FINDING THE MOST PREDICTIVE DATA SOURCE IN BIOLOGICAL DATA

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Sciences

By

Ushashi Chakraborty

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

October 2012

Fargo, North Dakota

North Dakota State University
Graduate School

Title

FINDING THE MOST PREDICTIVE DATA SOURCE IN BIOLOGICAL DATA

By

Ushashi Chakraborty

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr Anne Denton
Chair

Dr Juan Li

Dr Kendall Nygard

Dr Sumathy Krishnan

Approved:

04/05/2013	Dr Brian Slator
Date	Department Chair

ABSTRACT

Classification can be used to predict unknown functions of proteins by using known function information. In some cases, multiple sets of data are available for classification where prediction is only part of the problem, and knowing the most reliable source for prediction is also relevant. Our goal is to develop classification techniques to find the most predictive of the multiple data sets that we have in this project. We use existing classification techniques like linear and quadratic classifications and statistical relevance measures like posterior and log p analysis in our proposed algorithm, which is able to find the data set that is expected to give the best prediction. The proposed algorithm is used on experimental readings during cell cycle of yeast and it predicts the genes that participate in cell-cycle regulation and the type of experiment that provides evidence of cell cycle involvement for any particular gene.

ACKNOWLEDGEMENTS

I would like to thank my adviser, Dr. Anne Denton, for giving me an opportunity to work with her and to learn from her; for her constant support and encouragement; and for tremendous help of every fashion. My heartfelt gratitude to my other committee members: Dr. Kendall Nygard, Dr. Juan Jen Li and Dr. Sumathy Krishnan. Thanks to the Computer Science Department staff, Ms. Carole Huber, Ms. Stephanie Sculthorp and Ms. Lynn Thorp. Special thanks to Angshu Kar, Anuradha Boddeda, Indranil Ghosh, Sarthak Ahuja and Jianfei Wu. My humble regards to my parents who always showed me the right direction, my teachers and my family members. My deepest gratitude to Saurav and many other friends for their constant support and help.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1. INTRODUCTION

Databases are used to store large amounts of data and allow flexible retrieval of that information. Data-mining techniques can be utilized for gaining useful information from particularly large or complex data residing in such databases. Techniques for predicting categorical attributes are called classification. Classification uses existing information about the objects of interest for making predictions [1]. Protein function prediction has been a vast field of study in bioinformatics for the past few years. This field involves input and knowledge of biology, computer science and statistics. In the context of the thesis, the objective can be divided into two main parts - to predict the function of proteins associated with an unknown set of genes and to identify the data source which helps in the most reliable prediction of the protein function. The data used in this thesis involves multiple sources of data. We observed that using the data sources separately in our training for classification gave us different prediction results for the protein function of unknown genes. Hence, we infer that the reliability of our prediction results is dependent on the data source we use for training in our classification techniques.

In this study we have used a microarray analysis of four gene time-series datasets (temperature sensitive mutant methods alpha arrest, arrest by cdc15, cdc28 and elutriation) of *Saccharomyces cerevisiae* (yeast) from the Stanford Microarray Database [2, 3] originally posted by Spellman [4]. Microarray analysis is the study of gene expression while time-series analysis is studying those expressions at different time points. To analyze how different experimental methods affect the ability to predict of the protein function of from gene expression data, we divided the entire data source into four subsets of data as per the mutant methods. Within each of these data sources, half of the genes were put into a training set, and the other half were put in a test set. The division of training and test sets could be done in other ways as well, for example,

keeping more data in training than test set or by leaving a few records out of training set but in the scope of this thesis, we have not done so. The reason is we would like to predict the most reliable data source for every record and not leave any record out of classification. At this point, we saw that for any particular unknown gene, the protein function behavior would be differently predicted while using the various data sources separately for training in the classifiers. Therefore, we introduced a statistical relevance measure which would find out the most reliable data source for each gene. Using the statistical measure along with classification techniques, we were able to predict unknown protein function as well as find the most reliable data source.

The knowledge of the protein function of an unknown gene and the most reliable data source to predict the function can give us information of biological significance. In the current context, prediction of unknown protein function tells us which genes participate in cell cycle regulation and finding the most reliable data source for a particular gene signifies the mutant method that provides evidence of cell cycle involvement.

1.1. Problem Statement for Thesis

The study of protein function prediction has been helped substantially through gene expression experiments and their analysis using data-mining techniques. The techniques allow simultaneous measurement of expression levels of a certain number of genes [4].

Many algorithms exist for prediction based on individual sets of experiments. In this thesis, we focus on prediction from multiple sets of experiments and identify, which data set would be best for the said classification. By using multiple data sets for classification we, by means of the algorithm presented in this thesis, are able to identify the data source that gives the best prediction. Our focus in the thesis is specifically to construct a set of rules (equivalently prepare one single algorithm) that would help us identify the data set for each gene that would

give the best prediction. At this point, the following Table 1.1 would be a good a representation of the way the data look.

Table 1.1. Toy example showing a few genes with their respective expression level ratio at various time intervals for different cycles. The class label for the gene is also mentioned

Gene	Alpha			Cdc 15			Cdc 28			Elu			Class
	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3	Label
G1	0.07	0.12	0.03	1	0.12	0.54	-0.31	..	..	..	..	..	0
G2	-.09	0.24	0.19	1.2	-.15	0.15	1.3	..	..	..	..	..	**1**
G3	0.65	0.54	0.03	1.39	0.23	1	0.1	..	..	..	..	..	**1**
: :													
G2000	..	..	..	..	..	..	..	..	..	..	..	..	0
G2001	..	..	..	..	..	..	..	..	..	..	..	..	0
: :													
G7000	..	..	..	..	..	..	..	..	..	..	..	..	**1**

We have four data sets here: alpha, cdc 15, cdc 28 and elu. Each data set has many time-series-based microarray readings, and each row signifies a particular gene. Each gene in the training data belongs to a particular class label. The class labels in this project can have two values 0 and 1. Class label 1, mentioned in bold in the table, indicates that a particular gene participates in cell-cycle regulation. Class label 0 signifies that the gene did not participate in cell-cycle regulation [2, 3]. Also, there are very few items with class label 1 in the data that we use. Hence, it is of more significance to correctly predict those items with class label 1. By using this information and the proposed algorithm, we are going to find that data set for each gene which, when used for training, gives the best prediction. Using the above data sources separately, we run various classifiers using MATLAB 7 against each data source. We also compute a statistical relevance measure, discussed later in the thesis, analysis for each data source

separately and store the results. Using these data in our proposed algorithm, we try to find that data source which gives the highest statistical-relevance measure value. The algorithm then goes on to find the class label predicted for the most reliable data source for each gene. We use this set of prediction results against the ones we got by using the MATLAB classifier function and compare them. The data sources that we mention correspond to the different phases or mutant methods during cell cycle regulation of yeast. Cell-cycle regulation could happen in any of these phases. For example if gene x (gene x being any random gene in the test set) gets regulated in alpha phase, using the data source alpha should be sufficient for classification for gene x. On the other hand, if gene y (gene y being any random gene in the test set different than gene x) gets regulated in the elu phase, using the data set alpha will give a wrong prediction as the gene would not have regulated by then. Therefore, by using the proposed algorithm, we are finding the most reliable data source for each gene and this data source signifies the phase in which regulation has most likely happened for the same gene.

The following Table 1.2 shows values of the statistical relevance measure for each gene. The measures given in bold represent the highest value for a particular gene. We are proposing that, because a particular data set gives a higher relevance measure than the others, this data set, if used in training, is more likely to predict correctly than others. Hence, we use the prediction result corresponding to the data set with highest relevance measure for each gene. We will be using only that data source for prediction of each record that gave a higher statistical relevance measure value than others. Hence, unlike the other experiments that use the same data source for all records, using the proposed algorithm, we use the most reliable data source for each record.

