Monograph:
Dodd, T.J., Mitchinson, B. and Harrison, R.F. (2003) Multiple-model approach to non-linear kernel-based adaptive filtering. Research Report. ACSE Research Report no. 830 . Automatic Control and Systems Engineering, University of Sheffield

Multiple-Model Approach to Non-Linear Kernel-Based Adaptive Filtering

T.J. Dodd, B. Mitchinson and R.F. Harrison
Department of Automatic Control and Systems Engineering
The University of Sheffield, Sheffield S1 3JD, UK
e-mail: {t.j.dodd, r.f.harrison}@shef.ac.uk

Research Report No. 830
January 2003

Abstract

Kernel methods now provide standard tools for the solution of function approximation and pattern classification problems. However, it is typically assumed that all data are available for training. More recently, various approaches have been proposed for extending kernel methods to sequential problems whereby the model is updated as each new data point arrives. Whilst these approaches have proven successful in estimating the basic parameters, the problem of estimating the hyperparameters, which determine the overall model behaviour, remains essentially unsolved. In this paper a novel approach to the hyperparameters is presented based on a multiple model framework. An ensemble of models with different hyperparameters is trained in parallel, the outputs of which are subsequently combined based on a predictive performance measure. This new approach is successfully demonstrated on a standard benchmark time series problem.

1 Introduction

Kernel methods, including support vector machines, Gaussian processes, regularisation networks etc, have now found widespread acceptance as a tool for function approximation and pattern recognition. Significant factors in their success include a strong theoretical basis in terms of statistical learning theory, reproducing kernel Hilbert spaces (RKHS) and Bayesian statistics and also the inherent simplicity of the resulting models. More recently various approaches have been proposed to extend the class of kernel methods to the solution of sequential (online) learning problems (Csató and Opper 2002; Schölkopf and Smola 2002; Dodd, Kadirkamanathan, and Harrison 2003; Drezet 2001). Historically, similar methods include the method of potential functions (Aizerman, Braverman, and Rozonoer 1964) and the resource allocating network and its variants (Platt 1991; Kadirkamanathan and Niranjan 1993; Li, Sundararajan, and Saratchandran 2000).

The authors have developed a strong theoretical framework for sequential learning within RKHS (Dodd, Kadirkamanathan, and Harrison 2003). The basic model update is based on a stochastic gradient descent algorithm in the RKHS. However, this results in a model for which the number of terms (kernels) grows as new data points arrive. Therefore, various methods have been proposed to limit this growth in the model, called sparsity control (Dodd, Kadirkamanathan, and Harrison 2003; Drezet 2001). This typically takes the form of only including kernels which contribute "enough" to the model and removing any kernels for which removal will not significantly degrade it. Experience shows that this provides a practical method for keeping the model size at a manageable level.

A significant limitation in the application of these methods to sequential problems is how to deal with the hyperparameters. Such parameters control aspects of the solution such as its overall complexity and must be carefully tuned to ensure good generalisation. In this paper we introduce a novel approach by adopting a multiple model framework motivated by (Dodd and Harris 1999). In our approach we train an ensemble of models each with different combinations of hyperparameters. A measure of the predictive accuracy of these models in the recent past is then used to weight the model outputs at the current time instant to form a combined estimate. Those models for which the hyperparameters correspond well to the current conditions will be weighted correspondingly greater.

In the next section we describe our approach to sequential learning in kernel methods which is based on stochastic gradient descent type methods. A significant feature of this approach is the approach to ensuring that the models do not grow "too big". The multiple model framework is then introduced in Section 3. Finally, we demonstrate the application of the method to a standard benchmark time series problem.

2 Sequential Kernel Models

We are interested in the problem of approximating some unknown input-output mapping given only observation pairs, $\{x_i, y_i\}$, where $x_i \in \mathbb{R}^L, y_i \in \mathbb{R}$. A kernel model for such an approximation can be written in the general form

$$f(x) = \sum_{i=1}^{p} \alpha_i k(v_i, x) \tag{1}$$

where p is the number of terms, x is a generic input point, α_i are a set of (unknown) parameters and the $k(v_i, \cdot)$ are a set of kernel (basis) functions centred on the points v_i in the input space. The kernel functions are assumed to be positive definite for which the $f(\cdot)$ in (1) then belongs to a reproducing kernel Hilbert space (RKHS) with reproducing kernel $k(\cdot, \cdot)$. This general form includes various classes of standard models including radial basis function neural networks, regularisation networks, support vector machines, Gaussian processes, and Volterra series.

