Machine Learning, 48, 137–163, 2002

Structural Modelling with Sparse Kernels

S.R. GUNN s.r.gunn@ecs.soton.ac.uk
J.S. KANDOLA jsk97r@ecs.soton.ac.uk
ISIS Research Group, Department of Electronics and Computer Science, University of Southampton, UK

Editors: Yoshua Bengio and Dale Schuurmans

Abstract. A widely acknowledged drawback of many statistical modelling techniques, commonly used in machine learning, is that the resulting model is extremely difficult to interpret. A number of new concepts and algorithms have been introduced by researchers to address this problem. They focus primarily on determining which inputs are relevant in predicting the output. This work describes a transparent, advanced non-linear modelling approach that enables the constructed predictive models to be visualised, allowing model validation and assisting in interpretation. The technique combines the representational advantage of a sparse ANOVA decomposition, with the good generalisation ability of a kernel machine. It achieves this by employing two forms of regularisation: a 1-norm based structural regulariser to enforce transparency, and a 2-norm based regulariser to control smoothness. The resulting model structure can be visualised showing the overall effects of different inputs, their interactions, and the strength of the interactions. The robustness of the technique is illustrated using a range of both artifical and "real world" datasets. The performance is compared to other modelling techniques, and it is shown to exhibit competitive generalisation performance together with improved interpretability.

Keywords: Kernel methods, transparency, model interpretability, sparse structure, ANOVA

1. Introduction

The problem of empirical data modelling is germane to many applications. In empirical data modelling a process of induction is used to build up a model of a system from examples. Ultimately the quantity and quality of the observations will govern the performance of this model. However, the choice of modelling approach will also influence the performance of the model. By its observational nature, data obtained is finite and sampled; typically this sampling is non-uniform and due to the high dimensional nature of the problem, the data will form only a sparse distribution in the input space. Consequently, the problem is nearly always ill-posed (Poggio, Torre, & Koch, 1985) in the sense of Hadamard (1923). To address the ill-posed nature of the problem it is necessary to convert the problem to one that is well-posed. For a problem to be well-posed, a unique solution must exist that varies continuously with the data. Conversion to a well-posed problem is typically achieved with some form of capacity control, which aims to balance the fitting of the data with constraints on the model flexibility, producing a robust model that generalises successfully. Previous approaches to restoring the well posedness have included regularisation methods (Tikhonov, & Arsenin, 1977). In this paper, the method chosen is based around kernel methods due to their rigorous formulation and good generalisation ability for small sample sizes. Girosi (1997) and Smola (1998) have shown that kernel methods can be placed in a regularisation framework, which guarantees their well posedness; Support Vector Machines (SVMs) (Vapnik, 1995) and Gaussian Processes (GPs) (Rasmussen, 1996) are examples. For tutorial introductions to SVMs see Burges (1998), Cristianini and Shawe-Taylor (2000) or Gunn (1998). Given a dataset, $\mathcal{D} = \{(x^1, y^1), \ldots, (x^l, y^l) \mid x^i \in \mathbb{R}^d, y^i \in \mathbb{R}\}$, the model is a weighted linear summation of kernels,

$$f(x) = \sum_{i=1}^{l} \alpha_i K(x^i, x), \tag{1}$$

where these kernels are 'centred' on the data points. Consequently, the solution is opaque due to the large number of terms that will typically exist in this expansion. Furthermore, the multivariate basis functions can be difficult to interpret. The number of terms in the expansion can be reduced in some circumstances by enabling a proportion of the kernel multipliers to become zero. This can be achieved using a loss function that has a 'dead-zone', such as an ϵ-Insensitive loss function (Vapnik, 1995).

Whilst a predictive model may be the ultimate goal of modelling, it is often desirable and sometimes even essential to be able to interpret the final model structure. This is especially true in medical domains, where *black-box* models, such as traditional neural networks, bagged descision trees as well as kernel methods, are viewed with great suspicion (Wyatt, 1995; Plate, 1999). In situations where model interpretation is important, many researchers revert to using simpler, but more interpretable modelling methods, for example logistic regression. As Plate observes (Plate, 1999) there is a danger in using such simple models, since they typically suffer from the problem of model mismatch, and hence they may fail to discover an important relationship in the data because they lack the flexibility to model it. In this work we introduce interpretability, or *transparency*, by producing a parsimonious model, which has a sparse structural representation, but is flexible enough to avoid problems of model mismatch. The transparency is beneficial in that it enables the model to be validated and interpreted. Features that aid transparency are input selection and ways of decomposing the model into smaller more interpretable pieces that can be easily visualised. To address this issue we introduce a modified kernel model of the form,

$$f(x) = \sum_{i=1}^{l} \alpha_i \sum_{j} c_j K_j(x^i, x), \quad c_j \geq 0, \tag{2}$$

where the kernel is replaced by a weighted, c_j, linear sum of kernels, K_j. Transparency can then be introduced by a careful choice of the additive kernels, K_j and by making their weighting coefficients, c_j, sparse. In this paper we focus on the integration of an ANOVA (ANalysis Of Variance) representation to provide a transparent approach to modelling. ANOVA kernels (Stitson et al, 1999) have previously been used with SVMs, with promising performance. However, the difference here is to develop a technique that will select a sparse ANOVA kernel producing strong transparency. The ANOVA representation is motivated by the decomposition of a function into additive components, with the goal of representing the function by a subset of the terms from this expansion. A function may be decomposed into

$$f(x) = f_0 + \sum_{i}^{d} f_i(x_i) + \sum_{i<j}^{d} f_{i\otimes j}(x_i, x_j) + \cdots + f_{1\otimes 2\otimes \ldots \otimes d}(x), \tag{3}$$

where d is the number of inputs, f_0 represents the bias and the other terms represent the univariate, bivariate, etc., components. The notation x_i denotes the scalar value of input i. The basis functions are semi-local and are similar to the approaches used by Friedman (1991) in the Multivariate Adaptive Regression Splines (MARS) technique and in the Adaptive Spline Modelling of Observational Data (ASMOD) technique (Kavli & Weyer, 1995). The additive representation is advantageous when the higher order terms can be ignored, so that the resulting model is represented by a small subset of the ANOVA terms, which may be easily visualised. This produces a transparent model, in contrast to the majority of neural network models, providing the modeller with structural knowledge that can be used for both validation and model interpretation. Due to the curse of dimensionality (Bellman, 1961), an exhaustive search of the possible model structures is demanding. Even in the highly restrictive scenario, that the solution is a weighted linear combination of *fixed* basis functions, the parameter space has size 2^d. Extension to flexible basis functions, which is required for typical modelling, will only compound this dimension. Accordingly, greedy methods are typically used. ASMOD employs an evolutionary strategy to search the model space using a forward selection/backward elimination algorithm to select suitable refinements to a model. The MARS algorithm employs a recursive partitioning procedure to search the model space for an appropriate model. The drawback with both approaches is that they can become entrapped by local minima, due to the greedy nature of their search algorithms. A problem with deploying additive models in advanced flexible non-linear modelling methods is that they cannot provide a transparent model if the phenomenon being modelled contains high dimensional interactions. One possibility is to enforce transparency by constraining the order of possible interactions (e.g. restriction to univariate and bivariate terms only), providing a coarse, but interpretable structure, at the expense of structural integrity.

The aim of this work is to produce transparent models that generalise well, using a global approach to the modelling problem. This paper introduces a new SUpport vector Parsimonious ANOVA (SUPANOVA) technique to realise this goal. It will be shown that the technique is attractive since it can employ a wide range of loss functions (Smola, Schölkopf, & Müller, 1998), can produce interpretable models, and it is solved by breaking the problem down into simple convex optimisation problems, which can be implemented using readily available mathematical programming optimisers (Mészáros, 1998). The structure of the paper is as follows. The next section provides an overview of transparent modelling techniques: Section 4 introduces sparse additive kernel modelling, with a particular example, the SUPANOVA technique developed in Section 4, Section 5 describes the datasets which were used for evaluation, and the associated results. The paper ends with a discussion as to the applicability of interpretable modelling methods.

2. Transparent modelling methods

The interpretation of complex models has started to receive some attention within the machine learning community. Methods for enforcing or formulating additivity in various families of flexible models have been investigated by a number of researchers. Girosi (1995) shows that additive models can be formulated as regularisation networks, thereby allowing additive regularisers to be constructed. Moody and Rögnvaldsson (1996) discuss various smoothing terms for feedforward neural networks that penalise higher order derivatives with respect to the inputs; incorporation of a regularisation term pushes the model towards an additive structure (Plate, 1999). Other notable additive models include the Smoothing-Spline ANOVA (SS-ANOVA) model of Wahba (1994). This method is based on a Gaussian process model with a particular covariance function, and an additive structure.

A common approach to prevent model over-fitting is to impose a penalty constraint on the set of allowable functions which penalises the models parametric form (e.g. weight decay in neural network training) or penalises global smoothness properties (e.g. minimising curvature). A smoothness constraint essentially defines possible function behaviour in local neighbourhoods of the input space. Hence, the regulariser can be seen as imposing an ordering on the hypothesis space. However, when no prior knowledge is available about the data generating function, a large function space needs to be chosen so as to ensure that the approximation error will be small. As a consequence, imposing an order on this space is a difficult task. In learning theory there is also a need for sparse models, in which the smallest number of basis functions possible are used to approximate a function $f(x)$. In addition to a term that penalises the model parameters, an additional term to enforce sparseness of the model solution is introduced to act as a regulariser on the model structure. Both of these facets have been inspired by the well known principle of Ockham's razor.

The goal of transparency is to produce a model that not only performs predictions, but that can reveal the structure of the underlying data generating process. Additionally, since expert knowledge is typically qualitative, the resulting model can be validated. The use of a structural prior, in addition to a generic prior such as smoothness, can improve generalisation performance by constraining the hypothesis space. We now introduce some conventional techniques that encompass some form of transparency, and discuss their merits.

