Sparse Model Construction using Coordinate Descent Optimization

Xia Hong
School of Systems Eng.
U. of Reading
UK

Yi Guo
CSIRO
North Ryde, NSW 1670
Australia

Sheng Chen
ECS
U. of Southampton
UK

Junbin Gao
Charles Sturt University
Bathurst, NSW 2795
Australia

***Abstract*—We propose a new sparse model construction method aimed at maximizing a model's generalisation capability for a large class of linear-in-the-parameters models. The coordinate descent optimization algorithm is employed with a modified l_1- penalized least squares cost function in order to estimate a single parameter and its regularization parameter simultaneously based on the leave one out mean square error (LOOMSE). Our original contribution is to derive a closed form of optimal LOOMSE regularization parameter for a single term model, for which we show that the LOOMSE can be analytically computed without actually splitting the data set leading to a very simple parameter estimation method. We then integrate the new results within the coordinate descent optimization algorithm to update model parameters one at the time for linear-in-the-parameters models. Consequently a fully automated procedure is achieved without resort to any other validation data set for iterative model evaluation. Illustrative examples are included to demonstrate the effectiveness of the new approaches.**

***Index Terms*—lasso, linear-in-the-parameters model, regularization, leave one out errors, cross validation.**

I. Introduction

In data based modeling for the construction of mathematical models, one of the main aims should be good generalisation of the models, i.e. the capability to approximate system output for unseen data. A large class of nonlinear models including some types of neural networks can be classified as linear models which include statistically linear or linear-in-the-parameters models [1], [2]. These models have provable learning and convergence conditions and are well suited to be used for adaptive learning. They are amenable to parallel implementations, and have clear applications in many engineering applications [3]–[5]. Two important aspects in system identification are choosing parsimonious model structure and deriving robust model parameter estimates for a smooth prediction surface.

Fundamental to the evaluation of model generalisation capability is the concept of cross-validation (CV) [6], which can be used either in parameter estimation (e.g. tuning regularisation parameter [7], [8], forming new parameter estimates [9]), or to derive model selection criteria based on information theoretic principles [10], which regularises model structure in order to produce parsimonious models, since a parsimonious model is favored by these criteria. Cross validation as required in most algorithms for model generalization evaluation contributes significantly to overall computational overheads. Luckily for the linear-in-the-parameters models, the leave one out (LOO) mean square error (LOOMSE) can be calculated without actually splitting the training data set and estimating the associated models, by making use of the Sherman-Morrison-Woodbury theorem [11]. For linear models, the forward orthogonal least squares (OLS) algorithm efficiently constructs parsimonious models [12], [13], and has been a popular tool in associative neural networks such as fuzzy/neurofuzzy systems [14], [15], wavelet neural networks [16], [17]. It is shown the computation cost of LOOMSE is further reduced via recurcive computation, which is used as the model term selective criterion to in the forward orthogonal least squares (OLS) algorithm [18].

Regularization methods are developed to carry out parameter estimation and model structure selection simultaneously [19], [20]. It has been shown [21], [22] that the parameter regularization is equivalent to a maximized *a posterior* probability (MAP) estimate of parameters from Bayesian viewpoint by adopting a Gaussian prior for parameters. The regularization [7], [8] uses a penalty function on l^2 norms of the parameters. A regularization parameter is equivalent to the ratio of the related hyperparameter to the noise parameter, lending to an iterative evidence procedure for solving the optimal regularization parameters [19], [22].

Alternatively the model sparsity can be achieved by minimizing the l^1 norm of the parameters. The l^1 norm minimization is fundamental to the basis pursuit or least absolute shrinkage and selection operator (lasso) [23], [24]. Using the l_1-penalized cost function for a large class of linear-in-the-parameters models leads to a standard quadratic programming optimization problem. The advantage of lasso is that it can achieve much sparser models by forcing more parameters to zero, than models derived from the minimization of the l^p norm, as most l^p norms will produces small, but nonzero, values. The Bayesian interpretation for lasso is simply by adopting an Laplacian prior for parameters. By exploiting piecewise linearity of the problem, the least angle regression (LAR) procedure [25] is developed for solving the problem efficiently, facilitated by a *single* regularisation parameter setting. If the model performance is measured by the model predictive performance via a form of cross validation, the optimal regularization can be easily determined using line search.

The coordinate descent is a popular optimization tech-

nique by updating one variable at a time by minimizing a single-variable sub-problem. It is particularly appealing if the subproblem is simple. The coordinate descent algorithm has been recently successfully applied for penalized least squares problems [26], [27], where a fixed regulariser is decreasing along a path, along which the current solutions are used to as staring points to yield efficient solutions.

In this paper we propose a new coordinate descent optimization algorithm, within which the l^1 regularization is applied to estimate model parameters one at a time. We show that for a significant model term there exists a closed form of optimal LOOMSE regularization parameter for a single term model, which can be analytically computed without actually splitting the data set. Consequently we proposed a very simple method of simultaneously estimating the regularizer and parameter, forming the basis of our proposed sparse model construction algorithm for linear-in-the-parameters models. The method is very simple to implement without resort to any other validation data set for iterative model evaluation.

