CODEN IEJMAT *Internet Electronic Journal of Molecular Design* **2005**, *4*, 381–392 ISSN 1538–6414
BioChem Press http://www.biochempress.com

Inter*net* Electronic Journal of Molecular Design

June 2005, Volume 4, Number 6, Pages 381–392

Editor: Ovidiu Ivanciuc

Proceedings of the Internet Electronic Conference of Molecular Design 2004
IECMD 2004, November 29 – December 12, 2004

Modeling Structure Property Relationships with Kernel Recursive Least Squares

Rajshekhar, Abhijit Kulkarni, Valadi K. Jayaraman, and Bhaskar D. Kulkarni

Chemical Engineering and Process Development Division, National Chemical Laboratory, Dr. Homi Bhabha Road, Pune–411008, India

Received: October 27, 2004; Revised: March 5, 2005; Accepted: March 21, 2005; Published: June 30, 2005

Citation of the article:
Rajshekhar, A. Kulkarni, V. K. Jayaraman, and B. D. Kulkarni, Modeling Structure Property Relationships with Kernel Recursive Least Squares, *Internet Electron. J. Mol. Des.* **2005**, *4*, 381–392, http://www.biochempress.com.

Inter*net* Electronic Journal of Molecular Design

BioChem Press
http://www.biochempress.com

Modeling Structure Property Relationships with Kernel Recursive Least Squares [#]

Rajshekhar, Abhijit Kulkarni, Valadi K. Jayaraman, and Bhaskar D. Kulkarni *

Chemical Engineering and Process Development Division, National Chemical Laboratory, Dr. Homi Bhabha Road, Pune–411008, India

Received: October 27, 2004; Revised: March 5, 2005; Accepted: March 21, 2005; Published: June 30, 2005

Internet Electron. J. Mol. Des. **2005**, *4* (6), 381–392

Abstract

Motivation. Modeling structure property relationships accurately is a challenging task and newly developed kernel based methods may provide the accuracy for building these relationships.

Method. Kernelized variant of traditional recursive least squares algorithm is used to model two QSPR datasets.

Results. All the datasets showed a good correlation between actual and predicted values of boiling points with root mean squared errors (RMSEs) comparable to other conventional methods. For the datasets from Espinosa *et al*., KRLS showed good prediction statistics with *R* value in the range of 0.97–0.99 and *S* value in the range 5.5–8 as compared to multiple linear regression (MLR) with *R* value in the range 0.85–0.88 and *S* value in the range 22–26. For the dataset from Trinajstić *et al*., KRLS performed consistently well with *R* values lying in the range of 0.95–0.99 and *S* in the range of 5–10 as compared to MLR with *R* values in the range of 0.7–0.85 and *S* in the range of 25–30.

Conclusions. The KRLS method works better when more number of variables from the dataset are included as against other methods such as support vector learning or lazy learning technique which works better for smaller number of reduced relevant variables from the dataset.

Keywords. QSPR; quantitative structure–property relationships; multiple linear regression; kernel recursive least squares; support vector learning; lazy learning.

Abbreviations and notations

ALD, approximate linear dependence	RBF, radial basis function
KRLS, kernel recursive least squares	RMSE, root mean squared error
MLR, multiple linear regression	SVR, support vector regression

1 INTRODUCTION

There is a growing need to model the quantitative structure property relationships (QSPRs) as accurately as possible. As a result extensive efforts to develop new regression algorithms to facilitate modeling these relationships continue. Artificial neural networks were introduced to take into account the inherent non–linearities associated with the QSPR data [1–5]. But owing to certain drawbacks such as long training times, chances of overfitting, getting trapped in local minima while

[#] Presented in part at the Internet Electronic Conference of Molecular Design 2004, IECMD 2004.

* Correspondence author; phone: +91–20–5893095; fax: +91–20–5893041; E–mail: bdk@ems.ncl.res.in.

optimizing the weights etc., their utility becomes restricted and too expensive in terms of development. Recently, kernel based methods are gaining popularity in machine learning community since they provide an alternative to deal with the nonlinearity in the data elegantly and effectively. Some of these popular methods include support vector machines, kernel principal component analysis, kernel density estimation etc.

The idea, previously put forth to deal with nonlinearity in the data, is to map the data into some high dimensional (possibly infinite) feature spaces (usually termed as Hilbert spaces) with a view to make it amenable to linear methodologies. However, the dimensionality increases by many folds with the increase in number of features. Kernel functions were introduced particularly to deal with this problem of high dimensionality. With the advent of many desirable properties of these functions, several of the traditional techniques are now reformulated within this setting to deal with the nonlinearity in the data. In the present work, kernel recursive least squares (KRLS) algorithm is used, which is a kernelized variant of the traditional recursive least squares algorithm [6].

