A Mixed-Integer Programming Approach to Multi-Class Data Classification Problem

by

Fadime Yüksektepe Üney

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering

Koc University

September 2005

Koc University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Fadime Yüksektepe Üney

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

Metin Türkay, Ph. D. (Advisor)

____

Serpil Sayın, Ph. D.

____

Deniz Yüret, Ph. D.

____

Date: 09.05.2005

ABSTRACT

Data classification is an important data mining problem that aims to determine the membership of different data points to a number of different sets. Traditional approaches that are based on partitioning the data sets into two groups perform poorly for multi-class data classification problems. A new data classification method based on mixed-integer programming is presented in this thesis. The proposed approach is based on the use of hyper-boxes for defining boundaries of the classes that include all or some of the points in that set. A mixed-integer programming model is developed for representing existence of hyper-boxes and their boundaries. In addition, the relationships among the discrete decisions in the model are represented using propositional logic and then converted to their equivalent integer constraints using Boolean algebra. The proposed approach for multi-class data classification is illustrated on an example problem. The efficiency of the proposed method is tested on two separate data sets; the well-known IRIS data set and the protein folding type data set. The computational results on the illustrative example and the benchmark problems show that the simplicity and accuracy of the proposed method provides scientific insight into the multi-class data classification problems.

ACKNOWLEDGEMENTS

First, I would like to thank my advisor Ass. Prof. Metin Türkay for his great supervision and continuous support. He shared his knowledge and experience with me, paid attention to all the steps of my thesis and gave me the guidance that I required. I thank also Ass. Prof. Serpil Sayın and Ass. Prof. Deniz Yüret for participation in my thesis committee and for their supportive comments on my thesis.

I am also grateful to all my friends, Burcu, Canan, Sefa Nur, Alper, Aysun, Ahu, Tuğrul and Ertunç for being wonderful classmates, officemates and friends, Zeynep for her close friendship, and especially thank Semra not only for her valuable friendship but also for her helpful advice and being my confidant.

Finally, I thank my husband for always believing in me and for his support at every step of my life. I can not do anything without his patience, encouragement and love. Moreover, would like to thank my parents for their invaluable support. I am very lucky to be part of such a wonderful family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

i	training samples (i=*Sample1, Sample2, …, SampleI*)
j	test samples (j=*Sample1, Sample2, …, SampleI*)
k	class types (k=*Class1, Class2, …, ClassK*)
l	hyper-boxes that encloses a number of data points belonging to a class ($l=1,..,L$)
m	attributes ($m=1,..,M$)
n	bounds ($n=lo, up$)
C	small scalar
Q	large parameter
M	total number of attributes
L	total number of hyper-boxes
w	a non-zero vector
P	total number of attributes
ε	arbitrarily small positive number
a_{im}	value of the attribute m for the sample i
D_{ik}	class k that the data point i belong to
YB_l	Boolean variable to indicate whether the box l is used or not
YPB_{il}	Boolean variable to indicate whether the data point i is in box l or not
YBC_{lk}	Boolean variable to indicate whether box l represent class k or not
YPC_{ik}	Boolean variable to indicate whether the data point i is assigned to class k or not
$YPBN_{ilmn}$	Boolean variable to indicate whether the data point i is within the bound n with respect to attribute m of box l or not

$YPBM_{ilm}$	Boolean variable to indicate whether the data point i is within the bounds of attribute m of box l or not
$YP1_{ik}$	Boolean variable to indicate the type 1 misclassification of data points
$YP2_{ik}$	Boolean variable to indicate the type 2 misclassification of data points
X_{lmn}	the continuous variable that models bounds n for box l on attribute m
$XD_{l,k,m,n}$	the continuous variable that models bounds n for box l of class k on attribute m
$\vec{A}_j$	a vector for data point j composed of a_{jm} values
$\vec{P0}_{lmn}$	a vector for a extreme point of hyper box l for attribute m of bound n
$\vec{P1}_{lmn}$	another vector for a extreme point of hyper box l for attribute m of bound n
$\vec{W}_{jlmn}$	the difference vector between $\vec{A}_j$ and $\vec{P0}_{lmn}$
$\vec{V}_{jlmn}$	the difference vector between $\vec{A}_j$ and $\vec{P1}_{lmn}$
$C1_{jlmn}$	the dot product of the vectors $\vec{W}_{jlmn}$ and $\vec{V}_{jlmn}$
$C2_{jlmn}$	the dot product of $\vec{V}_{jlmn}$ by itself
b_{jlmn}	the ratio of $C1_{jlmn}$ to $C2_{jlmn}$
$Pb_{j}lmn$	the point where $\vec{A}_j$ is perpendicular to the edge between two extreme points

Chapter 1

INTRODUCTION

Data plays an important role in every decision process. Due to recent advances in hardware and software technologies, large quantities of data can be acquired, processed and stored. However, the ability to acquire, process and store the data is not sufficient in modern decision processes. The objective of data mining is to use discovered patterns to explain current behavior or to predict future outcomes. There are a large number of data mining methods and their implementations available. Data Classification is an important data mining problem that aims to determine the membership of different data points to a number of different sets [1].

1.1 Data Classification

Classification is a supervised learning strategy that analyzes the organization and categorization of data in distinct classes [2]. Generally, a training set, where all objects are already associated with known class labels, is used by classification approaches. The data classification algorithm learns from this training set by using input attributes and builds a model to classify new objects, in other words predicts output attribute values. Output attribute of the developed model is categorical. For instance, a bank could try to understand the behavior of its customers via analyzing their credit, and customers can be assigned three possible labels; “safe”, “risky”, and “very risky”. The generated model could be used to either accept or reject credit request in the future [1].

Classification, also known as supervised data mining technique, is one of the important strategies in knowledge discovery using databases. Classification has some significant differences from a related data mining technique, clustering. The class labels and the number of classes are not known in clustering. On the other hand, the class labels and the number of classes are known a priori for classification. In addition, there is not any output attribute in clustering; thus clustering algorithms try to group instances into two or more classes by using some measure of cluster quality [3]. Unlike clustering, prediction has an output attribute. However, the purpose of prediction is to determine future outcome rather than current behavior. In classification, output attribute is categorical, whereas the output attribute of predictive model can be either categorical or numerical. Classification emphasizes on building models that able to assign new instances to one of a set of well-defined classes [2].

Typical classification algorithms have three basic steps; model construction, model evaluation, and model use [1]. Each instance in training set is assumed to belong to a predefined class and using the input attributes the model is obtained in the model construction step. The model can be represented in different forms such as mathematical formulae, rule, etc. The next step, model evaluation, is the estimation of accuracy of the model based on a test set: known labels of each of the test samples are compared with the results of the model. The percentage of test set samples that are correctly classified by the model gives the accuracy value obtained. Selecting the instances of the test set is very critical: the test set must be independent of training set in order to get reliable results. Finally, the model is used to classify the unseen samples by assigning labels for them.

There are many application examples for classification in finance [2, 3], business [3], health care [2], sports [2], engineering [2, 4] and science [4]. In finance, especially in risk management, data classification is used to determine insurance rates, manage investment portfolios, and differentiate between individuals who have good or poor credit risks [3]. Furthermore, financial institutions use data classification to detect which customers are using which products so they can offer the right mix of products and services to better meet customer needs. Another application used by financial institutions is fraud detection in credit card and large cash transactions [2].

Customer Relationship Management (CRM) is a well-known application of data classification in business that involves the management of interactions with customers [3]. For this purpose, information related to each customer is collected and this data are used to increase the efficiency of interaction with the customers in all stages. In CRM, classification is generally used to assign a score to a particular customer or prospect indicating the likelihood that the individual will behave in such a way that revenues and customer satisfaction levels are improved. For example, the inclination to respond to a particular offer or to switch to a product form a competitor could be measured by a score. Moreover, characterization of customer segmentation into groups with similar behavior, such as buying a particular product, can be identified by classification. Consequently, data classification models can add tremendous value to organizations both in finance and business [2, 3].

In health care, medical diagnosis and the effectiveness of a treatment can be analyzed by the help of classification [2]. For instance, information about the patients who have had a heart attack or who have had not yet is collected. The risk of a person having a heart attack can be predicted using data classification methods. By using these risk values, some precautions are taken and some medical treatments are applied to the person with a high risk [2]. From the other perspective, the outcome of a back surgery for a patient can be guessed by using the information about the patients who have had similar treatment in the past [4]. In the case of sports, the data related to past matches between the teams are collected. While playing the chance games, gamblers use these past data and guess the result of the incoming match and the winner. This application has been also used for horse racing and lottery [2].

Data classification has a wide range of security related applications as well: fingerprint and facial recognition are the most studied topics. Another widely used application of data classification is in the area of bioinformatics; classification methods are being used in order to get valuable information on the characteristics of genes and proteins. Many classification methods are used in micro array analysis to predict sample phenotypes based on gene expression patterns [4]. Another problem in bioinformatics that attracted a lot of attention in the literature is the prediction secondary structure of a protein from its amino acid sequence [4]. Moreover, protein folding type prediction is also studied with different classification methods [4]. In conclusion, data classification is an important problem that has applications in a diverse set of areas ranging from business to bioinformatics.

There exists a broad range of methods for data classification problem including Decision Tree Induction [2], Association-Based Classification [2], Neural Networks (NN) [4], Bayesian Classifier [2], K-nearest Neighbor [2], Genetic algorithms [4], Rough Set Theory [2], Fuzzy Sets [4], Statistical Regression [2] and Support Vector Machines (SVM) [4]. Decision trees are the popular structures that look like flow-chart trees and are constructed using only those attributes best able to differentiate the concepts to be learned. A decision tree is built by initially selecting a subset of instances from a training set. The algorithm uses this subset and constructs the decision tree. The remaining training set instances are used to test the accuracy of the constructed tree. Conversely, association rules have the ability to find relationships in large databases without having the restriction of choosing a single output attribute [2]. A neural network is a data structure that tries to simulate the behavior of neurons in a biological brain. Network is composed of layers of units interconnected. From one unit to other, messages are passed along the connections.

Through this transfer, message can change based on the weight of the connection and the value in the node. Although the prediction accuracy is generally high, neural networks need long training time and understanding the learned function is very difficult [4]. Another classification method is a statistical classifier named as Bayesian Classifier that calculates explicit probabilities for hypothesis based on Bayes theorem and uses probabilistic learning. It performs well with large data sets and exhibits high accuracy [2]. K-nearest classifier has a different approach which classifies a new instance with the most common class of its k-nearest neighbors. Computational time can be improved by comparing instances to be classified with a subset of typical instances taken from each class represented in the data. However, this approach produces a high classification error rate when dealing with instances having irrelevant attributes [2]. Genetic Algorithms are used in data classification problems that are difficult to solve using conventional methods. It is based on Darwinian principle of natural selection; crossover and mutation are the most widely used genetic operators. In a basic genetic learning algorithm, a population P of n elements is initialized which often referred to as chromosomes. A fitness function is used to evaluate each element of current solution. If an element passes fitness criteria, it remains in P. By using genetic operators new elements are created and added to the population. This procedure is carried on until a specified termination condition is satisfied. Rough Set Theory (RST) can be approached as an extension of the Classical Set Theory [2]. Rough sets are considered as the sets with fuzzy boundaries, in other words the sets that cannot be precisely characterized using the available set of attributes. In data classification, it is inconvenient to describe the similarity among data with the indiscerniblity relation because two data x and z can not be guaranteed in the same class even though a couple of data x and y are contained in the same class and another couple of data y and z are also contained in the same class. In other words, the transitivity property is not always useful in the problem of data classification. This non-transitivity property is more salient for the data within the boundary region. For this reason, a tolerant relation appropriate for the data classification problem is studied by some researchers. In contrast, Fuzzy Sets are based on Fuzzy Logic [4]. Fuzzy logic is an extension of Boolean logic (YES or NO) dealing with the concept of partial truth. Whereas, classical logic holds that everything can be expressed in binary terms (0 or 1, yes or no), fuzzy logic replaces Boolean truth values with degrees of truth. It was first introduced in 1965 by Prof. Lotfi Zadeh at the University of California, Berkeley. Another method, Statistical Regression is a supervised technique that generalizes a set of numeric data by creating a mathematical equation relating one or more input attributes to a single output attribute. In general, linear regression is appropriate when the relationship between the input and output attributes is nearly linear [2]. SVMs are based on statistical learning theory that operate by finding a hyper surface that will split the groups so that the distance between the hyper surface and the nearest of the points in the groups has the largest value. SVMs generally give high accuracy values and are quite efficient [4].

While evaluating the data classification methods, some important properties of the model have to be considered in detail. Firstly, methods are usually evaluated on the test data. Prediction accuracy, ability of the model to correctly predict the class label, is a very considerable point for evaluation. Most of the comparisons between the models are done by looking directly to these prediction accuracy values. On the other hand, time to construct the model and time to use it also has a big role in real life applications. For a preferable data classification model, computational time must be reasonable. Thirdly, for an ideal data classifier, it should have a few parameters to tune in the system as possible. In Neural Networks, the weights between the nodes have to be adjusted. Since all of the existing weights need to be optimized, it is not easy to incorporate the domain knowledge and they possess a long training time. Moreover, it is difficult to understand the learned function. Similarly, SVM method has the biggest limitation of choosing the kernel function. Once the kernel is fixed, SVM classifiers have only one user-chosen parameter, error penalty.

However, kernel is a very important decision criterion. Another important characteristic of an ideal data classifier is the ability to form a decision boundary that minimizes the amount of misclassification for all of the overlapping classes in the training set.

Some of the methods mentioned above can only be used for the two class cases, such as yes (class1) or no (class 2). However, the number of classes to be classified is generally more than two in real life problems. Existing methods can be somehow modified or developed for multi-class case. In that situation, the accuracy values of the models decrease [4]. For instance, SVMs are originally a model for two class problems and are more effective. For multi-class case, combinations of SVMs should be used. Since SVMs use some approximation algorithms in order to reduce the computational time, increasing number of these approximation algorithms causes the degradation of classification performance. Thus, the performance does not improve as much as in binary case. Therefore, there is a need for new approaches that are able to address multi-class problems effectively. In this study, a novel mixed-integer programming approach for multi-class data classification problem has been developed. The proposed approach is based on the use of hyper-boxes for defining boundaries of the classes that include all or some of the points in that set. The computational results on the studied datasets show that the suggested method is accurate and efficient on multi-class data classification problems.

1.2 Contributions

This thesis presents a new mathematical programming method for multi-class data classification problems. A mixed-integer programming model is developed for representing existence of hyper-boxes which define the boundaries of the classes for the training set. The performance of the model is tested by the testing part of the proposed method. Main contributions of this thesis can be summarized as follows:

The suggested model can be used for both binary and multi-class cases without any modifications or additions. Hence, the performance of the model is always high and does not decrease in any class related changes.

On the other hand, the proposed model has only one parameter to be initialized and no need to adjust this parameter during the training of the model. Furthermore, the model can operate satisfactorily without a priori knowledge about the underlying distribution of the data. These properties make the model simple, easily understandable and attractive.

One of the most important contributions is that the proposed data classification method based on mixed-integer programming allows the use of hyper-boxes for defining boundaries of the classes that enclose all or some of the points in that set. Consequently, using hyper-boxes for multi-class data classification problems is accurate because of the well-construction of boundaries of each class. This lack of misclassifications in the training set indirectly effects and improves the prediction accuracy of the model. This is the reason behind the high classification accuracy values obtained by the developed model.

If we take a part from the computational time perspective, the construction of the suggested model is not time consuming. Preparation of the necessary data for the model could only take 3-5 minutes. However, time to run the model and get the results need some time related to the dimensionality of the problem. As observed from the worked data sets, this time is reasonable and less than the other methods used for these data sets.

The proposed model in this thesis has good accuracy values on the studied data sets. Hence, the developed multi-class data classification model is as accurate as the other models used including NN, SVM, Fuzzy Sets, Tolerant Rough Set, Singular Value Decomposition, etc.

In summary, by the development of this new approach, solutions to multi-class data classification problems can be obtained and improvement in the prediction accuracy is addressed. In addition to this, the simplicity and the understandability of the proposed model are preferable. In overall, the suggested model is attractive with respect to above characteristics.

1.3 Outline

This thesis contains five chapters. Chapter 2 provides a literature review on data classification summarizing distinct methods reported including the mathematical programming based approaches with the results on the studied datasets. The developed multi-class data classification approach is presented in Chapter 3. The mixed-integer programming formulation for the training part of the problem and the testing algorithm are discussed in detail. The method is also illustrated on a small illustrative example in Chapter 3. The applications of the proposed approach on two separate benchmark data sets are illustrated in Chapter 4: the well-known IRIS data set and the protein folding type data set. The efficiency of the proposed method on these benchmark problems is tested and results are examined in detail. The thesis is concluded with short summary, conclusions, and directions on future research work.

Chapter 2

LITERATURE REVIEW

Classification problems have been intensively studied by a diverse group of researchers including statisticians, engineers, biologists, computer scientists. There are variety of methods for solving classification problems such as neural networks (NN), fuzzy logic, support vector machines (SVM), K-nearest neighbor approach, tolerant rough sets, and linear programming [5]. In this chapter, a literature review on data classification methods and an important problem, prediction of folding type of proteins, is provided.

2.1 Literature Review on Data Classification Methods

An overall view of classification methods is published by Weiss & Kulikowski [6]. In this book, available classification and prediction methods from statistics, neural networks, machine learning and expert systems are reviewed. Hand [7] investigates the statistical approach of data classification and pattern detection in the fields of medicine, psychology and finance. The estimation of error rates in discriminant analysis is explored by Lachenbruch & Mickey [8]. In this study, leave-one-out cross-validations tests are proposed for error estimation. N (number of data points) separate times, the classification function is trained on all the data except for one point and a prediction is made for that point in leave-one-out cross-validation tests. Average error is computed and used to evaluate the model. The evaluation given by this cross-validation test error is good, but computing the result of leave-one-out tests takes very long time. Kendall et al. [9] give a comprehensive exposition about the statistical approach of data classification and advance theory of statistics. The book by Hertz et al. [10] is one of the most detailed and reliable information guides for neural network approach in pattern recognition. The book gives an introduction to neural computation and explains the theory of the neural network approach. On the other hand, Devijver & Kittler [11] concentrate on the K-nearest neighbor approach for data classification problems from the perspective of statistical approach. McLachlan studied on a thorough treatment of statistical procedures in discriminant analysis and pattern recognition [12].

One of the first papers published on data classification introduces fuzzy adaptive resonance theory (ART) which is a fast and reliable analog pattern clustering system. In this study, Carpenter and Grossberg combine the fuzzy logic with the idea of ART and try to develop an efficient classifier [13]. A general neural-network model for fuzzy logic control and decision systems including the data classification problem is discussed in [14]. Furthermore, Simpson [15] developed a fuzzy min-max classification neural network in which pattern classes are utilized as fuzzy sets. In this study, learning in the neural network was performed by properly placing and adjusting hyper boxes in the pattern space. Simpson defines a fuzzy set hyper-box as an n-dimensional box defined by a min and a max point with a corresponding membership function. The min-max membership function combination defines a fuzzy set, hyper-box fuzzy sets are aggregated to form a single fuzzy set class, and the resulting structure fits naturally into a neural network framework. Therefore, this classification system is referred as fuzzy min-max classification neural network. Since it uses only a min and a max point in the n-dimensional space and combines fuzzy sets with the neural network idea, this model has a different approach as compared to the proposed model in this thesis. All of the above classifications methods combine fuzzy logic with neural network. Thus, they are destined to have the same drawbacks as neural networks. The possibility of having the nonconvergent solution due to the wrong choice of initial weights and the possibility of having non-optimal solutions due to the local minima problem are important handicaps of neural network-based methods.

Rough set theory introduced by Pawlak [16] is a mathematical tool to deal with vagueness and uncertainty in machine learning and pattern recognition. Two applications of logic for classification using rough set approach are presented in [17]. The multi-model logics is employed for automatic feature selection while a rough-set-based inductive reasoning is used for discovering optimal feature set with respect to the quality of classification as well as for improving the performance of decision algorithms. Another approach in data classification is to use rough sets by tolerating the relationships among the objects for pattern classification [18]. A data classification method based on the tolerant rough set that combines the use of logic and the tolerance relation among the objects is presented in [19]. The performance of this approach is tested on the IRIS data [20] classification problem, which is also used in this thesis. Furthermore, Castro et al. [21] presented a method to learn maximal structure rules in fuzzy logic to deal with the IRIS data classification problem. Chen et al. [22], Hong et al. [23], Lin et al. [24] and Wu et al. [25] presented different methods to generate fuzzy rules from training instances based on genetic algorithms to deal with the IRIS problem. Most recently, Chen et al. [26] developed a new model based on distributions of training instances. They found two useful attributes of the IRIS data from the training instances that are more suitable to deal with the classification problem. Their proposed method achieves a higher average classification accuracy rate than existing methods.

