International Journal of Fuzzy Logic and Intelligent Systems, vol. 10, no. 4, December 2010, pp. 314-318
DOI : 10.5391/IJFIS.2010.10.4.314

Protein Secondary Structure Prediction using Multiple Neural Network Likelihood Models

Seong-gon Kim* and Yong-Gi Kim**

***CISE department, University of Florida, USA**
****Dept of Computer Science, Gyeongsang National University, Korea**
****Corresponding author: ygkim@gnu.ac.kr**

Abstract

Predicting Alpha-helicies, Beta-sheets and Turns of a proteins secondary structure is a complex non-linear task that has been approached by several techniques such as Neural Networks, Genetic Algorithms, Decision Trees and other statistical or heuristic methods. This project introduces a new machine learning method by combining Bayesian Inference with offline trained Multilayered Perceptron (MLP) models as the likelihood for secondary structure prediction of proteins. With varying window sizes of neighboring amino acid information, the information is extracted and passed back and forth between the Neural Net and the Bayesian Inference process until the posterior probability of the secondary structure converges.

Key Words : Bayesian inference, Likelihood model, Neural network, Protein secondary structure

1. Introduction

Machine learning approaches are best suited for areas where there are large amounts of data with little theory behind the correlation of the data. This situation particularly applies to the prediction of protein secondary structures, where there is clearly a relationship between the sequences of amino acids and the resulting structure, but little explanation as to why. Especially in modern days, the advancements in technology has provided vast amounts of protein secondary structure data, yet much is still unknown about the underlying biological mechanism of the structures.

Among the more popular machine learning methods used in this area are Artificial Neural Networks (ANN)[1,2] and Bayesian Inference methods. The ANN has the ability to take in large amounts of data and extract the relational information to approximate a model that closely fits the actual model that ties the data together, even through reasonable amount of missing data and noise. There has been a lot of work on predicting secondary structure using ANNs [3,4,5,6,7,8,9,10], sometimes in combination with other methods [11,12,13], and some of the of the more recent works include the PSI-BLAST[14] or PSIPRED[9,15] method that are still used today. PSIPRED is important in that it has so far been known to return the most accurate results in predicting secondary structures using protein profiles. Other methods that use protein profiles for structure prediction include PPRODO[16], SVMpsi[17,18], fragment assembly methods[19] and profile-profile alignment methods[20,21]. Using the same input vectors as PSIPRED, the nearest neighbor method is also used such as by PREDICT[22,23] to by step the training process as well.

Although there has been relatively good success with the ANN, there is much space to improve yet there is seemingly a limit to how much it actually has in recent times. This is mostly due to the rigid structure of the ANN that does not allow changes in the input as more information is extracted, as well as the explosion of free parameters that have to be computed as the input size grows.

Bayesian Inference, on the other hand, does not directly generate models like the ANN. It only concerns itself with assessing the value of the models with respect the available data and information. With a strong foundation in probability theory, Bayesian Inference provides a principled and rigorous approach to inference in situations of uncertainty such as predicting secondary structures. The main drawback is that asides being computationally intensive, the Bayesian Inference method can be overly sensitive to the likelihood model or prior probability set that it uses. Another issue is that fact that the Bayesian method assumes independency between neighboring amino acids[6].

2. Overview of the Method

The approach this paper makes is to use ANNs within a Bayesian framework to exploit the advantages of both methods while eliminating the drawbacks. Despite the rigid and non-dynamic structure of an ANN, we can use multiple ANN models for different sequence lengths of amino acids while assessing the value of each model within the Bayesian framework. This process is shown in [Fig. 1].

By using the ANNs as likelihood models, we can alternate between models of different window sizes and varying degrees of neighboring information to take advantage strong architectural diversity[10]. For each iteration in the Bayesian

Manuscript received Apr. 5, 2010; revised Sep. 24, 2010;
Accepted Sep. 30, 2010

process, the previous probability, or prior, of a particular amino acid being either an Alpha-helix, Beta-sheet, or Turn will be updated according to an ANN models outputs that also takes the prior into account as well. This leads to a convergence of probabilities that tell us which structure the amino acid in question most likely is.

