Study of multi-class classifiers : Application to Enzyme Classification Problem

A Dissertation Submitted in the Partial Fulfillment of the Requirements for the Award of Degree of

Master of Technology

in

Artificial Intelligence

By

SAIKIRAN V

09MCMI10

Department of Computer and Information Sciences

School of MCIS

University of Hyderabad

(P.O.) Central University, Gachibowli

Hyderabad - 500 046

Andhra Pradesh, India

May 3, 2011

CERTIFICATE

This is to certify that the project work entitled "**Study of multi-class classifiers : Application to Enzyme Classification Problem**" being submitted to University of Hyderabad by **SAIKIRAN V**, bearing Reg. No. 09MCMI10, in partial fulfillment for the award of the degree of Master of Technology in Artificial Intelligence, is a bonafide work carried out by him under my supervision.

Dr. S. Durga Bhavani
Project Supervisor,
Department of CIS,
University of Hyderabad.

Head ,
Department of CIS,
University of Hyderabad.

Dean,
School of MCIS,
University of Hyderabad.

DECLARATION

I, **Saikiran V**, here by declare that this dissertation entitled **"Study of multi-class classifiers : Application to Enzyme Classification Problem"** submitted by me under the guidance and supervision of **Dr. S Durga Bhavani** is a bonafide work. I also declare that it has not been submitted previously in part or in full to this University or other University or Institution for the award of any degree or diploma.

Date : Saikiran V

Place : 09MCMI10.

To,

My parents

Acknowledgments

I would like to express my profound thanks to my beloved teacher, erudite scholor and project supervisor **Dr. S. Durga Bhavani**, for her valuable suggestions, motivation, enthusiasm and keen personal interest throughout the progress of my project work. I could not have imagined having a better advisor as my guide. She has made available her support in number of ways which left me with invaluable experience. The project has been a learning and growing experience which laid a progressive path towards my career goals.

I am extremely grateful to our Head of the Department, **Prof. C.R.Rao**, for providing excellent computing facilities and a nice atmosphere for doing my project.

I take this opportunity to express my sincere thanks to **Prof Raju S.Bapi** and **Dr. T. Sobha Rani**, for their presence in our discussions and sharing the ideas without whose support and encouragement, I may not be able to reach the project goal.

At the outset, I would like to thank **The University of Hyderabad** for providing all the necessary resources for the successful completion of my course work.

Last but not the least, I would like to thank my classmates and other students of DCIS for extending their help and moral support .

With Sincere Regards,

Saikiran V.

Abstract

Multi-class classification problem has been solved using standard strategies of one-versus-all and all-versus-all . Some algorithms designed for binary classification can also be extended to address multi-class classification. In recent literature, a new system called Multiple classifier systems(MCS) are being used where rather than a single classifier, multiple classifiers are used. Their decisions are converted into a single decision by certain association techniques of combination, cooperation and selection of classifiers.

One of the fundamental tasks of bio-informatics is to classify proteins into one of their 6 enzyme classes. In this project, we consider the challenging multiclass problem called Enzyme classification(EC) problem, which involves the twin challenge of having multiple classes as well as unbalanced data across the classes. The challenge is to extract the function(EC class) from the structural features(binding site) rather than the sequence features. One of the issues is to find how the structural features(binding site) relate to function(EC class). Therefore, in this thesis, we consider the proteins whose binding sites are available. Bray et al [2] extracted 64 sequence and structural features from binding sites and achieved enzyme classification using Multiclass SVM with an accuracy of 33.1% which shows the challenge of the problem.

In this project, we have done extensive experimentation on this problem using combination of classifiers. We examined many issues involved with the imbalance of the dataset as well as the dependency of classification accuracy on the feature set and methods used. The best results we achieve on this same dataset is 56.6%.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 What is Classification ?

Classification is nothing but arranging the given data instance into predefined groups i.e, assigning one of several class labels to it.

- Given a **training set** of the form (X_i, y_i)
  - Each record in training set has a set of attributes X_i,and a classlabel y_i.
- Find a **model** for class attribute as a function of the values of other attributes.
- **Goal** : *previously unseen* records should be assigned a class as accurately as possible.
  - A **test set** is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

1.2 Machine learning methods for classification

Several machine learning techniques have been proposed to perform classification like Support vector machines, naiveBayes, Decision trees etc.

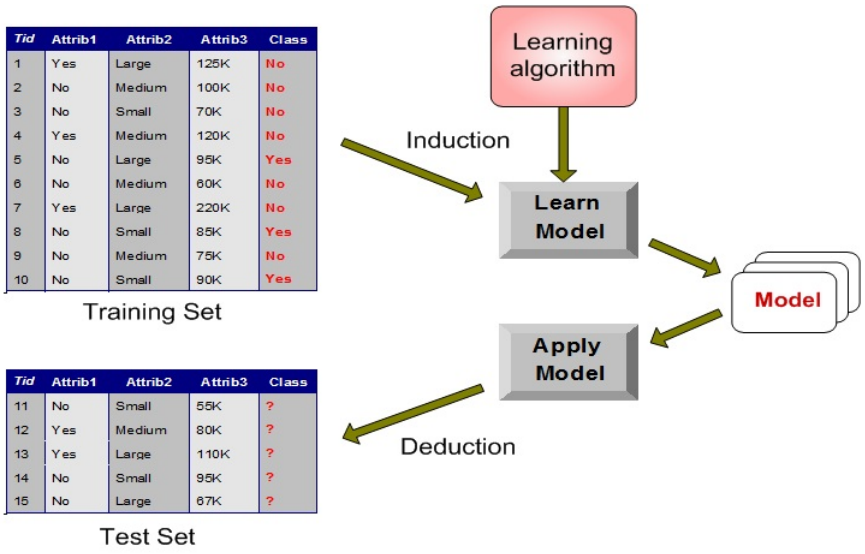

Figure 1.1: Classification Model

1.3 Multi-class classification

Multiclass classification is the special case in statistical classification of assigning one of several class labels to an input object. Unlike the better understood problem of binary classification, which requires discerning between the two given classes, the multiclass one is a more complex and less researched problem. In the past, multi-label classification was mainly motivated by the tasks of text categorization and medical diagnosis. Today, multi-label classification methods are seeing increased use in applications such as protein function classification, music categorization and semantic scene classification.

- Given training data of the form (x_i, y_i)
  where
  - $x_i = i_{th}$ example of the training set
  - $y_i \epsilon \{1, ..., k\}$ is its class label
- Goal : Find a learning model that classifies unseen new examples

1.4 Enzyme classification problem

Enzyme Classification is a challenging problem where enzymes are to be classified into one of the six enzyme classes - Hydrolase, Isomerase, Ligase, Lyase, Oxidoreductase and Transferase. This problem poses the twin challenge of having multiple classes as well as being a

	Binding Site (2008)	Whole protein (2008)	Binding Site (2010)	Whole protein (2010)
Hydrolase	355	269	742	713
Isomerase	62	48	120	129
Ligase	36	31	88	94
Lyase	107	81	158	169
Oxidoreductase	382	252	415	430
Transferase	390	306	686	724
Total	1332	987	2209	2259

Table 1.1: Dataset

highly unbalanced dataset. Each enzyme has several active sites where a ligand binds to the protein. The features of those sites are known Active-site(or Binding-site) features. Enzymes can be represented by their structural features, sequence features of whole protein as well as active-site. There is a question whether the active-site features alone can distinguish proteins. And which set of features has the highest power of distinguishing EC class is yet to be known. Extensive research is being done in this field of classifying enzymes using various features. For our project, we are using Amino acid composition, Fraction of Contact, Tightness of active site as the primary dataset. We have also worked on datasets with sequence features for active site as well as whole protein.

1.5 Aim

- To build a strong classification model to classify a new protein into one of the 6 enzyme classes based on the features extracted from the binding site.

1.6 R : package

In this thesis, we made use of R package [5] for developing programs for classification. R is a free software environment and language for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. It compiles and runs on a wide variety of UNIX platforms, Windows and Mac OS.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible.
- a wide range of add-on packages for machine learning techniques,neural networks etc.
- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hard copy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

Chapter 2

Literature

Classification is a supervised learning process aimed at producing a learned model from a labeled training set. Several techniques were proposed in literature to solve binary classification problem. But when it comes to multiclass classification, it becomes more delicate and difficult.

