Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2007)

Real Value Solvent Accessibility Prediction using Adaptive Support Vector Regression

Jayavardhana Gubbi, Alistair Shilton and Marimuthu Palaniswami
Department of Electrical and Electronic Engineering
The University of Melbourne
Vic - 3010, Australia.
Email: jrgl,apsh,swami@ee.unimelb.edu.au

Michael Parker
St. Vincent's Institute of Medical Research,
9, Princes Street, Fitzroy,
Vic - 3065, Australia.
Email: mparker@svi.edu.au

Abstract— **Knowledge of the secondary structure and solvent accessibility of a protein plays a vital role in prediction of fold, and eventually the tertiary structure of the protein. This paper deals with prediction of relative solvent accessibility, given only the amino-acid sequence. In this paper, we use an improved support vector regression (SVR) and new kernels for real valued prediction of solvent accessibility. In this regard, two main issues are addressed. First we address the problem of ϵ selection, which we found to be somewhat problematic in our earlier work (ϵ is a parameter with significant influence on noise insensitivity and generalization of SVRs). In particular, rather than employ the standard trial and error based approach, we used an improved tube shrinking method to find ϵ. Secondly, a novel kernel combining solvation model, electrostatic charge model and evolutionary information in the form of position specific scoring matrix (PSSM) is given. A new dataset of 472 proteins with less than 20% sequence identity is curated and used to evaluate the result. To make a more objective comparison with earlier methods, we use a standard dataset and show that the proposed scheme is better than the ones normally used in literature. We also report a lowest mean absolute error (MAE) so far of 0.12 on the standard dataset.**

I. Introduction

Knowledge of the secondary structure and solvent accessibility of a protein plays a vital role in predicting the tertiary structure of the protein. The protein folding problem can be defined as prediction of the complete three dimensional structure of a protein given only the amino-acid sequence. The folding free energy can be expressed as the summation of free energies due to intra molecular interaction and the interaction with the surrounding solvent molecules [24]. The problem of predicting interaction with surrounding solvent molecule has been shown to be more challenging. This essentially is to predict accessible surface area of a given residue in the protein. Most solvation models assume that the solvation energy of the solute is the sum of individual solvation energies of the residues. Hence it is important to know the solvation energy of the residues in a given environment. Moreover, this would also give an indication about the position of the residue with respect to the core of protein which will enable the calculation of accessible surface area of a residue [11], [24].

Unfortunately, calculating the solvation energy or accessible surface area of the residue is a non-trivial task. Relative Solvent Accessibility (RSA) helps us to express the accessible surface area in relative terms. Most of the early attempts at RSA prediction concentrated on predicting whether it is buried or exposed to solvents. These methods employ a binary classifier to predict whether the solvent is exposed or buried based on threshold of RSA, eg. 9% or 16%. Second generation included three states viz., buried, intermediate and exposed.

Polastri *et. al.* [26] use bidirectional recurrent neural network for protein solvent accessibility prediction. Yuan *et. al.* [36] use support vector machines (SVM) for two and three state RSA prediction reporting accuracies in the range of 70-73%. NETASA [1] was developed to predict the net accessible surface area and report results of about 71% on Manesh database. It uses a simple neural network architecture similar to PHD [27] and JPRED [5]. Kim and Park [16] use support vector machines (SVM) and 3D local descriptors. They call their system PSIsvm. They use PSSM matrix from PSI-BLAST and 3D local descriptors comprising of disulphide bridges, hydrophobic interactions and remote hydrogen bonds as features. They report accuracies of around 78-80% for two stage solvent accessibility classification for 16% and 25% buried state. In 2004, Nguyen and Rajapakse [22], [23] propose a two stage SVM approach which takes into account contextual relationships in the neighborhood. They report accuracies of over 90% using Manesh dataset. Sim *et. al.* [32] report slight improvement in predictions using a fuzzy k-nearest neighbor method. Recently, real value prediction has been developed to predict solvent accessibility particularly based on regression methods. RVP-net [2] was developed which predicts real valued solvent accessibility. In their work, they show the importance of real valued calculations over two stage predictions. Feed-forward neural network with multi-layer function mapping is employed. The network is trained with 80,000 residues. Gianese *et. al.* [7] use probability profiles of amino acids to predict RSA. Garg et. al. [6] use evolutionary information and feed-forward neural network and report improvement in accuracies by about a percent. Support Vector Regression has been quite popular in all other work which followed. Yuan and Bailey [35] demonstrate the application of regression approaches in predicting accessible surface area. They predict accessible surface area in squared angstroms rather than RSA. Yuan and Huang [37] use support vector regression and report the best possible mean absolute

1-4244-0710-9/07/$20.00 ©2007 IEEE

error of 17% on a database they create. Wagner *et. al.* [34] compare neural network and support vector regression for real valued RSA prediction. On a new dataset they report decrease in error rate by at least three percent compared to earlier proposed methods.