Table 1.2. Toy example showing a few genes with their respective log p and posterior values. The proposed algorithm helps to identify the data source with the highest log p and posterior values for each gene

Gene	Alpha		Cdc 15		Cdc 28		Elu		Most Predictive data source	
	Statistical Relevance Measure		Statistical Relevance Measure		Statistical Relevance Measure		Statistical Relevance Measure		Statistical Relevance Measure	
G1	0.67	0	**0.89**	**1**	0.88	0	0.74	1	**cdc 15**	**1**
G2	0.22	1	0.99	0	**1.00**	**0**	0.85	0	**cdc 28**	**0**
G3	0.89	0	0.98	0	**0.99**	**0**	0.94	0	**cdc 28**	**0**
:		:		:		:		:		:
G2000		:		:		:		:		:
G2001		:		:		:		:		:
:		:		:		:		:		:
G7000		:		:		:		:		:

We use the results obtained from our algorithm with both these relevance measures and finally compare the results. The results show the value of the proposed algorithm.

The objectives of this thesis can be summarized as (i) classifying and predicting class label of genes present in test set for each data set separately, (ii) calculating the statistical relevance measure for each record and finding the most reliable data source for every record, (iii) predicting class label of genes in test set by using the most reliable data source for classification, and (iv) comparing the prediction results of objective (iii) with the prediction result of objective

(i). The objectives have been explained pictorially below in Figures 1.1, 1.2 and, 1.3. Objective (i), as shown in Figure 1.1, is splitting data from the raw data source into training data set and test data set. Figure 1.1 starts with a raw set of data that we attained from Spellman Microarray Database [2, 3] and then we split this raw data into normalized training and test data. The objective is to predict correctly the class label of the genes present in test data by studying the genes present in training data.

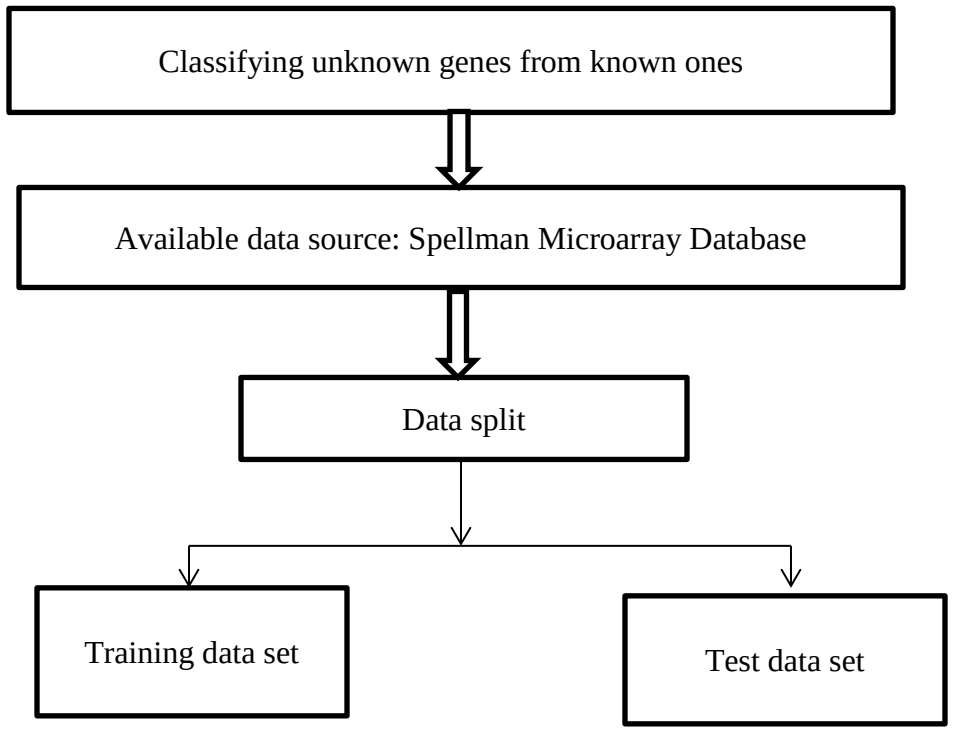

Figure 1.1. A pictorial representation of the data splitting done in this thesis.

Objectives (ii) and (iii) are shown together in Figure 1.2 is the classification part by implementing the proposed algorithm and finding the statistical relevance measure for each record. Objective (iv) is shown in Figure 1.3, is the comparison part between all the experiments done in this thesis, and to find the best predictive data source.

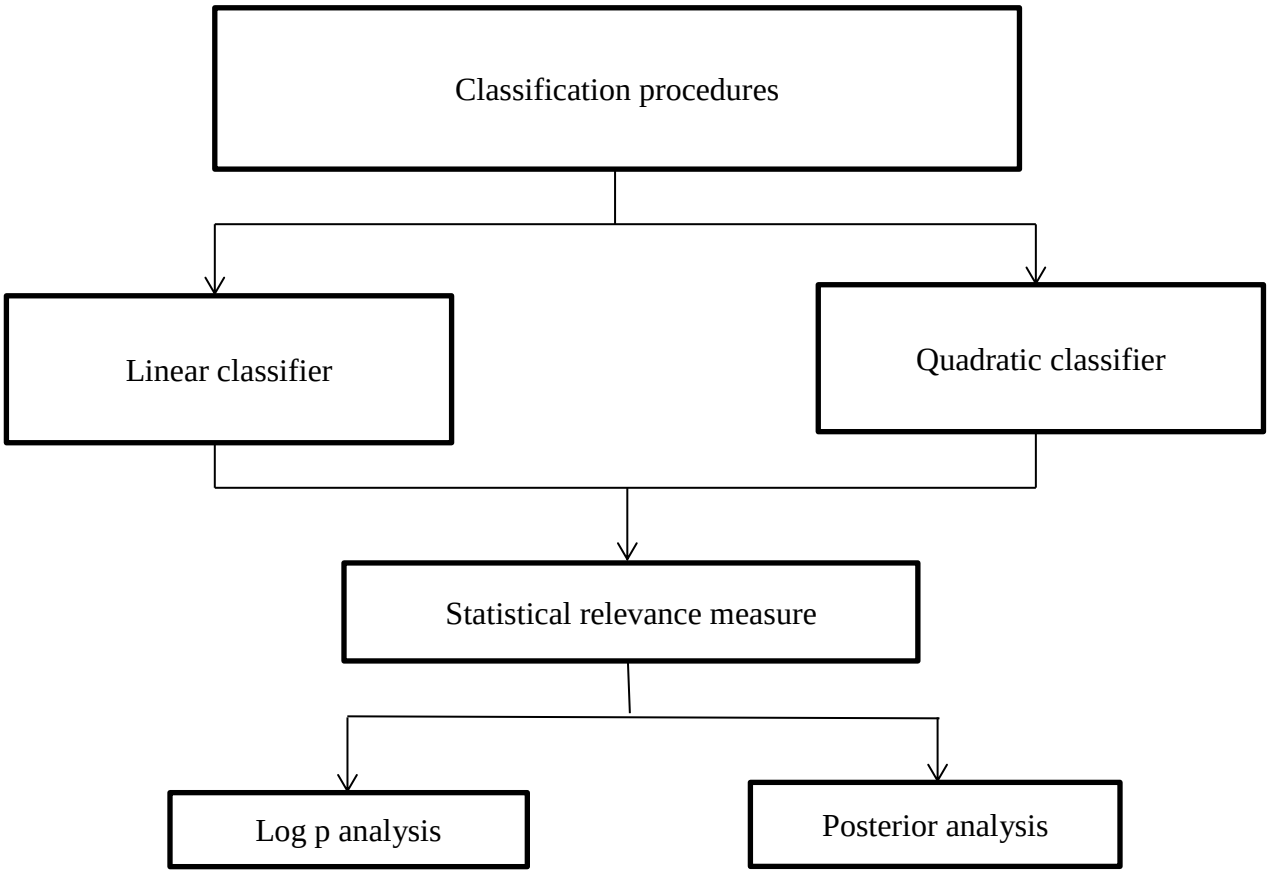

Figure 1.2. A pictorial overview of the classification algorithm used in this thesis.

