Sparsity-enforced Regression Based on Over-complete Dictionary

Peng Yang, Gongguo Tang and Arye Nehorai
Preston M. Green Department of Electrical and Systems Engineering
Washington University in St. Louis, St. Louis, MO 63130
Email: {yangp, gt2, nehorai}@ese.wustl.edu

***Abstract*—Nonlinear regression has broad applications in various research areas, and kernel-based regression is very popular in machine learning literature. However, the selection of basis-function parameters is often difficult. In this paper we propose a new sparsity-enforced regression method based on an over-complete dictionary. The over-complete dictionary comprises basis functions with quantized parameters, and we employ ℓ_1-regularized minimization to obtain a sparse weight vector of the basis. The ℓ_1-regularized minimization automatically selects the most suitable basis function parameters. Performance analysis shows that this new method provides improved regression accuracy with small model complexity as measured by the number of non-zero entries of the weight vector.**

I. Introduction

Kernel-based regression is one of the most popular nonlinear regression methods. It has broad applications in various areas, including signal processing, biology, ecology, and economics. In these applications, we have inputs and corresponding outputs, and wish to build a model to predict future outputs given future inputs. Well-known kernel regression algorithms include Support Vector Regression (SVR) [1], and Relevance Vector Machine (RVM) [2].

SVR is an extension of the support vector machine (SVM) [3] which was first proposed by Vapnik for classification. The SVM has become a classical and popular method in machine learning literature, and the performance is very satisfactory. However, in support vector regression, the parameters of the kernel function, the "cost" parameter C, and the error tolerance ϵ are often selected by using grid search and cross validation, which is difficult and time consuming. Tipping proposed the RVM (later named as Sparse Bayesian Learning) in [2], which employs the Bayesian framework. With the RVM, there is no need for trade-off parameter selection, and in general fewer supporting vectors are required. Two drawbacks with this method include convergence issues of the EM algorithm and high computational complexity. Also the kernel parameters have to be properly pre-specified.

In this paper, we propose a new sparsity-enforced regression algorithm based on an over-complete dictionary. The basic idea is to build an over-complete dictionary of basis functions, and use ℓ_1-constrained minimization to obtain the weight vector of the basis functions. To avoid the problem of selecting parameters for basis functions, we quantize the parameters in the parameter space, and generate multiple basis functions with the quantized parameters. This is motivated by the success of similar ideas used in sensor arrays [4] and radar [5]. It has been shown that by quantizing the direction of arrival (DOA) angle θ and forming an array manifold matrix $\mathbf{A}(\boldsymbol{\theta})$ based on the quantized angle vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_K]^T$, and then solving the ℓ_1-constrained optimization problem, the DOA estimation performance can be significantly improved . In our proposed approach, we quantize the parameters of the basis functions, and combine them to build an over-complete dictionary. Since the ℓ_1 constrained minimization enforces sparsity [6], only the most relevant basis functions, i.e., the basis functions with the most suitable parameters, are selected. Thus we can obtain a "sparse" combination of basis functions with parameters automatically selected through ℓ_1 minimization. The results on both synthetic and real world datasets show that the algorithm provides improved regression accuracy with a small number of active basis functions.

This paper is organized as follows. In Section II, we introduce the notation, model, and framework of our algorithm. In Sections III and IV, we present how to generate the over-complete dictionary and select the regularization parameter. In Section V, we evaluate the performance of the algorithm. A concluding summary is given in Section VI.

II. Notation, Model, and Framework of the Algorithm

A. Notation and Model

Let $\{\boldsymbol{x}_n \in \mathbb{R}^D\}_{n=1}^N$ denote the input data and $\{y_n \in \mathbb{R}\}_{n=1}^N$ denote the outputs corresponding to the inputs. We assume the inputs and outputs are related by the following model with additive noise:

$$y_n = g(\boldsymbol{x}_n; \boldsymbol{w}) + \epsilon_n, \tag{1}$$

where $\{\epsilon_n\}_{n=1}^N$ are independent noise samples, and $\boldsymbol{w}$ is a vector of model parameters. Our goal is to build the model $g(\boldsymbol{x}_n; \boldsymbol{w})$ based on the input and output data.