In this work, we employ a new variant of support vector regressor for real valued RSA prediction. A major difficulty with earlier methods employing support vector regressor is choosing ϵ which is usually done by trial and error (Although this fact is often skimmed over). The method described in this paper uses an improved tube shrinking method developed by Shilton *et. al.* [31] for automatically calculating ϵ. In [31], the co-authors have given the theoretical foundation of adaptive SVM and for the first time we use them to demonstrate its usefulness in application scenario. We also propose a novel kernel combining solvation model, electrostatic charge model and evolutionary information in the form of position specific scoring matrix (PSSM) and compare it with standard kernels. A new dataset is curated from recent version of CATH to validate the proposed technique. This dataset is "harder" than the ones used earlier as the maximum pairwise sequence identity is less than 20%. We use Manesh dataset [17] to compare our method objectively with previously proposed techniques. This dataset contains 215 non-homologous proteins with sequence identity less than 25%. 30 sequences are used for training and 185 proteins are used for testing. The proteins used for training include 1aba, 1abr, 1bdo, 1beo, 1bib, 1bmf, 1bnc, 1btm, 1btn, 1cem, 1ceo, 1cew, 1cfy, 1chd, 1chk, 1cyx, 1dea, 1del, 1dkz, 1dos, 1fua, 1gai, 1gpl, 1gsa, 1gtm, 1hav, 2i1b, 2sns, 3grs, 3mdd. The same set is used by several authors in literature [23], [22], [2], [1], [6] for comparison. Overall, we try to show that the combination of the features used and the kernel proposed performs better than the existing techniques in literature. The paper is organized as follows: Section II explains the dataset used, the features extracted and the evaluation methods. Section III introduces ϵ support vector regression followed by ν support vector regression and the *modified* ν support vector regression. Kernels are discussed in section IV. Results and discussions are presented in section V. Conclusions are given in section VI

II. Material and Methods

We construct the dataset from CATH version 2.6.0 released in April 2005 [25]. At the first stage, we select proteins with sequence length greater than 40 and with resolution of at least 2 Å. UniqueProt [21] with HSSP-value of 0 was used to eliminate identical sequences. After doing this, we are left with 472 proteins out of 10,000+ proteins with pairwise sequence identity less than 20% (PSA472 dataset (available on http://www.ee.unimelb.edu.au/ISSNIP/bioinf)). We get the secondary structure definitions from DSSP [14] algorithm. The 8 to 3 state reduction method used was H, G and I to H, E and B to E and all others to C where H stands for α Helix, E for β Strand and C for Coil. The solvent accessibility values extracted from DSSP program have been used. Relative Solvent Accessibility is defined as the ratio of solvent exposed surface area observed in the given protein (SA) to the maximum achievable solvent exposed surface area for that particular amino-acid (MSA):

$$RSA = \frac{SA}{MSA} \tag{1}$$

In a recent work, it was shown that the Empirical Atomic Solvation model [11] is the most effective out of the five implicit solvation models tested. This makes use of the atomic solvation parameters from Ooi *et. al*. [24]. Hence we make use of the free energy of hydration parameter from Ooi *et. al*. [24] as our first feature (denoted $\mathbf{x}_h$). The values reflect the contribution of each side chain to the thermodynamic parameters of hydration which give an indication of hydrophobicity and hydrophilicity.

Based on our earlier experiments, we make use of Grantham Polarity [8] (scale was obtained from http://au.expasy.org/tools/protscale.html) ($\mathbf{x}_c$) scale as the input for the new kernels we propose to use. We also extract probability of occurrence of amino acids in different secondary structure states (α, β and *C*) using Chou-Fasman method. The Chou-Fasman parameter for Helix(α) is given by $P_{\alpha i} = f_{\alpha i}/\langle f_\alpha \rangle$ where $\langle f_\alpha \rangle$=Number of Residues in Helix/Total Number of Residues and 'i' ranges over the set of amino-acid residues. Similar conformational parameters for strand $P_{\beta i}$ and coil $P_{\gamma i}$ are calculated ($\mathbf{x}_s$). We then extracted evolutionary information in the form of position specific scoring matrix (PSSM) generated by PSI-BLAST [3] using the non-redundant (NR) database. The low complexity regions, coiled-coil regions and transmembrane helices were filtered with *pfilt* [13]. We choose an E-value of 0.0001 and 10 iterations for PSI-BLAST. The BLOSUM62 matrix was used for multiple sequence alignment. We used the following function to scale the profile values from the range (-7,7) to the range (0,1) [15], [12](which is better suited for Support Vector Regression usage).

$$\hat{g}(x) = \begin{cases} 0.0 & x \leq -5 \\ 0.5 + 0.1x & -5 < x < 5 \\ 1.0 & x \geq 5 \end{cases} \tag{2}$$

where x is the value of the PSSM matrix. This results in a set of 25 features for every amino-acid in the newly created dataset. Instead of considering only one amino-acid, we use a window of length L around the residue to capture the local information. Another feature is added to every amino-acid to indicate whether it is at the edge of the protein sequence or in the middle. The final input to the support vector regressor is of length $25L + 1$.

For the evaluation of the proposed method, we use standard Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) values, namely:

$$\begin{aligned} \text{RMSE} &= \sqrt{\tfrac{1}{N}\sum_i (y_i - \hat{y}_i)^2} \\ \text{MAE} &= \tfrac{1}{N}\sum_i |y_i - \hat{y}_i| \end{aligned} \tag{3}$$

where y_i denotes the observed values and $\hat{y}_i$ denotes predicted values.

III. Support Vector Regression

Support Vector Machines [4] are a relatively new class of learning machines that have evolved from the concepts of structural risk minimization (SRM) and regularization theory. The major difference between support vector machines and many other neural network (NN) approaches is that instead of tackling problems using empirical risk minimization (ERM), SVMs use the concept of regularised ERM. This has enabled people to use SVMs with potentially huge capacities on smaller datasets without running into the usual difficulties of overfittting and poor generalization performance. The basic idea of SVM theory is to (implicitly) map the input data into higher dimensional feature space where the problem can be treated as a linear one. The SVM formulation is desirable due to its mathematical tractability and good generalization properties. In this section, we give standard ϵ-SV regression followed by ν-SV and modified ν-SV regression.