In the sections to follow, materials and methods section explains datasets used and the KRLS algorithm. Results and discussions section describes the results obtained and the pros and cons of the KRLS algorithm. Finally, conclusion section summarizes the salient features of the algorithm.

2 MATERIALS AND METHODS

2.1 Chemical Data

Datasets considered in the analysis are taken from literature. In QSPRs, molecular structural characteristics (geometric and electronic) are generally correlated with physicochemical properties of compounds. The structural characteristics are usually expressed in terms of molecular descriptors. The descriptors, which are routinely used include electronic, *e.g*. dipole moments, lipophilic, *e.g*. partition coefficients and topological, *e.g*. connectivity indices. Some molecular parameters like molar volume, parachor etc. are also used in correlating the physicochemical properties. Most frequently used topological indices proposed for QSPRs include Randić branching indices, valance molecular connectivity indices, Wiener path numbers, Kappa shape indices, and the electrotopological state indices. Many datasets have been published in the literature describing various descriptors correlating with different physicochemical properties. We have resorted to two such datasets, one is from Espinosa *et al.* [7] and the other is from Trinajstić *et al.* [8]. Both the datasets were built up to predict the boiling points of aliphatic hydrocarbons. Espinosa *et al.* [7] particularly dealt with alkanes, alkenes and alkynes family, whereas Trinajstić *et al.* [8] dealt with alkanes only. Espinosa *et al.* [7] obtained QSPRs from four valence molecular connectivity indices, a second–order Kappa shape index (^2k), dipole moment and molecular weight. These were obtained for first 140 alkanes, first 144 alkenes and 43 alkynes respectively. Since good amount of

experimental data is required to get significant correlation, in our analysis, we have dealt with only alkanes and alkenes datasets. Further details regarding the datasets can be found in Espinosa *et al.* [7]. Trinajstić *et al.* [8–10] considered different descriptors based on distance indices and two connectivity indices (total 12 molecular descriptors) for first 150 alkanes. In their work, they studied individual descriptors as well as combination of them to see how they correlate to boiling point. We used this prior knowledge about the data and considered only those descriptors, which correlate well. In our analysis, we have also considered all 12 descriptors together, something which was not done previously. Further details can be found in Trinajstić *et al*. [8]. Next we explain the foundations of the kernel theory in brief.

2.2 Kernel Theory and its connection to Reproducing Kernel Hilbert Spaces

To handle the nonlinear data in real practice, we transform the original input data (lower dimensional) to a potentially very high dimensional (sometimes infinite dimensional) linear feature space (usually termed as Hilbert spaces, $\hbar$), with the inner product defined, which is complete with respect to the corresponding norm. By doing so, one can linearly regress the data in that space [11]. The computations in the high dimensional feature space become intractable as the dimensionality and number of instances in the input space increase. To overcome this practical difficulty, kernel functions are introduced. A kernel is defined as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \bullet \Phi(\mathbf{x}_j) \tag{1}$$

where, $\Phi: R^d \rightarrow \hbar$ i.e. input space is mapped to a higher dimensional Hilbert space by mapping function Φ .

The idea of kernel functions is to perform operations in the input space rather than a very high dimensional feature space. In other words, as can be seen from Eq. (1), an inner product in feature space has an equivalent kernel in the input space. The kernel function can be chosen subject to Mercer's condition [11], which states that there exists a mapping Φ and it's expansion as given in Eq. (1), if and only if, for any g (**x**) we have

$$\int g(x)^2 \, dx \qquad \text{is finite} \tag{2}$$

And:

$$\int K(x, y) g(x) g(y) dx \, dy \geq 0 \tag{3}$$

Kernel functions like polynomial and Gaussian radial basis function have been very popular and extensively used in the literature [11]. Among these, the Gaussian radial basis function (RBF) kernel is very useful and we have employed this kernel in our computations. This kernel can be defined as:

$$K(x_i, x_j) = \exp\left(\frac{\|x_i - x_j\|^2}{-2\sigma^2}\right)$$

where σ is the kernel width parameter.

With the kernel trick, many traditional algorithms, like logistic regression, fisher discriminant analysis etc., have been modified to handle the nonlinearity in the data. KRLS is one such effective technique, which is explained in the next section.

2.3 Kernel Recursive Least Squares (KRLS) Algorithm

We start by giving a brief account of the traditional recursive least squares algorithm and then explain it's kernel variant.