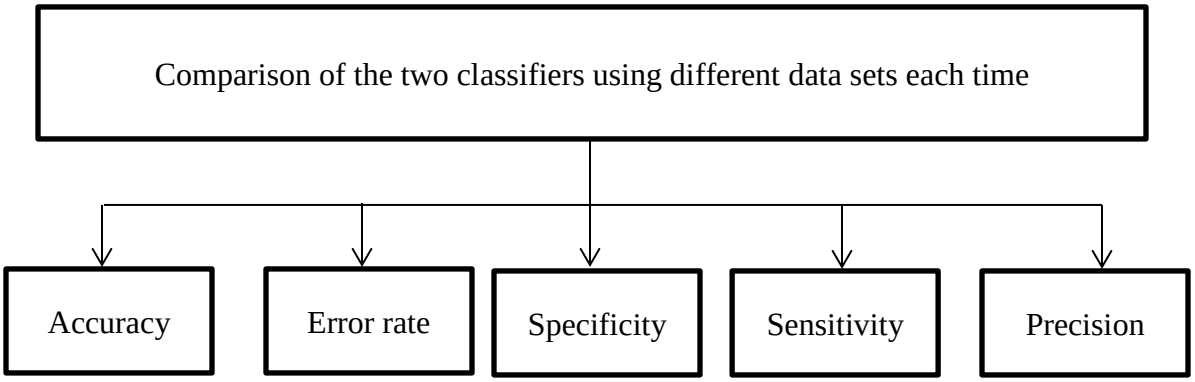

Figure 1.3. A pictorial overview of comparison metrics used for finding the most predictive data source.

The purpose of this thesis is that by virtue of a computer science algorithm we are able to find the most reliable data source which will have a significant biological significance which in the case of this thesis is finding the phase in which cell-cycle regulation has most likely occurred. Future work may show how suitable the algorithm is for data sources that are not biological in nature.

1.2. Organization of the Thesis

This thesis is organized according to the format that is recommended by the university. An overview for each chapter is given below.

Chapter 2 first explains Classification and its different types, along with its significance with respect to the thesis. It then goes on to present the Classification Algorithm used here. It gives an overview of the various prediction techniques used in this thesis and their significance. The ones that are used in the project are linear classification using the four data sets separately, quadratic classification using the four data sets separately, linear classification using log p and posterior and quadratic classification using log p and posterior. This chapter also defines log p and posterior, their usage and their significance.

Chapter 3 initiates discussion on the significance of the data that were chosen for this particular thesis. It talks about how the data were normalized so that the thesis used a fair data set. It talks in detail about microarray analysis and time series analysis. It explains the four stages or cycles that forms the crux of this research: alpha, cdc15, cdc 28 and elu. Then, it goes on to talk about classification and its different types. Various prediction methods are discussed, and their significance is categorized. The ones that are discussed are linear and quadratic classification. A specific method that has been described is the decision-tree classification method.

Chapter 4 talks about the results that each prediction technique has given and their corresponding plots. The metrics which would be helpful in evaluating the results have been defined, and their significance has been discussed. The metrics that have been used for results are mainly accuracy, error rate, specificity, sensitivity and precision. The plots that have been used for comparison between the metrics are specificity vs sensitivity.

Chapter 5 discusses the results given in the prior chapter and talks about their significance. It compares the results given in the plots and talks about which prediction technique was able to give the best results and why. It also talks about the most predictive data source in biological data according to this study. Finally, it talks about possible future work in this field.

CHAPTER 2. CLASSIFICATION ALGORITHM

In this chapter, we first explain what classification is, the various types of classifiers used in the present thesis, how we use multiple data sets and then the classification algorithm that the thesis proposes. Thereafter, we will explain how the multiple data sets are used in this algorithm.

2.1. Classification

Classification means identifying which group an unknown entity belongs to on the basis of observing some unique attributes [5]. Linear classification involves doing the same based on the value of a linear combination of attributes while quadratic classification does this based on a quadratic surface.

In two dimensions, linear classifier is a line. Figure 2.1 shows two types of classes represented by light dots and dark dots. Linear classification would segregate the two classes with straight lines. This type of classification might be simpler and faster than quadratic classification. However, if the members of all classes are intermingled, it is difficult to separate such classes using straight lines. As per the quadratic classification function used in MATLAB, quadratic classification fits multivariate normal densities with covariance estimates stratified by group. The quadratic classifier may not be a line. Figure 2.2 shows two types of classes represented by light dots and dark dots. Quadratic classification would segregate the two classes with any quadratic representation, which may be a line, circle, parabola, ellipse or hyperbole. Usually, quadratic classifiers give better results than linear classifiers. Our prediction results are also better in the case of quadratic classification than linear classification. The Figures 2.1 and 2.2 will help to explain why quadratic classifier is more likely to do a better classification and give better prediction results.

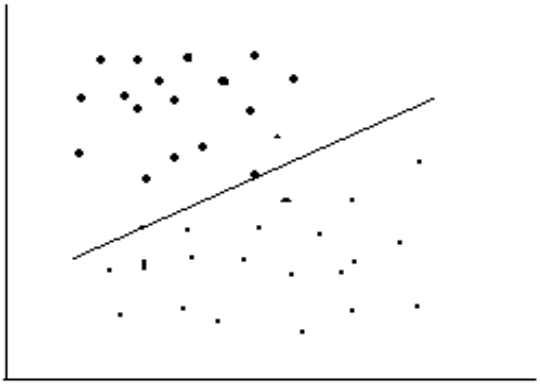

Figure 2.1. Linear classification.

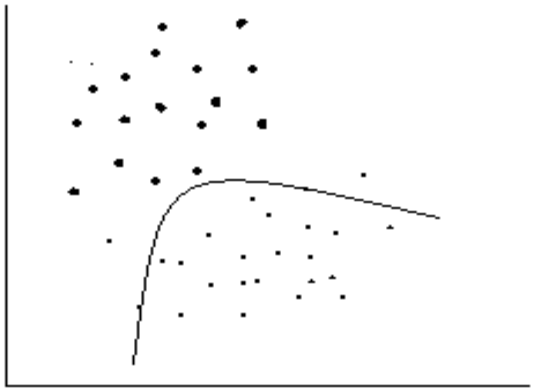

Figure 2.2. Quadratic classifcation.

As discussed in Chapter 1, let us take the example of the classification of animals into groups of mammals and non-mammals. Some important parameters to talk about when doing classification would be class label and attributes. We defined classification in the beginning of this section as grouping a set of unknown things into similar groups. In classification terms, we refer to those groups as being identified by a class label. Attributes are used to assign a class label to an object. We shall discuss class label and attribute in details in sections 2.1.1 and 2.1.2 respectively.

Classification can be based on several kinds of classifiers. Examples could be rule-based classifiers, Bayesian classifiers, nearest-neighbor classifiers or artificial neural network. Rule-based classifiers, as the name suggests, classify unknown entities into groups or classes based on certain rules or if-then scenarios [1]. Nearest-neighbor classifiers group unclassified objects based on the categorization of the nearest of a set of previously classified objects [6]. Bayes theorem can be expressed by the following equation:

$$P(Y|X) = (P(X|Y) * P(Y))/P(X).$$

The equation means that the probability, rather the conditional probability of Y when X is given, is directly proportional to the Bayes theorem and can be derived from conditional probability. The probability of Y occurring when X is true is equal to the quotient of the probability of X and Y both being true and only X being true.

$$P(Y|X) = P(Y \cap X) / P(X)$$

Similarly, we can say

$$P(X|Y) = P(Y \cap X) / P(Y).$$

From the above equations, we can say

$$P(Y \cap X) = P(Y|X) * P(X) = P(X|Y) * P(Y),$$

which brings us to Bayes theorem [1]

$$P(Y|X) = (P(X|Y) * P(Y))/P(X).$$

Neural networks are a technique of artificially mimicking or replicating the biological neural networks and then using the result as the base of study to solve classification problems.

2.1.1. Class Label

Class label identifies the group, in which the unknown entity will be placed after prediction based on studying its characteristics. The class label in this thesis is cell-cycle regulation, and the purpose of the experiments is to identify those genes that participate in cell-cycle regulation. As mentioned in Chapter 1, let us take the example of the classification of animals into groups of mammals and non-mammals. Now based on the study of the behavior of mammals and non-mammals, we keep some of those important behaviors in mind and check for them among the animals that we need to classify. As discussed before, we evaluate on the basis of body temperature and the ability to give birth, and based on the results, we classify the animals to their corresponding groups. This method is often known as the decision-tree

approach. For making a prediction, we should have our classes or class labels ready and then observe for any unique trait in the unknown entity which will help in the decision about which class label to which it belongs. For instance, in the example mentioned here, a unique trait would be “whether the body temperature of the animal is warm or cold?” Observing this trait helps us towards our prediction about whether the animal is a mammal or not. A tree has three types of nodes: a root node, internal nodes and the leaf nodes. The leaf nodes are the class labels while the root node and the internal nodes are those decision-making questions, or as Tan, Steinbach and Kumar [1] call them, the test attribute conditions. Starting from the root node, the test conditions are applied and follow the answers to either the appropriate internal nodes that leads to further decisions or to the leaf nodes that specify the desired class label.

