The Fourth International Conference on Computational Systems Biology (ISB2010)
Suzhou, China, September 9–11, 2010
Copyright © 2010 ORSC & APORC, pp. 258–265

An Improvement Based on Feature Selection in Enzyme Identification

Ji-Xiang Wei Fang-Zhen Li Sheng-Yuan Wang

School of Computer Science & Technology, Shandong Economic University, Jinan, China
Fzli1976@gmail.com

Liu-Huan Dong

CAS-MPG Partner Institute for Computational Biology, Shanghai Institutes for Biological Sciences, CAS, Shanghai, China
dlh@picb.ac.cn

Abstract Identification for enzymes is a prerequisite for understanding their functions. In recent years, machine learning has come to emerge in this field. In 2007, an automatic enzyme classifier based on support vector machine with feature vector from protein functional domain composition was built to identify enzymes from proteins. To achieve a better performance, in this paper, the method based on 1-rule was adopted to select proper number of features to build an automatic enzyme classifier with better accuracy. According to Jackknife cross-validation test, the 88.76% success rate achieved for enzyme/non-enzyme identification, which is about 2.7% higher than the success rate got in former experiment.

Keywords: enzyme; identification; support vector machine; 1-rule; jackknife cross-validation test

1 Introduction

With the development of technology, we could capture and store vast quantities of data. In the information age, one of the great challenges is finding patterns, trends and anomalies in these datasets, meanwhile summarizing them with simple quantitative models—turning data into information and turning information into knowledge. There has been stunning progress in data mining and machine learning. The synthesis of statistics, machine learning, information theory, and computing has created a solid science, with a firm mathematical base, and with very powerful tools.

Identification for enzymes is a prerequisite for understanding their functions. Traditionally, the comparison of sequence similarity methods were widely used in enzyme identification and classification, such as BLAST (Basic Local Alignment Search Tool) [6], which is a comparison of the similarity analysis tool in a DNA database or protein database.

In recent years, many research about identification and classification for enzymes with the method of machine learning has occurred in this work. As early as the year

2002, Jenson et al. first proposed a method based on primary sequence information to predict novel archaeal enzymes by applying Artificial Neuron Networks incorporating with features covering various sequence-related physicochemical features, such as signal peptide, trans-membrane and phosphoration properties [7]. In 2004, SVM (support vector machine) was imported to identify and classify enzymes with features consisting amino acid compositions by Cai and achieved success rates ranging from 50.0% to 95.7% for various enzyme categories [4, 5]. In 2007, Lu adopted SVM with feature vector based on protein functional domain composition knowledge to investigate this problem and the success rate of the experiment reached 86.03% for enzymes identification from protein [1].

In order to enhance the performance of SVM formerly used in [1], in this study we adopted 1-rule algorithm to select proper number of features for the input of SVM. Through our work, the success rate for identification for enzymes reached 88.76%, which was about 2.7% higher than the classifier constructed in [1].

2 materials and methods

2.1 Acquiring and Filtering the Ddatasets

To make comparison available, we use the same dataset as used in [1]. The original data covering 70537 enzymes came from Enzyme nomenclature database release 39.0 of February 2006 [http://www.expasy.org/ enzyme/]. Besides, 145271 non-enzyme proteins were colleted from UniProtKB/Swiss-Prot Release 49.3 of 21 March 2006 [http://www.expasy.org/sprot/] [8] excluding the enzymes mentioned above.

The datasets were filtered as follows [1]: (1) Remove enzymes with length less than 50 amino acids or more than 5000 amino acids. (2) Remove enzymes containing irregular characters such as B, J, O, U, X, Z. (3) Remove enzymes assigned to multiple categories. (4) Remove redundancy against homologous bias by using CD-HIT [9] and PISCES [10]. As a consequence, each enzyme in the filtered dataset shared no more than 25% identities with any other. (5) Remove enzymes without domain information in Pfam database and orphan proteins that possess Pfam [3] [http://www.sanger.ac.uk/Software/Pfam] domains that occur only once in our dataset. After being filtered, 2443 enzymes were obtained. (6) Randomly select 4668 (twice as big as the enzyme dataset) non-enzyme proteins which were filtered from the negative dataset in the same process as described above. Finally, a total dataset with 2443 enzymes and 4886 non-enzyme proteins were obtained.