2.1. *Multivariate linear regression*

A Multivariate Linear Regression (MLR) model is given by

$$y = w_0 + w_1x_1 + w_2x_2 + \cdots + w_dx_d, \tag{4}$$

where $w_1, \ldots, w_d$ are unknown parameters to be estimated, w_0 is a bias term and y is the predicted output. The unknown vector of parameters, w, can be estimated in the least squares sense. The uncertainty in each of these parametric values can be estimated to indicate the first order importance of these variables in contributing to the output. However, interpretation of parameter values and their associated uncertainty provides only a crude form of input selection and transparency, since the technique typically suffers from the problem of model mismatch.

2.2. *Graphical models*

The principal role of a graphical model is to convey the conditional independence structure in a dataset via a graphical representation. The notions of independence and conditional independence are a fundamental component of probability theory. Detailed studies of conditional independence properties can be found in Dawid (1979a, 1979b) or Lauritzen (1995). The graphical model can aid in the selection of input variables as part of a data-preprocessing strategy, retaining only those variables which are conditionally dependent upon the output. This provides a powerful tool for indicating variable influence as part of a model interpretation strategy. Let X' be a d-dimensional vector of random variables. A conditional independence graph, $G = (V, E)$, describes the association structure of X' by means of a graph, specified by the vertex set V and the edge set E (Whittaker, 1990). There is a directed edge between vertices i and j if the set E contains the ordered pair (i, j); vertex i is a parent of vertex j, and vertex j is a child of vertex i. An edge can be used to indicate *relevance* or *influence* between data variables. A graphical model is then a family of probability distributions, P_G, that is a Markov distribution over G where X'_a and X'_b represent the variables for which conditional independence is being tested for, given the other variables in the dataset X'_c. A graphical Gaussian model is obtained when only continuous random variables are considered. The conditional independence constraints are equivalent to specifying zeros in the inverse variance parameter corresponding to the absence of an edge in G (Whittaker, 1990). The critical part is the test employed to ascertain dependence. For example, a deviance statistic given by,

$$\mathrm{dev}(X'_b \perp\!\!\!\perp X'_c \mid X'_a) = -N \ln\left(1 - \mathrm{corr}^2_N(X'_b, X'_c \mid X'_a)\right) \tag{5}$$

has been used. This test statistic has an asymptotic χ^2 distribution with one degree of freedom. Elements in this deviance matrix, determine the significance of dependencies in the graphical model. A hypothesis test at a 95% confidence interval of the χ^2 distribution, is used to extract the significant effects. A limitation of the graphical Gaussian model can be attributed to the deviance statistic being inaccurate, since it depends on a linear correlation term. Hence, the graphical Gaussian model will only detect linear trends between the data variables. The Gaussian process network (Friedman & Nachman, 2000) has recently been introduced as a new family of continuous variable probabilistic networks that are based on Gaussian process priors to overcome the limitations of the graphical model. The priors that are used are semi-parametric in nature allowing marginal likelihoods for structural learning to be computed directly.

Recent work on Bayesian networks (also known as belief networks) has allowed the modelling of joint probability distributions in a number of systems. A Bayesian network is a graphical model that can be used to encode expert knowledge amongst a set of variables (Heckerman, 1999). A Bayesian network consists of two components. The first is a directed acyclic graph (DAG) in which each vertex corresponds to a random variable. In a manner similar to the graphical Gaussian model, this graph describes conditional independence properties of the represented distribution. The second component is a collection of conditional probability distributions that describe the conditional probability of each variable given its parents in the graph. Together, these two components can be shown to represent a unique probability distribution (Pearl, 1988). Bayesian networks have the advantage that they can be built from prior knowledge alone, although as Heckerman (1999) observes this is only realistic for problems consisting of a few variables and where definite prior knowledge exists. In recent years there has been a growing interest in learning Bayesian networks from data; see for example the work of Buntine (1991) and Heckerman (1995). The majority of this research has focused on learning the global structure, which corresponds to the edges of the DAG, of the network. Once a Bayesian network has been constructed to be able to determine various probabilities of interest from the model requires probabilistic inference. Although conditional independence is used in a Bayesian network to simplify probabilistic inference, exact inference in an arbitrary Bayesian network for discrete variables is NP-hard (Cooper, 1990). Even approximate inference (for example by the use of Monte Carlo methods) is NP-hard (Dagum & Luby, 1993).

2.3. Automatic relevance determination

To overcome the black-box nature of the Bayesian neural network, Automatic Relevance Determination (ARD) (MacKay, 1994; Neal, 1995) has been proposed as a method of input selection, and capacity control. In an ARD model, each input variable has an associated hyperparameter that controls the magnitudes of the weights on connections to that input. If the hyperparameter associated with an input is large the weights associated with it are likely to be small, and hence the input will have little effect on the output. Interpretation of these hyperparameter values enables an inputs influence on the network to be assessed, providing a method for knowledge extraction.

One of the main criticisms of the technique is the difficulty in determining the hyperparameter values. In a true Bayesian framework, parameters whose values are not known should be integrated out by a process referred to as marginalisation. However, in the commonly employed evidence framework a *maximum a posteriori* (MAP) approach is adopted (MacKay, 1994; Bishop, 1995). This approach finds values for the hyperparameters which maximise the posterior probability and then perform the remaining calculations with the hyperparameters set to these values. This is computationally equivalent to the *type-II* maximum likelihood method of Gull (1989). The hyperparameters are set to some initial values, and are then re-estimated. Empirical results (Penny & Roberts, 1998) have shown that the final solution obtained is sensitive to the initial values of the hyperparameters, causing the network to converge to a local rather than global minimum. The alternative method of integration by sampling, e.g., Markov Chain Monte Carlo (MCMC) methods have also been considered in the machine learning community (Neal, 1995). The main criticism of MCMC methods is that they are slow and it is usually difficult to monitor convergence. Another notable disadvantage of the MCMC method is that the posterior distribution over parameters, which captures all information inferred from the data about the parameters is stored as a set of samples which can be inefficient.

2.4. ASMOD

The Additive Spline Modelling of Observational Data (ASMOD) algorithm has been employed for finding interesting trends in data (Kavli & Weyer, 1995). In the ASMOD approach a set of piecewise polynomial basis functions are defined by a series of knots. The introduction of additional knots within the basis functions enables increasingly complex functions to be approximated, whilst an increase in the order allows potentially smoother functions to be obtained. The resulting model is a multidimensional polynomial surface which can be decomposed as a series of local, low order polynomials, which can be considered as a set of local kth order Taylor series approximation to the system. The model is constructed using a forward selection, backwards elimination algorithm that updates the model iteratively by selecting the best refinement from a set of possible refinements. These refinements can include: knot insertion, knot deletion, subnetwork deletion, as well as decreasing or increasing the order of the B-spline. At each stage in the model construction process an MSE based statistical significance measure (Gunn, Brown, & Bossley, 1997) can be used to select the optimal model refinement.

2.5. CART

The use of tree-based classification and regression has been widely used in the machine learning community. Popular methods for decision-tree induction are ID3 (Quinlan, 1986), C4.5 and CART (Classification And Regression Trees) (Breiman et al., 1984). To construct an appropriate decision tree, CART first grows a descision tree by determining a succession of splits (decision boundaries) that partition the training data into disjoint subsets. Starting from the root node that contains all the training data, an exhaustive search is performed to find the split that best reduces some minimum cost-complexity principle. The net result of this continual process is a sequence of trees of various sizes; the final tree selected is the tree that performs best when an independent test set is presented. Thus, the CART algorithm can be considered to consist of two stages: tree growing and tree pruning. Transparency can be introduced into this method simply by reading off which inputs are incorporated into the final tree structure.

3. Sparse kernel methods

To address some of the difficulties associated with the preceding methods, such as model mismatch and poor generalisation we propose a new additive sparse kernel method. An additive sparse kernel model extends a standard kernel model by replacing the kernel with a weighted linear sum of kernels,

$$f(x) = \sum_{i=1}^{l} \alpha_i \sum_{j=1}^{m} c_j K_j(x^i, x), \quad c_j \geq 0, \tag{6}$$

where K_j are positive definite functions and where the positivity constraints on the kernel coefficients, c_j, ensure that the complete kernel function is positive definite. Here, the term sparse refers to sparseness in the kernel coefficients c_j rather than the usual sparseness in the multipliers, α_i; sparseness in these multipliers can still be obtained by employing an appropriate loss function. A conventional kernel model regulariser will not enforce sparsity in the kernel coefficients and hence a more complex regulariser is required. The goal in selecting a sparse representation is to minimise the number of non-zero coefficients, c_i. This can be achieved with a p-norm on the kernel coefficients. As p increases the solution becomes less sparse and the computational complexity of the resulting optimisation problem is relaxed. Ideally a value of $p = 0$, which counts the number of terms in the expansion is attractive. This case is employed in the atomic decomposition of Chen (1995), but it results in a computationally hard combinatorial optimisation problem. Alternatively choosing a value of $p = 2$ produces a straightforward optimisation problem. This case is referred to as the method of frames or ridge regression, but crucially the sparseness within the expansion is now lost. A good compromise occurs when $p = 1$ producing a sparse solution, with a practical implementation. This penalty function has successfully been used in basis-pursuit de-noising (Chen, 1995). To enforce sparsity in the kernel expansion we consider a regularised cost functional of the form

$$\Phi(\alpha, c) = L(y, K(c)\alpha) + \lambda_\alpha \|\alpha\|^2_{K(c)} + \lambda_c \|c\|_1, \quad c_i \geq 0, \lambda_\alpha, \lambda_c > 0 \tag{7}$$

where L is the loss function, and $\lambda_\alpha, \lambda_c$ are regularisation parameters controlling the smoothness and sparsity of the kernel expansion respectively.