2.1.2. Attributes

Attributes are those characteristics that the test set has that help us to successfully classify them. They are defined as a property or characteristic of an object that may or may not be in another object [1]. Attributes are very important with respect to classification as they become the crux, or deciding factor behind classifying objects under a certain class label. Continuing on the example mentioned in Section 2.1.1 body temperature and the ability to give birth are the two deciding factors, characteristics or attributes. These attributes vary from one object, or in this case animal, to another. The attribute of one animal can be cold body temperature and does not give birth while another can have a hot body temperature and has the ability to give birth. Based on these attributes, we are able to decide into which class label, for example mammals or non-mammals, we can put the animals. With respect to the thesis, the attribute collections and their analysis have been pretty challenging. First, the attribute analysis involved learning and understanding the cell-cycle process explained more elaborately in Chapter 3. Second, the

attributes in this case involved the study of multiple data sets which is explained in the next section. The attributes in this thesis are the data readings at various time points made during the experiments.

2.1.3. Multiple Data Sets

As mentioned before, there were multiple data sets used, which made the problem more complex. The data sets used were collected at various stages in a cell cycle and the corresponding readings at multiple time intervals. The stages studied were alpha, cdc-15, cdc-28 and elu, the stages are explained in Chapter 3. Collecting data across stages in cell cycles and analyzing them to apply data mining and statistical methods were challenging because the change in function of proteins could be either because of a change in the stage of cell cycle or because of participation in regulation. The difference between the two situations is difficult to gauge and could cause a few discrepancies in the results.

2.2. Significance of Combining Classifiers and Using Multiple Data Sources

Using multiple data sources for prediction or for training, where the different data sources describe the same information, is a common data-mining technique. Finding which data source would be most reliable is the problem we have set out to solve. For a data source to be reliable for a particular record or gene in the context can be dependent on the record itself. The data that we used for this thesis are huge, and for a few records, data are not available. Reliability of the data source in the case of that particular record can be affected by such a thing. Because the data that we use involve more than one data source, if one data source has no information, for a particular record, there could be another data source which can be used for the training data for classification. With a large amount of data, the proposed algorithm uses the statistical relevance

measure to identify that data source for each record, which can be more reliable as compared to others.

A combination of classifiers can be very useful when one particular classifier is unable to give the most predictive information. In the literature, a lot has been discussed about the combination of classifiers [7, 8, 9], and there are some common classifier fusion methods, such as majority voting and evaluating statistical significance of each record.

Majority voting is a good technique when the number of data sources is odd. It has been used successfully in the classification of gene expression data [10], pattern recognition [11] and handwriting recognition [12]. Because we have four data sources, we have not used the majority voting technique in this thesis.

The other technique is to evaluate the statistical significance of each record. It has been used with respect to confidence-based classification of a poorly-differentiated tumor [13] and speaker-identification problem [14, 15]. This technique is a measurement-level [7] type of classifier fusion. Using such techniques typically improves the prediction results from earlier results based on running classifiers separately for each data source.

Coming back to the problem of missing data discussed at the beginning of this section, for genetic data, it is a very common issue, and then, classification for this type of data requires multiple data sources. We have incorporated two techniques from the techniques mentioned previously in this thesis: using multiple data sources and using the statistical significance of a record for classification. To validate that the better results that we get by using a statistical relevance measure would hold true generally, we also use different types of classifiers to do the same problem. Figure 2.3 will help explain the different types of classifiers: linear classifier, quadratic classifier, and support vector machine, used in this thesis.

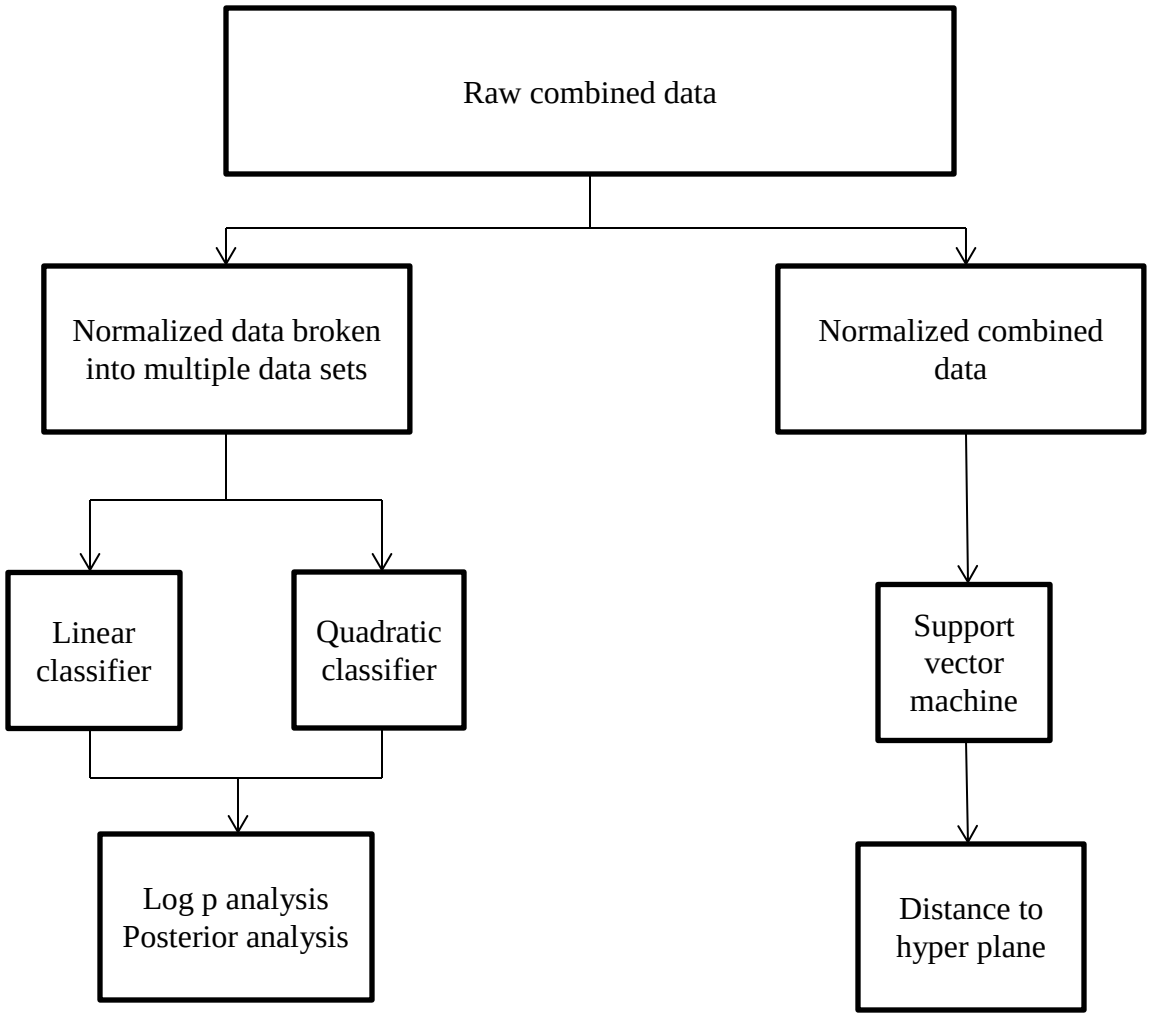

Figure 2.3. A pictorial representation of the different types of classifiers used in this thesis.

