Sparse Kernel Modelling: A Unified Approach

S. Chen[1], X. Hong[2], and C.J. Harris[1]

[1] School of Electronics and Computer Sciences, University of Southampton, Southampton SO17 1BJ, U.K. {sqc,cjh}@ecs.soton.ac.uk
[2] School of Systems Engineering, University of Reading, Reading RG6 6AY, U.K. x.hong@reading.ac.uk

Abstract. A unified approach is proposed for sparse kernel data modelling that includes regression and classification as well as probability density function estimation. The orthogonal-least-squares forward selection method based on the leave-one-out test criteria is presented within this unified data-modelling framework to construct sparse kernel models that generalise well. Examples from regression, classification and density estimation applications are used to illustrate the effectiveness of this generic sparse kernel data modelling approach.

1 Introduction

The objective of modelling from data is not that the model simply fits the training data well. Rather, the goodness of a model is characterised by its generalisation capability, interpretability and ease for knowledge extraction. All these desired properties depend crucially on the ability to construct appropriate sparse models by the modelling process, and a basic principle in practical data modelling is the parsimonious principle of ensuring the smallest possible model that explains the training data. Recently considerable research efforts have been focused on sparse kernel data modelling techniques [**?**,**?**,**?**,**?**,**?**,**?**]. Sparse kernel modelling methods typically use every training input data as a kernel. A sparse representation is then sought based on various criteria by making as many kernel weights to (near) zero values as possible. A different approach to these sparse kernel modelling methods is the forward selection using the orthogonal least squares (OLS) algorithm [**?**,**?**], developed in the late 80s for nonlinear system modelling, which remains highly popular for data modelling practicians.

Since its derivation, many enhanced variants of the OLS forward-selection algorithm have been proposed by incorporating the new developments from machining learning and the approach has extended its application to all the areas of data modelling, including regression, classification and kernel density estimation [**?**,**?**,**?**,**?**,**?**,**?**,**?**]. This contribution continues this theme, and it presents a unified framework for sparse kernel modelling that include all the three classes of data modelling applications, namely, regression, classification and probability density function (PDF) estimation. Based on this unified data-modelling framework, the OLS forward selection algorithm using the leave-one-out (LOO) test criteria and local regularisation (LR) is employed to construct sparse kernel models with

excellent generalisation capability. Experimental results are included to demonstrate the effectiveness of the OLS forward selection algorithm based on the LOO test criteria within the proposed unified data-modelling framework.

2 A Unified Framework for Data Modelling

Given the training data set, $D_N = \{\mathbf{x}_k, y_k\}_{k=1}^N$, where $\mathbf{x}_k = [x_{1,k}\ x_{2,k} \cdots x_{m,k}]^T \in \mathcal{R}^m$ is an observation sample and y_k is the target or desired response for $\mathbf{x}_k$, the task is to infer a kernel model of the form

$$\hat{y} = \hat{f}(\mathbf{x}; \boldsymbol{\beta}_N, \rho) = \sum_{i=1}^{N} \beta_i K_\rho(\mathbf{x}, \mathbf{x}_i) \tag{1}$$

to capture the underlying data generating mechanism, where $\hat{y}$ denotes the model output, $\boldsymbol{\beta}_N = [\beta_1\ \beta_2 \cdots \beta_N]^T$ is the kernel weight vector and $K_\rho(\bullet, \bullet)$ is the chosen kernel function with a kernel width ρ. Many types of kernel function can be employed and a commonly used one is the Gaussian function of the form

$$K_\rho(\mathbf{x}, \mathbf{c}_k) = \frac{1}{(2\pi\rho^2)^{m/2}} e^{-\frac{\|\mathbf{x}-\mathbf{c}_k\|^2}{2\rho^2}}, \tag{2}$$

where $\mathbf{c}_k \in \mathcal{R}^m$ is the k-th kernel centre vector. For regression and classification problems, the factor $\frac{1}{(2\pi\rho^2)^{m/2}}$ can be combined into kernel weights β_i. The generic kernel model (**??**) is defined by placing a kernel at each of the training input samples $\mathbf{x}_k$ and forming a linear combination of all the bases defined on the training data set. A sparse representation is then sought by selecting only N_s significant regressors from the full regressor set, where $N_s \ll N$.

