A Comparative Study between Fixed-size Kernel Logistic Regression and Support Vector Machines Methods for beta-turns Prediction in Protein

Murtada Khalafallah Elbashir

Faculty of Mathematical and Computer Sciences, University of Gezira, Wad Madani, 20, Sudan

murtadabashir@yahoo.com

ABSTRACT

Beta-turn is an important element of protein structure; it plays a significant role in protein configuration and function. There are several methods developed for prediction of beta-turns from protein sequences. The best methods are based on Neural Networks (NNs) or Support Vector Machines (SVMs). Although Kernel Logistic Regression (KLR) is a powerful classification technique that has been applied successfully in many classification problems, however it is often not found in beta-turns classification, mainly because it is computationally expensive. Fixed-Size Kernel Logistic Regression (FS-KLR) is a fast and accurate approximate implementation of KLR for large-scale data sets. It uses trust-region Newton's method for large-scale Logistic Regression (LR) as a basis, to solve the approximate problem, and Nystrom method to approximate the features' matrix. In this paper we used FS-KLR for beta-turns prediction and the results obtained are compared to those obtained with SVM. Secondary structure information and Position Specific Scoring Matrices (PSSMs) are utilized as input features. The performance achieved using FS-KLR is found to be comparable to that of SVM method. FS-KLR has an advantage of yielding probabilistic outputs directly and its extension to the multi-class case is well-defined. In addition its evaluation time is less than that of SVM method.

Key word: Fixed-size, Kernel Logistic Regression, Support Vector Machines

INTRODUCTION

Proteins are an important class of biological macromolecules present in all organisms. They play a key role in almost all biological processes. Protein structures describe the various levels of organization of protein molecules. There are four levels of protein structures, these levels are Primary structure, Secondary structure, Tertiary structure, and Quaternary structure. Turns are one of the basic elements in proteins secondary structure. Different turns are classified according to the separation between the two end residues. Beta-turns are the most common found type of turns that constitute approximately 25% of the residue in protein [1,2,3,4]. They play a significant role in protein configuration and function, and its formation is a vital stage during the protein folding. They were found to be more helpful in the context of molecular recognition and in modeling interactions between peptide substrates receptors, because they tend to be more solvent exposed than buried [5]. Beta-turns are further classified into different types according to the dihedral angles (ϕ and ψ) of the central two residues. The classification scheme proposed by Hutchinson and Thornton [6] recognizes nine distinct types of beta-turn: Types I, I', II, II', VIa1, VIa2, VIb, VIII, IV. In this classification, the most frequently occurring type is type IV which constitute approximately (35%) of the beta-turns. Types VIa1,VIa2, and VIb are rare types. A classifier that can be extended straight forwardly will be useful in predicting the different types of beta-turns.

Several prediction methods have been designed for beta-turns prediction. This is because beta-turns identification in protein is found to be helpful in fold recognition and drug design [7]. The beta-turn prediction methods can be divided into two main categories; machine learning approaches and statistical approaches. The machine learning approaches include BTPRED [8], BetaTPred2 [3, 9], MOLEBRNN [10], and NetTurnP [11], which are based on Artificial Neural Networks (ANN), Kim's method [12] based on K-nearest neighbor (KNN). The most recent prevailing methods are based on Support Vector Machines (SVMs) [7,13-16]. The statistical based approaches include the Chou-Fasman method [17], Thornton's algorithm [18], GORBYURN [19], the 1-4 & 2-3 correlation models [20], and the sequence-coupled model [21].

In general, the machine learning approaches outperform the statistical based methods. Almost all the current successful
machine learning approaches for beta-turns prediction utilize position specific scoring matrices and secondary structure information as features for
prediction. The best machine learning approaches devised for beta-turns prediction are based on ANN or SVM.

SVM is one of the kernel-based machine learning algorithms developed for binary classification. It was first proposed by Vapnik and Cortes based on statistical learning theory. It delivers state-of-the-art performance in real-world applications [22]. The SVM method requires solving a constrained quadratic optimization problem with a computational complexity of $O(n^3)$ where n is the number of training instances. An iterative chunking method, where the overall problem is divided into small active training set was designed to implement SVMs in large scale-dataset. The extreme form of chunking is the Sequential Minimal Optimization (SMO)[23]. LIBSVM, which is the state-of-the-art toolbox [24], uses the SMO solver described in [25]. LIBSVM has

been used by many of the current beta-turns prediction methods that are based on SVM, and it delivers high performance in predicting beta-turns [26,27]. However, SVM methods have weakness in that the training time for large-scale data sets such as beta-turns data sets sometimes is unrealistic.

Kernel Logistic Regression (KLR) is another type of kernel-based machine learning algorithms. It is the kernel version of Logistic Regression (LR), a well-known classifier in the field of statistical learning theory. It can be obtained by constructing the LR in higher-dimensional space using the kernel function. Unlike NNs and SVMs, KLR includes the probabilities of occurrences as a natural extension. Moreover KLR can be extended straight forwardly to handle multi-class classification problems, and it requires solving an only unconstrained optimization problem. However, KLR is not used in beta-turns prediction, because of its computational complexity. Karsmakers [28,29] proposed a fast and accurate approximate implementation of KLR for Automatic Speech Recognition (ASR). He described a different practical technique suited for large data sets, based on Fixed-Size Least Squares Support Vector Machines (FS-LSSVMs) [29] which he named Fixed Size Kernel Logistic Regression (FS-KLR). FS-KLR approximates the KLR

problem using Nystrom method, the approximate problem is solved using Newton's method for large-scale LR [30].