2.1 Solving multi-class problems

The aim of a classification algorithm is to assign a class label to each input example. For binary classification, the output classes can be either Positive or Negative whereas for a multiclass problem, the output class ranges from 1 to K. Several algorithms proposed to solve binary classification problem can be naturally extended to solve multiclass classification problem. The second way of solving this problem includes converting a multiclass problem into several binary classification problems. Another approach is to pose a hierarchy on the output space, the available classes, and perform a series of tests to detect the class label of new patterns.

2.1.1 Extensible algorithms

Extending the binary classification technique for some algorithms can solve multiclass classification problem [1]. Neural networks, decision tress, k-Nearest neighbor, naive Bayes come under this category.

Support vector machines

In the case of support vector machines, an instance in the training data is viewed as a n-dimensional vector. And a (n-1) dimensional hyperplane need to be found to separate those data points. Here, say the given data samples are :

$$D = \{(x_i, c_i | x_i \epsilon R^r, c_i \epsilon \{+1, -1\}\}$$

Now each data sample x_i can have either of the two values +1 or -1. And a hyperplane is to be found that separates all these data points in space based on its class. This hyperplane is built such that is maximizes the minimum distance from the separating hyperplane to the nearest data sample.

Naivebayes

In many domains, the performance of naive bayes classifier is comparable to that of neural networks and decision trees. Lets say each sample in the training set is described by a conjunction of attribute values and the decision values is taken from a finite set C. When a new instance is given with the attribute pair $(a_1, a_2, ...a_n)$, the maximum aposterior probability is found by the formula

$$MAP = argmax_{c_j \epsilon C} P(a_1, a_2, ...a_n | c_j) P(c_j)$$

So the class getting the highest maximum aposterior probability is assigned as the target class label.

k-Nearest neighbors

k-Nearest Neighbors is one of the oldest classification algorithms. Here, to classify a new unseen example, the distance from that sample to all the samples in the training data is found using some distance measure like euclidean distance or manhattan distance etc. Out of all the distances, the k smallest distances are found. Out of all these k training samples, the most represented class is found and assigned as the output target class to the new unseen example.

Decision trees

- Infer a split on training data to get a good generalization

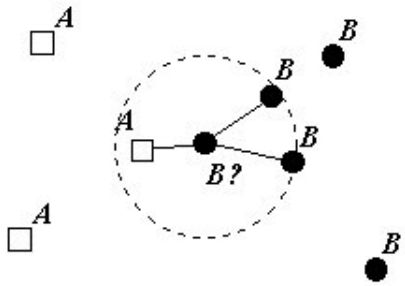

Figure 2.1: k-Nearest Neighbors

- Split made at each node by maximizing Information Gain
- Each leaf node corresponds to an output class label.
- When a new instance is given, it is classified by following the path from the root node to leaf node by testing the sample on some feature at each node. When leaf node is reached, the associated class label is assigned to the instance. A leaf node can refer to any of the K classes concerned. The decision tree algorithm can naturally be used for multiclass problem also.

Entropy

Entropy measures the *impurity* of S

$$Entropy(S) = \Sigma_i - p_i log_2 p_i$$

Information Gain

Gain(S,A) is the expected reduction in entropy due to sorting on A

$$Gain(S, A) = Entropy(S) - \Sigma_{v \epsilon Values(A)} \frac{S_v}{S} Entropy(S_v)$$

Neural networks

Multilayer feed-forward neural networks gives a natural support to multiclass problem. Multilayer feed-forward neural networks have a neuron in the output layer which gives an output 0 or 1 which indicates one of the two classes. For multiclass problem, rather

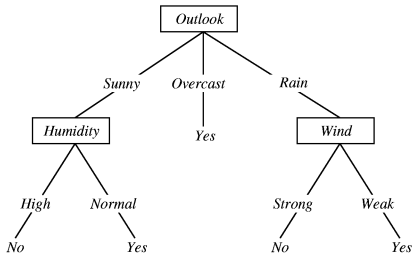

Figure 2.2: Decision Tree

than having one neuron in the output layer, we can have N binary neurons. The output codeword corresponding can be chosen in two ways.

- One-class per Coding
  - Each neuron identifies a given class.
  - Codeword for a class is 1 at that neuron and o for others .
  - So N = K neurons are required in o/p layer
  - When a new instance is given, the neuron with maximum output specifies class label.
  - For example, for a 4-class problem, the output codes looks as follows.

Class1	1000
Class2	0100
Class3	0010
Class4	0001

- Distributed output coding
  - Each class assigned a unique binary codeword from 0 to $(2^N - 1)$ where N specifies the number of neurons.
  - For new examples, output codeword is compared to that of the codewords of K classes, and the Nearest codeword is assigned as the winning class label.

Class1	00000
Class2	00011
Class3	11001
Class4	11110

2.1.2 Converting multiclass problem into binary classification problems

This is another approach to solve multiclass classification problem where the problem can be converted to several binary classification problems which can be solved very efficiently by binary classifiers. These binary classifiers can be anything like Support vector machines, decision trees, naiveBayes classifiers etc. The idea is similar to distributed coding where codewords are assigned to each class and then a number of binary classifiers are used. All the results of the binary classifiers determine the output class label of the new instance. There are several techniques [1] to decompose this multiclass problem into binary classification problems.

One-vs-All

In this technique, we convert the K-class problem into K binary classification problems. Here, each problem tries to classify a given class from all the other classes. So we need N=K binary classifier. K_{th} classifier is trained with the positive examples of class k and all the negative examples of other K-1 classes. When a new sample is given, the class associated with the classifier producing the maximum output is considered as the target class label and is assigned.

All-vs-All

In this technique,we convert the k-class problem into $\frac{K(K-1)}{2}$ binary classification problems. A binary classifier is built between each pair of classes. When a new instance is to be classified, it is given to all the $\frac{K(K-1)}{2}$ classifiers and a voting is performed between them. The class with the maximum votes is assigned as the target class label to the instance.

Unique one-vs-others method

Ding et al [3] proposed a method of combining One-vs-all and all-vs-all and developed a hybrid method which they call Unique one-vs-all method. The standard One-vs-all technique of classification is well known, where K-class problem is converted into K two-class problems where a binary classifier is built between each class and all the other classes.
In *Unique One-vs-Others method*, Ding et al [3] added a second step to one-vs-all method. In this second step, on all the classes with positive predictions, two way classifiers are built.

All the votes from these classifiers are taken and the class with the highest number of votes is considered the winner and assigned as the target class label

2.2 Multiple classifier systems

Normally, in decision making process, people prefer to take the opinions and ideas of multiple experts and based on them, reach a decision. In the same way, a classification system can use several classifiers and reach a decision based on all the outputs of the classifiers. Such a system is called *Multiple classifiers system* [7]. Based on the technique used for the association of the classifier decisions, there are three strategies.

Taxonomy of Multiple classifier systems :

- Combination of classifiers
- Cooperation of classifiers
- Selection of classifiers

2.2.1 Combination of classifiers

This strategy involves deciding using different opinions. This technique can be used in many fields like image processing, pattern recognition, face recognition etc. In those areas, the decision given by one sensor cannot be accurate because of noise etc. So its better not to take a stand based on that particular sensor. Rather use all the sensors and devise a system to fusion the results of all the sensors.

Lets say x_1, x_2, x_3 be the input features concerning a particular pattern, e_1, e_2, e_3 be three classifiers taking inputs and producing decisions l_1, l_2, l_3. These decisions are then combined using some mathematical technique and a final decision l is achieved.

Voting schemes

Different combination techniques are devised based on the classifier output.

