Orthogonal Least Squares Regression: An Efficient Approach for Parsimonious Modelling from Large Data

Sheng Chen

Communications, Signal Processing and Control Group
Electronics and Computer Science
Faculty of Physical and Applied Science
University of Southampton, Southampton SO17 1BJ, UK
E-mail: sqc@ecs.soton.ac.uk

Acknowledge: Dr Xia Hong, University of Reading; Professor Chris J. Harris, University of Southampton; Professor Stephen A. Billings, University of Sheffield

11th UK Workshop on Computational Intelligence
University of Manchester, Sept. 7-9, 2011

Outline

1. Orthogonal Forward Selection
   - Motivations
   - Previous Enhancements
   - Unified Data Modelling
2. Grey-Box Modelling
   - Incorporating Prior knowledge
   - Symmetric RBF Modelling
   - BVC RBF Modelling
3. Branch and Bound
   - Branch and Bound for Efficiency
   - Branch and Bound Aided OLS
4. Recent Extensions
   - New Enhancements

Outline

1. Orthogonal Forward Selection
   - Motivations
   - Previous Enhancements
   - Unified Data Modelling
2. Grey-Box Modelling
   - Incorporating Prior knowledge
   - Symmetric RBF Modelling
   - BVC RBF Modelling
3. Branch and Bound
   - Branch and Bound for Efficiency
   - Branch and Bound Aided OLS
4. Recent Extensions
   - New Enhancements

Nonlinear Identification

- In 80s, NARMAX identification of unknown **nonlinear** system

  $$\begin{aligned} y(k) &= f(u(k-1),\cdots,u(k-n_u),y(k-1),\cdots,y(k-n_y))+\epsilon(k) \\ &= f(\mathbf{x}(k))+\epsilon(k) \end{aligned}$$

  $y(k)$, $u(k)$ and $\epsilon(k)$: output, input and noise; system input vector with $m = n_u + n_y$:

  $$\begin{aligned} \mathbf{x}(k) &= [x_1(k)\cdots x_m(k)]^T \\ &= [u(k-1)\cdots u(k-n_u)\ y(k-1)\cdots y(k-n_y)]^T \end{aligned}$$

- Use **linear-in-the-parameters** nonlinear model

  $$\hat{y}(k) = \sum_{i=1}^{M} \theta_i p_i(k)$$

  $\{\theta_i\}$: unknown model weights; $\{p_i(k)\}$: fixed model bases, e.g. polynomial expansion, radial basis function, etc

- Utilise well-developed **linear** identification techniques

Parsimonious Principle

- Select subset of $M_s \ll M$ significantly model terms to overcome curse of dimensionality, overfitting, and poor generalisation
- Optimal subset selection intractable: candidate bases $M = 500$, subset size $M_s = 40 \Longrightarrow$ possible models to select from

$$\frac{M!}{M_s!(M - M_s)!} = \mathbf{2.2443 \times 10^{59}}$$

- Greedy-type forward subset selection

$$\left[\quad \overbrace{\mathbf{w}_1\ \mathbf{w}_2\ \cdots\ \mathbf{w}_{n-1}}^{\textbf{selected model terms}} \quad \middle| \quad \overbrace{\mathbf{p}_n\ \mathbf{p}_{n+1}\ \cdots\ \mathbf{p}_M}^{\textbf{candidate pool}} \right]$$

- Each time choose one term from candidate pool to add to subset model to maximally improve modelling performance

  $M = 500$ and $M_s = 40 \Longrightarrow$ candidate models to evaluate are:

$$\sum_{n=1}^{M_s} (M - n + 1) < M_s \times M = \mathbf{2 \times 10^4}$$

Orthogonal Decomposition

- Orthogonal decomposition of regression matrix: $\mathbf{P} = \mathbf{W}\mathbf{A}$ with

$$\mathbf{A} = \begin{bmatrix} 1 & \alpha_{1,2} & \cdots & \alpha_{1,M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \alpha_{M-1,M} \\ 0 & \cdots & 0 & 1 \end{bmatrix}$$

orthogonal $\mathbf{W} = [\mathbf{w}_1 \ \mathbf{w}_2 \cdots \mathbf{w}_M]$, $\mathbf{A}\theta = \mathbf{g}$ and equivalent model

$$\mathbf{y} = \mathbf{P}\theta + \epsilon \Leftrightarrow \mathbf{y} = \mathbf{W}\mathbf{g} + \epsilon$$

- Training error reduction ratio due to n-th model term

$$[\text{err}]_n = g_n^2 \mathbf{w}_n^T \mathbf{w}_n / \mathbf{y}^T \mathbf{y}$$

and training mean square error of n-term model

$$J^{(n)} = J^{(n-1)} - g_n^2 \mathbf{w}_n^T \mathbf{w}_n$$

Early Orthogonal Least Squares

- Orthogonal least squares methods and their application to non-linear system identification – S. Chen, S. A. Billings and W. Luo – International Journal of Control, 1989
  Google scholar citations: 645 ISI citations: 468 (July 2011) ECS EPrints downloads: average **1.5** per day

- Orthogonal least squares learning algorithm for radial basis function networks – S. Chen, C. F. N. Cowan and P. M. Grant – IEEE Transactions on Neural Networks, 1991
  Google scholar citations: 2166 ISI citations: 1555 (July 2011) ECS EPrints downloads: average **6** per day

Outline

1. Orthogonal Forward Selection
   - Motivations
   - Previous Enhancements
   - Unified Data Modelling
2. Grey-Box Modelling
   - Incorporating Prior knowledge
   - Symmetric RBF Modelling
   - BVC RBF Modelling
3. Branch and Bound
   - Branch and Bound for Efficiency
   - Branch and Bound Aided OLS
4. Recent Extensions
   - New Enhancements