The paper is organized as follows. Section II introduces the general linear-in-the-parameters problem and the proposed cost function. Section III formulates a single term model, derives LOOMSE for the single term model and presents the proposed regulariser estimation formula for minimizing LOOMSE. Section IV presents the proposed coordinate algorithm which cyclically update a one term model with other parameters fixed in turn. Simulated examples are utilized to demonstrate the efficacy of the proposed algorithm in Section V and some conclusions are given in Section VI.

II. Problem formulation

Consider the general nonlinear system represented by the nonlinear model:

$$y(k) = f(\mathbf{x}(k)) + e(k), \tag{1}$$

where $\mathbf{x}(k) \in \Re^m$ denotes the system input vector and $y(k)$ is the system output variable, respectively. $e(k)$ is the system white noise and $f(\bullet)$ is the unknown system mapping. The system model (1) is to be identified from an observation data set $D_N = \{\mathbf{x}(k), y(k)\}_{k=1}^N$ using some suitable functional which can approximate $f(\bullet)$ with arbitrary accuracy. One class of such functionals is the kernel regression model of the form:

$$y(k) = \hat{y}(k) + e(k) = \sum_{i=1}^{n_M} \theta_i \phi_i(\mathbf{x}(k)) + e(k), \tag{2}$$

where $\hat{y}(k)$ denotes the model output, θ_i are the model weights, $\phi_i(\mathbf{x}(k))$ are the regressors, and n_M is the total number of candidate regressors or model terms.

By letting $\boldsymbol{\phi}_i = [\phi_i(\mathbf{x}(1)) \cdots \phi_i(\mathbf{x}(N))]^T$, for $1 \le i \le n_M$, and defining

$$\mathbf{y} = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}, \quad \boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_{n_M}],$$
$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_{n_M} \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} e(1) \\ \vdots \\ e(N) \end{bmatrix}, \tag{3}$$

the regression model (2) can be written in the matrix form

$$\mathbf{y} = \boldsymbol{\Phi}\boldsymbol{\theta} + \mathbf{e}. \tag{4}$$

Let $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_{n_M}]^T$, with $\lambda_j > \delta$, $\forall j$. δ is a predetermined small positive number. Define the cost function

$$L(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2 + \sum_{j=1}^{n_M} \lambda_j |\theta|_j \tag{5}$$

where $\| \bullet \|$ denotes Euclidean norm. We highlight that we do need to have an individual regularization parameter associated with each model term, and this makes our objective function to be different from original lasso setting. The second term in (5) helps to achieve a sparse model, since for fixed $\boldsymbol{\lambda}$, we can find

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \{L(\boldsymbol{\lambda}, \boldsymbol{\theta})\} \tag{6}$$

via a standard quadratic programming algorithm. The resultant solutions have many parameters exactly as zeros. In order to obtain a model with good generalization, $\boldsymbol{\lambda}$ needs to be optimized with respect to the models predictive performance over an unseen data set. If all elements in $\boldsymbol{\lambda}$ are constrained to have the same value, then a grid search can locate the optimal $\boldsymbol{\lambda}$ efficiently, e.g. using ten fold cross validation.

III. Optimizing regularization parameter for one term model using LOOMSE

A. One term model and LOOMSE

We note that there is no analytical solution to (5) because of the correlated terms. If however there is only one term in the model, i.e. $n_M = 1$, $L(\boldsymbol{\lambda}, \boldsymbol{\theta})$ becomes

$$L(\lambda_1, \theta_1) = \|\mathbf{y} - \theta_1 \boldsymbol{\phi}_1\|^2 + \lambda_1 |\theta|_1 \tag{7}$$

Let $\theta_1^{(LS)} = \frac{\boldsymbol{\phi}_1^T \mathbf{y}}{\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1}$ denote the least square estimate. With λ_1 being fixed, by setting the subderivatives $\frac{\partial L(\lambda_1, \theta_1)}{\partial \theta_1} = 0$, we have

$$\boldsymbol{\phi}_1^T \mathbf{y} - \frac{\lambda_1}{2} \text{sign}(\theta_1) = \boldsymbol{\phi}_1^T \boldsymbol{\phi}_1 \theta_1 \tag{8}$$

where

$$\text{sign}(s) \begin{cases} = \quad 1 & \text{if} \quad s > 0 \\ = \quad -1 & \text{if} \quad s < 0 \\ \in [-1,\ 1] & \text{if} \quad s = 0 \end{cases} \tag{9}$$

yielding to the following simple solution:

$$\theta_1^{(lasso)} = \big(|\theta_1^{(LS)}| - \frac{\lambda_1}{2\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1}\big)_+ \text{sign}(\theta_1^{(LS)}) \tag{10}$$

where

$$z_+ = \begin{cases} z & \text{if} \quad z > 0 \\ 0 & \text{if} \quad z \le 0 \end{cases} \tag{11}$$