If a classifier gives crisp decisions, assigning a class to the input instance, then a simple voting scheme can be performed. In this voting scheme, each classifier decision is considered

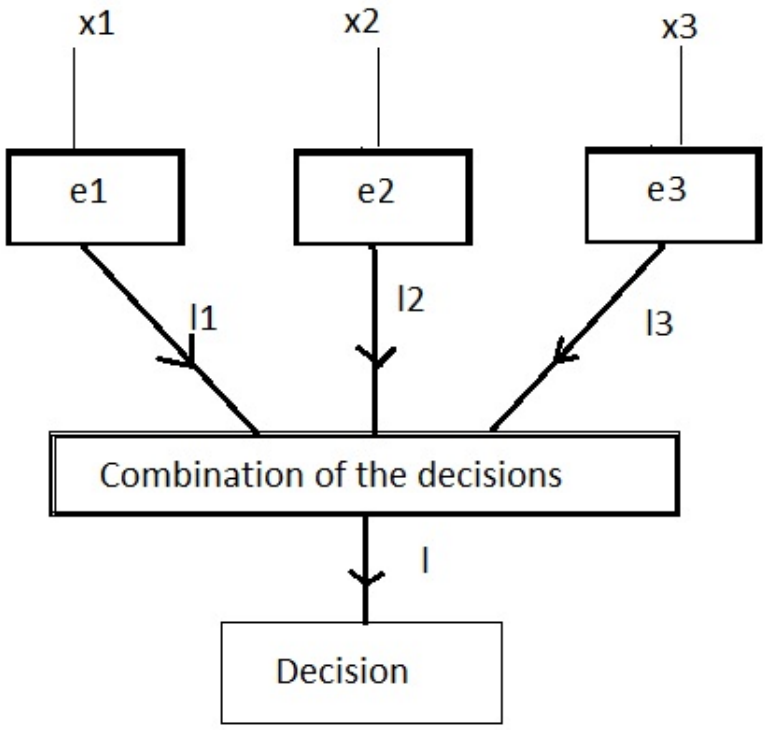

Figure 2.3: Combination of classifiers

as a vote for that class. In the end, which class get the highest votes, that particular class is assigned as the target class label to that instance.

Borda Count If a classifier gives a ranked decision, each class is ranked according to their preference. A ranking gives a class some points and Borda count [6] adds the points given to each class and picks the winner class with maximum points.

Other combinations Some other combination techniques exist like

- Combination by maximum rule
- Combination by minimum rule
- Combination by mean rule
- Combination by product rule
- Combination by probability theory
- Combination by belief theory

2.2.2 Cooperation of classifiers

In this strategy, the decisions of one or more classifiers are used to help or guide other classifiers to take decisions. So some information is passed in between the classifiers, unlike combination of classifiers where no information is passed between classifiers. Here, the information exchanged is nothing but the decision vector. Multilevel hierarchical classifiers used in neural networks can be viewed as cooperation of classifiers where classifiers in one level drive the decisions of classifiers in other levels. In this Figure 2.4, x_1, x_2, x_3 are the input features concerning a particular pattern, e_1, e_2, e_3 are three classifiers taking inputs and they exchange decision vectors and give decisions l_1, l_2, l_3.

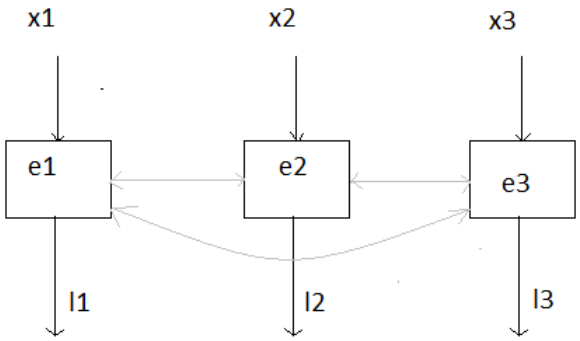

Figure 2.4: Cooperation of classifiers

2.2.3 Selection of classifiers

Generally, the best classifier changes depending on the problem and instance. So in this strategy, we intend to find which classifier will perform best for which kind of patterns. So initially, the training set is clustered into different partitions. And for each partition, different classifiers are used and tested to find the best classifier for each partitiond. Then when a new unseen instance is given, first it is found in which partition it will fall and then the best classifier for that partition is used to classify this new instance.

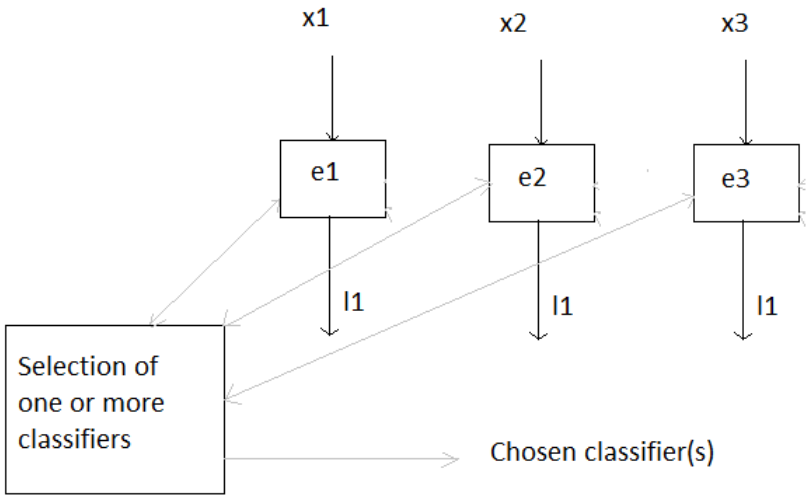

Figure 2.5: Selection of classifiers

2.3 Previous work on enzyme classification

Enzyme classification is a very challenging machine learning problem. Over the years, the number of protein structures are increasing rapidly. The enzyme classification scheme is a hierarchical system of organizing proteins into the six main classes hydrolases, isomerases, ligases, lyases, oxidoreductases and transferases. It is difficult to predict the enzyme class based on a particular set of features. Bray et al [2] achieved a highest prediction accuracy of **33.1%** using 74 features that are calculated for each enzyme which include *structural features, sequence features* and *size-associated features* of the whole protein as well as the active-sites. In this thesis, we investigate each of the methods listed in this chapter for the EC class problem.

Chapter 3

Classification using combination of classifiers

3.1 Dataset

The enzyme dataset of binding site containing the composition of 20 amino-acids as features is extracted from the Protein ligand interaction database(PLID) [4].

Dataset Used	:	PLID dataset
No. of Classes	:	6
Classes	:	Hydrolase, Isomerase, Ligase, Lyase, Oxidoreductase, Transferase

3.1.1 Initial binding site dataset :

S.No	Class name	-	No. of samples
1	Hydrolase	-	364
2	Isomerase	-	64
3	Ligase	-	36
4	Lyase	-	109
5	Oxidoreductase	-	383
6	Transferase	-	395
	Total	-	1351

Table 3.1: Initial binding site dataset

3.1.2 Addition of binding site features

2008 binding site dataset

Two new features of binding site - Fraction of Contact and Tightness were calculated and added to the existing 20 amino acid features. And the dataset is normalized and identical binding pockets are removed.

2010 binding site dataset

A new recent dataset is also extracted from the PLID [4] that contains the amino acid composition, fraction of contact and tightness values.

S.No	Class name	2008 dataset	2010 dataset
1	Hydrolase	355	742
2	Isomerase	62	120
3	Ligase	36	88
4	Lyase	107	158
5	Oxidoreductase	382	415
6	Transferase	390	686
	Total	1332	2209

Table 3.2: Binding site dataset with contact and tightness

3.1.3 Whole protein dataset

2008 Whole protein dataset :

Whole protein amino acid composition is obtained for all the proteins present in 2008 binding site dataset.

2010 Whole protein dataset :

A new recent dataset containing the amino acid composition of whole protein is also extracted.

S.No	Class name	2008 dataset	2010 dataset
1	Hydrolase	269	713
2	Isomerase	48	129
3	Ligase	31	94
4	Lyase	81	169
5	Oxidoreductase	252	430
6	Transferase	306	724
	Total	987	2259

Table 3.3: Whole protein dataset

3.2 Combination of classifiers by probability [7]

Combination of classifiers is a technique of getting an association among different classifiers to reach a single reliable decision.

Let $y_1, y_2, ...y_s$ be the decisions provided by classifiers $e_1, e_2, ...e_s$ for pattern x_k. Then $P(C_i)$ gives the prior probability of class C_i. $P(y_s|C_i)$ gives the probability that the decision of classifier s i.e.,y_s holds given a class C_i.