2-Norm Local Regularisation

- Instead of training error $\epsilon^T\epsilon$, consider regularised error criterion

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

where $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_M\}$

- Regularised error reduction ratio

$$[\mathrm{rerr}]_n = g_n^2 \left(\mathbf{w}_n^T\mathbf{w}_n + \lambda_n\right) / \mathbf{y}^T\mathbf{y}$$

- Evidence procedure for updating regularisation parameters

$$\lambda_n^{\mathrm{new}} = \frac{\gamma_n^{\mathrm{old}}}{K - \gamma^{\mathrm{old}}} \frac{\epsilon^T\epsilon}{g_n^2}, \quad 1 \le n \le M$$

$$\gamma_n = \frac{\mathbf{w}_n^T\mathbf{w}_n}{\lambda_n + \mathbf{w}_n^T\mathbf{w}_n} \quad \gamma = \sum_{n=1}^{M} \gamma_n$$

which has a Bayesian interpretation

An Illustrative Example

- Very sparse, and enhance performance
- Additionally help to determine appropriate subset model size

selection stage l	weight θ_l	regulariser λ_l
1	1.87494e+00	2.53227e-01
2	-1.70014e+00	1.81540e-01
3	-1.00970e+00	2.01490e-01
4	5.67310e-01	8.64601e-01
5	4.17979e-01	1.36357e+00
6	-1.51352e-01	6.93984e-01
7	**-9.49873e-10**	**5.67623e+07**
8	**-2.79967e-10**	**1.11770e+08**
9	**7.14157e-11**	**1.03860e+07**
10	**-2.05313e-12**	**1.92708e+08**
	⋮	

Optimal Experiment Designs

- LS estimate $\boldsymbol{\theta}_{\mathrm{LS}} = \left(\mathbf{P}^T\mathbf{P}\right)^{-1}\mathbf{P}^T\mathbf{y}$ of true parameter vector $\boldsymbol{\theta}$:

$$E\left[\boldsymbol{\theta}_{\mathrm{LS}}\right] = \boldsymbol{\theta}, \quad \mathrm{Cov}\left[\boldsymbol{\theta}_{\mathrm{LS}}\right] \propto \left(\mathbf{P}^T\mathbf{P}\right)^{-1}$$

- Optimal experiment designs prevent selection of oversized ill-posed model and overcome problem of high parameter estimate variances
- *A*-optimal design minimises trace of the covariance matrix $\mathrm{Cov}\left[\boldsymbol{\theta}_{\mathrm{LS}}\right]$, which in orthogonal decomposition space is

$$\mathrm{tr}\left[\left(\mathbf{W}^T\mathbf{W}\right)^{-1}\right] = \sum_{n=1}^{M} \frac{1}{\mathbf{w}_n^T\mathbf{w}_n}$$

- *D*-optimal design maximises determinant of design matrix

$$\det\left[\mathbf{W}^T\mathbf{W}\right] = \prod_{n=1}^{M} \mathbf{w}_n^T\mathbf{w}_n$$

Combined LROLS and *D*-Optimality

- Combined LROLS and *D*-optimality criterion

$$J_{CR}(\mathbf{g},\boldsymbol{\lambda},\beta) = J_R(\mathbf{g},\boldsymbol{\lambda}) + \beta \sum_{n=1}^{M} -\log\left(\mathbf{w}_n^T \mathbf{w}_n\right)$$

- Combined regularised error reduction and *D*-optimality ratio

$$[\mathrm{crerr}]_n = \left(g_n^2\left(\mathbf{w}_n^T\mathbf{w}_n + \lambda_n\right) + \beta \log\left(\mathbf{w}_n^T\mathbf{w}_n\right)\right)/\mathbf{y}^T\mathbf{y}$$

- Or selecting *n*-th model term by minimising combined criterion

$$J^{(n)} = J^{(n-1)} - g_n^2\left(\mathbf{w}_n^T\mathbf{w}_n + \lambda_n\right) - \beta \log\left(\mathbf{w}_n^T\mathbf{w}_n\right)$$

- 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, 1029–1036, June 2003

Leave-One-Out Cross Validation

- Highly desirable to select model terms by directly optimising model generalisation performance, instead of training MSE
- Model generalisation can be evaluated by test performance on data not used in training, and leave-one-out cross validation:
- "Remove" kth data from training set $D_K = \{\mathbf{x}(k), y(k)\}_{k=1}^{K}$, identify model $\hat{y}^{(n,-k)}$, and test error on data point not in training

$$\epsilon^{(n,-k)}(k) = y(k) - \hat{y}^{(n,-k)}(k)$$

- "Repeating" for each k leads to LOO MSE

$$J^{(n)} = \frac{1}{K} \sum_{k=1}^{K} \left(\epsilon^{(n,-k)}(k) \right)^2$$

a generalisation measure for model $\hat{y}^{(n)}$ identified with whole D_K

OLS-LOO Algorithm

- All above LOO cross validation steps are *virtual*, and orthogonal decomposition makes everything simple
- Leave-one-out error

$$\epsilon^{(n,-k)}(k) = \frac{\epsilon^{(n)}(k)}{\eta^{(n)}(k)}$$

- Modelling error of n-term model $\hat{y}^{(n)}$

$$\epsilon^{(n)}(k) = \epsilon^{(n-1)}(k) - w_n(k)g_n$$

$\epsilon^{(n-1)}(k)$ is modelling error of $(n-1)$-term model $\hat{y}^{(n-1)}$

- Leave-one-out weighting

$$\eta^{(n)}(k) = \eta^{(n-1)}(k) - \frac{w_n^2(k)}{\mathbf{w}_n^T\mathbf{w}_n + \lambda_n}$$