In this study FS-KLR is used for beta-turns prediction and the results compared to those obtained by the SVM. Secondary structure information predicted using four prediction methods and multiple sequence alignment in the form of PSI-BLAST-generated position-specific scoring matrices (PSSMs) [31] are utilized as input features. The results show that FS-KLR is as effective as SVM in predicting Beta-turns. In addition FS-KLR has the advantage of yielding probabilistic outputs, and its extension to multi-class classification is well-defined. This is appropriate for beta-turns' types prediction. Moreover, FS-KLR provides sparse solution which as a consequence relatively short evolution time.

State-of-the-Art Methods in beta-Turns Prediction

The state of the art beta-turns prediction methods are based on machine learning, namely ANN and SVMs. Shepherd et al (1999) used ANN to predict beta-turns in, they incorporated secondary structure information in the input data. In their method the total percentage of residues correctly classified as beta-turn or not-beta-turn was around 75% with predicted secondary structure information. Their method gave a Matthews correlation coefficient (MCC) of around 0.35 [8]. Kaur H et al (2003) developed a neural network-based method for the prediction of beta-turns in proteins by using multiple sequence alignment. Two feed-forward back-propagation networks with a single hidden layer were used where the first-sequence structure network was trained with PSSMs. The initial predictions from the first network and PSIPRED-predicted secondary structure [32] were used as input to the second structure-structure network to refine the predictions obtained from the first net. The final network produced an overall prediction accuracy of 75.5% when tested by seven-fold cross-validation on a set of 426 nonhomologous protein chains. The corresponding Matthews correlation coefficient value is 0.43 [9]. Kirschner A et al (2008) presented a bidirectional Elman-type recurrent neural network with multiple output layers (MOLEBRNN) capable of predicting multiple mutually dependent structural motifs and

demonstrated its efficiency in recognizing three aspects of protein structure: beta-turns, beta-turn types, and secondary structure.

In a seven-fold cross-validation experiment on a standard test dataset their method exhibited the total prediction accuracy of 77.9% and the Mathew's Correlation Coefficient of 0.45 [10]. Bent Petersen et al (2010) presented a neural network method, which they called NetTurnP, for prediction of two-class b-turns and prediction of the individual b-turn types, by use of evolutionary information and predicted protein sequence features. Their method achieved Qtotal of 78.2, and MCC of 0.50 on BT426 data set [11]. Pham TH et al (2003) introduced a SVM approach for prediction and analysis of beta-turns. They have investigated two aspects of applying SVM to the prediction and analysis of beta-turns. Their developed SVM method was called BTSVM, which predicts beta-turns of a protein from its sequence. The prediction results on the dataset of 426 non-homologous protein chains by seven-fold cross-validation technique achieved Qtotal of 75.8, and MCC of 0.44 using PSSMs [14]. Zhang Qet al (2005) developed a method of beta-turn prediction that uses the SVM algorithm together with predicted secondary structure information. Their SVM method achieved a MCC of 0.45 and Qtotal of 77.3% using a seven-fold cross-validation on a database of 426 non-homologous protein chains [15]. Hu X, and Li Q (2008) proposed SVM algorithm for predicting beta-turns and gamma-turns in the proteins by using the composite vector with increment of diversity, position conservation scoring function, and predictive secondary structures to express the information of sequence. They achieved overall prediction accuracy and the MCC in seven-fold cross-validation of 79.8% and 0.47, respectively on The 426 non-homologous protein chains dataset [16]. Ce Zheng and Kurgan (2008) proposed a method for the prediction of beta-turns based on SVM. Their method used features extracted from a window over the three state secondary structure predicted by an ensemble of four methods. they used feature selection to reduce the dimensionality of the feature vector. They achieved a Qtotal of 80.9% , and MCC of 0.47 on BT426 dataset [7].

Support Vector Machines

The Support Vector Machine (SVM) is a relatively new and promising classification and regression technique proposed by Vapnik and Cortes at AT&T Bell Laboratories. It classifies data by constructing a hyperplane or set of hyperplanes in a high or infinite dimensional space. Its theory follows the classical empirical risk minimization approach, which determines the classification decision function by minimizing the empirical risk as follows [28];

$$R = \frac{1}{l}\sum_{i=1}^{l} \left| f(x_i) - y_i \right| \quad \text{..... (1)}$$

where l and f are the number of examples and the classification decision function, respectively. The primary concern of the SVM is to determine an optimal separating hyperplane that gives low generalization error. Usually, the classification decision function in the linearly non-separable problem is represented by the following quadratic optimization problem;

$$\underset{\alpha}{Min}\ f(\alpha) = \frac{1}{2}\alpha^T Q\alpha - e^T\alpha, \quad \ldots\ldots (2)$$

Subject to

$$0 \le \alpha_i \le C, i = 1, 2, \ldots\ldots, l,$$
$$y^T\alpha = 0,$$

Where e is the vector of all ones, c is the upper bound of all variables, Q is a n by n symmetric matrix with $Q_{ij} = y_i y_j K(x_i, x_j)$ and $K(x_i, x_j)$ is the kernel function.

The matrix Q for large scale data is too large to be stored in the computer memory; there are many decomposition methods that are designed to solve this problem [33-35]. Unlike most optimization methods which update the whole vector α in each step of an iterative process, the decomposition method modifies only a subset of α per iteration. This subset, denoted as the working set B, leads to a small sub-problem to be minimized in each iteration. An extreme case is the Sequential Minimal Optimization (SMO), which restricts B to have only two elements. Then in each iteration one does

not require any optimization software in order to solve a simple two-variable problem. LIBSVM, which has gained wide popularity in machine learning and many other areas, is designed based on the above SMO [24].

There are many kernels are available that can be chosen; the popular ones are the following statistical kernels [33].