The direct solution of this problem is non-trivial, so an iterative method is introduced, whereby we solve two separate sub-problems: $\min_\alpha \Phi$ with c fixed; $\min_c \Phi$ with α fixed. The solution for a quadratic loss, $L(y, \hat{y}) = (y - \hat{y})^T(y - \hat{y})$, is given by

$$\begin{aligned}
\Phi(\alpha, c) &= \left\| y - \sum_i c_i K_i \alpha \right\|_2^2 + \lambda_\alpha \sum_i c_i \alpha^T K_i \alpha + \lambda_c \sum_i c_i, \quad \forall_p c_p \geq 0. \\
\alpha^* &= \arg\min_\alpha \alpha^T \left(\sum_i \sum_j c_i c_j K_i K_j + \lambda_\alpha \sum_k c_k K_k \right) \alpha - \left(2y^T \sum_l c_l K_l \right) \alpha \\
c^* &= \arg\min_c \sum_i \sum_j c_i c_j (\alpha^T K_i K_j \alpha) \\
&\quad + \sum_k c_k (\lambda_\alpha \alpha^T K_k \alpha + \lambda_c - 2y^T K_k \alpha), \quad \forall_p\, c_p \geq 0,
\end{aligned}$$

where y and $\hat{y}$ are vectors of target and predicted values respectively. The solution for an ϵ-Insensitive Loss, $L(y, \hat{y}) = \sum_i \max(0, |y_i - \hat{y}_i| - \epsilon)$ by,

$$\begin{aligned}
\Phi(\alpha, c) &= \left\| y - \sum_i c_i K_i \alpha \right\|_{1,\epsilon} + \lambda_\alpha \sum_i c_i \alpha^T K_i \alpha + \lambda_c \sum_i c_i, \quad \forall_p c_p \geq 0. \\
\alpha^* &= \arg \min_{\alpha = \alpha^+ - \alpha^-} (\alpha^+ - \alpha^-)^T \left(\lambda_\alpha \sum_k c_k K_k \right) (\alpha^+ - \alpha^-) \\
&\quad - \sum_i (\alpha^+ - \alpha^-) y_i + \sum_i (\alpha^+ + \alpha^-) \epsilon, \quad \forall_i 0 \leq \alpha_i^+, \alpha_i^- \leq \frac{1}{2\lambda_\alpha}. \\
c^* &= \arg \min_{c, \zeta^+, \zeta^-} \sum_i (\zeta_i^+ + \zeta_i^-) + \sum_j c_j (\lambda_\alpha \alpha^T K_j \alpha + \lambda_c), \\
&\quad \forall_{i,j} c_j \geq 0, \zeta_i^+, \zeta_i^- \geq 0, -\zeta^- - \epsilon \leq \sum_k c_k K_k \alpha \leq \zeta^+ + \epsilon.
\end{aligned}$$

where ζ^+ and ζ^- are slack variables. An attraction of this iterative technique is that it decomposes the problem into two simple convex optimisation problems. In the quadratic loss case the solution for α^* is given by simple matrix inversion, and for c^* by a bound constrained quadratic program. In the ϵ-insensitive case the solution for α^* is given by a box constrained quadratic program, and for c^* by a bound constrained linear program with linear constraints. Consequently, they can all be solved readily using a standard quadratic programming optimiser (Mészáros, 1998). A similarity can be drawn between this approach and Bayesian methods (MacKay, 1995) that employ a two stage iterative procedure, a parameter update and 'hyperparameter' update. However, unlike most Bayesian methods, the update stages consist of convex optimisation problems.

If λ_α and λ_c are known the solution can be obtained by,

$$\begin{array}{ll} \text{Initialise :} & \alpha_0^* = \arg\min_{\alpha} \Phi(\alpha, c_0^*), \quad c_0^* = \mathbf{1} \\ \text{Iteration :} & \begin{array}{l} \text{(a) } c_{i+1}^* = \arg\min_c \Phi(\alpha_i^*, c) \\ \text{(b) } \alpha_{i+1}^* = \arg\min_\alpha \Phi(\alpha, c_{i+1}^*). \end{array} \end{array}$$

In the quadratic case the second order partial derivatives with respect to α and c are always positive ensuring that every slice is convex. This fact combined with the knowledge that the solution is finite in α and c should ensure convergence to the global minimum. A similar result should be obtainable for the ϵ-insensitive loss function. The convergence properties of this algorithm will be studied in future work. In practice the situation is more complicated since λ_α and λ_c will not be known but will need to be estimated. Intuitively, both λ_c and λ_α should initially be set large and reduced gradually; reducing λ_α too quickly will over smooth the space making the sparse selection harder; reducing λ_c too quickly will tend to produce an over-sparse model. To provide a workable solution the method used in this paper uses an initialisation step and one iteration. In the initilisation step and part (b) of the iteration, λ_c does not enter the optimisation and as such does not need to be determined; λ_α can be determined using cross-validation (8-fold is used in this paper). The difficult part is determining the parameters in part (a) of the iteration. A possible method could fix λ_α at the value used in the initialisation step and select λ_c to obtain a comparable loss to that of the initialisation step. However, the method chosen, which was based on the best empirical performance, was to set $\lambda_\alpha = 0$ and to select λ_c such that the loss was equal to that of the validation error in the initialisation step. Alternative methods for determining these parameters will be investigated in future work. In the next section a particular class of sparse additive kernel model is introduced with some attractive transparency properties.

4. SUPANOVA

The SUPANOVA technique is designed to select a parsimonious model representation by selecting a small set of terms from the complete ANOVA representation (3). The technique is an additive kernel model, (6) with a particular choice of ANOVA kernels can be expressed as and hence we can employ the sparse kernel method described in the previous section to obtain its solution. This section considers some possibilities for ANOVA kernel models. The following theory is based upon Reproducing Kernel Hilbert Spaces (RKHS) (Aronszajn, 1950; Wahba, 1990). If K is a symmetric positive definite function, which satisfies Mercer's Conditions, then the kernel represents a legitimate inner product in feature space and it may be deployed within (6). The following two theorems (Aronszajn, 1950) are required in proving that ANOVA kernels satisfy Mercer's Conditions.

Theorem 1. *If k_1 and k_2 are both positive definite functions then so is $k_1 + k_2$.*

Theorem 2. *If k_1 and k_2 are both positive definite functions then so is $k_1 \otimes k_2$.*

It follows from Theorem 2 that multidimensional kernels can be obtained by forming tensor products of univariate kernels. A multivariate ANOVA kernel is given by the tensor product of a univariate kernel plus a bias term,

$$\begin{aligned} K_{\text{ANOVA}}(u, v) &= \prod_{i=1}^{d} (1 + k(u_i, v_i)) \\ &= 1 + \sum_{i}^{d} k(u_i, v_i) + \sum_{i<j}^{d} k(u_i, v_i) k(u_j, v_j) \\ &\quad + \cdots + \prod_{i=1}^{d} k(u_i, v_i). \end{aligned} \tag{8}$$

It follows from Theorems 1 and 2 that if k is a valid kernel then so is K_{ANOVA}. Considering (8) it is evident that the tensor product produces the ANOVA terms of (1), producing a flexible model. Another consequence of Theorems 1 and 2 is that each of the additive terms in the expansion (8) is also positive definite, and hence a valid kernel in its own right. This enables partial forms of (8) to be used as valid kernels, and this is the method employed within the SUPANOVA technique to produce parsimonious kernels. The choice of univariate kernel, k, will control the form of the final model. For simplicity, we shall restrict ourselves to the case where the same kernel is used for each dimension, although different univariate kernels could be deployed.

An attractive property of the kernel-based approach is that many functions commonly employed within modelling have kernels that satisfy Mercer's Conditions. Gaussian Radial Basis Function kernels have been successfully deployed in kernel methods. However, whilst they have some attractive properties from a regularisation perspective they are poor at modelling functions with different degrees of smoothness, and require the determination of an additional smoothing parameter. Multi-Layer Perceptron (MLP) kernels, using a set of sigmoidal functions, have also been used. However, the MLP kernel is only positive definite for particular values of its two controlling parameters, making deployment more difficult. Polynomial kernels have often been used and are cheap to compute. Their disadvantage is that in an ANOVA framework a high order polynomial will be required to model arbitrary functions. Splines are an attractive choice for modelling (Wahba, 1990) due to their ability to approximate arbitrary functions. Many types of splines have kernel representations, such as odd order B-splines and infinite splines. B-splines have been used in other modelling approaches and are favourable when a rule-base interpretation is desired (Brown & Harris, 1994). However, whilst they can have some computational advantages, the regularisation operator corresponding to a B-spline kernel representation has some weaknesses (Smola, 1998). This has been observed experimentally by the production of models with a tendency to oscillation (Gunn, 1998). An infinite spline incorporates the flexibility of a spline approach without the oscillation problem associated with B-splines, and this motivates it use within an ANOVA framework. Another advantage of the infinite spline kernel is that is has no scale, and therefore no associated scale parameter to determine. This is of great advantage in the SUPANOVA technique, since the ANOVA decomposition would introduce a multitude of such parameters which would need to be determined. The first order infinite spline kernel, which passes through the origin, is defined on the interval $[0, \infty)$ by,

$$k_{\text{spline}}(u, v) = \int_0^\infty (u - \tau)_+ (v - \tau)_+ d\tau, \tag{9}$$

where $(x)_+$ is equal to the positive part of x. The solution has the form of a piece-wise cubic polynomial,

$$k_{\text{spline}}(u, v) = uv + \frac{1}{2}(u + v)\min(u, v) - \frac{1}{6}(\min(u, v))^3, \tag{10}$$

and therefore the form of the SVM solution is a piecewise cubic with knots located at a subset of the data points. Multivariate spline kernels obtained from (10) will produce a lattice of piecewise multi-cubic functions.