More specifically we are interested in the case where the unknown input-output mapping corresponds to a NARMAX model and more particularly in this paper we address the case of nonlinear autoregressive models. The data are then time ordered and i represents time t_i with $x_i = [y_{i-1}, \ldots, y_{i-L+1}]^T$ a vector of lagged outputs. We call L the embedding dimension. In real applications we are then faced with the problem that the data arrive sequentially and we must therefore construct a model iteratively as each data point arrives.

A general sequential approach to learning kernel models is based on the method of stochastic gradient descent (Dodd, Kadirkamanathan, and Harrison 2003). Starting with an empty model we add a new kernel every time a new data point arrives such that $v_i = x_i$. The parameter corresponding to the new kernel is then set based on the method of stochastic gradient descent, i.e. $\alpha_i = \eta e_i$ where e_t is the prediction of the current model at the new data point, $e_i = f_{i-1}(x_i) - y_i$ and η is a learning parameter. Therefore, if the model at time $i-1$ has p terms the new model is given by

$$f_i(x) = \sum_{i=1}^{p} \alpha_i k(v_i, x) + \eta e_t k(x_t, x) = \sum_{i=1}^{p+1} \alpha_i k(v_i, x). \tag{2}$$

We assume the new model has $p+1$ terms and not i as we will subsequently introduce a sparsity control mechanism whereby the number of terms will not necessarily equal the time index. A significant problem with this (simple) approach is that the number of terms grows with the number of data points. Various approaches have been proposed to restrict this growth (Dodd, Kadirkamanathan, and Harrison 2003; Drezet 2001). These differ primarily in how terms are selected for removal. We describe a particular approach here which leads to the kernel LMS (KLMS) algorithm. We refer to removing terms as set reduction (SR). Any SR technique requires the removal of terms from the model, and thus has the potential to degrade the quality of the model. We seek to minimise this degradation.

We seek to remove those kernel functions that can be reasonably well represented as combinations of the kernel functions to be retained; we call this "sparsity control". Such techniques require the modification of the retained parameters because simply discarding a kernel will result in some (generally) undesirable degradation to the overall model. In this case, we may be able to distribute the contribution of the removed kernel amongst the retained kernel functions by adjustment of the retained parameters.

Defining the kernel (Gram) matrix, $K : K_{ij} = k(x_i, x_j)$ and assuming we have p kernel functions in the current model then we can partition the kernel matrix as follows

$$K = \left[\begin{array}{c|c} K_{p-1,p-1} & K_{p-1,1} \\ \hline K_{1,p-1} & k_{p,p} \end{array}\right] \tag{3}$$

where the last row and column correspond to the kernel to be removed. Note that we can always treat the kernel to be removed as the last row and column by a simple re-ordering of the kernel matrix.

After removal of the pth kernel function, we can choose a new set of $p-1$ multipliers $\{\beta_i\}$ such that the reduced model

$$f'(x) = \sum_{i=1}^{p-1} \beta_i k(v_i, x) \tag{4}$$

is, in some sense, the best possible approximation to the original model. Ideally, the discrepancy $\delta(x) = 0 \; \forall \; x$ where

$$\delta(.) = \sum_{i=1}^{p} \alpha_i k(v_i, \cdot) - \sum_{i=1}^{p-1} \beta_i k(v_i, \cdot). \tag{5}$$

