Malaysian Journal of Mathematical Sciences 2(2): 185-194 (2008)

Prediction of β – Turns Using Global Adaptive Techniques from Multiple Alignments in Neural Network

Zarita Zainuddin, Chan Siow Cheng and Lye Weng Kit
School of Mathematical Sciences, University Sains Malaysia
Email: zarita@cs.usm.my

ABSTRACT

A neural network-based method has been developed for the prediction of $\beta-$turns in proteins by using multiple sequence alignment. A feed-forward network with a single hidden layer is used where the sequence structure network is trained with multiple sequence alignment in the form of position-specific scoring matrices (PSSM). This paper concentrates on global adaptive techniques: Conjugate Gradient methods (Fletcher Reeves, Polak Ribiere and Powell Beale), Preconditioned Conjugate Gradient methods (Preconditioned Fletcher Reeves, Preconditioned Polak Ribiere and Preconditioned Powell Beale) and Levenberg Marquardt method in the training of multilayer perceptrons (MLP) neural network. The behavior of these training methods in the present study is reported. The Levenberg Marquardt method had been proved to be the most effective when tested and compared to other methods. It only takes 11 iterations to converge and yields overall performance, Q_{total} = 96.67%. Then Preconditioned Fletcher Reeves, Preconditioned Polak Ribiere and Preconditioned Powell Beale methods yield the same result of Q_{total} which is 93.33% with 333, 404 and 354 epochs accordingly. Lastly, Fletcher Reeves method yields the overall prediction accuracy of 93.33% after 689 epochs. On the other hand, Polak Ribiere and Powell Beale methods yield the same result of Q_{total} which is 90% with 531 and 468 iterations respectively.

Keyword: $\beta-$turns, neural network, prediction, secondary structure, multiple alignment protein

INTRODUCTION

The prediction of protein structure is a crucial topic in computational biology. One of the ultimate objectives of computational biology is reliable prediction of the three-dimensional structure of proteins due to the difficulty in obtaining high resolution crystallographic data (Cheng *et al*, 2005). An intermediate step, secondary structure prediction, has been used to simplify the complicated problems of protein 3D structure prediction.

The secondary structure can be classified into α helices, β sheets, loops and tight turns. The α helices and β sheets are two classic elements which play a significant role in complex macromolecular architecture during the last two decades. Hence, many methods for predicting these two elements have been developed rapidly. In contrast, with the α helices and β

sheets, just a few methods have been suggested for predicting the tight turns. Even the researchers did not put the attention focused on tight turns prediction but from both the structural and functional point of view, tight turns also play an important role in protein. Tight turns are usually found near the surface of proteins and this will help to recognize molecule between proteins. A polypeptide chain cannot fold into a compact globular structure by the absence of tight turns. Moreover, useful information about the definition of template structure for defining the new molecules is provided by tight turns (Chou, 2000).

In this paper we just focused on prediction of $\beta-\text{turns}$, one of the elements of tight turns. A neural network-based method is developed to predict the $\beta-\text{turns}$ in proteins by using multiple sequence alignment. Global adaptive techniques in particular, Conjugate Gradient method, Preconditioned Conjugate Gradient method and Levenberg Marquardt method, are proposed in training of multilayer perceptrons. The overall performances of the methods are investigated.

$\beta-$Turns

Tight turns are irregular structures with nonrepeating backbone torsion angles and often have at least one hydrogen bond. Tight turns are generally categorized as δ-turn, γ-turn, β-turn, α-turn and π-turn, which are formed by two, three, four, five and six amino acid residues respectively (Chou, 2000). A β-turn consists of 4 consecutive residues defined by position i, i+1, i+2, i+3 which are not present in alpha-helix. The distance between Calpha(i) and Calpha(i+3) is less than 7Å and the residues involved are not helical (Richardson, 1981). They are responsible for the compact globular shape of proteins because of the ability to reverse the protein chain direction within a span of several residues. On average, the β-turns constitute about 25% of the residues in globular proteins. β-turn formation is also an important stage in protein folding. Moreover, the occurrence of β-turns on solvent-exposed surfaces makes them a suitable candidate for molecular recognition processes and interactions between peptide substrates and receptors.

