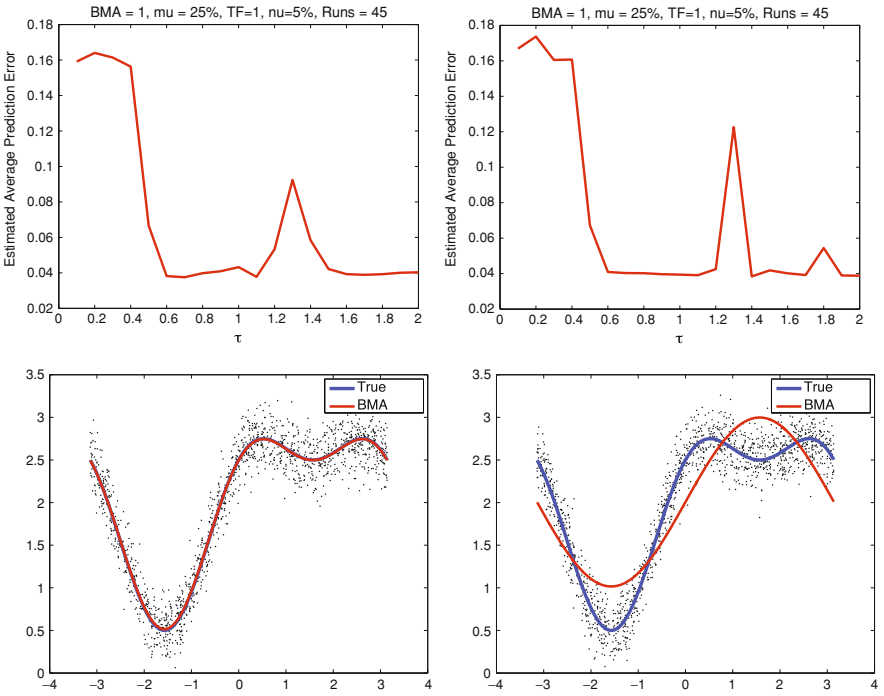


Fig. 9 Valley function in Fourier, TF = AS = 1. The *top row* is two further versions of the second performance curve. The APE as a function of the models may be quite rough. The *bottom row* shows the fit from the best value of τ and the fit from the worst value of τ. The curve on the left drops quickly to the error variability 1/4. The curve on the right is erratic due to the instability of predictive error

corresponds to a V formation, or a degenerate V formation. For instance, a decreasing pattern down to a minimal value at which the curve is constant only occurs for small model lists; AS $= 1$ and AS $= 3$ for instance are smaller lists than AS $= 2$. (The models in AS $= 3$ are larger than in AS $= 1, 2$ but fewer in number.) Thus, all decreasing patterns occur with them, except for Hill in its own basis with TF $= 1$ which we explain by overcompleteness. Hill, in its own basis with AS $= 3$, is a bit of an exception in the sense that when we used other functions we found a proper V-formation as expected. This partial exception may reflect that the sine basis is parsimonious for hill and so stabilizes effectively.

An increasing pattern often occurs when the basis used is wrong. It is exacerbated when the models are relatively small, AS $= 1$, and sometimes when AS $= 2$. Both situations make it difficult to include as many terms as necessary for good function approximation.

A complete V formation tends to appear weakly with AS $= 2$, and strongly with AS $= 3$. This is typically enhanced by larger values of TF. The exceptions are hill in its basis which we have already discussed and the MexicanHat with AS $= 3$ when Fourier and Chebyshev are combined. This latter case may be a very weak V, but combining bases may reduce bias so quickly that it seems constant initially, until increasing variance causes the rise in APE.

5 Implications for model list selection

Overall, our results show that, for well-behaved target functions, e.g., Hill and Valley, larger model lists or model lists with bigger models having more complex terms, tend to give model averages that have bias when they do not fit well and excessive variance when the full size of the model list is used for prediction. That is, strong clear V-formations in the second performance curves tend to be associated with models or model lists that are complex. Our results also suggest that there will be instability if an excessively rich dictionary is used and that otherwise there is little predictive advantage in large dictionaries.

A caveat to this is that when the basis used to form the BMA is quite different from the basis in which the target function is parsimoniously expressed it my be useful to average over more complex models, possibly with more complicated terms. Otherwise, as seen in the Hill with Chebyshev example AS $= 1, 2$ smaller models tend to do best. In these cases, it is possible that the roughness of the first performance curves accurately reflects the search over τ. In terms of the predictive accuracy correct or nearly correct model lists, as determined by the optimal τ, give performance curves of the first kind that decrease and are relatively smooth.