However, in general, this will not be possible and we will have to settle for approximate agreement between the old and new models at a finite set of locations in input space. The vectors $\{v_i\}$ are an obvious choice for this set of locations. We can write the discrepancy between the old and new models at the p kernel centres as

$$\begin{aligned} \delta &= \left[\begin{array}{c} \delta_{p-1} \\ \delta_p \end{array}\right] = K\alpha - \left[\begin{array}{c} K_{p-1,p-1} \\ K_{1,p-1} \end{array}\right] \beta \\ &= \left[\begin{array}{c} \left[\begin{array}{cc} K_{p-1,p-1} & K_{p-1,1} \end{array}\right] \alpha - K_{p-1,p-1}\beta \\ \left[\begin{array}{cc} K_{1,p-1} & k_{p,p} \end{array}\right] \alpha - K_{1,p-1}\beta \end{array}\right] \end{aligned}$$

where $\delta_{p-1} \in \mathbb{R}^{p-1}, \delta_p \in \mathbb{R}$ and $\alpha = [\alpha_1 \; ... \; \alpha_p]^T$ and $\beta = [\beta_1 \; ... \; \beta_{p-1}]^T$.

A reduction technique must define not only the new multipliers $\{\beta\}$ in terms of the original model (a projection), but also a deterioration metric, e_{max}, for the reduction, which measures how damaging to the quality of the model the reduction is. One adaptive strategy is to fix a maximum permissible deterioration, e_{max}, and remove a vector whenever this maximum is not exceeded; the model grows and shrinks as necessary. Another strategy, which may be more applicable when implementing the algorithm on hardware with hard resource

limits, is to fix an upper limit for p, p_{max}, and whenever p reaches this limit, to remove the kernel function that results in the minimum deterioration. We apply the former strategy.

The KLMS technique (Drezet 2001) obtains the β which results in $\delta_{p-1} = 0$, i.e. the reduced kernel model has the same output as the original model at the vectors that are retained, and is erroneous only at the removed vector. This projection is given by

$$\begin{aligned} \beta &= K_{p-1,p-1}^{-1} \left[\begin{array}{cc} K_{p-1,p-1} & K_{p-1,1} \end{array} \right] \alpha \\ &= \left[\begin{array}{cc} I & K_{p-1,p-1}^{-1} K_{p-1,1} \end{array} \right] \alpha. \end{aligned} \tag{6}$$

For this choice, we can write the remaining element of the discrepancy as

$$\begin{aligned} \delta_p &= \left[\begin{array}{cc} K_{1,p-1} & k_{p,p} \end{array} \right] \alpha \\ &\quad -K_{1,p-1} \left[\begin{array}{cc} I & K_{p-1,p-1}^{-1} K_{p-1,1} \end{array} \right] \alpha \\ &= \left(k_{p,p} - K_{1,p-1} K_{p-1,p-1}^{-1} K_{p-1,1} \right) \alpha_p \\ &= \kappa_p \alpha_p. \end{aligned}$$

where κ_p is the deterioration metric.

Using the same projection, but the modified deterioration metric $\kappa_p \alpha_p^2$, gives the technique described in (Dodd, Kadirkamanathan, and Harrison 2003), which we call Modified KLMS (MKLMS). This technique has a strong theoretical justification in terms of finding the unique orthogonal projection of the current model onto the space of models constructed by the removal of the kernel vector.

In summary then, the sequential learning task is to update the model for each new data point using (2). Subsequently, each of the kernels is assessed for removal and if the minimum discrepency over all kernels satisfies $\kappa_p^{min} < e_{max}$ then remove that kernel corresponding to κ_p^{min} and use projection (6).

3 Multiple Model Algorithm

Thus far we have not discussed the particular form of the kernel, $k(\cdot, \cdot)$ other than it must be positive definite. Various kernels are typically used; one of the most common being the Gaussian kernel

$$k(x, x') = \exp\left(\frac{\|x - x'\|^2}{2\sigma^2} \right). \tag{7}$$

A common feature of kernels is the presence of hyperparameters, in this case σ. These control the qualitative features of the approximation, e.g. larger σ gives smoother approximations. In our sequential approach we have also introduced additional hyperparameters which in this paper we have restricted to e_{max} used in controlling sparsity. These hyperparameters must either be chosen a priori or some non-trivial approach used to estimate them. The former is problematic as it can be difficult to choose an appropriate value and the latter often leads

to computationally expensive solutions which are not amenable to online methods and do not guarantee globally optimal solutions in any case. Instead we propose a multiple model approach whereby each model in an ensemble takes different hyperparameter values. The model outputs are then combined to form predictions.