2.2 Numeric Representation of Proteins based on Functional Domain Composition

To build a classifier with machine learning approaches, it's important to represent each protein in discrete numbers. Since an enzyme's category is closely related to its function domains, we can use these domains as characters to identify enzyme from other proteins. This can be realized by querying the proteins information in Pfam database [3] [http://www.sanger.ac.uk/ Software/Pfam/] (Version 19.0 on Nov 2005) recording the function domain information of proteins. The numeric representation of proteins is expressed as follows:

- Extract the functional domains of each protein. In our dataset, the 7329 proteins (enzymes and non-enzyme proteins) totally cover 2657 functional domains.
- Represent each protein in the form of 2657 dimension vector with each of 2657 domains as the vector base. E.g., if enzyme P48790 hit the domain PF04616 which is the 2017th record in the 2657 Pfam accessions, the 2017th element of the vector is set to 1, otherwise is set to 0, formulated as

$$P = \begin{pmatrix} p_1 \\ p_1 \\ \vdots \\ p_i \\ \vdots \\ p_{2657} \end{pmatrix} \quad (1)$$

where

$$p_i = \begin{cases} 1 & \text{when the protein hits the } i\text{th domain accession} \\ 0 & \text{otherwise} \end{cases}$$

2.3 1-rule

2.3.1 why we adopt feature selection

The reason we adopt feature selection could be described as follows.

In machine learning, the training data is used by one or more learning methods to construct classifiers. If lots of data is available, there is no problem: we take a large sample and use it for training; then take another independent large sample of different data and use it for testing. Provided that both samples are representative, the error rate on the test set will give a true indication of future performance. The real problem occurs when there is not a vast supply of data available. In many situations the training data must be classified manually—and so must the test data, of course, to obtain error estimates. This limits the amount of data that can be used for training, validation, and testing, and the problem becomes how to make the most of a limited dataset.

As we know, there are many different kinds of simple structure that datasets can exhibit. In one dataset, there might be a single feature that does all the work and the others may be irrelevant or redundant. In another dataset, the features might contribute independently and equally to the final outcome. There are also other structures that datasets can exhibit. This shows that the importance of different features is not the same with each other. Besides, we don’t want to generate perfect rules that guarantee to give the correct classification on all instances in the training set, but would rather generate “sensible” ones that avoid over fitting the training set and thereby stand a better chance of performing well on new test instances.

Summing up the two points above, we select proper number of features as training data to build a classifier.

2.3.2 1-rule

1-rule (1R) is a simple, cheap method that often comes up with quite good rules for characterizing the structure in data. When analyzing practical datasets, we strongly recommend the adoption of a "simplicity-first" methodology that often works very well [2]. There are many famous and simple rules which have the ability of selecting the informative features, such as t-test and F-score, but 1R is more applicable to nominal features, especially to binary features.

The idea of 1R is this: we make rules that test a single feature and branch it accordingly. Each branch corresponds to a different value of the feature. It is obvious that the best classification for each branch is to assign the class to it that occurs most often in the training data of this branch. Then the error rate of the rules can easily be determined. Just count the errors that occur on the training data, that is, the number of instances that do not have the majority class.

Each feature generates a different set of rules, one rule for one value of the feature. Evaluate the error rate for each feature's rule set and choose the proper number of features. Figure 1 shows the algorithm in the form of pseudo codes.

```
For each feature
{
  For each value of that feature, make a rule as follows:
  {
    count how often each class appears
    find the most frequent class
    make the rule assign that class to this feature-value.
  }
  Calculate the error rate of the rules for the feature.
}
Choose the proper number of features.
```

Figure 1. Pseudocode for 1R.

In our dataset, we consider functional domain as feature, and there are 2657 different functional domains. To see the 1-rule at work, take domain *i* as an example: supposing *n* enzymes have domain *i*, *m* non-enzyme proteins have domain *i* ($n<m$), *p* enzymes don't have domain *i*, *q* non-enzyme proteins don't have domain *i* ($p>q$), we'll get a set of rules as TABLE 1

TABLE 1. THE ERRORS CALCULATION OF DOMAIN I

Feature	Rules	Errors	Total errors
Domain *i*	P_i =1 ⇒ non-enzyme protein	$n/(n+m)$	$(n+q)/7329$
	P_i =0 ⇒ enzyme	$q/(q+p)$	

P_i =1 represents that the protein has domain i

P_i =0 represents that the protein doesn't have domain i

As TABLE 1 shows, the error rate of feature (domain *i*) is $(n+q)/7329$. This is how 1-rule works with domain *i*. According to the process above, we calculated the error rates of each feature and took them as the feature information.