The underlying data generating mechanism is governed by $y = f(\mathbf{x}) + \varepsilon$, where ε is a white process representing the observation noise. For regression problems, the unknown mapping $f : \mathcal{R}^m \to \mathcal{R}$. Regression is a supervised learning problem, as the desired response $y_k \in \mathcal{R}$ for the training data point $\mathbf{x}_k$ is given. For two-class classification problems, the unknown mapping $f : \mathcal{R}^m \to \{-1, +1\}$. The estimated class label for the pattern vector $\mathbf{x}_k$ is given by $\tilde{y}_k = \text{sgn}(\hat{y}_k)$ with

$$\text{sgn}(y) = \begin{cases} -1, & y \le 0, \\ +1, & y > 0. \end{cases} \tag{3}$$

Classification is also a supervised learning problem, since the correct label $y_k \in \{-1, +1\}$ for the training data point $\mathbf{x}_k$ is provided. For PDF estimation problems, the data $\{\mathbf{x}_k\}_{k=1}^N$ are drawn from a unknown density $f : \mathcal{R}^m \to \mathcal{R}_+$. Because f is a PDF, $f(\mathbf{x}) \ge 0$ for $\mathbf{x} \in \mathcal{R}^m$ and $\int_{\mathcal{R}^m} f(\mathbf{u})\, d\mathbf{u} = 1$. Thus, a kernel in a kernel density estimate must satisfy $K_\rho(\mathbf{x}, \mathbf{c}_k) \ge 0$ and $\int_{\mathcal{R}^m} K_\rho(\mathbf{u}, \mathbf{c}_k)\, d\mathbf{u} = 1$. Moreover the kernel weights must satisfy the nonnegative constraint

$$\beta_k \ge 0,\ 1 \le k \le N, \tag{4}$$

and the unity constraint

$$\boldsymbol{\beta}_N^T \mathbf{1}_N = 1, \tag{5}$$

where $\mathbf{1}_N$ denotes the vector of ones with dimension N. Kernel density estimation is an unsupervised learning problem because the desired response is unknown for each training data point $\mathbf{x}_k$. This difficult is circumvented by "inventing" a target function y_k for $\mathbf{x}_k$, so that the problem becomes a constrained regression one with the constraints (**??**) and (**??**). In particular, we choose y_k to be the value of the Parzen window estimate [**?**,**?**] at point $\mathbf{x}_k$. This choice of the desired response for density estimation is fully justified in [**?**].

Let the modelling error at training data point $\mathbf{x}_k$ be $\epsilon_k = y_k - \hat{y}_k$, where

$$\hat{y}_k = [K_\rho(\mathbf{x}_k, \mathbf{x}_1)\ K_\rho(\mathbf{x}_k, \mathbf{x}_2) \cdots K_\rho(\mathbf{x}_k, \mathbf{x}_N)]\ \boldsymbol{\beta}_N = \boldsymbol{\phi}^T(k)\boldsymbol{\beta}_N. \tag{6}$$

Define $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2 \cdots \boldsymbol{\phi}_N]$ with $\boldsymbol{\phi}_k = [K_\rho(\mathbf{x}_1, \mathbf{x}_k)\ K_\rho(\mathbf{x}_2, \mathbf{x}_k) \cdots K_\rho(\mathbf{x}_N, \mathbf{x}_k)]^T$ for $1 \le k \le N$, $\mathbf{y} = [y_1\ y_2 \cdots y_N]^T$ and $\boldsymbol{\epsilon} = [\epsilon_1\ \epsilon_2 \cdots \epsilon_N]^T$. The regression model (**??**) over the training data set D_N can then be expressed in the matrix form

$$\mathbf{y} = \boldsymbol{\Phi}\,\boldsymbol{\beta}_N + \boldsymbol{\epsilon}. \tag{7}$$