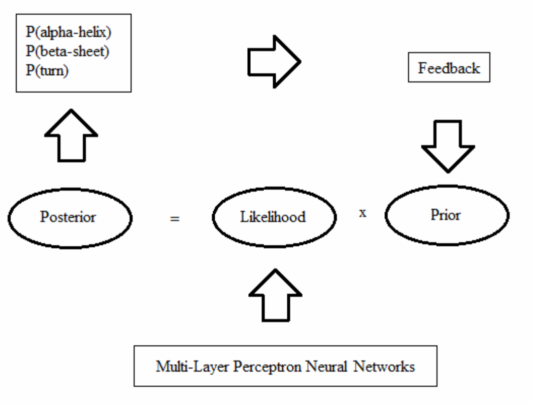

Fig. 1 Process of calculating the likelihood probability

Artificial Neural Networks are used to calculate the likelihood probability. The outputs are normalized and multiplied with the prior probabilities to find the posterior probability of each secondary structure. These probabilities are used as priors in the next iteration.

2.1 Multilayered Perceptron Neural Net Likelihood Model

We start by explaining the ANN architecture that will be used within the Bayesian framework. ANNs were originally developed to model the learning and processing mechanisms of the brain[1,2]. Although it is now known that the ANN does not accurately represent the workings of the brain, its ability to approximate non-linear functions in high dimensions proves ANN to be a valuable tool in searching for models to reasonably fit large amounts of data. The algorithm introduced in this paper uses one of the more simple types of networks – a standard 1 hidden layered fully connected Multilayered Perceptron (MLP) with sigmoid activation functions.

We first concern ourselves with the encoding of the sequence input and the interpretation of the output. Despite the initial intuition that encoding the sequence as input using physical and chemical features of potential relevance would outperform a more direct encoding scheme, due to the nature of the MLP, most of the information is discarded before reaching the output level. This algorithm uses a slight variation of the *orthogonal encoding scheme* where an amino acid would be encoded 20 nodes. Each of the nodes represents the 20 amino acids, while an input of all zeroes represents a terminal position within a window. Although this sparse encoding scheme has the disadvantage of being wasteful while increasing the number of network connections, it does not lose any data during the encoding process and is easier for the MLP to work with. A secondary structure was encoded with 3 nodes in a similar fashion, with each node representing an Alpha, Beta, or Turn structure. These encoding schemes were used to take a set number of neighboring amino acids within a certain window size as input along with each of their secondary structure, and the true secondary structure of the amino acid in question. The output is the amino acid in question, also represented by the orthogonal encoding scheme. Although the outputs do not represent the conditional probability of a secondary structure given the neighboring amino acid information, they are a representation of this probability with respect to the inputs, so the outputs are later normalized to be used as the likelihood probability within the Bayesian framework. The final architecture of the ANN with a window size of 3 is shown in [Fig. 2].

The next issue is deciding on the number of models and the different window sizes. It is known that large windows perform better in longer sequences since more information is used, while short windows perform better in shorter sequences of amino acids since much of the unneeded neighboring information is left out and the inputs become less noisy[10]. Previous work with similar architectures[24,4] have shown that a window size of 13 to 15 achieves the best performance. However, within the context of the Bayesian Inference process, we will also have to take into consideration the fact that the MLP is using prior probabilities as inputs as well. This means the algorithm will have to take in less neighboring prior information than the optimal window size while the Bayesian process is still converging. Assuming that 13 is the optimal size, we choose two smaller window sizes of 3 and 7, and so we have a total of three MLP models with window sizes W = {3, 7, 13}.

Finally, through experimentation, about 40 hidden nodes were shown to perform the best for all three MLP models.

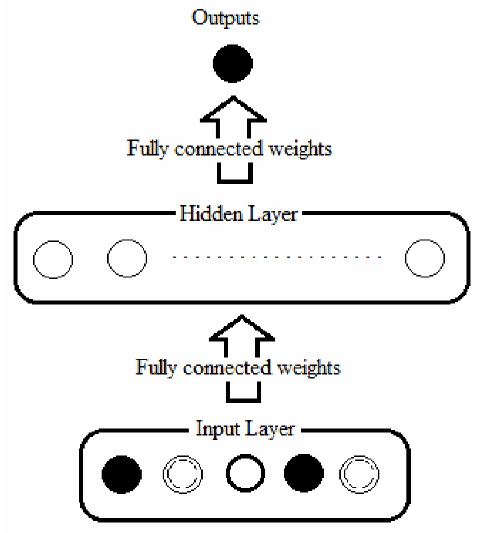

Fig. 2 An ANN with a window size of 3
N with a window size of 3

In [Fig.2], An ANN with a window size of 3 is shown. The solid colored inputs or output are each a group of 20 nodes representing the neighboring amino acids. The double-outlined inputs are each a group of 3 nodes representing the neighboring secondary structures. The single input with a bold outline represents a group of 3 input nodes representing the secondary structure of the amino acid in question. Normalizing the outputs gives us the likelihood probability.