In recent years, SVM has been considered as one of the most efficient methods for two-class classification problems [27]. SVM is a new classification technique developed by Vapnik and his group [28]. SVM is able to generate a separating hyper surface in order to maximize the margin and produce good generalization ability. However, the SVM has two important drawbacks. First, a combination of SVMs has to be used in order to solve the multi-class classification problems. Second, some approximation algorithms are used in order to reduce the computational time for SVMs while learning the large scale of data. On the other hand, this computational improvement could cause less efficient performance values. To overcome above problems, many variants of SVM are suggested including the use of SVM ensemble with bagging or boosting rather than using a single SVM [29]. Hsu et al. [30] compared the performance values of “all-together” and binary classification based methods such as “one-against-all”, “one-against-one” and directed acyclic graph (DAG) SVM. One-against-all method is the earliest used implementation for SVM multi-class classification. It constructs *k* SVM models where *k* is the number of classes. The *i*th SVM is trained with all of the examples in the *i*th class with positive labels, and all other examples with negative labels. One by one each class is separated from the others. On the other hand, one-against-one method constructs *k(k-1)/2* classifiers where each one is trained on data from two classes. In the testing part, if sign of the model says *x* is in the *i*th class, then the vote for for the *i*th class is added by one. Otherwise, the *j*th is increased by one. After all, *x* is predicted to be in the class with the largest vote. Direct acyclic graph SVM method’s training phase is same as the one-against-one method by solving *k(k-1)/2* binary SVMs. However, in the testing phase, is uses a rooted binary directed acyclic graph which has *k(k-1)/2* internal nodes and *k* leaves. Each node is a binary SVM of *i*th and *j*th classes. Given a test sample *x*, starting at root node, the binary decision function is evaluated. Then, it moves either left or right depending on the output value. Therefore, it goes through a path before reaching a leaf node which indicates the predicted class. Hsu et al. [30] came up with the result that “one-against-one” and DAG binary classification methods are more suitable for practical use than the other methods. Nevertheless, for solving multi-class SVM in one step, a much larger optimization problem is required so experiments are limited to small data sets.

In summary, a large number of data classification methods have been developed up to now; however each of them has some drawbacks which make them unattractive. Thus, researchers have been studying to develop more accurate and more efficient methods or to improve the existing methods.

2.2 Literature Review on Mathematical Programming Based Methods

There have been some attempts to solve classification problems using mathematical programming (MP). A survey of MP discriminant methods is given by Joachimsthaler and Stam [31]. The mathematical programming approach to linear discriminant analysis was first introduced in early 1980's. Since then, numerous mathematical programming models have appeared in literature. An extension and a complement to this, Erenguc and Koehler made a comprehensive review [32]. In their research, they formulate a typical mathematical programming (MP) approach as follows:

$$\text{Minimize} \quad f(w,c) \qquad (2.1)$$

$$\text{Subject to:} \quad X_1\, w \leq c1 \qquad (2.2)$$

$$X_2\, w \geq (c + \varepsilon)1 \qquad (2.3)$$

$$w \neq 0 \qquad (2.4)$$

By this general formulation MP approach tries to determine a scalar c and a non-zero vector $w \in R^p$ such that the hyper plane $w'x = c$ partitions the p-dimensional (p is the number of attributes) Euclidean space R^p into a closed half-space $w'x \leq c$ and an open half-space $w'x > c$. In the formulation, ε represents an arbitrarily small positive number. Interior and exterior deviation terms for each group are defined for MP approaches. An interior deviation is the deviation from the hyper plane of a point properly classified. An exterior deviation is the deviation from the hyper plane of a point improperly classified. Many distinct MP methods with different objective functions are developed in literature. Minimizing the maximum exterior deviation, minimizing the weighted sum of exterior deviations, minimizing a measure of exterior deviations while maximizing a measure of interior deviations, minimizing the number of misclassifications, and minimizing a generalized distance measure are some of the objective functions used by suggested MP based methods. Most of these methods modeled data classification as linear programming (LP) problems which optimize a distance function. Contrary to LP problems, mixed-integer linear programming (MILP) problems with minimizing the misclassifications on the design data set as an objective function are studied in this thesis. MP methods have certain advantages over the parametric ones. For instance, they are free from parametric assumptions and weights to be adjusted. Moreover, varied objectives and more complex problem formulations can easily be accommodated by using MP methods. On the other hand, obtaining a solution without any discriminating power, unbounded solutions and excessive computational effort requirement are some of the problems in MP based methods. Koehler [33] surveys these problems that may appear in MP formulations. There have been several attempts to formulate data classification problems as MILP problems [34-37]. Since MILP methods suffer from computational difficulties, the efforts are mainly focused on efficient solutions for two-group supervised classification problems. Although there exist ways to solve a multi-class data classification problem by means of solving several two-group problems, such approaches have also some drawbacks as mentioned in [38]. Hence, MP based heuristics are developed in order to tackle the multi class data classification problem directly. A heuristics extension of MP approach in order to improve the performance of multi-class supervised classification is proposed in [5]. In this research, a simulated data set is used to evaluate the performance of the developed heuristic. Computational results show that time consuming of the classification method decreases and solutions often close to global optimum are obtained.

2.3 Literature Review on Protein Folding Type Prediction

Proteins are the molecules of life that play a key role in realizing the functions of any biological organism. Discovery of the functions of proteins will enable us to understand the principles of life and working mechanisms of any organism. In the case of humans, this discovery will lead to the design of new drugs that will regulate the functions of proteins in order to improve the quality of life. Functions of proteins are highly correlated to their three dimensional structure. There exist some experimental methods to determine the protein structure including X-ray diffraction and nuclear magnetic resonance (NMR). These experimental methods require long experimental times and large amounts of resources. In order to overcome these shortcomings of experimental methods, researchers have developed a host of methods to predict the protein structures. Due to the importance of protein structure in understanding the biological and chemical activities in any biological system, protein structure determination and prediction has been a focal research subject in computational biology and bioinformatics.

Proteins are classified according to their secondary structure content, considering α-helices and β-sheets. Levitt et al. [39] were the first researchers who propose such a classification with four basic types. ”All-alpha” proteins consist almost entirely (at least 90%) of α -helices. ”All-beta” are the ones composed mostly of β-sheets (at least 90%) in their secondary structures. There are two intermediate classes which have mixed α-helices and β-sheets. ”Alpha/beta” proteins have approximately alternating, mainly parallel segments of α –helices and β-sheets. The last class,”alpha+beta” has mixture of all alpha and all beta regions, mostly in an anti-parallel fashion. It is postulated that overall folding type of a protein depends on its amino acid composition [40]. There have been several methods proposed to exploit this postulate for predicting folding type of a protein. Chou [41] developed a new prediction algorithm which incorporates coupling effect between different amino acid components. By the help of this component-coupled algorithm, prediction quality was significantly improved. Another important progress in this area was achieved by Bahar et al. [42]. In this study, they proposed a compact lattice model to clarify the success in predicting structural class from amino acid composition and achieved

81% accuracy value by the help of singular value decomposition method. In this method, each protein is represented by a 19-dimensional array of fluctuations in fractions of residues of different types. The *j*th element of this vector is the difference between the composition of the amino acid type *j* and the average fraction of amino acid *j* in the group of n structures. The distance of a protein from the four type of structural classes are calculated using 19-dimensional array of the protein by applying singular value decomposition method. The smallest of the four distances obtained for each protein determines the structural class of that protein. Although they use the same data set and mathematically identical method with Chou, their accuracy is somehow less. They explore this puzzling difference and came up with the result that the data files used in these studies are different. Chou used files that contained fewer residues (chains of amino acids) compared with intact Protein Data Bank (PDB) files. Eisenhaber et al. [43] found that component coupling effect between amino acid components did not improve the class prediction, using a different dataset constructed according to their definition. In order to clarify this paradox, Zhou [44], Chou et al. [45] and Cai [46] showed that component-coupled algorithm significantly improved the prediction accuracy. The reasons why Eisenhaber et al. come up with that result are misusing the component-coupled algorithm and using a conceptually incorrect rule to classify protein structural classes. On the other hand, Bu et al. [47] come up with a new idea, using amino acid index rather than composition in order to predict the structural classes. The overall predictive accuracy of the new proposed method for the jackknife test was 5-7% higher than the accuracy based only on the composition. However, many researchers continued studying on the first case, based on only the amino acid composition. Cai et al. [48] applied T. Kohonen's self-organization neural network on two data sets composed of 277 and 498 domains, respectively. They showed that this approach can be a powerful tool for protein structural class prediction. Furthermore, support vector machine (SVM) method was performed based on the same data sets by [49]. The SVM method applies for two class problems. Thus,”one-against other” method is used to transfer it into two class problems. Consequently, the prediction of folding types from amino acid composition alone is an important topic, which has been the object of many recent researches.

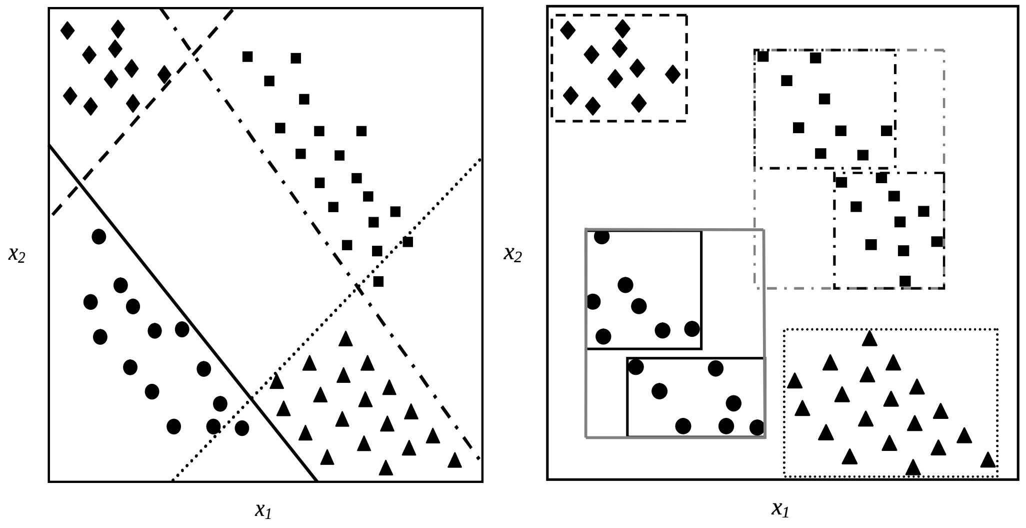
In conclusion, there exists restricted number of methods for multi class data classification problems in literature. This thesis addresses the need for efficient and reliable methods for multi class problems by introducing a new mixed-integer programming approach. Moreover, the important and widely used data sets, the well-known IRIS data set and the protein folding type data set, are studied to analyze the performance of the developed model. The results on these data sets show that the prediction accuracy of the developed model is better than the existing data classification models in literature. Furthermore, developed model gets rid of some drawbacks of the available multi class data classification models with only one adjustable parameter, rather short learning time, no need to know the underlying distribution of the data and well-construction of the class boundaries.

Chapter 3

PROBLEM FORMULATION FOR MULTI-CLASS DATA CLASSIFICATION

The objective in data classification is to assign data points that are described by several attributes into a predefined number of classes. Figure 3.1.a shows a schematic representation of the classification of multi-dimensional data using hyper-planes. Although the methods that are based on using hyper-planes to define the boundaries of classes can be efficient in classifying data into two sets, they are inaccurate and inefficient when data needs to be classified into more than two sets. On the other hand, the use of hyper-boxes for defining boundaries of the sets that include all or some of the points in that set as shown in Figure 3.1.b can be very accurate on multi-class problems.

It may be necessary to use more than one hyper-box in order to represent a single class as shown in Figure 3.1.b. When the classes that are indicated by square and circle data points are both represented by single hyper-box respectively, the boundaries of these hyper-boxes overlap requiring more than a single hyper-box. If a region of the attribute space is assigned to more than one class, it is possible that a new data point is classified into more than a single class. In order to eliminate this possibility, more than one hyper-box must be used to include all of the data points that belong to a class into the same class. A very important consideration in using hyper-boxes is the number of boxes used to define a class. If the total number of hyper-boxes is equal to the number of classes (i.e., exactly one hyper-box classifies all data points of the same class), then the data classification is very efficient. On the other hand; if there are as many hyper-boxes of a class as the number of data points in a class (i.e., each data point of a particular class is represented by a separate hyper-box), then the data classification is inefficient.



a) using hyper-planes b) using hyper-boxes

Figure 3.1. Schematic representation of classification of data.

The data classification problem based on this new idea is considered in two parts: training and testing. The objective of the training part is to determine the characteristics of the data points that belong to a certain class and differentiate them from the data points that belong to other classes. That is to say, boundaries of the classes are formed by the construction of hyper-boxes. After the distinguishing characteristics of the classes are determined, then the effectiveness of the classification must be tested. In order to do that, the predictive accuracy of the developed model is performed on a test data set.

3.1. Training Problem Formulation for Multi-Class Data Classification

Training part studies are performed on a training data set composed of a number of instances *i*. The data points are represented by the parameter a_{im} that denotes the value of attribute *m* for the instance *i*. The class *k* that the data point *i* belongs to are given by the set D_{ik}. Each existing hyper-box *l* will enclose a number of data points belonging to the class *k*.

Moreover, bounds *n* (*lower, upper)* of each hyper-box will be determined by solving the training problem.

Given these parameters and the sets, the following Boolean variables are sufficient to model the data classification problem with hyper-boxes as depicted in Figure 3.1.b. YB_l is a Boolean variable that indicates whether the box *l* is used or not. The position (inside or outside) of the data point *i* with regard to box *l* is represented by YPB_{il} . The assigned class *k* of box *l* and data point *i* is symbolized by YBC_{lk} and YPC_{ik} , respectively. If the data point *i* is within the bound *n* with respect to attribute *m* of box *l*, the Boolean variable $YPBN_{ilmn}$ is "yes", otherwise "no". Similarly, $YPBM_{ilm}$ represents the Boolean variable which indicates whether the data point *i* is within the bounds of attribute *m* of box *l* or not. Finally, $YP1_{ik}$ and $YP2_{ik}$ indicate the misclassification type 1 and type 2 of data points.

The relationships among these Boolean variables can be represented using the following propositional logic [50-52]:

$$\underline{\bigvee_{l}} YPB_{il} \quad \forall i \tag{3.1}$$

$$\underline{\bigvee_{k}} YPC_{ik} \quad \forall i \tag{3.2}$$

$$\underline{\bigvee_{l}} YPB_{il} \Leftrightarrow \underline{\bigvee_{k}} YPC_{ik} \quad \forall i \tag{3.3}$$

$$YB_l \Rightarrow \bigvee_{k} YBC_{lk} \quad \forall l \tag{3.4}$$

$$YBC_{lk} \Rightarrow \bigvee_{i} YPB_{il} \quad \forall l,k \tag{3.5}$$

$$YBC_{lk} \Rightarrow \bigvee_{i} YPC_{ik} \quad \forall l,k \tag{3.6}$$

$$\bigwedge_{n} YPBN_{ilmn} \Rightarrow YPBM_{ilm} \quad \forall i,l,m \tag{3.7}$$

$$\bigwedge_{m} YPBM_{ilm} \Rightarrow YPB_{il} \quad \forall i,l \tag{3.8}$$

$$YPC_{ik} \Rightarrow YP1_{ik} \qquad \forall i,k \notin D_{ik} \tag{3.9}$$

$$\neg YPC_{ik} \Rightarrow YP2_{ik} \qquad \forall i,k \in D_{ik} \tag{3.10}$$

The Eqs. (3.1) and (3.2) state that every data point must be assigned to a single hyper-box, *l*, and a single class, *k*, respectively. The equivalence between Eqs. (3.1) and (3.2) is given in Eq. (3.3); indicating that if there is a data point in the class *k*, then there must be a hyper-box *l* to represent the class *k* and vice versa. The existence of a hyper-box implies the assignment of that hyper-box to a class as shown in Eq. (3.4). If a class is represented by a hyper-box, there must be at least one data point within that hyper-box as in Eq. (3.5). In the same manner, if a hyper-box represents a class, there must be at least a data point within that class as given in Eq. (3.6). The Eq. (3.7) represents the condition of a data point being within the bounds of a box in attribute *m*. If a data point is within the bounds of all attributes of a box, then it must be in the box as shown in Eq. (3.8). When a data point is assigned to a class that it is not a member of, type 1 penalty applies as indicated in Eq. (3.9). When a data point is not assigned to a class that it is a member of, type 2 penalty applies as given in Eq. (3.10). In mathematical programming applications, there is one-to-one correspondence between a Boolean variable and a binary variable [50]. The value of True for a Boolean variable is equivalent to the value of 1 for the corresponding algebraic variable. The same argument applies between False and 0. In addition, it is possible to obtain exact algebraic equivalent of propositional logic expressions. The correspondence between propositional logic expressions and their equivalent algebraic representations are given in Table 3.1.

Table 3.1. The correspondence between propositional logic and algebraic inequalities.

Logical Relation	**Boolean Expression**	**Algebraic Constraint**
OR ($\vee$)	$Y_1 \vee Y_2 \vee \ldots \vee Y_r$	$y_1+y_2+\ldots+y_r{}^3 1$
AND ($\wedge$)	$Y_1 \wedge Y_2 \wedge \ldots \wedge Y_r$	$y_1{}^3 1$ $y_2{}^3 1$ $\ldots$ $y_r{}^3 1$
EXCLUSIVE OR ($\underline{\vee}$)	$Y_1 \underline{\vee} Y_2 \underline{\vee} \ldots \underline{\vee} Y_r$	$y_1+y_2+\ldots+y_r=1$
IMPLICATION ($Y_1 \Rightarrow Y_2$)	$\neg Y_1 \vee Y_2$	$1-y_1+y_2{}^3 1$
EQUIVALANCE ($Y_1 \Leftrightarrow Y_2$)	$(\neg Y_1 \vee Y_2) \wedge (Y_1 \vee \neg Y_2)$	$y_1-y_2=0$

Using the information in Table 3.1, Boolean equations that represent the construction of hyper-boxes are one by one converted to corresponding algebraic formats. The algebraic equivalents of the propositional logic expressions in Eqs. (3.1)-(3.10) are given as follows:

$$\sum_l ypb_{il} = 1 \quad \forall i \tag{3.11}$$

$$\sum_k ypc_{ik} = 1 \quad \forall i \tag{3.12}$$

$$\sum_l ypb_{il} = \sum_k ypc_{ik} \quad \forall i \tag{3.13}$$

$$\sum_k ybc_{lk} \le yb_l \quad \forall l \tag{3.14}$$

$$ybc_{lk} - \sum_i ypb_{il} \le 0 \quad \forall l,k \tag{3.15}$$

$$ybc_{lk} - \sum_i ypc_{ik} \le 0 \quad \forall l,k \tag{3.16}$$

$$\sum_n ypbn_{ilmn} - ypbm_{ilm} \le \mathrm{N}-1 \quad \forall i,l,m \tag{3.17}$$

$$\sum_m ypbm_{ilm} - ypb_{il} \le \mathrm{M} - 1 \quad \forall i,l \tag{3.18}$$

$$ypc_{ik} - yp1_{ik} \le 0 \quad \forall i,k \notin D_{ik} \tag{3.19}$$

$$ypc_{ik} + yp2_{ik} \ge 1 \quad \forall i,k \in D_{ik} \tag{3.20}$$

In order to define the boundaries of hyper-boxes, two continuous variables are defined. X_{lmn} is the one that models bounds n for box l on attribute m. Correspondingly, bounds n for box l of class k on attribute m are symbolized by the continuous variable $XD_{l,k,m,n}$.