A. ϵ SV Regression

Suppose we are given a training set:

$$\begin{array}{l}\Theta = (\mathbf{x}_1, z_1), (\mathbf{x}_2, z_2), ..., (\mathbf{x}_N, z_N) \\ \mathbf{x}_i \in \Re^{d_L} \\ z_i \in \Re\end{array} \tag{4}$$

which is assumed to have been generated based on some unknown but well defined map $\hat{g} : \Re^{d_L} \rightarrow \Re$, so that $z_i = \hat{g}(\mathbf{x}_i) + \text{noise}$. We define (implicitly, as will be seen shortly) a map $\boldsymbol{\varphi} : \Re^{d_L} \rightarrow \Re^{d_H}$. Using this map, the aim is to find a non-linear approximation g to $\hat{g}$ with the form:

$$g(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) + b \tag{5}$$

which is a linear function of position in feature space (but nonlinear in input space by virtue of the map $\boldsymbol{\varphi}$). The usual ϵ-SVR method of selecting $\mathbf{w}$ and b is to minimize the regularized risk functional:

$$\begin{array}{ll}\min\limits_{\mathbf{w},b,\xi,\xi^*} & R(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{N}\mathbf{1}^T\boldsymbol{\xi} + \frac{C}{N}\mathbf{1}^T\boldsymbol{\xi}^* \\ \text{such that:} & (\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) + b) \geq z_i - \epsilon - \xi_i \\ & (\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) + b) \leq z_i + \epsilon + {\xi_i}^* \\ & \boldsymbol{\xi}, \boldsymbol{\xi}^* \geq \mathbf{0}\end{array} \tag{6}$$

where $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ characterizes the complexity of the model and $\frac{1}{N}\mathbf{1}^T\boldsymbol{\xi} + \frac{1}{N}\mathbf{1}^T\boldsymbol{\xi}^*$ the empirical risk associated with it. The constant $C > 0$ controls the trade-off between empirical risk minimization (potential over-fitting) if C is large and complexity minimization (potential under-fitting) if C is small. The constant $\epsilon > 0$ in eq. 6 is included to give the model a degree of noise insensitivity (assuming that ϵ is well matched to the noise present in the training data). Using lagrange multiplier techniques, the dual form of eq. 6 is [30]:

$$\begin{array}{ll}\min\limits_{\alpha} & L(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{z} + \epsilon\left|\boldsymbol{\alpha}\right|^T\mathbf{1} \\ \text{such that:} & -\frac{C}{N}\mathbf{1} \leq \boldsymbol{\alpha} \leq \frac{C}{N}\mathbf{1} \\ & \mathbf{1}^T\boldsymbol{\alpha} = 0\end{array} \tag{7}$$

where $G_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$ and $|\boldsymbol{\alpha}|$ is the elementwise mod (ie. $|\boldsymbol{\alpha}| \in \Re^N$, $|\boldsymbol{\alpha}|_i = |\alpha_i|$). We also note that:

$$g(\mathbf{y}) = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{y}) + b$$

B. $\nu - SV Regression$

One difficulty with ϵ-SV is the selection of ϵ, which usually requires a trial-and-error approach. To overcome this problem, Scholkopf *et. al.* [28] introduced the ν-SVR formulation, which includes an additional term in the primal problem to trade-off the tube size (ϵ, no longer a constant) against model complexity and empirical risk. From [28], the primal formulation is:

$$\begin{array}{ll}\min\limits_{\mathbf{w},b,\xi,\xi^*,\epsilon} & R = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\nu\epsilon + \frac{C}{N}\mathbf{1}^T\boldsymbol{\xi} + \frac{C}{N}\mathbf{1}^T\boldsymbol{\xi}^* \\ \text{such that:} & (\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) + b) \geq z_i - \epsilon - \xi_i \\ & (\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) + b) \leq z_i + \epsilon + {\xi_i}^* \\ & \boldsymbol{\xi}, \boldsymbol{\xi}^* \geq \mathbf{0} \\ & \epsilon \geq 0\end{array} \tag{8}$$

where $\nu > 0$ is a constant. The associated dual is [28], [30]:

$$\begin{array}{ll}\min\limits_{\alpha} & L(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{z} \\ \text{such that:} & -\frac{C}{N}\mathbf{1} \leq \boldsymbol{\alpha} \leq \frac{C}{N}\mathbf{1} \\ & \mathbf{1}^T\boldsymbol{\alpha} = 0 \\ & \mathbf{1}^T\left|\boldsymbol{\alpha}\right| = C\nu\end{array} \tag{9}$$

where $\mathbf{G}$ is as before. The advantage of this form lies in the properties of the constant ν. It can be shown [28] that:

- $\frac{N_E}{N} \leq \nu$, where N_E is the number of error vectors ($|g(\mathbf{x}_i) - z_i| > \epsilon$, $\alpha_i = C$) in the training set.
- $\frac{N_S}{N} \geq \nu$, where N_S is the number of support vectors ($|g(\mathbf{x}_i) - z_i| \geq \epsilon$, $\alpha_i > 0$) in the training set.

The advantage here is that ν is connected directly to the sparsity of the resulting regressor (where sparsity is the proportion of zero multipliers $\alpha_i = 0$), which is in most cases much easier to select.