Let an orthogonal decomposition of the regression matrix $\boldsymbol{\Phi}$ be $\boldsymbol{\Phi} = \mathbf{W}\,\mathbf{A}_N$, where $\mathbf{A}_N$ is the $N \times N$ upper triangular matrix with unity diagonal elements, and $\mathbf{W} = [\mathbf{w}_1\ \mathbf{w}_2 \cdots \mathbf{w}_N]$ with orthogonal columns satisfying $\mathbf{w}_i^T \mathbf{w}_j = 0$, if $i \neq j$. The regression model (**??**) can alternatively be expressed as

$$\mathbf{y} = \mathbf{W}\,\mathbf{g}_N + \boldsymbol{\epsilon}, \tag{8}$$

where the weight vector $\mathbf{g}_N = [g_1\ g_2 \cdots g_N]^T$ satisfies the triangular system $\mathbf{A}_N\boldsymbol{\beta}_N = \mathbf{g}_N$. The model (**??**) is equivalently expressed by

$$\hat{y}_k = \mathbf{w}^T(k)\,\mathbf{g}_N, \tag{9}$$

where $\mathbf{w}^T(k) = [w_{k,1}\ w_{k,2} \cdots w_{k,N}]$ is the k-th row of $\mathbf{W}$.

3 Orthogonal-Least-Squares Algorithm

As established in the previous section, the regression, classification and PDF estimation can all be unified within the common regression modelling framework. Therefore, the OLS forward selection based on the LOO test criteria and local regularisation (OLS-LOO-LR) [**?**] provides an efficient algorithm to construct a sparse kernel model that generalise well.

3.1 Sparse Kernel Regression Model Construction

The LR aided least squares solution for the weight parameter vector $\mathbf{g}_N$ can be obtained by minimising the following regularised error criterion [**?**]

$$J_R(\mathbf{g}_N, \boldsymbol{\lambda}) = \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} + \mathbf{g}_N^T\boldsymbol{\Lambda}\mathbf{g}_N, \tag{10}$$

where $\boldsymbol{\lambda} = [\lambda_1 \ \lambda_2 \cdots \lambda_N]^T$ is the vector of regularisation parameters, and $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_N\}$. Applying the evidence procedure results in the following iterative updating formulas for the regularisation parameters [**?**]

$$\lambda_i^{\text{new}} = \frac{\gamma_i^{\text{old}}}{N - \gamma^{\text{old}}} \frac{\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}}{g_i^2}, \ 1 \leq i \leq N, \tag{11}$$

where g_i for $1 \leq i \leq N$ denote the current estimated parameter values, and

$$\gamma = \sum_{i=1}^{N} \gamma_i \quad \text{with} \quad \gamma_i = \frac{\mathbf{w}_i^T \mathbf{w}_i}{\lambda_i + \mathbf{w}_i^T \mathbf{w}_i}. \tag{12}$$

Typically a few iterations (less than 10) are sufficient to find a (near) optimal $\boldsymbol{\lambda}$. The use of LR is known to be capable of providing very sparse solutions [**?**,**?**].

For regression, the OLS-LOO-LR algorithm selects a sparse model by incrementally minimising the LOO mean square error (MSE) criterion, which is a measure of the model's generalisation performance [**?**,**?**,**?**]. At the n-th stage of the OLS selection procedure, an n-term model is selected. The LOO test error, denoted as $\epsilon_k^{(n,-k)}$, for the selected n-term model is defined as [**?**,**?**]

$$\epsilon_k^{(n,-k)} = \epsilon_k^{(n)} / \eta_k^{(n)}, \tag{13}$$

where $\epsilon_k^{(n)}$ is the usual n-term modelling error and $\eta_k^{(n)}$ is the associated LOO error weighting. The LOO MSE for the model with a size n is then defined by

$$J_n = \frac{1}{N} \sum_{k=1}^{N} \left(\epsilon_k^{(n,-k)} \right)^2 = \frac{1}{N} \sum_{k=1}^{N} \left(\epsilon_k^{(n)} \right)^2 / \left(\eta_k^{(n)} \right)^2. \tag{14}$$