Generally, the data have their own structure and different features may show a relationship with each other. But with the method 1R, it uses all the features and allows them to make contributions to the decisions equally importantly and independently of each other. This is unrealistic, of course: what makes real-life datasets interesting is that the features are certainly not equally important or independent. But it leads to a simple scheme that again works surprisingly well in practice. According to the result shown below, the classifier gets quite good identification accuracy. This indicates the rules produced by testing single feature can replace the complex structure.

2.4 Testing

In order to get a proper set of features for building a classifier we tested different numbers of features. First, we sorted the features in ascending order according to the error rates. The reason for choosing the ascending order is that the rules belonged to the features with smaller error rates are more suitable to judge whether the protein is enzyme or not. Second, we selected different numbers of features as test data. The features were selected from front to back and the number of features increased by 100.

3 result and discussion

According to the procedure mentioned above, 2443 enzymes and 4886 non-enzymes were represented by binary vectors of 2657 dimensions. Through the method based on 1-rule, we extracted all the features information and selected different numbers of features as the input of SVM^{light} [http://www.cs.cornell.edu/People/tj/svm_light/] [11]. Then, the classifier which was used for enzyme/non-enzyme identification was built by utilizing SVM^{light} software package. SVMlight provide several common kernels, such as linear kernel, polynomial kernel and radial basis function kernel. In light of the analysis in [1], the linear kernel was chosen, and other parameters were set to default values.

In our experiment, 26 runs were done, each of which differed by the number of features by 100 (see Table 2).

3.1 Jackknife Cross-validation Test was Adopted to Evaluate the Predictor Performance

In order to evaluate the performance with different training datasets, we selected jackknife cross-validation tests to complete this work.

Jackknife cross-validation is a quite objective and rigorous way in algorithm evaluation, and it performed as follow:

- To identify enzyme/non-enzyme for each protein P in the dataset consisting of 2443 enzymes and 4886 non-enzymes, we trained the SVM^{light} with the dataset excluding P and examined the performance on P. The identification succeeded if it correctly predicted the category of the query protein. The success rate for enzyme/non-enzyme identification can be calculated according to equation (1):

$$\begin{cases} \text{success rate for enzymes} = \dfrac{\text{number of correctly predicted enzymes}}{\text{number of true enzymes}} \\ \text{success rate for non-enzymes} = \dfrac{\text{number of correctly predicted non-enzymes}}{\text{number of non-enzyme}} \\ \text{success rate for overall} = \dfrac{\text{number of correctly predicted proteins}}{\text{total number of proteins}} \end{cases} \quad (2)$$

TABLE 2. JACKKNIKFE TEST OF ENZYME/NON-ENZYME IDENTIFICATION

number of Feature	**success rate (%)**		
	enzyme	**non-enzyme**	**overall**
100	31.31	98.26	75.94
200	47.48	96.73	80.31
300	57.84	95.93	83.23
400	65.62	95.48	85.52
500	72.37	94.84	87.35
600	77.45	94.41	88.76
700	77.81	93.78	88.44
800	77.32	93.04	87.80
900	77.28	92.80	87.64
1000	77.45	92.45	87.42
1100	77.08	92.28	87.26
1200	76.91	92.06	87.04
1300	77.04	92.18	87.11
1400	76.95	92.02	87.01
1500	76.42	91.98	86.82
1600	75.89	91.92	86.59
1700	76.18	91.63	86.51
1800	76.42	91.55	86.52
1900	76.67	91.49	86.56
2000	76.59	91.40	86.52
2100	76.71	91.26	86.42
2200	76.91	91.16	86.40
2300	76.59	91.20	86.33
2400	76.50	91.10	86.23

2500	76.46	91.08	86.21
2600	76.22	90.95	86.04

As the TABLE 2 shows, in enzyme/non-enzyme identification, the success rate of jackknife cross validation test for enzyme, non-enzyme and overall all changed with the increasing number of training features. When the number of features was 600, the classifier based on our method got the best performance and the success rate for overall achieved 88.76%. When number is above 600, with the number increasing, the success rates appear the tendency of declining slowly.