Given M models, each corresponding to a different set of hyperparameters, then the sequential learning procedure described in the previous section can be applied to estimate the parameters of each model. Given the a priori selected hyperparameters each set of model parameters will then be optimal under the assumptions given in the previous section.

We denote the jth model in the ensemble at time instant i by $f_{j,i}$ and the corresponding estimate of the $(i+p)$th sample using this model by $\hat{y}^j_{i+p}$. Such an estimate will use the regressor of lagged outputs $\hat{y}^j_{i+p-1}, \ldots, \hat{y}^j_{i+1}, y^j_i, \ldots, y^j_{i+p-L}$ where the terms, $\hat{y}$, are ,themselves, estimates resulting in an iterated prediction.

We then write the prediction at time $i+p$ of the multiple model given observations up to time, i, as

$$\hat{y}_{i+p|i} = \sum_{j=1}^{M} w_{j,i} \hat{y}^j_{i+p} \tag{8}$$

where the $\{w_{j,i}\}$ is a set of M, to be determined, weighting parameters, one for each model.

The weighting parameters are calculated using a moving average squared error (MASE) defined, for each model, by

$$\eta_{j,i} = \sum_{l=0}^{R} \left(\hat{y}^j_{i-l+1|i-l} - y_{i-l+1} \right)^2 \quad \forall \ \ j \in [1, M] \tag{9}$$

where R is an a priori chosen prediction horizon.

The MASE for model j, $\eta_{j,i}$, is a historical measure of the one step ahead prediction performance of the model over the previous R samples. A low value for $\eta_{j,i}$ indicates that model i made accurate one step ahead predictions at the last R time instants. The value of R dictates how localised this measure is. For $R = 0$ then the MASE only assesses the model accuracy on the current sample. This is unlikely to give satisfactory results as there is no averaging of the model performance over a number of samples. We should therefore set $R > 0$, however what particular value to choose is still an open question. This, together with the number of models, M, remain the only parameters which cannot be incorporated into the multiple model framework. We therefore refer to R and M as metaparameters. Some discussion on this point is given for the example in Section 4.2.

The reciprocal MASE is then given by

$$\gamma_{j,i} = \frac{1}{\eta_{j,i}} \quad \forall \ \ j \in [1, M] \tag{10}$$

and, finally, the weighting for the jth model is calculated using

$$w_{j,i} = \frac{\gamma_{j,i}}{\sum_j \gamma_{j,i}} \quad \forall \quad j \in [1, M]. \tag{11}$$

By defining the model weightings as such ensures that, as we would expect, the constraint $\sum_{j=1}^{M} w_{j,i} = 1$ for all i is satisfied.

This algorithm then provides a method for combining, sequentially, the outputs of the individual models in the ensemble. Those models which have performed best over the receding horizon, R, will be given more weighting accordingly. This ensures that the output of the combination is biased to those models which are (locally in time) best. Given the measure, MASE, used in weighting the models then, given the recent prediction horizon used, the combination should perform at least as well as the best single model in most cases. However, depending on the value of R this may not always be the case. Further investigation is ongoing into this point.