Using a complete ANOVA kernel (8) has drawbacks when it comes to interpretation of the model, due to the large number of terms within the expansion. To introduce enhanced transparency we employ a parsimonious ANOVA kernel. Considering the expansion of (8) an additional set of positive coefficients, c_i, are introduced,

$$\begin{aligned} K_{\text{ANOVA}}(u, v) = c_0 + \sum_i^d c_i k(u_i, u_i) + \sum_{i<j}^d c_{i,j} k(u_i, u_i) k(u_j, u_j) \\ + \cdots + c_{1,2,\ldots,d} \prod_{i=1}^d k(u_i, u_i). \end{aligned} \tag{11}$$

Consequently the resulting kernel is a weighted linear sum of kernels, and a parsimonious model solution can be obtained by using the method of the previous section.

Since the univariate ANOVA term is constrained to pass through the origin, bivariate and higher order terms will be constrained to be zero along their axes. Consequently the parsimonious model will not simply consist of the single highest order ANOVA term, but will favour low order terms in preference to high order terms. The ANOVA terms in the parsimonious model can be recovered from the final SVM expansion. For example, the univariate terms are given by,

$$f_g(x) = c_g \sum_{i=1}^l \alpha_i k\left(x_g^i, x_g\right), \tag{12}$$

and the bivariate terms are given by,

$$f_{g\otimes h}(x) = c_{g,h}\sum_{i=1}^{l}\alpha_i k\big(x_g^i, x_g\big)k\big(x_h^i, x_h\big), \tag{13}$$

where α_i are the Lagrange multipliers obtained from the complete ANOVA kernel solution. However, the computation required to solve the optimisation problem is extremely demanding due to the combinatorial nature of the problem and the curse of dimensionality (Bellman, 1961) associated with the full ANOVA expansion. To overcome this problem the ANOVA expansion can be truncated to simplify the problem, since if transparency is to be obtained the selected terms should be of low order. This technique contrasts with other parsimonious techniques, such as MARS and ASMOD, in that it aims to find a full model and sub-select the significant terms. The drawback with the MARS and ASMOD approaches is that they are local, and can suffer from entrapment in local minima within the construction process. Additionally, they may not be strictly well-posed. A further attraction of the SUPANOVA technique is that it decomposes the problem into three simple convex optimisation problems. An important issue is the form of solution produced when highly correlated inputs exist. The combination of the regularisers, (7) will produce a model that is distributed for two or more identical inputs; if a $\|c\|_0$ regulariser was used the model would not be distributed. In the case when the inputs are only highly correlated, the technique will produce a sparse model, and therefore a simple correlation test could be employed to identify the limiting case.

5. Experiments

Four modelling problems were used to assess the performance of the SUPANOVA approach. In each experiment, 90% of the data was used for training and validating the model and 10% of the data was used for estimating the generalisation performance. The SUPANOVA algorithm was executed multiple times for each problem, using both an ϵ-Insensitive and a quadratic loss function, with the whole data set "randomly" partitioned into the training and test sets. The capacity control parameter λ_α was determined using 8-fold cross validation, combined with an automatic search procedure, which locates a local minimum of the validation error.

5.1. Additive data modelling

To demonstrate the performance of the technique an artificial modelling problem proposed by Friedman (1991) was used. This is appropriate in that it concerns the modelling of an additive function, which has a sparse representation in an ANOVA framework. The model is a ten input function, with five redundant inputs, given by

$$f(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20\left(x_2 - \frac{1}{2}\right)^2 + 10x_4 + 5x_5 + \mathcal{N}(0, 1.0), \tag{14}$$

where $\mathcal{N}$ is zero mean, unit variance, additive Gaussian noise, corresponding to approximately 20% noise, and the inputs were generated independently from a uniform distribution in the interval $[0, 1]^{10}$. The experiments were performed using 200 examples, 180 for training and 20 for estimating the generalisation performance. This was repeated 50 times for each loss function producing a total of 100 models.

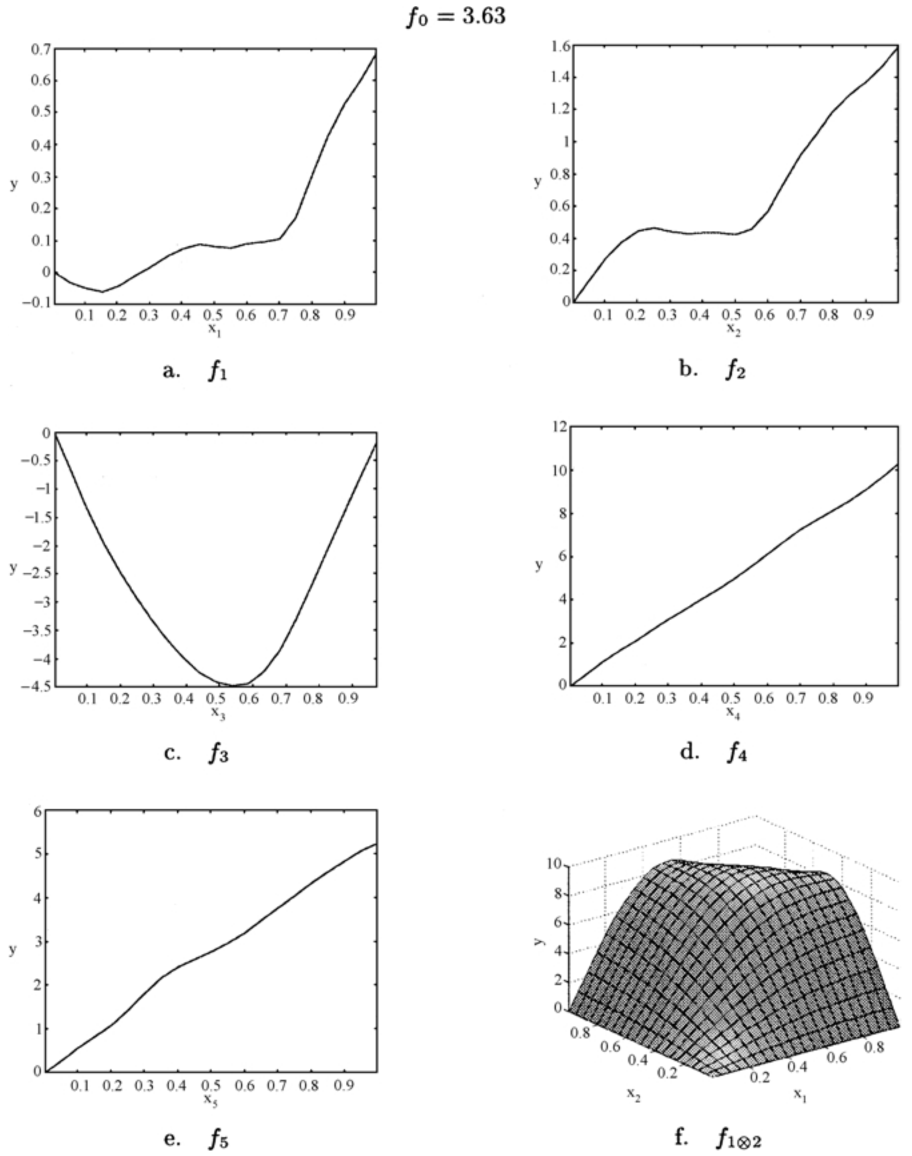
Figure 1 illustrates one of the 100 models, obtained from the SUPANOVA technique. It can be seen that it has selected 7 interaction terms (bias, five univariates, and one bivariate) from a possible 1024 terms. Each plot shows the overall effect that the ANOVA term which was selected (f_i) has on the output. The axes represent the contribution of the selected term on the output given the data. Table 1 demonstrates that the difference in the mean of the estimated generalisation error between a full ANOVA model is twice as high as the error for the parsimonious ANOVA model. These results were corroborated by the results using the ϵ-Insensitive function. Comparing the two different loss functions shows that, for this particular data-set, there is very little performance difference. The ANOVA terms selected by the 100 models are shown in Table 2. The difference column expresses the fraction of models which produced inconsistant selection in this term. The results show a high consistency, demonstrating the potential of the technique.

Table 1. SUPANOVA results for the additive data set ($\epsilon = 1.0$).

		Estimated generalisation error					
Loss function		Stage I		Stage III		Linear model	
Training	Testing	Mean	Variance	Mean	Variance	Mean	Variance
Quadratic	Quadratic	4.84	1.20	2.22	2.54	6.53	3.60
ϵ-Insensitive	ϵ-Insensitive	0.93	0.04	0.47	0.11	1.17	0.08
ϵ-Insensitive	Quadratic	4.88	1.59	2.32	2.24	6.61	3.79

Table 2. SUPANOVA terms selected for the additive data set ($\epsilon = 1.0$).

Terms	Quadratic	ϵ-Insensitive	"Difference"
bias	50	50	0.00
x_1	50	50	0.00
x_2	50	50	0.00
x_3	34	32	0.16
x_4	50	50	0.00
x_5	50	50	0.00
$x_1 \otimes x_2$	49	50	0.02
$x_3 \otimes x_8$	5	3	0.08
$x_3 \otimes x_9$	4	5	0.10
$x_3 \otimes x_{10}$	0	5	0.10
$x_4 \otimes x_5$	1	0	0.02



a. f_1 b. f_2

c. f_3 d. f_4

e. f_5 f. $f_{1\otimes 2}$

Figure 1. Visualisation of the selected ANOVA terms using a quadratic additive model (1 of 50) when applied to the additive dataset.

Table 3 shows the consistency of the ARD input selection method using MacKay's evidence framework for the additive data. The input variables have been ranked in order of the size of the hyperparameter controlling that input. It is evident that the relevant inputs are not extracted as successfully as the SUPANOVA technique. However, employing a MCMC method, Table 4, shows that the ARD method is much more consistent and comparable

Table 3. Ranked importance of input variables when using evidence framework on Friedman's additive problem.