Linear kernel: $K(x, x') = x^T x'$

Polynomial kernel: $K(x, x') = (c + x^T x')^d, c \ge 0, d > 0$

Radial Basis Function (RBF) kernel: $K(x, x') = \exp(-\|x - x'\|^2 / 2\sigma^2)$

Fixed-Size Kernel Logistic Regression

Kernel Logistic Regression (KLR) is the kernel version of logistic regression, which is a well-known statistical model for classification. Unlike LR, KLR enables the classification of linearly non-separable problems by transferring the input features to a higher-dimensional space, via the kernel trick. The kernel is a transformation function that must satisfy Mercer's necessary and sufficient conditions, which state that a kernel function must be expressed as an inner product and must be positive semi-definite. Similar to LR, KLR can be fitted using Maximum Likelihood Estimate (MLE) [28].

Iteratively Reweighted Least Square (IRLS) algorithm is one of the most popular techniques used to find the MLE of the LR models. IRLS is a nonlinear optimization algorithm that uses a series of Weighted Least Squares (WLS) sub-problems to search for the MLE. It is a special case of Fisher's scoring method, a quasi-Newton algorithm that replaces the objective function's Hessian with the Fisher information. For LR, IRLS is a special form of Newton's method in

which each iteration finds the WLS estimates for a given set of weights, which are used to construct a new set of weights. KLR also can be fitted effectively using IRLS [29].
Unlike SVMs, KLR does not use risk minimization principle, but it is based on conditional maximum likelihood inference, which results in estimates of a posteriori class probabilities via logit stochastic models.

$$P(Y=-1|X=x;f)=\frac{\exp(f(x))}{1+\exp(f(x))} \quad \text{..... (3)}$$

$$P(Y=1|X=x;f)=\frac{1}{1+\exp(f(x))}$$

Where $f(x)=w^T\varphi(x)+b$, w is the vector of the KLR parameters, and b is the intercept.
The solution w can be expressed in terms of α computed using IRLS iteration as

$$w=\sum_{i=1}^{N}\alpha_i\varphi(x_i) \quad \text{.....(4)}$$

In the dual representation the function values $f(x)$ in the KLR logit models can be computed as follows

$$f(x)=\sum_{i=1}^{N}\alpha_i K(x,x_i)+b \quad \text{..... (5)}$$

Where $K(x,x_i)=\varphi(x)^T\varphi(x_i)$
The IRLS method is suitable for small size problems, but for large scale problems this methods become computationally expensive. Based on Fixed-Size Least Squares Support Vector Machines (FS-LSSVMs) Karsmakers [29, 30] implemented a Fixed-Size variant of the standard KLR formulation (FS-KLR) which does easily scale to very large data sets. In his method he adopted Nystrom approximation method.
In Nystrom approximation the kernel matrix will be decomposed into eigenvalues/eigenvectors matrices in the form.

$$K_{n\times n}=U_n\Lambda_n U_n^T \quad \text{.....(6)}$$

Where Λ_n=diag(λ_i), and $\lambda_1\geq\lambda_2\geq\ldots\geq\lambda_n\geq 0$ are the eigenvalues of the matrix K, U_n is the matrix of the eigenvectors that correspond to the eigenvalues, and n is the number of the data points.
The first p eigenvectors and eigenvalues form the matrices U and Λ respectively can be selected, where

$p \leq n$, to approximate the kernel matrix. This approximation is motivated by it is widely usage e.g. principal component analysis. Using this approximation can reduce the computational cost drastically. However computing the eigen-decomposition is also computationally expensive. To reduce the computational cost of computing the eigen-decomposition we can select small sample of size $m \leq n$ from the features matrix to construct the following eigen-problem:

$$K_{m\times m} = U_m \Lambda_m U_m^T \quad \text{.....(7)}$$

Now we can extend the eigenvalues/ eigenvectors of the $K_{m\times m}$ to all points using the following Nystrom approximation.

$$\tilde{\lambda}_i^{(n)} = \frac{n}{m}\lambda_i^{(m)} \quad \text{..... (8)}$$

$$\tilde{u}_i^{(n)} = \sqrt{\frac{m}{n}}\frac{1}{\lambda_i^{(m)}} K_{n,m} u_i^{(m)} \text{..... (9)}$$

where $\lambda_i^{(m)}$, and $u_i^{(m)}$ are the i_{th} eigenvalue/ eigenvector of the $m\times m$ eigen-problem, and $K_{n,m}$ is the appropriate $n\times m$ sub matrix of K.

The selected sample of size m from the feature matrix can be called Prototype Vectors (PVs). These PVs can be selected using K-center clustering. The use of K-center clustering is justified in [36], which observed that the Nystrom low-rank approximation depends crucially on the quantization error induced by encoding the sample set with landmark points. This suggests that one can simply use the clusters obtained with a k-center (such as k-means) algorithm, which finds a local minimum of the quantization error (Note that using this technique the PVs do not necessarily coincidence with the training data). The PVs selection methods using K-center clustering, suffer from the fact that they will select outliers as prototypes. In cases where the number of PVs is relatively small, the fraction of prototypes chosen to represent the non-outlier and outlier data is unbalanced and therefore the classification performance will not be optimal. When the number of PVs is increased, the performance will also increase to

that of KLR. Hence taking into account outliers can result in a sparser model. The sparse kernel logistic regression problem is solved in the primal space using the Newton's based method. Newton trust region algorithm, which is given in [37]. This algorithm yielded the best performance compared to state-of-the-art alternatives. A good balance between convergence speed and cost per iteration is found in that low cost approximate Newton steps are taken in the beginning of the algorithm and full Newton directions at the end for fast convergence.