The following mixed-integer programming problem models the training part of data classification method using hyper-boxes:

$$\min \quad z = \sum_i \sum_k \left(yp1_{ik} + yp2_{ik} \right) + c \sum_l yb_l \tag{3.21}$$

subject to

$$XD_{lkmn} \le a_{im} ypb_{il} \quad \forall i,k,l,m,n \,|\, n = lo \tag{3.22}$$

$$XD_{lkmn} \ge a_{im} ypb_{il} \quad \forall i,k,l,m,n \,|\, n = up \tag{3.23}$$

$$XD_{lkmn} \le Q\ ybc_{lk} \quad \forall k,l,m,n \tag{3.24}$$

$$\sum_k XD_{lkmn} = X_{lmn} \forall l,m,n \tag{3.25}$$

$$ypbn_{ilmn} \ge (1/Q)(X_{lmn} - a_{im}) \quad \forall i,l,m,n \,|\, n = up \tag{3.26}$$

$$ypbn_{ilmn} \ge (1/Q)(a_{im} - X_{lmn}) \quad \forall i,l,m,n \,|\, n = lo \tag{3.27}$$

Eqs. (3.11)-(3.20)

$$X_{lmn}, XD_{lkmn} \ge 0,\ yb_l, ybc_{lk}, ypb_{il}, ypc_{ik}, ypbn_{ilmn}, ypbm_{ilm}, yp1_{ik}, yp2_{ik} \in \{0,1\} \tag{3.28}$$

The objective function of the mixed-integer programming problem is to minimize the misclassifications in the data set with the minimum number of hyper-boxes. In order to eliminate unnecessary use of hyper-boxes, the unnecessary existence of a box is penalized with a small scalar *c* in the objective function. The lower and upper bounds of the boxes are given in Eqs. (3.22) and (3.23), respectively. The lower and upper bounds for the hyper-boxes are determined by the data points that are enclosed within the hyper-box. Eq. (3.24) enforces the bounds of boxes exist if and only if this box is assigned to a class. Eq. (3.25) is used to relate the two continuous variables that represent the bounds of the hyper-boxes. The position of a data point with respect to the bounds on attribute *m* for a hyper-box is given in Eqs. (3.26) and (3.27). The binary variable $ypbn_{ilmn}$ helps to identify whether the data point *i* is within the hyper-box *l*. Two constraints, one for the lower bound and one for the upper bound, are needed for this purpose (Eqs. (3.26) and (3.27)). Since these constraints establish a relation between continuous and binary variables, a large parameter, *Q*, is included in them. The model includes all of the algebraic constraints on binary variables that are constructed from propositional logic. Finally, last constraint gives non-negativity and integrality of decision variables. By using this MIP formulation, a training set can be studied and the bounds of the classes can be determined for a data classification problem.

3.2. Testing Problem for Multi-Class Data Classification

The testing problem for multi-class data classification using hyper-boxes is straight forward. If a new data point whose membership to a class is not known arrives, it is necessary to assign this data point to one of the classes. There are three possibilities for a new data point when determining its class:

i. the new data point is within the boundaries of a single hyper-box
ii. the new data point is within the boundaries of more than one hyper-box
iii. the new data point is not enclosed in any of the hyper-boxes determined in the training problem

When the first possibility is realized for the new data point, the classification is made by directly assigning this data to the class that was represented by the hyper-box enclosing the data point. Since eliminating the shared areas between the constructed hyper-boxes introduces new constraints into the training problem that makes it computationally very difficult to be solved, there exists a possibility for a new data point to be within the boundaries of more than one hyper-box. In that case, the data point is assigned to the classes of the hyper-boxes that enclose this specific data point. The proportion of the number of correct classes to the number of total assigned classes to that data point determines the effect of that data point to the accuracy of the model. For example, if a data point belonging to Class 1 is enclosed by two different hyper-boxes (Box1 and Box2) whose classes are Class 1 and Class 2, then this data point is assumed to be classified with 50% accuracy and the number 0.5 is added to the number of correctly classified test samples. In the case when the third possibility applies, the assignment of the new data point to a class requires some analysis. If the data point is within the lower and upper bounds of all but not one of the attributes (i.e., m') defining the box, then the shortest distance between the new point and the hyper-box is calculated using the minimum distance between hyper-planes defining the hyper-box and the new data point. The number of hyper-planes that must be evaluated for determining the minimum distance between the new data point and the hyper-box is given by ***2(M-1)*** where ***M*** is the total number of attributes. The minimum distance between the new data point ***j*** and the hyper-box is calculated using Eq. (3.29) considering the fact that the minimum distance is given by the normal of the hyper-plane.

$$\min_{l,m,n}\left\{\left|\left(a_{jm}-X_{lmn}\right)\right|\right\} \tag{3.29}$$

When the data point is between the bounds of smaller than or equal to M-2 attributes, then the shortest distance between the point and the hyper-box is obtained by calculating the minimum distance between edges of the hyper-box and the new point. The number of edges in a hyper-box is equal to $M2^{M-1}$. Thus, $ML2^{M-1}$ *(L:* total number of hyper-boxes*)* distance calculations have to be performed for each new data point and minimum of them must be selected. An edge is a finite segment consists of the points of a line that are between two extreme points X_{lmn} and $X_{lm'n}$. The data point *j* is represented by the vector $\vec{A}_j$ which is composed of a_{jm} values and $\vec{P0}_{lmn}$ and $\vec{P1}_{lmn}$ are the vector forms of two extreme points. The minimum distance between the new data point *j* and one of the segments of the hyper-box determined by two extreme points is calculated using Eq. (3.30) where (·) indicates the dot product of the matrices.

$$\vec{W}_{jlmn} = \vec{A}_j - \vec{P0}_{lmn} \tag{3.30}$$

$$\vec{V}_{jlmn} = \vec{P1}_{lmn} - \vec{P0}_{lmn} \tag{3.31}$$

$$C1_{jlmn} = (W_{jlmn} \cdot V_{jlmn}) / \|W_{jlmn}\|\|V_{jlmn}\| \tag{3.32}$$

$$C2_{jlmn} = (V_{jlmn} \cdot V_{jlmn}) / \|V_{jlmn}\|\|V_{jlmn}\| \tag{3.33}$$

$$b_{jlmn} = C1_{jlmn} / C2_{jlmn} \tag{3.34}$$

$$Pb_jlmn = P0_{jlmn} + b_{jlmn} \cdot V_{jlmn} \tag{3.35}$$

$$\min_{l,n}\left\{\sqrt{\sum_m (a_{jm} - p_{b\,jlmn})^2}\right\} \tag{3.36}$$

When the data point is not within the lower and upper bounds of any attributes defining the box, then the shortest distance between the new point and the hyper-box is calculated using the minimum distance between extreme points of the hyper-box and the new data point. The number of extreme points in a hyper-box is given by 2^M. Therefore, it is necessary to perform $2^M L$ *(L:* total number of hyper-boxes*)* distance calculations for each new data point and select the minimum distance. The minimum distance between the new data point *j* and one of the extreme points of the hyper-box is calculated using Eq. (3.37).

$$\min_{l,n}\left\{\sqrt{\sum_m \left(a_{jm}-X_{lmn}\right)^2}\right\} \qquad (3.37)$$

The following algorithm assign a new data point *j* with attribute values a_{jm} to class *k*:

Step 0: Initialize *inAtt(l,m)*=0.

Step 1: For each *l* and *m*, if

$$X_{lmn} \le a_{jm} \le X_{lmn'} \qquad \forall n = lo, n' = up \qquad (3.38)$$

Set *inAtt(l,m)=inAtt(l,m)*+1.

Step 2: If *inAtt(l,m)=M*, then go to Step 3. Otherwise, continue.
If *inAtt(l,m)≤M-1*, then go to Step 4.

Step 3: Assign the new data point to class *k* where ybc_{lk} is equal to 1 for the hyper-box in Step 2. Stop.

Step 4: Calculate the minimum given by Eq. (3.29) and set the minimum as *min1(l)*. Calculate the minimum given by Eq. (3.36) and set the minimum as *min2(l)*. Calculate the minimum given by Eq. (3.37) and set the minimum as *min3(l)*. Select the minimum between *min1(l)*, *min2(l)* and *min3(l)* to determine the hyper-box *l* that is closest to the new data point *j*. Assign the new data point to class *k* where ybc_{lk} is equal to 1 for the hyper-box *l*. Stop.

The application of the proposed approach on an example is illustrated in the next section.

3.3. Illustrative Example

We applied the mixed-integer programming method on a set of 16 data points in 4 different classes given in Figure 3.2. The data points can be represented by two attributes, 1 and 2.

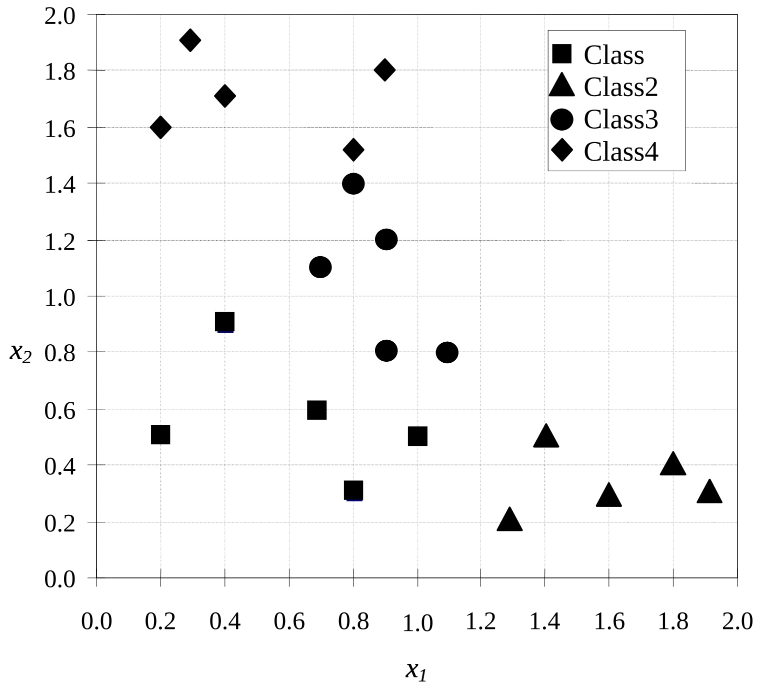


Figure 3.2. Data points in the illustrative example and their graphical representation.

There are a total of 20 data points; 16 of these points were used in training and 4 of them used in testing. The training problem classified the data into 4 four classes using 5 hyper-boxes as shown in Figure 3.3. It is interesting to note that Class1 requires two hyper-boxes while the other classes are represented with a single hyper-box only. The reason for having two hyper-boxes for Class1 is due to the fact that a single hyper-box for this class would include one of the data points that belong to Class3. In order to eliminate inconsistencies in training data set, the method included one more box for Class1.

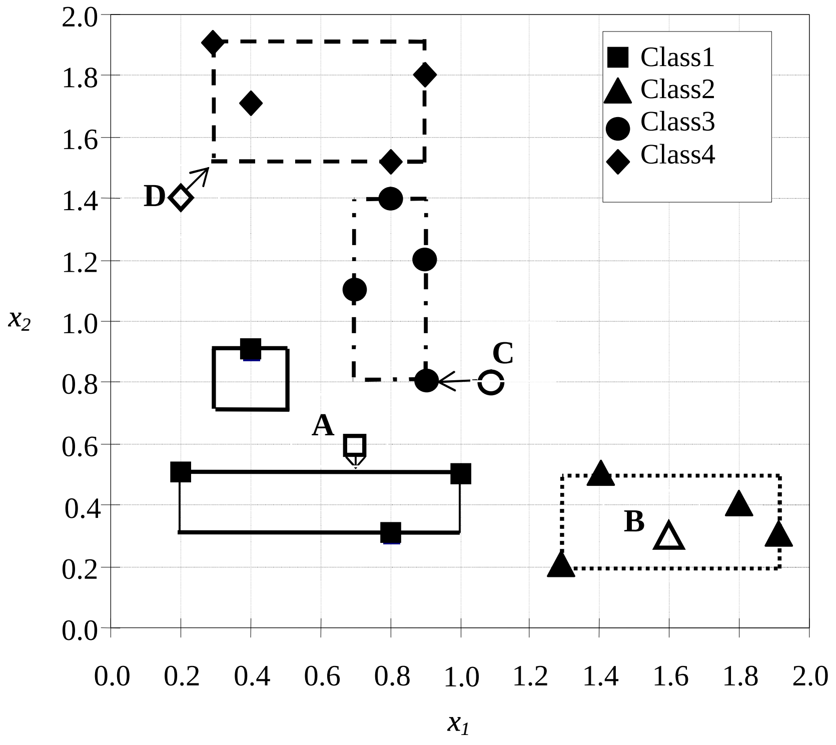


Figure 3.3. Hyper-boxes that classify the data points in the illustrative example.

After the training is successfully completed, the test data is processed to assign them to hyper-boxes that classify the data perfectly. The assignment of the test data point B to Class2 is straightforward since it is included in the hyper-box that classifies Class2 (i.e., *inAtt(l,m)=N* for this data point). The test data in Class1 is assigned to one of the hyper-boxes that classify Class1. Similarly, the test data in Class3 is also assigned to the hyper-box that classifies Class3. Since the test data in these classes are included within the bounds of one of the two attributes, the minimum distance is calculated as the normal to the closest hyper-plane to these data points. In the case of data point that belongs to Class4, it is assigned to its correct class since the closest extreme point of a hyper-box classifies Class4. This extreme point of the hyper-box 5 classifying Class4 is given by ($X_{5,1,lo}$,$X_{5,2,lo}$).

The test problem also classified the data points with 100% accuracy as shown in Figure 3.3.

This illustrative example is also tested by different data classification models existing in the literature in order to compare the results and to measure the performance of the proposed model. Table 3.2 shows the examined models and their outcomes for this small illustrative example.

Table 3.2. Comparison of different classification models for illustrative example.

CLASSIFICATION MODEL	**PREDICTION ACCURACY**	**MISCLASSIFIED SAMPLE(S)**
Neural Networks[a]	75%	A
Support Vector Machines[b]	75%	D
Bayesian Classifier[c]	75%	C
K-nearest Neighbor Classifier[c]	75%	A
Statistical Regression Classifiers[c]	75%	C
Decision Tree Classifier[c]	50%	A, C
Proposed MILP approach	100%	-

[a] iDA implementation in MS Excel[53]. [b] SVM implementation in Matlab [54].
[c] WEKA [55].

Neural Networks, Support Vector Machines, Bayesian, K-nearest Neighbor and Statistical Regression classifiers have only one misclassified instance which leads to 75% accuracy value as shown in Table 3.2. Neural Networks and K-nearest Neighbor classifier predicts the class of test sample A as Class 3. Support Vector Machine method misclassifies test sample D and assigns it to Class 1 while Bayesian and Statistical Regression classifier classifies test sample C as belonging to Class 2. On the other hand, Decision Tree classifier gives the lowest accuracy value (50%) with two misclassifications. Sample A and sample C is classified as Class 3 and Class 2, respectively. Consequently, proposed method in this thesis classifies all of the test samples accurately and achieves 100% accuracy. As a result, suggested mixed-integer programming approach performs better than other data classification methods that are listed in Table 3.2 for this small example. Thus, this new method can be attractive for real life data classification problems. For further investigation to the performance of the developed mixed-integer programming approach, two distinct benchmark problems are examined in the next chapter of the thesis.

Chapter 4

RESULTS

The performance of the proposed multi-class data classification method is evaluated on two important benchmark problems; IRIS and protein folding type. The prediction results and comparison with other data classification methods are examined in this chapter.

4.1. Evaluation of the Method on IRIS Data Set

In this part of the study, the efficiency of the proposed model is tested on the well-known IRIS data set. IRIS data published by Fisher [20] is selected due to the reason that it has been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in centimeters on 50 iris specimens from each of three species, *Iris setosa, I. versicolor,* and *I. virginica*. This data set is extensively studied in the pattern recognition [56-58].

We selected 24 data samples randomly for the training problem where each class is represented by exactly the same number of samples. Table 4.1 shows the training set used in order to solve the mixed-integer programming problem presented in Section 3.1. Using the above 24 samples, the MILP problem defined in Section 3.1 is modeled in GAMS [59] and solved using ILOG CPLEX Solver version 9.0 [60].

Table 4.1. IRIS data training set.

SAMPLES	CLASS I (SETOSA)				CLASS II (VERSICOLOR)				CLASS III (VIRGINICA)			
	SL	SW	PL	PW	SL	SW	PL	PW	SL	SW	PL	PW
1	5.7	4.4	1.5	0.4	5.6	3	4.1	1.3	6.5	3.0	5.8	2.2
2	4.8	3.0	1.4	0.1	5.5	2.3	4.0	1.3	6.3	2.9	5.6	1.8
3	5.2	4.1	1.5	0.1	5.7	3.0	4.2	1.2	6.3	2.5	5	1.9
4	5.5	4.2	1.4	0.2	6.7	3.1	4.7	1.5	6.7	3.1	5.6	2.4
5	4.9	3.1	1.5	0.1	5.8	2.7	4.1	1.0	6.0	3.0	4.8	1.8
6	4.8	3.4	1.6	0.2	5.0	2.3	3.3	1.0	5.8	2.7	5.1	1.9
7	5.4	3.7	1.5	0.2	6.0	2.7	5.1	1.6	6.5	3.2	5.1	2.0
8	5.2	3.5	1.5	0.2	5.0	2.0	3.5	1.0	6.8	3	5.5	2.1

SL: Sepal Length, SW: Sepal Width, PL: Petal Length, PW: Petal Width.

The training problem is solved approximately in 3 CPU seconds on a notebook computer with Inter Pentium M 1.5 GHz Processor and 256 MB of RAM. The characteristics of the constructed model are listed in Table 4.2. The solution of the proposed model with the default settings of CPLEX version 9.0 indicates that the optimal solution is found at the root node in 15.39 CPU seconds. Besides, the solution is obtained without any misclassification of samples, 5 hyper-boxes are formed, one box for Setosa and two boxes for Virginica and Versicolor classes. Table 4.3 gives the bounds on each attribute for constructed hyper-boxes and assigned classes of each box.

Table 4.2. Characteristics of the constructed model.

ITEM	VALUE
# of continuous variables	305
# of binary variables	1,652
# of constraints	2,871
# of nodes	0
# of iterations	1,115
Solver memory (MB)	4.5
CPU time (sec)*	**3.13**

*Notebook computer with Inter Pentium M 1.5 GHz, Processor and 256 MB of RAM.

Table 4.3. Bounds of hyper-boxes constructed by training problem.

HYPER-BOXES & CLASSES	SEPAL LENGTH		SEPAL WIDTH		PETAL LENGTH		PETAL WIDTH	
	Lower	Upper	Lower	Upper	Lower	Upper	Lower	Upper
1 - Setosa	4.7	5.7	3	4.4	1.4	1.6	0.1	0.4
2 - Versicolor	5	5.6	2	3	3.3	4.1	1	1.3
3 - Virginica	5.8	6.7	2.7	3.2	4.8	5.6	1.8	2.4
4 - Versicolor	5.7	6.7	2.7	3.1	4.1	5.1	1	1.6
5 - Virginica	6.3	6.8	2.5	3	5	5.8	1.8	2.2

After classifying the training data perfectly, the test set given in Table 4.4 is studied to assign them to constructed hyper-boxes by applying the method explained in Section 3.2. The assignment of data in the test set to classes is done without a prior knowledge on their membership to a class.

There are three possibilities for a new data: either it is enclosed in an existing hyper-box, or it is enclosed by more than one hyper-boxes, or it is outside the region enclosed by a hyper-box. Data points like data 17 that are within the bounds of the hyper-box 1 are assigned to Setosa class. This assignment is straightforward; however there are also some data points that are not enclosed in any of the hyper-boxes found in the training problem. For example, test data point 18 in Class III (with the attribute values 6.7, 3.3, 5.7, 2.5) is a typical example for this type of data points. The closest hyper-box to this data is calculated using Eqs. (3.29) and (3.30). Then the hyper-box that is closest to this data point is (3); thus it is assigned to the class “Virginica”. On the other hand, there are not any data points enclosed by two or more hyper-boxes for this data set.

Table 4.4. IRIS data test set.