2.2 Bayesian Inference Scheme

Now that the MLP models have been set, we turn to the overall Bayesian framework that the MLPs will run in. The Bayesian equation that is used is shown below.

$$P(D|H) = P(H|D) * P(D)$$

$$\text{Posterior} = \text{likelihood} * \text{prior}$$

The reason the denominator is missing from the RHS of the Bayes equation is because its only function is to act as a normalization factor for the LHS of the equation. Within the Bayesian Inference process, each time a set of probabilities are found, they are normalized before the next iteration so the denominator is not required.

Each MLP model serves as the likelihood model while a 3 x *n* table of probabilities are kept to serve as the posterior and prior, where *n* = the length of the sequence and each row contains the probabilities of the amino acid being one of the 3 Alpha-helix, Beta-sheet or Turn structure. Starting with the first amino acid in the sequence, the algorithm runs through entire sequence one letter at a time, extracting the neighboring amino acid sequence information according to the set window size of the particular MLP as well as each of their prior probabilities from the probability table. This information is the used as input for the Feedforward algorithm and the output is normalized, multiplied to each respective prior, then set in a posterior probability table. Once the algorithm runs through the entire sequence, the posterior probability table is used as the prior probability table for the next iteration. This process is visually shown in [Fig. 3].

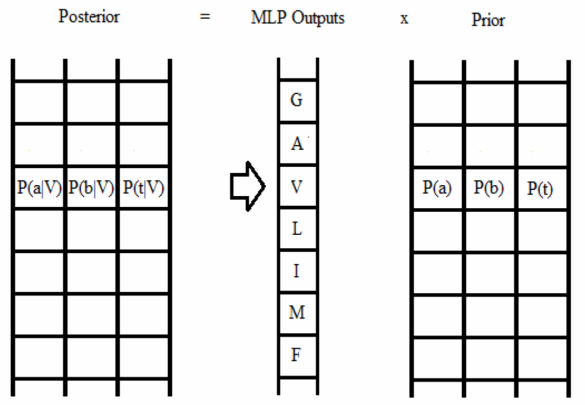

Fig. 3 Process of Bayesian Inference Scheme

In [Fig. 3], the arrow represents the amino acid the MLP is running on. Alpha, Beta, and Turn structures are represented as lowercase a, b, and t, respectively. P(S) is the probability of being secondary structure S and P(S|V) is the conditional probability of being secondary structure S given that the amino acid is V, where S = {a, b, t}.

The algorithm alternates between MLP models of different window sizes for the iterations and this process continues until there is convergence. Experimentation shows that the order in which the MLPs of different window sizes are used was not important since the direction in which the probabilities converge stays the same in each case and so the Bayesian process effectively eliminates the ordering dependencies.

One final issue with this Bayesian process is the sensitivity to the initial prior probabilities. It is shown that it is possible to get structure information from position specific scoring matrices for machine learning procceses[9]. Although these scoring matrices are not actual probabilities, like the outputs for the MLP, they contain information that represents the structural probabilities according to the position of each amino acid. The prior probability table used in this case was initialized to the normalized Chou-Fasman parameters[25].

Finally, the amino acid is classified as the secondary structure with the highest probability from the posterior table.

3. Experimentation and Analysis

Prior to running the algorithm, the MLP models had to be trained offline. 50 proteins selected from the Protein Data Bank (PDB)[26] were used as the training data. The training data was also carefully chosen to include the same number of proteins from each of the five classes; 10 random proteins each from classes of all Alpha, all Beta, Alpha and parallel Beta-sheets, Alpha and anti-parallel Beta-sheets, and multi-domain proteins. 10 different proteins, 2 from each of the five classes, were also used as a validation set. Holley and Karplus[4] had achieved an overall predictive accuracy of 63.2% using an MLP model with the same encoding techniques to predict secondary structures, and a similar result was attained with these MLP models in predicting the amino acids. The Bayesian Inference process was used with the three MLP models on 5 validation sets, each consisting of 2 proteins from each of the 5 classes. [Tab. 1] and [Tab. 2] show the results using the validation sets of all Alpha, all Beta, Alpha and parallel Beta-sheets, Alpha and anti-parallel Beta-sheets, and multi-domain proteins.