Clearly if $\delta \ge 2|\boldsymbol{\phi}_1^T \mathbf{y}|$, $\theta_1^{(lasso)} = 0$. This is also the case for any $\lambda_1 > 2|\boldsymbol{\phi}_1^T \mathbf{y}|$. If $\delta < 2|\boldsymbol{\phi}_1^T \mathbf{y}|$, and as we decrease λ_1 from $2|\boldsymbol{\phi}_1^T \mathbf{y}|$ to δ, $\theta_1^{(lasso)}$ increases its magnitude monotonically from zero to

$$\theta_1^{(B)} = \big(|\theta_1^{(LS)}| - \frac{\delta}{2\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1}\big)_+ \text{sign}(\theta_1^{(LS)}). \tag{12}$$

Whether a given $\boldsymbol{\phi}_1$ may be reliably excluded from the model can be indicated by the magnitude of $\boldsymbol{\phi}_1^T \mathbf{y}$, the cross correlation between the model term and system output. For any data set the cross correlation of an insignificant term may be small, but cannot be exactly zero. However an insignificant term may be assessed via its sample cross correlation to the model output for randomly re-sampled data sets, which will have different signs over different data sets. However if $|\boldsymbol{\phi}_1^T \mathbf{y}|$ is sufficiently large, then the sign change due to sampling is unlikely, then it indicates that the term should be included in the model.

Consider the general model selection problem from a set of models produced using different setting of regularization parameters of λ_1. In this contribution we introduce analytically choosing the regularization parameter for a significant model term with $2|\boldsymbol{\phi}_1^T \mathbf{y}| > \delta_1$, based on the concept of leave out out cross validation but without actually splitting the data set, where δ_1 is an empirically predetermined positive value. Denote a predictor as $\hat{y}(k, \lambda_1)$ if it is identified using all N data points. The idea of LOO is that, each data point in the estimation data set D_N is sequentially set aside in turn, a model is estimated using the remaining $(N-1)$ data, and the prediction error is calculated based on the data point that was removed. That is, for $k = 1, \cdots, N$, the model is estimated by removing the k^{th} data point from the estimation set. The output of the model based on $(N-1)$ data points (with the k^{th} data point removed) is denoted by $\hat{y}^{(-k)}(k, \lambda_1)$, and the LOO prediction error is calculated as

$$e^{(-k)}(k, \lambda_1) = y(k) - \hat{y}^{(-k)}(k, \lambda_1) \tag{13}$$

Finally the leave one out mean square error (LOOMSE) is obtained by computing the average of all these prediction errors as $J(\lambda_1) = E\left[[e^{(-k)}(k, \lambda_1)]^2\right]$. For models obtained with different setting of regularization parameters of λ_1, the one associated with the minimal LOOMSE is chosen, i.e.

$$\lambda_1^{opt} = \arg\{\min_{\lambda_1}\{J(\lambda_1) = \frac{1}{N}\sum_{k=1}^{N}[e^{(-k)}(k, \lambda_1)]^2\}\} \tag{14}$$

and the resultant model is selected.

In the following we show that LOOMSE based on (2) can be evaluated efficiently without actually sequentially splitting the estimation data set, if there is only one term in the model. For a model with nonzero $\theta_1^{(lasso)}$, i.e. $2|\boldsymbol{\phi}_1^T \mathbf{y}| > \delta_1$, it is clear that $\theta_1^{(lasso)}$ satisfies

$$\theta_1^{(lasso)} = \big(\boldsymbol{\phi}_1^T \mathbf{y} - \frac{\lambda_1}{2}\text{sign}(\theta_1^{(LS)})\big)/\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1 \tag{15}$$

The model residual is

$$e(k, \lambda_1) = y(k) - \phi_1(k)\big(\boldsymbol{\phi}_1^T \mathbf{y} - \frac{\lambda_1}{2}\text{sign}(\theta_1^{(LS)})\big)/\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1 \tag{16}$$

If the data sample indexed at k is removed from estimation data set, the leave one out lasso parameter estimator is obtained by using only $(N-1)$ data points as

$$\theta_1^{(lasso,-k)} = \big[\big(\boldsymbol{\phi}_1^{(-k)}\big)^T \mathbf{y}^{(-k)} - \frac{\lambda_1}{2}\text{sign}(\theta_1^{(LS,-k)})\big]/\big(\boldsymbol{\phi}_1^{(-k)}\big)^T \boldsymbol{\phi}_1^{(-k)} \tag{17}$$

in which $\boldsymbol{\phi}_1^{(-k)}$ and $\mathbf{y}^{(-k)}$ denote the regressor and output vector respectively, with the kth element removed from $\boldsymbol{\phi}_1$ and $\mathbf{y}$, with the relations of

$$\big(\boldsymbol{\phi}_1^{(-k)}\big)^T \boldsymbol{\phi}_1^{(-k)} = \boldsymbol{\phi}_1^T \boldsymbol{\phi}_1 - [\phi_1(k)]^2 \tag{18}$$

$$\big(\boldsymbol{\phi}_1^{(-k)}\big)^T \mathbf{y}^{(-k)} = \boldsymbol{\phi}_1^T \mathbf{y} - \phi_1(k) y(k) \tag{19}$$