Then the combination information concerning the class of x_k represented by aposterior probability is given by

$$P(C_i|y_1, y_2, ...y_s) = \frac{P(C_i)\prod_{s=1}^{S} P(y_s|C_i)}{\sum_{n=1}^{l} P(C_n)\prod_{s=1}^{S} P(y_s|C_n)}$$

Here $P(C_i|y_1, y_2, ...y_s)$ gives the probability that the instance x_k belongs to class C_i given the individual classifier decisions$y_1, y_2, ...y_s$. And then, The class label is assigned as follows

$$SC(x_k) = \begin{cases} C_j, & if P(C_i|y_1, y_2, ...y_s) = max_{i=1}^{l} P(C_i|y_1, y_2, ...y_s) \\ & P(C_i|y_1, y_2, ...y_s) \geq T \\ C_{l+1}, & otherwise \end{cases}$$

3.2.1 Results

In our experimentation, each instance is given to three different classifiers - support vector machines, k-nearest neighbor and naive bayes classifier. Their decisions are taken and combination by probability theory technique is applied. For example,

- Sample 257 is given to SVM,NB and KNN.

  Output of SVM is Hydrolase

Output of NB is Oxidoreductase

Output of KNN is Hydrolase

Here,

$y_1 = 1$, $y_2 = 6$, $y_3 = 1$

So, to find the combination probabilities of each class

$$P(C_1).P(y_1|C_1).P(y_2|C_1).P(y_3|C_1) = \frac{242}{900}\frac{182}{242}\frac{62}{242}\frac{177}{242} = 0.03789$$

$$P(C_2).P(y_1|C_2).P(y_2|C_2).P(y_3|C_2) = \frac{42}{900}\frac{17}{42}\frac{14}{42}\frac{6}{42} = 0.00089$$

$$P(C_3).P(y_1|C_3).P(y_2|C_3).P(y_3|C_3) = \frac{24}{900}\frac{8}{24}\frac{6}{24}\frac{5}{24} = 0.00046$$

$$P(C_4).P(y_1|C_4).P(y_2|C_4).P(y_3|C_4) = \frac{72}{900}\frac{36}{72}\frac{16}{72}\frac{21}{72} = 0.00259$$

$$P(C_5).P(y_1|C_5).P(y_2|C_5).P(y_3|C_5) = \frac{256}{900}\frac{70}{256}\frac{96}{256}\frac{36}{256} = 0.00410$$

$$P(C_6).P(y_1|C_6).P(y_2|C_6).P(y_3|C_6) = \frac{264}{900}\frac{60}{264}\frac{107}{264}\frac{34}{264} = 0.00347$$

So, $\sum_{n=1}^{l} P(C_n) \prod_{s=1}^{S} P(y_s|C_n) = 0.049429$

$$P(C_1|y_1, y_2, y_3) = 0.766$$

$$P(C_2|y_1, y_2, y_3) = 0.018$$

$$P(C_3|y_1, y_2, y_3) = 0.009$$

$$P(C_4|y_1, y_2, y_3) = 0.052$$

$$P(C_5|y_1, y_2, y_3) = 0.082$$

$$P(C_6|y_1, y_2, y_3) = 0.070$$

So, Class 1 i.e., **Hydrolase**, whose value came highest, is assigned as classlabel to the input. The example is shown in Figure 3.1.

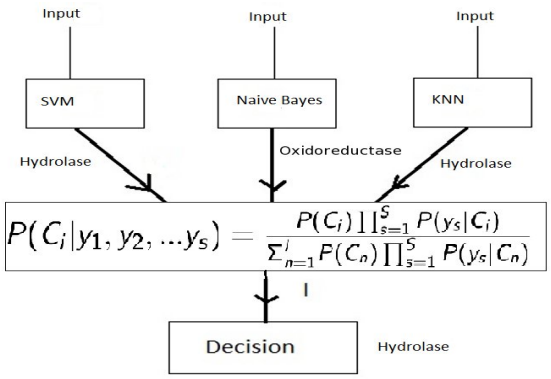

Figure 3.1: Combination technique : Example

Combination technique on initial binding site dataset

The results of applying combination technique on the PLID dataset is shown Table 3.4. Also individual classifier accuracies(in %) are specified.

Classifier Used	Hydrolase	Isomerase	Ligase	Lyase	Oxido-reductase	Transferase	Average Accuracy
SVM	63	0	0	0	46.4	53.4	27.1
NB	9.8	1.5	5.	18.5	12.5	24.5	19.5
KNN	46.4	7.8	11.1	3.7	43.3	40.7	24.6
SVM, NB	62.2	0	0	0	47	52.5	26.9
SVM, KNN	41.8	4.7	5	5	50.6	39.8	16.9
NB, KNN	42.6	9.5	2	6	44.9	39.3	24
SVM, NB & KNN	41.8	7.9	5	3.7	53.25	44.19	28.53

Table 3.4: Combination technique on initial binding site dataset

The results of applying combination technique using 5-fold cross validation on the PLID dataset is shown in Table 3.5.

Classifier Used	Normal classification	Using 5-fold cross validation
SVM	27.1	26.46
NB	19.5	19.4
KNN	24.6	26.21
SVM, NB	26.9	27.9
SVM, KNN	16.9	23.9
NB, KNN	24	23.6
SVM, NB & KNN	28.53	24.48

Table 3.5: Combination with 5-fold cross validation on initial binding site dataset

Here, the combination technique is not producing any good results on the initial dataset. So, the same technique is applied on 2008 and 2010 binding site datasets and whole protein datasets. *All the experiments are done with 5-fold cross validation.*

Results of combination technique on 2008 and 2010 datasets

The results of applying combination technique using 5-fold cross-validation on the 2008 and 2010 binding site dataset(with contact and tightness) as well as 2008 and 2010 whole protein datasets are shown in Table 3.6. Also individual classifier accuracies are specified. All the experiments are done with 5-fold cross validation.

From Table 3.6, it can be observed that combining classifiers is making an incremental improvement, not a significant improvement on binding site with SVM,NB & KNN combination achieving 29.5% as compared with SVM alone with 28.3% on 2008 binding site dataset. Certainly, combination of classifiers seems to be improving the results for whole protein dataset with single SVM achieving 43% when compared to combination of classifiers achieving 50%.

Classifier Used	binding site 2008	whole protein 2008	binding site 2010	whole protein 2010
SVM	28.3	28.08	34.0	43.04
NB	23.9	26.51	26.8	32.12
KNN	26.3	20.97	29.9	42.65
SVM, NB	27.8	27.84	34.0	43.14
SVM, KNN	29.36	25.19	38	48.73
NB, KNN	27.7	20.5	33.8	49.04
SVM, NB & KNN	29.50	26.2	37.24	50.12

Table 3.6: Combination results on 2008 and 2010 datasets

3.3 Unbalanced datasets

The above mentioned dataset is highly unbalanced where ligase, isomerase and lyase have much lower number of samples compared to hydrolase, oxidoreductase and transferase. As a result, a number of false positives occur as the higher classes dominate the predictions of lower classes.

3.3.1 Make dataset balanced

To avoid this problem of imbalance of classes in the dataset, we first find the least instance class (in this case, it is Ligase).And we take only that many instances from every class as there are in the least instance class (n = no. of samples in least instance class). This process is repeated in three sets. In all the sets, all the samples from least instance class are taken. For other classes, take different n samples for different sets.

3.3.2 Results

Results of balanced binding site dataset

The results of applying combination technique using 5-fold cross validation on the 2008 and 2010 Balanced PLID dataset with Contact and Tightness is shown in Table 3.7. Firstly, balancing the data itself is improving the accuracy to 30.6% which is comparable to the results of Bray et al [2]. It can also be seen that combining classifiers do not have any positive impact on the results.

Classifier Used	2008 dataset with contact and tightness			2010 dataset with contact and tightness		
	Set1	Set2	Set 3	Set1	Set2	Set 3
SVM	31.5	30.2	30.5	40.9	43.6	37.3
NB	21.7	23.2	25.4	31.9	33.3	27.1
KNN	23.1	29.6	29.6	29.0	33.3	30.7
SVM, NB	30.6	29.6	29.0	40.7	43.4	37.3
SVM, KNN	29.2	26.9	30.5	39.8	43	35.4
NB, KNN	21.7	25.7	24.5	30.3	33	31.5
SVM, NB & KNN	30.1	29.6	29.5	38.3	43.4	35.7

Table 3.7: Combination results on balanced binding site datasets

Results of balanced whole protein dataset

The results of applying combination technique using 5-fold cross validation on the 2008 and 2010 Whole protein dataset is shown in Table 3.8. Also individual classifier accuracies

are specified.