Multiple sequence alignment

Multiple sequence alignment is an alignment of n>2 sequences obtained by inserting gaps ('-') into sequences such that the resulting sequences have all lengths, L, and can be arranged in matrix of N rows and L columns where each column represents a homologous position. The basic information from a multiple alignment of protein sequences is the position and nature of the

conserved regions in each member of the group. Conserved sequence regions correspond to functionally and structurally important parts of the protein. Sequences can be aligned across their entire length (global alignment) or only in certain regions (local alignment). Global alignments need to use gaps (representing insertions/deletions) while local alignments can avoid them by aligning regions between gaps.

Global Adaptive Techniques

In this study, global adaptive techniques will be adopted in the training of multilayer perceptrons (MLP). These techniques use global knowledge of the state of the entire network, such as the direction of the overall weight update vector. Conjugate Gradient methods (Fletcher Reeves, Polak Ribiere and Powell Beale), Preconditioned Conjugate Gradient methods (preconditioned Fletcher Reeves, preconditioned Polak Ribiere and preconditioned Powell Beale) and Levenberg Marquardt method are investigated.

Conjugate Gradient (CG) Methods

The CG method is an algorithm for finding the nearest local minimum of a function of n variables based on the premise that the gradient of the function can be computed. It uses conjugate direction instead of local gradient for going downhill. If the vicinity of the minimum has the shape of a long, narrow valley, the minimum is reached in far fewer steps than would be the case when using the steepest descent method. The method proceeds by generating vector sequences of iterates, residuals corresponding to the iterates, and search directions used in updating the iterates and their residuals. Although the length of these sequences can become large, only a small number of vectors need to be kept in memory.

In most CG algorithms, the step size is adjusted at each iteration. A search is made along the conjugate gradient direction to determine the step size that minimizes the performance function along that line in order to reduce the number of iterations. The modifications of the CG algorithm are discussed in relation to a quadratic function for the case in which the Hessian is unknown but in which the objective function values and gradients are available. The next search direction $p^{(i+1)}$ is a linear combination of $r^{(i+1)}$ and $p^{(i)}$, $p^{(i+1)} = r^{(i+1)} + \beta^{(i)} p^{(i)}$. The modifications are all based on the algebraic

manipulation of the formula β_i in such a way that A is eliminated. We discuss three well-known modifications:

i. Fletcher-Reeves Update (Flecher & Reeves, 1964)

$$\beta^{(i)} = \frac{\left(g^{(i)}\right)^T g^{(i)}}{\left(g^{(i-1)}\right)^T g^{(i-1)}}$$

ii. Polak-Ribiere (1971) Update (Polak, 1971)

$$\beta^{(i)} = \frac{\left(\Delta g^{(i-1)}\right)^T g^{(i)}}{\left(g^{(i-1)}\right)^T g^{(i-1)}}$$

iii. Powell-Beale Restarts

One such reset method was proposed by (Powell, 1977), based on an earlier version proposed by Beale. This technique restarts if there is very little orthogonally left between the current gradient and the previous gradient. This is tested with the following inequality:

$$\left\left(g^{(i-1)}\right)^T g^{(i)}\right\| \geq 0.2\left\|g^{(i)}\right\|^2$$

If this condition is satisfied, the search direction is reset to the negative of the gradient.

Preconditioned Conjugate Gradient (PCG) Methods

Lack of robustness is a widely recognized weakness of iterative solutions, relative to direct solutions. Both the efficiency and robustness of iterative techniques can be improved by preconditioning. Preconditioning is simply a means of transforming the original linear system into one which has the same solution, but which is likely to be easier to solve with an iterative solution. It is used to decrease the time needed to solve a linear system with an iterative method (Evans, 1994). In general, the reliability of iterative techniques, when dealing with various applications, depends much on the quality of the preconditioner. The convergence rate of iterative methods depends on the spectral properties of the coefficient matrix. Hence, one may attempt to transform the linear system into one that is equivalent to the

original in the sense that it has the same solution, but that has more favorable spectral properties.