3.2 Our Classifier is better than the one trained on the dataset without feature selection

As shown in TABLE 2, the best number of features was 600. For comparison the success rates obtained upon selected 600 features and those obtained without feature selection (see [1]) were listed in TABLE 3. As shown in this table, the success rates obtained by our improvement were 77.45% for enzyme, 94.41% for non-enzyme and 88.76% for overall, which were all notably higher than the success rates obtained by the SVM trained on the dataset without feature selection. This is due to the noise raised by redundant features since the SVM used in [1] took the whole dataset as the training data, but didn't select proper features as training data. Through our work, we can see that when the number of features is above 600, some of the features and their rules are over fitting the training data and don't make the classifier get better performance on future test data, even becoming counterproductive. Our method based on feature selection with the 1-rule algorithm tested different number of features and finally remove some of features which were bad to generate high-precision classifier. Therefore, our improvement is an effective means to increase identification accuracy.

TABLE 3. CONTRAST WITH PREVIOUS WORK

Method	**Success rate (%)**		
	enzyme	**Non-enzyme**	**over**
SVM without feature selcetion	76.30	90.89	86.03
Our method	77.45	94.41	88.76

4 conclusion

Optimizing the machine learning process is an important part in getting better classifier or the structure of the data. 1R is an easy algorithm but does astonishingly well in comparison with state of the art learning methods. In solving this problem, we use the method based on 1R to select the proper features as the training data and the result outperforms the classifier of [1]. The success rate for identifying enzyme and non-enzyme achieves 88.76% which is about 2.7% higher than that got in [1]. Further analysis performed on the predictor confirmed that the rules that test a single feature are a viable alternative to more complex structures. In all, the result demonstrates that our method based on 1R is a very effective improvement in machine learning applying to enzyme identification. In addition, we also could use simple rudimentary techniques such as 1R firstly, and then progress to more sophisticated learning methods.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 30600121, 10701070), the Doctoral Foundation of Shandong Province, China (No. 2007BS09002) and the Natural Science Foundation of Shandong Province, China (No. ZR2009GQ015).

References

[1] Lu, L. et al., “ECS: An automatic enzyme classifier based on functional domain composition”, Computat. Biol.chem 2007
[2] Ian H.Witten, Eibe Frank, Data Ming-Practical Machine Learning Tools and Techniques, second edition, Morgan Kaufmann Publishers
[3] Bateman, A., Birney, E., Durbin, R., Eddy, S.R., Howe, K.L., Sonnhammer, E.L.L., 2000. The Pfam protein families database. Nucleic Acids Res. 28, 263–266.
[4] Cai, C.Z., Han, L.Y., Ji, Z.L., Chen, Y.Z., 2004a. Enzyme family classification by support vector machines. Proteins: Struct. Funct. Bioinform. 55, 66–77
[5] Cai, Y.D., Ricardo, P.W., Jen, C.H., Chou, K.C., 2004b. Application of SVM to predict membrane protein types. J. Theor. Biol. 226, 373–376.
[6] Altschul, S.F., Gish, W., Miller, W., Meyers, E.W., Lipman, D.J., 1990. Basic local alignment search tool. J. Mol. Biol. 215, 403–410.
[7] Jensen, L.J., Skovgaard, M., Brunak, S., 2002. Prediction of novel archaeal enzymes from sequence-derived features. Protein Sci. 11, 2894–2898.
[8] Wu, C.H., Amos Bairoch, R.A., Natale, D.A., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M.J., Mazumder, R., O’Donovan, C., Redaschi, N., Suzek, B., 2006. The Universal Protein Resource (UniProt): an expanding universe of protein information. Nucleic Acids Res. 34, D187–D191.
[9] Li, W., Godzik, A., 2006. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 22, 1658–1659.
[10] Wang, G., Dunbrack Jr., R.L., 2003. PISCES: a protein sequence culling server. Bioinformatics 19, 1589–1591.
[11] Scolk, B., Burges, f.C., Smola, A., 1999. Making large-scale SVM learning practical. In: Advances in Kernel Methods—Support Vector Learning. MITPress.