$w_n(k)$ is kth element of nth model column $\mathbf{w}_n$

OLS-LOO Procedure

- Thus, leave-one-out mean square error $J^{(n)}$ can be evaluated efficiently
- Moreover $J^{(n)}$ is "locally convex" with respect to model size n, and there exists an "optimal" model size M_s such that
  - For $n \leq M_s$: $J^{(n)}$ decreases as n increases
  - while $J^{(M_s)} \leq J^{(M_s+1)}$
- Regularised OLS algorithm can readily used, but selection of nth model term is based on minimisation of $J^{(n)}$
- 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, 898–911, 2004

Outline

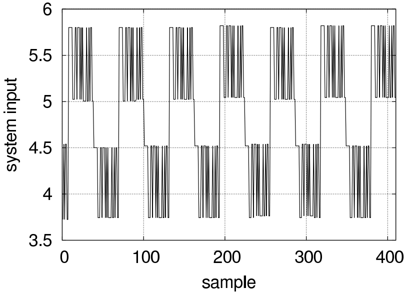
1. Orthogonal Forward Selection
   - Motivations
   - Previous Enhancements
   - Unified Data Modelling
2. Grey-Box Modelling
   - Incorporating Prior knowledge
   - Symmetric RBF Modelling
   - BVC RBF Modelling
3. Branch and Bound
   - Branch and Bound for Efficiency
   - Branch and Bound Aided OLS
4. Recent Extensions
   - New Enhancements

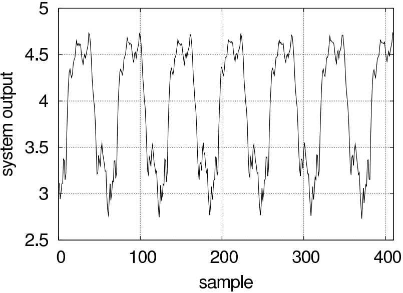
Unified Regression Framework

- Originally derived for regression, all algorithms can be applied to classification and density estimation as well
  - Regression and classification are supervised learning, while density estimation is unsupervised learning
- Two-class classification: give training set $D_K = \{\mathbf{x}(k), y(k)\}_{k=1}^{K}$, where $y(k) \in \{-1, +1\}$, OLS forward selection based on
  - Fisher ratio of interclass difference to intraclass spread
  - Leave-one-out misclassification rate
- Probability density function estimation: give training set $D_K = \{\mathbf{x}(k)\}_{k=1}^{K}$, construct Parzen window estimate on D_K
  - Use PW estimate at $\mathbf{x}(k)$ as $y(k) \rightarrow$ regression problem
  - Weights must be nonnegative and add up to unity

Engine Data Set

- Data collected from a Leyland TL11 turbocharged, direct injection diesel engine operated at low engine speed
- System input $u(k)$ is fuel rack position, and system output $y(k)$ is engine speed





- First 210 data points for training, and last 200 data for testing

Engine Data Results

- Training data $\{\mathbf{x}(k), y(k)\}_{k=1}^{K}$ with $K = 210$, and

$$\mathbf{x}(k) = [y(k-1)\ u(k-1)\ u(k-2)]^T$$

- LROLS-LOO: Gaussian RBF, RBF variance σ^2 determined separately by cross validation
- SVM: Gaussian kernel, kernel variance σ^2, regularisation parameter and error band determined separately by cross validation
- Experimental results:

algorithm	model size	training MSE	test MSE
LROLS-LOO	22	0.000453	0.000490
SVM	92	0.000447	0.000498

Boston Housing Data

- Regression benchmark, comprised 506 data points with 14 variables
  - Predict median house value from remaining 13 attributes
  - 456 data points were randomly selected for training and remaining 50 data points for testing
  - Average results were given over 100 repetitions
  - Gaussian kernel was used
- Experimental results:

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

The SVM model is overfitted, due to the difficulties in finding near optimal values for three hyperparameters, kernel variance, regularisation parameter and error band

Diabetes Data Set

- Two-class, feature space dimension $m = 8$; 100 realisations, each having 468 training patterns and 300 test patterns
- Experimental results:

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
ROLS-LOO	23.00 ± 1.70	6.0 ± 1.0

Data and first 7 results from:

`http://ida.first.fhg.de/projects/bench/benchmarks.htm`

Thyroid Data Set

- Two-class, feature space dimension $m = 5$; 100 realisations, each having 140 training patterns and 75 test patterns
- Experimental results:

algorithm	test error rate %	model size
RBF-Network	4.52 ± 2.12	8
AdaBoost RBF-Network	4.40 ± 2.18	8
LP-Reg-AdaBoost	4.59 ± 2.22	8
QP-Reg-AdaBoost	4.35 ± 2.18	8
AdaBoost-Reg	4.55 ± 2.19	8
SVM	4.80 ± 2.19	not available
Kernel Fisher Discriminant	4.20 ± 2.07	140
ROLS-LOO	4.80 ± 2.20	4.6 ± 1.0