MATERIAL AND METHODS

Data Sets

The SVM and FS-KLR were trained and tested on three non-redundant data sets BT426, BT547, and BT823, they were downloaded from http://biomine.ece.ualberta.ca /BTNpred/ BTNpred.html. These data sets contain chains with at least one beta-turn and they have X-ray

crystallographic resolution better than 2.0. All protein chains have less than 25% sequence similarity, to ensure that there is very little homology in the training. The dataset BT426 consists of 426 protein chains. It was created by Guruprasad and Rajkumar [38], and consists of 96339 amino acids. In total 23,580 amino acids, corresponding to 24.9% of all amino acids, have been assigned to be located in b-turns. The data set BT426 has been used by the majority of recent b-turn prediction methods and, therefore it can be a good benchmark for comparing different beta-turns prediction methods. BT547 and BT823 data sets contain 547 and 823 protein sequences respectively. The total number of residue is 104522, and 150969 in BT547, and BT823 respectively. They were constructed for training and testing COUDES [4].

Features Vector

The features in the above mentioned data sets include PSSMs, and secondary structure information. It has been shown that the PSSMs contributed significantly to the accuracy of beta-turns prediction [7]. The PSSMs are in the form of $20*M$, where M represents the sequence length, they can be generated using the Position-Specific Iterative Basic Local Alignment Search Tool (PSI-BLAST) program against National Center for

Biotechnology Information (NCBI) non-redundant (nr) sequence database. PSI-BLAST is a sequence similarity search method, in which a query protein or nucleotide sequence is compared to nucleotide or protein sequences in a target database to identify regions of local alignment and report those alignments that score above a given score threshold. The PSSMs values are scaled to values between 0 and 1. The secondary structures were predicted as three structures: helix, strand and coils using Four secondary structure prediction methods, PSIPRED v2.5 [32, 39], JNET [40], TRANSSEC [41], and PROTEUS2 [41]. These secondary structures information were encoded using unary encoding scheme, in which the helix are encoded as (1 0 0), the coils as (0 1 0), and the strands as (0 0 1). The feature vector is computed using a window of size seven over the PSSMs and the predicted secondary structures that are centered on the predicted residue. So the total number of the features that are based on PSSMs and secondary structure information is (20*7+4*3=152). Another 64 features were added to the features, 4 of them are the confidence score of the central amino acid using the four prediction methods, 48 features representing a binary value for a specific configuration of the secondary structure using the four methods for the central and two adjacent residue, and 12 features representing the ratio between the number of residues in a given secondary structure and the window size. Thus the total number of features is 216. Features selection methods based on information gain and CHI-squared were employed to reduce the features to 90.

Experimental Setup

For the SVM we used LIBSVM software package for the training and testing. The Radial Basis Function (RBF) is used as a kernel function. The two parameters c and gamma of the SVM were optimized using the default grid search approach. For the FS-KLR we used the package FS-KLR, which was downloaded from ftp://ftp.esat.kuleuven.ac.be/pub/sista/karsmakers/ software / index.html. The parameter of FS-KLR is optimized based on cross validation approach. Selecting the number of PVS from the feature matrix is an important task in FS-KLR, as mentioned above this number of PVs will be used to approximate the kernel matrix. A relatively small number of PVs will yield low

performance where as a big number of PVS will increase the evaluation time. To select the optimum number of PVs we used cross

validation approach starting with relatively small number, and adding more vectors to the PVs until a point where adding more vectors does not improve the classification performance significantly reached.

In order to evaluate a prediction method it is necessary to have different data sets for training and testing. The jackknife test is the most objective and rigorous cross validation method compared with independent data set test and sub-data set test [42]. In a full jackknife test of N proteins, one protein is removed from the set, the training is done on the remaining $N-1$ proteins and the test is done on the removed protein. This process is repeated N times by removing protein in turn. Since this training technique is very time consuming most of the recent beta-turns prediction methods use seven-fold cross-validation to assess their performances. Also we used seven-fold cross- validation to assess the accuracy of SVM and FS-KLR.

In seven-fold cross-validation, the data sets will be divided into seven subsets, each containing equal number of proteins. Each set is an unbalanced set that retains the naturally occurring proportion of beta-turns. Six of the seven subsets were merged together to form a training set that was used to train both the SVM and the FS-KLR methods, and the seventh was used for validation. This process was repeated seven times in order to have a different set for validation each time. The final prediction results are taken as the average of the results from the seven testing sets.

Performance Measures

The measures that are used in this study can be divided in the following two categories: Threshold dependent measures, these measures relies on the following quantities: *TP* (true positives) is the number of correctly classified beta-turn residues, *TN* (true negatives) is the number of correctly classified non beta -turn residues, *FP* (false positives) is the number of non -beta turn incorrectly classified as beta -turn residues, and *FN* (false negatives) is the number of beta -turn incorrectly classified as non -beta -turn residues. The threshold dependent measures that are used in this study are: Qtotal (prediction accuracy), Qpred (Probability of correct prediction), Qobserved(sensitivity or coverage), and Matthews Correlation Coefficient (MCC). Qtotal (prediction accuracy) is defined as the percentage of correctly classified residues.

$$Q_{total} = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad \text{…...……(10)}$$

Qpredicted or Probability of correct prediction, it is the percentage of correctly predicted beta-urns among the predicted beta-turns

$$\text{Qpredicted} = \frac{TP}{TP + FP} \times 100 \quad \text{……(11)}$$

Qobserved, it is the percentage of correctly predicted beta-turns among the observed (true) beta-urns or (percent coverage)

$$\text{Qobserved} = \frac{TP}{TP + FN} \times 100 \quad \text{.....(12)}$$

Because of the imbalanced dataset (25% beta-turns), Qtotal is a poor measure by itself, as it is possible to achieve a Qtotal of 75% if all residues were predicted to be non-beta-turns. As a result, MCC is very important measure that takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient between the observed and predicted binary classifications.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \quad \text{.....(13)}$$

One of the problems with the threshold dependent measures is that they measure the performance on a given threshold. They fail to use all the information provided by a method for evaluation. For this reason we employed a threshold independent measure, which is the Receiver Operating Characteristic (ROC) or simply ROC curve. From the ROC curve we can calculate the area under the curve (AUC), which measures the ability of a method to correctly classify beta-turns and non-beta-turns residues.

