

# Analyzing the Performance of Popular Methods used in Pattern Classification in the field of Bioinformatics

Fitzroy Nembhard

Morgan State University  
finem1@morgan.edu  
Bioinformatics 513, Fall 2011  
November 30, 2011

## ABSTRACT

Pattern classification is the organization of patterns into groups of patterns sharing the same set of properties. Methods of pattern classification are used to classify recurring patterns in data on the basis of either a priori knowledge or information directly extracted from the data. Thus, the resulting learning strategy is characterized as either supervised or unsupervised learning [4]. These methods can be applied to the field of Bioinformatics. Using secondary structure data obtained from JPred (a consensus secondary structure prediction server), we investigate the performance of the K-nearest neighbor algorithm, the projections method using SVD, Gradient Descent, and the BackPropagation network, which are popular methods used in pattern classification. We apply and observe the responses of these algorithms to three encoding techniques, namely the standard vector encoding, hydrophobic/hydrophilic charged alphabet, and the volume alphabet. These methods use different metrics such as Euclidian distances, machine learning, and matrix decomposition techniques to estimate decision boundaries. Our experiment shows that though the Backpropagation network is a universal approximator, it has to be trained properly using the right number of neurons to achieve optimal results. On the other hand, the projections method using SVD works well with high dimensional unique vectors, the k-nearest neighbor works differently given different values of  $k$ , and the gradient descent algorithm works better when the data follows a linear pattern. Conclusively, the choice of a

classifier depends on the kind of data to which it is being applied.

## 1. INTRODUCTION

The development and application of computational algorithms and techniques to analyze data regarding biological structures is known as structural bioinformatics. There is a major challenge in bioinformatics, which has to do with accurately predicting the occurrence of local secondary structures, namely alpha helices, beta sheets and coils. However, evolutionary information resulting from improved searches and larger databases has boosted prediction accuracy over the years [2]. Some of the methods widely used in Secondary Structure Prediction include Chou-Fasman, GOR (Garnier, Osgathorpe, and Robinson), Neural Network models, Nearest-neighbor methods and HMM (Homology Modeling Methods). Another method involves analyzing the X-ray diffraction patterns of crystallized proteins [3]. The more popular methods combine prediction techniques from several methods. In this paper, I will examine four methods used in pattern classification, which can also be applied to the secondary structure prediction problem. The methods are K-nearest neighbors (KNN), Projection using Singular Value Decomposition (SVD), Gradient Descent network, and a single-layer Back Propagation network.

## 2. METHODS

After the data was downloaded from the JPred server, we created a structure to house all sequences and their classification using the Matlab software. Using a 'sliding window' of length 17, we formulated a training set to be used as input data to each algorithm and network. We then encoded the secondary structure categorization of each amino acid (where  $H \Rightarrow$  alpha helix,  $E \Rightarrow$  beta sheet and  $_ \Rightarrow$  other) by taking into account the central position of each input sequence (i.e. the 9<sup>th</sup> letter). Using an identity matrix, 'E' is represented by the vector  $[1\ 0\ 0]^T$ ,  $H$  by  $[0\ 1\ 0]^T$  and '\_' is represented by  $[0\ 0\ 1]^T$ . Once this was done, we ran tests to determine the classification of unknown sequences using each of the methods below and calculated their performances based on their output.

### 2.1 K-nearest Neighbor

In pattern recognition, the k-nearest neighbor algorithm is a method for classifying objects based on closest training examples in the feature space. KNN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. KNN is one of the simplest of machine learning algorithms, where an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors ( $k$  is a positive integer, typically small). If  $k = 1$ , then the object is simply assigned to the class of its nearest neighbor. We ran our experiment using  $k=1$  and  $k=3$ . We computed the distance between each test vector and the known classification vector, and then assign a classification based on the class with the closest neighbors.