2.3.1 Recursive Least Squares (RLS)

The RLS algorithm is used to recursively train a linear regression model [12], which can be expressed in the following parametric form,

$$f(x) = \langle b, w, \phi(x) \rangle \tag{4}$$

where, $\phi(x)$ is a feature vector associated with the input variable vector **x** and b is the bias term. The weights **w** can be adjusted in a manner such that the bias term becomes zero. In such a case, the regression model reduces to a simpler form,

$$f(x) = \langle w, \phi(x) \rangle = \phi(x)^T w \tag{5}$$

The objective of the learning algorithm is to minimize,

$$g(w) = \sum_{i=1}^{n} (f(x_i) - y_i)^2 \tag{6}$$

with respect to the w vector. In the simple least squares algorithm all the points in the training set are considered simultaneously. However in RLS algorithm each training data point is considered one by one and improving the weight vector in the process.

The optimal weight vector can be expressed as,

$$w = \sum_{i=1}^{n} \alpha_i \phi(x_i) \tag{7}$$

and the regression model becomes,

$$f(x) = \phi(x)^T \phi(x) \alpha \tag{8}$$

2.3.2 Kernel Recursive Least Squares (KRLS)

As can be seen from the Eq. (8), the regression model can be expressed as a function of inner product of feature vectors corresponding to input vectors. The basic idea behind the kernel machine is that if applied on a set of input vectors it can represent the inner product of feature vectors as [6]:

$$f(x) = \sum_{i=1}^{n} K(x_i, x)\alpha_i \tag{9}$$

By minimizing the cost function, we can get the theoretical value of α (same as in case of least squares). However, *the matrix inversion method* allows us to compute the α vector recursively (same as in case of RLS) i.e. a stream of training data points can be sampled sequentially.

This approach will however lead to some severe problems if size of training set increases to some considerable extent. The problems arise due to overfitting, memory limitations (storage of kernel matrix) and the instability in inverting the large kernel matrix (as will be required in the α vector calculation).

To avoid these shortcomings, sparsification method for the pruning of training data is necessary. By making use of this method the training data can be stored in a compact form i.e. only a fraction of training data will be actually used for the training purpose. As explained in the paper by [6], the sparsification procedure can be justified as follows. Although the dimensions of the feature space can be very large but the effective dimensionality of the manifold spanned by the training feature vectors may be significantly low. Hence, the solution to any optimization problem can be expressed by a set of linearly independent feature vectors that approximately span this manifold, as long as it satisfies the conditions required by the theorem. The sparsification procedure will be discussed in details in the next section.

2.3.3 Sparsification Method

In the sparsification procedure, the linearly independent training data points will be stored in a Dictionary Set [6]. Assume that after sampling (i–1) number of data points, the dictionary set is D_{i-1}. Now consider next training data point, this data point will be added in the dictionary set if it is approximately linearly independent of the dictionary set vectors. To prove the linear dependency of the new data vector on the dictionary vectors the approximate linear dependence (ALD) test is performed, *i.e*. we will find a coefficient vector C such that,

$$\delta_i = \min_C \left\| \sum_{j=1}^{m-1} C_j \phi(x_j) - \phi(x_i) \right\|^2 \leq \nu \tag{10}$$

where ν is an important tuning parameter that determines the level of sparsity. Expanding Eq. (10),

$$\delta_i = \min_C \left\{ \sum_{j,l=1}^{m-1} C_j C_l \langle \phi(x_j), \phi(x_l) \rangle - 2\sum_{j=1}^{m-1} C_j \langle \phi(x_j), \phi(x_i) \rangle + \langle \phi(x_i), \phi(x_i) \rangle \right\} \tag{11}$$

we can express the cost function for the ALD test in terms of the inner product of the feature vectors. We can apply the kernel trick once again to get the cost function as:

$$\delta_i = \min_C \left\{ C^T K_{i-1} C - 2C^T K_{i-1}(x_i) + K_{ii} \right\} \tag{12}$$

where, $\left[K_{i-1}\right]_{j,l} = k(x_j, x_l)$, $(K_{i-1}(x_i))_j = k(x_j, x_i)$ and $K_{ii} = k(x_i, x_i)$.

Solving for optimality of the cost function, we can get:

$$C_i = K_{i-1}^{-1} K_{i-1}(x_i) \text{ and } \delta_i = K_{ii} - K_{i-1}(x_i)^T C_i \leq \nu \tag{13}$$

If $\delta_i \leq \nu$, we don't need to include the data point under consideration into the dictionary set. But if $\delta_i > \nu$, then we should include it into the dictionary set.

2.3.4 Computation of α vector

While updating the α vector online (*i.e*. recursively), we are faced with two situations:

(*a*) The new training data point will be added in the dictionary set and

(*b*) The new training data point will not be added in the dictionary set.