Classifier Used	2008 whole protein			2010 whole protein		
	Set1	Set2	Set 3	Set1	Set2	Set 3
SVM	36.59	34.3	26.5	41.68	38.9	40.1
NB	29.37	33.2	21.5	31.79	30.8	33.5
KNN	16.67	16.8	17.1	36.13	36.2	35.1
SVM, NB	36.11	33.2	26.5	41.33	38.9	40.3
SVM, KNN	33.97	29.9	26.7	41.67	38.2	40.1
NB, KNN	21.51	26.3	24.3	38.33	36.4	39.2
SVM, NB & KNN	36.03	30.6	29.9	40.77	38.6	40.1

Table 3.8: Combination results on balanced whole protein datasets

3.3.3 Top ranking accuracies

To improve the classifier results, the Combination of classifiers is modified to take the top k ranks.

In this method of Top k Ranking [2], we first find the probabilities that an instance belongs to a particular class. Out of all the 6-class probabilities, rather than assigning the top probability class, consider top k probable classes and if the target class is either of those k classes, consider it as a successful positive classification. Otherwise, it is not a successful classification. For example, in Top 2 Ranking, we consider top 2 probable classes.

The above mentioned Top ranking scheme is tested on balanced 2008 dataset with top 2 and 3 ranks by taking equal trainsets. The results are shown in Table 3.9 and Table 3.10 .

These experiments give us a further insight into the amount of error that the classifier is making in labeling a protein. Considering top 2 ranks is increasing the accuracy of classifier from 30% to about 50% which is a significant increase in classification performance.

Classifier Used	Dataset with contact and tightness		
	Set1	Set2	Set 3
SVM, NB	50.5	52.9	47.5
SVM, KNN	45.7	45.7	51
NB, KNN	43	46.3	42.6
SVM, NB & KNN	48.2	54.5	47.8

Table 3.9: Combination results with top 2 ranks

Classifier Used	Dataset with contact and tightness		
	Set1	Set2	Set 3
SVM, NB	61.5	67.7	61
SVM, KNN	58.7	59	63.4
NB, KNN	55.8	60.8	61.5
SVM, NB & KNN	61.2	66	61.8

Table 3.10: Combination results with top 3 ranks

3.3.4 Substitution group features

As an additional experiment on features, we did one experiment evaluating substitution group features for this classification problem. There are 6 substitution groups into which all the 20 amino acids of proteins are grouped into. So classification of proteins can be done based on the substitution group features also.
The information about substitution groups is given in Tabel 3.11

Results

The substitution group features are built for 2008 binding site dataset. Combination of classifiers by probability techniques is applied on the dataset with substitution group features along with contact and tightness. And the accuracies(in %) were observed as shown

Substitution group	Associated amino acids
Group 1	HIS(H), ARG(R), LYS(K)
Group2	ASP(D), GLU(E), ASN(N). GLN(Q)
Group3	CYS(C)
Group4	SER(S), THR(T), PRO(P), ALA(A), GLY(G)
Group5	MET(M), ILE(I), LEU(L), VAL(V)
Group6	PHE(F), TYR(Y), TRP(W)

Table 3.11: Substitution group features

in Table 3.12. Clearly, substitution group features are not helping to improve the classification performance.

Classifier Used	Substitution Group features with contact and tightness		
	Set1	Set2	Set 3
SVM	27.7	33.9	23.7
NB	27.4	28.5	24.2
KNN	20.9	27.4	26.8
SVM, NB	28.6	33.9	25
SVM, KNN	24.1	31.6	24.2
NB, KNN	20.9	24.7	25.1
SVM, NB & KNN	27.4	33.6	25.6

Table 3.12: Combination results on substitution group features

Chapter 4

Classification with expert classifiers on sequence features

4.1 Extraction of sequence features

Till now, experiments were done on the datasets with amino acid composition, fraction of contact and tightness as features . Now, some more sequence features were calculated. The new features calculated were :

- molecular weight
- length
- average residue weight
- charge
- isoelectricpoint
- normalized hydrophobic score

The above features were calculated for both the binding site and whole protein datasets of 2008. So now the binding site dataset becomes a 28-feature dataset along with amino acid composition, fraction of contact and tightness for binding site. And the whole protein dataset becomes a 26-feature dataset.

4.2 Review of unique one-vs-all method [3]

The standard one-vs-others technique of classification is well known, where K-class problem is converted into K two-class problems where a binary classifier is built between each class and all the other classes. When a test instance is given, ideally only one positive label should be obtained. But it may not be the case and ambiguous decision vector can be obtained.

For those cases, Ding et al [3] proposed *Unique One-vs-Others method* where a second step is added. In this second step, on all the classes with positive predictions, two-way classifiers are built on all the pairs of positively predicted classes. All the votes from these classifiers are taken and the class with the highest number of votes is considered the winner and assigned as the target class label.

4.3 Implementation and results

For each class, first build an expert classifier. For that, take all the samples of that class and take an equal percentage of samples from all the rest of the classes to make an equally approximate negative sample set. Now take $(\frac{1}{6})^{th}$ of each class as validation set. And the rest is used for training and testing. For example, in 2008 binding site dataset with sequence features, there are 355 samples belonging to Hydrolase class. So for building expert classifiers for Hydrolase, take all 355 samples of Hydrolase as positive class samples. And for negative class samples, take 37% from all the remaining 5 classes to make up around 355 samples. This 355 samples from Hydrolase and the other 355 samples from remaining 5 classes becomes the dataset for building expert classifier for Hydrolase class. In the same way, for each class, a svm , naivebayes and a knn classifier are built on that class against the remaining five classes for 2008 binding site and whole protein datasets and the accuracies(in %) are observed as in Table 4.1 and Table 4.2.

For 2008 binding site dataset with sequence features :

Expert classifiers for Hydrolase			
Classifier	Hydrolase	Non-Hydrolase	AverageAccuracy
SVM	61.4	68.1	**64.8**
NB	47.7	76.6	62.2
KNN	44.2	74.8	57.5
Expert classifiers for Isomerase			
Classifier	Isomerase	Non-Isomerase	AverageAccuracy
SVM	37.6	59.4	48.5
NB	32.6	59.0	45.8
KNN	50.7	64.3	**57.5**
Expert classifiers for Ligase			
Classifier	Ligase	Non-Ligase	AverageAccuracy
SVM	52.1	77.0	**64.5**
NB	49.3	71.5	60.4
KNN	36.4	66.5	51.5
Expert classifiers for Lyase			
Classifier	Lyase	Non-Lyase	AverageAccuracy
SVM	52.4	57.3	**54.9**
NB	36.5	65.5	51
KNN	43.7	52.7	48.2
Expert classifiers for Oxidoreductase			
Classifier	Oxidoreductase	Non-Oxidoreductase	AverageAccuracy
SVM	70.0	61.8	**65.9**
NB	43.5	77.2	60.4
KNN	55.2	71.7	63.5
Expert classifiers for Transferase			
Classifier	Transferase	Non-Transferase	AverageAccuracy
SVM	61.4	67.7	**64.5**
NB	43.1	77.7	60.4
KNN	55.1	68.2	61.7

Table 4.1: Building expert classifiers for 2008 binding site dataset with sequence features

Here, for hydrolase, svm is performing best with 64.8% accuracy.

For 2008 whole protein dataset with sequence features:

Expert classifiers for Hydrolase			
Classifier	Hydrolase	Non-Hydrolase	AverageAccuracy
SVM	63.1	74.3	**68.7**
NB	41.8	86.3	64
KNN	49.8	55.3	52.5
Expert classifiers for Isomerase			
Classifier	Isomerase	Non-Isomerase	AverageAccuracy
SVM	35.0	54.4	44.7
NB	45.0	43.6	44.3
KNN	50.0	61.1	**55.5**
Expert classifiers for Ligase			
Classifier	Ligase	Non-Ligase	AverageAccuracy
SVM	60.0	78.2	**69.1**
NB	52.0	74.8	63.4
KNN	44.0	58.7	51.3
Expert classifiers for Lyase			
Classifier	Lyase	Non-Lyase	AverageAccuracy
SVM	58.6	66.6	**62.6**
NB	78.6	42.9	60.8
KNN	52.9	49.9	51.4
Expert classifiers for Oxidoreductase			
Classifier	Oxidoreductase	Non-Oxidoreductase	AverageAccuracy
SVM	60.5	62.5	**61.5**
NB	81.4	38.0	59.7
KNN	57.6	53.4	55.5
Expert classifiers for Transferase			
Classifier	Transferase	Non-Transferase	AverageAccuracy
SVM	69.0	73.7	**71.3**
NB	47.1	79.2	63.1
KNN	57.8	57.6	57.7

Table 4.2: Building expert classifiers for 2008 whole protein dataset with sequence features

Here, for isomerase, knn is performing best with 55.5% accuracy.