Here we use kernel-based regression, namely, the model $g(\boldsymbol{x}_n; \boldsymbol{w})$ is a weighted sum of the basis functions $\{\psi(\boldsymbol{x}; \boldsymbol{\theta}_m)\}_{m=1}^M$:

$$g(\boldsymbol{x}; \boldsymbol{w}) = \sum_{m=1}^{M} w_m \psi(\boldsymbol{x}; \boldsymbol{\theta}_m), \tag{2}$$

This work was supported by National Science Foundation Grants No. CCF-1014908 and CCF-0963742, AFOSR Grant No. FA9550-11-1-0210, and ONR Grant No. N000140810849.

978-1-4577-2105-2/11/$26.00 ©2011IEEE

where $\boldsymbol{\theta}_m \in \mathbb{R}^P$ is the parameter of the mth kernel function. Denote $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{D\times N}$, $\boldsymbol{y} = [y_1, y_2, \cdots, y_N]^T \in \mathbb{R}^N$, $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_N] \in \mathbb{R}^{P\times M}$, and $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \cdots, \epsilon_N]^T \in \mathbb{R}^N$. We then rewrite (1) in a more concise matrix form:

$$\boldsymbol{y} = \boldsymbol{\Psi}(\mathbf{X}; \boldsymbol{\Theta})\boldsymbol{w} + \boldsymbol{\epsilon}, \tag{3}$$

where $\boldsymbol{\psi}(\boldsymbol{x}; \boldsymbol{\theta}_m) = [\psi(\boldsymbol{x}_1; \boldsymbol{\theta}_m), \psi(\boldsymbol{x}_2; \boldsymbol{\theta}_m), \cdots, \psi(\boldsymbol{x}_N; \boldsymbol{\theta}_m)]^T \in \mathbb{R}^N$, and $\boldsymbol{\Psi}(\mathbf{X}, \boldsymbol{\Theta}) = [\boldsymbol{\psi}(\boldsymbol{x}; \boldsymbol{\theta}_1), \boldsymbol{\psi}(\boldsymbol{x}; \boldsymbol{\theta}_2), \cdots, \boldsymbol{\psi}(\boldsymbol{x}; \boldsymbol{\theta}_M)] \in \mathbb{R}^{N\times M}$.

B. Algorithm Framework

Regression based on model (3) involves estimating the parameters $\boldsymbol{\theta}$ of the basis functions and the weight vector $\boldsymbol{w}$ of the basis. In our algorithm, we do not estimate the kernel parameters directly. Instead, we build an over-complete dictionary $\boldsymbol{\Psi}$ consisting of basis functions with the parameters sampled in the parameter space, and use the ℓ_1-regularized optimization to obtain the sparse weight vector. In this paper, we employ the Gaussian Radial Basis as the kernel function

$$\psi(\boldsymbol{x}; \boldsymbol{c}_i, \sigma_j^2) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_i\|_2^2}{\sigma_j^2}\right), \tag{4}$$

where the centers $\boldsymbol{c}_i$ are taken as the input data points, a standard choice according to [1], [2] and [3]. For each center $\boldsymbol{c}_i = \boldsymbol{x}_i$, we generate J different $\sigma_{i_j}^2$'s following the uniform distribution in logarithm scale (i.e., $\log \sigma_{i_j}^2 \sim \mathrm{U}[\log \sigma_{\text{low}}^2, \log \sigma_{\text{high}}^2]$), obtain the basis functions $\boldsymbol{\psi}_{i,i_j} = \boldsymbol{\psi}(\boldsymbol{x}; \boldsymbol{x}_i, \sigma_{i_j}^2)$, and combine these basis functions to construct the over-complete dictionary

$$\boldsymbol{\Psi} = [\underbrace{\boldsymbol{\psi}_{1,1_1}, \cdots, \boldsymbol{\psi}_{1,1_J}}_{\text{common center } \boldsymbol{x}_1}, \cdots, \boldsymbol{\psi}_{i,i_j}, \cdots, \underbrace{\boldsymbol{\psi}_{N,N_1}, \cdots, \boldsymbol{\psi}_{N,N_J}}_{\text{common center } \boldsymbol{x}_N}], \tag{5}$$