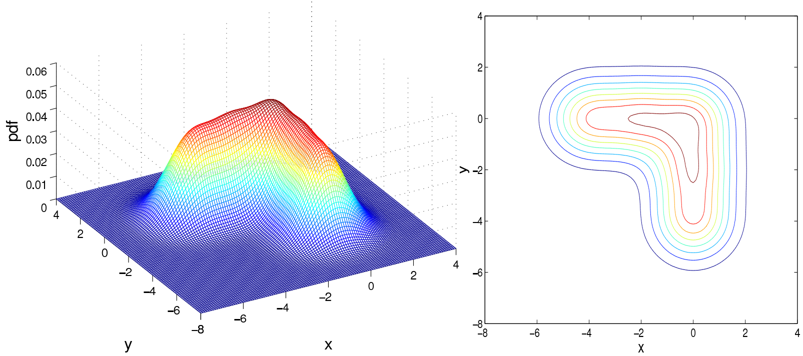
Data and first 7 results from:

`http://ida.first.fhg.de/projects/bench/benchmarks.htm`

2-D Density Example

$$p(x_1,x_2)=\sum_{i=1}^{5}\frac{1}{10\pi}e^{-\frac{(x_1-\mu_{i,1})^2}{2}}e^{-\frac{(x_2-\mu_{i,2})^2}{2}}$$

Means of 5 Gaussians: $[0.0\ -4.0]$, $[0.0\ -2.0]$, $[0.0\ 0.0]$, $[-2.0\ 0.0]$, $[-4.0\ 0.0]$



- Estimation set $K=500$, and experiment repeated 100 times

2-D Density Example Results

- Kernel width was obtained separately via cross validation
- L_1 test error and numerical approximation of Kullback-Leibler divergence are used to assess an estimator
- Average kernel number obtained by OLS with D-optimality is 8
- GMM: Gaussian mixture model estimate, number of mixture componenets set to 8
- RSDE: reduced set density estimate (Girolami & He, 2003)
- Experimental results:

estimator	PW	OLS D-opt	RSDE	GMM
$L_1 \times 10^3$	3.62 ± 0.44	3.24 ± 0.56	3.63 ± 0.36	3.68 ± 0.67
KLC $\times 10^2$	3.42 ± 0.55	3.47 ± 1.30	3.54 ± 0.49	3.39 ± 0.87
kernel no.	500	7.9 ± 0.8	13.2 ± 3.0	8
maximum	500	9	21	8
minimum	500	6	6	8

6-D Density Example

- True density was mixture of three Gaussian distributions

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

- with

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

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

$$\begin{aligned}\bar{\boldsymbol{\mu}}_3 &= [0.0\ 0.0\ 0.0\ 0.0\ 0.0\ 0.0]^T,\\ \bar{\boldsymbol{\Gamma}}_3 &= \text{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\}\end{aligned}$$

- Estimation set $K = 600$, while experiment is repeated 100 times

6-D Density Example Results

- Kernel width was obtained separately via cross validation
- Average kernel number obtained by OLS with *D*-optimality design is 8.4
- GMM: number of mixture componenets set to 8
- RSDE: reduced set density estimate (Girolami & He, 2003)
- Experimental results:

estimator	PW	OLS *D*-opt	RSDE	GMM
L_1 $\times 10^5$	3.52 ± 0.16	2.78 ± 0.23	2.74 ± 0.50	1.74 ± 0.29
kernel no.	600	8.4 ± 0.9	14.2 ± 3.6	8
maximum	600	10	25	8
minimum	600	6	8	8

Outline

1. Orthogonal Forward Selection
   - Motivations
   - Previous Enhancements
   - Unified Data Modelling
2. Grey-Box Modelling
   - Incorporating Prior knowledge
   - Symmetric RBF Modelling
   - BVC RBF Modelling
3. Branch and Bound
   - Branch and Bound for Efficiency
   - Branch and Bound Aided OLS
4. Recent Extensions
   - New Enhancements

Motivations

- Like many existing data modelling methods, the approach discussed so far is a black-box model, which is appropriate
  - if no *a priori* information exists regarding underlying data generating mechanism
- Known prior knowledge concerning underlying process should be incorporated into model structure explicitly
- How to incorporate prior knowledge to form grey-box model is highly problem dependent, and is really an *art*
- Two types of prior information are considered
  - Underlying process exhibits known symmetry property
  - Underlying process obeys set of boundary value constraints
- Existing learning algorithms can be applied to resulting grey-box models without any modification and added complexity

Outline

1. Orthogonal Forward Selection
   - Motivations
   - Previous Enhancements
   - Unified Data Modelling
2. Grey-Box Modelling
   - Incorporating Prior knowledge
   - Symmetric RBF Modelling
   - BVC RBF Modelling
3. Branch and Bound
   - Branch and Bound for Efficiency
   - Branch and Bound Aided OLS
4. Recent Extensions
   - New Enhancements

Symmetric RBF Network

- Unknown system $f(\bullet)$ possesses odd symmetry $f(-\mathbf{x}) = -f(\mathbf{x})$
  - e.g. from physics, underlying optimal discriminant function for BPSK digital signals has old symmetry
- RBF model with standard node

$$p_i(k) = \varphi\left(\|\mathbf{x}(k) - \mathbf{c}_i\|/\sigma\right)$$

cannot guarantee to have odd symmetry

- Symmetric RBF model with symmetric RBF node

$$p_i(k) = \varphi\left(\|\mathbf{x}(k) - \mathbf{c}_i\|/\sigma\right) - \varphi\left(\|\mathbf{x}(k) + \mathbf{c}_i\|/\sigma\right)$$

guarantees to obey same odd symmetry as underlying process

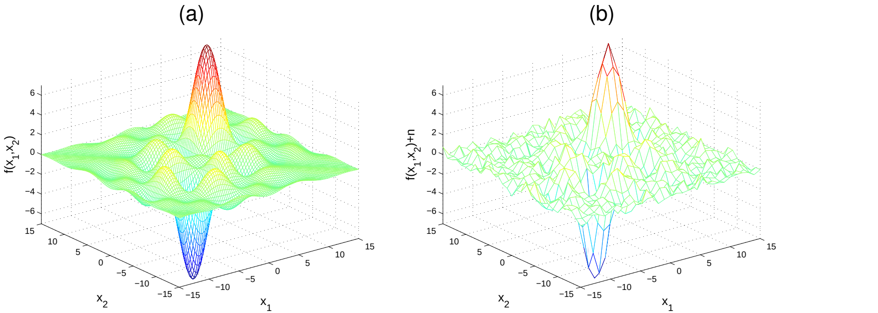
  - incorporate prior information naturally into model structure
  - all RBF learning methods are readily applicable