	Input variables									
Dataset	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1	2nd	1st	3rd	–	–	5th	–	–	4th	–
2	2nd	1st	3rd	7th	–	6th	–	4th	5th	7th
3	1st	3rd	2nd	7th	–	–	–	5th	4th	6th
4	2nd	1st	3rd	5th	–	–	–	4th	–	–
5	2nd	1st	4th	5th	7th	6th	–	–	3rd	–
6	1st	2nd	3rd	4th	6th	–	–	–	–	5th
7	1st	2nd	3rd	4th	–	–	–	–	–	–
8	1st	2nd	3rd	4th	7th	–	–	5th	6th	–
9	2nd	1st	3rd	4th	6th	–	–	5th	–	–
10	1st	2nd	3rd	5th	6th	4th	7th	–	–	–

Table 4. Ranked importance of input variables when using MCMC resampling on Friedman's additive problem.

	Input variables									
Dataset	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1	1st	2nd	3rd	4th	5th	–	–	–	–	–
2	1st	2nd	3rd	4th	5th	6th	–	–	–	–
3	1st	2nd	3rd	4th	5th	–	–	6th		
4	2nd	1st	3rd	4th	5th	–	–	6th	–	–
5	1st	2nd	3rd	4th	5th	–	–	–		–
6	1st	2nd	3rd	4th	6th	–	–	–		5th
7	1st	3rd	2nd	5th	4th	–	–	–	–	–
8	3rd	1st	2nd	4th	5th	–	–		–	–
9	1st	2nd	3rd	4th	5th	–	–		6th	–
10	1st	2nd	3rd	4th	5th	–	–	–	–	–

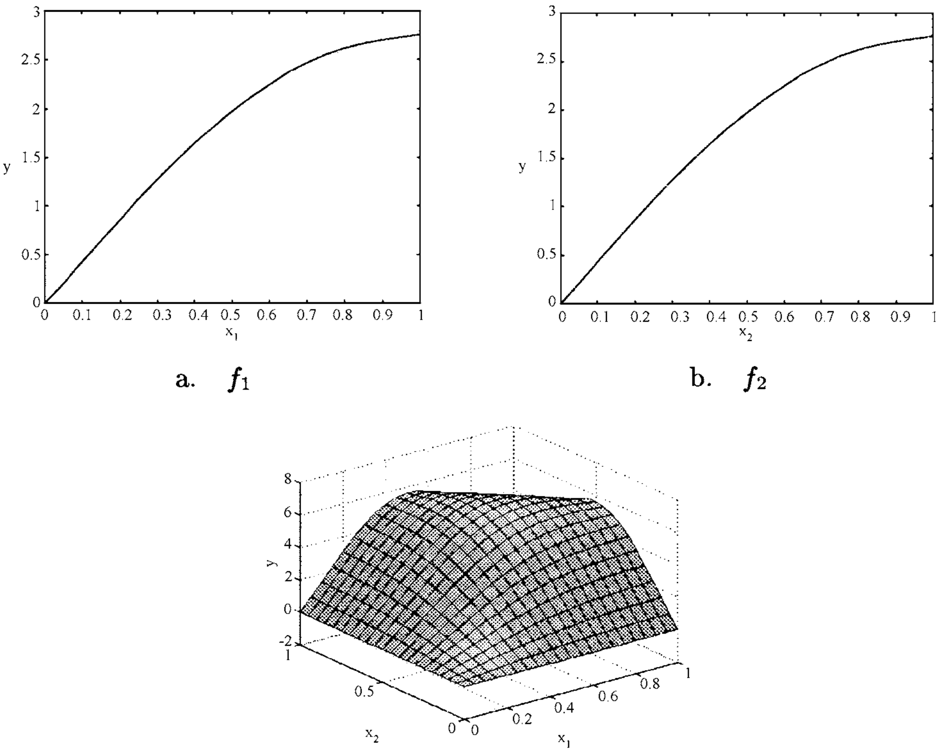
to the SUPANOVA technique. MCMC methods are advantageous since they make no assumptions concerning the form of the underlying probability distribution such as whether it can be approximated by a Gaussian distribution. Nonetheless the ARD technique as it stands is incapable of determining higher order interactions and is restricted to simple input selection.

One point of interest is brought out by the results. The spline kernel employed will produce ANOVA terms which are zero at the origin, and hence bivariate terms will be zero along both axes, which is illustrated by the $x_1 \otimes x_2$ term in figure 1. Accordingly, the additive model should not require the univariate terms x_1, x_2 to model the data generated by (14).

5.1.1. Simplified additive data modelling. To investigate the inclusion of the two univariate terms x_1, x_2, further the generating function was simplified to a two input function,

$$f(\mathbf{x}) = 10\sin(\pi x_1 x_2). \tag{15}$$

A 15 by 15 grid of points on [0, 1] × [0, 1] were used to induce a new model. In this experiment the regularisation parameter λ_{alpha}, was controlled manually, and varied over a wide range. The result for a larger value of regularisation is shown in figure 2. It is evident that the technique has modelled the function using both the univariate and bivariate terms. This is in contrast to a technique that uses a small amount of regularisation, in which the function is entirely modelled by the bivariate term. This behaviour can be explained by considering the way the regularisation term penalises the spline basis functions. The regularisation term is penalising the square of the amplitude of the basis functions. Hence, as this term becomes more significant the optimisation problem can attain a lower value by decomposing the single bivariate term into a combination of bivariate and univariate ANOVA terms. In the



a. f_1

b. f_2

c. $f_{1\otimes 2}$

Figure 2. Visualisation of ANOVA terms when deploying regularisation effects ($C = 10$).

initialisation stage where the ANOVA model space is large, it will be necessary to employ a significant amount of regularisation to control the capacity of the flexible model. Therefore, this behaviour will be common when a ridge regression type regulariser is employed. This problem could be addressed by considering alternative regularisation operators/kernels. It also explains the fact that the quadratic term was extracted less consistently than the other terms, which is evident from Table 2. However, its consequence will be to introduce ANOVA terms that are factors of a main effect and as such this is not an overriding problem, since the main effect terms typically have a low dimension. In the case when the main effect term has a high dimension, transparency has already been lost.

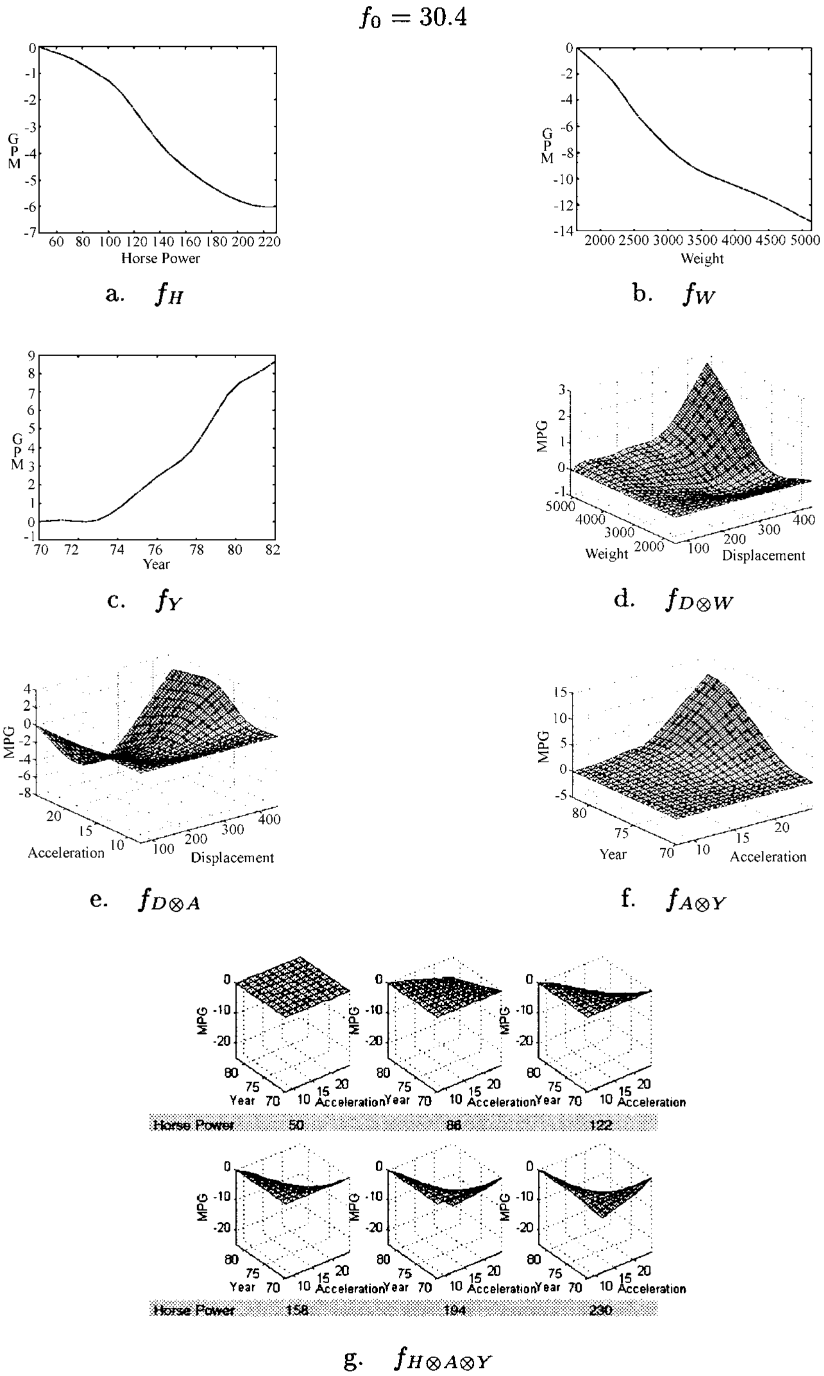
5.2. Automobile miles per gallon (AMPG) data modelling

The performance of the SUPANOVA approach to a real data-set is demonstrated by application to the problem of modelling automobile miles per gallon data (Blake & Merz, 1998). The AMPG data set contains the miles travelled, per gallon of fuel consumed, for various different cars. The input variables measure six characteristics of a car; the number of cylinders (discrete), displacement, horsepower, weight, acceleration and model year (discrete). The goal is to discover a relationship between the AMPG and the cars' characteristics. After removing a small number of entries with missing values from the original data set, the experiments were performed using 392 examples, 352 for training and validation and 40 for estimating the generalisation performance.