C. Modified ν-SV Regression

In this section we describe Modified ν-SV Regression developed by Shilton *et. al.* [31], [30]. In [31], the co-authors have given the theoretical foundation of adaptive SVM and for the first time we use this method to demonstrate its usefulness in application scenario. One practical difficulty with the standard ν-SV regressor is the complexity of the constraint set, and in particular the presence of the constraint $\mathbf{1}^T\left|\boldsymbol{\alpha}\right| = C\nu$. We would like to remove this constraint without losing the ability to automatically select ϵ based on another, more useful parameter, ν. Consider the primal form of the standard ν-SV regression in eq. 8. The term $C\nu\epsilon$ is effectively a linear regularization term for the variable ϵ (in much the same way that $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ is a regularization term for the variable $\mathbf{w}$). Replacing this with a quadratic regularisation term $\frac{C\nu}{2}\epsilon^2$, we get the new regularised risk functional:

$$\begin{array}{ll}\min\limits_{\mathbf{w},b,\xi,\xi^*,\epsilon} & R = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C\nu}{2}\epsilon^2 + \frac{C}{N}\mathbf{1}^T\boldsymbol{\xi} + \frac{C}{N}\mathbf{1}^T\boldsymbol{\xi}^* \\ \text{such that:} & (\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) + b) \geq z_i - \epsilon - \xi_i \\ & (\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) + b) \leq z_i + \epsilon + {\xi_i}^* \\ & \boldsymbol{\xi}, \boldsymbol{\xi}^* \geq \mathbf{0}\end{array} \tag{10}$$

where, once again, $\nu > 0$ is a constant. The associated dual is [31], [30]:

$$\begin{array}{l} \min\limits_{\alpha} L(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{H}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{z} \\ \text{such that:} \quad -\frac{C}{N}\mathbf{1} \leq \boldsymbol{\alpha} \leq \frac{C}{N}\mathbf{1} \\ \qquad\qquad \mathbf{1}^T\boldsymbol{\alpha} = 0 \end{array} \tag{11}$$

where $\mathbf{H} = \mathbf{G} + \frac{1}{C\nu}\text{sgn}(\boldsymbol{\alpha})\,\text{sgn}(\boldsymbol{\alpha})^T$, $\mathbf{G}$ is as before and $\text{sgn}(\boldsymbol{\alpha})$ is the elementwise sigmoid (ie. $\text{sgn}(\boldsymbol{\alpha}) \in \Re^N$, $\text{sgn}(\boldsymbol{\alpha})_i = \text{sgn}(\alpha_i)$). It may also be seen [31], [30] that:

$$\epsilon = \frac{1}{C\nu}\mathbf{1}^T|\boldsymbol{\alpha}| \tag{12}$$

Considering eq. 11, it should be noted that:

- The hessian matrix $\mathbf{H}$ is positive semi-definite and the constraints are linear. Hence there will be no global minima.
- While the modified ν-SV regression method incorporates tube-shrinking into its design, the constraint set of eq. 11 is no more complex than the standard ϵ-SVR dual given by eq. 7.

It can also be shown [30], [31] that:

$$\frac{1}{\nu}\frac{N_E}{N} \leq \epsilon \leq \frac{1}{\nu}\frac{N_S}{N} \tag{13}$$

and hence ν is once again connected with the sparsity of the regressor, simplifying its selection.

IV. Kernels

The function $K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\varphi}(\mathbf{x}_i)^T\boldsymbol{\varphi}(\mathbf{x}_j)$ is called the kernel function. It is not difficult to show that our approximation function $g(\mathbf{x})$ may be written in terms of the kernel function:

$$g(\mathbf{y}) = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{y}) + b \tag{14}$$

The feature map $\boldsymbol{\varphi} : \Re^{d_L} \rightarrow \Re^{d_H}$ are hidden by the kernel function. It is well known that for any function $K : \Re^{d_L} \times \Re^{d_L} \rightarrow \Re$ satisfying Mercer's condition [10], [29], [20] there exists an associated set of feature map $\boldsymbol{\varphi} : \Re^{d_L} \rightarrow \Re^{d_H}$ (although calculating these maps may not be a trivial exercise). Mercer's condition states that $K : \Re^{d_L} \times \Re^{d_L} \rightarrow \Re$ must be a continuous, non-negative definite, symmetric kernel. Indeed, we may start with such a kernel function and, with no knowledge of $\boldsymbol{\varphi}$ at all, optimize and use an SV-regressor. This is referred to as *Kernel trick* [29].

Instead of employing a standard kernel function, to effectively make use of the features extracted we have constructed the following new Mercer kernel using closure properties [33]:

$$k(\mathbf{x}, \mathbf{y}) = k_h(\mathbf{x}_h, \mathbf{y}_h)k_c(\mathbf{x}_c, \mathbf{y}_c) + k_s(\mathbf{x}_s, \mathbf{y}_s) + k_e(\mathbf{x}_e, \mathbf{y}_e) \tag{15}$$

where $\mathbf{x}$ and $\mathbf{y}$ represent the input data with subscript 'h' denoting kernel for evaluating hydrophobic values, 'c' for evaluating polarity values, 's' for evaluating Chou-Fasman secondary structure conformational parameters and 'e' for evaluating evolutionary information extracted in the form of PSSM matrix. The four sub-kernels are defined as follows:

Hydrophobicity and Polarity Sub-Kernel: The hydrophobicity sub kernel k_h is a simple dot product of the free energy values within a window of length L. Jaramillo *et. al.* [11] use a simple summation in their solvation model, to good effect. This motivates us to use a similar model in our kernel function. The polarity sub kernel k_c is similar, but with Grantham polarity scales. The two sub-kernels are represented as shown in eq. 16. The values d_h and d_c help in capturing the local correlation [29]. w is a triangular window with positive real numbers which is used to emphasize the central residue.

$$\begin{array}{rcl} k_h(x_h, y_h) & = & \left[\frac{1}{L}\sum\limits_{i=1}^{L} w(i)x_h(i)y_h(i)\right]^{d_h} \\ k_c(x_c, y_c) & = & \left[\frac{1}{L}\sum\limits_{i=1}^{L} w(i)x_c(i)y_c(i)\right]^{d_c} \end{array} \tag{16}$$

Sub-Kernel to infer the Secondary Structure State: To obtain this kernel, we sum the P_α, P_β and P_c values over a window of length L. The maximum value of the three is considered as the output. Mathematically:

$$k_s(x_s, y_s) = \max\left[\begin{array}{l} \frac{1}{L}\sum\limits_{i=1}^{L} x_{s\alpha}(i)y_{s\alpha}(i), \\ \frac{1}{L}\sum\limits_{i=1}^{L} x_{s\beta}(i)y_{s\beta}(i), \\ \frac{1}{L}\sum\limits_{i=1}^{L} x_{s\gamma}(i)y_{s\gamma}(i) \end{array}\right] \tag{17}$$

where the three terms in max represent the Chou-Fasman parameters calculated for each state α, β and C. The idea here is to pick up the most favorable secondary structure state in the given window.

PSSM Sub-Kernel: We use a simple dot-product [18], [19] to combine PSSM values of the two vectors. The values are between 0 and 1 and to improve the local correlation we raise the entire equation to power d_e. so:

$$k_e(x_e, y_e) = \left[\frac{\sum\limits_{j=1}^{20}\sum\limits_{i=1}^{L} x_e(i,j)y_e(i,j)}{\sum\limits_{j=1}^{20}\sum\limits_{i=1}^{L} x_e(i,j)\sum\limits_{j=1}^{20}\sum\limits_{i=1}^{L} y_e(i,j)}\right]^{d_e} \tag{18}$$

All of the newly defined kernels are symmetric in nature. The kernels k_h and k_c are dot-product kernels with element scaling, and hence satisfy Mercer's condition. k_s is the maximum of three simple dot products and hence is a valid kernel. k_e can be written in the dot product form thusly:

$$k_e(x_e, y_e) = \left[\sum_{i,j}\left[\frac{x_e(i,j)}{\sum\limits_{k,l} x_e(k,l)}, \frac{y_e(i,j)}{\sum\limits_{k,l} y_e(k,l)}\right]\right]^{d_e} \tag{19}$$

and hence also satisfies Mercer's condition.

We compare the proposed kernels with standard Radial Basis Function (RBF) and Polynomial kernels defined as follows:

- RBF: $K(x, y) = exp\left(\frac{-\|x-y\|^2}{\gamma}\right), \gamma > 0.$
- Polynomial: $K(x, y) = \left(\gamma x^T y + r\right)^d, \gamma > 0.$

TABLE I

COMPARISON WITH STANDARD KERNELS WITH PROPOSED KERNEL IN TERMS OF MAE AND SUPPORT VECTORS.

	Support Vectors	0-5	5-10	10-15	15-20	20-25	Global MAE
RBF(Gamma = 7)	3746	0.10	0.12	0.10	0.09	0.08	0.11
RBF (Gamma = 10)	3641	**0.10**	**0.12**	0.10	0.09	0.08	**0.11**
Proposed Kernel (Weighted)	4955	0.17	0.13	**0.08**	**0.06**	**0.05**	0.12
Proposed Kernel (Non - Weighted)	5161	0.20	0.14	0.09	0.09	0.06	0.14
Polynomial Kernel (d = 3)	5296	0.21	0.16	0.12	0.08	0.03	0.15

TABLE II

MAE VALUES FOR VARIOUS VALUES OF ν.

ν	0-5	5-10	10-15	15-20	20-25	25-30	30-35	35-40	40-45	45-50	50-100
0.01	0.46	0.40	0.35	0.30	0.25	0.20	0.15	0.10	0.05	0.01	0.13
0.05	0.39	0.34	0.29	0.24	0.19	0.15	0.10	0.05	0.02	0.05	0.18
0.1	0.34	0.29	0.25	0.20	0.15	0.11	0.07	0.03	0.04	0.08	0.21
0.2	0.29	0.26	0.21	0.17	0.13	0.09	0.06	0.05	0.07	0.10	0.22
0.3	0.27	0.24	0.20	0.16	0.12	0.08	0.06	0.05	0.08	0.11	0.23
0.4	0.26	0.23	0.20	0.16	0.11	0.08	0.07	0.06	0.08	0.11	0.23
0.5	0.25	0.23	0.19	0.15	0.11	0.08	0.07	0.06	0.08	0.11	0.23
1	0.27	0.24	0.20	0.16	0.12	0.08	0.06	0.05	0.07	0.11	0.23
2	0.25	0.23	0.19	0.15	0.11	0.08	0.07	0.06	0.08	0.11	0.23

Results are shown in table I. 0, 5, 10, etc. in table I are solvent accessibility thresholds. *Weighted* indicates that the three sub-kernels are weighted unequally ($0.25, 0.25, 0.5$ respectively in this experiment). *Non-weighted* means the weights are equal to 1.