A preconditioned system can be written as

$$M^{-1}Ax = M^{-1}b$$

in which the system possesses the same solution as the original, where the matrix M is called the preconditioning matrix or preconditioner. Matrix, A, is symmetric and positive definite. The search direction for each iteration is determined by

$$p^{(i)} = M^{-1} \cdot \left(-g^{(i)}\right) + \beta^{(i)} p^{(i-1)}$$

The manner in which the constant $\beta^{(i)}$ is computed distinguishes the various versions of the Preconditioned Conjugate Gradient algorithm.

i. For the preconditioned Fletcher-Reeves method, the procedure is

$$\beta^{(i)} = \frac{\left(g^{(i)}\right)^T M^{-1}\left(g^{(i)}\right)}{\left(g^{(i-1)}\right)^T M^{-1}\left(g^{(i-1)}\right)}$$

ii. For the preconditioned Polak-Ribiere method, the procedure is

$$\beta^{(i)} = \frac{\left(\Delta g^{(i-1)}\right)^T M^{-1}\left(g^{(i)}\right)}{\left(g^{(i-1)}\right)^T M^{-1}\left(g^{(i-1)}\right)}$$

iii. The preconditioned Powell-Beale Restart Procedure method can be improved by periodically resetting the search direction to the negative of the gradient,

$$p^{(i+1)} = M^{-1} \cdot \left(-g^{(i+1)}\right)$$

For this technique, we will restart if there is very little orthogonally left between the current gradient and the previous gradient. This is tested with the following inequality

$$\left\| \left(g^{(i)}\right)^T \cdot M^{-1} \cdot g^{(i+1)} \right\| \geq 0.2 \left\| M^{-1} \cdot g^{(i+1)} \right\|^2$$

If this condition is satisfied, the search direction is reset to the negative of the gradient.

Levenberg-Marquardt (LM) Method

The LM algorithm is the most widely used optimization algorithm. It outperforms simple gradient descent and other conjugate gradient methods in the solution of a wide variety of problems (Hagan et al., 1996). The LM algorithm is an iterative technique that locates the minimum of a multivariate function that is expressed as the sum of squares of non-linear real valued functions. It has become a standard technique for solving non-linear least-squares problems, widely adopted in a board spectrum of disciplines. The LM algorithm provides a nice and optimal comprise between the speed of Newton's method and the guaranteed convergence of the Steepest Descent method. When the current solution is far from the correct one, the algorithm behaves like the Steepest Descent method, slow, yet guaranteed to converge. When the current solution is close to the correct solution, it mimics the Gauss-Newton method.

Neural Network Architectures

The neural network that was used for this research is the feed-forward neural network. Multilayer perceptrons (MLPs) which are capable of finding solutions on a much wider range of problems were used in this paper.

Training a feed forward network is an iterative process that involves repeatedly presenting the training set (containing exemplar patterns with known target outputs) to the network. After each presentation, the network parameters (or weights) are adjusted so that the networks' total error for all patterns in the set (as measured by an appropriate error function) is progressively reduced. This type of training is known as supervised learning and the algorithm for adjusting the network weights is the training method. We choose more advanced training methods i.e. the CG, PCG and LM methods as they frequently require far fewer training iterations than backpropagation.

The data set was divided into the training set and the testing set. The training set consisted of all 90 known sequences which were selected from the PDB (Protein Data Bank). There were 45 $\beta - \text{turns}$ sequences and 45 $\text{non} - \beta - \text{turns}$ sequences. The testing set comprised of 30 unknown sequences which are chosen randomly from the PDB too. The target outputs always consisted of two binary numbers, i.e. [1, 0] or [0, 1] for all patterns in the training set used in this research. If the sequence was a $\beta - \text{turns}$, the target output was set at [1, 0] and classified into category 1, otherwise, it was set at [0, 1] and classified into category 0. A data training set is a preliminary process.

Position specific scoring matrice (PSSM) is used as a direct input to the neural network. The matrix has 20 rows X 10 columns. 20 rows represent the 20 amino acids while the 10 columns represent the length of the target sequence. Each element represents the frequency of the occurrence of each 20 amino acid at a particular position in the alignmnent.

RESULTS AND DICUSSIONS

The straightforward measure of a method's overall performance of Q_{total} (Shepherd et al. 1999), the percentage of correctly classified residues, is given as

$$Q_{total} = \left(\frac{p+n}{l} \right) \times 100\%$$

where p = number of correctly-classified $\beta - \text{turns}$ sequences

n = number of correctly-classified $\text{non} - \beta - \text{turns}$ sequences

l = total number of sequences.

The regression results for the global adaptive techniques are shown in Table 1 and the simulation results for the overall performance of Q_{total} are shown in Table 2.