Figure 3 illustrates one of the 100 models, obtained from the SUPANOVA technique. It can be seen that it has selected 8 interaction terms (bias, 3 univariate, 3 bivariate and one trivariate) from a possible 64 terms. Table 5 demonstrates that the difference in the mean of the estimated generalisation error between a full ANOVA model and a parsimonious ANOVA model is negligible. However, it also demonstrates that the parsimonious kernel has a lower variance and hence suggests that it is more robust. These results were corroborated by the results using the quadratic loss function. Comparing the two different loss functions shows that, for this particular data-set, there is very little performance difference. Inspection of the ANOVA terms selected by the 100 models shows a high consistency, and confirms the robustness of the technique. The transparency of the terms is evident from figure 3, although the trivariate is harder to interpret. An example of model validation is demonstrated by the

Table 5. Mean and associated variance for the SUPANOVA results when applied to the additive data set ($\epsilon = 1.0$).

		Estimated generalisation error					
Loss function		Stage I		Stage III		Linear model	
Training	Testing	Mean	Variance	Mean	Variance	Mean	Variance
Quadratic	Quadratic	6.97	7.39	7.08	6.19	11.4	11.0
ϵ-Insensitive	ϵ-Insensitive	0.48	0.04	0.49	0.03	1.80	0.11
ϵ-Insensitive	Quadratic	7.07	6.52	7.13	6.04	11.72	10.94



a. f_H

b. f_W

c. f_Y

d. $f_{D \otimes W}$

e. $f_{D \otimes A}$

f. $f_{A \otimes Y}$

g. $f_{H \otimes A \otimes Y}$

Figure 3. Visualisation of the ANOVA terms from an ϵ-Insensitive AMPG model (1 of 50).

ability to verify the trends in the interaction terms. All the trends are consistent with prior knowledge about the problem and the univariate year term is of particular interest. It can be seen that before 1973 this term has no effect on the MPG, but after 1973 there is a sharp rise in MPG; this could be a consequence of the oil crisis.

5.3. *Boston housing data*

The Boston housing dataset originates from the work of Harrison and Rubinfield (1978) who were interested in the effect of air pollution on housing prices. The data concerns the median price in 1970 of owner-occupied houses in 506 census tracts within the Boston metropolitan area. Twelve attributes pertaining to each census tract are available for use in predicting the median price. The input variables are: *Crime rate*—per capita crime rate by town, *% Residential land*—proportion of residential land zoned for lots over 25,000 sq.ft., *% Non-retail Business*—proportion of non-retail business acres per town, *Nitric Oxides*—Nitric oxides concentration (parts per 10 million), *Mean no. of rooms*—Average number of rooms per dwelling, *% built pre 1940*—Proportion of owner-occupied units built prior to 1940, *distance to job centre*—weighted distance to five Boston employment centres, *Access to Highways*—index of accessibility to radial highways, *Property Tax*—full value property tax per $10,000, *Pupil:Teacher Ratio*—Pupil teacher ratio by town and *% Blacks*—$1000(Blks - 0.63)^2$ where Blks is the proportion of blacks by town.

As Neal (1995) observes the data is 'messy' in several regards. Some of the attributes are not actually measured on a per-tract basis, but only for larger regions. The median prices for the highest-priced tracts appear to be censored. Censoring is suggested by the fact the highest median price of exactly $50,000 is reported for sixteen of the tracts. Considering these potential problems, it appears unreasonable to expect that the distribution of the target variable, given the input variables, is Gaussian.

Work carried out by Husmeier (1999) on using an ensemble of Bayesian neural networks trained using an Expectation-Maximisation (EM) algorithm and incorporating automatic relevance determination (ARD) was able to select "relevant" inputs. The input variables rooms, distance to employment centres, access to radial highways, property tax, and the percentage of lower status in the population were selected as being relevant inputs. Husmeier observes, and this is confirmed in our approach, that the variable crime seems to be an irrelevant input. The remaining input variables show an ambiguous behaviour. Figure 4 illustrates the model obtained from the SUPANOVA technique Table 6. Fourteen interaction terms (bias, 4 univariate, 8 bivariate) were selected as being important by the SUPANOVA technqiue. Inspection of the ANOVA terms selected by the 100 models shows a high consistency, and confirms the robustness of the technique. An example of the terms chosen are shown in figure 4. The trends depicted are broadly consistent with prior knowledge about the problem.

5.4. *Materials data*

To assess the behaviour of the SUPANOVA technique in high noise situations a "real world" dataset was considered. A commercial processing-properties dataset for DC cast aluminium plate is considered, concentrating on prediction of the mechanical property 0.2% proof stress. This dataset is illustrative of the problems and challenges that arise in real world modelling; sparsely distributed data and highly correlated inputs. The raw dataset consists of ten input variables and 290 data pairs covering alloy composition and thermomechanical processing information. The ten input variables were; final gauge (FG), Cu, Fe, Mg, Mn, Si (all in weight percent), cast slab length (SL), solution treatment time (STT), percentage stretch (%st.) and reduction-ratio (RR).

5.4.1. Automatic relevance determination. The Bayesian neural network using ARD was trained in the same manner as it was for Friedman's artificial dataset. Figure 5 shows the variation of training and test set errors for increasing numbers of hidden nodes. The optimal network structure was determined to have seven hidden nodes since this corresponds to the lowest error on the test set.

Table 7 shows the mean ARD hyperparameter values (and associated standard deviation over the ten datasets) indicating the influence of each variable on the output for the optimal

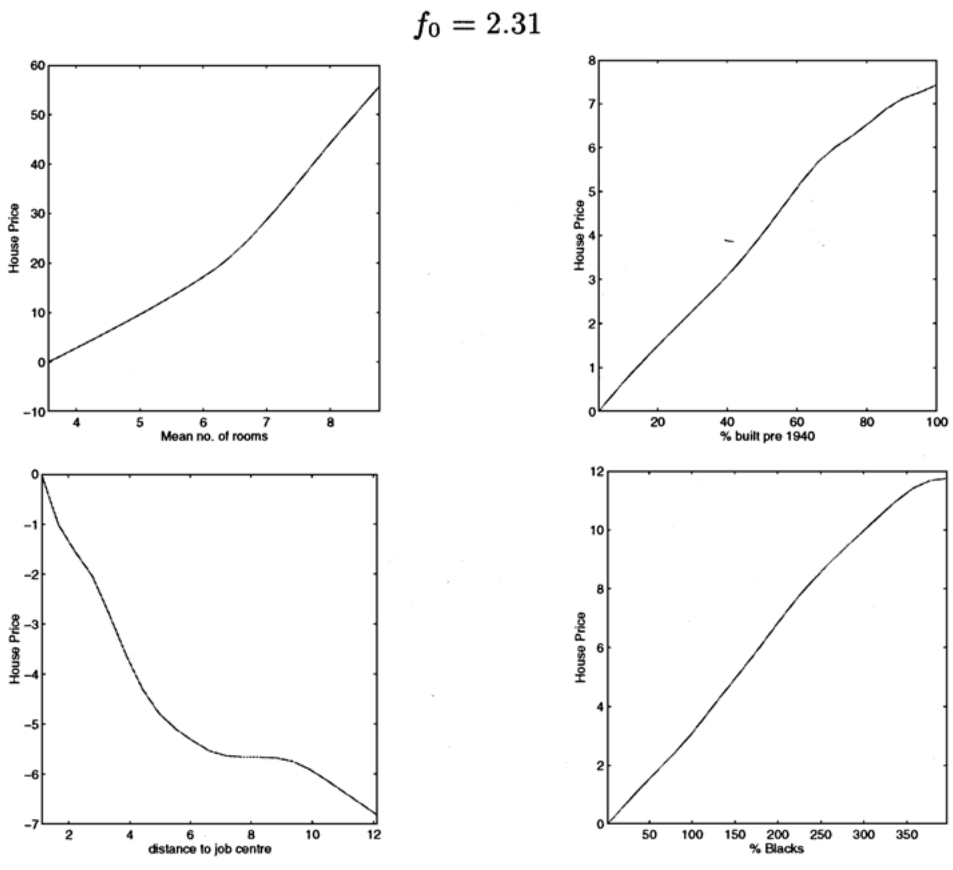


Figure 4. Visualisation of some of the selected ANOVA terms from the Boston house price data.

(*Continued on next page*.)