SAMPLES	ClASS I (Setosa)				Class II (Versicolor)				Class III (Virginica)			
	SL	SW	PL	PW	SL	SW	PL	PW	SL	SW	PL	PW
1	4.5	2.3	1.3	0.3	4.9	2.4	3.3	1	6.7	3.3	5.7	2.1
2	5	3.5	1.6	0.6	6.2	2.2	4.5	1.5	7.3	2.9	6.3	1.8
3	4.3	3	1.1	0.1	5.5	2.6	4.4	1.2	4.9	2.5	4.5	1.7
4	5	3.5	1.3	0.3	6	3.4	4.5	1.6	6.9	3.1	5.4	2.1
5	5.1	3.8	1.9	0.4	5.8	2.7	3.9	1.2	5.8	2.8	5.1	2.4
6	5	3.4	1.5	0.2	6.1	2.9	4.7	1.4	6.5	3	5.5	1.8
7	5.1	3.7	1.5	0.4	5.7	2.9	4.2	1.3	7.7	3.8	6.7	2.2
8	5.1	3.8	1.5	0.3	5.9	3	4.2	1.5	5.7	2.5	5	2
9	4.6	3.4	1.4	0.3	6.9	3.1	4.9	1.5	7.2	3	5.8	1.6
10	5.4	3.4	1.7	0.2	5.2	2.7	3.9	1.4	7.7	3	6.1	2.3
11	5.8	4	1.2	0.2	7	3.2	4.7	1.4	6.9	3.2	5.7	2.3
12	4.9	3	1.4	0.2	5.7	2.6	3.5	1	7.4	2.8	6.1	1.9
13	5	3.2	1.2	0.2	6.6	2.9	4.6	1.3	7.2	3.2	6	1.8
14	5	3	1.6	0.2	6	2.9	4.5	1.5	6.4	2.7	5.3	1.9
15	5.1	3.8	1.6	0.2	6.6	3	4.4	1.4	7.9	3.8	6.4	2
16	4.7	3.2	1.3	0.2	6.1	2.8	4	1.3	6.2	2.8	4.8	1.8
17	5.4	3.4	1.5	0.4	6.4	3.2	4.5	1.5	6.4	3.2	5.3	2.3
18	4.6	3.2	1.4	0.2	5.5	2.5	4	1.3	6.7	3.3	5.7	2.5
19	4.9	3.1	1.5	0.2	5.8	2.6	4	1.2	6.4	2.8	5.6	2.1
20	5	3.4	1.6	0.4	6.4	2.9	4.3	1.3	6.3	2.8	5.1	1.5
21	4.4	2.9	1.4	0.2	5.5	2.4	3.8	1.1	6.5	3	5.2	2
22	4.6	3.6	1	0.2	5.1	2.5	3	1.1	6.8	3.2	5.9	2.3
23	5.1	3.3	1.7	0.5	5.6	2.5	3.9	1.1	7.7	2.8	6.7	2
24	5	3.3	1.4	0.2	6.3	3.3	4.7	1.6	6.1	3	4.9	1.8
25	5.1	3.5	1.4	0.2	5.4	3	4.5	1.5	6.1	2.6	5.6	1.4
26	5	3.6	1.4	0.2	6.7	3.1	4.4	1.4	7.1	3	5.9	2.1
27	5.4	3.9	1.3	0.4	5.6	3	4.5	1.5	5.6	2.8	4.9	2
28	5.1	3.3	1.7	0.5	5.6	2.7	4.2	1.3	6.3	3.3	6	2.5
29	5.5	3.5	1.3	0.2	6	2.2	4	1	6.4	2.8	5.6	2.2
30	4.8	3.4	1.9	0.2	5.9	3.2	4.8	1.8	6.3	2.7	4.9	1.8
31	4.6	3.1	1.5	0.2	6.3	2.5	4.9	1.5	6.4	3.1	5.5	1.8
32	5.1	3.5	1.4	0.3	5.7	2.8	4.1	1.3	7.7	2.6	6.9	2.3
33	5.3	3.7	1.5	0.2	5.6	2.9	3.6	1.3	6.9	3.1	5.1	2.3
34	4.8	3.1	1.6	0.2	6.5	2.8	4.6	1.5	6.7	3	5.2	2.3
35	4.7	3.2	1.6	0.2	6.1	3	4.6	1.4	6.3	3.4	5.6	2.4
36	4.8	3	1.4	0.3	5.7	2.8	4.5	1.3	6	2.2	5	1.5
37	5.1	3.4	1.5	0.2	5.5	2.4	3.7	1	7.2	3.6	6.1	2.5
38	5.7	3.8	1.7	0.3	6.1	2.8	4.7	1.2	5.8	2.7	5.1	1.9
39	5.2	3.4	1.4	0.2	6.8	2.8	4.8	1.4	6.2	3.4	5.4	2.3
40	4.4	3.2	1.3	0.2	6.2	2.9	4.3	1.3	7.6	3	6.6	2.1
41	4.4	3	1.3	0.2	6.7	3	5	1.7	5.9	3	5.1	1.8
42	5.4	3.9	1.7	0.4	6.3	2.3	4.4	1.3	6.7	3	5	1.7

For each of the data in the test set, the method is applied and the test data is assigned classes. Then, the accuracy of the proposed model is checked by comparing the assigned and original classes of samples. At the end of the comparisons, it is realized that 123 samples are assigned correctly to their original classes. On the other hand, 3 samples, data 30 of Class II with attributes (5.9, 3.2, 4.8, 1.8); data 20 and data 42 of Class III with attributes (6.3, 2.8, 5.1, 1.5) and (6.7, 3, 5, 1.7), are misclassified. Table 4.5 summarizes the above results by the help of confusion matrix.

Table 4.5. Classification performances.

Assigned / **Original**	**SETOSA**	**VERSICOLOR**	**VIRGINICA**
SETOSA	42	0	0
VERSICOLOR	0	41	1
VIRGINICA	0	2	41

Table 4.6. Comparison of different classification models for IRIS data set.

CLASSIFICATION MODEL	**PREDICTION ACCURACY**
Castro's Method [21]	96.6%
Hong-and-Lee's Method [23]	95.57%
Wu-and-Chen's Method [25]	96.21%
Chen-and-Fang's Method [26]	97.33%
Back-propagation Neural Networks [19]	96.7%
Obj. Func.-based Unsupervised Neural Networks [19]	90.7%
Fuzzy C-means [19]	90%
Tolerant Rough Set [19]	98%
SVM (one-against-all) [30]	96%
SVM (one-against-one) [30]	97.33%
SVM (DAG) [30]	97.33%
Proposed MILP approach	97.62%

From Table 4.5, it can be seen that, the overall accuracy of the proposed model to the IRIS data set is 97.62%. A comparison of the average classification accuracy rate of the proposed model with that of the existing methods is shown in Table 4.6. Castro et al. [21] solved IRIS data classification problem by presenting a method to learn maximal structure rules in fuzzy logic and achieved 96.6% accuracy. While dealing with IRIS problem, Hong et al. [23] develop a model to generate fuzzy rules and membership functions from training examples. Their approach can only reach the accuracy value of 95.57%. Wu et al. [25] use the similar fuzzy rules and membership function idea with some extensions and increase the accuracy to 96.21%. Most recently, Chen et al. [26] present a new model based on distributions of training instances. Two useful attributes of the IRIS data from training instances that are more suitable to deal with are found by some statistical calculations. Their proposed model achieves an accuracy value of 97.33%. Models proposed by Castro et al. [21], Wu et al. [25] and Chen et al. [26] are very simple and not time consuming. However, they do not have sufficient accuracy values with respect to the other developed models for IRIS data classification problem. Kim [19] proposes a new data classification method based on the tolerant rough set that extends the existing equivalent rough set. In this study, Kim compares tolerant rough set (TRS) approach with the back-propagation neural networks (BPNN), objective function-based unsupervised neural networks (OFUNN) and fuzzy C-means (FCM) models. Their accuracy values are 98.0, 96.7, 90.7, and 90 in the case of TRS, BPNN, OFUNN and FCM, respectively [19]. Hsu and Lin [30] examine the SVM methods used for multi-class data classification. In this research, they compare the performance values of binary classification based methods such as “one-against-all”, “one-against-one” and direct acyclic graph (DAG) SVM. The accuracy values of multi-class SVM methods’ used by Hsu and Lin are listed in Table 4.6. “one-against-one” and DAG SVM methods achieve the accuracy value of 97.33% and dominate the “one-against-all” SVM method which is 96% accurate on IRIS data set.

When we compare the classification performances of different classification algorithms listed in Table 4.6, the proposed mixed-integer programming (MIP) approach in this thesis is accurate and efficient. Moreover, in the case of simplicity, MIP approach is simpler and easily understandable than the other approaches like SVM, TRS, BPNN, OFUNN and FCM. Furthermore, there is no need to do many iterations and the solution time of MIP approach is very short (3 CPU seconds).

4.2. Evaluation of the Method on Protein Folding Type Data Set

The correlation between amino acid composition and protein folding types has been widely studied during last decade [41-49]. Knowledge of fractions of 20 amino acids is considered to be sufficient alone for predicting the structural class of a given protein [40]. Since the improvement in the prediction of structural classes is used for the discovery of the functions of proteins, this discovery will lead to better understanding of biological systems. Therefore, protein folding type problem is an important problem in bioinformatics and computational biology.

4.2.1 Proteins

Proteins are large molecules indispensable for existence and proper functioning of biological organisms. Proteins are used in structure of cells, which are main constituents of larger formations like tissues and organs. Besides their necessity for structure, they are also required for proper functioning and regulation of organisms such as enzymes, hormones, antibodies. A protein molecule is the chain(s) of amino acids (also called residues). A typical protein (Figure 4.1) contains 200 – 300 amino acids but this may increase up to 27,000 in a single chain.

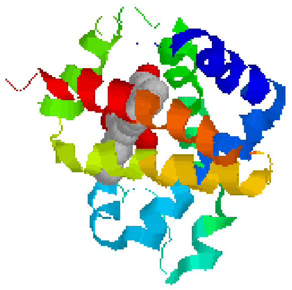


Figure 4.1. A typical protein molecule.

Amino acids, subunits of proteins, are organic compounds that contain a basic amino (NH_2) group, an acidic carboxyl (COOH) group and a side chain attached to an alpha carbon atom. Although carboxyl and basic amino groups of all amino acids are the same, their side chains differ providing diversity. There are 20 types of amino acids in nature and their names, 3-letter representations and single-letter representations are provided in Table 4.7.

Table 4.7. List of amino acids, their three-letter and single-letter representations.

Amino Acid	**Three-Letter Representation**	**Single-Letter Representation**	**Amino Acid**	**Three-Letter Representation**	**Single Letter Representation**
alanine	ALA	A	leucine	LEU	L
arginine	ARG	R	lysine	LYS	K
asparagine	ASN	N	methionine	MET	M
aspartic acid	ASP	D	phenylalanine	PHE	F
cysteine	CYS	C	proline	PRO	P
glutamic acid	GLU	Q	serine	SER	S
glutamine	GLN	E	threonine	THR	T
glycine	GLY	G	tryptophan	TRP	W
histidine	HIS	H	tyrosine	TYR	Y
isoleucine	ILE	I	valine	VAL	V

Residue content and order in chain(s) is unique for each protein just like specificity of gene sequence. The sequence and types of side chains determine all properties of protein; its three-dimensional shape, and chemical and biological functions. Residue order is determined by the base sequence of nucleotides (building blocks of genes) in the gene coding the protein.

Starting with the sequence of residues in the chain(s) making up protein, there are 4 basic structural phases: primary structure, secondary structure, tertiary structure and quaternary structure.

The primary structure is the sequence of amino acids that the protein molecule consists of. The primary structure of protein “1ABA” which has a total of 87 residues is showed in Figure 4.2. Only the plain chain(s) formed by attachment of amino acids to each other is considered; higher dimensional shapes, arrangement of peptide bonds, angles between chemical bonds in amino acids and interactions between any parts of residues are all ignored. The amino acid content and order dictates the shape of the protein molecule, its spatial and biochemical properties.

1ABA: MFKVYGYDSN IHKCGPCDNA KRLLTVKKQP FEFINIMPEK
GVFDDEKIAE LLTKLGRDTQ IGLTMPQVFA PDGSHIGGFD QLREYFK

Figure 4.2. Primary structure of protein 1ABA

The secondary structure (folding type) of a segment of polypeptide chain is the local spatial arrangement of its main-chain atoms without regard to the conformation of its side chains or to its relationship with other segments. This is the shape formed by amino acid sequences due to interactions between different parts of molecules. There are mainly three types of secondary structural shapes: α-helices, β-sheets and other structures connecting these such as loops, turns or coils. Alpha-helices are spiral strings formed by hydrogen bonds between CO and NH groups in residues. Beta-sheets are plain strands formed by stretched polypeptide backbone. When β-sheets come together, hydrogen bonds form between C=O and NH groups of residues of adjacent chains, keeping them together. Connecting structures do not have regular shapes; they connect α-helices and β-sheets to each other.

Proteins are classified according to their secondary structure content, considering α-helices and β-sheets. Levitt and Chothia [39] were the first to propose such a classification with four basic types. "All-alpha" proteins consist almost entirely of α-helices. "All-beta" are the ones composed mostly of β-sheets in their secondary structures. There are two intermediate classes which have mixed α-helices and β-sheets. "Alpha/beta" proteins have approximately alternating, mainly parallel segments of α–helices and β-sheets. The last class, "alpha+beta" has mixture of all alpha and all beta regions, mostly in an antiparallel fashion. Table 4.8 gives the definitions of four structural classes of proteins in detail.

Table 4.8. Definitions of Protein Structural Classes [42].

Structural Class	**Proportion of α-helices**	**Proportion of** b**-sheets**
All-α proteins	≥ 40%	≤ 5%
All-b **proteins**	≥ 40%	≤ 5%
α+b **proteins** (≥%60 antiparallel β-sheets)	≥ 15%	≥ 15%
α/b **proteins** (≥%60 parallel β-sheets)	≥ 15%	≥ 15%

4.2.2 Evaluation of the Method

A good training database is important for improving the accuracy of prediction. For this reason, the selection of proteins for the training database is carried out according to following criteria: (i) a typical or distinguishable feature for each of the folding types concerned, (ii) a good quality of structure, (iii) as many non-homologous structures as possible. 120 structure-known proteins are selected and classified into 30 α, 30 β, 30 α+β, 30 α/β proteins (Table 4.9).

Table 4.9. The PDB (Protein Data Bank) codes of the 4x30 = 60 representative proteins in the training database[a]

Instances	All α	All β	α+β	α/β
1	1AVH,A*	1ACX,A	1CTF,A	1ABA,A
2	1BAB,B	1CD8,A	1DNK,A	1BKS,B
3	1C5A,A	1CDT,A	1EME,A	1CIS,A
4	1CPC,A	1CID,A	1FXI,A	1DBP,A
5	1CPC,L	1DFN,A	1FXI,B	1DHR,A
6	1ECO,A	1HIL,A	1FXI,C	1EAF,A
7	1FCS,A	1HIV,A	1FXI,D	1ETU,A
8	1FHA,A	1MAM,H	1HSB,A	1GPB,A
9	1FIA,B	1PAZ,A	1LTS,A	1KKJ,A
10	1HBG,A	1REI,A	1PPN,A	1OFV,A
11	1HBB,A	1TEN,A	1RND,A	1OVB,A
12	1HIG,A	1TFG,A	2AAK,A	1PFK,A
13	1LE4,A	1TLK,A	2ACH,A	1Q21,A
14	1LTS,C	2ALP,A	2ACT,A	1S01,A
15	1MBC,A	2AVI,A	2PHY,A	1SBP,A
16	1MBS,A	2AYH,A	2PRF,A	1SBT,A
17	1RPR,A	2CTX,A	2RNT,A	1TIM,A
18	1POC,A	2RAC,A	2VAA,A	1TRE,A
19	1TRO,A	2LAL,A	3IL8,A	1ULA,A
20	256B,B	2ILA,A	3MON,A	2CTC,A
21	2CCY,A	2OMP,A	3SC2,A	2FOX,A
22	2LH1,A	2SNV,A	3SIC,I	2HAD,A
23	2LHB,A	2VAA,B	3SSI,A	2LIV,A
24	2LIG,A	3CD4,A	4BLM,A	2PGD,A
25	2ZTA,A	3HHR,C	4LZT,A	2TMD,A
26	2MHB,A	4GCR,A	4TMS,A	3GBP,A
27	2MHB,B	7API,B	5HOH,A	5CPA,A
28	3HDD,A	8FAB,A	5TLI,A	5P21,A
29	4MBA,A	8FAB,B	8CAT,A	8ABP,A
30	4MBN,A	8I1B,A	9RSA,A	8ATC,A

[a] The PDB code is constituted by the first four characters according to Brookhaven National Labroratory, and the fifth character after the comma used here to indicate a specific chain of a protein.

Composition values of training proteins are listed in Appendix A. The training dataset is a corrected and reorganized version of the same sets of proteins as used by Chou [41] and Bahar et al. [42]. Using the 120 training set samples, the proposed model is solved in GAMS [59] by using the solver ILOG CPLEX Solver version 9.0 [60].

The training problem is solved approximately in 210 CPU seconds on a notebook computer with Inter Pentium M 1.5 Ghz Processor and 256 MB of RAM. The characteristics of the constructed model for protein folding type problem are listed in Table 4.10. The solution of the proposed model with the default settings of CPLEX version 9.0 indicates that the optimal solution is found at the root node in 210 CPU seconds. Without any misclassification of samples, 16 hyper-boxes are formed, (4 boxes for each class). The bounds on each attribute for constructed hyper-boxes and assigned classes of each box are given in Appendix B.

Table 4.10. Characteristics of the constructed model.

ITEM	VALUE
# of continuous variables	16580
# of binary variables	22036
# of constraints	27516
# of nodes	0
# of iterations	10176
Solver memory (MB)	22
CPU time (sec)*	210

*Notebook computer with Inter Pentium M 1.5 GHz, Processor and 256 MB of RAM.

The results of the self-consistency test of the proposed MILP method and other methods used the same training data set for protein folding type problem are listed in Table 4.11. These rates are training accuracy values indicating the models performance of clustering of proteins of different classes. Singular Value Decomposition (SVD) method presented by Bahar et al. [42] groups All α and All β classes with 100% accuracy. However, some proteins belonging to α+β and α/β classes are misclassified. Thus, the training accuracies of these classes are only 96.7% and 93.3% for α+β and α/β classes, respectively. SVD method's overall training accuracy is 97.5%. On the other hand, Cai et al. [48] suggested a Neural Network approach which leads to lower training accuracy values with respect to the other applied methods. Overall rate of correct prediction of each class in the training set is only 93.5% with NN approach. Component Coupled (CC) algorithm is introduced by Chou et al. [45] and training set performance of this method is higher than NN method but not sufficient at all with a value of 94.2%. The prevailing result is observed by using Support Vector Machine (SVM) method suggested by Cai et al. [45]. All of the proteins in the training set are classified without any misclassification. Thus, 100% overall prediction accuracy is achieved. Similar to SVM approach, the proposed Mixed-Integer Linear Programming (MILP) approach classifies all of structural classes with 100% accuracy.

Table 4.11. Results of self-consistency test.

Algorithm	**All α**	**All** b	**α**+b	**α**/b	**Overall**
SVD [42]	100%	100%	96.7%	93.3%	97.5%
NN [48]	98.6%	93.4%	96.3%	84.6%	93.5%
SVM [49]	100%	100%	100%	100%	**100%**
CC [45]	95.7%	93.4%	95.1%	92.3%	94.2%
MIP	100%	100%	100%	100%	**100%**

After classifying the training data perfectly, the test set given in Table 4.12 is studied to assign them to constructed hyper-boxes by applying the testing problem algorithm. The assignment of data in the test set to structural classes is done without a prior knowledge on their membership to a class. The testing data set is also a corrected and reorganized version of the same sets of proteins as used by Chou [41] and Bahar et al. [42].

Table 4.12. Testing set composed of 60 instances.

All α	All β	α+β	α/β
1AJH,A	1AB9,B	1BW4,A	1M6B,A
1BBL,A	1ATX,A	1CYO,A	1NIP,B
1HUU,A	1BJ8,A	1DNK,A	1SBP,A
1IFA,A	1BLN,B	1GLA,G	1XAD,A
1MDN,A	1COB,A	1OVO,A	2MIN,A
1MRR,A	1EGF,A	1QR5,A	4ICD,A
1PRC,H	1EPP,E	1SHA,A	7AAT,A
1RNR,A	1HCC,A	1THO,A	9RUB,B
4CPV,A	1IXA,A	1UUG,A	
	1MDA,A	1WQM,A	
	1NN2,A	1XOB,A	
	1SHF,A	2AAA,A	
	1TIE,A	2ABH,A	
	1TNF,A	2PIA,A	
	2ACH,B	2MS2,A	
	2SOD,B	2SN3,A	
	3VGC,B	2TAA,A	
	43CA,A	2TDM,A	
	5NN9,A	3COX,A	
	3SC2,B	4ENL,A	
		4INS,B	
		4RCR,H	
		4RHN,A	

*The letters after the comma represents the chain of the protein.

As mentioned in testing problem formulation part, there are three possible situations for a new protein to be settled down: either it is enclosed in an existing hyper-box, or it is between the bounds of more than one hyper-box, or it is outside the region enclosed by a hyper-box constructed in the training part. Proteins that are within the bounds of a hyper-box are assigned directly to the representing class of that box. This assignment is easy and simple; however some proteins are not enclosed in any of the hyper-boxes found in the training problem. In this case, closest hyper-box to this protein is calculated using Eqs.