The Figure 2.3 shows the different types of classifiers used in this thesis and the different methods used for each one of them. We begin with raw data straight from a data source and then normalize them. The normalization process is explained in Chapter 3. Then, the data are broken into multiple data sets for running the quadratic and linear classifiers while the entire data are used together for analysis using support vector machines. For quadratic and linear classifiers, the statistical relevance measure (log p and posterior analysis for this thesis) is calculated for each

record. For the support vector machine, the MATLAB function for running svm classify was run, and results were generated and compared with the ones obtained using the proposed algorithm.

2.3. Related Work

The significance of classifiers [16, 17, 18] is a topic where a lot of work has been done. However, most of these approaches are based on an assumption that input data may have different or higher classification accuracy for one data set (which is a part of the input data) as compared to others. In the perspective of the thesis, such is not the case. We are trying to find that data set that would give higher classification accuracy as opposed to others for an individual object. One of the often-used methods to differentiate between class labels on the basis of attributes is discriminant analysis [19]. We use linear and quadratic classifiers for our thesis. Linear discriminant analysis is supposed to have lower accuracy results for multi-class classification [20] and component analysis [21]. The poorer results are because of the assumptions on the covariance matrix. With quadratic discriminant analysis, on the other hand, the input data are assumed to have a normal distribution [22], resulting in more accurate results. In regard of this thesis, we see that, using quadratic classifier, we always get better results as compared to a linear classifier. Another method that is used often is support vector machines [23]. Support vector machines are utilized to create models based on learning or studying the training data to analyze data and recognize patterns. Support vector machines construct a hyper plane in infinite dimensional space. The distance between the hyper plane and training data points forms the basis for classification study [24, 25]. Distance from hyper plane using support vector machine implementation, SVM^{perf} [26], is also considered for classification. Another method used is conformal prediction where results are compared using a nearest-neighbor approach [27]. It is quite similar to the approach that we have undertaken. It involves using

information from a prediction for a future prediction. Conformal prediction has many applications in data mining [28, 29]. In the thesis, we use statistical relevance measures for each gene and its corresponding data to generate a matrix. We choose the record with the highest statistical relevance measure and use this information for our prediction.

2.4. Introduction to Algorithm

Classification algorithms are used extensively to predict protein function [30]. However, the idea here is not only to predict protein function, but also to find the most predictive data source for each record.

2.4.1. Predictive Data Source Algorithm

As mentioned in section 2.2, that the motive for this algorithm is to find the most predictive data source for the purpose of predicting from multiple sources. To do so, we calculate a statistical relevance measure for each record and each data set. We claim that using the data set that gives us the highest relevance measure for our prediction will give us better results. The relevance measures that we have calculated here are log p and posterior analysis values for each gene separately for each data set. Gene x will have different log p and posterior analysis values for alpha, cdc 15, cdc 28 and elu. We use the algorithm to find the maximum log p and posterior analysis value as well as the corresponding data set that gives us that result for each gene. We now use this data set for our prediction for gene x. Figure 2.4 explains pictorially what the algorithm strives to do. The figure is made for any particular gene and for posterior analysis results. The same pattern is followed for log p results. It shows that the multiple data sets are run with the various classifiers and for each data set the posterior values are calculated. Thereafter, the algorithm calculates the maximum posterior value for each record and returns the corresponding data set. This data set is then used for further prediction.

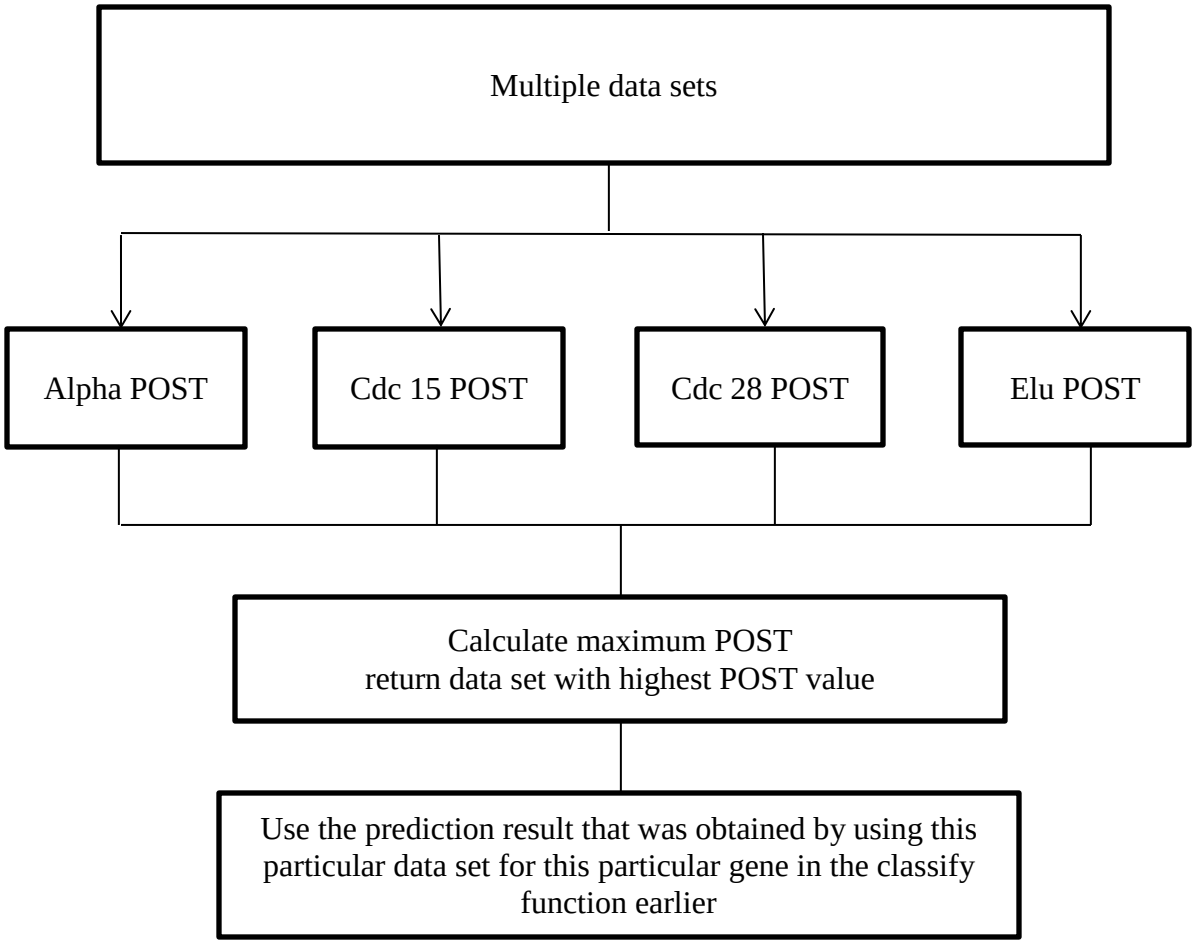

Figure 2.4. A pictorial representation of the Predictive Data Source Algorithm.

2.4.2. Outline of Algorithm

Before the algorithm was run, we first ran a quadratic classification for each data set separately using MATLAB classifiers. We had predictions for each data set. We then used the MATLAB function to find out log p and posterior values for each gene and for each data set separately. These values were stored as Microsoft Excel datasheets or .csv files. For example, the datasheet having posterior values was called posterior.csv and was used as input data for our proposed algorithm. We used Excel macros to find the maximum log p or posterior value as well as the corresponding data set for it. The proposed algorithm finds the initial prediction result by

using the quadratic classifier for the data set that gives the highest log p and posterior value for a particular gene. Instead of using same data set while predicting the class label, by finding the data set that gives highest log p or posterior value, we use only that dataset while predicting results.

2.4.2.1. Algorithm: Predictive Data Source

```
1. StreamReader sr = File.OpenText("posterior.csv");
2. String tmp;
3. String[] tmpArr;
4. while ((tmp = sr.ReadLine()) != null)
5. {
6. if (!tmp.StartsWith("Largest Class"))
7. {
8. tmpArr = tmp.Split(',');
9. tw.WriteLine(tmpArr[0] + '\t' + tmpArr[1] + '\t' + tmpArr[2] + '\t' + tmpArr[3] + '\t' +
   tmpArr[4]+'\t'+tmpArr[col[tmpArr[0]]]);
10. }
11. }
```

2.4.3. Linear Classification Using Data Sets Separately

The four data sets are used separately for these predictions. Linear classification was explained in Section 2.1. Also, as discussed earlier, the data sets are taking part in four different cycles: the alpha factor arrest, arrest of cdc15, cdc28 temperature-sensitive mutants and elutriation [2, 31]. Instead of using all the data together, i.e., including the expression level of the genes in every time series in each cycle, the data were separated with respect to each cycle. We

obtained different training and test sets for the alpha, cdc 15, cdc 28 and elu cycles. Each of these data sets was used separately in the MATLAB 7.0 statistical toolbox.