From [Tab. 1] we can see that the accuracy predicting proteins of mostly single secondary structures was higher than that of mixed secondary structures. Using multiple window size MLPs did show its advantages, and not only were these results significantly better compared to that of using a single MLP for structure prediction as reported by Holley and Karplus[4], but the algorithm did reasonably well even against the proteins of mixed structures. Using multiple MLPs allowed the algorithm to take into account not only neighboring information that were nearby, but also the information that are faraway on the sequence yet actually geometrically close when the proteins are folded in reality.

Table 1. The results using the validation sets

CLASS	Alpha			Beta			Turn			Accuracy
	a	b	t	a	b	t	a	b	t	
A	284	26	61	36	95	47	23	14	68	.6834
B	129	27	39	42	263	21	19	32	72	.7205
A/B	156	29	89	57	87	32	28	9	76	.5666
A+B	103	9	29	52	67	47	20	11	67	.5852
A, B	88	9	60	67	97	52	11	5	42	.5267

In [Tab. 1], the rows are the different classes the proteins were selected from. All Alpha, all Beta, Alpha and parallel Beta-sheets, Alpha and anti-parallel Beta-sheets, and multi-domain proteins are represented as A, B, A/B, A + B, A, B, respectively. Each column shows the predicted structures a, b, and t, for the true structures Alpha, Beta and Turn.

Table 2. The accuracy of prediction each of the three structures

	Alpha	Beta	Turn
Accuracy	.6678	.5734	.6539

Comparing the prediction results of the mixed structure proteins, it is also clear that multi domain proteins were harder to predict than the proteins with all parallel or anti-parallel beta-sheets. Although this is intuitive as multi-domain proteins have a more complex underlying structure, it also signifies that the sequencing and parallelism of the beta-sheets does play a role in deciding the secondary structure of the amino acids. This is also confirmed by the fact that the accuracy of Alpha-helices is significantly higher than the Beta-sheets in [Tab. 2], and this correlation of data and the parameters also gives us some insight into the underlying biological mechanism that decides the secondary structures of proteins.

In [Tab. 2], it can be clearly seen that the Turn predictions have a high prediction accuracy compared to the Beta-sheet as well. This was most likely due to the usage of orthogonal encoding of the amino acids for input. Since the neighboring amino acid MLP inputs for the portion of the window that falls off the sequence is set to all zero and since Turns usually occur at the end of protein sequences, the MLPs easily detected and predicted the Turns near the end of the sequences by the presence of the zeros as input.

Overall, a comparison with the 63.2% accuracy that Holley and Karplus[4] had previously achieved shows that an MLP used within a Bayesian framework does significantly better at a maximum 72.05% accuracy. This is nearly a 9% increase in the best accuracy results which also show that standard Neural Network methods can easily be enhanced through a Bayesian scheme.

4. Conclusion and Future work

An algorithm to predict protein secondary structures using Bayesian Inference and MLPs was presented. The algorithm took advantage of using multiple MLPs to estimate the likelihood probability for the Bayesian Inference process and as a result, more information could be extracted for the prediction. This led the algorithm to produce better results than using a single MLP and shows potentials for further investigation.

Only simple inputs, namely the neighboring amino acids and secondary structures, were considered in this experiment but showed promising results. By providing more information to the MLP models such as the parallelism of the Beta-sheets, the geometrical angles, and the amino acids, or more experiments using various Neural Network models could provide more insight into the relationship between the amino acids and the secondary structures they form, as well as an improvement in prediction accuracy.

References

[1] Bishop, C.M., *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[2] Haykin, S., *Neural Networks and Learning Machines*, Third Edition, Pearson Inc, ISBN-10:0-13-147139-2, 2008.

[3] Bohr, H., Bohr, J., Brunak, S., Cotterill, R.M.J., Lautrup, B., Nørskov, L., Olsen, O.H., and Petersen, S. B., "Protein secondary structures and homology by neural networks: The α-helices in rhodopsin", *FEBS Letters,* 241, pp.223–228, 1988.

[4] Holley, L.H., and Karplus, M., "Protein secondary structure prediction with a neural network", *Proc. Nat. Acad. Sci.* USA, 86, pp.152–156, 1989.

[5] Kneller, D.G., Cohen, F.E., and Langridge, R., "Improvements in protein secondary structure prediction by an enhanced neural network", *J. Mol. Biol.*, 214, pp.171–182, 1990.

[6] Stolorz, P., Lapedes, A., and Xia, Y., "Predicting protein secondary structure using neural net and statistical methods", *J. Mol. Biol.,* 225, pp.363–377, 1992.

[7] Rost B., and Sander, C., "Improved prediction of protein secondary structure by use of sequence profiles and neural networks", *Proc. Nat. Acad. Sci. USA*, 90, pp.7558–7562, 1993.