Symmetric Function Modelling

(a) Underlying function

$$f(x_1, x_2) = 10\left(\frac{\sin(x_1 - 5)\sin(x_2 - 5)}{(x_1 - 5)(x_2 - 5)} - \frac{\sin(x_1 + 5)\sin(x_2 + 5)}{(x_1 + 5)(x_2 + 5)}\right)$$

shown on the grid of 90601 points, and (b) 961 noisy training data points $y = f(x_1, x_2) + \epsilon$, where ϵ is Gaussian noise of zero mean and variance 0.16



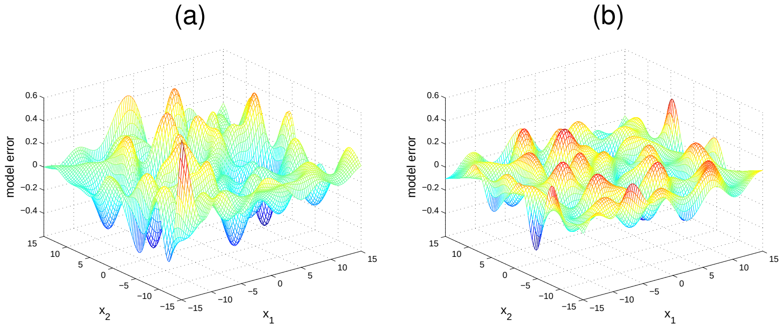
Symmetric Modelling Results

- Every training data used as a RBF centre with $M = K = 961$, RBF variance $\sigma^2 = 8.0$ was determined separately using cross validation
- Local regularisation assisted OLS algorithm with LOO MSE was used to automatically select sparse RBF / SRBF model
- Mean square error MSE $= E[(y - \hat{y})^2]$ was calculated over noisy training set and a separate noisy test set
- Mean modelling error MME $= E[(f(x_1, x_2) - \hat{f}(x_1, x_2))^2]$ was defined over grid of 90601 points noise-free $f(x_1, x_2)$, with $\hat{f}$ denoting estimated mapping

	model size	training MSE	test MSE	test MME
RBF	105	0.1543	0.2047	0.0294
SRBF	68	0.1566	0.1839	0.0093

Symmetric Modelling (continue)

(a) modelling error $f(x_1, x_2) - \hat{f}(x_1, x_2)$ of standard RBF model, and
(b) modelling error $f(x_1, x_2) - \hat{f}(x_1, x_2)$ of symmetric RBF model



Results Analysis

- By incorporating prior information, SRBF offers significantly better generalisation performance than standard RBF
  - Mean modelling error is three times smaller
- OLS algorithm selecting M_s model terms from K-term candidate set, where $M_s \ll K$, has complexity

  $$C = \big(M_s + 1\big) \times K \times \mathcal{O}(K)$$

  For SRBF, $M_s = 68$, while for standard RBF, $M_s = 105$
  - Thus, complexity of SRBF model construction is about half of complexity for constructing standard RBF model
- Computational requirements of a symmetric node is more than that of standard one, but SRBF has few RBF units
  - Prediction complexity of two models are similar

Outline

1. Orthogonal Forward Selection
   - Motivations
   - Previous Enhancements
   - Unified Data Modelling
2. Grey-Box Modelling
   - Incorporating Prior knowledge
   - Symmetric RBF Modelling
   - BVC RBF Modelling
3. Branch and Bound
   - Branch and Bound for Efficiency
   - Branch and Bound Aided OLS
4. Recent Extensions
   - New Enhancements

Boundary Value Constraints

- Underlying system satisfies a set of boundary value constraints

$$f(\mathbf{x}_j) = d_j, \ 1 \leq j \leq L$$

  $\mathbf{x}_j$ and d_j, $1 \leq j \leq L$, are known

  - These BVCs may represent the fact that at some critical regions, there is a complete knowledge about system

- Any identified model $\hat{f}$ is required to strictly meet these BVCs

$$\hat{f}(\mathbf{x}_j) = d_j, \ 1 \leq j \leq L$$

  - RBF model with standard node $p_i(k) = \varphi\left(\|\mathbf{x}(k) - \mathbf{c}_i\|/\sigma\right)$ cannot meet these BVCs

- Using BVCs as constraints dramatically complicates learning

  - Efficient state-of-the-art learning methods cannot be applied directly

BVC-RBF Network

- Boundary value constraint-RBF model takes the form

$$\hat{y}(k) = \hat{f}(\mathbf{x}(k)) = \sum_{i=1}^{M} p_i(\mathbf{x}(k))\theta_i + g(\mathbf{x}(k))$$

- with novel RBF node structure

$$p_i(\mathbf{x}) = h(\mathbf{x})\varphi(\|\mathbf{x} - \mathbf{c}_i\|/\sigma)$$

- **Geometric mean** of data sample $\mathbf{x}$ to BVCs $\mathbf{x}_j$, $1 \leq j \leq L$

$$h(\mathbf{x}) = \sqrt[L]{\prod_{j=1}^{L} \|\mathbf{x} - \mathbf{x}_j\|}$$

- Since $h(\mathbf{x}_j) = 0$ at any boundary point $\mathbf{x}_j$, node $p_i(\mathbf{x})$ has property of **zero forcing** at any $\mathbf{x}_j$