Due to sparsification, the cost function (for regression model) reduces to the form,

$$g(\alpha) = \left\| \phi_i^T \phi_i \alpha - y_i \right\|^2 = \left\| A_i K_i \alpha - y_i \right\|^2 \tag{14}$$

Where α is approximated to be equal to $A_i \alpha$. A_i is introduced to counter the ill effects of sparsification. i.e. $A_i \alpha$ is a vector of m (size of the dictionary set) "reduced" coefficients.

For the two cases, the optimal value of α vector can be computed recursively using the *matrix inversion method*.

Case (*a*)

$$K_i = \begin{bmatrix} K_{i-1} & K_{i-1}(X_i) \\ K_{i-1}(X_i)^T & K_{ii} \end{bmatrix} \tag{15}$$

$$K_i^{-1} = \frac{1}{\delta_i} \begin{bmatrix} \delta_i K_{i-1}^{-1} + C_i C_i^T & -C_i \\ -C_i^T & 1 \end{bmatrix} \tag{16}$$

$$\alpha_i = K_i^{-1} (A_i^T A_i)^{-1} A_i^T y_i = \begin{bmatrix} \alpha_{i-1} - \frac{C_i}{\delta_i} (y_i - K_{i-1}(x_i)^T \alpha_{i-1}) \\ \frac{1}{\delta_i} (y_i - K_{i-1}(x_i)^T \alpha_{i-1}) \end{bmatrix} \tag{17}$$

As required in case (*b*), we have to update the matrix P also as,

$$P_i = \begin{bmatrix} P_{i-1} & 0 \\ 0^T & 1 \end{bmatrix} \tag{18}$$

Case (*b*). Here there will not change the kernel matrix (as the dictionary set remains same).

$$\alpha_i = K_i^{-1} P_i A_i^T y_i = \alpha_{i-1} + K_i^{-1} q_i (y_i - K_{i-1}(x_i)^T \alpha_{i-1} \tag{19}$$

where $P_i = (A_{i-1}^T A_{i-1})^{-1} = P_{i-1} - \frac{P_{i-1} C_i C_i^T P_{i-1}}{1 + C_i^T P_{i-1} C_i}$ and:

$$q_i = \frac{P_{i-1} C_i}{1 + C_i^T P_{i-1} C_i} \tag{20}$$

2.4 KRLS Algorithm (with sparsification)

(1) Initialize the Dictionary Set.
(2) Select the first element from the training set and put into the dictionary set. Compute the kernel weight vector (alpha) for the dictionary set.
(3) Get the next sample from the training set and perform the approximate linear dependence (ALD) test for it.
(4) If ALD test error is less than the threshold value ν, go to step (6).
(5) Add the new sample in the dictionary set. Update the kernel weight vector. Go to step (7).
(6) Keep the dictionary set unchanged. Update the kernel weight vector.
(7) If training set has any element left, go to step (3).
(8) Use the dictionary set and kernel weight vector in the testing phase.

Having described the datasets and the algorithm, we now move on to explain the results obtained.

3 RESULTS AND DISCUSSION

The results obtained are summarized in Tables 1–4. The kernel methods depend on the selection of properly optimized kernels [13–15]. We tried several kernel functions including linear, quadratic and Gaussian form of RBF kernel. Comparison of RBF kernel with linear kernel in terms of prediction RMSEs on all the datasets is shown in Table 1. We found that RBF kernel performed consistently well and hence selected and used it in all the simulations. There are two parameters (ν and σ), which need proper attention to get good performance of the KRLS algorithm. The parameter, ν, controls the size of the dictionary set and was set as 10^{-5}. The kernel width parameter was optimized in a range of [0.1, 20] with 0.1 as step size *i.e.* while training the algorithm, width parameter goes on changing from iteration to iteration while ν remains constant. The kernel width value with least root mean squared error (RMSE) is treated as an optimal value for the particular

range specified above. This optimal value is used in predictions.

Table 1. Prediction RMSEs for linear and RBF kernel

Dataset	Linear Kernel	RBF Kernel
Alkane (Espinosa *et al.* [7])	5.23	2.39
Alkene (Espinosa *et al.* [7])	11.51	8.49
Alkane (Trinajstić *et al.* [8])–12 descriptors	3.43	2.94
Alkane– 2–TI descriptor only	21.61	13.81
Alkane– 3–TI descriptor only	22.68	21.31
Alkane– connectivity index only	10.03	5.68

Table 2. Results on the dataset due to Espinosa *et al.* [7]

QSPR Dataset	Optimal Kernel parameter width	Dictionary set size	RMSE
Alkanes	3.1	31	**2.39**
Alkenes	3.7	32	**8.49**

Table 3. Results on the Alkanes dataset due to Trinajstić *et al.* [8]

Descriptor considered	Optimal Kernel parameter width	Dictionary set size	RMSE KRLS^{+}	RMSE Lazy learning	RMSE SVR^{+}
2–TI*	0.4	32	13.81	**1.88**	2.13
3–TI*	0.2	22	21.31	**0.70**	0.93
Connectivity index	0.4	8	5.68	**0.60**	0.73
All 12 descriptors	4.7	26	**2.94**	17.08	17.71

*2–TI: 2–dimensional topological index, 3–TI: 3–dimensional topological index
$^{+}$ SVR: Support vector regressor, KRLS: Kernel recursive least squares

Standard machine learning steps were adopted while making the partitions in the data as training set (approximately 2/3rd of the original data) and test set (remaining 1/3rd) for predictions. The figures below show the prediction plots (cf. Figure 1–6) of KRLS algorithm wherein actual boiling point is plotted against predicted boiling point.