4.4 Combination of expert classifiers

4.4.1 Expert classifier for each enzyme class

Now, out of all the classifiers built for each class against all other classes, select the classifier that performs best in identifying that class against the other classes. These will be the expert classifiers for the corresponding classes.

Class	Corresponding Expert	
	Binding site with sequence features(2008)	Whole protein sequence features(2008)
Hydrolase	SVM	SVM
Isomerase	KNN	KNN
Ligase	SVM	SVM
Lyase	SVM	SVM
Oxidoreductase	SVM	SVM
Transferase	SVM	SVM

Table 4.3: Final experts for 2008 datasets with sequence features

In the Table 4.3, it is described how each class is associated with an expert classifier of its own. So , when a new instance is given, it is given to all the 6 expert classifiers , say, Expert_Hydrolase_SVM, Expert_Isomerase_KNN, Expert_Ligase_SVM, Expert_Lyase_SVM, Expert_Oxidoreductase_SVM and Expert_Transferase_SVM.

4.4.2 Results of combination of expert classifiers :

The decisions from all the expert classifiers will be 1s or 0s telling if the instance belongs to their class or not. For example, if a decision vector is (1,0,0,0,1,0) means that Expert_Hydrolase_SVM tells the instance belongs to hydrolase and Expert_Oxidoreductase_SVM tells the instance belongs to oxidoreductase. Now we need to resolve among these two classes.

- If the decision vector has a single 1, then the corresponding class is assigned as the target class label.
- If the decision vector has more than one 1s, then combination technique by accuracy is applied.
  - The testset accuracies of each expert classifier is noted.
  - When the decision vector gives more than one 1s, out of the classes getting a positive prediction, which class has got the highest testset accuracy is found.
  - That corresponding class is assigned as the target class label.

This method is clearly not working for this dataset, which can be seen from the results in Table 4.4. Hence, there is a need for refining this approach.

Class	Binding site with sequence features(2008)	Whole protein sequence features(2008)
Hydrolase	51.6	29.5
Isomerase	0	0
Ligase	0	0
Lyase	1.5	
Oxidoreductase	22.7	40.4
Transferase	50.7	80.3
Average	21.1	26.5

Table 4.4: Combination of expert classifiers of 2008 datasets with sequence features

4.5 Refined approach : expert classifiers followed by n_{C_2} classifiers

One refinement to the combination of expert classifiers can be done [3] by adding a second step.

- Give the new instance to expert classifiers

- If the decision vector of expert classifiers has a single 1, give the corresponding class as the target class label.

- Otherwise, between all the classes that got positive predictions by experts, build two-way classifiers i.e., if there are n 1s in the decision vector, n_{C_2} two-way classifiers are built (NOTE: if the decision vector has all 0s or all 1s, then two-way classifiers are built between all the classes) and the instance is given to these n_{C_2} classifiers and the votes are collected. Now, the class having the maximum votes is assigned as the target class label.

4.5.1 Results of binding site with sequence features:

Class	only AA	AA + Seq. features	AA+Contact +Tightness	AA+Seq.features+ Contact+Tightness
Hydrolase	18.3	26.6	23.3	26.6
Isomerase	0	27.2	18.1	27.2
Ligase	50.0	33.3	50.0	33.3
Lyase	17.6	17.6	11.7	17.6
Oxidoreductase	43.5	43.5	38.7	45.1
Transferase	41.5	36.9	30.7	27.6
Average	28.5	30.8	28.7	29.6

Table 4.5: Refined approach results on binding site with sequence features

The above mentioned method is applied on different combinations of features of the dataset like

- only amino acid composition

- amino acid composition and sequence features

- amino acid composition, fraction of contact and tightness

- complete feature set : amino acid composition, sequence features, fraction of contact and tightness

The results are shown in Table 4.5. The whole idea of experts by Ding et al [3] seems to be working on the dataset represented by amino acids and sequence features (AA+Seq.features) with an accuracy of 30.8%.

4.5.2 Results of whole protein with sequence features:

The method specified in the previous section is used on this 2008 whole protein dataset with sequence features. The resulting accuracies are as follows :

Class	Whole protein with sequence features
Hydrolase	38.6
Isomerase	12.5
Ligase	50.0
Lyase	45.5
Oxidoreductase	35.7
Transferase	39.2
Average	36.9

Table 4.6: Refined approach results on whole proteins with sequence features

It can be observed that adding n_{C_2} classifiers in the second stage improves the classification rate significantly from 26.5% to 36.9%.

4.6 Using 6_{C_2} all-vs-all classifiers

In the above method, Expert classifiers are followed by n_{C_2} classifiers. This can be generalized and the expert classifiers step can be removed completely. And all-versus-all method can be used. Here, simply two-way classifiers are to be built between all the classes. These classifiers are built on training sets with equal number of samples from both classes. When a new instance comes, it is given to all these $\frac{K(K-1)}{2}$ classes, where K is the total no. of classes. And the results of all these classifiers are obtained and based on the majority voting, whichever class got the highest number of votes, that class is assigned as the target class label.

The results of applying these all-vs-all classifiers are shown in Table 4.7. Considering 6_{C_2} classifiers alone on binding site dataset is achieving 25.9% accuracy as compared to 30.8% for refined approach of experts followed by n_{C_2}.

Class	Binding site with sequence features(2008)	Whole protein sequence features(2008)
Hydrolase	50.0	36.3
Isomerase	27.2	25.0
Ligase	50.0	83.3
Lyase	11.7	81.8
Oxidoreductase	37.0	33.3
Transferase	24.6	27.4
Average	25.9	47.8

Table 4.7: Results of 6_{C_2} classifiers on datasets with sequence features

4.7 Expert classifiers based on different feature sets

Previously, classification is done using a single feature set where an expert classifier is built on that feature set. Now we use different feature sets [3] from the existing dataset to build different expert classifiers for the same class. For the binding site dataset with sequence features, it is already learnt from the previous section that the expert classifier for Hydrolase is SVM. Now we need to build SVM on Hydrolase-vs-remaining classes. Rather than building one SVM classifier, we build four different SVM classifiers using the following different feature sets.

- SVM on only Amino acid composition
- SVM on Amino acid composition, fraction of contact, tightness
- SVM on Amino acid composition, sequence features
- SVM on Amino acid composition, sequence features, fraction of contact, tightness

For whole protein dataset with sequence features also, different feature sets are made use of to build different expert classifiers for the same class. Here, two feature sets are possible. For example, when we need to build SVM on Hydrolase-vs-remaining classes, rather than building one SVM classifier, we build two different SVM classifiers using the following different feature sets.

- SVM on only Amino acid composition
- SVM on Amino acid composition, sequence features

4.7.1 Classification by voting

- For binding site dataset, for each class, 4 expert classifiers are built. So for 6 classes, a total of 24 classifiers are built. So when a new instance to be classified is given, it is given to all the 24 classifiers and cumulative voting is performed.
- For whole protein dataset, for each class, 2 expert classifiers are built. So for 6 classes, a total of 12 classifiers are built. So when a new instance to be classified is given, it is given to all the 12 classifiers and cumulative voting is performed.

And based on maximum voting rule, the class with highest number of votes is assigned as the target class label. The results are shown in Table 4.8.
Just as the saying *too many cooks spoil the broth*, this effort of using the knowledge of 24 different experts to arrive at a decision on the class has not worked.

4.7.2 Expert classifiers followed by n_{C_2} classifiers

In first stage of this method, for binding site, 24 expert classifiers(4 experts on different feature sets for each class) are built and for whole protein dataset, 12 expert classifiers(2 experts on different feature sets for each class) are built.
And when the new unseen instance is given, first it is given to all the 24 expert classifiers (12 expert classifiers in the case of whole protein dataset) and cumulative voting is done. If a unambiguous decision if obtained, then the positively predicted class is assigned as the target class label. If an ambiguous decision vector is obtained, a second stage of two-way classifiers between each pair of classes n which got a positive prediction in first stage is added. Again the instance is given to n_{C_2} classifiers and based on majority voting rule, the winning class label is found. The results are shown in Table 4.9.