RESULTS

As mentioned in the experimental setup section, the set of features that are composed of PSSMs and secondary structure information were used as the input to the optimized FS-KLR and SVM classifiers to perform prediction of beta-turns. The evaluation of the quality of these predictions was performed based on seven-fold cross-validation on the BT426 dataset, see Table 1. The table shows that FS-KLR results in high MCC value. It can be seen from the table that the difference in the Qtotal (prediction accuracy) is relatively small as both of the FS-KLR and SVM provide the accuracy which is close to 80.5%, but it is higher for the SVM.

Table 1.Performance comparison between the FS-KLR and the SVM based on the seven-fold cross-validation test on theBT426 dataset

Method	MCC	Qtotal	Qpredicted	Qobserved	AUC
FS-KLR	0.48	80.5	60.0	63.0	0.861
SVM	0.46	80.9	62.8	53.8	0.847

Although Qtotal favors SVM, but because of the unbalanced data set (25% beta-turns), Qtotal is a poor measure by itself, as it is possible to achieve Qtotal of 75% by predicting all residues to be non-beta-turns. Instead FS-KLR achieved a relatively large MCC, which will also regarded as a balanced measure. The Qobserved of the FS-KLR is higher by 9.2% than the QObserved of the

SVM. The Qobserved value obtained by the FS-KLR method shows that 63% of actual β- turns were correctly predicted. We note that Qpredicted value of FS-KLR is 2.2% lower than the Qpredicted of the SVM method, but from the definition of the Qobserved and the Qpredicted, the increase in the Qobserved can compensate the decrease in the Qpredicted.

Table 2.Performance comparison between the FS-KLR and the SVM based on the seven-fold cross-validation test on theBT547 dataset

Method	MCC	Qtotal	Qpredicted	Qobserved	AUC
FS-KLR	0.48	80.4	59.4	63.3	0.860
SVM	0.45	80.9	63.0	50.5	0.847

Table 3.Performance comparison between the FS-KLR and the SVM based on the seven-fold cross-validation test on theBT823 dataset

Method	MCC	Qtotal	Qpredicted	Qobserved	AUC
FS-KLR	0.47	81.1	58.2	61.0	0.860
SVM	0.45	81.3	60.3	54.2	0.842

Besides BT426 dataset used for training and testing, we utilized two additional datasets, i.e. BT547 and BT823 datasets, to validate the performance of the FS-KLR and SVM. Table 2 shows the results obtained based on the seven-fold cross-validation with these datasets. These results show that for the BT547 dataset, the FS-KLR method obtains 0.5% lower Qtotal,3% higher MCC,12.8% better Qobserved, and 3.6% lower Qopredicted
when compared with the SVM method. The Qobserved of FS-KLR shows that more than 63% of the observed beta-turns are correctly predicted, we emphasize that this is a high percentage in beta-turns prediction and can compensate the percentage of correctly predicted beta-turns among the predicted beta-turns (Qpredicted) given that the difference in Qpredicted between FS-KLR and SVM is relatively small, while the difference in Qobserved is relatively high, and it fevours FS-KLR. Similarly, for the BT823 dataset, FS-KLR obtains 2% and 6.8% higher MCC and Qobserved, respectively, and 0.2% decrease in Qtotal and 2.1 decreases in Qpredicted as shown in Table 3.
We observe that the FS-KLR is very stable over all the three data sets. The Qtotal values range between 81.1% and 80.5%. The same trend of stable prediction is seen for all other performance measures as well; as we see that the Qpredicted values range between 60.0% and 58.2%, Qobserved between 61.0% and 63.3%, and MCC between 0.47 and 0.48, see Tables1, 2, and 3.

This rules out possibility of over fitting the BT426 dataset due to the performed design.

It can be seen from the Tables 1, 2, and 3 that SVM has higher Qtotal and lower MCC, the same can be seen for Qpredicted and Qobserved.It is obviously not possible to compare the methods objectively, therefore, Besides using the above measures, we calculated the area under the ROC curve in the b-turn prediction, as mentioned is a threshold independent measure. The performance of both FS-KLR and SVM has been assessed using the ROC curve as shown in Figure 1, the corresponding areas under the ROC curves are 0.861, 0.860 and 0.860 for the FS-KLR on the BT426, BT547, and BT832, respectively. And the areas under the ROC curves for the SVM on these data sets are 0.847, 0.847,and 0.842 respectively.