Calculating Free Parameters

To calculate the free parameters, we selected 20% of proteins with minimum length of 55 and belonging to each class (All α, All β, $\alpha + \beta$, Few Secondary Structures) as defined by CATH. This was divided into two sets of 15% and 5% as training and testing sets respectively. As described earlier, the new SVR formulation eliminates ϵ and introduces ν which is a free parameter. For several values of ν we calculated MAE and RMSE. We found that above a certain value (approx 0.5 for our problem) the effect of ν is negligible. Based on this we choose $\nu = 2$ for all our experiments. It may be seen that this value should be increased for larger data sets. The MAE and RMSE values for various values of ν are shown in table II.

The window length L was chosen experimentally by variation from $L = 1$ to $L = 19$. MAE for various values of window length L is given in table III. From the table we chose a constant window length of $L = 11$ for rest of our experiments.

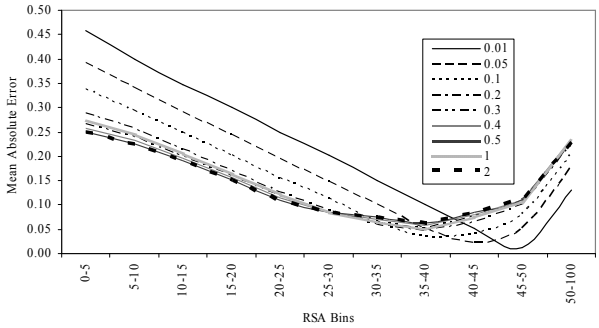

Fig. 1. Plot of MAE vs regression bins for various values of ν

V. RESULTS AND DISCUSSION

3-fold cross-validation (CV) was carried out on PSA472 data set after dividing the data into three sets randomly. We performed cross-validation using both RBF kernel and the novel kernel presented here. We found that the number of support vectors using RBF kernel was 30.81% and using the proposed kernel was 38.62% of the total training vectors. The results of the CV are as shown in table IV. The other major result of this paper is the use of modified support vector regression where the system automatically chooses the value of ϵ. We have shown that a reasonably high value of ν (wherein $\nu > 1$), which is independent of the noise present in the system, gives good results consistently.

Finally, we compared our method with the other real value prediction methods in literature [23], [6], [1], [7]. Manesh dataset [17] was used to make this objective comparison as this was the commonly used dataset. Table V summarizes classification results of several systems for different RSA

TABLE IV

MAE VALUES FOR 3 FOLD CROSS-VALIDATION

	MAE		RMSE	
SA Bins	RBF	Proposed Kernel (Weighted)	RBF	Proposed Kernel (Weighted)
0-5	0.12	0.15	0.14	0.19
5-10	0.12	0.14	0.14	0.17
10-15	0.10	0.09	0.12	0.11
15-20	0.08	0.08	0.10	0.09
20-25	0.07	0.07	0.09	0.08
25-30	0.07	0.06	0.08	0.08
30-35	0.07	0.08	0.09	0.10
35-40	0.09	0.10	0.12	0.13
40-45	0.13	0.12	0.15	0.15
45-50	0.17	0.17	0.19	0.19
50-100	0.26	0.29	0.29	0.32
Overall	0.13	0.15	0.15	0.19

TABLE III
MAE VALUES FOR VARIOUS VALUES OF L

L	0-5	5-10	10-15	15-20	20-25	25-30	30-35	35-40	40-45	45-50	50-100
1	0.44	0.38	0.33	0.28	0.23	0.18	0.13	0.08	0.04	0.02	0.16
3	0.35	0.30	0.26	0.20	0.15	0.11	0.07	0.04	0.05	0.10	0.21
5	0.29	0.24	0.22	0.17	0.12	0.08	0.06	0.07	0.08	0.12	0.21
7	0.27	0.23	0.22	0.15	0.12	0.09	0.07	0.06	0.09	0.11	0.20
9	0.27	0.22	0.21	0.15	0.13	0.08	0.07	0.06	0.08	0.10	0.21
11	0.25	0.22	0.21	0.15	0.13	0.09	0.07	0.07	0.08	0.10	0.20
13	0.24	0.21	0.20	0.15	0.13	0.09	0.07	0.08	0.09	0.11	0.20
15	0.23	0.19	0.19	0.14	0.12	0.09	0.07	0.08	0.10	0.11	0.21
17	0.23	0.19	0.19	0.14	0.12	0.09	0.07	0.09	0.09	0.11	0.22
19	0.63	0.58	0.55	0.50	0.47	0.42	0.37	0.31	0.24	0.23	0.14

thresholds T. Threshold T is used to indicate whether a residue is buried ($\leq T$) or exposed ($> T$). As it can be seen from table V, the proposed method (ASVM) performs better than all other methods for RSA thresholds $> 20\%$. For other thresholds, our method is the *second* best. The first eleven columns of table V indicate two state classification (buried or exposed). The 12th and the 13th columns indicate three state classification (buried, intermediate and exposed). The last column gives mean absolute error (MAE) for real valued prediction. We report the best MAE to date (0.12) on Manesh dataset.