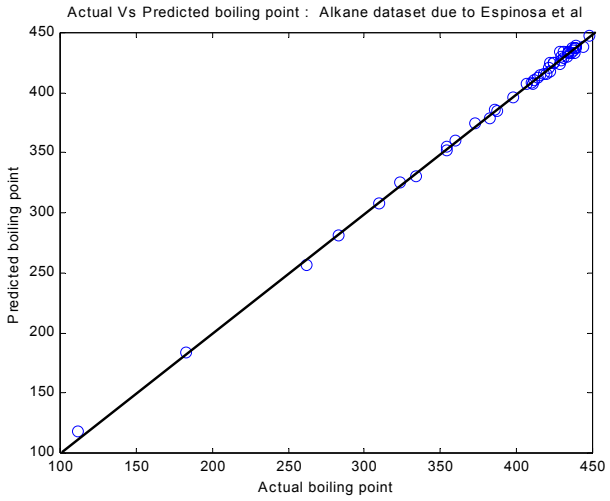

Figure 1. Experimental *vs*. predicted boiling point: alkanes dataset from Espinosa *et al.* [7]

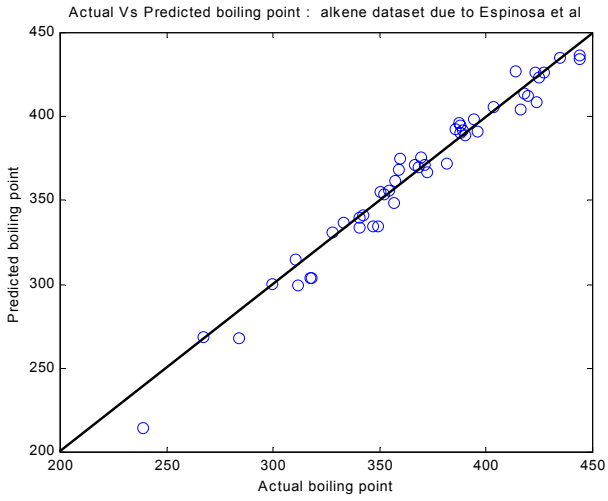

Figure 2. Actual *vs* predicted boiling point: alkenes dataset from Espinosa *et al.* [7]

Firstly, the datasets due to Espinosa *et al.* [7] were analyzed. All the descriptors in the original data were considered while building the model. As can be seen from Table 2, descriptors in both the

datasets, viz. alkanes and alkenes, correlate well with prediction RMSEs 2.39 and 8.49 respectively. *R* and *S* statistics (cf. Table 4) eventually show that KRLS performed better than multiple linear regression (MLR) and equivalently to support vector regression and lazy learning. Direct comparison with previous work due to Espinosa *et al.* [7] is not possible due to different performance parameters and simulation setups. However, the prediction plots ultimately indicate that KRLS performed well.

Table 4. Comparison of methods based on prediction statistics (R and S)

Dataset	R				S			
	MLR [a]	**KRLS**	SVR	LL	MLR	**KRLS**	SVR	LL
Alkane (Espinosa *et al.* [7])	0.88	**0.99**	0.98	0.99	22.15	**5.93**	7.85	5.93
Alkene (Espinosa *et al.* [7])	0.85	**0.97**	0.96	0.97	25.2	**7.15**	8.2	7.15
Alkane (Trinajstić *et al.* [8]) 12 descriptors	0.7	**0.99**	0.99	0.99	30.2	**5.9**	5.9	5.9
Alkane (2–TI descriptor)	0.75	**0.95**	0.95	0.95	29.1	**9.17**	9.17	9.17
Alkane (3–TI descriptor)	0.7	**0.98**	0.95	0.98	30.2	**6.23**	9.17	6.23
Alkane (connectivity index)	0.84	**0.95**	0.94	0.95	26.5	**9.18**	9.5	9.18

[a] MLR: Multiple linear regression, KRLS: Kernel recursive least squares, SVR: Support vector regression, LL: Lazy learning