(3.29) and Eqs. (3.30). On the other hand, some proteins are enclosed by two hyper-boxes belonging to different classes. In that case, these proteins are assumed to classified with 50% percent and the value 0.5 is added to the total number of correct classification. For each member of the test data set, testing algorithm is applied and an assignment to a structural class is done. After all, the accuracy of the developed model is checked by comparing the original and assigned structural classes of proteins. At the end of the testing, it is realized that 33 proteins in the test set are correctly classified. On the other hand, 13 proteins are misclassified and 8 proteins are enclosed by two hyper-boxes. Table 4.13 shows the confusion matrix of the protein folding type problem.

Table 4.13. Confusion Matrix.

Assigned / **Original**	**All α**	**All β**	**α+β**	**α/β**
All α	7	2	0	0
All β	1	16	3	1
α+β	3.5	2	15.5	2
α/β	0	2	1.5	4.5

The overall accuracy of the proposed model on the protein folding type problem is 71.66%. A comparison of the average classification accuracy rate of the developed model with that of the existing methods is shown in Table 4.14.

Table 4.14. Results of test set.

Method	**All α**	**All** b	**α**+b	**α**/b	**Overall**
SVD [42]	66.7%	90.1%	81%	66.7%	81%
NN [48]	68.6%	85.2%	86.4%	56.9%	74.7%
SVM [49]	74.3%	82%	87.7%	72.3%	79.4%
CC [45]	84.3%	82%	81.5%	67.7%	79.1%
MIP	77.7%	76.2%	66.1%	56.25%	**71.66%**

NN approach suggested by Cai et al. [48] gives prediction accuracy results with 74.7%. SVM method applied by Cai et al. [48] and CC algorithm proposed by Chou et al. [45] have accuracy values very close to each other, 79.4% and 79.1%, respectively. Furthermore, SVD method applied by Bahar et al. [42] gives an accuracy value of 81%. On the other hand, the developed model in this thesis reaches the accuracy value of 71.66%. In order to compare these supervised classification models, the P-value analysis based on hypothesis testing are carried out. P value represents the difference between two models with 95% confidence. If P value is greater than 2, the difference between the results of the models is not due to chance. Otherwise, the accuracies of the models are very close to each other and no significant improvement achieved. P value can be calculated using Eq. 4.1. In this equation, E_1 and E_2 are the error rates of two models; q is the average of error rates; n_1 and n_2 are the number of instances in the test sets of two models.

$$P = \frac{|E_1 - E_2|}{\sqrt{q(1-q)(1/n_1 + 1/n_2)}} \qquad 4.1$$

For each of the methods in Table 4.14, MIP approach is one by one tested and P values are calculated (Table 4.15). As it can be seen from the Table 4.15, the P values are not greater than 2. Thus the difference between the methods SVD, NN, SVM, CC and proposed MIP method is not significant. Therefore, the proposed approach can also be used for protein folding type prediction.

Table 4.15. P values of the compared models.

Compared Models	**Parameters**	**P value**
SVD - MIP	E_1=0.19, E_2=0.2834, n_1=n_2=60	1.2
NN – MIP	E_1=0.253, E_2=0.2834, n_1=277, n_2=60	0.48
SVM – MIP	E_1=0.206, E_2=0.2834, n_1=277, n_2=60	1.26
CC - MIP	E_1=0.209, E_2=0.2834, n_1=n_2=60	1.21

Chapter 5

CONCLUSION

Every decision process depends critically on the ability of data acquisition, process and storage. Since realization of underlying behavior of the system that the data comes from is very important, data mining, the process of finding useful patterns in data, plays a critical role in data analyses. There exist a large number of data mining methods and implementations. Data classification is one of the important data mining approaches that tries to analyze and categorize the data in distinct classes. In this thesis study, a new method for multi-class data classification problem is proposed and the performance of this approach is compared with existing multi-class data classification methods on two widely used challenging data sets; the well-known IRIS data set and protein folding type data set.

The developed multi-class data classification approach based on mixed-integer programming is described in Chapter 3. In the training part of the proposed approach, the characteristics of data points belonging to a certain class are determined by the construction of hyper-boxes. The hyper-boxes define the boundaries of the classes that include all or some of the points in that set. In order to represent the existence of hyper-boxes and their boundaries, a mixed-integer programming model is developed. In addition, the relationships among the discrete decisions in the model are presented using propositional logic and then converted to their equivalent integer constraints using Boolean algebra. After distinguishing characteristics of the classes are determined in the training part, the performance of the model is tested by the algorithm introduced in testing problem formulation part. If a new data point with an unknown membership arrives, it is necessary to assign this data point to one of the classes. There are three possible situations for a new protein to be positioned: either an existing hyper-box encloses the data point, or two or more hyper-boxes enclose the data point or it is outside the region enclosed by a hyper-box constructed in the training part. Data points that are within the bounds of a hyper-box are assigned directly to the representing class of that box. For the data points that are enclosed by two or more hyper-boxes, the value added to the number of correctly classified data points is determined by the percentage of correctly assigned hyper-boxes. On the other hand, closest hyper-box is calculated using Eqs. (3.29) and Eqs. (3.30) for the data points that are not enclosed in any of the hyper-boxes found in the training problem. For each member of the test data set, testing algorithm is applied and assignments to a class are done. After all, by checking the original classes of the test set samples the performance of the developed model is evaluated. The proposed model is illustrated on a small example consists of 16 samples in the training and 4 samples in the test set. By this illustrative example, the main steps of the developed mixed-integer programming approach are understood. Moreover, the comparison of the results of distinct models available for data classification is performed. The suggested model's result is accurate and efficient in this small example with regard to the other models listed in Table 3.2.

In Chapter 4, the applications of the proposed approach on two different benchmark data sets are illustrated. One of them is the IRIS data set which is widely studied in pattern recognition and discriminant analysis. When we compare the classification performances of distinct classification algorithms used this data set, developed MIP approach performs better result. Castro's [21], Hong-and-Lee's [23], Wu-and-Chen's [25] and Chen-and-Fang's [26] membership function based methods are simple; however they do not have sufficient accuracy values with respect to the other models. On the other hand, TRS approach suggested by Kim [19] has a rather high accuracy but it is more complicated than the other approaches. In addition, multi-class SVM methods [30] are not only time consuming methods but also own time consuming calculations. To sum up, the proposed MIP approach can be attractive with its simplicity and accuracy for this well-known benchmark problem.

Protein folding type problem is an important research topic in bioinformatics and computational biology. Thus, the developed model is tested on widely studied protein folding type data set. Since the model well constructs the boundaries of the classes, the training part self-consistency test results are higher than the methods such as SVD [42], NN [48] and CC [45]. The result of this data set shows that the proposed MIP approach with 71.66% accuracy is as accurate as the SVD, SVM, NN and CC methods.

In conclusion, this thesis introduces a new accurate mathematical programming method for multi-class data classification problem. One of the most important characteristics of the proposed approach is allowing the use of hyper-boxes for defining the boundaries of the classes that enclose all or some of the points in that set. In other words, if necessary, more than one hyper-box is constructed for a specific class through the training part studies. Moreover, well-construction of the boundaries of each class provides the lack of misclassifications in the training set and indirectly improves the accuracy of the model. In addition, the model does not need to know the underlying distribution of the training data set and learns from the training set in a reasonable time. With only one parameter to be initialized, the suggested model is simple and easily understandable. Furthermore, the proposed model can be used for both binary and multi-class data classification problems without any modifications or additions. Hence, the performance of the model does not depend on the class related changes. The accuracy, simplicity and understandability of the proposed model are favorable. In overall, the suggested model is attractive with respect to above characteristics.

The advantage of the mathematical programming approach in the context of supervised classification lies in its power to model more complex real world problems.

Future studies should further evaluate the performance of the proposed approach on different complex real data sets. Moreover, when the size of training data set increases, proposed approach needs more computational time to solve the training part problem and construct the hyper-boxes for the classes. Thus, this computational complexity of the model should be studied and improved. Moreover, since the protein folding type problem has 20 attributes, some Eigen value analyses could be performed and the most prevailing attributes could be determined in order to decrease the number of attributes which lead us to small size problems. This model size decreasing works could also be used in real life problems.

BIBLIOGRAPHY

[1] M. S. Chen, J. Han, and P. S. Yu, Data Mining: An overview from a database perspective, IEEE Transactions Knowledge and Data Engineering, 8 (1996), 866-883.

[2] R. J. Roiger, and M. W. Geatz, Data Mining-A Tutorial Based Primer, Addison Wesley Press, (2003).

[3] H. Edelstein, Building Profitable Customer Relationships with Data Mining, Two Crows Corporation, (2003).

[4] A. Jagota, Data Analysis and Classification for Bioinformatics, Bay Press, (2000).

[5] J. Adem, W. Gochet, Mathematical programming based heuristics for improving LP-generated classifiers for the multi-class supervised classification problem, European Journal of Operational Research ,168 (2006), 181-199.

[6] S. M. Weiss, and C. A. Kulikowski, Computer systems that learn: classification and prediction methods from statistics, neural networks, machine learning and expert systems, Morgan Kaufmann, San Mateo, CA, (1991).

[7] D. J. Hand, Discrimination and Classification, John Wiley, Chichester, (1981).

[8] P. Lachenbruch, and R. Mickey, Estimation of error rates in discriminant analysis, Technometrics, 10 (1968), 1–11.

[9] M. G. Kendall, A. Stuart, and J. K. Ord, The advanced Theory of Statistics, Volume 3, Design and Analysis and Time Series, Chapter 44, Griffin, London, (1983).

[10] J. Hertz, A. Krogh, and R. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, (1991).

[11] P. A. Devijver, and J. V. Kittler, Pattern Recognition: A Statistical Approach, Prentice Hall, Englewood Cliffs, (1982).

[12] G. J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, John Wiley, New York, (1992).

[13] G. Carpenter, S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, Comput. Vision Graphics Image Understanding, 37 (2) (1987), 54-115.

[14] C. T. Lin, and C. S. G. Lee, Neural-network-based fuzzy logic control and decision system, IEEE Transactions on Computers, 40 (12) (1991), 1320-1336.

[15] P. Simpson, Fuzzy min-max neural networks- Part 1: Classification, IEEE Transactions Neural Networks, 3 (5) (1992), 776-786.

[16] Z. Pawlak, Rough set, International Journal Foundations of Computer Science, 11(1982), 341-356.

[17] J. Banzan, H. Nguyen, A. Skowron, J. Stepaniuk, Some logic and rough set applications for classifying objects, ICS Research Report No. 34, Warsaw University of Technology, (1994).

[18] H. Nguyen, A. Skowron, P. Synak, Discovery of data patterns with applications to decomposition and classification problems, Rough Sets in Knowledge Discovery, Physica-Verlag, Wurzburg, (1998).

[19] D. Kim, Data classification based on tolerant rough set, Pattern Recognition, 34 (2001), 1613-1624.

[20] R. Fisher, The use of multiple measurements in taxonomic problem, Ann. Eugenics, 7 (1936), 179-188.

[21] J. L. Castro, J. J. Castro-Schez, and J. M. Zurita, Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems, Fuzzy Sets and Systems, 101 (1999), 331-342.

[22] Y. C. Chen, and S. M. Chen, Constructing membership functions and generating fuzzy rules using genetic algorithms, Proceedings of the 2001 Ninth National Conference on Fuzzy Theory and Its Applications, Chungli, TAoyuan, Taiwan, Republic of China, (2001), 195-200.

[23] T. P. Hong, and C.Y. Lee, Induction of fuzzy rules and membership functions from training examples, Fuzzy Sets and Systems, 84 (1996), 33-47.

[24] H. L. Lin, and S. M. Chen, A new method for generating weighted fuzzy rules from training instances using genetic algorithms, Proceedings of the 6th Conference on Artificial Intelligence and Applications, Kaohsiung, Taiwan, Republic of China, (2001), 628-633.

[25] T. P. Wu, S. M. Chen, A new method for constructing membership functions and fuzzy rules from training examples, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 29 (1999), 25-40.

[26] S. M. Chen, and Y. D. Fang, A new approach for handling the Iris data classification problem, International Journal of Applied Science and Engineering, 1 (2005), 37-49.

[27] Y. Yajima, Linear programming approaches for multi category support vector machines, European Journal of Operational Research, 162 (2) (2005), 514-531.

[28] C. Cortes, V. Vapnik, Support vector network, Machine Learning, 20 (1995), 273-297.

[29] H. Kim, S. Pang, H. Je, D. Kim, S.Y. Bang, Constructing support vector machine ensemble, Pattern Recognition, 36 (2003), 2757-2767.

[30] C. W. Hsu, and C. J. Lin, A comparison of methods for multi-class support vector machines, IEEE Transactions on Neural Networks, 13 (2) (2002), 415-425.

[31] E. A. Joachimsthaler and A. Stam, Mathematical programming approaches for the classification problem in two-group discriminant analysis, Multivariate Behavioral Research, 25(1990), 427-454.

[32] S.S. Erenguc, G.J. Koehler, Survey of mathematical programming models and experimental results for linear discriminant analysis, Managerial and Decision Economics, 11 (1990), 215-225.

[33] G. J. Koehler, Considerations for mathematical programming models in discriminant analysis, Managerial and Decision Economics, 11 (1990), 227-234.

[34] S.M. Bajgier, A.V. Hill, An experimental comparison of statistical and linear programming approaches to the discriminant problem, Decision Sciences, 13 (1982), 604-618.

[35] W.V. Gehrlein, General mathematical programming formulations for the statistical classification problem, Operations Research Letters, 5 (6) (1986), 299-304.

[36] J.M. Littschwager, C. Wang, Integer programming solution of a classification problem, Management Science, 24 (14) (1978), 1515-1525.

[37] A. Stam, E.A. Joachimsthaler, A comparison of a robust mixed-integer approach to existing methods for establishing classification rules for the discriminant problem, European Journal of Operations Research, 46 (1) (1990), 113-122.

[38] D. Tax, R. Duin, Using two-class classifiers for multi class classification, Proceedings 16th International Conference on Pattern Recognition, Quebec City, Canada, Vol. II, IEEE Computers Society Press, Los Alamitos, (2002), 124-127.

[39] M. Levitt, and C. Chothia, Structural patterns in globular proteins, Nature, 261 (1976), 552-558.

[40] H. Nakashima, K. Nishikawa, and T. Ooi, The folding type of a protein is relevant to the amino acid composition, J. Biochem., 99 (1986), 152-162.

[41] K. C. Chou, Does the folding type of a protein depend on its amino acid composition?, FEBS Letters, 363 (1995), 127-131.

[42] I. Bahar, A. R. Atilgan, R. L. Jernigan, and B. Erman, Understanding the recognition of protein structural classes by amino acid composition PROTEINS: Structure, Function, and Genetics, 29 (1997), 172-185.

[43] F. Eisenhaber, C. Frommel, and P. Argos, Prediction of secondary structural content of proteins from their amino acid composition alone. II. The paradox with secondary structural class PROTEINS: Structure, Function, and Genetics, 25 (1996), 169-179.

[44] G. P. Zhou, An intriguing controversy over protein structural class prediction Journal of Protein Chemistry, 17 (1998), 729-738.

[45] K. C. Chou, W. M. Liu, G. M. Maggiora, and C. T. Zhang, Prediction and classification of domain structural classes, PROTEINS: Structure, Function, and Genetics, 31 (1998), 97-103.

[46] Y. D. Cai, Is it a paradox or misinterpretation?, PROTEINS: Structure, Function, and Genetics, 43 (2001), 336-338.

[47] W. S. Bu, Z. P. Feng, Z. Zhang, and C. T. Zhang, Prediction of protein (domain) structural classes based on amino-acid index, Eur. J. Biochem., 266 (1999), 1043-1049.

[48] Y. D. Cai, and G. P. Zhou, Prediction of protein structural classes by neural network, Biochimie., 82 (2000), 783-785.

[49] Y. D. Cai, X. J. Liu, X. B. Xu, and G. P. Zhou, Support vector machines for predicting protein structural class, BMC Bioinformatics, 2 (3) (2001).

[50] T.M. Cavalier, P.M. Pardalos, and A.L. Soyster, Modeling and integer programming techniques applied to propositional calculus, Computers and Operational Research, 17 (6) (1990), 561-570.

[51] R. Raman, and I.E. Grossmann, Relation between MILP modeling and logical inference for chemical process synthesis, Computers and Chemical Engineering, 15 (2) (1991), 73-84.

[52] M. Turkay, and I.E. Grossman, Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs-multiple size regions, Industrial and Engineering Chemistry Research, 35 (1996b), 2611-2623.

[53] iData Analyzer, Version 2.0, Information Acumen Corporation.

[54] G. Cawley, Matlab Support Vector Machine Toolbox, School of Information Systems, University of East Anglia.

[55] WEKA (Waikato Environment for Knowledge Analysis), Version 3.4.5, University of Waikato, New Zealand, (199-2005).

[56] B. Bay, The UCI KDD Archive, Department of Information and Computer Science, University of California, Irvine, CA, (1999) (downloadable from website http://kdd.ics.uci.edu).

[57] H. Kim, S. Pang, H. Je, D. Kim, and S.Y. Bang, Constructing support vector machine ensemble, Pattern Recognition, 36 (2003), 2757-2767.

[58] B. Schölkopf, C. Burges, and V. Vapnik, Extracting support data for a given task, Proceedings of the First International Conference on Knowledge Discovery and Data Mining, Montreal, Canada, (1995).

[59] A. Brooke, D. Kendrick, A. Meeraus, and R. Raman, GAMS: A User's Guide, GAMS Development Co., Washington, DC, (1998).

[60] Ilog, (2003). ILOG CPLEX 9.0 User's Manual.

APPENDIX A: Composition Values of Training and Test Set Proteins

Table A.1. Compositions of training proteins belong to All-α for the first 10 amino acids.

PROT	CHAIN	M	K	T	A	Y	R	S	F	V	H
1AVH	A	0.0250	0.0688	0.0719	0.0813	0.0375	0.0594	0.0656	0.0406	0.0500	0.0094
1BAB	B	0.0068	0.0753	0.0479	0.1027	0.0205	0.0205	0.0342	0.0548	0.1233	0.0616
1C5A	A	0.0274	0.1507	0.0137	0.1233	0.0685	0.0548	0.0137	0.0137	0.0274	0.0137
1CPC	A	0.0123	0.0494	0.0741	0.1481	0.0617	0.0370	0.0988	0.0370	0.0494	0.0062
1CPC	L	0.0349	0.0349	0.0465	0.1802	0.0349	0.0581	0.0814	0.0233	0.0756	0.0000
1ECO	A	0.0294	0.0735	0.0662	0.1250	0.0147	0.0221	0.0662	0.1029	0.0662	0.0294
1FCS	A	0.0131	0.1242	0.0261	0.1111	0.0196	0.0327	0.0392	0.0392	0.0588	0.0719
1FHA	A	0.0273	0.0656	0.0383	0.0710	0.0492	0.0383	0.0656	0.0328	0.0328	0.0546
1FIA	A	0.0612	0.0714	0.0612	0.0510	0.0408	0.0612	0.0408	0.0204	0.1020	0.0000
1HBB	A	0.0142	0.0780	0.0638	0.1489	0.0213	0.0213	0.0780	0.0496	0.0922	0.0709
1HBG	A	0.0340	0.0816	0.0068	0.2041	0.0204	0.0204	0.0680	0.0272	0.0680	0.0340
1HIG	A	0.0290	0.1449	0.0362	0.0507	0.0290	0.0435	0.0725	0.0725	0.0580	0.0145
1LE4	A	0.0278	0.0486	0.0417	0.1042	0.0278	0.1181	0.0486	0.0069	0.0625	0.0069
1LTS	C	0.0000	0.0488	0.0732	0.0000	0.0976	0.0732	0.0976	0.0244	0.0488	0.0000
1MBC	A	0.0131	0.1242	0.0327	0.1111	0.0196	0.0261	0.0392	0.0392	0.0523	0.0784
1MBS	A	0.0131	0.1242	0.0327	0.0915	0.0131	0.0327	0.0458	0.0458	0.0392	0.0850
1POC	A	0.0224	0.0896	0.0821	0.0299	0.0597	0.0448	0.0746	0.0373	0.0373	0.0522
1RPR	A	0.0317	0.0476	0.0635	0.0952	0.0159	0.0635	0.0476	0.0317	0.0000	0.0317
1TRO	A	0.0370	0.0370	0.0370	0.0926	0.0185	0.0833	0.0556	0.0093	0.0463	0.0185
256B	B	0.0283	0.1226	0.0472	0.1604	0.0189	0.0377	0.0189	0.0189	0.0377	0.0189
2CCY	A	0.0234	0.1172	0.0469	0.2188	0.0000	0.0156	0.0391	0.0313	0.0234	0.0078
2LH1	A	0.0065	0.0915	0.0523	0.1373	0.0131	0.0065	0.0588	0.0458	0.1111	0.0327
2LHB	A	0.0336	0.0872	0.0671	0.1409	0.0268	0.0336	0.0872	0.0537	0.0805	0.0134
2LIG	A	0.0549	0.0244	0.0549	0.1402	0.0427	0.0488	0.0793	0.0366	0.0183	0.0183
2MHB	A	0.0071	0.0780	0.0638	0.1135	0.0213	0.0213	0.0922	0.0496	0.0851	0.0709
2MHB	B	0.0068	0.0753	0.0205	0.0959	0.0205	0.0274	0.0411	0.0548	0.1233	0.0616
2ZTA	A	0.0303	0.1515	0.0000	0.0303	0.0303	0.0909	0.0303	0.0000	0.0909	0.0303
3HDD	A	0.0000	0.1167	0.0500	0.0667	0.0167	0.1500	0.0833	0.0500	0.0000	0.0000
4MBA	A	0.0205	0.0753	0.0137	0.1986	0.0000	0.0274	0.0890	0.1027	0.0685	0.0068
4MBN	A	0.0131	0.1242	0.0327	0.1111	0.0196	0.0261	0.0392	0.0392	0.0523	0.0784

Table A.2. Compositions of training proteins belong to All-α for the last 10 amino acids.