Figure 2 shows the average mean absolute error obtained for all the amino acids in Manesh dataset [17]. Hydrophobic amino acids Valine (V), Isoleucine (I), Leucine (L) and Phenylalanine (F) give the lowest mean absolute error. This indicates the system's sensitiveness to the hydrophobic residues. Figures 3 and 4 show the plot of observed and predicted relative solvent accessibility values for proteins 1PDA and 1SLU from Manesh dataset. 1PDA gives the lowest mean absolute error of 0.09 and 1SLU gives the highest mean absolute error of 0.156. From the plots, it is clear that the values predicted follows the observed values very closely. If we compare only the recent SVR method [23] with our method, we report better performance with single stage SVR against two-stage SVR. This result is important as it emphasizes the fact that the data representation used is contributing significantly. In [23], authors choose $\epsilon = 0.001$. Choosing this value can be a bit tricky and our system is free from this parameter. The correct value of noise insensitivity parameter, ϵ, is important to get the best result from a support vector regressor. Other than data representation, the improved performance of our single stage SVR compared to the two-stage SVR [23] is due to automatic calculation of ϵ.

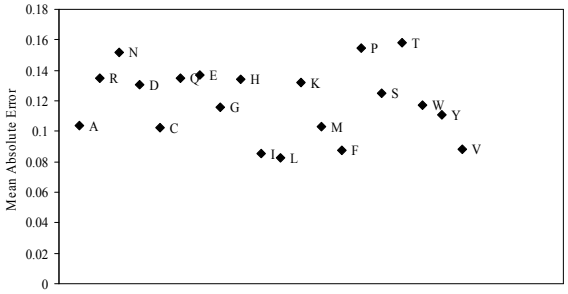

Fig. 2. Average MAE values for different amino acids in Manesh dataset

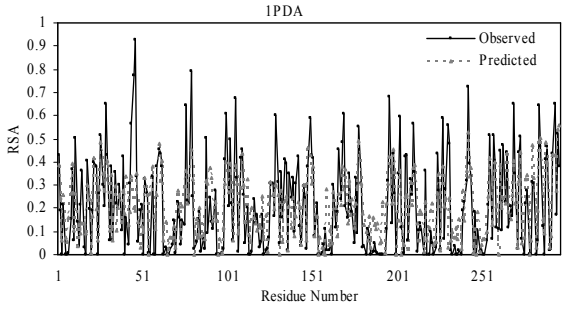

Fig. 3. Predicted and Observed RSA for protein 1PDA in Manesh dataset. 1PDA gives the lowest MAE of 0.090

VI. CONCLUSION

Real valued relative solvent accessibility prediction using adaptive support vector regression is proposed. Novel kernels are employed which combine secondary structure statistics, solvation model, electrostatic model and evolutionary information in the form of PSSM. A new variant of support vector regression which is free from choosing ϵ, the noise insensitivity parameter is presented for the first time in an application scenario. A new dataset containing 472 proteins has been curated (PSA472) from recent version of CATH with sequence identity less than 20% to validate our method. We get overall mean absolute error of 0.13 and 0.15 on PSA472 dataset. The proposed method is compared with a few other methods using Manesh dataset [17]. On this set we report the lowest mean absolute error (0.12) to date. The usefulness of the proposed technique is demonstrated by making use of it in protein topology prediction with highly encouraging results [9].

REFERENCES

[1] S. Ahmad and M. M. Gromiha. Netasa: Neural network based prediction of solvent accessibility. *Bioinformatics*, 18:819–824, 2002.

[2] S. Ahmad, Gromiha M. M., and A. Sarai. Rvp-net: online prediction of real valued accessible surface area of proteins from single sequences. *Bioinformatics*, 19:1849–1851, 2003.

[3] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blasat: a new generation of protein database search programs. *Nucleic Acid Research*, 27(17):3389–3402, 1997.

[4] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20(3):273–297, 1995.

TABLE V

PERFORMANCE COMPARISON OF ADAPTIVE SUPPORT VECTOR REGRESSION AND OTHER REAL VALUED PREDICTION METHODS IN LITERATURE USING MANESH DATASET [17]. USING DIFFERENT THRESHOLDS, TWO CLASS CLASSIFICATION IS CARRIED OUT AND PERFORMED. LAST COLUMN IS THE OVERALL MAE FOR DIFFERENT SYSTEMS. IF THRESHOLD CONTAINS TWO VALUES, IT MEANS IT IS A THREE CLASS CLASSIFICATION

Threshold	0	5	10	20	25	30	40	50	60	70	80	10-20	25-50	MAE
ASVR	88.7	77.1	76.8	**77.7**	**77.8**	**78.1**	**81.3**	**88.2**	**94.5**	**97.9**	**99.4**	**64.3**	**68**	**0.12**
NguyenSVR [23]	-	**81.1**	**78.5**	77.6	77.3	-	-	79.5	84.3	89.9	95.0	-	-	0.15
Garg [6]	-	74.9	77.2	77.7	-	77.8	78.1	80.5	85.3	90.7	95.1	-	-	0.17
Ahmad [1]	87.9	74.6	71.2	-	70.3	-	-	75.9	-	-	-	63	55	0.18
Giaenese [7]	**89.5**	75.7	73.4	-	71.6	-	-	76.2	-	-	-	-	-	-

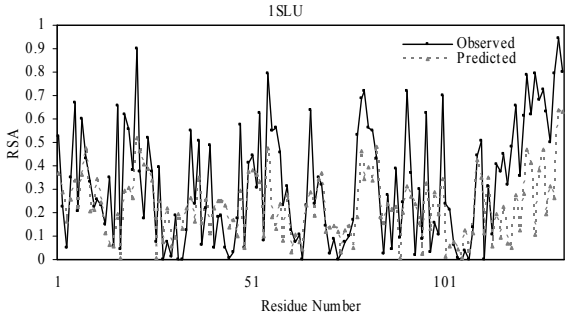

Fig. 4. Predicted and Observed RSA for protein 1SLU in Manesh dataset. 1SLU gives the highest MAE of 0.156

[5] J. A. Cuff and G. J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins*, 34:508–519, 1999.