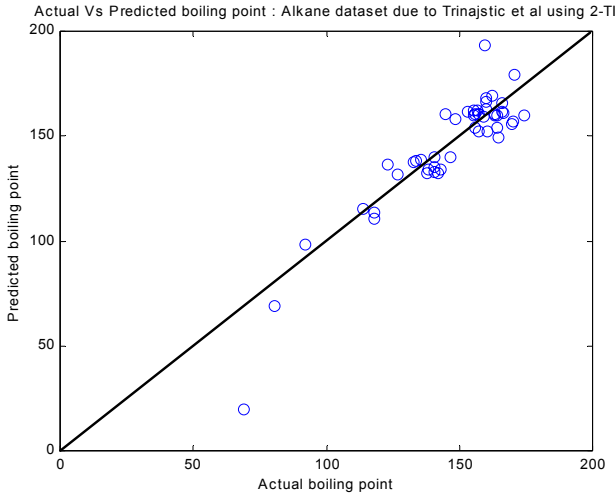

Figure 3. Actual *vs*. predicted boiling point: alkane dataset from Trinajstić *et al.* [8] using 2–TI

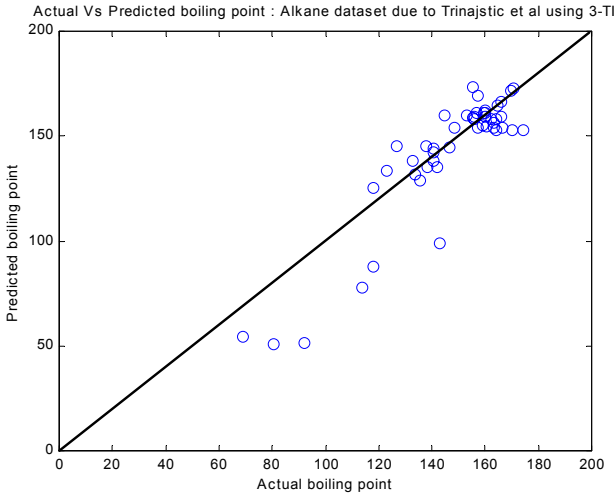

Figure 4. Actual *vs*. predicted boiling point: alkane dataset from Trinajstić *et al.* [8] using 3–TI

Next, the alkanes data reported by Trinajstić *et al.* [8–10] was analyzed. We initially considered only three descriptors *viz*. 2–dimensional topological index, 3–dimensional topological index and connectivity index. The earlier work reported that these descriptors correlate well in predicting the boiling points. This observation was also justified in the work reported by Kumar *et al.* [16]. We took advantage of this knowledge and modeled the QSPR with these individual descriptors. Overall, it is found that KRLS is correlating well in predicting the boiling points. Among these three descriptors, 3–dimensional topological index is found to be correlating well. This is consistent with the observations made by Kumar *et al*. [16]. The prediction *R* values for all the three descriptors lie in the range of 0.95–0.99 and *S* values lie in the range of 6–10. The prediction plots for individual descriptors are as shown in Figures 3–5.

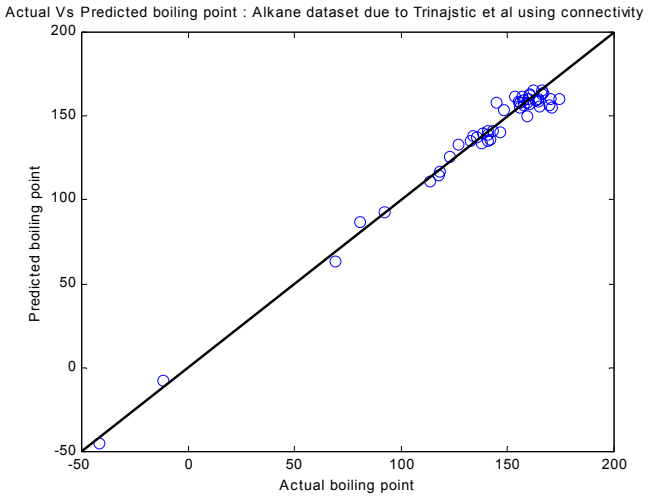

Figure 5. Actual *vs*. predicted boiling point: alkane dataset from Trinajstić *et al.* [8] using connectivity index.