This LOO MSE can be computed efficiently due to the fact that $\epsilon_k^{(n)}$ and $\eta_k^{(n)}$ can be calculated recursively according to [**?**,**?**]

$$\epsilon_k^{(n)} = \epsilon_k^{(n-1)} - w_{k,n} g_n \tag{15}$$

and

$$\eta_k^{(n)} = \eta_k^{(n-1)} - w_{k,n}^2 / \left(\mathbf{w}_n^T \mathbf{w}_n + \lambda_n \right), \tag{16}$$

respectively, where $w_{k,n}$ is the k-th element of $\mathbf{w}_n$. The selection is carried out as follows. At the n-th stage of the selection procedure, a model term is selected among the remaining n to N candidates if the resulting n-term model produces the smallest LOO MSE J_n. The selection procedure is terminated when

$$J_{N_s+1} \geq J_{N_s}, \tag{17}$$

yielding an N_s-term sparse model. The LOO statistic J_n is at least locally convex with respect to the model size n [**?**]. Thus, there exists an "optimal" model size N_s such that for $n \leq N_s$ J_n decreases as n increases while the condition (**??**) holds. The sparse regression model selection procedure is now summarised.

Initialisation: Set $\lambda_i = 10^{-6}$ for $1 \leq i \leq N$, and set iteration index $I = 1$.
Step 1: Given the current $\boldsymbol{\lambda}$ and with the following initial conditions

$$\epsilon_k^{(0)} = y_k, \eta_k^{(0)} = 1,\ 1 \leq k \leq N,\ \text{and}\ J_0 = \mathbf{y}^T\mathbf{y}/N, \tag{18}$$

use the OLS-LOO procedure [**?**] to select a subset model with N_I terms.
Step 2: Update $\boldsymbol{\lambda}$ using (**??**) and (**??**) with $N = N_I$. If the maximum iteration number (e.g. 10) is reached, stop; otherwise set $I+ = 1$ and go to *Step 1*.

3.2 Sparse Kernel Classifier Construction

The same LOO cross validation concept [**?**] is adopted to provide a measure of classifier's generalisation capability. Denote the test output of the LOO n-term model evaluated at the k-th data sample of D_N not used in training as $\hat{y}_k^{(n,-k)}$. The associated LOO signed decision variable is defined by

$$s_k^{(n,-k)} = \text{sgn}(y_k)\hat{y}_k^{(n,-k)} = y_k\hat{y}_k^{(n,-k)}, \tag{19}$$

where $\text{sgn}(y_k) = y_k$ since the class label $y_k \in \{-1, +1\}$. The LOO misclassification rate can be computed by

$$J_n = \frac{1}{N}\sum_{k=1}^{N}\mathcal{I}_d\left(s_k^{(n,-k)}\right), \tag{20}$$

where the indication function is defined by $\mathcal{I}_d(y) = 1$ if $y \leq 0$ and $\mathcal{I}_d(y) = 0$ if $y > 0$. The LOO misclassification rate J_n can be evaluated efficiently because $s_k^{(n,-k)}$ can be calculated very fast [**?**]. Specifically, express the LOO signed decision variable as $s_k^{(n,-k)} = \psi_k^{(n)}/\eta_k^{(n)}$. The recursive formula for $\eta_k^{(n)}$ is given in (**??**), while $\psi_k^{(n)}$ can be represented using the recursive formula [**?**]

$$\psi_k^{(n)} = \psi_k^{(n-1)} + y_k g_n w_{k,n} - w_{k,n}^2/\left(\mathbf{w}_n^T\mathbf{w}_n + \lambda_n\right). \tag{21}$$

The OLS-LOO-LR algorithm described in Subsection **??** can readily be applied to select a sparse kernel classifier with some minor modifications. Moreover, extensive empirical experience has suggested that all the regularisation parameters λ_i, $1 \leq i \leq N$, can be set to a small positive constant λ, and there is no need to update them using the evidence procedure. The sparse kernel classifier selection procedure based on this OLS-LOO algorithm is now summarised.