Class	Binding site with sequence features(2008)	Whole protein sequence features(2008)
Hydrolase	21.6	11.3
Isomerase	0	12.5
Ligase	0	0
Lyase	11.7	18.1
Oxidoreductase	40.32	47.6
Transferase	58.4	74.5
Average	22.03	27.3

Table 4.8: Results of expert classifiers using different feature sets

Class	Binding site with sequence features(2008)	Whole protein sequence features(2008)
Hydrolase	28.3	29.5
Isomerase	9.0	25.0
Ligase	33.3	50.0
Lyase	11.7	45.4
Oxidoreductase	41.9	38.0
Transferase	30.7	31.3
Average	25.8	36.5

Table 4.9: Results of refined approach on different feature sets

After doing vast experimentation on datasets with sequence features, we come to a conclusion that sequence features are not helping to improve the classification rate. So we perform all experiments using the methods listed in this chapter on the basic framework of features i.e., amino acid composition. This experimentation is shown in the next chapter.

Chapter 5

Classification with expert classifiers on binding site features

5.1 Unique one-vs-others method [3]

Ding et al [3] proposed *Unique One-vs-Others method* where a second step is added to one-vs-all method. In this second step, on all the classes with positive predictions, two way classifiers are built on all the pairs of positively predicted classes. All the votes from these classifiers are taken and the class with the highest number of votes is considered the winner and assigned as the target class label.

5.2 Implementation and results

For each class, first build training set by taking all the samples of that class and an equal percentage of samples from all the rest of the classes to make an equally approximate negative sample set.

Using those training sets, for each class, a svm , naivebayes and a knn classifier are built on that class against the remaining five classes and the accuracies(in %) on 2008 and 2010 binding site and whole protein datasets are shown in Table 5.1, Table 5.2, Table 5.3 and Table 5.4.

For 2008 binding site dataset with contact and tightness :

Expert classifiers for Hydrolase			
Classifier	Hydrolase	Non-Hydrolase	AverageAccuracy
SVM	61.4	68.1	**64.8**
NB	47.7	76.6	62.2
KNN	44.2	74.8	57.5
Expert classifiers for Isomerase			
Classifier	Isomerase	Non-Isomerase	AverageAccuracy
SVM	37.6	59.4	48.5
NB	32.6	59.0	45.8
KNN	50.7	64.3	**57.5**
Expert classifiers for Ligase			
Classifier	Ligase	Non-Ligase	AverageAccuracy
SVM	52.1	77.0	**64.5**
NB	49.3	71.5	60.4
KNN	36.4	66.5	51.5
Expert classifiers for Lyase			
Classifier	Lyase	Non-Lyase	AverageAccuracy
SVM	52.4	57.3	**54.9**
NB	36.5	65.5	51
KNN	43.7	52.7	48.2
Expert classifiers for Oxidoreductase			
Classifier	Oxidoreductase	Non-Oxidoreductase	AverageAccuracy
SVM	70.0	61.8	**65.9**
NB	43.5	77.2	60.4
KNN	55.2	71.7	63.5
Expert classifiers for Transferase			
Classifier	Transferase	Non-Transferase	AverageAccuracy
SVM	61.4	67.7	**64.5**
NB	43.1	77.7	60.4
KNN	55.1	68.2	61.7

Table 5.1: Building expert classifiers for 2008 Binding site dataset with contact and tightness

For example, in the case of hydrolase, svm is performing best with 64.8% accuracy.

For 2008 whole protein dataset :

Expert classifiers for Hydrolase			
Classifier	Hydrolase	Non-Hydrolase	AverageAccuracy
SVM	63.1	74.3	**68.7**
NB	41.8	86.3	64
KNN	49.8	55.3	52.5
Expert classifiers for Isomerase			
Classifier	Isomerase	Non-Isomerase	AverageAccuracy
SVM	35.0	54.4	44.7
NB	45.0	43.6	44.3
KNN	50.0	61.1	**55.5**
Expert classifiers for Ligase			
Classifier	Ligase	Non-Ligase	AverageAccuracy
SVM	60.0	78.2	**69.1**
NB	52.0	74.8	63.4
KNN	44.0	58.7	51.3
Expert classifiers for Lyase			
Classifier	Lyase	Non-Lyase	AverageAccuracy
SVM	58.6	66.6	**62.6**
NB	78.6	42.9	60.8
KNN	52.9	49.9	51.4
Expert classifiers for Oxidoreductase			
Classifier	Oxidoreductase	Non-Oxidoreductase	AverageAccuracy
SVM	60.5	62.5	**61.5**
NB	81.4	38.0	59.7
KNN	57.6	53.4	55.5
Expert classifiers for Transferase			
Classifier	Transferase	Non-Transferase	AverageAccuracy
SVM	69.0	73.7	**71.3**
NB	47.1	79.2	63.1
KNN	57.8	57.6	57.7

Table 5.2: Building experts for 2008 whole protein dataset

For example, knn is performing best for isomerase with an accuracy of 55.5% **For 2010**

binding site dataset with contact and tightness :

Expert classifiers for Hydrolase			
Classifier	Hydrolase	Non-Hydrolase	AverageAccuracy
SVM	69.7	77.9	**73.8**
NB	68.7	63.8	66.2
KNN	68.4	67.4	67.9
Expert classifiers for Isomerase			
Classifier	Isomerase	Non-Isomerase	AverageAccuracy
SVM	55.0	69.6	**62.3**
NB	38.0	71.6	54.8
KNN	47.0	68.0	57.5
Expert classifiers for Ligase			
Classifier	Ligase	Non-Ligase	AverageAccuracy
SVM	60.0	76.0	**68.0**
NB	61.3	52.0	56.6
KNN	64.0	53.3	58.7
Expert classifiers for Lyase			
Classifier	Lyase	Non-Lyase	AverageAccuracy
SVM	43.1	78.9	**61.0**
NB	36.9	72.5	54.7
KNN	46.2	66.7	56.5
Expert classifiers for Oxidoreductase			
Classifier	Oxidoreductase	Non-Oxidoreductase	AverageAccuracy
SVM	72.2	63.6	**67.9**
NB	53.9	70.9	62.4
KNN	57.4	61.6	59.5
Expert classifiers for Transferase			
Classifier	Transferase	Non-Transferase	AverageAccuracy
SVM	77.9	68.8	**73.3**
NB	53.0	76.8	64.9
KNN	54.9	76.7	65.8

Table 5.3: Building experts for 2010 binding site dataset with contact and tightness

For example, svm is performing best for Lyase with an accuracy of 61.0%

For 2010 whole protein dataset :

Expert classifiers for Hydrolase			
Classifier	Hydrolase	Non-Hydrolase	AverageAccuracy
SVM	72.9	84.7	**78.8**
NB	57.3	85.7	71.5
KNN	69.4	79.2	74.3
Expert classifiers for Isomerase			
Classifier	Isomerase	Non-Isomerase	AverageAccuracy
SVM	59.1	61.9	60.5
NB	66.4	50.5	58.5
KNN	63.6	62.9	**63.2**
Expert classifiers for Ligase			
Classifier	Ligase	Non-Ligase	AverageAccuracy
SVM	71.2	61.3	**66.2**
NB	70	48	59
KNN	83.8	48.0	65.9
Expert classifiers for Lyase			
Classifier	Lyase	Non-Lyase	AverageAccuracy
SVM	64.3	74.5	**69.4**
NB	75.7	45.5	60.6
KNN	67.1	66.2	66.7
Expert classifiers for Oxidoreductase			
Classifier	Oxidoreductase	Non-Oxidoreductase	AverageAccuracy
SVM	72.5	64.6	**68.6**
NB	71.9	46.2	59
KNN	80.0	50.3	65.2
Expert classifiers for Transferase			
Classifier	Transferase	Non-Transferase	AverageAccuracy
SVM	87.0	65.8	**76.4**
NB	59.0	72.2	65.6
KNN	73.6	68.6	71.1

Table 5.4: Building experts for 2010 whole protein dataset

For example, svm is performing best for oxidoreductase with an accuracy of 68.6%

5.3 Combination of expert classifiers

5.3.1 Expert classifier for each enzyme class

Now, out of all the classifiers built for each class against all other classes, select the classifier that performs best in identifying that class against the other classes. These will be the expert classifiers for the corresponding classes.