The leave one out error evaluated at k is given by

$$\begin{aligned} e^{(-k)}(k, \lambda_1) &= y(k) - \theta_1^{(LS,-k)} \phi_1(k) \\ &= y(k) - \frac{\phi_1(k)}{\big(\boldsymbol{\phi}_1^{(-k)}\big)^T \boldsymbol{\phi}_1^{(-k)}} \\ &\quad \times \big[\big(\boldsymbol{\phi}_1^{(-k)}\big)^T \mathbf{y}^{(-k)} - \frac{\lambda_1}{2}\text{sign}(\theta_1^{(LS,-k)})\big] \end{aligned} \tag{20}$$

From (18), we have

$$\frac{\phi_1(k)}{\big(\boldsymbol{\phi}_1^{(-k)}\big)^T \boldsymbol{\phi}_1^{(-k)}} = \frac{\phi_1(k)/\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1}{1 - [\phi_1(k)]^2/\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1} \tag{21}$$

Substitute (19) and (21) into (20)

$$\begin{aligned} e^{(-k)}(k, \lambda_1) &= y(k) - \frac{\phi_1(k)}{1 - [\phi_1(k)]^2/\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1} \\ &\quad \times \big[\boldsymbol{\phi}_1^T \mathbf{y} - \phi_1(k) y(k) - \frac{\lambda_1}{2}\text{sign}(\theta_1^{(LS,-k)})\big]/\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1 \\ &= \frac{1}{1 - [\phi_1(k)]^2/\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1} \\ &\quad \times \big[y(k) - \phi_1(k)\big(\boldsymbol{\phi}_1^T \mathbf{y} - \frac{\lambda_1}{2}\text{sign}(\theta_1^{(LS,-k)})\big)/\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1\big] \end{aligned} \tag{22}$$

If $\text{sign}(\theta_1^{(LS,-k)}) = \text{sign}(\theta_1^{(LS)})$, then by applying (16) we have

$$e^{(-k)}(k, \lambda_1) = w^{(1)}(k) e(k, \lambda_1) \tag{23}$$

where $w^{(1)}(k) = \frac{1}{1 - [\phi_1(k)]^2/\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1} > 0$. The leave one out mean square error (LOOMSE) can be calculated as

$$J(\lambda_1) = \sum_{k=1}^{N} [w^{(1)}(k)]^2 e^2(k, \lambda_1) \tag{24}$$

by assuming that $\text{sign}(\theta_1^{(LS,-k)}) = \text{sign}(\theta_1^{(LS)})$ holds for most data samples. We point out that in order for $\text{sign}(\theta_1^{(LS,-k)})$ and $\text{sign}(\theta_1^{(LS)})$ to be different, $\theta_1^{(LS)}$ needs to be very close to zero, which would be a violation to the assumption $2|\boldsymbol{\phi}_1^T \mathbf{y}| > \delta_1$. Hence we can treat $J(\lambda_1)$ in (24) as the exact LOOMSE

for sufficiently large value of δ_1. The setting of δ_1 is that it should be large enough for LOOMSE, but not too large in order to allow significant model terms to be included.

We further note that

$$e(k,\lambda_1) = \varepsilon(k) + \frac{\lambda_1}{2\boldsymbol{\phi}_1^T\boldsymbol{\phi}_1}\phi_1(k)\text{sign}(\theta_1^{(LS)}) \tag{25}$$

where $\varepsilon(k) = y(k) - \theta_1^{(LS)}\phi_1(k)$ is the model residual of least square estimate. By setting $\frac{\partial J(\lambda_1)}{\partial \lambda_1} = 0$, we obtain λ_1 as

$$\lambda_1 = -2\text{sign}(\theta_1^{(LS)})\boldsymbol{\phi}_1^T\boldsymbol{\phi}_1\boldsymbol{\phi}_1^T\mathbf{W}^{(1)}\boldsymbol{\varepsilon}/\boldsymbol{\phi}_1^T\mathbf{W}^{(1)}\boldsymbol{\phi}_1 \tag{26}$$

where $\mathbf{W}^{(1)} = \text{diag}\{[w^{(1)}(1)]^2, \cdots, [w^{(1)}(N)]^2\}$, where $\boldsymbol{\varepsilon} = [\varepsilon(1), \cdots, \varepsilon(N)]^T \in \Re^N$. We then calculate

$$\begin{aligned}\lambda_1^{opt} = \max\big[\min\big[2|\boldsymbol{\phi}_1^T\mathbf{y}|,\\ -2\text{sign}(\theta_1^{(LS)})\boldsymbol{\phi}_1^T\boldsymbol{\phi}_1\boldsymbol{\phi}_1^T\mathbf{W}^{(1)}\boldsymbol{\varepsilon}/\boldsymbol{\phi}_1^T\mathbf{W}^{(1)}\boldsymbol{\phi}_1\big], \delta\big]\end{aligned} \tag{27}$$

in order to satisfy the constraint that $\delta \le \lambda_1 \le 2|\boldsymbol{\phi}_1^T\mathbf{y}|$.