The classify function in MATLAB uses three arguments: sample, training and group. It classifies each record of the data in the sample into one of the groups in training. Sample and training must be matrices with the same number of columns. Group is a grouping variable for training. Training and group must have the same number of rows. In this thesis, the sample would be the test set (alpha test set, cdc 15 test set, cdc 28 test set and elu test set): training would be the training set (alpha training set, cdc 15 training set, cdc 28 training set and elu training set); and group would be the class label for the training set. The output class indicates the class label to which each row of the test set has been assigned, and is of the same type as the class label. The function would give a matrix with same number of rows as the group or class label which would be the predicted group or class label for the sample or the test set. Usually, the classify function in MATLAB treats NaNs or empty strings in the group as missing values and ignores the corresponding rows of training. However, in this thesis, all the missing values, or NANs, were replaced with 0. The reason for this is that, in microarray experiments, the expression levels are deciphered on the basis of the logarithmic ratio of red by green expression. In the case of NAN, we can be neutral and assume that the red and green expression level is same, and hence their ratio would be 1. The logarithmic value of 1 is 0, and hence all NAN values were replaced with 0.

The Figure 2.5 shows us a graphical representation of how a sample training class label plot looks. A sample result plot or test class label plot is shown in Chapter 3. There are very few genes which participate in the cell cycle regulation or have class label 1.

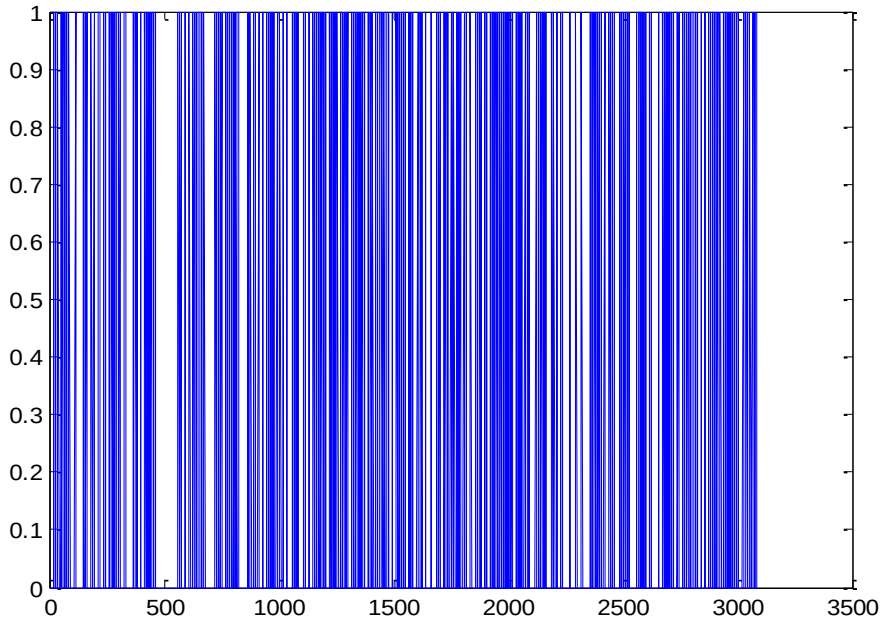

Figure 2.5. A sample training class label plot. The X-axis represents the gene number, and the Y-axis represents expression level.

For the first technique, a linear classifier was used for each of the data sets separately. As mentioned in the help text of the statistical toolbox in MATLAB, a linear classifier "fits a multivariate normal density to each group, with a pooled estimate of covariance" [32, 33]. In MATLAB, this sort of classification is done by default with the classify function.

2.4.4. Quadratic Classification Using Data Sets Separately

As also implied in section 2.2, in most cases, quadratic classification gives better results than linear classification. Hence, the classify function in MATLAB was used as a quadratic classifier. In this case, the classify function has four arguments. In addition to the arguments mentioned in Section 3.1, the extra parameter that was used here is the type. The arguments are sample, training, group and type wherein the type specifies the type of discriminant function.

Various types could be diaglinear, quadratic, diagquadratic or mahalanobis. If the type is not mentioned, by default, the type is taken as linear.

In this technique, we have used the type as quadratic. As mentioned in the help text of the statistical toolbox in MATLAB, a quadratic classifier "fits multivariate normal densities with covariance estimates stratified by group" [32, 33]. Once again, all the data sets have been used separately. As explained in Section 3.1, the rest of the arguments remain the same, hence the output is also test class label. It will be seen in Chapter 4 and 5 whether we get better results, as expected, from this type of classification.

2.4.5. Linear and Quadratic Classification Using Log p and POST

After using linear and quadratic classifiers on the test set, log p and posterior probability calculations were used for both linear and quadratic classification. To define the terms log p and posterior, let us see what the help text in the discriminant analysis chapter of the statistical toolbox in MATLAB says about them.

"[class,err, POSTERIOR] = classify(...) returns a matrix POSTERIOR of estimates of the posterior probabilities that the jth training group was the source of the ith sample observation, i.e., Pr(group j|obs i)." [32, 33]

"[class,err,POSTERIOR,logp] = classify(...) returns a vector logp containing estimates of the logarithms of the unconditional predictive probability density of the sample observations, p(obs i) = p(obs i|group j)Pr(group j) over all groups." [32, 33]

In simpler language and in the context of this thesis, posterior probability can be explained as the probability of any random gene participating in cell-cycle regulation, which is the conditional probability that a particular training set has been used for the classification. Log p can be explained as the summation of the product of conditional probability of a random training

set being used for a particular gene and the probability of using the training set. From the explanation and using Bayes theorem as explained in Section 2.2, we can mathematically describe posterior probability and log p as follows.

$$\text{Posterior Probability} = \text{Probability}\{\text{gene y} \mid \text{training set x}\}$$

$$= (\text{Probability}\{\text{training set x} \mid \text{gene y}\} * \text{Probability}\{\text{gene y}\}) / \text{Probability}\{\text{training set x}\}$$

$$\text{Log P} = \sum \text{Probability}\{\text{gene y} \mid \text{training set x}\} * \text{Probability}\{\text{training set x}\}$$

Posterior analysis has been conducted in microarray experiments before [34, 35] for other types of biological data. However, because the challenge of this thesis is using multiple data sets, predicting a gene, y, also depended on which data sets were being used. In some case while, say gene y was predicted right when we used the alpha data set, it was predicted wrong when we used the cdc-15 data set. By using the posterior probability, we developed a matrix of these probabilities for the occurrence of gene y in regulation, and we used the data set which gave the maximum probability.

Linear and quadratic classifiers were both used separately for result analysis. The differences between linear and quadratic classifiers are covered in Sections 3.1 and 3.2, and they hold the same significance in these experiments, too.

CHAPTER 3. BIOLOGICAL DATA

In this thesis, the initial goal was to predict whether a set of genetic data participated in cell-cycle regulation based on given data by using classification techniques. The next part of the objective was to analyze the prediction results and to find the most predictive data source.

The four datasets that were retrieved from the Stanford Microarray Database (SMD) were organized in a single table in such a way that, at one glance, we could find the expression-level measure at a specific time for a specific gene in a specific cycle. This table, as mentioned before, had data of all genes of yeast synchronized by four different methods (alpha, cdc15, cdc28 and elu) and their expression levels at various time points of the Saccharomyces cerevisiae cell cycle. The data that were obtained from the SMD were raw data and had huge disparities, so the data were normalized. Normalization was done by calculating the row mean and row standard deviation of the entire data set. Then, each value was normalized by subtracting the row mean from it and then dividing by the row standard deviation. After this, two separate tables were made from the main table. All the even-numbered rows in the main table (that was of the form of an Excel spreadsheet) were kept in one spreadsheet and were called the Training Set while the rest of the rows, or the odd-numbered rows were put in a separate spreadsheet and named the Test Set. The Training Set and Test Set had expression-level measures of genes taking part in all the four cycles.