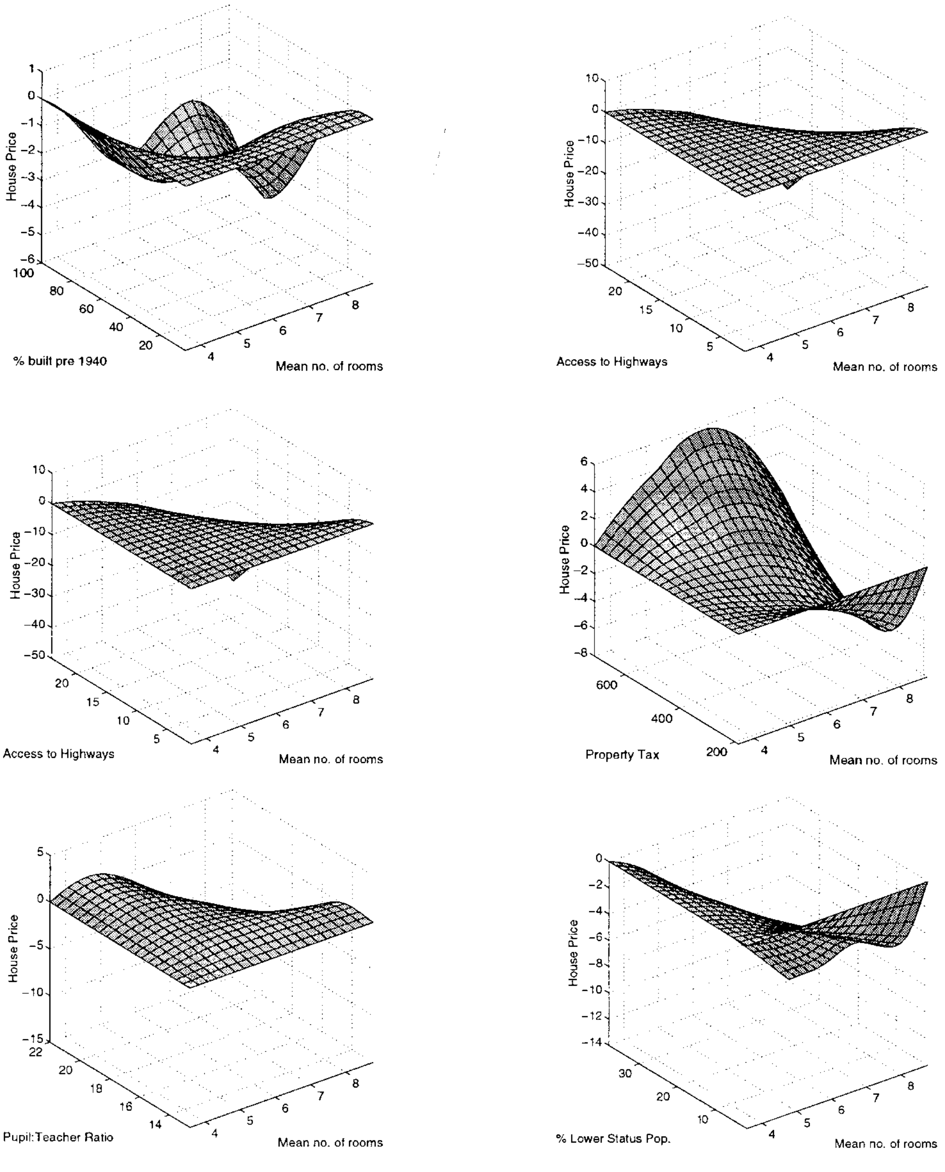


Figure 4. (*Continued*).

model structure, and Table 9 shows the ranked selection of each input variable for each of the ten models trained. From the values quoted final gauge (FG), silicon (Si), percentage stretch (%st.) and slab length (SL) exhibit the largest values. Three of these four inputs are consistent with those inputs selected by the graphical Gaussian model and the MLR. The mean MSE obtained for the training data was 57.6, whilst that for the test data was 90.5 representing a difference of 9.5 MPa between training and test performance.

Table 6. Summary of the SUPANOVA terms selected when using the AMPG data set.

Terms	Quadratic	ϵ-Insensitive	"Difference"
Bias	50	50	0.00
C	3	1	0.08
D	35	8	0.66
H	2	20	0.44
W	50	50	0.00
Y	50	50	0.00
$C \otimes D$	9	26	0.54
$C \otimes W$	0	4	0.08
$C \otimes A$	1	11	0.24
$C \otimes Y$	2	18	0.40
$D \otimes W$	35	44	0.38
$C \otimes A$	42	43	0.16
$H \otimes Y$	10	5	0.18
$W \otimes Y$	2	1	0.06
$A \otimes Y$	50	47	0.06
$C \otimes D \otimes W$	0	1	0.02
$C \otimes W \otimes A$	0	1	0.02
$C \otimes W \otimes Y$	0	1	0.02
$C \otimes A \otimes Y$	0	7	0.14
$D \otimes H \otimes W$	1	2	0.06
$H \otimes A \otimes Y$	50	49	0.02
$W \otimes A \otimes Y$	0	4	0.08
$C \otimes D \otimes W \otimes A$	0	1	0.02
$C \otimes D \otimes A \otimes Y$	4	0	0.08

($\epsilon = 2.5$), (C-No of cylinders, D-Displacement, H-Horse Power, W-Weight, A-Acceleration, Y-Year) (All remaining terms were zero).

Table 7. Mean ARD hyperparameter values for seven hidden nodes using the evidence framework.

	FG	Cu	Fe	Mg	Mn	Si	SL	STT	%st.	RR
$1/\alpha$	0.134	0.046	0.075	0.164	0.05	0.485	0.281	0.066	0.184	0.09
std	0.08	0.05	0.03	0.24	0.04	0.15	0.18	0.03	0.07	0.04

The SUPANOVA technique was applied to the ten input materials dataset, of the possible 1024 different terms in the full ANOVA expansion, only 12 terms were chosen as being significant. The full selection of terms is given in Table 8. The univariate terms selected were the bias, Mg, Si, STT, %st., the bivariate terms were FG⊗Mg, FG⊗RR, Cu⊗Si, Fe⊗Si, Mn⊗SL, Si⊗RR, and the trivariates terms FG⊗Cu⊗Si and Fe⊗Si⊗%.st. Examples of

Table 8. Summary of ANOVA terms selected when applied to the commerical materials dataset.

	Dataset									
Components	1	2	3	4	5	6	7	8	9	10
Cu	–	×	×	–	–	–	–	–	–	×
Mg	×	–	–	×	×	×	×	×	×	–
Si	×	×	×	×	×	×	×	×	×	×
STT	×	×	×	×	×	×	×	×	×	×
%st.	×	×	–	×	×	–	×	×	×	×
FG⊗Mg	×	×	×	×	×	×	×	×	×	×
FG⊗%st.	–	×	×	×	×	–	×	×	×	×
FG⊗RR	×	×	×	×	×	×	×	×	×	×
Cu⊗Si	×	×	×	×	×	×	×	×	×	×
Fe⊗Si	×	×	×	×	×	×	×	×	×	–
Mn⊗SL	×	×	–	×	×	–	×	×	×	–
Si⊗RR	×	×	–	×	×	–	×	×	×	–
FG⊗Cu⊗Si	×	×	×	×	×	×	×	×	×	×
FG⊗Mg⊗%st.	–	–	–	–	–	–	–	–	×	–
Cu⊗Mg⊗%st.	–	–	–	–	–	–	–	–	×	–
Fe⊗Si⊗%st.	×	×	–	×	×	–	×	×	×	×
Fe⊗Si⊗RR	–	–	×	–	×	×	–	–	–	×
Fe⊗SL⊗RR	–	×	–	–	–	–	–	–	×	–
Fe⊗Si⊗SL⊗RR	–	–	–	–	–	–	×	–	–	–

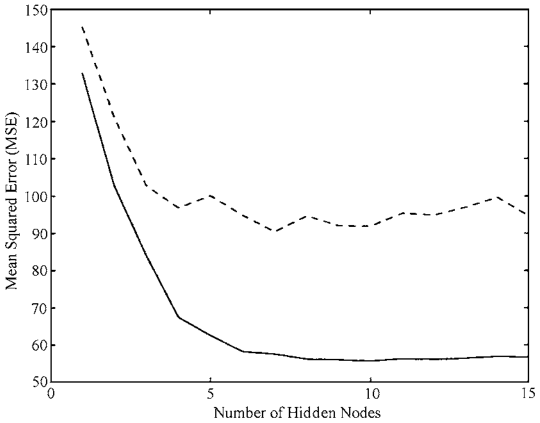


Figure 5. Variation of mean training and test MSE for a Bayesian MLP trained with varying numbers of hidden nodes.

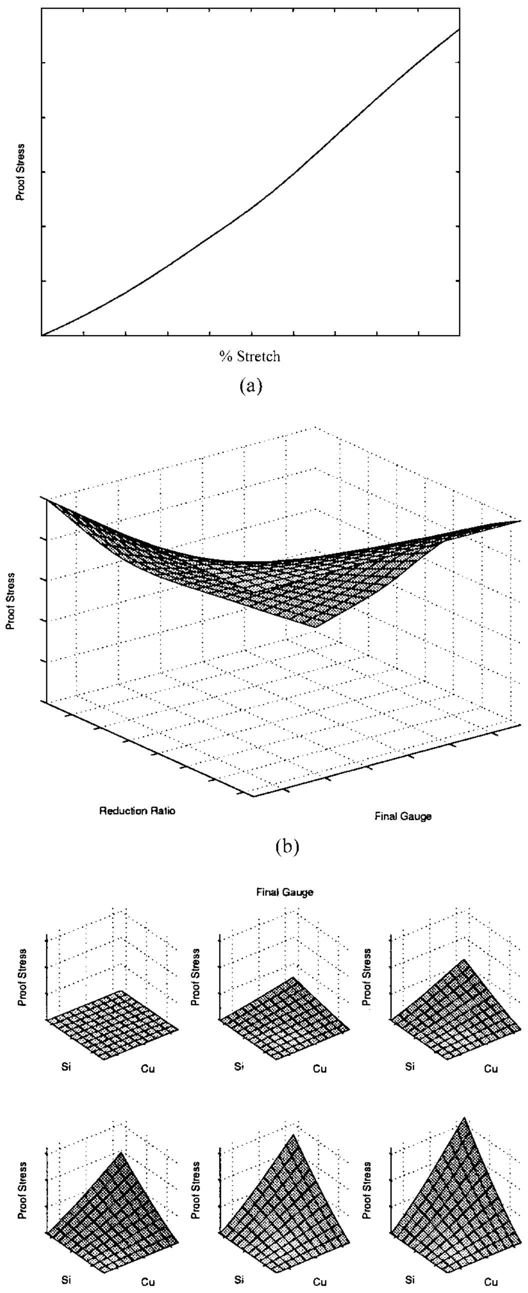


Figure 6. Examples of univariate, bivariate and trivariate interaction terms obtained from SUPANOVA applied to the commerical materials dataset. (a) Univariate interaction term, (b) Bivariate interaction term, (c) Trivariate interaction term.