Setting λ to a small positive number, and with the following initial conditions

$$\psi_k^{(0)} = 0 \quad \text{and} \quad \eta_k^{(0)} = 1 \quad \text{for} \quad 1 \leq k \leq N, \quad \text{and} \quad J_0 = 1, \tag{22}$$

use the OLS-LOO procedure [**?**] to select a subset model with N_s terms.

The LOO misclassification rate J_n is also locally convex with respect to the classifier's size n. Thus there exists an optimal model size N_s such that for $n \leq N_s$ J_n decreases as n increases, while $J_{N_s} \leq J_{N_s+1}$. Therefore the selection procedure is automatically terminated with a subset classifier containing only N_s significant kernels.

3.3 Sparse Kernel Density Estimator Construction

Since the kernel density estimation problem can be expressed as a constrained regression modelling, the OLS-LOO-LR algorithm detailed in Subsection **??** can be used to select a sparse kernel density estimate. After the structure determination using the OLS-LOO-LR algorithm, a sparse N_s-term subset kernel model is obtained. Let $\mathbf{A}_{N_s}$ denote the subset matrix of $\mathbf{A}_N$, corresponding to the selected N_s-term subset model. The kernel weight vector $\boldsymbol{\beta}_{N_s}$, computed from $\mathbf{A}_{N_s}\boldsymbol{\beta}_{N_s} = \mathbf{g}_{N_s}$, may not satisfy the constraints (**??**) and (**??**). However, we can recalculate $\boldsymbol{\beta}_{N_s}$ using the multiplicative nonnegative quadratic programming (MNQP) algorithm [**?**,**?**]. Since N_s is very small, the extra computation involved is small. Formally, this task is defined as follows. Find $\boldsymbol{\beta}_{N_s}$ for the model

$$\mathbf{y} = \boldsymbol{\Phi}_{N_s}\boldsymbol{\beta}_{N_s} + \boldsymbol{\epsilon}, \tag{23}$$

subject to the constraints

$$\beta_i \geq 0,\ 1 \leq i \leq N_s, \tag{24}$$

$$\boldsymbol{\beta}_{N_s}^T \mathbf{1}_{N_s} = 1, \tag{25}$$

where $\boldsymbol{\Phi}_{N_s}$ denotes the selected subset regression matrix and $\boldsymbol{\beta}_{N_s}^T = [\beta_1\ \beta_2 \cdots \beta_{N_s}]$. The kernel weight vector can be obtained by solving the following constrained nonnegative quadratic programming

$$\begin{aligned} &\min_{\boldsymbol{\beta}_{N_s}} \{\tfrac{1}{2}\boldsymbol{\beta}_{N_s}^T \mathbf{C}_{N_s}\boldsymbol{\beta}_{N_s} - \mathbf{v}_{N_s}^T\boldsymbol{\beta}_{N_s}\} \\ &\text{s.t. } \boldsymbol{\beta}_{N_s}^T \mathbf{1}_{N_s} = 1 \text{ and } \beta_i \geq 0,\ 1 \leq i \leq N_s, \end{aligned} \tag{26}$$

where $\mathbf{C}_{N_s} = \boldsymbol{\Phi}_{N_s}^T\boldsymbol{\Phi}_{N_s} = [c_{i,j}] \in \mathcal{R}^{N_s \times N_s}$ is the related design matrix and $\mathbf{v}_{N_s} = \boldsymbol{\Phi}_{N_s}^T\mathbf{y} = [v_1\ v_2 \cdots v_{N_s}]^T$. Although there exists no closed-form solution for this optimisation problem, the solution can readily be obtained iteratively using a modified version of the MNQP algorithm [**?**].