B. Fast parameter estimate calculation

We are interested in whether the computational cost can be reduced to minimal in estimating $\theta_1^{(lasso)}$. When $2|\boldsymbol{\phi}_1^T\mathbf{y}| > \delta_1$, our parameter estimate can be obtained by plugging (27) into (10) to yield $\theta_1^{(lasso)}$ with three possible results. If $\lambda_1 = \delta$, then $\theta_1^{(lasso)} = \theta_1^{(B)}$. If $\lambda_1 = 2|\boldsymbol{\phi}_1^T\mathbf{y}|$, then $\theta_1^{(lasso)} = 0$. Otherwise we obtain a nonzero $\theta_1^{(lasso)}$ with same sign, but smaller in magnitude than $\theta_1^{(B)}$.

Consider substituting (26) into the following equation;

$$\theta_1^{(test)} = \big(|\theta_1^{(LS)}| - \frac{\lambda_1}{2\boldsymbol{\phi}_1^T\boldsymbol{\phi}_1}\big)\text{sign}(\theta_1^{(LS)}) \tag{28}$$

which resembles (10), except that it is a continuous function without thresholding operation as in (10). $\theta_1^{(test)}$ is the same as $\theta_1^{(lasso)}$ of (10) if both $\theta_1^{(LS)}$ and $\theta_1^{(test)}$ have the same sign. Otherwise it indicates $\theta_1^{(lasso)} = 0$ for (10) with thresholding. Noting that $\boldsymbol{\varepsilon} = \mathbf{y} - \boldsymbol{\phi}_1\theta_1^{(LS)}$, we obtain

$$\theta_1^{(test)} = \frac{\boldsymbol{\phi}_1^T\mathbf{W}^{(1)}\mathbf{y}}{\boldsymbol{\phi}_1^T\mathbf{W}^{(1)}\boldsymbol{\phi}_1} \tag{29}$$

Thus we can use the following three extremely simple rules to determine $\theta_1^{(lasso)}$;

1) If $2|\boldsymbol{\phi}_1^T\mathbf{y}| < \delta_1$, set $\theta_1^{(lasso)} = 0$. Otherwise goto Step 2).
2) If $\text{sign}(\theta_1^{(test)}) \ne \text{sign}(\theta_1^{(LS)})$ then set $\theta_1^{(lasso)} = 0$. Otherwise goto 3).
3) Calculate both $\theta_1^{(test)}$ and $\theta_1^{(B)}$, and set $\theta_1^{(lasso)}$ as the one with the smaller magnitude.

IV. The Proposed Coordinate Descent Algorithm

In this section we present the proposed coordinate descent algorithm, incorporated with the fast calculation method in Section IV, for solving (5) in which $\boldsymbol{\lambda}$ are optimized based on LOOMSE. For a model with nonorthogonal terms, (10) cannot be extended into vector form. In order to exploit the simplicity of (27) and (10), we optimize each parameter one at a time using coordinate descent algorithm. Given an initial solution of $\boldsymbol{\theta}$, each element in $\boldsymbol{\theta}$ is optimized in turn by holding other elements fixed. Using the coordinate descent algorithm, we fit the associated parameter based on a single term model towards the model residual that has not yet been explained by all the fixed parameters. Hence (27) and (10) are still usable with appropriate modification as detailed in the following. The procedure is repeated until the parameters converge.

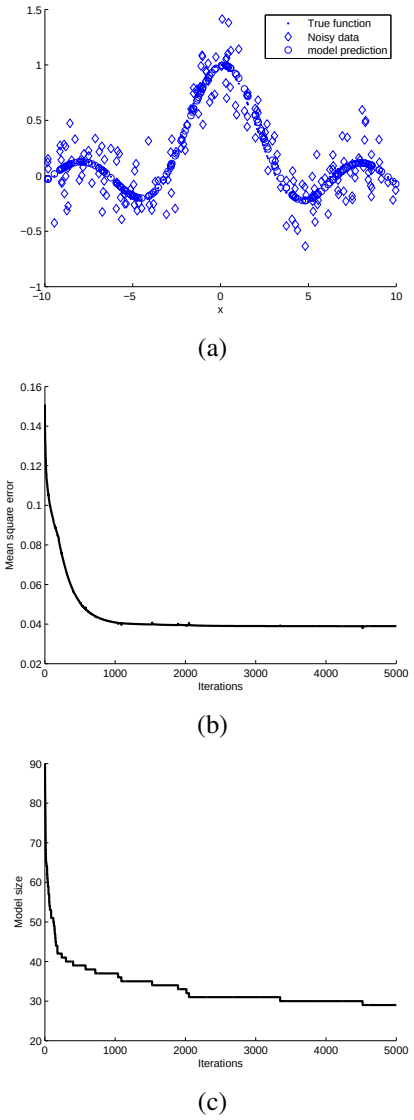

Fig. 1. The modeling results of the illustrative scalar function problem.