Class	Corresponding Expert			
	Binding site 2008	Binding site 2010	Whole protein 2008	Whole protein 2010
Hydrolase	SVM	SVM	SVM	SVM
Isomerase	KNN	SVM	KNN	KNN
Ligase	SVM	SVM	SVM	SVM
Lyase	SVM	SVM	SVM	SVM
Oxidoreductase	SVM	SVM	SVM	SVM
Transferase	SVM	SVM	SVM	SVM

Table 5.5: Final experts for each dataset

From the Table 5.5, it is now known how each class is associated with an expert classifier of its own. So , when a new instance is given, it will be given to all the 6 expert classfiers , say, Expert_Hydrolase_SVM, Expert_Isomerase_KNN, Expert_Ligase_SVM, Expert_Lyase_SVM, Expert_Oxidoreductase_SVM and Expert_Transferase_SVM.

5.3.2 Results of combination of expert classifiers

The decision vector now obtained from all these experts will tell if an instance belongs to corresponding class or not. For example, a decision vector (1,0,0,0,1,0) means that Expert_Hydrolase_SVM tells the instance belongs to hydrolase and Expert_Oxidoreductase_SVM tells the instance belongs to oxidoreductase. Now we need to resolve among these two classes.

- If the decision vector has a single 1, then the corresponding class is assigned as the

target class label.

- If the decision vector has more than one 1s, then combination technique by test set accuracies is applied.

Table 5.6 shows the results of combination of expert classifiers on binding site and whole protein datasets. It should be observed that only using expert classifiers is not achieving good classification performance. The classification rate is only 23.3% and not going beyond 30% due to the unavailability of the dataset. This inadequacy is not allowing the classifier to do creditable classification.

Class	Datasaet			
	Binding site 2008	Binding site 2010	Whole protein 2008	Whole protein 2010
Hydrolase	68.3	54.0	31.8	70.3
Isomerase	0	0	0	5.2
Ligase	0	7.6	0	7.1
Lyase	11.7	0	9.0	20.6
Oxidoreductase	53.2	62.8	35.7	64.2
Transferase	6.1	73.2	82.3	66.3
Average	23.2	32.9	26.4	39.0

Table 5.6: Combination of expert classifiers

5.4 Refined approach : expert classifiers followed by n_{C_2} classifiers

One refinement to the combination of expert classifiers can be done by adding a second step.

- Give the new instance to expert classifiers
- If the decision vector of expert classifiers has a single 1, then the corresponding class is returned as the target class label.

- Otherwise, build two-way classifiers between all the classes that got positive predictions by experts. So if there are n 1s in the decision vector, n_{C_2} two-way classifiers are built (NOTE: if the decision vector has all 0s or all 1s, then two-way classifiers are built between all the classes) and the instance is given to these n_{C_2} classifiers and the votes are collected. Now, the class having the maximum votes is assigned as the target class label.

It can be observed from Table 5.7 that adding n_{C_2} classifiers in the second stage improves the classification rate significantly. With a 50% increase in the dataset, the performance accuracy in fact doubled from 28.7% to 50.8% on the binding site dataset.

Results of expert classifiers followed by n_{C_2} classifiers:

Class	Datasaet			
	Binding site 2008	Binding site 2010	Whole protein 2008	Whole protein 2010
Hydrolase	23.3	53.2	34.0	54.2
Isomerase	18.1	55.0	37.5	73.6
Ligase	50.0	53.8	66.6	71.4
Lyase	11.7	28.5	54.5	48.2
Oxidoreductase	38.7	62.8	33.3	51.4
Transferase	30.7	51.7	33.3	43.6
Average	28.7	50.8	43.2	57.1

Table 5.7: Refined approach results

5.5 Using 6_{C_2} all-vs-all classifiers

In the above method, Expert classifiers are followed by n_{C_2} classifiers. This can be generalized and the expert classifiers step can be removed completely. And all-versus-all method can be used. Here,simply two-way classifiers are built between all the classes. A new unseen

instance is given to all these $\frac{K(K-1)}{2}$ classes, where K is the total no. of classes. And the votes from these classifiers are obtained and based on the majority voting rule, the winning class label is obtained.

Using all-against-all classifiers:

Class	Datasaet			
	Binding site 2008	Binding site 2010	Whole protein 2008	Whole protein 2010
Hydrolase	1.6	46.7	43.1	57.6
Isomerase	27.2	60.0	50.0	68.4
Ligase	33.3	76.9	100	71.4
Lyase	11.7	46.4	90.9	62.0
Oxidoreductase	27.4	61.4	35.7	55.7
Transferase	30.7	48.2	27.4	53.7
Average	22.0	**56.6**	57.8	**61.5**

Table 5.8: Results of using 6_{C_2} one-vs-one classifiers

Clearly from the Table 5.8, just *all-vs-all* method is classifying best with 56.6% accuracy on 2010 binding site dataset. One may wonder why so much experimentation has to be done before arriving at such a conclusion. It should be observed that on a smaller dataset of 2008, the performance of all-vs-all approach is 22% as compared to 29% achieved by expert classifiers and 30% achieved by combination technique by probability. So it is important to study all the multiclass classification methods available.

Chapter 6

Conclusion and future work

In literature, enzyme classification is being done using only protein sequence data. This thesis addresses the issue of classifying the protein as an enzyme based on features from binding sites alone i.e., relating structure(binding site) to a protein function(enzyme), which is viewed as a challenging multiclass problem.

Previously, Bray et al [2] reported 33.3% accuracy in enzyme classification. We investigated several techniques of multiclass classification to address this twin problem of multiclass classification along with unbalanced dataset. Combination of classifiers by probability technique is making only an incremental improvement, not a significant improvement on unbalanced binding site dataset with SVM, naive bayes and KNN achieving 29.5% as compared to SVM alone with 28%. This data is further balanced and this balancing the data improved the accuray to 30.6% which is comparable to that of Bray et al[2]. We did experimentation on substitution groups whose results are not improving.
Sequence features were then calculated for the binding site dataset and combination technique as well as expert classification technique is used. The refined approach of expert classifiers [3] seems to be working on the dataset with sequence features with an accuracy of 30.8%. Whereas considering 6_{C_2} classifiers alone is achieving an accuracy of 25.9%, which is less compared to 30.8%. Also, the method of experts based on different feature sets is not working as the number of experts is too many making it difficult to reach a consensus. Since sequence features are not helping improve the classification accuracy, only amino acid composition is used. When only amino acid composition is used, the classification rate is not going beyond 23% by applying expert classifiers. When refined approach of

expert classifiers is used, it is observed that a 50% increase in the dataset(2010) doubled the classification performance from 28.7% to 50.8%. Finally, using all-vs-all method on 2010 binding set dataset achieved the highest accuracy of 56.6% whereas on the smaller 2008 dataset, performance of all-vs-all is low when compared to other methods.

In conclusion, extensive state-of-art experimentation has been done on this challenging dataset containing binding site features of proteins and we achieved a classification accuracy of 56.6% for enzyme classification. And it is observed that classification rate can be improved with more available dataset. It is not clear whether the experiments validate that binding site features have strong link to function(EC class).

Clearly the inadequacy of the dataset is restricting the classifier performance. So it makes it necessary to apply boosting techniques like AdaBoost and SMOTE analysis to enhance the dataset. And all the methods listed in this thesis has to be applied on that enhanced dataset for multiclass enzyme classification.

Bibliography

[1] Mohamed Aly. Survey on multiclass classification methods. *http://www.vision.caltech.edu/malaa/research/aly05multiclass.pdf*, chap 1.2:23, Nov 2005.

[2] Bray T, Doig AJ, Warwicker J. Sequence and structural features of enzymes and their active sites by ec class. *Mol Biol J.*, 386(5):1423–1436, 2009.

[3] Ding C, Dubchak I. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics.*, 17:349–358, 2001.

[4] IICT. Protein ligand interaction database, *http://203.199.182.73/gnsmmg/databases/plid/*.

[5] R. The r project for statistical computing, *http://www.r-project.org/*, April 2010.

[6] R Ranawana and V.Palade. Multiclassifier Systems: Reiew and roadmap for developers. *Int.J.Hybrid Intell.Syst3.*, 1:35, 2006.

[7] V Gunes,M Menard S.Petitrenaud. Multiple classifier systems: Tools and methods. *Handbook of pattern recognition and computer vision,4th Edition.*, 42:1273–1280, 2002.