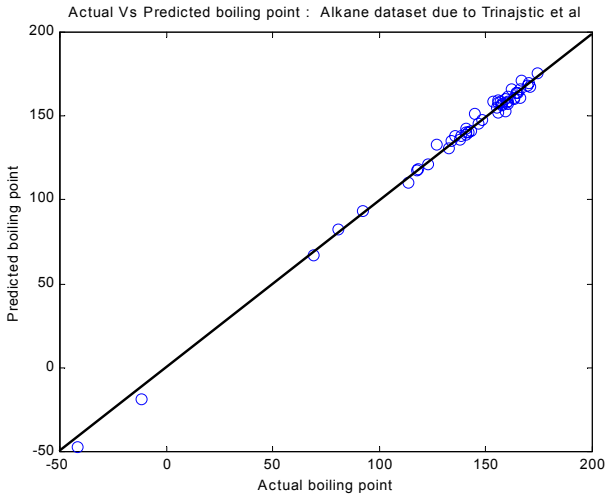

Figure 6. Actual *vs*. predicted boiling point: alkane dataset from Trinajstić *et al.* [8] using 12 descriptors.

If we compare the descriptorwise results with that reported by Kumar *et al.* [16], it is observed that lazy learning regressor and support vector regressor outperforms KRLS with RMSE values as shown in Table 3. One possible reason is that lazy learning is a powerful local learner and support vector regressor, which is a kernel–enabled method, is based on strong foundations of statistical learning theory and structural risk minimization principle which gives it a natural advantage to generalize well. But if we consider the relative training time to train all these algorithms, KRLS requires less time since in support vector regression and lazy learning, optimization of algorithm parameters (model selection) take much time. However, the errors by SVR and lazy learning algorithms are less than experimental error. This may be attributed to the fact that these methods are quite powerful and sometimes they may overfit. To avoid this problem, lot of simulations on unseen observations are required to guarantee their generalization. But they have advantage that they can be easily implemented online also.

Further in our analysis, we combined all the 12 descriptors reported in the original dataset to build the model. Here KRLS outperformed lazy learning as well as support vector regressor (cf. Table 3 and [16]). The prediction plot is as shown in Figure 6. Prediction statistics (R and S) are as shown in the Table 4. The results imply that KRLS performs better with the more number of variables in the dataset whereas support vector regressor and lazy learning regressor works better with the relevant variables, which contain maximum information for good correlation. Since KRLS algorithm takes help of only dictionary set while predicting the unseen cases, the increased number of computations due to increased number of variables can be compensated with the less computations with the less number of observations in the dictionary set.

To summarize the results, KRLS performed well on all the investigated datasets. The attractive

features of the algorithm include less training time, less computations while predicting the unseen cases (cf. dictionary set size in Tables 2 and 3). With these desirable features, algorithm can be further exploited to tackle more difficult real world problems.

4 CONCLUSIONS

The real world QSPR datasets often have large degree of nonlinearity associated with them that make it difficult to develop accurate correlations. Kernel recursive least squares, a variant of traditional recursive least squares algorithm, is used in the present work to model two different QSPR datasets for predicting the boiling points of aliphatic hydrocarbons, namely alkanes and alkenes. Kernel functions facilitate to deal with the nonlinearity in data effectively by allowing one to work in input space instead of very high dimensional feature space. The algorithm gives a good correlation between predicted and actual values of boiling points in case of both the datasets. For the datasets due to Espinosa *et al.* [7], KRLS showed good prediction statistics with *R* value in the range of 0.97–0.99 and *S* value in the range 5.5–8 as compared to multiple linear regression (MLR) with *R* value in the range 0.85–0.88 and *S* value in the range 22–26. For the dataset due to Trinajstić *et al.* [8], KRLS performed consistently well with *R* values lying in the range of 0.95–0.99 and *S* in the range of 5–10 as compared to MLR with *R* values in the range of 0.7–0.85 and *S* in the range of 25–30. Small dictionary set size indicates reduced number of computations in prediction of the unseen observations. With the desirable properties like less training time, reduced computations due to small dictionary set size, the algorithm may be quite useful to model other kind of structural relationships like structure activity relationships or structure mobility relationships.

Acknowledgment

Financial assistance from Department of Science and Technology (DST), New Delhi is gratefully acknowledged. Abhijit is grateful to the Council of Scientific and Industrial Research (CSIR), New Delhi for research fellowship.

5 REFERENCES

[1] M. Wagener, J. Sadowski and J. Gasteiger, Autocorrelation of Molecular Surface Properties for Modeling Corticosteroid Binding Globulin and Cytosolic Ah Receptor activity by Neural Networks, *J. Am. Chem. Soc.* **1995**, *117*, 7769–7775.

[2] J. Koziol, Neural Network Modeling of Melting Temperatures for Sulfur–Containing Organic Compounds, *Internet Electron. J. Mol. Des.* **2002**, *1*, 80–93, http://www.biochempress.com.

[3] E. S. Goll and P. C. Jurs, Prediction of the Normal Boiling Points of Organic Compounds from Molecular Structures with a Computational Neural Network Model, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 974–983.

