Rule Clustering and Super-rule Generation for Transmembrane Segments Prediction

Jieyue He[1,2], Bernard Chen[2], Hae-Jin Hu[2],
Robert Harrison[2,3,4], Phang C. Tai[3], Yisheng Dong[1] and Yi Pan[2,*]
[1]*Department of Computer Science*
Southeast University, Nanjing 210096, China
Email: jieyuehe@seu.edu.cn
[2]*Department of Computer Science,*[3]*Department of Biology*
Georgia State University, Atlanta, GA 30303-4110, USA
Email: pan@cs.gsu.edu
[*]*Corresponding author*
[4]*GCC Distinguished Cancer Scholar*

Abstract

The explanation of a decision is important for the acceptance of machine learning technology in bioinformatics applications such as protein structure prediction. In past research, we have already combined SVM with decision tree to extract rules for understanding transmembrane segments prediction. However, rules we have gotten are as many as about 20,000. This large number of rules makes them difficult for us to interpret their meaning. In this paper, a novel approach of rule clustering (SVM_DT_C) for super-rule generation is presented. We use K-means clustering to cluster huge number of rules to generate many new super-rules. The experimental results show that the super-rules produced by SVM_DT_C can be analyzed manually by a researcher, and these super-rules are not only new but also achieve very high transmembrane prediction accuracy (exceeding 95%) most of the times.

1. Introduction

Recent years, there have been many studies focusing on improving the accuracy of transmembrane segments prediction, and many good results have been achieved [1,2]. In spite of these new results, the existing methods do not explain the process of how a learning result is reached and why a prediction decision is made. The explanation of a decision made is important for the acceptance of machine learning technology in bioinformatics applications such as protein structure prediction. In past research, we have already combined SVM with decision tree to extract rules for understanding transmembrane segments prediction. However, rules we have gotten are as many as about 20,000. Such a large number of rules are difficult for researchers to interpret and analyze. Clearly, it will not be satisfactory for researchers to simply use arbitrary small subset of rules because the subset of rules can't cover all the data in the domain. Therefore, in this paper, a novel approach of rule clustering (SVM_DT_C) for super-rule generation is presented. We use K-means clustering to cluster huge number of rules based on similarity, and then aggregate the rules in each cluster to generate new super-rules. These super-rules represent the consensus rule pattern and the essential underlying relationship of classification. Because the super-rules come from each of clusters, the researchers not only can understand the general trend and ignore the noise but also can interactively focus on the key aspects of the domain by using super-rules and selectively view the original detail rules in the corresponding of cluster.

2. Methods

Support vector machines (SVM)[3] have shown strong generalization ability in a number of application areas, including protein structure prediction while a decision tree has good comprehensibility. By combining SVM and decision tree, we can produce a

large number of rules for understanding transmembrane segments prediction. On the other hand, the goal of clustering is to reduce the amount of data by categorizing or grouping similar data items together. Usually, one of the motivations for using clustering algorithms is to provide automated tools to help in constructing categories or taxonomies [4]. Therefore, we develop a hybrid approach (SVM_DT_C) which combined SVM, decision tree and clustering methods to produce super-rules. This approach proceeds in four steps. Firstly, we use the combined orthogonal and Blosum62 matrix as encoding schemes to train the SVM. Secondly, in order to achieve better training results, we use correct prediction results from SVM to feed into a decision tree. Thirdly, we decode the rules into logical rules with biological meaning according to encoding schemes. Finally, we aggregate similar rules through K-means clustering into several clusters and get some cluster score matrices, which indicate the importance of each amino acid in a particular position for prediction. Through the cluster score matrices, super-rules are generated.