which is an $N \times NJ$ matrix. Then we employ model (3) for the regression. The ℓ_1-constrained minimization problem for sparsity enforcement is formulated as

$$\begin{array}{ll} \min\limits_{\boldsymbol{w}} & \|\boldsymbol{w}\|_1 \\ \text{subject to} & \|\boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{w}\|_2^2 \le \beta^2, \end{array} \tag{6}$$

where β^2 specifies the error tolerance. Equation (6) is equivalent to the unconstrained optimization problem

$$\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1, \tag{7}$$

which we will employ in this paper. Problem (7) with the first term multiplied by $\frac{1}{2}$ is often called the LASSO [7].

III. Generation of the Over-complete Dictionary

The quality of the over-complete dictionary plays an important role in the performance of the algorithm. On the one hand, model (3) represents the data accurately only if the dictionary is indeed "over-complete", in that it contains the kernels with proper parameters for the regression. On the other hand, the dictionary $\boldsymbol{\Psi}$ should be as uncorrelated as possible, as implied by the restricted isometry property (RIP) [6] [8]. In our algorithm, we have no prior knowledge about the cardinality of the signal support, but the RIP still provides insights on the selection of the kernel parameters (see the full paper [9]). Since the centers of the Gaussian kernels are taken as the data points, the only parameter we need to select is the spread parameter.

The selection of the spread parameter σ^2 of the Gaussian kernel has been studied by several researchers, e.g. [10],[11]. However, proper selection of σ^2 with low computational complexity is difficulty. In our method, instead of accurately selecting the spread parameter σ^2 of the Gaussian kernel, we need to specify only a proper range $[\sigma_{\text{low}}^2, \sigma_{\text{high}}^2]$ from which to sample the parameter. Since the Gaussian kernel is a "local kernel", we select the spread parameter range based on the distance from the center $\boldsymbol{c}_j$ to its K-nearest neighbors [12], [13]. The lower bound of σ^2 for the center $\boldsymbol{c}_j$ is selected to be

$$\sigma_{\text{low}}^2 = \left[\frac{1}{K}\sum_{k=1}^{K} \|\boldsymbol{c}_j - \tilde{\boldsymbol{x}}_k\|_2\right]^2, \tag{8}$$

where $\{\tilde{\boldsymbol{x}}_k\}_{k=1}^K$ are the K-nearest neighbors of center $\boldsymbol{c}_j$. Since the Gaussian kernel is "local", the spread parameter should not be too large. However, some features of the input may contribute little to the output, in which case a "large" spread parameter is desired to "smooth out" the contributions of such features. A reasonable choice of the upper bound is $\sigma_{\text{high}}^2 = D^p$, where D is the dimension of $\boldsymbol{x}$, and p is a constant. A good empirical choice of p is $p \in [1, 2]$. Note that the input features are normalized to the range $[0, 1]$.

One major drawback of the Gaussian Radial Basis is that it assumes equal contributions from each element of the input vector $\boldsymbol{x}$, which is often not the case. For better performance, we modify the standard Gaussian Radial Basis function as follows:

$$\psi(\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{\sigma}) = \exp\left[(\boldsymbol{x} - \boldsymbol{c})^T \operatorname{diag}(\boldsymbol{\sigma}^2)^{-1}(\boldsymbol{x} - \boldsymbol{c})\right], \tag{9}$$

where $\boldsymbol{\sigma}^2 = [\sigma_1^2, \cdots, \sigma_d^2]^T$. The different weights on different input features result in less regression error and a more "sparse" weight vector. If prior knowledge about the function to be approximated is available, we can choose more appropriate kernel functions other than the Gaussian kernel (see [9]), or even combine multiple types of kernels to obtain better results.

