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 predicting its structure 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
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 $\epsilon$ which is usually done by trial and error (Although this fact is often skinned over). The method described in this paper uses an improved tube shrinking method developed by Shilton et al. [31] for automatically calculating $\epsilon$. 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, 1bd0, 1beo, 1bif, 1bmf, 1bnc, 1btm, 1bti, 1cem, 1ceo, 1cew, 1cfy, 1chd, 1ckh, 1cxy, 1dea, 1del, 1dkz, 1dos, 1fua, 1gai, 1gpl, 1gsa, 1gtm, 1hav, 2iib, 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 $\epsilon$ support vector regression followed by $\nu$ support vector regression and the modified $\nu$ 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 resolution of at least 2 Å. UniProt [21] with HSSP-value of $\theta$ 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 $\alpha$ Helix, E for $\beta$ 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):

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

(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 $x_0$). 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 PDI values (8) (scale was obtained from http://au.expasy.org/tools/protscale.html) ($x_0$) 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 ($\alpha$, $\beta$ and C) using Chou-Fasman method. The Chou-Fasman parameter for Helix($\alpha$) is given by $P_{\alpha} = f_{\alpha}/(f_0)$ where $f_0$=Number of Residues in Helix/Total Number of Residues and ‘$i$’ ranges over the set of amino-acid residues. Similar conformational parameters for strand $P_{\beta}$ and coil $P_{\beta}$ are calculated ($x_0$). 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).

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

(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:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| y_i - \hat{y}_i \right|$$

(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 overfitting and poor generalization performance. The basic idea of SVM theory is to (implicitly) map the input data into a higher dimensional feature space where the problem can be treated as a linear one. The SVM formulation is desirable due to its empirical tractability and good generalization properties. In this section, we give standard SVM regression followed by $\nu$-SV and modified $\nu$-SV regression.

#### A. $\epsilon$-SV Regression

Suppose we are given a training set:

$$\Theta = (x_1, z_1), (x_2, z_2), \ldots, (x_N, z_N)$$

where $x_i \in \mathbb{R}^d$, $z_i \in \mathbb{R}$

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

$$g(x) = w^T \varphi(x) + b$$

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

$$\min_{w, b, \xi, \xi^*} R(w, b, \xi, \xi^*) = \frac{1}{2} w^T w + \frac{C}{N} \sum_i \xi_i + \frac{\epsilon}{N} \sum_i \xi^*_i$$

such that:

$$w^T \varphi(x_i) + b \geq z_i - \xi_i$$

$$\xi_i, \xi_i^* \geq 0$$

(6)

where $\frac{1}{2} w^T w$ characterizes the complexity of the model and $\frac{C}{N} \sum_i \xi_i + \frac{\epsilon}{N} \sum_i \xi^*_i$ 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 $\epsilon$ is well matched to the noise present in the training data). Using lagrange multiplier techniques, the dual form of eq. 6 is [30]:

$$\min \alpha \in \mathbb{R}^N \frac{1}{2} \alpha^T \alpha - \alpha^T \xi + \epsilon |\alpha|_1$$

such that:

$$-\frac{\epsilon}{N} \leq \alpha \leq \frac{C}{N}$$

(7)

$$1^T \alpha = 0$$

where $G_{i,j} = K(x_i, x_j)$ and $|\alpha|$ is the elementwise (mod. i.e. $|\alpha| \in \mathbb{R}^N, |\alpha_i| = |\alpha_i|$). We also note that:

$$g(y) = \sum_i \alpha_i K(x_i, y) + b$$

#### B. $\nu$-SV Regression

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

$$\min_{w, b, \xi, \xi^*} R = \frac{1}{2} w^T w + C\nu - \frac{C}{N} \sum_i \xi_i + \frac{\epsilon}{N} \sum_i \xi^*_i$$

such that:

$$w^T \varphi(x_i) + b \geq z_i - \nu + \xi_i$$

$$\xi_i, \xi^*_i \geq 0$$

(8)