The pseudo-code of our SVM_DT_C algorithm is shown in Figure 1. Suppose we are given a training data set $S=\{(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)\}$, where x_i is the feature vector and y_i is the expected class label or target of the i-th training instance. At first, SVMs are trained using N-fold cross validation. That is, for data set S, we divided it into N subsets with similar sizes (k) and similar distribution of classes. We perform the tests for N runs, each with a different subset as the test set (Te_svm_i,i=1…N) and with the union of the other N-1 subsets as the training set (Tr_svm_i, i=1…N). Then, from each test set (Te_svm_i, i=1…N), based on the result of prediction, we select cases that are correctly predicted by SVM into new data set (S_i_svm, i=1…N). We use the original test data Te_svm_i,i=1…N as test data set (Te_dt_i,i=1…N) and the union of the other N-1 subsets S_i_svm as the training set (Tr_dt_i,i=1…N) to train a decision tree and induce the rule sets. Thirdly, we decode the rules into logical rules (L_R) with biological meaning. Finally, K-means clustering is used to cluster the huge number of rules generated in step 3 based on similarities, and then according to the cluster score matrices aggregate the rules in each cluster to generate new super-rules (S_R). Since a support vector machine usually has strong generalization ability and we select the new data set from the correct result of SVM as our inputs to DT, we believe that some bad ingredients of S, such as the noise, may be reduced by the process of SVMs, and some weak cases may be sieved by SVMs. It is indicated that new data set Si_svm data is better than the original training data set S for rule induction. This is the reason why we use support vector machine as a pre-process of decision tree.

```
Input: training set S
Output: Super-rule set S_R
Process:
for i=1 to N {   /*generate N sets with similar sizes (k)
                 of train data and test data */
 {Tr_svm_i,  Te_svm_i }=Create_cross_validation_data(S)
}
for i=1 to N { /* for each set, train SVM, then select from
                 P_i_svm data into new data set S_i_svm */
   P_i_svm=SVM(Tr_svm_i, Te_svm_i)
   S_i_svm=Φ
   for j=1 to k {
        if P_i_svm_j is correct
          S_i_svm= S_i_svm ∪ P_i_svm_j
   }
}
for i=1 to N {/* generate N sets of train data for decision
                 tree using N-fold cross validation */
   Te_dt_i = Te_svm_i
   Tr_dt_i = Tr_dt_i ∪ {S_j_svm, j=1,..N, j≠ i}
}
for i=1 to N {  /* Each set is fed into decision tree to induce
                 rules, and then decode rule into logical
                 rule(L_R_i)  */
   R_i=dt (Tr_dt_i, Te_dt_i)
   L_R_i=decode(R_i)
}
S_R= Φ
for i=1 to N {  /* for each R_i set, clustering to generate super-
                 rule(S_R) */
   {rule_number, Matrix_i}=k-means (L_R_i)
   S_R_i=create_super_rule(Matrix_i )
   S_R = S_R ∪ S_R_i
}
```

Figure 1 SVM_DT_C algorithm

3. Experiments and Results

In this study, the data set given by Rost et al. is tested and this is labeled as data set of 165 low-resolution. According to Rost et al. [1], the 165 proteins is expert-curated set from the SWISS-PROT database which was originally collected by Möller et al. [5]. For encoding scheme, we use combined orthogonal and Blosum62 matrix. The orthogonal encoding scheme is the simplest profile which assigns a unique binary vector to each residue, such as (1, 0, 0…), (0, 1, 0…), (0, 0, 1…) and so on. The Blosum62 matrix represents the "log-odds" scores for the likelihood that a given amino acid pair will interchange

with one another and it contains the general evolutionary information among the protein families [6]. For the preliminary screening, the performance of each matrix is compared with that of the combined matrix. Since the combined matrix showed a better performance than the orthogonal or the Blosum62 matrix taken alone, it is adopted for training the SVM.

In the experiments, firstly, to train the SVM, we selected the kernel function $K(x,y) = e^{-\lambda\|x-y\|^2}$ based on the previous studies, and the parameter of the kernel function λ and the regularization parameter C were optimized based on tests [7]. With the data set, we ran 7-fold cross validation in the experiments. In each run, we fed the training data into SVM^{light} to get the model and used test data as validation. We use decision tree of C5.0 and C5.0 rules to produce the rules, and get 7 group rule sets. Then, we parsed the rule sets produced by C5.0 and obtained the logical rules with biological meaning by decoding the rules according to the encoding schemes used. Finally, we used k-means to cluster rules according to similarity of rules. K-means is the most widely used method in partitioning categories due to its fast speed and easy understanding. The method uses something called *centroid* which is the mean point in a cluster. K-means minimizes the intra-cluster distance between any point in the cluster and the centroid. We applied this method in our classification rules clustering process by combining similar rules together to generate more general and error-tolerance rules. We use random method to generate initial centroids positions; we set K equals to 20 for transmembrane prediction rules, 30 or 35 (depends on the result) for non-transmembrane prediction rules. Comparison of percent of rule numbers of 7 groups of super-rules for different accuracy ranges of predicting 'T' is shown as Table 1.

Table 1. Comparison of percent of rule numbers of 7 groups of super-rules for different accuracy ranges of predicting 'T'

Accuracy	1T (%)	2T (%)	3T (%)	4T (%)	5T (%)	6T (%)	7T (%)
95-100	27.7	12.5	27.8	30.8	25.5	37.1	27.1
90-95	25.5	20.0	13.9	20.5	21.6	14.3	16.7
85-90	14.9	17.5	19.4	15.4	15.7	8.6	16.7
80-85	21.3	17.5	13.9	12.8	17.6	8.6	14.6
<80	10.6	32.5	25.0	20.5	19.6	31.4	25.0