IV. Selection of The Regularization Parameter

After generating the over-complete dictionary $\boldsymbol{\Psi}$, we need to solve the ℓ_1-regularized minimization problem (7), where λ is the regularization parameter. The objective function $\|\boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1$ is a trade-off between the regression error $\|\boldsymbol{y} - \boldsymbol{\Psi}\boldsymbol{w}\|_2^2$ and the complexity of the model, denoted by $\|\boldsymbol{w}\|_1$. Large λ leads to high regression error, while small λ leads to over-fitting. Chen, Donoho and Saunders [14] showed that a good choice for the regularization parameter λ is $\sigma\sqrt{2\log N}$, where σ^2 is the variance of the error, and N is the number of samples. However, since the additive noise here is not necessarily Gaussian, and the error terms consist both additive noise and the modeling error caused by representing the regression function as a linear combination of the basis in the dictionary, it is extremely difficult to estimate the error variance σ^2. In this paper, we discuss two methods for regularization parameter selection: the Bayesian information criterion, and cross validation.

The Bayesian information criterion (BIC) [15][16] is mostly used for model selection. It represents a trade-off between the model complexity and the modeling error. The model with the minimum BIC is selected as the optimal model. The BIC is expressed as

$$\text{BIC} = -2 \cdot \mathcal{L} + (\log N) \cdot d, \tag{10}$$

where $\mathcal{L}$ is the log-likelihood function, N is the sample size, and d is the complexity of the model. In this case the complexity is denoted by the number of basis functions used in the model, i.e., $d = \|\boldsymbol{w}\|_0$. Under the Gaussian model, let σ^2 denote the variance of the Gaussian noise, and then the log-likelihood function is

$$\begin{aligned} \mathcal{L} &= \log\left[\frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp\left(-\frac{\sum_{i=1}^{N}(y_i - g(x_i))^2}{2\sigma^2}\right)\right] \\ &= -\frac{N}{2}\log(\sigma^2) - \frac{N\hat{\sigma}^2}{2\sigma^2} - C, \end{aligned} \tag{11}$$

where $\hat{\sigma}^2 = \frac{\sum(y_i - g(x_i))^2}{N}$ is the mean squared regression error, and C is a constant. Plugging (11) into (10), we obtain that

$$\text{BIC} = N\log(\sigma^2) + N\frac{\hat{\sigma}^2}{\sigma^2} + (\log N) \cdot d. \tag{12}$$

When the noise variance σ^2 is unknown, we use the minimum of the mean squared regression error $\hat{\sigma}^2$ on validation dataset as an estimator for it. The regression error $\hat{\sigma}^2$ consists of both signal noise and modeling error. For $\hat{\sigma}^2_{\min}$, the modeling error is negligible compared to the noise variance, and thus it is a suitable estimator of the noise variance.

Another method for regularization parameter selection we employ is the cross validation (CV). The CV makes no assumption on the noise and is relatively stable. The major issue with CV is its computational load.

V. Experiments and Performance

In this section we evaluate the performance of our algorithm on multiple datasets. CVX Matlab software [17] is employed for ℓ_1-regularized minimization. The detailed settings of experiments are available in the full paper [9].

A. 1-Dimensional Synthetic Function

We first test our algorithm on 1-D synthetic datasets from [2] and [18]. The synthetic functions are listed in Table I and the comparison of results are available in Table II.

The performance of our algorithm on the simple uni-scale functions, e.g. the Sinc datasets, is similar to that of RVM. However, on more complex and multi-scale functions, our algorithm outperform the other methods. Besides, in RVM, the spread parameters of Gaussian kernels have to be properly pre-specified, while in our algorithm, they are automatically chosen by the ℓ_1-regularized minimization. We also compare different methods for the selection of the regularization parameter in Table III.

TABLE I: Description of 1-D Synthetic Datasets

Dataset	Function Expression	Variable x	Noise ϵ
Sinc 1	$y = \text{sinc}(x) + \epsilon$	$[-2\pi, 2\pi]$	$\text{N}(0, 0.01)$
Sinc 2	$y = \text{sinc}(x) + \epsilon$	$[-2\pi, 2\pi]$	$\text{U}(-0.1, 0.1)$
Zigzag	$y = \sin x^2 \cos x^2 - 0.25x + \epsilon$	$[0, 3]$	$\text{N}(0, 0.05)$
Rhythm	$y = \left[\frac{\bmod (x,11) - 5}{8}\right]^3 + \epsilon$	$[0, 20]$	$\text{N}(0, 0.01)$
Polyn	$y = \sum_{i=1}^{5} ix^{i-1} + \epsilon$	$[0, 1]$	$\text{N}(0, 0.1)$