### 2.2 Projection using SVD

In linear algebra and functional analysis, a projection is a linear transformation  $P$  from a vector space to itself such that  $P^2 = P$ . It leaves its image unchanged. A projection matrix  $P$  is an  $n \times n$  square matrix that gives a vector space projection from  $\mathbb{R}^n$  to a subspace  $W$  [6]. The columns of  $P$  are the projections of the standard basis vectors, and  $W$  is the image of  $P$ . A square matrix  $P$  is a projection matrix iff  $P^2 = P$ .

Matrix decomposition techniques such as the Singular Value Decomposition can lead to the

construction of subspaces that can mathematically categorize subsets of sequences into families. The application of linear subspaces to achieve pattern classification consists of applying orthogonal projection operators based upon the training classes [5]. Given a  $n \times m$  matrix  $A$ , the SVD "decomposes"  $A$  into the following:

$$A_i = U_i \Sigma_i V_i^T$$

where  $U_i$  is  $n \times n$  orthogonal matrix,  $\Sigma_i$  is a  $n \times m_i$  matrix whose diagonal contains the singular values and  $V_i$  is an  $m_i \times m_i$  orthogonal matrix.

Given a sequence vector  $x$  of unknown classification, by applying the SVD projections to the data, we determined its classification by computing the relative magnitudes of  $\|P_{\alpha helices} \hat{x}\|$ ,  $\|P_{\beta sheets} \hat{x}\|$ , and  $\|P_{other} \hat{x}\|$ . The SVD is a functional approach in that  $P_{\tilde{U}} = P_A$ , where, by using Matlab,  $\tilde{U} = U(:, rank(A))$  and  $P_{\tilde{U}} = \tilde{U} \tilde{U}^T$ .

### 2.3 Gradient Descent

Gradient Descent is an algorithm for finding the nearest local minimum of a function whose gradient can be computed. Also known as the method of steepest descent or a linear network, the gradient descent method starts at a point  $P_0$  and, as many times as needed, moves from  $P_i$  to  $P_{i+1}$  by minimizing along the line extending from  $P_i$  in the direction of  $-\nabla f(P_i)$ , the local downhill gradient [7]. The goal is to obtain a set of weights,  $w$ , for computing the linear decision boundary where  $g(x) = w^T x + w_0$ , and minimizing the mean squared error at the same time.

Using the weights obtained from this linear classifier, we formulated a matrix,  $A = [TestVector; ones(1, N)]$ , then computed the network response to the test vector by computing  $netout = w * A$  and then classifying this output based on the maximum value in each column.

### 2.4 BackPropagation Network

Neural Network Models are widely used in secondary structure prediction. These networks are composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. An example is A BackPropagation (BackProp) network, which consists of at least three layers: an *input* layer, at least one intermediate *hidden* layer, and an *output* layer.

Units are connected in a feed-forward fashion with input units fully connected to units in the hidden layer and hidden units fully connected to units in the output layer. The output of a BackProp network can be interpreted as a classification decision.

In a BackProp network, learning occurs during a training phase in which each input pattern in a training set is applied to the input units and then propagated forward. The pattern of activation arriving at the output layer is then compared with the correct (associated) output pattern to calculate an error signal. The error signal for each such target output pattern is then backpropagated from the outputs to the inputs in order to appropriately adjust the weights in each layer of the network.

After a BackProp network has learned the correct classification for a set of inputs, it can be tested on a second set of inputs to see how well it classifies untrained patterns.

To utilize the BackProp network, we pass the encoded data matrix along with each known classification to our algorithm which then computes  $W^* = \{w^*, \tau^*, \alpha^*\}$ . Given sequences of unknown classification, we encoded them using each of the three encoding techniques mentioned and computed the network response to the test vectors by computing

$$network\_response = \alpha(\sigma(w^*testvector - \tau))$$

## 2.5 Encoding Methods

### Standard Vector Encoding, hydrophobic/hydrophilic charged alphabet, and volume alphabet

The orthogonal or standard vector encoding is commonly used to encode sequence data. For this method, each symbol (4 nucleotides for DNA and 20 amino acids for proteins) is used to create a  $k$ -dimensional unit vector, where  $k$  is the number of symbols. Sequences are encoded as a vector by concatenating the appropriate unit vectors.