When target functions are not so well behaved, e.g., the MexicanHat and Tooth examples, having, for instance, high curvature, it is easy for a mismatch of bases to lead to either instability (too rich a class of approximands) or to the preference for many models when the average is formed from individual terms.

Strong advocates of parsimony would conjecture that when the target function is a finite sum of basis elements, best predictive results would be obtained when the approximating basis is the same as the target function basis. In fact, our computations, on balance, do not appear to support this, except for two special cases noted next. Aside from these, the distinction $f^* \in \mathsf{span}\mathbb{B}$ versus $f^* \notin \mathsf{span}\mathbb{B}$ does not seem to affect predictive error when enough basis elements can be added so the approximation error is smaller than the variation in the data.

The two special cases where basis matching between target and approximation seems to be predictively helpful occur for very rich classes of models and for lists of very small models. Indeed, with rich model classes, when $f^* \in \mathsf{span}\mathbb{B}$ some overcompleteness may be helpful. With lists of small models the bias variance tradeoff is affected by $f^* \notin \mathsf{span}\mathbb{B}$. In particular, when $f^* \notin \mathsf{span}\mathbb{B}$ small models in the wrong basis cannot approximate it well.

As a generality, approximation methods perform better when there is a mechanism for removing basis elements or models that prove to be of little or no use. However, this is most important when the sample size is large because otherwise, as n increases, the chance of including poor terms increases. For smaller sample sizes such as those we have used here, we suggest pruning would be of little benefit predictively. In larger sample settings, pruning would likely only be helpful past a specific sample size dependent on the target function and basis used. In effect, therefore, we have addressed the more limited goal of searching for the first model lists that were big enough for good prediction rather than genuinely optimal. Nevertheless, we have demonstrated that model list search and consequent uncertainty is an essential component of a comprehensive uncertainty analysis.

References

Alonso A, Pena D, Romo J (2006) Introducing model uncertainty by moving blocks bootstrap. Stat Papers 47:167–179
Berger J, Barbieri M (2004) Optimal predictive model selection. Ann Stat 32:870–897
Chen SS (1995) Basis pursuit. Ph.D. thesis, Department of Statistics, Stanford University. http://www-stat.stanford.edu/schen
Chen SS, Donoho DL, Saunders MA (1998) Atomic decomposition by basis pursuit. SIAM J Sci Comput 20:33–61
Chen SS, Donoho DL, Saunders MA (2001) Atomic decomposition by basis pursuit. SIAM Rev 43:129–159
Clarke J, Clarke B (2009) Prequential analysis of complex data. Stat Anal Data Min (to appear)
Czado C, Raftery A (2006) Choosing the link function and accounting for link uncertainty in generalized linear models using Bayes factors. Stat Papers 47:419–442
Daubechies I (1988) Time-frequency localization operators: a geometric phase space approach. IEEE Trans Inf Theory 34:605–612
Dawid AP (1984) Present position and potential developments: some personal views. Statsitical theory. The prequential approach. J R Stat Soc B 147:278–292
Dawid AP, Vovk V (1999) Prequential probability: principles and properties. Bernoulli 5:125–162
Draper D (1995) Assessment and propagation of model uncertainty. J R Stat Soc B 57:45–97
George EI (2000) The variable selection problem. J Am Stat Assoc 95:1304–1308
George EI, McCulloch RE (1993) Variable selection via Gibbs sampling. J Am Stat Assoc 88:881–889
Gustafson P, Clarke B (2004) Decomposing posterior variance. J Stat Plan Inference 119:311–327
Hjort NL, Claeskens G (2003) Frequentist model average estimators. J Am Stat Assoc 98:879–899
Hoeting JA, Madigan D, Raftery AE, Volinsky CT (1999) Bayesian model averaging: a tutorial. Stat Sci 14:382–401
Kass R, Raftery A (1995) Bayes factors. J Am Stat Assoc 90:773–795
Leamer E (1978) Specification searches, ad hoc inference with nonexperimental data. Wiley, New York
Minka T (2000) Bayes model averaging is not model combination. Technical report. http://citeseerx.ist.psu.edu
Raftery AE, Madigan D, Hoeting JA (1997) Bayesian model averaging for linear regression models. J Am Stat Assoc 92:179–191
Toutenberg H, Shalabh (2002) Prediction of response values in linear regression models from replicated experiments. Stat Papers 43:423–433
Wolpert D (1992) Stacked generalization. Neural Netw 5:241–259
Wong A (1995) An averaging approach to prediction. Stat Papers 36:253–264