[6] A. Garg, H. Kaur, and G. P. S. Raghava. Real value prediction fo solvent accessibility in proteins using multiple sequence alignment and secondary structure. *Proteins: Structure, Functions and Bioinformatics*, 61:318–324, 2005.

[7] G. Gianese, F. Bossa, and S. Pascarella. Improvement in prediction of solvent accessibility by probability profiles. *Proteins Engineering*, 16(12):987–992, 2003.

[8] R. Grantham. Amino acid difference formula to help explain protein evolution. *Science*, 185:862–864, 1974.

[9] J. Gubbi, A. Shilton, M. W. Parker, and M. Palaniswami. Protein topology classification using two-stage support vector machines. *Genome Informatics - To appear*, 17(2), 2006.

[10] D. Haussler. Convolution kernels on discrete structures. *Technical Report*, UCS-CRL-99(10), 1999.

[11] A. Jaramillo and S. J. Wodak. Computational protein design is a challenge for implicit solvation model. *Biophysical Journal*, 88:156–171, 2005.

[12] G. L. Jayavardhana Rama, M. Palaniswami, D. Lai, and M. W. Parker. A study on the effect of physico-chemical properties in protein secondary structure prediction. In *Applied Artificial Intelligence*, pages 609–616. World Scientific, 2006.

[13] D. T. Jones, W. R. Taylor, and J. M. Thornton. A model recognition approach to the prediction of all-helical membrane protein structure and topology. *Biochemistry*, 33:30383049, 1994.

[14] W. Kabsch and C. Sander. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577–2637, 1983.

[15] H. Kim and H. Park. Protein secondary structure prediction based on an improved support vector machines approach. *Protein Engineering*, 16(8):553–560, 2003.

[16] H. Kim and H. Park. Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3d local descriptor. *PROTEINS: Structure, Function, and Bioinformatics*, 54:557–562, 2004.

[17] H. N. Manesh, M. Sadeghi, S. Arab, and A. M. and Movahedi. Prediction of protein surface accessibility with information theory. *Proteins*, 42:452–459, 2001.

[18] L. J. McGuffin, K. Bryson, and D. T. Jones. The psipred protein structure prediction server. *Bioinformatics*, 16:404–405, 2000.

[19] L. J. McGuffin, R. T. Smith, K. Bryson, S. A. Sorensen, and D. T. Jones. High throughput profile-profile based fold recognition for the entire human proteome. *BMC Bioinformatics*, 7(288), 2006.

[20] J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Transactions of the Royal Society of London*, 209(A), 1909.

[21] S. Mika and B Rost. Uniqueprot: creating representative protein-sequence sets. *Nucleic Acids Research*, 31(13):3789–3791, 2003.

[22] M. N. Nguyen and J. C. Rajapakse. Two-stage support vector machines to protein relative solvent accessibility prediction. In *Proceedings of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 67–72, 2004.

[23] M. N. Nguyen and J. C. Rajapakse. Two-stage support vector regression approach for predicting accessible surface areas of amino acids. *PROTIENS: Structure, Function, and Bioinformatics*, 63:542–550, 2006.

[24] T. Ooi, M. Oobatake, G. Nemethy, and H. A. Scheraga. Accessible surface areas as a measure of the thermodynamic parameters of hydration of peptides. *Proc. of Natl. Acad. Sci.*, 84:3086–3090, 1987.

[25] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton. Cath- a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1108, 1997.

[26] G. Pollastri, P. Baldi, P. Fariselli, and R. Casadio. Prediction of coordination number and relative solvent accessibility in proteins. *PROTEINS: Structure, Function, and Bioinformatics*, 47:142–153, 2002.

[27] B. Rost. Phd: predicting one-dimensional protein structure by profile based neural networks. *Methods in Enzymology*, 266:525–539, 1996.

[28] B. Scholkopf, P. Bartlett, A. Smola, and R. Williamson. Shrinking the tube: A new support vector egression algorithm. *Advances in Neural Information Processing Systems*, 11:330–336, 1999.

[29] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and beyond.* MIT Press, USA, 2002.

[30] A. Shilton. Design and training of support vector machines. *PhD Thesis, The University of Melbourne*, 2006.

[31] A. Shilton and M. Palaniswami. A modified ν-SV method for simplified regression. In *Proceedings of the International Conference on Intelligent Sensing and Information Processing*, pages 422–427, 2004.

[32] J. Sim, S-Y. Kim, and J. Lee. Prediction of protein solvent accessibility using fuzzy k-nearest neighbor method. *Bioinformatics*, 21:2844–2849, 2005.

[33] J. S. Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[34] M. Wagner, R. Adamczak, A. Porollo, and J. Meller. Linear regression models for solvent accessibility prediction in proteins. *Journal of Computational Biology*, 12(3):355–369, 2005.

[35] Z. Yuan and T. L. Bailey. Prediction of protein solvent profile using svr. In *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, pages 2889–2892, 2004.

[36] Z. Yuan, K. Burrage, and J. S. Mattick. Prediction of solvent accessibility using support vector machines. *Proteins: Structure, Functions and Genetics*, 48:566–570, 2002.

[37] Z. Yuan and B. Huang. Prediction of protein accessible surface areas by support vector regression. *Proteins:Structure, Function and Bioinformatics*, 57:558–564, 2004.