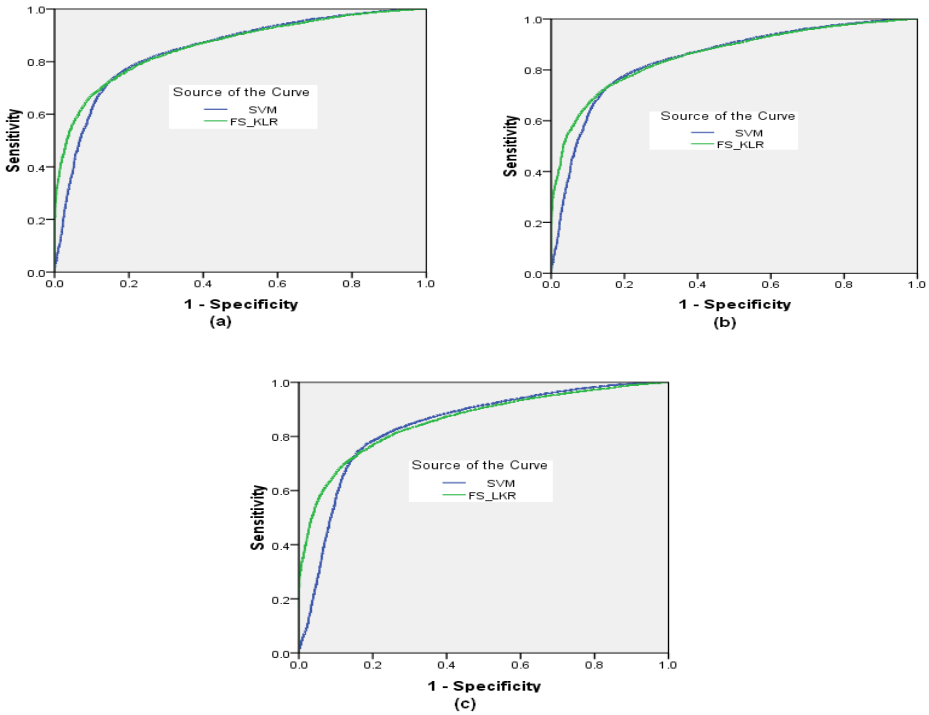

Figure 1. (a) The ROC curve for BT426 data set, (b) The ROC curve for BT547data set, (c) The ROC curve for BT823 data set

Figure 2 shows the average execution time of FS-KLR and SVM in function of the number of training instances. The input data used for the figure is from BT426. It is clear that the execution time of support vector machine increases rapidly as the number of training instances increases, while the execution time of FS-KLR increases slowly as the number of instances increases.

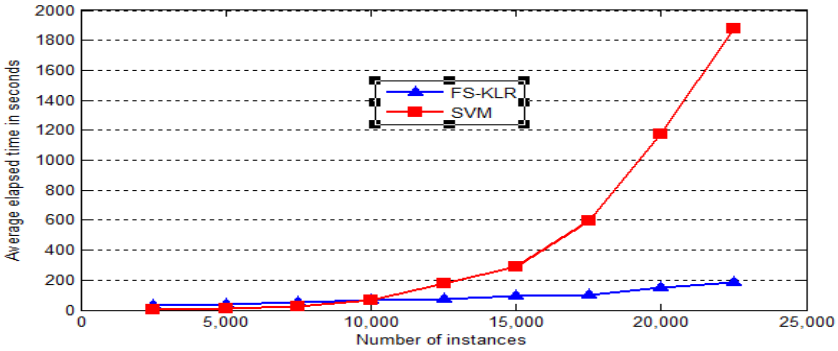

Figure 2. average execution time of FS-KLR and SVM in function of the number of training instances

DISCUSSIONS

Performance comparison of different methods plays a vital role in the development of any field of science. In this study we compared the performance of FS-KLR and SVM for beta-turn prediction in protein. Beta-turns play and important roles due to the following facts: (i) it is important for a polypeptide chain to fold into compact globular folds; (ii) beta-turns usually occur on the exposed surface of a protein and hence it is involved in molecular recognition process; (iii) also play important role in protein folding and stability; and (iv) beta-turns are also involved in the biological activity of peptides as the bioactive structures that interact with other

molecules such as receptors, enzymes, or antibodies. Recent years have seen interest in mimicking beta-turns for the synthesis of medicines. Thus, beta-turn is an important component of protein structure whose prediction can provide enormous information to the researchers working in the field of protein structure prediction, so the prediction of beta-turns would not only aid in overall tertiary structure prediction but also assists in fold recognition studies. Throughout the preceding research on beta-turn prediction, predictors based on machine learning method emphasize selecting proper features to improve prediction performance. Now secondary structures and PSSMs are widely used in the predictions, and have been proven to be the most helpful features. Using these features FS-KLR acheives comparable results to the SVM method. To design a method that can be applied in beta-turn prediction, there are four main concerns, these concerns are: (1) the size of the data set which the method is processing , (2) the need for dealing with input examples of variable length, (3) the desire to have probabilistic outcomes, and (4) the need to perform multiclass classification. When the dataset is very large such as the beta-turns data, people neglect the last two concerns and concentrates on selecting classifier that deal with large datasets effectively. Since SVM methods are designed in a way that can handle large scale data sets they become the choice for most of the beta-turns classification purpose. However, SVM do not address the last two concerns. KLR is not used in large scale data sets such as beta-turns data classification although it provides elegant solution to the last two concerns, simply because it is inapplicable in such data sets. The last two concerns are very important for beta-turns classification, since there is a need for multiclass classification for the beta-turns type. FS-KLR extends the applicability of KLR for large scale data sets. This way FS-KLR can address all of the aforementioned concerns.

As stated in the experimental setup subsection, selecting the number of PVs from the features matrix using K-means clustering algorithm is an important task in FS-KLR. Figure 3 shows that with small number of PVs we can achieve good accuracy. It shows also that selecting large number of PVs will not increase the accuracy much; it will rather increase the processing time. The input data used for the figure is from BT426 data set. In general Figure 3 shows that we can select a number of vectors from the feature matrix that is by far less than the number of instances without any significant decrease in the MCC and the accuracy of the solution.