TABLE 1: The regression result for global adaptive techniques

Algorithm	***Slope***	***y-intercept***	***Correlation coefficient***
Fletcher Reeves	0.833	8.11e-17	0.866
Polak Ribiere	0.769	2.43e-16	0.809
Powell Beale	0.766	0.0526	0.783
Preconditioned Fletcher Reeves	0.833	8.11e-17	0.866
Preconditioned Polak Ribiere	0.833	8.11e-17	0.866
Preconditioned Bowell Beale	0.833	8.11e-17	0.866
Levenberg Marquardt	0.929	0	0.929

TABLE 2: The simulation results for the overall performance of Q_{total}

Algorithm	**Epochs**	**Total correct pattern (over 30 sequences)**	**Overall performance, Q**
Fletcher Reeves	689	28	93.33
Polak Ribiere	531	27	90.00
Powell Beale	468	27	90.00
Preconditioned Fletcher Reeves	333	28	93.33
Preconditioned Polak Ribiere	404	28	93.33
Preconditioned Bowell Beale	354	28	93.33
Levenberg Marquardt	11	29	96.67

From the results obtained, the Levenberg Marquardt proved to be the most effective, when tested and compared to the other methods. It only took 11 iterations to converge and yielded an overall performance, Q_{total} of 96.67%. Preconditioned Fletcher Reeves, Preconditioned Polak Ribiere, Preconditioned Bowell methods yielded a similar Q_{total} of 93.33% taking 333, 404 and 354 epochs respectively to converge. Lastly, the Conjugate Gradient methods demonstrated the slowest learning speed when compared to other methods. In fact, the Fletcher Reeves yielded the overall prediction accuracy of 93.33% after 689 epochs. On the other hand, Polak Ribiere and Powell Beale methods yielded the same Q_{total} of 90% with 531 and 468 iterations respectively.

An evaluation of the networks on the testing sets revealed that the Levenberg Marquardt method obtained the best correlation coefficient, R, of closely to 1. This indicates that the corresponding targets explained the variation in the network outputs very well.

CONCLUSION

The numerical evidence shows that PCG and LM methods do perform better than CG methods. The PCG methods with the use of a preconditioner unlike the normal CG methods succeeded in lowering the condition number of the input matrix by clustering the eigen values. Due to this property, it can be surmised that PCG algorithms based on the preconditioned can achieve higher convergence rates than CG methods without getting stuck at local minima as CG methods did. Overall, the results show that LM has the best performance in the training of MLP as it has the highest convergence accuracy and speed. Thus, it can be postulated that the LM algorithm can provide a good compromise between the speed of Newton's method and the guaranteed convergence of the Steepest Descent technique.

REFERENCES

Cheng, H., Sen, T. Z., Kloczkowski, A. & Jernigan, R. L. 2005. Prediction of protein structure by mining structural fragment database. *Polymer*. **46**, 4314-4321.

Richardson, J.S. 1981. The anatomy and taxonomy of protein structure. *Adv. Protein Chem.* **34**, 167-339.

Chou, K. C. 2000. Prediction of tight turns and their types in proteins. *Analytical Biochemistry*. **286**(16), 1-16.

Fletcher, R. & Reeves, C. M. 1964. Function minimization by conjugate gradients. *Computer Journal*. **7**, 149-154.

Polak, E. 1971. *Compuatational methods in optimization*, New York: Academic press.

Powell, M. J. D. 1977. Restart procedures for the conjugate gradient method. *Mathematical Programming*. **12**, 241-254.

Henikoff, J. G. and Henikoff, S. 1996. Using substitution probabilities to improve position-specific scoring matrices. *Comput. Appl. Biosci*. **12**(2), 135-143.

Tatusov, R. L., Altschul, S. F. and Koonin, E. V. 1994. Detection of conserved segments in proteins: iterative scanning of sequence databases with alignment blocks. *Proc Natl Acad Sci*. **91**(25), 12091-12095.

Lawrence, C. E., Altschul, S. F. and Boguski. 1993. Detecting subtle sequence signals a gibbs sampling strategy for multiple alignment. *Journal of Science*. **262**, 208-214.

Evans, D. J. 1994. *Preconditioned Iterative Methods*. Switzerland: Gordon and Breach Science Publisher.

Hagan, M.T., Demmh, H. B. and Beale, M. 1996. *Neural Network Design*. Singapore: Thomson Learning.