4 Example: Laser Data

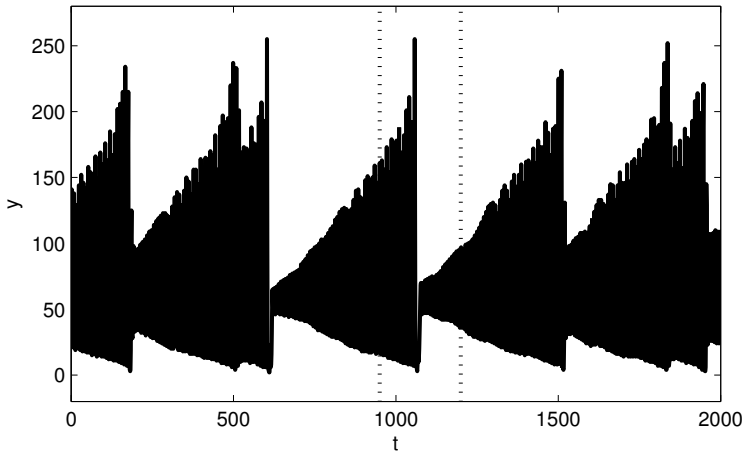
4.1 Data Description

As an example of real chaotic (nonlinear) data we examine a standard data set which formed part of the Santa Fe time series competition as data set A (Weigend and Gershenfeld 1994) and which has previously been used to assess kernel methods for system identification in the batch case (Dodd and Harris 2002). The data are recordings of the output intensity as recorded from a Far-Infrared-Laser in a chaotic state. The data is very clean with a signal-to-noise ratio of approximately 300 and corresponds to a stationary, low-dimensional chaotic behaviour (Hübner, Weiss, Abraham, and Tang 1993). Whilst the data set is very predictable on the shortest time scales (relatively simple oscillations) the global events are harder to predict. In fact, (Paluš 1993) describes the chaoticity of the laser data as somewhere between the strongly chaotic Lorenz systems and the weakly chaotic Rössler system.

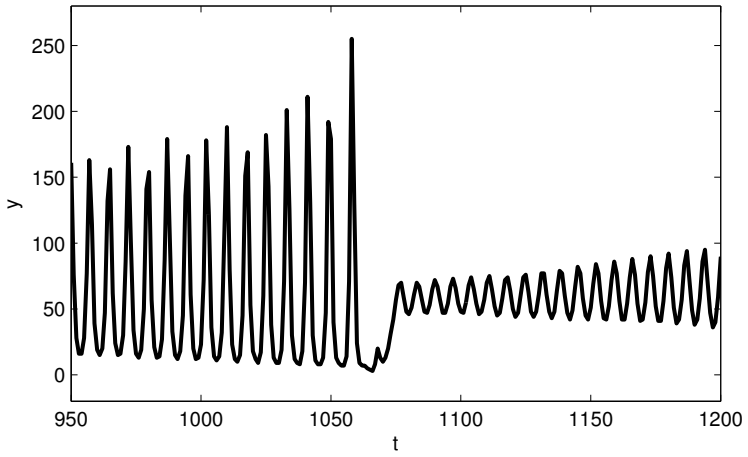
In (Paluš 1993) the correlation dimension of the laser data was estimated to be about 2.05, and if we take the next highest integer then this corresponds to an embedding dimension of three. This is in accordance with (Weigend 1994) who estimated the dimension as three using an entropy (information) based criterion. The dimension of the underlying manifold has been estimated to be close to two (Paluš 1993). In contrast (Casdagli and Weigend 1993) found that an embedding dimension of approximately eight gives the most accurate forecasts. More generally the best results in the competition used embedding dimensions of 25, 32, 50 and 200 (Casdagli and Weigend 1993). For the purposes of our experiments we chose embedding dimensions of 6, 12 and 18.

The original dataset supplied for training in the competition consisted of 1000 points. In addition to this an additional 9000 points was then made available to provide unseen data for testing purposes. We used the composition of

the original data set together with the first 1,000 points of the continuation, Figure 1. This ensured the data contained a variety of dynamic behaviours for training. For the purposes of assessing the prediction performance we have high-



(a)



(b)

Figure 1: Laser time series data set, (a) all 2000 samples with a mode change delineated and highlighted in (b) corresponding to samples 951-1200.

lighted some distinct sections of the data. We determine that a mode change commences approximately at the last strong peak before a collapse and subsequently lasts for 100 samples. Within our data set we have highlighted four major mode changes, at [181, 280], [601, 700], [1061, 1160], [1511, 1610]. In addition there exist two minor mode changes at around 520 and 1850 samples, which are ignored. A further mode change at around 1950 is ignored, since we measure the MSE only over the sample range 51-1950. We also define regions

of the data where the series is particularly well-behaved (read, 'particularly periodic'); these are given by [301, 500], [701, 900], [1151, 1351], [1601, 1800] and are called 'stable regions'. It is expected that prediction accuracy will be good in these stable regions and less good during mode changes.