Specifically consider representing (2) as a single term model

$$\tilde{y}_j(k) = \theta_j\phi_j(\mathbf{x}(k)) + e(k), \tag{30}$$

where

$$\tilde{y}_j(k) = y(k) - \sum_{i=1, i\ne j}^{n_M}\tilde{\theta}_i\phi_i(\mathbf{x}(k)) \tag{31}$$

i.e. θ_i for $i \ne j$ are fixed at $\tilde{\theta}_i$. Clearly the desired model response $\tilde{y}_j(k)$ of (28) is simply the partial model residual

subject to $\tilde{\theta}_i$ for $i \neq j$.

The objective function is

$$L(\lambda_j, \theta_j) = \|\tilde{\mathbf{y}}_j - \theta_j \boldsymbol{\phi}_j\|^2 + \lambda_j |\theta|_j \tag{32}$$

where $\tilde{\mathbf{y}}_j = [\tilde{y}_j(1), \cdots, \tilde{y}_j(N)]^T \in \Re^N$. To minimize $L(\lambda_j, \theta_j)$ with respect to θ_j, we have solution similar to (15) as

$$\theta_j^{(lasso)} = \big(|\theta_j^{(PLS)}| - \frac{\lambda_j^{opt}}{2\boldsymbol{\phi}_j^T \boldsymbol{\phi}_j}\big)_+ \text{sign}(\theta_j^{(PLS)}) \tag{33}$$

where $\theta_j^{(PLS)} = \frac{\boldsymbol{\phi}_j^T \tilde{\mathbf{y}}_j}{\boldsymbol{\phi}_j^T \boldsymbol{\phi}_j}$ denotes the (partial) least square estimate by fitting θ_j using $\tilde{y}_j(k)$ as the target.

The regularization parameter minimizing LOOMSE has a similar form to (27) given by

$$\begin{aligned} \lambda_j^{opt} = \max \big[\min \big[& 2|\boldsymbol{\phi}_j^T \tilde{\mathbf{y}}_j|, \\ & - 2\text{sign}(\theta_j^{(PLS)}) \boldsymbol{\phi}_j^T \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \mathbf{W}^{(j)} \boldsymbol{\varepsilon}_j / \boldsymbol{\phi}_j^T \mathbf{W}^{(j)} \boldsymbol{\phi}_j \big], \delta \big] \end{aligned} \tag{34}$$

by replacing $\mathbf{y}$ with $\tilde{\mathbf{y}}_j$, and $\boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon}_j = [\varepsilon_j(1), \cdots, \varepsilon_j(N)]^T \in \Re^N$, in which $\varepsilon_j(k) = \tilde{y}_j(k) - \theta_j^{(PLS)} \phi_j(k)$ is model residual using all current model parameters. $\mathbf{W}^{(1)}$ is replaced by $\mathbf{W}^{(j)} = \text{diag}\{[w^{(j)}(1)]^2, \cdots, [w^{(j)}(N)]^2\}$, in which $w^{(j)}(k) = \frac{1}{1 - [\phi_j(k)]^2 / \boldsymbol{\phi}_j^T \boldsymbol{\phi}_j} > 0$.

Rather than directly calculating (34) and (33), which are for analysis, in our proposed algorithm below we use fast parameter estimation method as analyzed in Section IV. Denote $\bar{\boldsymbol{\phi}}_j = \mathbf{W}^{(j)} \boldsymbol{\phi}_j$, $\alpha_j = \boldsymbol{\phi}_j^T \boldsymbol{\phi}_j$ and $\beta_j = \boldsymbol{\phi}_j^T \mathbf{W}^{(j)} \boldsymbol{\phi}_j$. These vector/variables are stored in memory to minimize the computational cost, e.g. (see Step 3) below).

Initialize all $\tilde{\theta}_i$ as zeros. we repeat the following four steps for $j = 1, 2, \cdots, n_M,\ 1, 2, \cdots, n_M,\ 1, 2, \cdots$ until convergence.