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

$$\min \alpha \in \mathbb{R}^N \frac{1}{2} \alpha^T \alpha - \alpha^T \xi + \epsilon |\alpha|_1$$

such that:

$$-\frac{C}{N} \leq \alpha \leq \frac{C}{N}$$

$$1^T \alpha = 0$$

$$\frac{1}{2} \nu = C\nu$$

(9)

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

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

The advantage here is that $\nu$ 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 $\nu$-SV Regression

In this section we describe Modified $\nu$-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 $\nu$-SV Regressor is the complexity of the constraint set, and in particular the presence of the constraint $1^T \alpha = C\nu$. We would like to remove this constraint without losing the ability to automatically select $\epsilon$ based on another, more useful parameter, $\nu$. Consider the primal form of the standard $\nu$-SV regression in eq. 8. The term $C\nu$ is effectively a linear regularization term for the variable $\epsilon$ (in much the same way that $\frac{1}{2} w^T w$ is a regularization term for the variable $w$). Replacing this with a quadratic regularization term $C\nu^2$, we get the new regularised risk functional:

$$\min_{w, b, \xi, \xi^*} R = \frac{1}{2} w^T w + C\nu^2 - \frac{C}{N} \sum_i \xi_i + \frac{\epsilon}{N} \sum_i \xi^*_i$$

such that:

$$w^T \varphi(x_i) + b \geq z_i - \nu + \xi_i$$

$$\xi_i, \xi^*_i \geq 0$$

(10)
where, once again, \( \nu > 0 \) is a constant. The associated dual is [31], [30]:
\[
\min L(\alpha) = \frac{1}{2} \alpha^T H \alpha - \alpha^T z
\]
such that:
\[
\frac{-C}{N} \leq \alpha \leq \frac{C}{N},
\]
\[
1^T \alpha = 0
\]
where \( H = G + \frac{1}{C} \text{sgn}(\alpha) \text{sgn}(\alpha)^T \), \( G \) is as before and \( \text{sgn}(\alpha) \) is the elementwise sigmoid (ie. \( \text{sgn}(\alpha) \in \mathbb{R}^N \), \( \text{sgn}(\alpha)_i = \text{sgn}(\alpha_i) \)). It may also be seen [31], [30] that:
\[
\epsilon = \frac{1}{2} N \| \alpha \|
\]
Considering eq. 11, it should be noted that:
- The hessian matrix \( H \) is positive semi-definite and the constraints are linear. Hence there will be no global maxima.
- While the modified \( \nu \)-SV regression method incorporates tube-shrinking into its design, the constraint set of eq. 11 is no more complex than the standard \( \epsilon \)-SVR dual given by eq. 7.

It can also be shown [30], [31] that:
\[
\frac{1}{\nu} N \leq \epsilon \leq \frac{1}{\nu} N
\]
and hence \( \nu \) is once again connected with the sparsity of the regressor, simplifying its selection.

### IV. Kernels

The function \( K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \) is called the kernel function. It is not difficult to show that our approximation function \( g(X) \) may be written in terms of the kernel function:
\[
g(y) = \sum_i \alpha_i K(x_i, y) + b
\]

The feature map \( \varphi : \mathbb{R}^{d_L} \rightarrow \mathbb{R}^{d_H} \) are hidden by the kernel function. It is well known that for any function \( K : \mathbb{R}^{d_L} \times \mathbb{R}^{d_L} \rightarrow \mathbb{R} \) satisfying Mercer’s condition [10], [29], [20] there exists an associated set of feature map \( \varphi : \mathbb{R}^{d_L} \rightarrow \mathbb{R}^{d_H} \) (although calculating these maps may not be a trivial exercise). Mercer’s condition states that \( K : \mathbb{R}^{d_L} \times \mathbb{R}^{d_L} \rightarrow \mathbb{R} \) must be a continuous, non-negative definite, symmetric kernel. Indeed, we may start with such a kernel function and, with no knowledge of \( \varphi \) 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]:
\[
k(x, y) = k_h(x_0, y_0) k_c(x_v, y_v) + k_h(x_s, y_s) + k_c(x_e, y_e)
\]
where \( x \) and \( y \) 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 \( d_e \) help in capturing the local correlation [29]. \( \nu \) is a triangular window with positive real numbers which is used to emphasize the central residue.