Table 9. Ranked importance of the input variables.

	Input variables									
Dataset	FG	Cu	Fe	Mg	Mn	Si	SL	STT	%st.	RR
1	4th	–	–	–	–	1st	2nd	–	3rd	–
2	2nd	–	–	–	–	1st	4th	5th	3rd	6th
3	5th	–	–	3rd	6th	1st	2nd	–	4th	–
4	–	–	–	1st	4th	2nd	3rd	–	5th	6th
5	–	–	3rd	–	–	1st	–	–	2nd	4th
6	–	–	–	–	–	1st	2nd	–	3rd	4th
7	5th	–	–	4th	–	2nd	1st	–	3rd	–
8	3rd	–	–	–	–	2nd	1st	–	–	–
9	5th	–	–	4th	–	1st	2nd	–	3rd	–
10	2nd	5th	7th	–	–	1st	4th	–	3rd	6th

these are illustrated in figure 6. Table 9 shows the stability of these terms across the ten different data partitions.

These regression surfaces represent interaction terms; to see the overall effect of an input all the interaction terms associated with that input must be considered.

Figure 6 is an example of a univariate interaction term. This allows visualisation of the global contribution of percentage stretch on proof stress, as an independent effect, as it does not appear in any other terms. Interpretation of the bivariate (figure 6(b)) and trivariate (figure 6(c)) can be less straightforward. By looking at all 12 terms of the type shown in figure 6 the entire structure of the model is defined. The mean MSE for the training set was 61.4 whilst the generalisation MSE was 80.8 (giving a difference in error values of 8.9 MPa).

6. Conclusion

An approach to modelling with an emphasis on good generalisation and model interpretation has been described. Model interpretation is achieved through a interactive model representation providing input selection and enhanced visualisation. To obtain a accurate representation a large hypothesis space must be considered, and consequently a technique is warranted that has excellent capacity control. In this respect, kernel methods have been extended to allow the incorporation of a parsimonious kernels. An example if this, the SUPANOVA technique, has been described that decomposes the problem into three simple convex optimisation problems, which can be solved efficiently. With application to four datasets we have shown the the additive structure of the parsimonious SUPANOVA technique can aid in the understanding of complex relationships that can exist in data generated from physical systems.

Acknowledgments

The authors would like to thank Ian Nabney, Radford Neal and Steve Roberts for helpful discussions and comments on ARD.

References

Aronszajn, N. (1950). Theory of reproducing kernels. *Trans. Amer. Math. Soc., 686*, 337–404.

Bellman, R. (1961). *Adaptive control processes*. Princeton, NJ: Princeton University Press.

Bishop, C. (1995). *Neural networks for pattern recognition*. Oxford: Oxford University Press.

Blake, C., & Merz, C. (1998). UCI Repository of machine learning databases.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Wadsworth Inc.

Brown, M., & Harris, C. J. (1994). *Neurofuzzy adaptive modelling and control*. Hemel Hempstead: Prentice Hall.

Buntine, W. (1991). Theory refinement on bayesian networks. In B. D. D'Ambrosio, P. Smets, & P. P. Bonissone (Eds.), *Proc. Seventh Annual Conference on Uncertainty Artificial Intelligence*. San Francisco, CA, (pp. 52–60), San Mateo, CA: Morgan Kaufmann Publishers.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Journal of Data Mining and Knowledge Discovery, 2*, 121–167.

Chen, S. (1995). Basis pursuit. Ph.D. thesis, Department of Statistics, Stanford University.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines*. Cambridge: Cambridge University Press.

Dagum, P., & Luby, M. (1993). Approximating probabilistic inference in bayesian belief networks is NP-hard. *Artificial Intelligence, 60*, 141–153.

Dawid, A. (1979a). Conditional independence in statistical theory (with discussion). *Journal of the Royal Statistical Society B, 41:1*, 1–31.

Dawid, A. (1979b). Some misleading arguments concerning conditional independence. *Journal of the Royal Statistical Society B, 41:2*, 249–252.

Friedman, J. (1991). Multivariate adaptive regression splines. *The Annals of Statistics, 19*, 1–141.

Friedman, N., & Nachman, N. (2000). Gaussian process networks. In Proc. Sixteenth Conf. on Uncertainty in Artificial Intelligence (UAI), to appear.

Girosi, F. (1997). An equivalence between sparse approximation and support vector machines. A.I. Memo 1606, MIT Artificial Intelligence Laboratory.

Girosi, F., Jones, M., & Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation, 7*, 219–269.

Gull, S. (1989). Developments in maximum entropy data analysis. In J. Skilling (Ed.), *Maximum entropy and bayesian methods*. Dordrecht: Kluwer Academic Publishers.

Gunn, S. R. (1998). Support vector machines for classification and regression. Technical Report ISIS-1-98, Department of Electronics and Computer Science, University of Southampton.

Gunn, S. R., Brown, M., & Bossley, K. M. (1997). Network performance assessment for neurofuzzy data modelling. In *Intelligent Data Analysis*, (pp. 313–323).

Hadamard, J. (1923). *Lectures on the cauchy problem in linear partial differential equations*. Yale University Press.

Harrison, D., & Rubinfield, D. (1978). Hedonic housing prices and the demand for clean air. *Journal of Enviromental Economics and Management, (5)*, 81–102.

Heckerman, D. (1999). A tutorial on learning with bayesian network, *Learning in graphical models*, Cambridge, MA: MIT Press.

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning, 20*, 197–243.

Husmeier, D. (1999). *Neural networks for conditional probability estimation*. Berlin: Springer-Verlag Publishers.

Kandola, J. S., & Gunn, S. R. (2000). On the use of advanced inductive methods for knowledge extraction from complex datasets. *Submitted to Journal of Data Mining and Knowledge Discovery*.

Kavli, T., & Weyer, E. (1995). On ASMOD—an algorithm for building multivariable spline models. In G. I. K. J. Hunt & K. Warwick (Eds.), *Advances in neural networks for control systems*, Springer series on Advances in Industrial Control. Berlin: Springer Verlag, pp. 83–104.

Lauritzen, S. (1995). *Graphical models*. Oxford: Oxford University Press.

MacKay, D. (1994). Bayesian non-linear modelling for the prediction competition. *ASHRAE Transactions: Symposia*, OR-94-17-1.

MacKay, D. (1995). Ensemble learning and evidence maximization. Technical Report, Cavendish Laboratory, Dept. Physics, University of Cambridge.

Mészáros, C. (1998). The BPMPD interior point solver for convex quadratic problems. Technical Report WP 98-8, Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest.

Moody, J. E., & Rognvaldsson, T. S. (1996). Smoothing regularisers for projective basis function networks. Technical Report OGI CSE TR 96-006, Dept. Computer Science and Engineering, Oregan Graduate Institute of Science and Technology.

Neal, R. (1995). *Bayesian learning for neural networks*. Berlin: Springer-Verlag Publishers.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. San Mateo, CA: Morgan Kaufmann Publishers.

Penny, W., Roberts, S. (1998). Bayesian classification using neural networks—how useful is the evidence framework. *Neural Networks, 12*, 877–892.

Plate, T. (1999). Accuracy versus interpretability in flexible modelling: Implementing a tradeoff using Gaussian process models. *Behaviourmetrika special issue on "Interpreting Neural Network Models," 26*, 29–50.

Poggio, T., Torre, V., & Koch, C. (1985). Computational vision and regularization theory. *Nature, 317*, 314–319.

Quinlan, J. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106.

Rasmussen, C. (1996). Evaluation of gaussian processes and other methods for nonlinear regression.

Smola, A., Schölkopf, B., & Müller, K.-R. (1998). General cost functions for support vector regression. In T. Downs, M. Frean, & M. Gallagher (Eds.), *Proc. of the Ninth Australian Conf. on Neural Networks*. Brisbane, Australia (pp. 79–83). University of Queensland.

Smola, A. J. (1998). Learning with Kernels. Ph.D. thesis, Technische Universität Berlin.

Stitson, M., Gammerman, A., Vapnik, V., Vovk, V., Watkins, C., & Weston, J. (1999). Support vector regression with ANOVA decomposition kernels. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in Kernel methods—support vector learning*. Cambridge, MA (pp. 285–292). Cambridge, MA: MIT Press.

Tikhonov, A. N., & Arsenin, V. Y. (1977). *Solutions of ill-posed problems*. Washington, D.C.: W. H. Winston.

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer.

Wahba, G. (1990). *Splines models for observational data*. Philadelphia: Series in Applied Mathematics, Vol. 59, SIAM.

Wahba, G., Wang, Y., Gu, C., Klein, R., & Klein, B. (1994). *Structured machine learning for 'soft' classification with smoothing spline ANOVA and stacked tuning, testing and evaluation*. In J. Cowan, G. Tesaro & J. Alspector (Eds.), *Advances in neural information processing (NIPS)*, vol. 6, San Mateo, CA: Morgan Kauffman.

Whittaker, J. (1990). *Graphical models in applied multivariate statistics*. Chichester, UK: John Wiley and Sons.

Wyatt, J. (1995). Nervous about artificial neural networks? (commentary). *The Lancet, 346*, 1175–1177.

Received August 16, 2000
Revised January 23, 2001
Accepted January 23, 2001
Final manuscript January 26, 2001