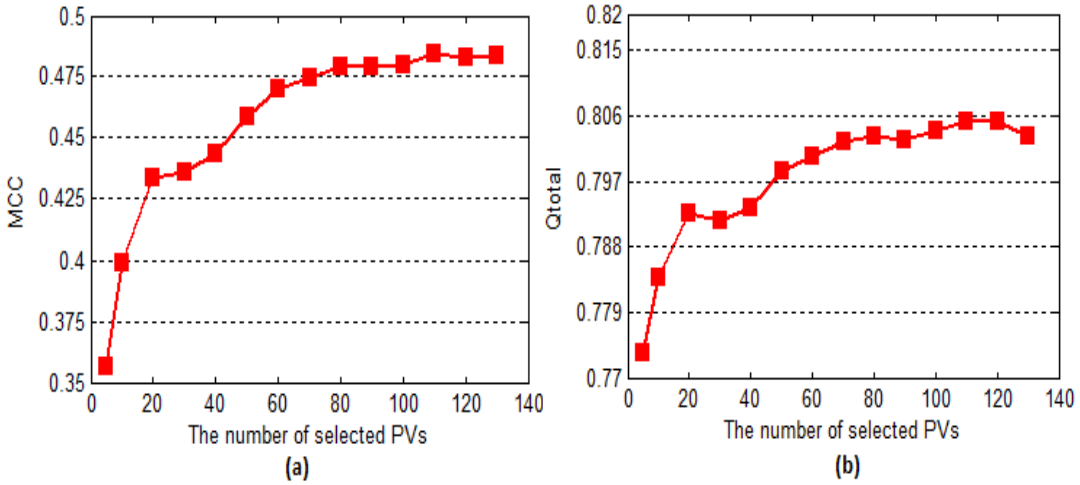

Figure 3. (a) The number of selected PVs in function of the MCC , (b) The number of selected PVs in function of the Qtotal

CONCLUSIONS

This paper provides a benchmarking of FS-KLR and SVM methods for beta-turns prediction. FS-KLR is proposed by Karsmakers to obtain a fast and accurate approximate implementation of KLR for Automatic Speech Recognition (ASR). FS-KLR has not been applied before in beta-turns prediction. These two methods are trained and tested on three data sets using seven-fold cross-validation. PSSMs and secondary structure predicted using four different prediction methods are used as input for both of the FS-KLR and SVM methods. The MCC of FS-KLR is better than SVM; where as the accuracy of SVM is slightly better than that of FS-KLR. The ROC curve is the same for both of the methods. FS-KLR has an advantage of yielding probabilistic output, and its extension to multi-class prediction is well defined. In addition the evaluation time of FS-KLR is found to be less than that of SVM.

REFERENCES

Chou, K.C. (2000). Prediction of tight turns and their types in proteins. Anal. Biochem. 286, 1–16

Chou, K.C., Blinn, J.R. (1997). Classification and prediction of beta-turn types. J. Protein Chem. 16, 575–595.

Kaur, K.S., Raghava, G.P.(2004). A neural network method for prediction of beta-turn types in proteins using evolutionary information. Bioinformatics 16, 2751–2758.

Fuchs, P.F.J.,Alix,A.J.P. (2005).High accuracy prediction of b-turn and their types using propensities and multiple alignments. Proteins 59,828–839.

Rose GD, Gierasch LM, Smith JA (1985). Turns in peptides and proteins. Adv Protein Chem. 37, 100-9.

Hutchinson EG, Thornton JM (1994). A revised set of potentials for b-turn formation in proteins, Protein Sci, 3, pp.2207-2216.

Ce Zheng, Lukasz Kurgan (2008), Prediction of beta-turns at over 80% accuracy based on an ensemble of predicted secondary structures and multiple alignments, BMC Bioinformatics 9:340.

Shepherd AJ, Gorse D, Thornton JM (1999), Prediction of the location and type of beta-turns in proteins using neural networks. Protein Sci 8:1045-55.

Kaur H, Raghava GPS (2003), Prediction of β-turns in proteins from multiple alignment using neural network. Protein Sci , 12:627-34.

Kirschner A, Frishman D (2008), Prediction of beta-turns and beta-turn types by a novel bidirectional Elman-type recurrent neural network with multiple output layers (MOLEBRNN). Gene, 422(1–2):22-9.

Petersen B, Lundegaard C, Petersen TN (2010) NetTurnP—neural network prediction of beta-turns by use of evolutionary information and predicted protein sequence features. PLoS One 5:e15079

Kim S (2004), Protein β-turn prediction using nearest-neighbor method. Bioinformatics, 20:40-4.

Cai YD, Liu XJ, Xu XB, Chou KC (2002), Support vector machines for the classification and prediction of beta-turn types. J Peptide Sci, 8:297-301.

Pham TH, Satou K, Ho TB(2003), Prediction and analysis of beta-turns in proteins by support vector machine. Genome Inform 2003, 14:196-205.

Zhang Q, Yoon S, Welsh WJ (2005), Improved method for predicting β-turn using support vector machine. Bioinformatics, 21:2370-4.

Hu X, Li Q (2008), Using support vector machine to predict beta and gamma-turns in proteins. J Comput Chem , 29(12):1867-75.
Chou PY, Fasman G (1974). Conformational parameters for amino acids in helical, β-sheet and random coil regions calculated from proteins. Biochemistry 1974, 13:211-22.
Wilmot CM, Thornton JM (1988), Analysis and prediction of the different types of β-turns in proteins. J Mol Biol, 203:221-32.
Wilmot CM, Thornton JM (1990), β-Turns and their distortions: a proposed new nomenclature. Protein Eng 1990, 3:479-93.
Zhang CT, Chou KC (1997), Prediction of beta-turns in proteins by 1–4 & 2–3 correlation model. Biopolymers, 41:673-702.
Chou KC(1997), Prediction of beta-turns. J Peptide Res, 49:120-144.
Nello Cristianini, John Shawe-Taylor (2000), An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods, Cambridge university press.
J.C. Platt (1999), Fast training of support vector machines using sequential minimal optimization, Advances in Kernel Methods: Support Vector Learning, MITPress, pp. 185-208.
C.C. Chang, C.J. Lin (2001), LIBSVM : a library for support vector machines, Software available at http: // www. csie. ntu. edu. tw/ ˜ cjlin/ libsvm.
R.-E. Fan, P.-H. Chen, C.-J. Lin (2005), Working set selection using second order information for training SVM, Journal of Machine Learning Research, Vol.6, pp. 1889-1918.
Zehui Tang, Tonghua Li, Rida Liu, Wenwei Xiong, Jiangming Sun, Yaojuan Zhu and Guanyan Chen (2011), Improving the performance of b-turn prediction using predicted shape strings and a two-layer support vector machine model, BMC bioinformatics.