Specifically, the iterative updating equations for $\boldsymbol{\beta}_{N_s}$ are given by [**?**,**?**]

$$r_i^{<t>} = \beta_i^{<t>}\left(\sum_{j=1}^{N_s} c_{i,j}\beta_j^{<t>}\right)^{-1},\ 1 \leq i \leq N_s, \tag{27}$$

$$h^{<t>} = \left(\sum_{i=1}^{N_s} r_i^{<t>}\right)^{-1}\left(1 - \sum_{i=1}^{N_s} r_i^{<t>} v_i\right), \tag{28}$$

$$\beta_i^{<t+1>} = r_i^{<t>}\left(v_i + h^{<t>}\right), \tag{29}$$

where the superindex $^{<t>}$ denotes the iteration index and h is the Lagrangian multiplier. During the iterative procedure, some of the kernel weights may be driven to (near) zero [**?**,**?**]. The corresponding kernels can then be removed from the kernel model, leading to a further reduction in the subset model size.

algorithm	model size	training MSE	test MSE
OLS-LOO-LR	58.6 ± 11.3	12.9690 ± 2.6628	17.4157 ± 4.6670
SVM	243.2 ± 5.3	6.7986 ± 0.4444	23.1750 ± 9.0459

Table 1. Comparison of modelling accuracy for the Boston housing data set. The results were averaged over 100 realizations and quoted as the mean±standard deviation.

4 Empirical Data Modelling Results

Boston housing data set. This was a regression benchmark data set, available at the UCI repository [**?**]. The data set comprised 506 data points with 14 variables. The task was to predict the median house value from the remaining 13 attributes. From the data set, 456 data points were randomly selected for training and the remaining 50 data points were used to form the test set. Because a Gaussian kernel was placed at each training data sample, there were $N = 456$ candidate regressors in the full regression model (**??**). The kernel width for the OLS-LOO-LR algorithm was determined via a grid-search based cross validation. The support vector machine (SVM) algorithm with the ε-insensitive cost function was also used to construct the regression model for this data set, as a comparsion. The three learning parameters of the SVM algorithm, the kernel width, error-band and trade-off parameters, were tuned via cross validation. Average results were given over 100 repetitions, and the two sparse Gaussian kernel models obtained by the OLS-LOO-LR and SVM algorithms, respectively, are compared in Table **??**.

For the particular computational platform used in the experiment, the recorded average run time for the OLS-LOO-LR algorithm when the kernel width was fixed was 200 times faster than the SVM algorithm when the kernel width, error-band and trade-off parameters were chosen. It can be seen from Table **??** that the OLS-LOO-LR algorithm achieved better modelling accuracy with a much sparser model than the SVM algorithm. The test MSE of the SVM algorithm was poor. This was probably because the three learning paremeters, namely the kernel width, error-band and trade-off parameters, were not tuned to the optimal values. For this regression problem of input dimension 13 and data size $N \approx 500$, the grid search required by the SVM algorithm to tune the three learning parameters was expensive and the optimal values of the three learning parameters were hard to find.

Diabetes data. This two-class classification benchmark data set was originated in the UCI repository [**?**] and the data set used in the experiment was obtained from [**?**]. The feature space dimension was $m = 8$. There were 100 realisations of the data set, each having 468 training patterns and 300 test patterns. Seven existing state-of-the-art radial basis function (RBF) and kernel classifiers were compared in [**?**,**?**]. The results given in [**?**] were reproduced in Table **??**. For the first 5 methods studied in [**?**], the nonlinear RBF network with 15 optimised Gaussian units was used. For the SVM algorithm with Gaussian kernel, no average model size was given in [**?**] but it could safely be assumed that it was

algorithm	test error rate	model size
RBF-Network	24.29 ± 1.88	15
AdaBoost RBF-Network	26.47 ± 2.29	15
LP-Reg-AdaBoost	24.11 ± 1.90	15
QP-Reg-AdaBoost	25.39 ± 2.20	15
AdaBoost-Reg	23.79 ± 1.80	15
SVM	23.53 ± 1.73	not available
Kernel Fisher Discriminant	23.21 ± 1.63	468
OLS-LOO	23.00 ± 1.70	6.0 ± 1.0

Table 2. Average classification test error rate in % over the 100 realizations of the diabetes data set. The first 7 results were quoted from [**?**].

much larger than 40. The kernel Fisher discriminant was the non-sparse optimal classifier using all the $N = 468$ training data samples as kernels.