At this point, we retrieve another table from the Stanford Microarray Database, the class label. The class label table tells us the names of those genes that take part in the microarray hybridization and are regulated in yeast cell cycles. The class label is also divided into two tables with respect to our existing tables, i.e., Training Set and Test Set. The genes that are a part of the class label and take part in microarray hybridization to show the regulated behavior in yeast cell

cycles as well as a part of the Training Set now are put into table called Training Set Class Label. Likewise, the genes that are a part of the class label and take part in microarray hybridization to show regulated behavior in yeast cell cycles as well as a part of the Test Set now are put in a table called Test Set Class Label.

To summarize what was discussed above, we had four tables: the Training Set, the Test Set, the Training Set Class Label and the Test Set Class Label. Apropos to our objectives mentioned in Section 2.1 and the beginning of this chapter, our prime objective was to predict the names of the genes present in the Test Set Class Label. To do such a prediction it was necessary to study the Training Set and the Training Set Class Label. The studying of the training and test sets were done by overall by all techniques mentioned in Sections 3.1, 3.2 and 3.3. However, each of those techniques had different prediction results, as we will see in Chapter 4. Let us discuss each technique that was implemented for this prediction.

In bioinformatics, microarray analysis plays an important role. Microarray analysis is a high-throughput process which tells us how different genes are relatively expressed in an organism [2, 4]. In the microarray process, the ribonucleic acid (RNA) of an organism is extracted. Its complementary deoxyribonucleic acid (cDNA) is prepared and fluorescently labeled. It is later hybridized to a slide where small oligonuclotides are present. If a respective gene is present and expressed, the fluorescence levels of these genes are measured, and later, this microarray data can be used to find the relative presence and expression of particular genes in an organism [4].

3.1. Material and Methods

In the present project, initially four gene time-series data sets (alpha, cdc15, cdc28 and elu) of *Saccharomyces cerevisiae* (yeast) from the Stanford Microarray Database [3, 36] were

retrieved, originally posted by Spellman [2]. To know more about the data sets, a brief description of the cell-cycle regulation in yeast is needed. The cell cycle of yeast is a process where a parent yeast cell produces two daughter cells which contain similar genetic information as in the parent cell [37]. The cell cycle includes a DNA replication process where there are two main steps: a DNA synthesis step called the S-phase and a mitosis step (M-phase). These two steps in DNA replication are separated by two gaps, known as G1 and G2 [38]. Various genes in *Saccharomyces cerevisiae* are regulated at different time points of their cell cycle. Thus, there may be different genes and expression patterns in yeast which need to be studied for a better understanding of the cell-cycle regulation of yeast. Proper regulation of the genes will help the yeast to function normally.

Synchronization is a method that can be performed on yeast cells to understand their cell-cycle events where cells are sorted either at a particular time point in their life cycle or by their size and temperature sensitivity. Alpha factor arrest, elutriation, and arrest of cdc15 and cdc28 temperature-sensitive mutants are some of the synchronization methods available [31, 2]. Elutriation is a one cell-cycle synchrony method; two-cycle synchrony is by alpha; and three-cell cycle synchrony is by cdc15 method. Spellman et al. [2] extracted the RNA from yeast cells synchronized by different methods (alpha, cdc15, cdc28 and elu) at various time points in their life cycles. They later used the RNA in microarray hybridization, and analyzed the data to identify genes regulated in yeast cell cycles. The data were made available in the Stanford Microarray Database.

Microarray data include information related to genes of various species and may also include data at various time points of cell-cycle regulation in an organism [2]. However, predicting genes and their behavior in the organism after microarray analysis is extremely

difficult, especially when the data are enormous and include repeated measurements [39]. Classification can help understand the data by using a supervised learning method where the data are divided into training and test sets. A training set generally includes classes (characteristics or gene functions and behaviors) that are known, and a test set includes data where classes are unknown. Thus, the goal of classification is to predict the unknown data using information from known attributes, in this case, prediction of test set using training set details using a decision-tree algorithm. Discriminant analysis with MATLAB or other software can later be used to evaluate how significant the predictions are.

Many algorithms, such as the decision-tree algorithm, k-nearest-neighbor analysis, artificial neural networks and support vector machines (SVM) help us in classifying data [40, 41, 42]. There are binary (two classes involved) and multi-class type methods available in classification. Cross validation becomes an important aspect in classification, where multiple ways of breaking up data sets in different ways are used to derive training and test sets.

The main objectives of the present project were (i) dividing gene expression data in yeast which have been synchronized by four methods as described earlier (alpha, cdc15, cdc28 and elu) into training and test sets, (ii) classifying the data to know which genes were being expressed that help in cell-cycle regulation and (iii) finding the best possible classification/prediction. The cell-cycle gene regulators were the primary classifiers. A decision-tree algorithm (using MATLAB) was used to decide which genes were being expressed in the yeast samples that were synchronized by various methods.

Four data sets originally posted by Spellman et al. [2] were retrieved from the Stanford Microarray Database, and they were organized in a single table. That table had data for all yeast genes synchronized by four different methods (alpha, cdc15, cdc28 and elu) and their expression

levels at various time points of the Saccharomyces cerevisiae cell cycle. The time series data sets included genes and their expression microarray data collected at 7-minute intervals up to 140 minutes for the alpha method, 10-minute intervals up to 300 minutes for the cdc15 and cdc28 methods, and 30-minute intervals up to 6.5 hours for the elutriation method (Spellman et al. [2]). Later, these gene data were divided into two categories, training and test sets (Figures 3.1 and 3.2).

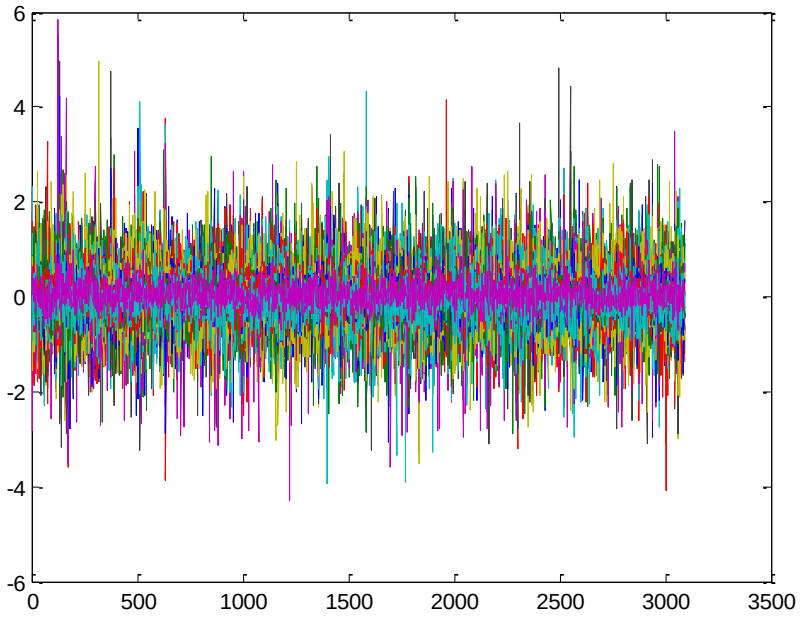

Figure 3.1. Training gene dataset of yeast. The X-axis represents gene number, and the Y-axis represents gene expression level or logarithmic ratio of red by green.

The Figure 3.1 is a MATLAB generated plot for the training data set combining the multiple data sets. There are a total of more than 7000 genes in the entire data set, and the training and test sets each have more than 3000 genes as the figures 3.1 and 3.2 show. From the figures, it is clear that there are some large disparities in expression level or logarithmic ratio of red by green among the genes because the graphs were plotted from raw data. While some genes

give expression levels as high as 6, there are some genes that have expression levels as low as -4. Using these data would not be good for prediction. Hence, the data were normalized, and the normalization process was explained at the beginning of this chapter.