For amino acids, where  $k = 20$ , the  $j^{th}$  amino acid where  $1 \leq j \leq 20$  is represented by a 20 dimensional vector that is assigned a one at the  $j^{th}$  position and zero in every other position [5].

The standard encoding approach may be expanded by categorizing the standard amino acid alphabet into families that take into account physical and

chemical characteristics. Entries within the data matrices may be weighted based upon their hydrophobicity, charge and volume. The hydrophobicity index is a measure of the relative hydrophobicity, or how soluble an amino acid is in water.

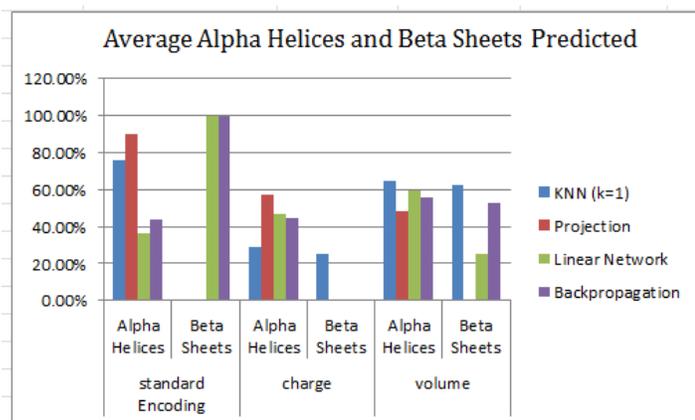
A peptide's volume can be estimated from the molecular weight of the peptide and an average protein partial specific volume. The partial specific volume of a protein is the ratio between its volume and molecular weight. Similarly, proteins that have to bind to positively-charged molecules have surfaces rich with negatively charged amino acids like glutamate and aspartate, while proteins binding to negatively-charged molecules have surfaces rich with positively charged chains like lysine and arginine.

## 3. RESULTS

All four methods investigated behaved differently for each encoding technique.

Our results show that the Back-propagation network achieved higher results than the linear network on average, for the total number of beta sheets and alpha helices predicted. See table 1 and figure 1. For the standard encoding, using 200 neurons and 10,000 iterations for the network training, the BackProp network accrued a mean squared error of  $1.3 \times 10^{-3}$ , while the linear network accrued  $3.03 \times 10^{-5}$  after 20,000 iterations. Using the charge alphabet encoding technique, the linear network had a total MSE of 20.12 while the BackProp network totaled 2.71. The error accrued using the volume charge alphabet encoding scheme was also significantly lower for the BackProp network than the linear network (3.12 to 14.78).

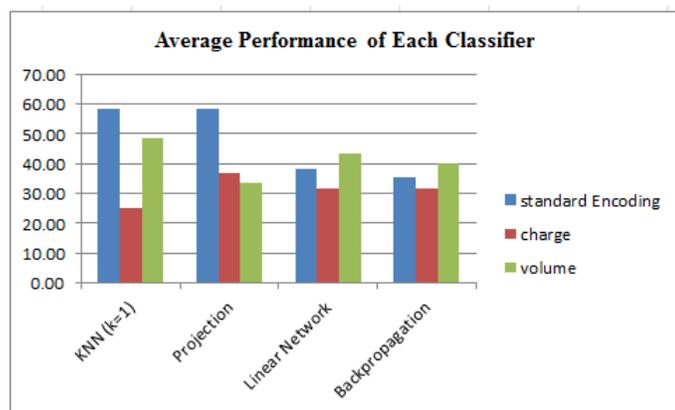
The projections method performed extremely well, especially for the standard encoding approach. This can be attributed to the fact that the dimensions are large and by reducing dimensionality issues, the SVD approach is very reliable. When the vectors are unique, the Euclidian distance will be unique, so the  $k$ -nearest neighbor works well in cases such as the standard encoding technique, achieving (~58%) accuracy.