The OLS-LOO algorithm was applied to construct sparse Gaussian kernel classifiers for this data set, and the results averaged over the 100 realisations are also listed in Table **??**. It can be seen that the proposed OLS-LOO method compared favourably with the existing benchmark RBF and kernel classifier construction algorithms, both in terms of classification accuracy and model size.

Six-dimensional density estimation. The underlying density to be estimated was given by

$$f(\mathbf{x}) = \frac{1}{3}\sum_{i=1}^{3} \frac{1}{(2\pi)^{6/2}} \frac{1}{\det^{1/2}|\boldsymbol{\Gamma}_i|} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Gamma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)} \tag{30}$$

with

$$\begin{aligned} \boldsymbol{\mu}_1 &= [1.0\ 1.0\ 1.0\ 1.0\ 1.0\ 1.0]^T, \\ \boldsymbol{\Gamma}_1 &= \mathrm{diag}\{1.0, 2.0, 1.0, 2.0, 1.0, 2.0\}, \end{aligned} \tag{31}$$

$$\begin{aligned} \boldsymbol{\mu}_2 &= [-1.0\ -1.0\ -1.0\ -1.0\ -1.0\ -1.0]^T, \\ \boldsymbol{\Gamma}_2 &= \mathrm{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\}, \end{aligned} \tag{32}$$

$$\begin{aligned} \boldsymbol{\mu}_3 &= [0.0\ 0.0\ 0.0\ 0.0\ 0.0\ 0.0]^T, \\ \boldsymbol{\Gamma}_3 &= \mathrm{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\}. \end{aligned} \tag{33}$$

A training data set of $N = 600$ randomly drawn samples was used to construct kernel density estimates, and a separate test data set of $N_{\text{test}} = 10,000$ samples was used to calculate the L_1 test error for the resulting estimate according to

$$L_1 = \frac{1}{N_{\text{test}}} \sum_{k=1}^{N_{\text{test}}} \left| f(\mathbf{x}_k) - \hat{f}(\mathbf{x}_k; \boldsymbol{\beta}_N, \rho) \right|. \tag{34}$$

The experiment was repeated $N_{\text{run}} = 100$ different random runs.

Simulation was used to test the proposed combined OLS-LOO-LR and MNQP algorithm and to compare its performance with the Parzen window estimator as

method	L_1 test error	kernel number
Parzen window estimate	$(3.5195 \pm 0.1616) \times 10^{-5}$	600 ± 0
SKD estimate of [**?**]	$(4.4781 \pm 1.2292) \times 10^{-5}$	14.9 ± 2.1
OLS-LOO-LR/MNQP	$(3.1134 \pm 0.5335) \times 10^{-5}$	9.4 ± 1.9

Table 3. Performance comparison for the six-dimensional three-Gaussian mixture.

well as our previous sparse kernel density (SKD) estimation algorithm [**?**]. The algorithm of [**?**], although also based on the OLS-LOO-LR regression framework, is very different from the current combined OLS-LOO-LR and MNQP algorithm. In particular, it transfers the kernels into the corresponding cumulative distribution functions and uses the empirical distribution function calculated on the training data set as the target function of the unknown cumulative distribution function. Moreover, in the work of [**?**], the unity constraint is met by normalising the kernel weight vector of the final selected model, which is nonoptimal, and the nonnegative constraint is ensured by adding a test to the OLS forward selection procedure, which imposes considerable computational cost.

The optimal kernel width was found to be $\rho = 0.65$ for the Parzen window estimate and $\rho = 1.2$ for both the previous SKD algorithm and the combined OLS-LOO-LR and MNQP algorithm, respectively, via cross validation. The results obtained by the three density estimator are summarised in Table **??**. It can be seen that the proposed combined OLS-LOO-LR and MNQP algorithm yielded sparser kernel density estimates with better test performance.