\[
k_h(x_h, y_h) = \left[ \frac{1}{L} \sum_{i=1}^{L} w(i) x_h(i) y_h(i) \right]^{d_h}
\]
\[
k_c(x_c, y_c) = \left[ \frac{1}{L} \sum_{i=1}^{L} w(i) x_c(i) y_c(i) \right]^{d_e}
\]

Sub-Kernel to infer the Secondary Structure State: To obtain this kernel, we sum the \( P_a, P_b \) and \( P_c \) values over a window of length \( L \). The maximum value of the three is considered as the output. Mathematically:
\[
k_e(x_e, y_e) = \max \left[ \frac{1}{L} \sum_{i=1}^{L} x_e(i) y_e(i), \frac{1}{L} \sum_{i=1}^{L} x_a(i) y_a(i), \frac{1}{L} \sum_{i=1}^{L} x_c(i) y_c(i) \right]
\]
where the three terms in \( \max \) represent the Chou-Fasman parameters calculated for each state \( \alpha, \beta \) 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:
\[
k_e(x_e, y_e) = \left[ \frac{1}{L} \sum_{i=1}^{L} x_e(i) y_e(i) \right]^{d_e}
\]

All of the newly defined kernels are symmetric in nature. The kernels \( k_h \) and \( k_e \) are dot-product kernels with element scaling, and hence satisfy Mercer’s condition. \( k_h \) 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:
\[
k_e(x_e, y_e) = \left[ \frac{1}{L} \sum_{i=1}^{L} x_e(i) y_e(i) \right]^{d_e}
\]

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:** \( K(x, y) = (\gamma x^T y + r)^d, \gamma > 0 \).
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, α + β, 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 ν = 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.

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 ν > 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.
Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CICB 2007)

TABLE III

MAE VALUES FOR VARIOUS VALUES OF L

<table>
<thead>
<tr>
<th>L</th>
<th>0-5</th>
<th>5-10</th>
<th>10-15</th>
<th>15-20</th>
<th>20-25</th>
<th>25-30</th>
<th>30-35</th>
<th>35-40</th>
<th>40-45</th>
<th>45-50</th>
<th>50-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.44</td>
<td>0.38</td>
<td>0.33</td>
<td>0.28</td>
<td>0.23</td>
<td>0.18</td>
<td>0.13</td>
<td>0.08</td>
<td>0.04</td>
<td>0.02</td>
<td>0.16</td>
</tr>
<tr>
<td>3</td>
<td>0.35</td>
<td>0.30</td>
<td>0.26</td>
<td>0.20</td>
<td>0.15</td>
<td>0.11</td>
<td>0.07</td>
<td>0.04</td>
<td>0.02</td>
<td>0.01</td>
<td>0.05</td>
</tr>
<tr>
<td>5</td>
<td>0.29</td>
<td>0.24</td>
<td>0.22</td>
<td>0.17</td>
<td>0.12</td>
<td>0.08</td>
<td>0.06</td>
<td>0.07</td>
<td>0.08</td>
<td>0.12</td>
<td>0.21</td>
</tr>
<tr>
<td>7</td>
<td>0.27</td>
<td>0.23</td>
<td>0.22</td>
<td>0.15</td>
<td>0.12</td>
<td>0.09</td>
<td>0.07</td>
<td>0.06</td>
<td>0.09</td>
<td>0.11</td>
<td>0.20</td>
</tr>
<tr>
<td>9</td>
<td>0.27</td>
<td>0.22</td>
<td>0.21</td>
<td>0.15</td>
<td>0.13</td>
<td>0.08</td>
<td>0.07</td>
<td>0.06</td>
<td>0.08</td>
<td>0.10</td>
<td>0.21</td>
</tr>
<tr>
<td>11</td>
<td>0.25</td>
<td>0.22</td>
<td>0.21</td>
<td>0.15</td>
<td>0.13</td>
<td>0.09</td>
<td>0.07</td>
<td>0.07</td>
<td>0.08</td>
<td>0.10</td>
<td>0.20</td>
</tr>
<tr>
<td>13</td>
<td>0.24</td>
<td>0.21</td>
<td>0.20</td>
<td>0.15</td>
<td>0.13</td>
<td>0.09</td>
<td>0.07</td>
<td>0.08</td>
<td>0.09</td>
<td>0.11</td>
<td>0.20</td>
</tr>
<tr>
<td>15</td>
<td>0.23</td>
<td>0.19</td>
<td>0.19</td>
<td>0.14</td>
<td>0.12</td>
<td>0.09</td>
<td>0.07</td>
<td>0.08</td>
<td>0.10</td>
<td>0.11</td>
<td>0.21</td>
</tr>
<tr>
<td>17</td>
<td>0.23</td>
<td>0.19</td>
<td>0.19</td>
<td>0.14</td>
<td>0.12</td>
<td>0.09</td>
<td>0.07</td>
<td>0.09</td>
<td>0.11</td>
<td>0.11</td>
<td>0.22</td>
</tr>
<tr>
<td>19</td>
<td>0.63</td>
<td>0.58</td>
<td>0.55</td>
<td>0.50</td>
<td>0.47</td>
<td>0.42</td>
<td>0.37</td>
<td>0.31</td>
<td>0.24</td>
<td>0.23</td>
<td>0.14</td>
</tr>
</tbody>
</table>