[8] Rost B., and Sander, C., "Prediction of protein secondary structure at better than 70% accuracy", *J. Mol. Biol.,* 232, pp.584–599, 1993.

[9] Jones, D.T., "Protein secondary structure prediction based on position-specific scoring matrices", *J. Mol. Biol.,* 292, pp.195–202, 1999.

[10] Petersen, T.N., Lundegaard, C., Nielsen, M., Bohr, H., Bohr, J., Brunak, S., Gippert, G.P., and Lund, O., "Prediction of protein secondary structure at 80% accuracy", *Proteins,* 41, pp.17–20, 2000.

[11] Zhang, X., Mesirov, J., and Waltz, D., "Hybrid system for protein secondary structure prediction", *J. Mol. Biol.,* 225, pp.1049–1063, 1992.

[12] Maclin, R., and Shavlik, J., "Using knowledge-based neural networks to improve algorithms: Refining the Chou–Fasman algorithm for protein folding", *Machine Learning,* 11, pp.195–215, 1993.

[13] Riis, S.K., and Krogh, A., "Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments", *J. Comput.Biol.,* 3, pp.163–183, 1996.

[14] Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., and Lipman, L.J., "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", *Nucl. Acids Res.,* 25:3389–3402, 1997.

[15] McGuffin, L.J., Bryson, K., and Jones, J.T., "The PSIPRED protein structure prediction server", *Bioinformatics,* 16, pp.404–405, 2000.

[16] Sim, J., Kim, S.-Y., and Lee, J., "PPRODO: prediction of protein domain boundaries using neural network", *Proteins: Structure, Function, and Bioinformatics* 59, pp. 627-632, 2005.

[17] Kim, H., and Park, H., "Protein secondary structure prediction based on an improved support vector machines approach", *Protein Eng.* 16, pp.553-560, 2003.

[18] Kim, H., and Park, H., "Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3D local descriptor", *Proteins: Structure, Function, and Bioinformatics* 54, pp.557-562, 2004.

[19] Lee, J., Kim, S.-Y., Joo, K., Kim, I., and Lee, J., "Prediction of protein tertiary structure using PROFESY, a novel method based on fragment assembly and conformational space annealing", *Proteins: Structure, Function, and Bioinformatics* 56, pp.704-714, 2004.

[20] Ginalski, K., et al., "ORFeus: detection of distant homology using sequence profiles and predicted secondary structure," *Nucleic Acids Res.* 31, pp.3804-3807, 2003.

[21] Sim, J., Kim, S.-Y., Lee, J., and Yoo, A., "Predicting the threedimensional structures of proteins: combined alignment approach", *J. Korean Phys. Soc.* 44, pp.611-616, 2004.

[22] Joo, K., Lee, J., Kim, S.-Y., Kim, I., and Lee, S.J., "Profile-based nearest neighbor method for pattern recognition", *J. Korean Phys. Soc.* 44, pp.599-604, 2004.

[23] Joo, K., Kim, I., Kim, S.-Y. Lee, J., and Lee, S.J., "Prediction of the secondary structures of proteins by using PREDICT, a nearest neighbor method on pattern space", *J. Korean Phys. Soc.* 45, pp.1441-1449, 2004.

[24] Qian, N., and Sejnowski, T.J., "Predicting the secondary structure of globular proteins using neural network models", *J. Mol. Biol.,* 202, pp.865–884, 1988.

[25] Chou, P.Y., and Fasman, G.D., "Prediction of the secondary structure of proteins from their amino acid sequence", *Adv. Enzymol. Relat. Areas Mol. Biol.,* 47, pp.45–148, 1978.

[26] Protein Data Bank (PDB): http://www.rcsb.or

Seong-Gon Kim received the B.S. degree in computer science from University of Illinois at Urbana-Champaign, Illinois, U.S.A in 2008, and the M.S. degree in computer science and engineering from University of Florida, Florida, U.S.A. in 2010. He is currently a Researcher/Engineer in LG Electronics User Platform Lab. His current research interests include machine learning, pattern recognition, and intelligent robotics.

Yong-Gi Kim received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea in 1978, the M.S. degree in computer science from University of Montana, U.S.A. in 1987, and the Ph.D. degrees in computer and information sciences from Florida State University, U.S.A. in 1991. He was a Visiting Scholar in the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Illinois, U.S.A., from 2008 to 2009. He is currently a Professor in the Department of Computer Science, Gyeongsang National University, Korea. His current research interests include soft computing, intelligent systems and autonomous underwater vehicle