Table 2. One example of super-rules by SVM_DT_C and explanation

Logical rule with biological meaning	IF Sq[3] in {A,C,G,I,L,M,F,S,T,W,Y,V} AND Sq[4] in {A,C,G,I,L,M,F,T,W,Y,V} AND Sq[6] in {A,C,G,I,L,M,F,T,W,Y,V} AND Sq[7] in {I,F} AND Sq[8] in {I,L,M,F,V} AND Sq[9] in {C,G,I,L,F,W,Y,V} AND Sq[10] in {A,C,G,I,L,M,F,P,S,T,W,Y,V} AND Sq[11] in {A,C,G,H,I,L,M,F,S,T,W,Y,V} AND Sq[13] in {I,L,M,F,W,Y,V} THEN St[7]=T accuracy 100.00%, support 0.30%
Rule explanation	If the amino acid the fourth, the third, and the first positions before the target position is one of {A,C,G,I,L,M,F,S,T,W,Y,V},{A,C,G,I,L,M,F,T,W,Y,V}, {A,C,G,I,L,M,F,T,W,Y,V},respectively , at the same time, the first, the second, the third, the fourth, the sixth amino acid following the target position is one of {I,L,M,F,V},{C,G,I,L,F,W,Y,V}, {A,C,G,I,L,M,F,P,S,T,W,Y,V},{A,C,G,H,I,L,M,F,S,T,W,Y,V},{I,L,M,F,W,Y,V},respectively, and the target amino acid is one of {I,F}, the prediction at the target position is 'T' (transmembrane) with accuracy 100 % and support 0.30 % .

Table 3. One of the cluster score matrices of {A,R,N,D,C,Q,E,G,H.I}

	A	R	N	D	C	Q	E	G	H	I
1	91	95	100	100	95	100	87	100	95	37
2	25	29	25	8	25	25	20	29	0	8
3	25	16	33	20	25	25	33	29	16	20
4	20	16	20	20	16	16	20	12	12	8
5	25	25	29	25	29	25	29	29	20	25
6	100	100	95	45	100	95	100	100	25	50
7	25	29	25	29	25	25	25	25	25	33
8	29	29	29	29	29	29	33	29	25	37
9	66	75	75	62	75	79	75	79	4	16
10	25	29	25	25	33	33	29	29	16	16
11	8	8	8	8	8	8	12	8	0	4
12	41	37	37	29	41	37	29	37	4	20
13	0	0	0	0	0	0	0	0	0	0

From Table 1, we can see that the percent of rule numbers with prediction accuracy over 90 is about 60%. It means that most of the super-rules generated have high quality. Two examples of super-rules and explanation are shown in Table 2. These super-rules are very useful in guiding biological experiments. In the clustering process, we get the cluster score matrixes, such as Table 3. The entries in the matrices indicate the profile of the amino acids in each of the 13 positions in a window for transmembrane segments prediction. For example, the value of 95 in the first row and the second column indicates that the amino acid R has 95% possibility for prediction.

4. Conclusion

To extract rules for understanding transmembrane segments prediction is important for the acceptance of decision results in bioinformatics. However, for thousands of rules, it is difficult for one to analyze and interpret. This paper proposes an approach of rule clustering (SVM_DT_C) for super-rule generation. The experiments show that most of the super-rules generated not only have high quality (the percent of rules with accuracy over 90 is about 60%), but also are different from the original rules before clustering because these super-rules are produced by aggregating the original detail rules and indicate the general trend.

Acknowledgement

The authors would like to thank Professor Thorsten Joachims for making SVM^{light} software available and thank RuleQuest Research for ten-day evaluation licence of C5.0 software available and Professor Burkhard Rost for providing the 165 low-resolution data sets.

This research was supported in part by the scholarship under the State Scholarship Fund of China, and the U.S. National Institutes of Health (NIH) under grants R01 GM34766-17S1, P20 GM065762-01A1, and the U.S. National Science Foundation (NSF) under grants ECS-0196569, and ECS-0334813. This work was also supported by the Georgia Cancer Coalition and computer hardware used was supplied by the Georgia Research Alliance.

References

[1] C.P Chen, A. Kernytsky, and B. Rost "Transmembrane helix predictions revisited" Protein Science, 2002. vol. 11, pp.2774-2791

[2] A.R. Sikder and A.Y. Zomaya "An overview of protein-folding techniques: issues and perspectives" Int. J. Bioinformatics Research and Applications, 2005, Vol.1,issure 1,pp.121-143.

[3] J.R Quinlan, "Improved Use of Continuous Attributes in C4.5.", J. Artificial Intelligence Research, 1996, vol. 4, pp. 77-90,

[4] J.Han "Data mining Concepts and Techniques", Morgan Kaufmann Publishers (2001.

[5] S. Möller, Kriventseva, E. V. Apweiler, "A collection of well characterized integral membrane proteins", Bioinformatics, 2000, vol. 16, pp. 1159-1160.

[6] S. Henikoff, and J.G Henikoff, "Amino acid substitution matrices from protein blocks" PNAS, 1992,.vol. 89, pp. 10915-10919.

[7] H.Hu, R.Harrison, P.C.Tai, and Y.Pan, "Transmembrane Segments Prediction with Support Vector Machine Based on High Performance Encoding Schemes", Proc. of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, La Jolla, California, USA. , Oct. 7-8, 2004.