PROT	CHAIN	L	D	E	G	P	N	W	Q	I	C
1AVH	A	0.11875	0.078125	0.090625	0.06875	0.01563	0.01875	0.00313	0.0375	0.05625	0.003125
1BAB	B	0.12329	0.047945	0.054795	0.089041	0.04795	0.0411	0.0137	0.02055	0	0.013699
1C5A	A	0.0411	0.054795	0.09589	0.041096	0.0137	0.0411	0	0.05479	0.068493	0.082192
1CPC	A	0.08025	0.049383	0.037037	0.080247	0.03704	0.04321	0.00617	0.03086	0.049383	0.012346
1CPC	L	0.08721	0.075581	0.02907	0.081395	0.01744	0.0407	0	0.02907	0.052326	0.017442
1ECO	A	0.04412	0.066176	0.036765	0.080882	0.03676	0.03676	0.00735	0.02941	0.066176	0
1FCS	A	0.11765	0.039216	0.091503	0.071895	0.02614	0.01307	0.01307	0.03268	0.058824	0
1FHA	A	0.12022	0.081967	0.087432	0.038251	0.01639	0.06557	0.00546	0.06011	0.032787	0.016393
1FIA	A	0.12245	0.061224	0.040816	0.05102	0.02041	0.08163	0	0.10204	0.010204	0
1HBB	A	0.12766	0.056738	0.028369	0.049645	0.04965	0.02837	0.00709	0.00709	0	0.007092
1HBG	A	0.07483	0.040816	0.020408	0.136054	0.02041	0.02041	0.01361	0.04762	0.054422	0.006803
1HIG	A	0.07246	0.072464	0.065217	0.036232	0.01449	0.07246	0.00725	0.05797	0.050725	0
1LE4	A	0.16667	0.048611	0.118056	0.048611	0.00694	0	0.02083	0.09722	0	0
1LTS	C	0.04878	0.073171	0.097561	0.02439	0	0.07317	0	0.09756	0.097561	0.02439
1MBC	A	0.11765	0.045752	0.091503	0.071895	0.02614	0.00654	0.01307	0.03268	0.058824	0
1MBS	A	0.12418	0.052288	0.091503	0.078431	0.02614	0.01961	0.01307	0.01961	0.052288	0
1POC	A	0.06716	0.08209	0.037313	0.08209	0.03731	0.03731	0.01493	0.00746	0.029851	0.074627
1RPR	A	0.15873	0.111111	0.111111	0.031746	0	0.04762	0	0.04762	0.031746	0.031746
1TRO	A	0.17593	0.037037	0.12037	0.046296	0.03704	0.0463	0.01852	0.05556	0.027778	0
256B	B	0.09434	0.113208	0.075472	0.028302	0.03774	0.0566	0	0.0566	0.028302	0
2CCY	A	0.09375	0.039063	0.078125	0.078125	0.05469	0.01563	0.02344	0.0625	0.015625	0.015625
2LH1	A	0.0915	0.039216	0.091503	0.045752	0.03268	0.03922	0.01961	0.02614	0.058824	0
2LHB	A	0.06711	0.073826	0.053691	0.040268	0.04027	0.01342	0.01342	0.01342	0.053691	0.006711
2LIG	A	0.10976	0.04878	0.042683	0.042683	0.01829	0.07317	0.0061	0.11585	0.018293	0.006098
2MHB	A	0.14894	0.06383	0.021277	0.070922	0.04255	0.02837	0.00709	0.00709	0	0.007092
2MHB	B	0.13014	0.047945	0.068493	0.09589	0.03425	0.04795	0.0137	0.0274	0	0.006849
2ZTA	A	0.18182	0.030303	0.181818	0.030303	0	0.06061	0	0.0303	0	0
3HDD	A	0.1	0	0.116667	0.016667	0.01667	0.06667	0.01667	0.08333	0.05	0
4MBA	A	0.07534	0.054795	0.034247	0.075342	0.0411	0.06164	0.0137	0.0137	0.027397	0
4MBN	A	0.11765	0.045752	0.091503	0.071895	0.02614	0.00654	0.01307	0.03268	0.058824	0

Table A.3. Compositions of training proteins belong to All-β for the first 10 amino acids.

PROT	CHAIN	M	K	T	A	Y	R	S	F	V	H
1ACX	A	0.0000	0.0093	0.0926	0.1852	0.0278	0.0093	0.1389	0.0463	0.0833	0.0093
1CD8	A	0.0088	0.0351	0.0614	0.0614	0.0351	0.0614	0.1140	0.0877	0.0526	0.0088
1CDT	A	0.0500	0.1667	0.0333	0.0500	0.0500	0.0333	0.0500	0.0000	0.0833	0.0000
1CID	A	0.0226	0.0734	0.0678	0.0452	0.0113	0.0282	0.1130	0.0282	0.0791	0.0000
1DFN	A	0.0000	0.0000	0.0333	0.1000	0.1000	0.1333	0.0000	0.0333	0.0000	0.0000
1HIL	A	0.0138	0.0645	0.1106	0.0415	0.0461	0.0276	0.1429	0.0369	0.0599	0.0092
1HIV	A	0.0202	0.0606	0.0808	0.0303	0.0101	0.0404	0.0101	0.0202	0.0707	0.0101
1MAM	H	0.0138	0.0369	0.1198	0.0645	0.0553	0.0323	0.1429	0.0276	0.0829	0.0092
1PAZ	A	0.0407	0.1057	0.0407	0.1057	0.0325	0.0081	0.0407	0.0244	0.0976	0.0244
1REI	A	0.0093	0.0374	0.1028	0.0561	0.0748	0.0187	0.1308	0.0280	0.0280	0.0000
1TEN	A	0.0111	0.0556	0.1333	0.0444	0.0333	0.0444	0.0667	0.0222	0.0444	0.0000
1TFG	A	0.0089	0.0625	0.0357	0.0714	0.0625	0.0357	0.0893	0.0268	0.0357	0.0179
1TLK	A	0.0260	0.0714	0.0519	0.0649	0.0260	0.0195	0.0714	0.0325	0.0714	0.0130
2ALP	A	0.0101	0.0101	0.0909	0.1212	0.0202	0.0606	0.1010	0.0303	0.0960	0.0051
2AVI	A	0.0156	0.0703	0.1641	0.0391	0.0078	0.0625	0.0703	0.0547	0.0547	0.0078
2AYH	A	0.0140	0.0654	0.0888	0.0561	0.0794	0.0093	0.0701	0.0607	0.0514	0.0187
2CTX	A	0.0000	0.0704	0.1268	0.0423	0.0141	0.0704	0.0423	0.0423	0.0563	0.0141
2ILA	A	0.0194	0.0710	0.0774	0.0903	0.0452	0.0194	0.0581	0.0581	0.0452	0.0194
2LAL	A	0.0000	0.0608	0.1326	0.0718	0.0387	0.0221	0.0773	0.0718	0.0884	0.0110
2OMF	A	0.0088	0.0529	0.0618	0.0853	0.0853	0.0324	0.0471	0.0559	0.0676	0.0029
2RAC	A	0.0476	0.0762	0.0762	0.1238	0.0381	0.0190	0.0286	0.0381	0.1143	0.0476
2SNV	A	0.0331	0.0662	0.0795	0.0596	0.0265	0.0530	0.0662	0.0397	0.0861	0.0397
2VAA	B	0.0505	0.0909	0.0808	0.0303	0.0505	0.0303	0.0606	0.0404	0.0505	0.0505
3CD4	A	0.0000	0.1044	0.0714	0.0275	0.0055	0.0275	0.0989	0.0330	0.0659	0.0110
3HRR	C	0.0148	0.0640	0.0788	0.0246	0.0493	0.0443	0.0739	0.0345	0.0788	0.0197
4GCR	A	0.0402	0.0115	0.0287	0.0115	0.0862	0.1149	0.0747	0.0517	0.0345	0.0287
7API	B	0.0556	0.1389	0.0556	0.0000	0.0000	0.0000	0.0556	0.1111	0.1111	0.0000
8FAB	A	0.0047	0.0519	0.1038	0.0943	0.0519	0.0236	0.1368	0.0189	0.0708	0.0094
8FAB	B	0.0134	0.0536	0.0804	0.0625	0.0491	0.0402	0.1384	0.0357	0.1071	0.0134
8I1B	A	0.0263	0.0987	0.0263	0.0263	0.0329	0.0263	0.0987	0.0526	0.0789	0.0197

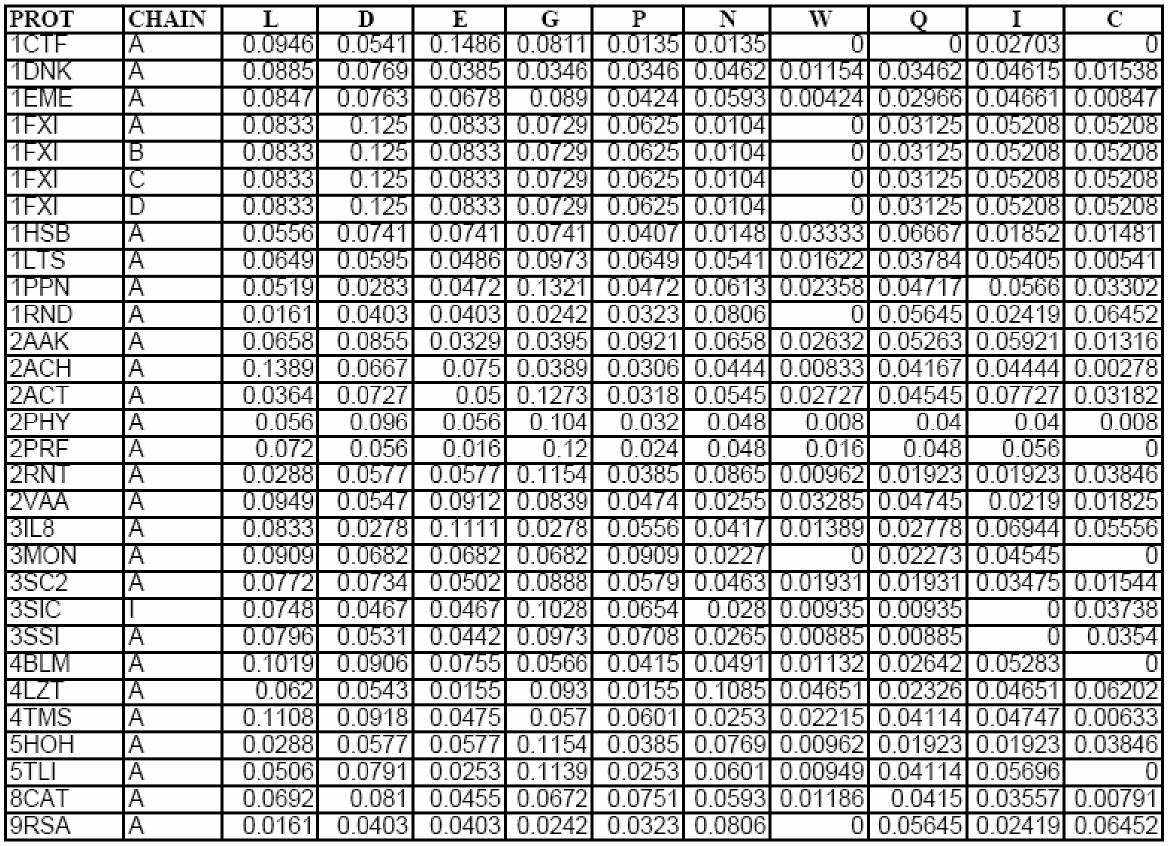
Table A.4. Compositions of training proteins belong to All- β for the last 10 amino acids.

PROT	CHAIN	L	D	E	G	P	N	W	Q	I	C
1ACX	A	0.0370	0.0463	0.0093	0.1296	0.0556	0.0370	0.0000	0.0370	0.0093	0.0370
1CD8	A	0.1316	0.0351	0.0439	0.0614	0.0614	0.0439	0.0175	0.0439	0.0088	0.0263
1CDT	A	0.0833	0.0167	0.0167	0.0333	0.0667	0.0833	0.0000	0.0000	0.0500	0.1333
1CID	A	0.1130	0.0169	0.0960	0.0508	0.0508	0.0395	0.0169	0.0960	0.0395	0.0113
1DFN	A	0.0333	0.0333	0.0333	0.1000	0.0333	0.0000	0.0333	0.0333	0.1000	0.2000
1HIL	A	0.0645	0.0553	0.0415	0.0645	0.0507	0.0507	0.0184	0.0507	0.0323	0.0184
1HIV	A	0.1212	0.0404	0.0404	0.1313	0.0606	0.0303	0.0202	0.0606	0.1212	0.0202
1MAM	H	0.0829	0.0323	0.0323	0.0922	0.0645	0.0184	0.0230	0.0323	0.0138	0.0230
1PAZ	A	0.0488	0.0407	0.0813	0.0650	0.0650	0.0569	0.0000	0.0244	0.0894	0.0081
1REI	A	0.0748	0.0467	0.0187	0.0748	0.0561	0.0187	0.0093	0.1215	0.0748	0.0187
1TEN	A	0.0778	0.1111	0.0889	0.0556	0.0556	0.0333	0.0111	0.0222	0.0889	0.0000
1TFG	A	0.0982	0.0536	0.0357	0.0357	0.0625	0.0625	0.0268	0.0357	0.0625	0.0804
1TLK	A	0.0390	0.0844	0.1818	0.0844	0.0455	0.0390	0.0065	0.0065	0.0325	0.0325
2ALP	A	0.0505	0.0101	0.0202	0.1616	0.0202	0.0657	0.0101	0.0455	0.0404	0.0303
2AVI	A	0.0547	0.0391	0.0547	0.0859	0.0156	0.0781	0.0313	0.0234	0.0547	0.0156
2AYH	A	0.0467	0.0561	0.0374	0.1215	0.0327	0.0888	0.0374	0.0187	0.0374	0.0093
2CTX	A	0.0141	0.0845	0.0000	0.0563	0.0845	0.0423	0.0141	0.0141	0.0704	0.1408
2ILA	A	0.0968	0.0581	0.0516	0.0323	0.0387	0.0645	0.0129	0.0516	0.0839	0.0065
2LAL	A	0.0442	0.0608	0.0276	0.0663	0.0387	0.0829	0.0166	0.0331	0.0552	0.0000
2OMF	A	0.0588	0.0794	0.0412	0.1412	0.0118	0.0882	0.0059	0.0382	0.0353	0.0000
2RAC	A	0.0381	0.0476	0.0857	0.0667	0.0667	0.0190	0.0095	0.0095	0.0381	0.0095
2SNV	A	0.0530	0.0530	0.0662	0.1258	0.0464	0.0331	0.0199	0.0132	0.0397	0.0000
2VAA	B	0.0303	0.0606	0.0606	0.0202	0.0909	0.0404	0.0202	0.0505	0.0707	0.0202
3CD4	A	0.1154	0.0549	0.0495	0.0714	0.0330	0.0495	0.0165	0.0879	0.0549	0.0220
3HRR	C	0.0640	0.0443	0.0936	0.0443	0.0591	0.0542	0.0394	0.0394	0.0493	0.0296
4GCR	A	0.0747	0.0747	0.0517	0.0805	0.0460	0.0345	0.0230	0.0575	0.0345	0.0402
7API	B	0.0556	0.0000	0.0556	0.0278	0.1389	0.0833	0.0000	0.0556	0.0556	0.0000
8FAB	A	0.0519	0.0283	0.0472	0.0566	0.0802	0.0283	0.0189	0.0660	0.0330	0.0236
8FAB	B	0.0714	0.0357	0.0179	0.0938	0.0536	0.0357	0.0223	0.0313	0.0223	0.0223
8I1B	A	0.0921	0.0461	0.0724	0.0461	0.0592	0.0592	0.0066	0.0724	0.0526	0.0066

Table A.5. Compositions of training proteins belong to α+β for the first 10 amino acids.

PROT	CHAIN	M	K	T	A	Y	R	S	F	V	H
1CTF	A	0	0.1486	0.027	0.2297	0	0.0135	0.02703	0.01351	0.10811	0
1DNK	A	0.0154	0.0346	0.0577	0.0846	0.0577	0.0462	0.11538	0.04231	0.09615	0.02308
1EME	A	0.0254	0.0805	0.0678	0.0381	0.0424	0.0297	0.04237	0.05085	0.07203	0.04237
1FXI	A	0	0.0417	0.0729	0.0938	0.0625	0.0104	0.05208	0.01042	0.07292	0.01042
1FXI	B	0	0.0417	0.0729	0.0938	0.0625	0.0104	0.05208	0.01042	0.07292	0.01042
1FXI	C	0	0.0417	0.0729	0.0938	0.0625	0.0104	0.05208	0.01042	0.07292	0.01042
1FXI	D	0	0.0417	0.0729	0.0938	0.0625	0.0104	0.05208	0.01042	0.07292	0.01042
1HSB	A	0.0222	0.0333	0.0815	0.0852	0.0519	0.0852	0.05185	0.02593	0.05926	0.03704
1LTS	A	0.0162	0.0054	0.0324	0.0649	0.0973	0.0973	0.06486	0.02703	0.04865	0.04324
1PPN	A	0	0.0472	0.0377	0.066	0.0896	0.0566	0.06132	0.01887	0.08491	0.00943
1RND	A	0.0323	0.0806	0.0806	0.0968	0.0484	0.0323	0.12097	0.02419	0.07258	0.03226
2AAK	A	0.0329	0.0329	0.0395	0.0724	0.0329	0.0724	0.09211	0.03947	0.04605	0.00658
2ACH	A	0.0278	0.0667	0.0667	0.0778	0.025	0.0333	0.07778	0.05556	0.05278	0.025
2ACT	A	0.0091	0.0273	0.0818	0.0818	0.0636	0.0227	0.05455	0.02273	0.07727	0.00455
2PHY	A	0.04	0.088	0.056	0.072	0.04	0.016	0.04	0.072	0.072	0.016
2PRF	A	0	0.048	0.096	0.136	0.04	0.024	0.064	0.032	0.096	0.008
2RNT	A	0	0.0192	0.0577	0.0673	0.0865	0.0096	0.14423	0.03846	0.07692	0.02885
2VAA	A	0.0146	0.0401	0.0693	0.0693	0.062	0.0766	0.04015	0.0219	0.05839	0.0292
3IL8	A	0	0.125	0.0278	0.0417	0.0139	0.0694	0.06944	0.04167	0.06944	0.02778
3MON	A	0	0.0909	0.0227	0.0455	0.1136	0.1136	0.04545	0.04545	0.04545	0
3SC2	A	0.0116	0.027	0.0386	0.1004	0.0695	0.0425	0.07336	0.05019	0.0695	0.03475
3SIC	I	0.0187	0.028	0.0748	0.1495	0.028	0.0374	0.07477	0.02804	0.1215	0.01869
3SSI	A	0.0265	0.0177	0.0708	0.1593	0.0265	0.0354	0.07965	0.02655	0.11504	0.0177
4BLM	A	0.0189	0.0906	0.0792	0.0981	0.0226	0.0566	0.04151	0.02642	0.0566	0.00377
4LZT	A	0.0155	0.0465	0.0543	0.093	0.0233	0.0853	0.07752	0.02326	0.04651	0.00775
4TMS	A	0.019	0.0633	0.0506	0.0633	0.0443	0.038	0.04114	0.06013	0.05063	0.06013
5HOH	A	0	0.0192	0.0481	0.0865	0.0865	0.0096	0.14423	0.03846	0.07692	0.02885
5TLI	A	0.0063	0.0348	0.0791	0.0886	0.0886	0.0316	0.08228	0.03165	0.06962	0.02532
8CAT	A	0.0198	0.0494	0.0435	0.0711	0.0395	0.0632	0.04941	0.06126	0.06719	0.0415
9RSA	A	0.0323	0.0806	0.0806	0.0968	0.0484	0.0323	0.12097	0.02419	0.07258	0.03226

Table A.6. Compositions of training proteins belong to α+β for the last 10 amino acids.