**Figure 1:** Bar chart comparing the performance of each classifier based on alpha and beta sheet classification

Classifier	Standard Encoding		charge		volume	
	Alpha Helices	Beta Sheets	Alpha Helices	Beta Sheets	Alpha Helices	Beta Sheets
KNN (k=1)	3/4	0	2/7	1/4	2/3	5/8
Projection	8/9	0	4/7	0	1/2	0
Linear Network	1/3	1	1/2	0	3/5	1/4
Backpropagation	4/9	1	4/9	0	5/9	1/2

**Table 1:** The average number of accurately predicted alpha helices and beta sheets by each classifier. These numbers were calculated by dividing the number correct by the number of vectors.



**Figure 2:** The average performance of each classifier for each encoding method.

## 4. DISCUSSION

By running a total of five hold out tests (90% for training and 10% testing), we were able to calculate the performance of the four methods we used in our experiment. Appendices 1-3 show the detailed results obtained for each algorithm, which we tested using each encoding approach.

We compared the network classification with the actual classification by dividing the number of correctly predicted vectors by the number of vectors tested.

Our results show that for the standard encoding approach, k-nearest neighbor (k=1) achieved similar results to the projections method. Both achieved (~58%) total accuracy. However, the projections method accurately predicted (90%) of the alpha helices, while k-nearest neighbor predicted (76%).

The back-propagation achieved higher percentages than the linear network in most cases. This was expected since the BackProp is a universal approximator.

## 5. CONCLUSION

We investigated the performance of the K-nearest neighbor algorithm, the projections method using SVD, Gradient Descent, and the BackPropagation network, which are popular methods used in pattern classification. Using standard vector encoding, the charge alphabet, and the volume alphabet, we showed that as a lazy classifier, the k-nearest neighbor yields better responses to unique data, since it classifies data based on distances. The projection method using SVD works better with unique matrices of large dimensions, given its ability to decompose the data matrices. On the other hand, the Backpropagation network is a universal approximator, which has to be trained properly using an ideal number of neurons if a desired response is expected. It performs better than the gradient descent method, which is a linear classifier that works better with data of a linear nature.

## 6. REFERENCES

- [1.] Pattern Classification, Imed Hammouda & Jakub Rudzki  
(<http://www.cs.tut.fi/~kk/webstuff/PatternClassificationKalvot.pdf>)
- [2.] B. Rost, Review: Protein Secondary Structure Prediction Continues to Rise, *Journal of Structural Biology* 134, p204-218, 2001
- [3.] Junger Tang, Design and Implementation of a Neural Network For the Evaluation of Protein Secondary Structure Prediction Using the DSSP and Chou-Fasman Algorithms  
<<http://web.mit.edu/rsi/www/pdfs/papers/95/jungert.pdf>>
- [4.] P. G. Wodehouse, Bioinformatics and Pattern Recognition Come Together, *Journal of Pattern Recognition Research* 1 (2006) 37-41
- [5.] Eric Sakk and Iyanuoluwa E. Odebode. Vector Space Information Retrieval Techniques for Bioinformatics Data Mining
- [6.] Moslehian, Mohammad Sal; Rowland, Todd; and Weisstein, Eric W. "Projection Matrix." From MathWorld--A Wolfram Web Resource.  
<http://mathworld.wolfram.com/ProjectionMatrix.html>
- [7.] Weisstein, Eric W. "Method of Steepest Descent." From MathWorld--A Wolfram Web Resource.  
<http://mathworld.wolfram.com/MethodofSteepestDescent.html>
- [8.] Devin McAuley, The BackPropagation Network: Learning by Example, Updated for BrainWave 2.0 by Simon Dennis 1999, <http://itee.uq.edu.au/~cogs2010/cmc/chapters/BackProp/>