5 Conclusions

A regression framework has been proposed for sparse kernel modelling, which unifies the supervised regression and classification problems as well as the unsupervised PDF learning problem. An OLS algorithm has been developed for selecting sparse kernel models that generalise well, based on the LOO test criteria and coupled with local regularisation. For sparse kernel density estimation, a combined approach of the OLS-LOO-LR algorithm and multiplicative nonnegative quadratic programming has been proposed, with the OLS-LOO-LR algorithm selecting a sparse kernel density estimate while the MNQP algorithm computing the kernel weights of the selected model to meet the constraints for density estimate. Empirical data modelling results involving regression, classification and density estimation have been presented to demonstrate the effectiveness of the proposed unified data modelling framework based on the OLS-LOO-LR algorithm, and the results shown have confirmed that this unified sparse kernel regression framework offers a state-of-the-art for data modelling applications.

References

1. V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

2. M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Machine Learning Research*, vol.1, pp.211–244, 2001.
3. F. Sha, L.K. Saul and D.D. Lee, "Multiplicative updates for nonnegative quadratic programming in support vector machines," *Technical Report*, MS-CIS-02-19, University of Pennsylvania, USA, 2002.
4. B. Schölkopf and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.
5. V. Vapnik and S. Mukherjee, "Support vector method for multivariate density estimation," in: S, Solla, T. Leen and K.R. Müller (Eds.), *Advances in Neural Information Processing Systems*. MIT Press, 2000, pp.659–665.
6. M. Girolami and C. He, "Probability density estimation from optimally condensed data samples," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.25, no.10, pp.1253–1264, 2003.
7. S. Chen, S.A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol.50, no.5, pp.1873–1896, 1989.
8. S. Chen, C.F.N. Cowan and P.M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol.2, no.2, pp.302–309, 1991.
9. S. Chen, X. Hong and C.J. Harris, "Sparse kernel regression modelling using combined locally regularized orthogonal least squares and D-optimality experimental design," *IEEE Trans. Automatic Control*, vol.48, no.6, pp.1029–1036, 2003.
10. S. Chen, X. Hong, C.J. Harris and P.M. Sharkey, "Sparse modelling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol.34, no.2, pp.898–911, 2004.
11. S. Chen, "Local regularization assisted orthogonal least squares regression," *Neurocomputing*, vol.69, no.4-6, pp.559–585, 2006.
12. S. Chen, X. Hong and C.J. Harris, "Sparse kernel density construction using orthogonal forward regression with leave-one-out test score and local regularization," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol.34, no.4, pp.1708–1717, 2004.
13. S. Chen, X. Hong and C.J. Harris, "An orthogonal forward regression technique for sparse kernel density estimation," *Neurocomputing*, to appear, 2007.
14. X. Hong, P.M. Sharkey and K. Warwick, "Automatic nonlinear predictive model construction algorithm using forward regression and the PRESS statistic," *IEE Proc. Control Theory and Applications*, vol.150, no.3, pp.245–254, 2003.
15. X. Hong, S. Chen and C.J. Harris, "Fast kernel classifier construction using orthogonal forward selection to minimise leave-one-out misclassification rate," in: *Proc. 2nd Int. Conf. Intelligent Computing* (Kunming, China), Aug.16-19, 2006, pp.106–114.
16. E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol.33, pp.1066–1076, 1962.
17. B.W. Silverman, *Density Estimation for Statistics and Data Analysis*. London: Chapman Hall, 1986.
18. R.H. Myers, *Classical and Modern Regression with Applications* (2nd Edition). Boston, MA: PWS Pub. Co. 1990.
19. `http://www.ics.uci.edu/~mlearn/MLRepository.html`
20. `http://ida.first.fhg.de/projects/bench/benchmarks.htm`
21. G. Rätsch, T. Onoda and K.R. Müller, "Soft margins for AdaBoost," *Machine Learning*, vol.42, no.3, pp.287–320, 2001.