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 insensitivity to the hydrophobic residues. Figures 3 and 4 show the plot of observed and predicted relative solvent accessibility values for proteins IPDA and 1SLU from Manesh dataset. IPDA 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 ε = 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.

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

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

<table>
<thead>
<tr>
<th>Threshold</th>
<th>0</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>10-20</th>
<th>25-50</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASVR</td>
<td>8.87</td>
<td>77.1</td>
<td>76.8</td>
<td>77.7</td>
<td>77.8</td>
<td>78.1</td>
<td>81.3</td>
<td>88.2</td>
<td>94.5</td>
<td>97.9</td>
<td>99.4</td>
<td>64.3</td>
<td>68</td>
<td>0.12</td>
</tr>
<tr>
<td>NguyenSVR [23]</td>
<td>-</td>
<td>81.4</td>
<td>78.5</td>
<td>77.6</td>
<td>77.3</td>
<td>-</td>
<td>-</td>
<td>80.5</td>
<td>85.3</td>
<td>90.7</td>
<td>95.1</td>
<td>-</td>
<td>-</td>
<td>0.17</td>
</tr>
<tr>
<td>Garg [6]</td>
<td>-</td>
<td>74.9</td>
<td>77.2</td>
<td>77.7</td>
<td>-</td>
<td>77.8</td>
<td>78.1</td>
<td>80.5</td>
<td>85.3</td>
<td>90.7</td>
<td>95.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ahmad [1]</td>
<td>87.9</td>
<td>74.6</td>
<td>71.2</td>
<td>-</td>
<td>70.3</td>
<td>-</td>
<td>-</td>
<td>75.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>63</td>
<td>55</td>
<td>0.18</td>
</tr>
<tr>
<td>Giarense [7]</td>
<td>89.5</td>
<td>75.7</td>
<td>73.4</td>
<td>-</td>
<td>71.6</td>
<td>-</td>
<td>-</td>
<td>76.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

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


Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Gubbi, J; Shilton, A; Palaniswami, M; Parker, M

Title:
Real value solvent accessibility prediction using adaptive support vector regression

Date:
2007-01-01

Citation:

Publication Status:
Published

Persistent Link:
http://hdl.handle.net/11343/34871