BVC-RBF Offset Function

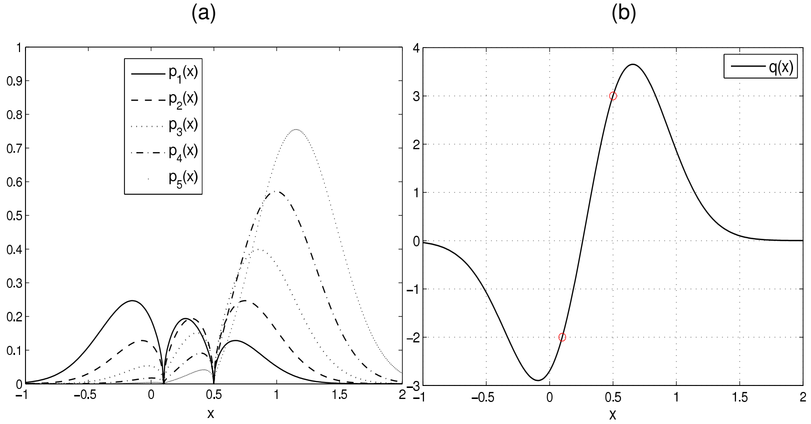
- **Offset function** $$g(\mathbf{x}) = \sum_{j=1}^{L} \alpha_j e^{-\frac{\|\mathbf{x}-\mathbf{x}_j\|^2}{\tau}}$$
- τ is a positive scalar, $\boldsymbol{\alpha} = [\alpha_1 \ \alpha_2 \cdots \alpha_L]^T$ is obtained by solving $g(\mathbf{x}_j) = d_j$, $1 \leq j \leq L$, i.e. $\boldsymbol{\alpha} = \mathbf{G}^{-1}\mathbf{d}$, with $\mathbf{d} = [d_1 \ d_2 \cdots d_L]^T$ and

$$\mathbf{G} = \begin{bmatrix} 1 & e^{-\frac{\|\mathbf{x}_1-\mathbf{x}_2\|^2}{\tau}} & \cdots & e^{-\frac{\|\mathbf{x}_1-\mathbf{x}_L\|^2}{\tau}} \\ e^{-\frac{\|\mathbf{x}_2-\mathbf{x}_1\|^2}{\tau}} & 1 & \ddots & e^{-\frac{\|\mathbf{x}_2-\mathbf{x}_L\|^2}{\tau}} \\ \vdots & \ddots & \ddots & \vdots \\ e^{-\frac{\|\mathbf{x}_L-\mathbf{x}_1\|^2}{\tau}} & e^{-\frac{\|\mathbf{x}_L-\mathbf{x}_2\|^2}{\tau}} & \cdots & 1 \end{bmatrix}$$

- Offset function $g(\mathbf{x})$ passes all predetermined boundary values $f(\mathbf{x}_j) = g(\mathbf{x}_j) = d_j$, $1 \leq j \leq L$, and it is completely determined by BVCs but does not depend on D_K

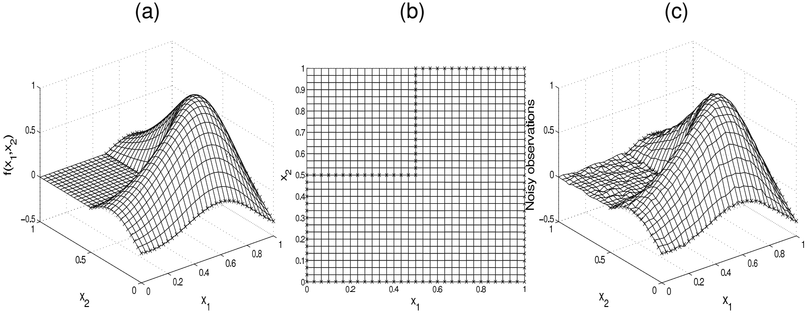
BVC-RBF Illustration

- One-dimensional function $f(x)$ with two BVCs: $f(0.1) = -2$, $f(0.5) = 3$
- Five RBFs with zero forcing at two boundary points (a), and offset passing function $g(x)$ (b)



BVC-Function Modelling

(a) Underlying function $f(x_1, x_2)$ shown on grid of 961 points, (b) $L = 120$ BVCs given by coordinates marked as cross points, and (c) 961 noisy training points, with Gaussian noise of zero mean and variance 0.01^2

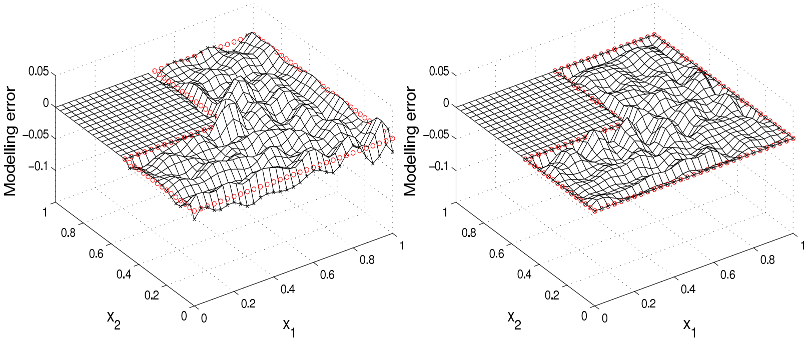


- OLS algorithm with training MSE and *D*-optimality was used to automatically identify standard RBF and BVC-RBF models
- RBF variance $\sigma^2 = 0.01$ was determined by cross validation, $\tau = 0.04$, and *D*-optimality weighting $\beta = 10^{-5}$