4.2 Sequential Prediction

The results described in this section were arrived at using the KLMS algorithm for model reduction with the following hyperparameter values

$$\sigma \in \{10, 20, 40, 70, 100\}, \quad e_{\max} \in \{0.1, 0.5, 0.9\}$$

resulting in a total of 15 models.

Various results were generated for different prediction horizons, R, as used for the model weighting. It was found that for $R < 7$ then certain single models outperformed the multiple model under some conditions. Eventually a value of $R = 15$ was selected which provided a reasonable set of illustrative results.

The prediction performance was measured as the MSE over an iterated prediction horizon of 20 steps ahead ($p = 20$), normalised by the MSE of the zero model. Four different sets of results were obtained over (i) mode change regions, (ii) the non-mode-change regions, (iii) the stable regions, and (iv) the entire data set. In each case, samples $1 - 50$ and $1951 - 2000$ were excluded to avoid difficulties with end effects.

Experiments were performed over a variety of data pre-processing approaches and embedding dimensions. In addition to the raw data, pre-processing to give zero mean and zero median was applied. In principle these various models could be incorporated into the multiple model paradigm. In practise it was found that the raw data performed best and the case of an embedding dimension of six demonstrated much clearer performance of the combination model over the individual models. The results presented therefore correspond to this case only.

Figure 2 shows the normalised (with respect to the zero model) MSE performance of the multiple model and single models together with persistence (prediction is taken as the current value) and zero models for comparison. The results show the iterated prediction NMSE for prediction horizons of 1-20 steps ahead. The results for the 15 single models are shown as grey regions between the best and worst performing models at each prediction horizon.

5 Concluding Remarks

The principal conclusions we can draw from these results are:

- except during mode changes, every one of the single models outperforms the zero model;
- the combination always outperforms every one of the single models;
- the combination always outperforms the persistence model; and

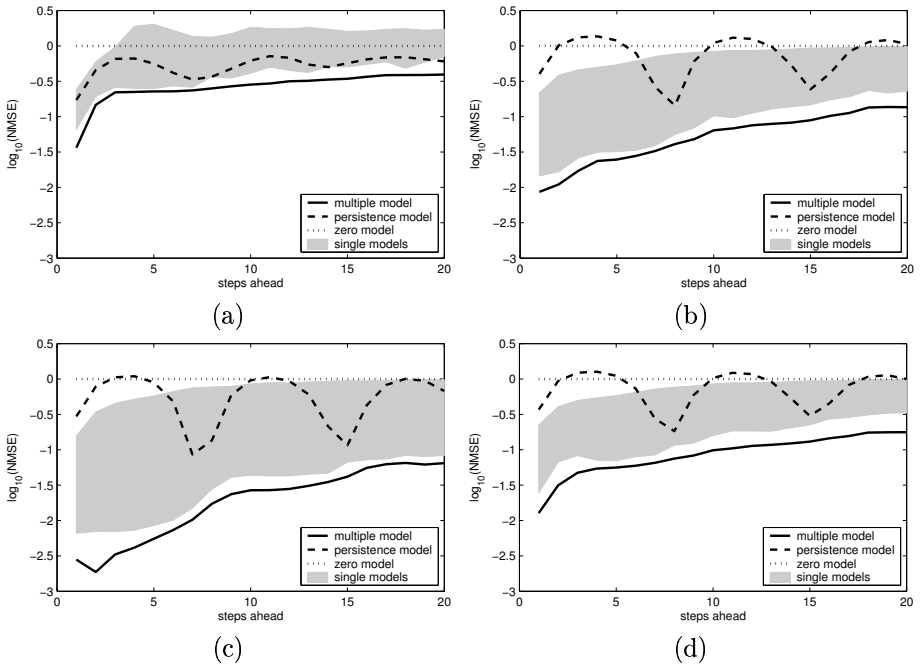


Figure 2: Multi-step-ahead NMSE measured over (a) mode changes, (b) outside mode changes, (c) stable regions, and (d) whole data set.