PROT	CHAIN	L	D	E	G	P	N	W	Q	I	C
1CTF	A	0.0946	0.0541	0.1486	0.0811	0.0135	0.0135	0	0	0.02703	0
1DNK	A	0.0885	0.0769	0.0385	0.0346	0.0346	0.0462	0.01154	0.03462	0.04615	0.01538
1EME	A	0.0847	0.0763	0.0678	0.089	0.0424	0.0593	0.00424	0.02966	0.04661	0.00847
1FXI	A	0.0833	0.125	0.0833	0.0729	0.0625	0.0104	0	0.03125	0.05208	0.05208
1FXI	B	0.0833	0.125	0.0833	0.0729	0.0625	0.0104	0	0.03125	0.05208	0.05208
1FXI	C	0.0833	0.125	0.0833	0.0729	0.0625	0.0104	0	0.03125	0.05208	0.05208
1FXI	D	0.0833	0.125	0.0833	0.0729	0.0625	0.0104	0	0.03125	0.05208	0.05208
1HSB	A	0.0556	0.0741	0.0741	0.0741	0.0407	0.0148	0.03333	0.06667	0.01852	0.01481
1LTS	A	0.0649	0.0595	0.0486	0.0973	0.0649	0.0541	0.01622	0.03784	0.05405	0.00541
1PPN	A	0.0519	0.0283	0.0472	0.1321	0.0472	0.0613	0.02358	0.04717	0.0566	0.03302
1RND	A	0.0161	0.0403	0.0403	0.0242	0.0323	0.0806	0	0.05645	0.02419	0.06452
2AAK	A	0.0658	0.0855	0.0329	0.0395	0.0921	0.0658	0.02632	0.05263	0.05921	0.01316
2ACH	A	0.1389	0.0667	0.075	0.0389	0.0306	0.0444	0.00833	0.04167	0.04444	0.00278
2ACT	A	0.0364	0.0727	0.05	0.1273	0.0318	0.0545	0.02727	0.04545	0.07727	0.03182
2PHY	A	0.056	0.096	0.056	0.104	0.032	0.048	0.008	0.04	0.04	0.008
2PRF	A	0.072	0.056	0.016	0.12	0.024	0.048	0.016	0.048	0.056	0
2RNT	A	0.0288	0.0577	0.0577	0.1154	0.0385	0.0865	0.00962	0.01923	0.01923	0.03846
2VAA	A	0.0949	0.0547	0.0912	0.0839	0.0474	0.0255	0.03285	0.04745	0.0219	0.01825
3IL8	A	0.0833	0.0278	0.1111	0.0278	0.0556	0.0417	0.01389	0.02778	0.06944	0.05556
3MON	A	0.0909	0.0682	0.0682	0.0682	0.0909	0.0227	0	0.02273	0.04545	0
3SC2	A	0.0772	0.0734	0.0502	0.0888	0.0579	0.0463	0.01931	0.01931	0.03475	0.01544
3SIC	I	0.0748	0.0467	0.0467	0.1028	0.0654	0.028	0.00935	0.00935	0	0.03738
3SSI	A	0.0796	0.0531	0.0442	0.0973	0.0708	0.0265	0.00885	0.00885	0	0.0354
4BLM	A	0.1019	0.0906	0.0755	0.0566	0.0415	0.0491	0.01132	0.02642	0.05283	0
4LZT	A	0.062	0.0543	0.0155	0.093	0.0155	0.1085	0.04651	0.02326	0.04651	0.06202
4TMS	A	0.1108	0.0918	0.0475	0.057	0.0601	0.0253	0.02215	0.04114	0.04747	0.00633
5HOH	A	0.0288	0.0577	0.0577	0.1154	0.0385	0.0769	0.00962	0.01923	0.01923	0.03846
5TLI	A	0.0506	0.0791	0.0253	0.1139	0.0253	0.0601	0.00949	0.04114	0.05696	0
8CAT	A	0.0692	0.081	0.0455	0.0672	0.0751	0.0593	0.01186	0.0415	0.03557	0.00791
9RSA	A	0.0161	0.0403	0.0403	0.0242	0.0323	0.0806	0	0.05645	0.02419	0.06452

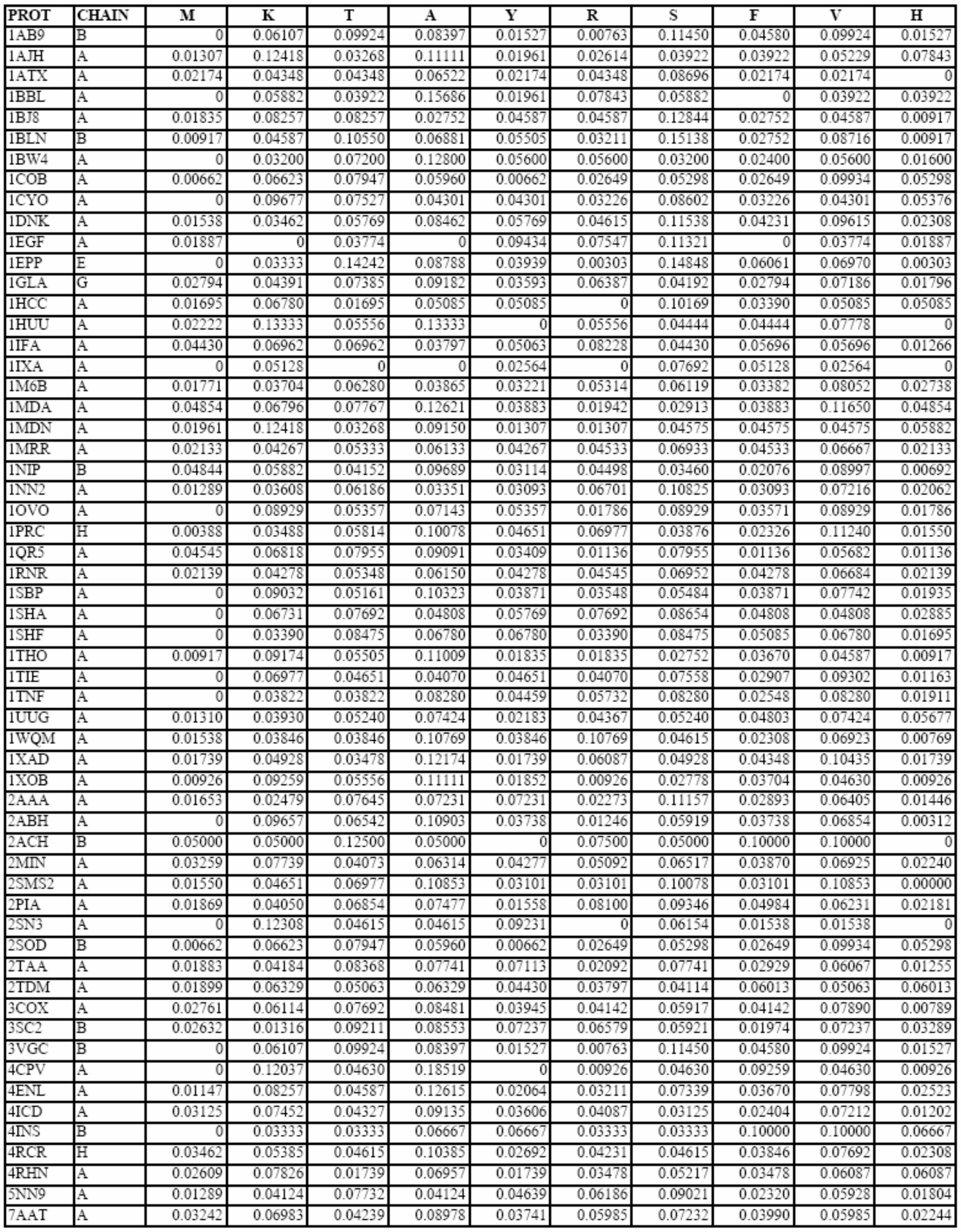
Table A.7. Compositions of training proteins belong to α/β for the first 10 amino acids.

PROT	CHAIN	M	K	T	A	Y	R	S	F	V	H
1ABA	A	0.03448	0.10345	0.04598	0.034483	0.0345	0.0345	0.02299	0.08046	0.04598	0.022989
1BKS	B	0.03778	0.04786	0.0529	0.108312	0.0302	0.0479	0.04786	0.03275	0.04786	0.035264
1CIS	A	0.01515	0.10606	0.01515	0.106061	0.0303	0.0606	0.01515	0	0.13636	0
1DBP	A	0.01476	0.08856	0.04797	0.136531	0.0111	0.0221	0.03321	0.02583	0.10701	0.01107
1DHR	A	0.0332	0.06224	0.07054	0.120332	0.0124	0.0332	0.08299	0.02905	0.07469	0.016598
1EAF	A	0.02058	0.06584	0.04115	0.135802	0.0165	0.0453	0.0535	0.03704	0.07407	0.020576
1ETU	A	0.02646	0.0582	0.07937	0.063492	0.0265	0.0556	0.02646	0.03439	0.09788	0.029101
1GPB	A	0.02494	0.05701	0.04276	0.074822	0.0428	0.0748	0.03444	0.04513	0.07363	0.026128
1KKJ	A	0.02387	0.04535	0.05489	0.124105	0.0358	0.0525	0.0358	0.04057	0.08592	0.0358
1OFV	A	0.00592	0.04734	0.04734	0.065089	0.0473	0.0118	0.07101	0.04734	0.05917	0
1OVB	A	0.00629	0.08805	0.0566	0.09434	0.0377	0.044	0.06918	0.03145	0.06918	0.018868
1PFK	A	0.0375	0.04375	0.04688	0.084375	0.0344	0.0688	0.04375	0.03125	0.07813	0.021875
1Q21	A	0.02339	0.05848	0.06433	0.064327	0.0526	0.0702	0.04678	0.02924	0.08772	0.017544
1S01	A	0.01455	0.04364	0.04727	0.138182	0.0327	0.0073	0.13455	0.01455	0.10909	0.021818
1SBP	A	0	0.09032	0.05161	0.103226	0.0387	0.0355	0.05484	0.03871	0.07742	0.019355
1SBT	A	0.01818	0.04	0.04727	0.134545	0.0364	0.0073	0.13455	0.01091	0.10909	0.021818
1TIM	A	0.0081	0.08907	0.04049	0.11336	0.0162	0.0324	0.04858	0.03239	0.09312	0.032389
1TRE	A	0.02745	0.06275	0.03529	0.176471	0.0196	0.0314	0.04314	0.02353	0.07451	0.031373
1ULA	A	0.04152	0.04152	0.0519	0.076125	0.0311	0.0588	0.04844	0.0519	0.07958	0.027682
2CTC	A	0.00977	0.04886	0.08469	0.065147	0.0619	0.0358	0.10423	0.05212	0.04886	0.026059
2FOX	A	0.03623	0.07246	0.03623	0.043478	0.0217	0.0145	0.05797	0.03623	0.07246	0
2HAD	A	0.03548	0.03871	0.05161	0.096774	0.0355	0.0484	0.04194	0.07419	0.04516	0.016129
2LIV	A	0.01453	0.0843	0.05233	0.127907	0.0378	0.0203	0.0407	0.02907	0.07558	0.011628
2PGD	A	0.02697	0.07261	0.03527	0.091286	0.027	0.0415	0.05602	0.04772	0.04979	0.024896
2TMD	A	0.01646	0.05761	0.05075	0.090535	0.0508	0.059	0.05624	0.03018	0.06173	0.028807
3GBP	A	0.0228	0.09772	0.04235	0.130293	0.0261	0.0195	0.0456	0.01954	0.08795	0.009772
5CPA	A	0.00977	0.04886	0.08469	0.068404	0.0619	0.0358	0.10423	0.05212	0.05212	0.026059
5P21	A	0.0241	0.04819	0.06627	0.066265	0.0542	0.0663	0.04819	0.03012	0.09036	0.018072
8ABP	A	0.02941	0.09804	0.05229	0.101307	0.0196	0.0261	0.04575	0.03922	0.08497	0.009804
8ATC	A	0.02581	0.04839	0.05806	0.109677	0.0258	0.0484	0.06452	0.03871	0.07097	0.035484

Table A.8. Compositions of training proteins belong to α/β for the last 10 amino acids.

PROT	CHAIN	L	D	E	G	P	N	W	Q	I	C
1ABA	A	0.08046	0.08046	0.05747	0.091954	0.0575	0.0345	0	0.04598	0.06897	0.022989
1BKS	B	0.09572	0.04534	0.07053	0.108312	0.0453	0.0277	0.00252	0.04282	0.06045	0.012594
1CIS	A	0.07576	0.07576	0.10606	0.030303	0.0455	0.0303	0.01515	0.06061	0.07576	0
1DBP	A	0.09225	0.08118	0.04059	0.084871	0.0332	0.059	0	0.06273	0.04797	0
1DHR	A	0.08714	0.04564	0.04564	0.103734	0.0332	0.0332	0.02905	0.0332	0.03734	0.016598
1EAF	A	0.13169	0.04527	0.04527	0.061728	0.0658	0.0247	0.00823	0.03704	0.06173	0.00823
1ETU	A	0.07407	0.0582	0.09259	0.108466	0.0503	0.0159	0.00265	0.01852	0.07407	0.007937
1GPB	A	0.09264	0.05819	0.0772	0.057007	0.0428	0.0546	0.01425	0.03563	0.06057	0.010689
1KKJ	A	0.07637	0.04773	0.0716	0.085919	0.0453	0.0334	0	0.04773	0.05251	0.004773
1OFV	A	0.07101	0.09467	0.06509	0.106509	0.0178	0.0651	0.02367	0.07101	0.07692	0.005917
1OVB	A	0.0566	0.0566	0.06918	0.09434	0.0314	0.0252	0.01887	0.03774	0.04403	0.050314
1PFK	A	0.07813	0.07188	0.06563	0.11875	0.025	0.025	0.00313	0.01563	0.0875	0.01875
1Q21	A	0.07602	0.08187	0.07602	0.064327	0.0175	0.0234	0	0.06433	0.06433	0.017544
1S01	A	0.05455	0.04364	0.01455	0.12	0.0509	0.0545	0.01091	0.03636	0.04727	0.003636
1SBP	A	0.06774	0.08387	0.06129	0.067742	0.0387	0.0581	0.02258	0.03226	0.05806	0
1SBT	A	0.05455	0.04	0.01455	0.12	0.0509	0.0618	0.01091	0.04	0.04727	0
1TIM	A	0.06883	0.05263	0.06883	0.109312	0.0283	0.0243	0.02024	0.03644	0.06883	0.016194
1TRE	A	0.06667	0.03529	0.08235	0.086275	0.0275	0.0392	0.00784	0.04314	0.07451	0.011765
1ULA	A	0.08997	0.04498	0.06228	0.093426	0.0519	0.0381	0.01038	0.04498	0.04152	0.013841
2CTC	A	0.07818	0.05212	0.0456	0.074919	0.0326	0.0423	0.0228	0.03909	0.0684	0.006515
2FOX	A	0.05797	0.06522	0.13768	0.101449	0.0217	0.058	0.02174	0.01449	0.1087	0.021739
2HAD	A	0.09032	0.08387	0.05806	0.058065	0.0742	0.0323	0.01935	0.04194	0.04516	0.012903
2LIV	A	0.06686	0.07558	0.04942	0.098837	0.0436	0.0407	0.00872	0.05814	0.05814	0.005814
2PGD	A	0.09544	0.06432	0.04772	0.107884	0.0311	0.0373	0.0166	0.03527	0.07261	0.018672
2TMD	A	0.0631	0.06859	0.0631	0.089163	0.0494	0.0316	0.02058	0.04115	0.05213	0.019204
3GBP	A	0.07492	0.09446	0.04235	0.071661	0.0326	0.0684	0.01629	0.04886	0.04886	0
5CPA	A	0.07492	0.03909	0.0456	0.074919	0.0326	0.0554	0.0228	0.03583	0.0684	0.006515
5P21	A	0.06627	0.08434	0.07831	0.066265	0.0181	0.0241	0	0.06627	0.06627	0.018072
8ABP	A	0.0719	0.06863	0.06863	0.094771	0.049	0.0327	0.01634	0.03595	0.05229	0.003268
8ATC	A	0.12258	0.06774	0.04516	0.048387	0.0387	0.0484	0.00645	0.04516	0.04839	0.003226

Table A.9. Composition values of test data set proteins' for the first 10 amino acids.