1) Generate the partial model residual vector $\tilde{\mathbf{y}}_j$ according to (31).
2) If $2|\boldsymbol{\phi}_j^T \tilde{\mathbf{y}}_j| > \delta_1$, then set $\theta_j^{(lasso)} = 0$ and goto Step 6); Otherwise goto 3).
3) Calculate

$$\theta_j^{(PLS)} = \boldsymbol{\phi}_j^T \tilde{\mathbf{y}}_j / \alpha_j \tag{35}$$

$$\theta_j^{(B)} = \text{sign}(\theta_j^{(PLS)}) \big(|\theta_j^{(PLS)}| - \frac{\delta}{2\alpha_j}\big) \tag{36}$$

$$\theta_j^{(test)} = \bar{\boldsymbol{\phi}}_j^T \tilde{\mathbf{y}}_j / \beta_j \tag{37}$$

4) If $\text{sign}(\theta_j^{(PLS)}) \neq \text{sign}(\theta_j^{(test)})$, then set $\theta_j^{(lasso)} = 0$ and goto Step 6); Otherwise goto 5).
5) Set

$$\theta_j^{(lasso)} = \text{sign}(\theta_j^{(PLS)}) \min(|\theta_j^{(B)}|, |\theta_j^{(test)}|) \tag{38}$$

6) Set $\tilde{\theta}_j = \theta_j^{(lasso)}$.

The algorithm is terminated when a predetermined number of iterations is reached. Clearly with only two inner product calculations, the computational cost of updating a single parameter is extremely cheap. However the overall convergence rate of any coordinate descent algorithm can be problem dependent and is difficult to analyze. Thus the convergence of the proposed algorithm is still an open problem.

V. Numerical examples

Example 1: Consider using a RBF network to approximate an unknown scalar function

$$f(x) = \frac{\sin(x)}{x} \tag{39}$$

A data set of two hundred points was generated from $y = f(x) + \xi$, where the input x was uniformly distributed in [-10,10] and the noise ξ was Gaussian with zero mean and standard deviation 0.2. The data were very noisy. The Gaussian function

$$\phi_i(x) = \exp(-\frac{(x - c_i)^2}{2\tau^2}) \tag{40}$$

was used as the basis function to construct an RBF model, with a kernel width $\tau^2 = 10$. All the two hundred data points were used as the candidate RBF centre set for c_i. The proposed algorithm is applied with the parameters set as $\delta_1 = 2$ and $\delta = 0.03$. At the beginning all parameters are initialized as zeros (empty model). The snapshot of the modeling process is presented in Figure 1. It can be seen that over the iterations, the mean square error (MSE) is reduced rapidly to 0.04 at 1000 iterations. The model increases to a large model size at the beginning and reduces its size to 29 at 2000 iterations. By comparing the model prediction this 29-term model with noisy data and the true function in Figure 1(a), it is seen that the proposed method is capable of constructing sparse model to represent true function for this example.

Example 2: A simulated two dimensional nonlinear time series is given by

$$\begin{aligned} y(k) \ = \ & (0.8 - 0.5 \exp(-y^2(k-1)) y(t-1) \\ & -(0.3 + 0.9 \exp(-y^2(k-1)) y(t-2) \\ & +0.1 \sin(\pi y(k-1)) \end{aligned} \tag{41}$$

500 data samples were generated given $y(0) = 1$, $y(1) = 1$. The first 100 data points were used for training, and remaining data samples were used for model validation. We set $\mathbf{x} = [y(k-1), y(k-2)]^T$ and used 100 training data points as the candidate RBF centre set for $\mathbf{c}_i$. The Gaussian function $\phi_i(\mathbf{x}) = \exp(-\frac{(\|\mathbf{x} - \mathbf{c}_i\|^2}{2\tau^2})$ was used, with a kernel width $\tau = 0.3$. The proposed algorithm is applied with the parameters set as $\delta_1 = 0.001$ and $\delta = 0.5$, resulting a sparse RBF model with model size 36 at 5000 iterations. The modeling mean square error for the validation data set by the resultant 36-term RBF model is 5×10^{-5}. The phase plot of the actual data and that of model predictions using the 36-term RBF model over the validation data set is shown in Figure 2.

VI. Conclusions

We proposed a new coordinate descent optimization algorithm for sparse model construction of linear-in-the-parameter models from observational data. By using a penalized l_1 least square cost function together with the coordinate descent

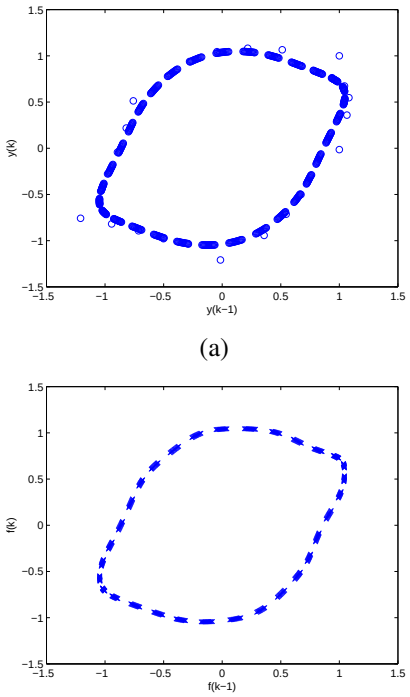

Fig. 2. The phase plots of the nonlinear time series example. (a) Time series data and (b) One-step ahead prediction over the validation data set.

framework, both the model parameter and the regularization parameter are estimated one at a time by minimizing the leave one out mean square error (LOOMSE). We derive a closed form of optimal LOOMSE regularization parameter for a single (assumingly significant) term model, for which we show that the LOOMSE can be analytically computed without actually splitting the data set leading to a very simple parameter estimation method. We then integrate the new results within the coordinate descent optimization algorithm and develop the model construction algorithm for linear-in-the-parameters models. Consequently a fully automated procedure is achieved without resort to any other validation data set for iterative model evaluation. Illustrative examples are included to demonstrate the effectiveness of the new approaches. Future researches will be focused on its applications to more practical signal processing problems.