BVC-Function Modelling Results

	model size	training MSE (inside D_K)	test MME (inside boundary)	test MME (on boundary)
RBF	91	1.6894×10^{-4}	1.0229×10^{-4}	2.1249×10^{-4}
BVC-RBF	68	1.0736×10^{-4}	4.3787×10^{-5}	7.2598×10^{-11}

Modelling error $f(x_1, x_2) - \hat{f}(x_1, x_2)$ of standard RBF (a) and BVC-RBF (b)


(a)

(b)

Outline

1. Orthogonal Forward Selection
   - Motivations
   - Previous Enhancements
   - Unified Data Modelling
2. Grey-Box Modelling
   - Incorporating Prior knowledge
   - Symmetric RBF Modelling
   - BVC RBF Modelling
3. Branch and Bound
   - Branch and Bound for Efficiency
   - Branch and Bound Aided OLS
4. Recent Extensions
   - New Enhancements

Motivations

*n*th stage of OLS forward subset selection

$$\left[\overbrace{\mathbf{w}_1\ \mathbf{w}_2\ \cdots\ \mathbf{w}_{n-1}}^{\textbf{selected subset model}} \;\middle|\; \overbrace{\mathbf{p}_n\ \mathbf{p}_{n+1}\ \cdots\ \mathbf{p}_M}^{\textbf{candidate set } \mathcal{S}}\right]$$

- choose one term from candidate set $\mathcal{S}$ as $\mathbf{w}_n$ to add to subset model which maximumly improves modelling performance

With Branch and bound, *n*th stage of OLS forward subset selection

$$\left[\overbrace{\mathbf{w}_1\ \mathbf{w}_2\ \cdots\ \mathbf{w}_{n-1}}^{\textbf{selected subset model}} \;\middle|\; \overbrace{\mathbf{p}_n\ \mathbf{p}_{n+1}\ \cdots\ \mathbf{p}_{M_n}}^{\textbf{candidate set } \mathcal{S}} \;\middle|\; \overbrace{\mathbf{p}_{M_n+1}\ \mathbf{p}_{M_n+2}\ \cdots\ \mathbf{p}_M}^{\textbf{infeasible set } \bar{\mathcal{S}}}\right]$$

- choose one term from candidate set $\mathcal{S}$ as $\mathbf{w}_n$ to add to subset model, and check any candidate in $\mathcal{S}$ can be safely removed to infeasible set $\bar{\mathcal{S}}$ (will not be considered in subsequent stages)

What is Branch and Bound

- An evaluation procedure for all candidate solutions by using upper and lower estimated bounds of the quantity optimised, leading to large subsets of fruitless candidates being discarded
  - Branching: successively dividing a candidate solution set into subsets
  - Bounding: computing upper and lower bounds for a given subset
- Let candidate set be divided into two disjoint subsets, $\mathcal{A}$ and $\mathcal{B}$, and a bounding function is based on current best solution
  - If lower bound for $\mathcal{A}$ is greater than current best solution, it is discarded, and search space is reduced to $\mathcal{B}$
- It is often difficult to design a branch and bound strategy for specific problem
  - For OLS algorithm, it can be implemented effectively

Outline

1. Orthogonal Forward Selection
   - Motivations
   - Previous Enhancements
   - Unified Data Modelling
2. Grey-Box Modelling
   - Incorporating Prior knowledge
   - Symmetric RBF Modelling
   - BVC RBF Modelling
3. Branch and Bound
   - Branch and Bound for Efficiency
   - Branch and Bound Aided OLS
4. Recent Extensions
   - New Enhancements

Branch and Bound OLS with *A*-Optimality

- OLS selection based on training MSE and *A*-optimality

$$J^{(n)} = J^{(n-1)} - \frac{1}{K} g_n^2 \mathbf{w}_n^T \mathbf{w}_n + \frac{\beta}{\mathbf{w}_n^T \mathbf{w}_n}$$

β: *A*-optimality weighting, K: the full candidate set size

- nth stage, a candidate from $\mathcal{S}$ is selected as $\mathbf{w}_n$, which has minimum $J^{(n)}$

- **Theorem**. Consider another candidate $\mathbf{p}_j$ in $\mathcal{S}$, let

$$\mathbf{w}^{(-)} = \mathbf{p}_j - \sum_{i=1}^{n-1} \alpha_{i,j}^{(-)} \mathbf{w}_i \text{ with } \alpha_{i,j}^{(-)} = \frac{\mathbf{p}_j^T \mathbf{w}_i}{\mathbf{w}_i^T \mathbf{w}_i}$$

If

$$\left(\mathbf{w}^{(-)}\right)^T \mathbf{w}^{(-)} < \frac{\beta}{J^{(n)}}$$

$\mathbf{p}_j$ can **safely** be removed from $\mathcal{S}$ into $\bar{\mathcal{S}}$

Complexity Saving

- Number of column orthogonalisations and cost function evaluations for conventional OLS forward selection

$$C_{\mathrm{OLS}} = \sum_{n=1}^{M_s} (K - n + 1)$$

- For branch and bound OLS forward selection, this number is

$$C_{\mathrm{BB-OLS}} = \sum_{n=1}^{M_s} (M_n - n + 1)$$

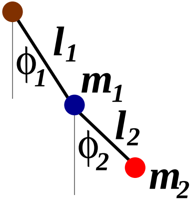
  with $M_{n+1} \le M_n$ and $M_1 = K$

- Empirical results obtained in practice show that typically 20% to 40% saving of computational cost is likely