PROT	CHAIN	M	K	T	A	Y	R	S	F	V	H
1AB9	B	0	0.06107	0.09924	0.08397	0.01527	0.00763	0.11450	0.04580	0.09924	0.01527
1AJH	A	0.01307	0.12418	0.03268	0.11111	0.01961	0.02614	0.03922	0.03922	0.05229	0.07843
1ATX	A	0.02174	0.04348	0.04348	0.06522	0.02174	0.04348	0.08696	0.02174	0.02174	0
1BBL	A	0	0.05882	0.03922	0.15686	0.01961	0.07843	0.05882	0	0.03922	0.03922
1BJ8	A	0.01835	0.08257	0.08257	0.02752	0.04587	0.04587	0.12844	0.02752	0.04587	0.00917
1BLN	B	0.00917	0.04587	0.10550	0.06881	0.05505	0.03211	0.15138	0.02752	0.08716	0.00917
1BW4	A	0	0.03200	0.07200	0.12800	0.05600	0.05600	0.03200	0.02400	0.05600	0.01600
1COB	A	0.00662	0.06623	0.07947	0.05960	0.00662	0.02649	0.05298	0.02649	0.09934	0.05298
1CYO	A	0	0.09677	0.07527	0.04301	0.04301	0.03226	0.08602	0.03226	0.04301	0.05376
1DNK	A	0.01538	0.03462	0.05769	0.08462	0.05769	0.04615	0.11538	0.04231	0.09615	0.02308
1EGF	A	0.01887	0	0.03774	0	0.09434	0.07547	0.11321	0	0.03774	0.01887
1EPP	E	0	0.03333	0.14242	0.08788	0.03939	0.00303	0.14848	0.06061	0.06970	0.00303
1GLA	G	0.02794	0.04391	0.07385	0.09182	0.03593	0.06387	0.04192	0.02794	0.07186	0.01796
1HCC	A	0.01695	0.06780	0.01695	0.05085	0.05085	0	0.10169	0.03390	0.05085	0.05085
1HUU	A	0.02222	0.13333	0.05556	0.13333	0	0.05556	0.04444	0.04444	0.07778	0
1IFA	A	0.04430	0.06962	0.06962	0.03797	0.05063	0.08228	0.04430	0.05696	0.05696	0.01266
1IXA	A	0	0.05128	0	0	0.02564	0	0.07692	0.05128	0.02564	0
1M6B	A	0.01771	0.03704	0.06280	0.03865	0.03221	0.05314	0.06119	0.03382	0.08052	0.02738
1MDA	A	0.04854	0.06796	0.07767	0.12621	0.03883	0.01942	0.02913	0.03883	0.11650	0.04854
1MDN	A	0.01961	0.12418	0.03268	0.09150	0.01307	0.01307	0.04575	0.04575	0.04575	0.05882
1MRR	A	0.02133	0.04267	0.05333	0.06133	0.04267	0.04533	0.06933	0.04533	0.06667	0.02133
1NIP	B	0.04844	0.05882	0.04152	0.09689	0.03114	0.04498	0.03460	0.02076	0.08997	0.00692
1NN2	A	0.01289	0.03608	0.06186	0.03351	0.03093	0.06701	0.10825	0.03093	0.07216	0.02062
1OVO	A	0	0.08929	0.05357	0.07143	0.05357	0.01786	0.08929	0.03571	0.08929	0.01786
1PRC	H	0.00388	0.03488	0.05814	0.10078	0.04651	0.06977	0.03876	0.02326	0.11240	0.01550
1QR5	A	0.04545	0.06818	0.07955	0.09091	0.03409	0.01136	0.07955	0.01136	0.05682	0.01136
1RNR	A	0.02139	0.04278	0.05348	0.06150	0.04278	0.04545	0.06952	0.04278	0.06684	0.02139
1SBP	A	0	0.09032	0.05161	0.10323	0.03871	0.03548	0.05484	0.03871	0.07742	0.01935
1SHA	A	0	0.06731	0.07692	0.04808	0.05769	0.07692	0.08654	0.04808	0.04808	0.02885
1SHF	A	0	0.03390	0.08475	0.06780	0.06780	0.03390	0.08475	0.05085	0.06780	0.01695
1THO	A	0.00917	0.09174	0.05505	0.11009	0.01835	0.01835	0.02752	0.03670	0.04587	0.00917
1TIE	A	0	0.06977	0.04651	0.04070	0.04651	0.04070	0.07558	0.02907	0.09302	0.01163
1TNF	A	0	0.03822	0.03822	0.08280	0.04459	0.05732	0.08280	0.02548	0.08280	0.01911
1UUG	A	0.01310	0.03930	0.05240	0.07424	0.02183	0.04367	0.05240	0.04803	0.07424	0.05677
1WQM	A	0.01538	0.03846	0.03846	0.10769	0.03846	0.10769	0.04615	0.02308	0.06923	0.00769
1XAD	A	0.01739	0.04928	0.03478	0.12174	0.01739	0.06087	0.04928	0.04348	0.10435	0.01739
1XOB	A	0.00926	0.09259	0.05556	0.11111	0.01852	0.00926	0.02778	0.03704	0.04630	0.00926
2AAA	A	0.01653	0.02479	0.07645	0.07231	0.07231	0.02273	0.11157	0.02893	0.06405	0.01446
2ABH	A	0	0.09657	0.06542	0.10903	0.03738	0.01246	0.05919	0.03738	0.06854	0.00312
2ACH	B	0.05000	0.05000	0.12500	0.05000	0	0.07500	0.05000	0.10000	0.10000	0
2MIN	A	0.03259	0.07739	0.04073	0.06314	0.04277	0.05092	0.06517	0.03870	0.06925	0.02240
2SMS2	A	0.01550	0.04651	0.06977	0.10853	0.03101	0.03101	0.10078	0.03101	0.10853	0.00000
2PIA	A	0.01869	0.04050	0.06854	0.07477	0.01558	0.08100	0.09346	0.04984	0.06231	0.02181
2SN3	A	0	0.12308	0.04615	0.04615	0.09231	0	0.06154	0.01538	0.01538	0
2SOD	B	0.00662	0.06623	0.07947	0.05960	0.00662	0.02649	0.05298	0.02649	0.09934	0.05298
2TAA	A	0.01883	0.04184	0.08368	0.07741	0.07113	0.02092	0.07741	0.02929	0.06067	0.01255
2TDM	A	0.01899	0.06329	0.05063	0.06329	0.04430	0.03797	0.04114	0.06013	0.05063	0.06013
3COX	A	0.02761	0.06114	0.07692	0.08481	0.03945	0.04142	0.05917	0.04142	0.07890	0.00789
3SC2	B	0.02632	0.01316	0.09211	0.08553	0.07237	0.06579	0.05921	0.01974	0.07237	0.03289
3VGC	B	0	0.06107	0.09924	0.08397	0.01527	0.00763	0.11450	0.04580	0.09924	0.01527
4CPV	A	0	0.12037	0.04630	0.18519	0	0.00926	0.04630	0.09259	0.04630	0.00926
4ENL	A	0.01147	0.08257	0.04587	0.12615	0.02064	0.03211	0.07339	0.03670	0.07798	0.02523
4ICD	A	0.03125	0.07452	0.04327	0.09135	0.03606	0.04087	0.03125	0.02404	0.07212	0.01202
4INS	B	0	0.03333	0.03333	0.06667	0.06667	0.03333	0.03333	0.10000	0.10000	0.06667
4RCR	H	0.03462	0.05385	0.04615	0.10385	0.02692	0.04231	0.04615	0.03846	0.07692	0.02308
4RHN	A	0.02609	0.07826	0.01739	0.06957	0.01739	0.03478	0.05217	0.03478	0.06087	0.06087
5NN9	A	0.01289	0.04124	0.07732	0.04124	0.04639	0.06186	0.09021	0.02320	0.05928	0.01804
7AAT	A	0.03242	0.06983	0.04239	0.08978	0.03741	0.05985	0.07232	0.03990	0.05985	0.02244

Table A.10. Composition values of test data set proteins' for the last 10 amino acids.

PROT	CHAIN	L	D	E	G	P	N	W	Q	I	C
1AB9	B	0.06870	0.04580	0.03817	0.08397	0.02290	0.05344	0.03053	0.03817	0.04580	0.03053
1AJH	A	0.11765	0.04575	0.09150	0.07190	0.02614	0.00654	0.01307	0.03268	0.05882	0
1ATX	A	0.02174	0.02174	0.02174	0.17391	0.04348	0.08696	0.04348	0.02174	0.06522	0.13043
1BBL	A	0.13725	0.05882	0.09804	0.07843	0.01961	0.05882	0	0.01961	0.03922	0
1BJ8	A	0.05505	0.06422	0.07339	0.02752	0.06422	0.04587	0.03670	0.02752	0.08257	0.00917
1BLN	B	0.08257	0.02752	0.03670	0.08257	0.05505	0.02752	0.02752	0.02294	0.02294	0.02294
1BW4	A	0.04800	0.08000	0.00800	0.09600	0.04800	0.06400	0.04000	0.06400	0.03200	0.04800
1COB	A	0.05298	0.07285	0.05298	0.16556	0.03974	0.03974	0	0.01987	0.05960	0.01987
1CYO	A	0.08602	0.06452	0.12903	0.06452	0.03226	0.03226	0.01075	0.02151	0.05376	0
1DNK	A	0.08846	0.07692	0.03846	0.03462	0.03462	0.04615	0.01154	0.03462	0.04615	0.01538
1EGF	A	0.07547	0.07547	0.03774	0.11321	0.03774	0.05660	0.03774	0.01887	0.03774	0.11321
1EPP	E	0.05758	0.06364	0.01515	0.11515	0.03939	0.01818	0.01515	0.02727	0.05455	0.00606
1GLA	G	0.07186	0.05389	0.07984	0.08583	0.02794	0.03992	0.02595	0.03992	0.06786	0.00998
1HCC	A	0.03390	0.03390	0.10169	0.11864	0.10169	0	0.01695	0.01695	0.06780	0.06780
1HUU	A	0.05556	0.05556	0.08889	0.07778	0.04444	0.04444	0	0.02222	0.04444	0
1IFA	A	0.12658	0.01266	0.08861	0.01266	0.00633	0.06329	0.02532	0.08228	0.05063	0.00633
1IXA	A	0.05128	0.10256	0.10256	0.12821	0.05128	0.10256	0.02564	0.02564	0.02564	0.15385
1M6B	A	0.09984	0.04670	0.04831	0.08857	0.05314	0.07085	0.01288	0.04187	0.03543	0.05797
1MDA	A	0.03883	0.03883	0.08738	0.06796	0.06796	0.01942	0.00971	0.00971	0.03883	0.00971
1MDN	A	0.11765	0.05229	0.09150	0.09804	0.02614	0.02614	0.01307	0.04575	0.03922	0
1MRR	A	0.09867	0.06667	0.08533	0.03467	0.03467	0.04533	0.01867	0.05867	0.07467	0.01333
1NIP	B	0.07266	0.05882	0.10035	0.09689	0.02768	0.04498	0	0.02422	0.07612	0.02422
1NN2	A	0.04124	0.06959	0.03866	0.09536	0.03866	0.06443	0.03093	0.03093	0.06959	0.04639
1OVO	A	0.05357	0.05357	0.03571	0.05357	0.07143	0.10714	0	0	0	0.10714
1PRC	H	0.12016	0.06589	0.07364	0.07364	0.06977	0.00775	0.01550	0.03488	0.03101	0.00388
1QR5	A	0.06818	0.09091	0.07955	0.07955	0.01136	0.02273	0	0.05682	0.10227	0
1RNR	A	0.09893	0.06684	0.08556	0.03476	0.03476	0.04545	0.01872	0.05882	0.07487	0.01337
1SBP	A	0.06774	0.08387	0.06129	0.06774	0.03871	0.05806	0.02258	0.03226	0.05806	0
1SHA	A	0.10577	0.03846	0.06731	0.06731	0.02885	0.04808	0.00962	0.03846	0.02885	0.02885
1SHF	A	0.08475	0.08475	0.10169	0.06780	0.03390	0.03390	0.03390	0.01695	0.03390	0
1THO	A	0.11927	0.10092	0.04587	0.08257	0.04587	0.03670	0.01835	0.02752	0.08257	0.01835
1TIE	A	0.09302	0.06977	0.08721	0.08140	0.05814	0.02326	0.01744	0.05814	0.03488	0.02326
1TNF	A	0.11465	0.03185	0.06369	0.07006	0.06369	0.04459	0.01274	0.06369	0.05096	0.01274
1UUG	A	0.10480	0.03057	0.06114	0.07860	0.06987	0.04367	0.02620	0.06114	0.04367	0.00437
1WQM	A	0.06154	0.06154	0.02308	0.08462	0.01538	0.07692	0.03846	0.04615	0.03846	0.06154
1XAD	A	0.10435	0.04928	0.09565	0.09565	0.06957	0.02029	0.00870	0.00870	0.03188	0
1XOB	A	0.12037	0.10185	0.04630	0.08333	0.04630	0.03704	0.01852	0.02778	0.08333	0.01852
2AAA	A	0.07645	0.08471	0.03512	0.08264	0.03926	0.05165	0.02273	0.02479	0.05992	0.01860
2ABH	A	0.07477	0.06231	0.04673	0.10592	0.04673	0.05296	0.02492	0.04361	0.05296	0
2ACH	B	0.05000	0.02500	0.02500	0	0.07500	0.07500	0	0.05000	0.10000	0
2MIN	A	0.05703	0.06110	0.08147	0.09369	0.04073	0.03055	0.01833	0.02240	0.07332	0.01833
2SMS2	A	0.05426	0.03101	0.03876	0.06977	0.04651	0.07752	0.01550	0.04651	0.06202	0.01550
2PIA	A	0.07477	0.07477	0.06542	0.07165	0.05296	0.03427	0.01246	0.02181	0.04050	0.02492
2SN3	A	0.07692	0.03077	0.09231	0.13846	0.06154	0.04615	0.01538	0.01538	0	0.12308
2SOD	B	0.05298	0.07285	0.05298	0.16556	0.03974	0.03974	0	0.01987	0.05960	0.01987
2TAA	A	0.07113	0.09205	0.02510	0.08577	0.04393	0.05439	0.01883	0.03766	0.05649	0.02092
2TDM	A	0.11076	0.09177	0.04747	0.05696	0.06013	0.02532	0.02215	0.04114	0.04747	0.00633
3COX	A	0.06312	0.05523	0.03748	0.11637	0.04536	0.05720	0.01775	0.03748	0.04931	0.00197
3SC2	B	0.08553	0.05263	0.02632	0.07895	0.06579	0.02632	0.03947	0.03947	0.03289	0.01316
3VGC	B	0.06870	0.04580	0.03817	0.08397	0.02290	0.05344	0.03053	0.03817	0.04580	0.03053
4CPV	A	0.08333	0.12963	0.05556	0.07407	0	0.02778	0	0.01852	0.04630	0.00926
4ENL	A	0.09174	0.07110	0.05734	0.08486	0.03440	0.04358	0.01147	0.02064	0.05046	0.00229
4ICD	A	0.07452	0.06010	0.08413	0.09615	0.04808	0.03606	0.01442	0.02644	0.08894	0.01442
4INS	B	0.13333	0	0.06667	0.10000	0.03333	0.03333	0	0.03333	0	0.06667
4RCR	H	0.09231	0.05000	0.06154	0.09615	0.08846	0.03077	0.01154	0.01923	0.05000	0.00769
4RHN	A	0.07826	0.06957	0.04348	0.11304	0.06087	0.01739	0.00870	0.04348	0.09565	0.01739
5NN9	A	0.04639	0.05928	0.05412	0.07732	0.05670	0.06701	0.03608	0.02062	0.06443	0.04639
7AAT	A	0.07980	0.04988	0.05736	0.07481	0.04239	0.04239	0.01746	0.03741	0.05985	0.01247

APPENDIX B: Bounds of Boxes Constructed for Protein Folding Type Problem

Table B.1 Bounds of first 8 boxes constructed for protein folding type problem.

BOUNDS		BOX1	BOX2	BOX3	BOX4	BOX5	BOX6	BOX7	BOX8
		ALP	ALP	ALP	ALP	BET	BET	BET	BET
M	L	0	0	0	0	0	0	0	0
	U	0.0612	0.0303	0.0370	0.0317	0.0134	0.0505	0.0556	0.0500
K	L	0.0244	0	0.0244	0.0244	0	0	0	0
	U	0.1226	0.1515	0.1507	0.1515	0.0536	0.0909	0.1389	0.1667
T	L	0	0	0	0	0.0263	0.0263	0.0263	0
	U	0.0671	0.0732	0.0465	0.0821	0.0926	0.0909	0.1641	0.1268
A	L	0	0	0	0	0	0	0	0
	U	0.2041	0.2188	0.1802	0.1489	0.1852	0.1212	0.1238	0.1000
Y	L	0	0	0	0	0	0	0	0
	U	0.0427	0.0976	0.0685	0.0617	0.0491	0.0853	0.0553	0.1000
R	L	0	0.0065	0.0065	0	0	0	0	0
	U	0.0612	0.0909	0.1181	0.1500	0.0402	0.0606	0.0625	0.1333
S	L	0.0137	0	0.0137	0	0	0	0	0
	U	0.0890	0.0976	0.0814	0.0988	0.1389	0.1010	0.1429	0.1308
F	L	0	0	0	0	0	0	0	0
	U	0.1029	0.0496	0.0233	0.0725	0.0463	0.0559	0.1111	0.0877
V	L	0	0	0	0	0	0	0	0
	U	0.1020	0.0909	0.0756	0.1233	0.1071	0.0960	0.1143	0.0861
H	L	0	0	0	0	0	0	0	0
	U	0.0340	0.0709	0.0185	0.0850	0.0134	0.0505	0.0476	0.0397
L	L	0	0.0411	0.0411	0.0411	0	0.0141	0.0141	0.0141
	U	0.1224	0.1818	0.1759	0.1818	0.0714	0.0588	0.0829	0.1316
D	L	0	0	0	0	0	0	0	0
	U	0.1132	0.0732	0.0756	0.1111	0.0463	0.0794	0.1111	0.0845
E	L	0	0.0204	0	0.0204	0	0	0	0
	U	0.0755	0.1818	0.1204	0.1818	0.0179	0.0606	0.1818	0.0960
G	L	0	0.0167	0.0167	0	0	0.0202	0.0202	0
	U	0.1361	0.0781	0.0814	0.0959	0.1296	0.1616	0.0922	0.1313
P	L	0	0	0	0	0.0118	0.0118	0.0118	0
	U	0.0411	0.0547	0.0370	0.0496	0.0556	0.0909	0.1389	0.0845
N	L	0	0	0	0	0	0	0	0
	U	0.0816	0.0732	0.0463	0.0725	0.0370	0.0882	0.0833	0.0888
W	L	0	0	0	0	0	0	0	0
	U	0.0137	0.0234	0.0208	0.0196	0.0223	0.0202	0.0394	0.0374
Q	L	0.0071	0	0	0	0	0	0	0
	U	0.1159	0.0976	0.0972	0.0833	0.0370	0.0505	0.0660	0.1215
I	L	0	0	0	0	0	0	0.0088	0.0088
	U	0.0662	0.0976	0.0685	0.0588	0.0223	0.0707	0.0894	0.1212
C	L	0	0	0	0	0	0	0	0
	U	0.0068	0.0244	0.0822	0.0746	0.0370	0.0303	0.0325	0.2000

Table B.2 Bounds of last 8 boxes constructed for protein folding type problem.

		BOX9	BOX10	BOX11	BOX12	BOX13	BOX14	BOX15	BOX16
BOUNDS		APB	APB	APB	APB	ASB	ASB	ASB	ASB
M	L	0	0	0	0	0	0	0	0
	U	0.0400	0.0278	0.0329	0.0187	0.0415	0.0378	0.0165	0.0182
K	L	0	0.0054	0	0.0054	0	0.0387	0.0387	0
	U	0.0880	0.0909	0.1250	0.1486	0.0725	0.1061	0.0576	0.0436
T	L	0.0227	0.0227	0	0	0.0151	0	0	0
	U	0.0818	0.0815	0.0960	0.0748	0.0663	0.0794	0.0847	0.0473
A	L	0	0.0381	0.0381	0	0	0	0	0.0344
	U	0.1593	0.0938	0.1360	0.2297	0.1358	0.1765	0.0905	0.1382
Y	L	0	0	0	0	0	0.0111	0	0
	U	0.0865	0.1136	0.0625	0.0973	0.0542	0.0387	0.0619	0.0364
R	L	0	0.0096	0	0.0096	0	0	0.0072	0
	U	0.0632	0.1136	0.0766	0.0973	0.0748	0.0688	0.0590	0.0073
S	L	0.27	0.27	0	0	0	0.0151	0	0
	U	0.1442	0.0823	0.1230	0.1442	0.0580	0.0830	0.1042	0.1345
F	L	0	0.0104	0.0104	0	0	0	0	0
	U	0.0720	0.0556	0.0451	0.0385	0.0742	0.0805	0.0521	0.0145
V	L	0	0	0.0454	0.0454	0	0	0.0451	0.0451
	U	0.1150	0.0849	0.1025	0.1215	0.0904	0.1364	0.0617	0.1091
H	L	0	0	0	0	0	0	0	0
	U	0.0601	0.0432	0.0292	0.0432	0.0277	0.0358	0.0288	0.0218
L	L	0	0.0162	0	0	0.0545	0.0545	0	0
	U	0.1108	0.1389	0.1019	0.0946	0.1317	0.1226	0.0782	0.0545
D	L	0	0	0	0.0277	0	0	0	0
	U	0.0960	0.1250	0.1250	0.0595	0.0843	0.0945	0.0947	0.0436
E	L	0	0.155	0	0	0.0145	0	0.0145	0
	U	0.0577	0.0833	0.1111	0.1486	0.1377	0.1061	0.0651	0.0145
G	L	0	0	0	0	0	0	0	0
	U	0.1273	0.1321	0.1200	0.1154	0.1014	0.1188	0.1065	0.1200
P	L	0	0	0	0	0	0.0175	0.0175	0
	U	0.0751	0.0909	0.0921	0.0654	0.0742	0.0575	0.0494	0.0509
N	L	0.0104	0	0	0	0	0.0158	0	0.0158
	U	0.0865	0.1085	0.0658	0.0769	0.0580	0.0684	0.0651	0.0618
W	L	0	0	0	0	0	0	0	0
	U	0.0273	0.0465	0.0328	0.0162	0.0217	0.0290	0.0237	0.0109
Q	L	0	0	0	0	0.0144	0	0	0.0144
	U	0.0565	0.0667	0.0526	0.0378	0.0663	0.0627	0.0710	0.0400
I	L	0	0	0	0	0	0.0373	0.0373	0
	U	0.0773	0.0570	0.0694	0.0541	0.1087	0.0875	0.0769	0.0473
C	L	0	0	0	0	0	0	0	0
	U	0.0645	0.0620	0.0556	0.0385	0.0217	0.0503	0.0192	0.0036

VITA

Fadime Yüksektepe Üney completed the high school in Salihli Sekine Evren Anatolian High School, Manisa, in 1998. She received her B. Sc. degree in Chemical Engineering from Istanbul Technical University (ITU), Istanbul, in 2003. Since 2003 she is in the M. Sc. program in Industrial Engineering of Koc University as a teaching/research assistant. For the completion of the program, she has studied the thesis “A Mixed-Integer Programming Approach to Multi-Class Data Classification Problem”. She will be a Ph.D. candidate in Industrial Engineering /Operations Management at Koc University.