- the performance is least good for mode changes and improves substantially in the non-mode-change regions, being best in the "stable regions".

These conclusions are as expected and demonstrate that the multiple model framework does indeed provide a practical solution to the problem of hyperparameter estimation. Although the approach to combining the models has not been optimised we see that the combination always performs better than the best single model. This is to be expected given that, at worst, we would expect the combination to consist of only the best single model. However, in practise we see that this model can always be improved in our example using an appropriate weighted combination with other models.

We therefore believe that we have presented a useful approach to removing the need for a priori selection or estimation the hyperparameters. More generally this approach needs to be extended to take account of all the possible hyperparameters. This presents a significant computational challenge which we are seeking to address. In addition our approach obviously needs to be tested on a variety of other example problems. Finally, the asymptotic performance of the ensemble as the number of models increases, and the comparative performance to the best single model are also being investigated. In the current approach we have assumed the models are conditionally independent. By taking account of the correlations between the model outputs within, for example, a Kalman

filter or recursive least squares based combination approach, we believe that the results can be further improved.

Acknowledgement

The authors would like to thank the UK EPSRC for its financial support under Grant No. GR/R15726/01.

References

Aizerman, M., E. Braverman, and L. Rozonoer (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control (translation of Automatika i Telemekhanika) 25*(6), 821–837.

Casdagli, M. and A. Weigend (1993). Exploring the continuum between deterministic and stochastic modeling. In A. Weigend and N. Gershenfeld (Eds.), *Time Series Prediction: Forecasting the Future and Understanding the Past*, Reading, MA, pp. 347–369. Addison-Wesley.

Csató, L. and M. Opper (2002). Sparse on-line Gaussian processes. *Neural Computation 14*, 641–668.

Dodd, T. and C. Harris (1999, July). Committees of Gaussian kernel based models. In *Proceedings of the 2nd International Conference on Information Fusion*, California, USA, pp. 281–288.

Dodd, T. and C. Harris (2002). Identification of nonlinear time series via kernels. *International Journal of Systems Science 33*(9), 737–750.

Dodd, T., V. Kadirkamanathan, and R. Harrison (2003). Function estimation in Hilbert space using sequential projections. In *Proceedings of the IFAC International Conference on Intelligent Control Systems and Signal Processing.*

Drezet, P. (2001). *Kernel Methods and their Application to Systems Identification and Signal Processing.* Ph. D. thesis, The University of Sheffield.

Hübner, U., C. Weiss, N. Abraham, and D. Tang (1993). Lorenz-like chaos in NH_3-FIR lasers. In A. Weigend and N. Gershenfeld (Eds.), *Time Series Prediction: Forecasting the Future and Understanding the Past*, Reading, MA, pp. 73–104. Addison-Wesley.

Kadirkamanathan, V. and M. Niranjan (1993). A function estimation approach to sequential learning with neural networks. *Neural Computation 5*(6), 954–975.

Li, Y., N. Sundararajan, and P. Saratchandran (2000, July). Analysis of minimal radial basis function network algorithm for real-time identification of nonlinear dynamic systems. *IEE Proceedings on Control Theory and Applications 147*(4), 476–484.

Paluš, M. (1993). Identifying and quantifying chaos by using information-theoretic functionals. In A. Weigend and N. Gershenfeld (Eds.), *Time Series Prediction: Forecasting the Future and Understanding the Past*, Reading, MA, pp. 387–413. Addison-Wesley.

Platt, J. (1991). A resource-allocating network for function interpolation. *Neural Computation 3*, 213–225.

Schölkopf, B. and A. Smola (2002). *Learning with Kernels*. The MIT Press.

Weigend, A. (1994). Time series analysis and prediction. Technical report, Department of Computer Science and Institute of Cognitive Science, University of Colorado.

Weigend, A. and N. Gershenfeld (Eds.) (1994). *Time Series Prediction: Forecasting the Future and Understanding the Past*, Volume XV of *Santa Fe Institute Studies in the Sciences of Complexity*. Addison-Wesley.