X. Hong, S. Chen and C.J. Harris, "*A*-optimality orthogonal forward regression algorithm using branch and bound," *IEEE Trans. Neural Networks*, Vol.19, No.11, 1961–1967, 2008

Double Pendulum Results

- Modelling performance for lower pendulum angle ϕ_2
- Integration time span of 200 s at sampling rate of 0.2 s
- First 800 data samples were used in training and last 200 data samples for model testing
- Gaussian RBF variance $\sigma^2 = 3.0$ was set empirically
- Conventional OLS with training MSE and *A*-optimality, and branch and bound aided one



weighting	training MSE		test MSE		model size		BB cost
β	Conv.	BB	Conv.	BB	Conv.	BB	reduction
10^{-11}	0.000127	0.000176	0.000316	0.000515	31	29	**23.02%**
10^{-12}	0.000081	0.000088	0.000196	0.000174	33	35	**20.0%**
10^{-13}	0.000062	0.000078	0.000163	0.000262	42	38	**35.1%**
10^{-14}	0.000046	0.000061	0.000176	0.000162	48	39	**42.8%**

Outline

1. Orthogonal Forward Selection
   - Motivations
   - Previous Enhancements
   - Unified Data Modelling
2. Grey-Box Modelling
   - Incorporating Prior knowledge
   - Symmetric RBF Modelling
   - BVC RBF Modelling
3. Branch and Bound
   - Branch and Bound for Efficiency
   - Branch and Bound Aided OLS
4. Recent Extensions
   - New Enhancements

Elastic-Net OLS

- Elastic net orthogonal forward regression criterion

$$J_{EN}(\mathbf{g}, \lambda_1, \lambda_2) = \epsilon^T \epsilon + \lambda_1 \|\mathbf{g}\|_2 + \lambda_2 \|\mathbf{g}\|_1$$

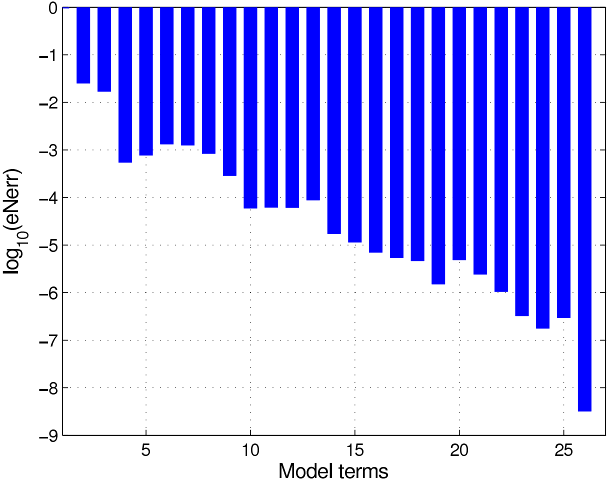
  - Maintain sparsity of LASSO, 1-norm regularisation drives many weights to exactly zero
  - Not as aggressive as LASSO in excluding correlated terms, owing to 2-norm regularisation

- Efficient two level learning
  - At upper level, PSO optimises λ_1 and λ_2 based on LOO MSE values from lower level
  - At lower level, given multiple λ_1 and λ_2 from upper level, perform multiple orthogonal forward selections

- X. Hong and S. Chen, “Automatic kernel regression modeling using elastic net orthogonal forward regression assisted by particle swarm optimization,” submitted to *IEEE Trans. Neural Networks*

Engine Data Set

- Exactly 26 non-zero erro-reduction-ratio (err) terms are selected



- Training MSE: 0.000447, testing MSE: 0.000470

Tunable “Kernel” Modelling

- Tunable “kernel”

$$p_i(k) = \varphi\left(\left(\mathbf{x}(k) - \mathbf{c}_i\right)^T \mathbf{\Sigma}_i^{-1} \left(\mathbf{x}(k) - \mathbf{c}_i\right)\right)$$

  - Centre $\mathbf{c}_i$ and covariance matrix $\mathbf{\Sigma}_i$ are not fixed but parameters to be learnt
- Kernels are optimised by PSO based on LOO criterion one by one in efficient orthogonal forward regression
  - A unified approach for regression, classification and density estimation
- Offer advantages of smaller model size, better generalisation, and less computational complexity in learning, in comparison with “fixed” kernel approach
- S. Chen, X. Hong and C.J. Harris, “Particle swarm optimization aided orthogonal forward regression for unified data modelling,” *IEEE Trans. Evolutionary Computation*, vol.14, no.4, pp.477–499, 2010

Imbalanced Classification

- Highly imbalanced two-class classification problems are widely found in practice
- Construct a Parzen window density estimate based on the positive class training data
- Over-sample the positive class by drawing synthetic samples according to the estimated density
- Apply the PSO aided tunable RBF classifier to the re-balanced data
- M. Gao, X. Hong, S. Chen and C.J Harris, “PDFOS: PDF estimation based over-sampling for imbalanced two class problems,” submitted to *IEEE Trans. Neural Networks*

Conclusions

- The celebrated OLS algorithm has evolved into state-of-the-arts for parsimonious modelling from large data
- Previous enhancements discussed include
  - Local regularisation, optimal experimental design, and leave-one-out cross validation
  - Incorporating prior knowledge naturally for efficient grey-box modelling
  - Implementing branch and bound for further computational efficiency enhancement
- Some very recent extensions have been briefly discussed
- Maintain simplicity and efficiency of original algorithm, which are so appealing to data modelling practitioners