[4] R. C. Schweitzeri and J. B. Morris, The Development of a Quantitative Structure Property Relationship (QSPR) for the Prediction of Dielectric Constant Using Neural Networks, *Anal. Chim. Acta.* **1999**, *384*, 285–303.

[5] O. Ivanciuc, T. Ivanciuc, and A. T. Balaban, Quantitative Structure–Property Relationships for the Normal Boiling Temperatures of Acyclic Carbonyl Compounds, *Internet Electron. J. Mol. Des.* **2002**, *1*, 252–268, http://www.biochempress.com.

[6] Y. Engel, S. Mannor, R. Meir, The Kernel Recursive Least Squares Algorithm, *IEEE Trans. Sig. Proc*. **2004**, *52* (8), 2275–2285.
[7] G. Espinosa, D. Yaffe, Y. Cohen, A. Arenas, F. Giralt, Neural Network Based Quantitative Structural Property Relations (QSPRs) for Predicting Boiling Points of Aliphatic Hydrocarbons*, J. Chem. Inf. Comput. Sci*. **2000**, *40*, 859–879.
[8] Z. Mihalić, S. Nikolić, N. Trinajstić, Comparative study of molecular descriptors derived from the distance matrix, *J. Chem. Inf. Comput. Sci*. **1992**, *32*, 28–37.
[9] B. Lučić, D. Juretić, S. Nikolić, N. Trinajstić, The Structure–Property Models Can Be Improved Using the Orthogonalized Descriptors, *J. Chem. Inf. Comput. Sci*. **1995**, *35*, 532–538.
[10] M. Randić, N. Trinajstić, Comparative structure–property studies: The connectivity basis, *J. Mol. Struct*. (*Theochem*) **1993**, *284*, 209–221.
[11] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.
[12] R.H. Myers, *Classical and Modern Regression with Applications*, 2nd Edn., Boston, MA: PWS–KENT Publishing company, 1994.
[13] R. Burbidge, M. Trotter, B. Buxton, and S. Holden, Drug design by machine learning: support vector machines for pharmaceutical data analysis, *Comput. Che*m. **2001**, *2*6, 5–14.
[14] O. Ivanciuc, Support Vector Machine Classification of the Carcinogenic Activity of Polycyclic Aromatic Hydrocarbons, *Internet Electro*n. J. *Mo*l. *De*s. **2002**, *1*, 203–218, http://www.biochempress.com.
[15] O. Ivanciuc, Support Vector Machine Identification of the Aquatic Toxicity Mechanism of Organic Compounds, *Internet Electro*n*. J. Mol. Des*. **2002**, 1, 157–172, http://www.biochempress.com.
[16] R. Kumar, Abhijit Kulkarni, Valadi K. Jayaraman, Bhaskar D. Kulkarni, Structure–Activity Relationships using Locally Linear Embedding Assisted by Support Vector and Lazy Learning Regressors, *Internet Electron*. *J. Mol. Des*. **2004**, *3*, 118–133, http://www.biochempress.com.

Biographies

Rajshekhar is a graduate student in Chemical Engineering Department at Indian Institute of technology Kharagpur, Kharagpur.

Abhijit Kulkarni is a senior research fellow in the chemical engineering division of National Chemical Laboratory, Pune, India. His research interests include machine–learning applications in process engineering and process modeling, simulation and optimization. He obtained his bachelor's degree from University of Pune and Master's degree from Birla Institute of Technology and Science, Pilani. Presently he is doing PhD at University of Pune.

Valadi K. Jayaraman is a senior scientist in the chemical engineering division of the National Chemical Laboratory, Pune, India (jayaram@che.ncl.res.in). His interests include chemical and bio–reaction engineering, applications of artificial–intelligence tools in engineering, process modeling, optimization and control. Jayaraman has been visiting faculty to Indian universities and has taught many core chemical engineering courses to graduate students. He obtained his bachelor's and master's degrees in chemical engineering from the Univ. of Madras and his Ph.D. while working at National Chemical Laboratory. He has over 50 international publications. He has recently received the Herdillia award from the Indian Institute of Chemical Engineers (IIChE) for excellence in basic research.

Bhaskar D. Kulkarni is a senior scientist and heads the chemical engineering division of the National Chemical Laboratory (NCL), Pune, India. He has been with NCL for over 25 years. His interests are stochastic processes, non–linear systems, chemical reaction engineering, applications of artificial–intelligence tools in engineering, process modeling, optimization and control. A fellow of the Indian National Science Academy, National Academy of Sciences, National Academy of Engineering, and Third World Academy of Sciences, he has received numerous awards for his work. He has published three books and over 200 technical papers in prestigious international journals. Kulkarni obtained his bachelor's and master's degrees in chemical engineering from Laxminarayan Institute of Technology in Nagpur, India, and received his Ph.D. degree while working at NCL.