TABLE II: (Root) Mean Squared Error of 1-D Datasets

	SVM		RVM		SER	
Dataset	RMSE	kernels	RMSE	kernels	RMSE	kernels
Sinc 1	0.0378	45.2	**0.0326**	6.7	0.0330	10.8
Sinc 2	0.0215	44.3	**0.0187**	7.0	0.0192	11.2

	Bagging.LMS		RVM		SER	
Dataset	MSE	kernels	MSE	kernels	MSE	kernels
Zigzag	0.0275	N/A	0.0089	7.3	**0.0059**	23.2
Rhythm	0.0022	N/A	0.0025	8.9	**0.0017**	21.4
Polyn	0.0128	N/A	0.0060	11.4	**0.0047**	7.1

TABLE III: Comparison of the BIC, Validation, and Cross Validation for the selection of λ

	BIC		Validation		CV	
Dataset	(R)MSE	kernels	(R)MSE	kernels	(R)MSE	kernels
Sinc 1	0.0348	1**0.2**	0.0376	11.3	**0.0330**	10.8
Sinc 2	0.0196	**10.7**	**0.0191**	11.1	0.0192	11.2
Zigzag	0.0069	**16.5**	0.0075	31.5	**0.0059**	23.2
Rhythm	0.0022	**15.8**	0.0023	33.7	**0.0017**	21.4
Polyn	0.0074	**6.4**	0.0065	6.7	**0.0047**	7.1

B. Multi-resolution Performance

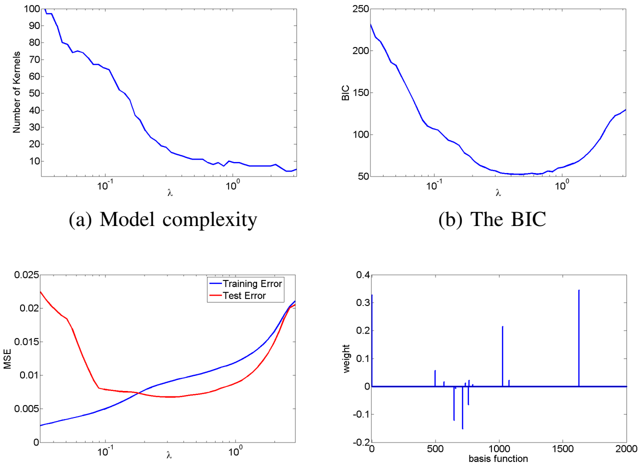
Many regression algorithms employ a uniform kernel width σ^2, which would result in poor regression results when the function to be approximated has multi-scale structure. In our method, since we have multiple quantized kernel widths for each kernel center, the kernel width chosen by the ℓ_1 minimization is actually adaptive to the local property of the function to be approximated. Fig. 2 shows the advantage of our algorithm using an example from [19].

C. Benchmark Comparisons

We finally test our algorithm on synthetic and real world benchmark datasets. The synthetic datasets used are the Friedman #1, #2 and #3 datasets [20], and the real world dataset

TABLE IV: Mean Squared Error on Benchmark Datasets

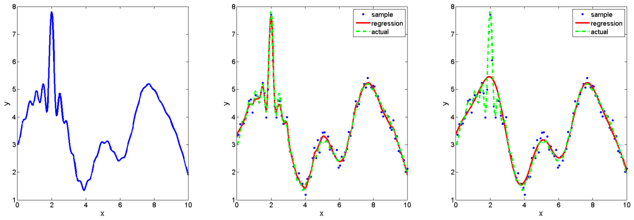
	SVM		RVM		SER	
Dataset	MSE	kernels	MSE	kernels	MSE	kernels
Friedman 1	2.92	116.6	2.80	59.4	**1.30**	103.3
Friedman 2	4140	110.3	3505	6.9	**3012**	21.7
Friedman 3	0.0202	106.5	0.0164	11.5	**0.0103**	50.8
Boston	8.04	142.8	7.46	39.0	**7.19**	173.0



(a) Model complexity (b) The BIC

(c) Mean squared error (d) Sparse weight vector $\boldsymbol{w}$

Fig. 1: Regression on Sinc 1 dataset. (a) shows how the model complexity, measured by the number of active kernels, changes with λ. (b) and (c) present the Bayesian information criterion (BIC), the training error, and the test error as functions of λ. (d) shows the sparse weight vector $\boldsymbol{w}$ at the optimal λ selected by the BIC.