Xiaobo Shi,XiuzhenHun, ShaoboLi, XingxingLiu (2011), Prediction of b-turn types in proteinby using composite vector, Journal of Theoretical Biology 286, pp. 24-30.
P. Karsmakers (2010), Sparse kernel-based models for speech recognition, PhD thesis, Katholieke Universiteit Leuven,Arenberg Doctoral School of Science, Engineering & Technology, Belgium, may.
P. Karsmakers, K. Pelckmans, J. A. K. Suykens (2007), Multi-class kernel logistic regression: a fixed-size implementation, In Proc. of the international joint conference in neural networks (IJCNN), Orlando, Florida, U.S.A, pp. 1756-1761.
C.-J. Lin, R.C. Weng, S.S. Keerthi (2008), Trust Region Newton Method for Large-Scale Logistic Regression, Journal of Machine Learning Research, Vol. 9, pp. 623-646.
Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997), Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25, 3389-3402.
McGuffin LJ, Bryson K, Jones DT (2000), The PSIPRED protein structure prediction server. Bioinformatics, 16(4), 404-5.
E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In Proceedings of CVPR'97, pages 130–136, New York, NY, (1997). IEEE.

Thorsten Joachims (1998) Making large-scale SVM learning practical. In Bernhard Sch¨olkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, Advances in Kernel Methods- Support Vector Learning, Cambridge, MA, MIT Press.
John C. Platt (1998), Fast training of support vector machines using sequential minimal optimization. In Bernhard Sch¨olkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, Advances in Kernel Methods - Support Vector Learning, Cambridge, MA, MIT Press.

K. Zhang, I.W. Tsang, J.T. Kwok (2008) Improved Nystrom low-rank approximation and error analysis, In Proc. of the 25th International Conference on Machine Learning, Helsinki, Finland, pp. 1232-1239
C.-J. Lin, R.C. Weng, S.S. Keerthi (2008), Trust Region Newton Method for Large-Scale Logistic Regression, Journal of Machine Learning Research, Vol. 9, pp.623-646.
Guruprasad K, Rajkumar S. (2000), Beta-and gamma-turns in proteins revisited:a new set of amino acid turn-type dependent positional preferences and potentials, J Biosci, 25(2), pp.143-156.
Bryson K, McGuffin LJ, Marsden RL, Ward JJ, Sodhi JS, Jones DT (2005), Protein structure prediction servers at University College London.Nucl Acids Res, W36-38.
Cuff JA, Barton GJ (2000), Application of multiple sequence alignment profiles to improve protein secondary structure prediction.Proteins , 15:502-11.
Montgomerie S, Sundararaj S, Gallin WJ, Wishart DS (2006), Improving the accuracy of protein secondary structure prediction using structural alignment. BMC Bioinformatics, 14:301.
Chou K, Zhang C. (1995), Prediction of protein structural classes. Critical Reviews in Biochem And Mol Biol. 11:275–349.

دراسة مقارنة لطريقتي ال**انحدار اللوجستى المبنى على الكيرنال ثابت الحجم** و**الات المتجه الداعم في تقدير** التفاف بيتا في البروتين

مرتضى خلف الله البشير

كلية العلوم الرياضية والحاسوب, جامعة الجزيرة, ود مدنى , السودان

murtadabashir@yahoo.com

التفاف بيتا هو عنصر مهم جداً في بنية البروتين ويلعب دوراً هاماً في تكوين البروتين ووظيفته. هناك عدة طرق تم تطويرها للتنبؤ بالتفاف بيتا من سلسة البروتين. وتعتمد أفضل الطرق على الشبكات العصبية (NNS) أوالات المتجه الداعم (SVMs). وعلى الرغم من أن الانحدار اللوجستى المبنى على الكيرنال (KLR) هو أسلوب تصنيف قوى وتم تطبيقه بنجاح في العديد من مشاكل التصنيف، ولكنه لم يستخدم في التنبؤ أو تصنيف التفاف بيتا ، وذلك لأنه يحتاج الي ذاكرة حاسوب كبيرة ومعالج ذي سرعة عالية. هنالك تطبيق تقريبي سريع ودقيق للانحدار اللوجستى المبنى على الكيرنال يدعى الانحدار اللوجستى المبنى على الكيرنال ثابت الحجم (FS-KLR) يمكن أن يستخدم في البيانات ذات الحجم الكبير. ويستخدم أسلوب الثقة لمنطقة نيوتن للانحدار اللوجستى (LR) للبيانات الكبيرة كأساس لحل مشكلة التقريب، وطريقة نيستروم لتقريب مصفوفة الصفات. في هذه الورقة قمنا بإستخدام الانحدار اللوجستى المبنى على الكيرنال ثابت الحجم (FS-KLR) للتنبؤ بالتفاف بيتا وتم مقارنة النتائج التي تم الحصول عليها مع تلك التي حصلت باستخدام آلة المتجه الداعم (SVM) . معلومات البنية الثانوية للبروتين و مصفوفات PSSMs استخدمت كصفات مدخلة للطريقتين. الاداء الذى تحصلنا عليه بإستخدام الانحدار اللوجستى المبنى على الكيرنال ثابت الحجم كان مقارنا بالذي تحصلنا عليه بإستخدام الة المتجه الداعم (SVM). الانحدار اللوجستى المبنى على الكيرنال ثابت الحجم لديه ميزة انتاج مخرجات احتمالية مباشرة وامتداده ليغطي حالات اصناف متعددة معرف بصورة جيدة بالإضافة الى أن زمن تنفيذه أقل من الزمن المطلوب لتنفيذ الة المتجه الداعم (SVM).