VII. Acknowledgement

Junbin Gao and Xia Hong acknowledge that this work is supported by ARC via Grant DP130100364.

References

[1] C. J. Harris, X. Hong, and Q. Gan, *Adaptive Modelling, Estimation and Fusion from Data: A Neurofuzzy Approach*, Springer-Verlag, 2002.

[2] M. Brown and C. J. Harris, *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall, Hemel Hempstead, 1994.

[3] A. E. Ruano, *Intelligent Control Systems using Computational Intelligence Techniques*, IEE Publishing, 2005.

[4] R. Murray-Smith and T. A. Johansen, *Multiple Model Approaches to Modelling and Control*, Taylor and Francis, 1997.

[5] S. G. Fabri and V. Kadirkamanathan, *Functional Adaptive Control: An Intelligent Systems Approach*, Springer, 2001.

[6] M. Stone, "Cross validatory choie and assessment of statistical predictions," *Journal of the Royal Statistical Society, Series B*, vol. 36, pp. 117–147, 1974.

[7] S. Chen, Y. Wu, and B. L. Luk, "Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks," *IEEE Trans. on Neural Networks*, vol. 10, pp. 1239–1243, 1999.

[8] M. J. L. Orr, "Regularisation in the selection of radial basis function centers," *Neural Computation*, vol. 7, no. 3, pp. 954–975, 1995.

[9] X. Hong and Billings, "Parameter estimation based on stacked regression and evolutionary algorithms," *IEE Proc. - Control Theory and Applications*, vol. 146, no. 5, pp. 406–414, 1998.

[10] L. Ljung and T. Glad, *Modelling of Dynamic Systems*, Prentice Hall, Englewood Cliffs, NJ, 1994.

[11] R. H. Myers, *Classical and modern regression with applications*, PWS-KENT, Boston, 2nd edn., 1990.

[12] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their applications to non-linear system identification," *International Journal of Control*, vol. 50, pp. 1873–1896, 1989.

[13] M. J. Korenberg, "Identifying nonlinear difference equation and functional expansion representations: the fast orthogonal algorithm," *Annals of Biomedical Engineering*, vol. 16, pp. 123–142, 1988.

[14] L. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Trans. on Neural Networks*, vol. 5, pp. 807–814, 1992.

[15] X. Hong and C. J. Harris, "Neurofuzzy design and model construction of nonlinear dynamical processes from data," *IEE Proc. - Control Theory and Applications*, vol. 148, no. 6, pp. 530–538, 2001.

[16] Q. Zhang, "Using wavelets network in nonparametric estimation," *IEEE Trans. on Neural Networks*, vol. 8, no. 2, pp. 1997, 1993.

[17] S. A. Billings and H. L. Wei, "The wavelet-narmax representation: A hybrid model structure combining polynomial models with multiresolution wavelet decompositions," *International Journal of Systems Science*, vol. 36, no. 3, pp. 137 – 152, 2005.

[18] X. Hong, P. M. Sharkey, and K. Warwick, "Automatic nonlinear predictive model construction using forward regression and the PRESS statistic," *IEE Proc.-Control Theory Appl.*, vol. 150, no. 3, pp. 245–254, 2003.

[19] S. Chen, X. Hong, and C. J. Harris, "Sparse kernel regression modelling using combined locally regularised orthogonal least squares and D-optimality experimental design," *IEEE Trans. on Automatic Control*, vol. 48, no. 6, pp. 1029–1036, 2003.

[20] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. R. Stasti. Soc. B*, vol. 67, no. 2, pp. 301–320, 2005.

[21] S. Chen, "Locally regularised orthogonal least squares algorithm for the construction of sparse kernel regression models," in *Proceedings of 6th Int. Cof. Signal Processing*, Beijing, China, 2002, pp. 1229–1232.

[22] D. J. C. MacKay, *Bayesian Methods for Adaptive Models*, Ph.D. thesis, California Institute of Technology, USA, 1991.

[23] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[24] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of Royal Statistical Society. Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[25] B. Efron, I. Johnstone, T. Hastie, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–451, 2004.

[26] J. Friedman, T. Hastie, and R Tibshirami, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010.

[27] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirami, "Pathwise coordinate descent," *The Annals of Statistics*, vol. 1, no. 2, pp. 302–332, 2007.