(a) Multi-scale function (b) Adaptive kernel width(c) Uniform kernel width

Fig. 2: An example of a 1-D function with multi-scale structure. The original function is plotted in (a), and the regression results using our algorithm and uniform kernel width are plotted in (b) and (c), respectively.

is the Boston Housing dataset. The input $\boldsymbol{x}$ of Friedman #1 function is 10-dimensional, with 5 inactive input features. The input dimensions of Friedman #2, Friedman #3 and Boston Housing datasets are 4, 4, and 13, respectively. The regression results on these benchmark datasets averaged over 100 repetitions are presented in Table IV. The results indicate that our algorithm performs very well on datasets with multi-dimensional inputs. Also, the sparsity enforcement results in a smaller number of active kernels compared to the SVM on average.

VI. Discussion and Conclusion

In this paper, we proposed and implemented a new sparsity-enforced regression algorithm based on an over-complete dictionary. We built the dictionary by quantizing the parameters of basis functions, and employed ℓ_1-regularized minimization to obtain the weight vector of the basis functions. The sparsity enforcement automatically selects the most suitable basis functions. We discussed details for dictionary generation and the selection of the regularization parameter, and tested our algorithm on multiple synthetic and real world datasets. The advantages of our method include automatic selection of kernel parameters, small model complexity, and improved regression results, especially for functions with multi-scale structure and/or different dependence on input features. Further improvements can be made on methods for regularization parameter selection and algorithms for ℓ_1-regularized minimization. Also we expect to extend this new regression algorithm to classification problems.

References

[1] V. N. Vapnik, S. E. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," in *Advances in Neural Information Processing Systems 9*. MIT Press, 1996, pp. 281–287.

[2] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[3] V. N. Vapnik, *The nature of statistical learning theory*. New York: Springer, 1995.

[4] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Signal Process. Mag.*, vol. 53, pp. 3010–3022, Aug. 2005.

[5] S. Sen, G. Tang, and A. Nehorai, "Multiobjective optimization of ofdm radar waveform for target detection," *IEEE Signal Process. Mag.*, vol. 59, pp. 639–652, Feb. 2011.

[6] E. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, pp. 21–30, Mar. 2008.

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

[8] E. Candès, "Compressive sampling," in *Proc. of the International Comgress of Mathematicians*, Madrid, Spain, 2006.

[9] P. Yang, G. Tang, and A. Nehorai, "Kernel-based nonlinear regression using over-complete dictionary and sparsity enforcement," *Submitted for publication*, 2011.

[10] W. Wang, Z. Xu, W. Lu, and X. Zhang, "Determination of the spread parameter in gaussian kernel for classification and regression," *Neurocomputing*, vol. 55, pp. 643–663, 2003.

[11] J. Yuan, L. Bo, K. Wang, and T. Yu, "Adaptive spherical gaussian kernel in sparse bayesian learning framework," *Expert Systems with Applications*, vol. 36, pp. 3982–3989, 2009.

[12] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, pp. 281–294, 1989.

[13] N. Benooudjit, C. Archambeau, A. Lendasse, J. Lee, and M. Verleysen, "Width optimization of the gaussian kernels in radial basis function networks," in *Euro. Symp. on Artificial Neural Networks*, Belgium, Apr. 2002, pp. 425–432.

[14] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 389–403, 2001.

[15] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*. New York: Springer, 2009.

[16] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[17] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, Apr. 2011.

[18] Y. Wu, C. Wang, and S. C. Ng, "Bagging.LMS: A bagging-based linear fusion with Least-Mean-Square error update for regression," in *TENCON 2006 IEEE Region 10 Conference*, 2006, pp. 1–4.

[19] W.-F. Zhang, D.-Q. Dai, and H. Yan, "Framelet kernels with applications to support vector regression and regularization networks," *IEEE Trans. Syst., Man, Cybern. B*, vol. 40, no. 4, pp. 1128–1144, Aug. 2010.

[20] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1–67, Mar. 1991.