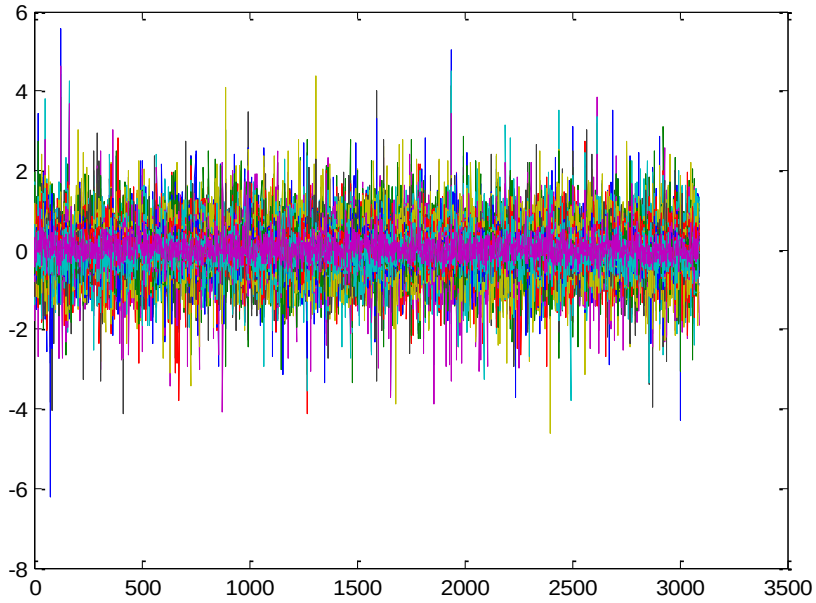

Figure 3.2. Test gene dataset of yeast. The X-axis represents gene number, and the Y-axis represents gene expression level or logarithmic ratio of red by green.

Figure 3.2 is a MATLAB generated plot for the test data set combining the multiple data sets. Here, too, we see a lot of disparity in the expression-level. Hence, the test data have been normalized for the experiments.

The normalization was done for each time-point for the microarray results for all the genes taken together, so for one particular time-point, the mean and standard deviation for the microarray readings for all the genes at that time-point were calculated. Then, each reading was normalized by subtracting the mean from it and dividing by the standard deviation. This was done so that the comparison between genes would be fair. The reading of one gene at a time-

point and for another gene at the same point should be made under similar circumstances to lessen any kind of outside interference in the results.

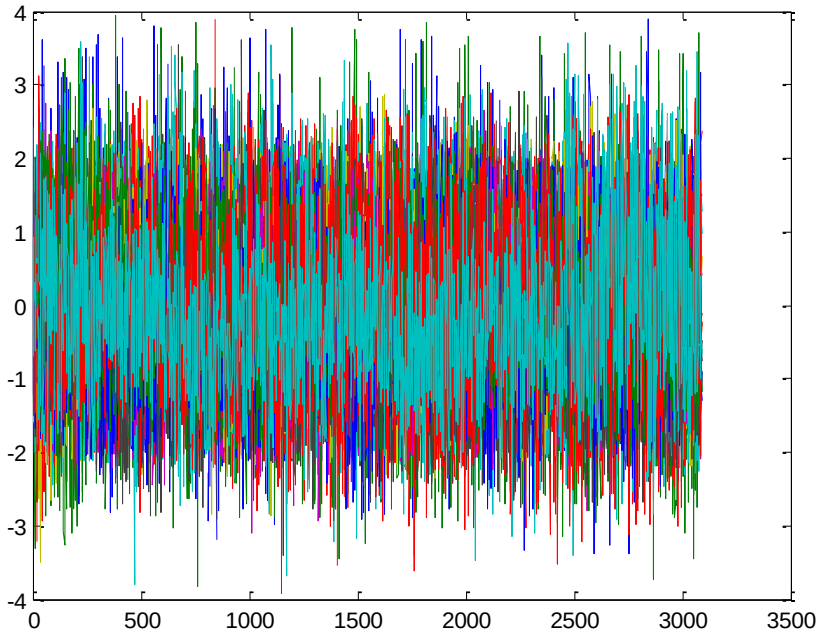

Figure 3.3. Training gene normalized dataset of yeast. The X-axis represents the gene number, and the Y-axis represents gene expression level or logarithmic ratio of red by green.

The Figure 3.3 shows a plot based on readings of the logarithmic ratio of red by green of the normalized training data. We can that the disparity between readings for each gene is less compared to Figure 3.1. However, it is imperative to mention that the normalization process has just ensured fair comparison but has not been involved in difference in the pattern of the results. Another thing to note here is that, for this data set, all the readings that were unreadable, or NaN, have been assumed to be 0. The reason why these readings have been assumed to be 0 is because, if the red by green ratio is assumed to be 1the logarithmic value of red by green would then be 0.

CHAPTER 4. RESULTS AND PLOTS

In my analysis of the results, I have focused on some known and often-used parameters to compare the prediction results. These parameters are accuracy, specificity, sensitivity, precision and F1 measure [1]. The plot mainly used for comparison and other analysis is the specificity vs sensitivity plot. A detailed explanation of said parameters is given below.

4.1. Comparison Metrics in Detail

The comparison parameters used here are dependent on, mostly, four values for each experimental result. These values are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). The explanation of each is given as follows.

- TP = the number of times when the prediction of a gene expression in the training set is 1, while expression of the same gene in the testing set is 1
- TN = the number of times when the prediction of a gene expression in the training set is 0, while expression of the same gene in the testing set is 0
- FP = the number of times when the prediction of a gene expression in the training set is 0, while expression of the same gene in the testing set is 1
- FN = the number of times when the prediction of a gene expression in the training set is 1, while the expression of the same gene in the testing set is 0

Table 4.1, more commonly known as the confusion matrix, summarizes the above definitions. We will be using this table commonly to compare all our results while running to various classifiers on various data sets and while using log p and posterior analysis on the classifiers. The tables presented in this chapter will tell us at a glance how many True Positives and True Negatives were predicted by each of the classifiers. Using the data in the tables we shall find accuracy, specificity, sensitivity and precision for the experiments.

Table 4.1. General confusion matrix of cell-cycle regulation gene presence in yeast

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**True Positive (TP)**	**False Negative (FN)**
Cell-Cycle Regulated Gene Absent	**False Positive (FP)**	**True Negative (TN)**

So with respect to the values used in the confusion matrix, we can define and explain our parameters. Accuracy is the measure of how correct our predictions are. It can be defined by using the above-mentioned values as follows.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

It is often believed that, the higher the accuracy, the better the prediction is and a very high accuracy can result only because of a good prediction. Even though that is true in most cases, a high accuracy might not always mean a great prediction. Consider a case where we trying to predict the N genes where N = 1000. From the 1000 genes, only 10 participate in cell-cycle regulation, or have class label value, 1. Hypothetically, say our prediction predicts only 2 of those 10 genes correctly, or as a value 1, and are the rest are predicted as without expression, or value 0. The accuracy measure measurement of such a prediction would be

$$\text{Accuracy} = (2 + 990) / 1000 = 0.992 \text{ or } 99.2\%$$

That is a very high accuracy rate for a prediction which could only correctly identify 20% of the expressing genes. This parameter does not reflect the true correctness of prediction in the case of a large number of data, especially if the data have a very large number of expressionless genes compared to expressing genes as in this case. Here we are trying to predict the function of

3090 genes, of which only 384 genes actually show the said expression which amounts to only 12% of the total genes. Only measuring the accuracy would not give us a clear idea about our predictions in this case.

Specificity is the measure of how correctly we have predicted the under-expressed genes as under-expressed. Specificity can be defined by using the above mentioned values as follows.

$$\text{Specificity} = TN / (TN + FP)$$

A high specificity ensures that the prediction recognizes all the true negatives. In our prediction, it is imperative that we get a very high specificity because the number of genes that are under-expressed is very large. We have 2706 under-expressed genes in the test set of the total 3090 genes. Hence, it is natural for the classify function to predict a high number of under-expressed genes, but the specificity value measures how correct this prediction is.

Sensitivity, or Recall, is the measure of how correctly we have predicted the over-expressed genes as over-expressed. Sensitivity can be defined by using the above-mentioned values as follows.

$$\text{Sensitivity} = TP / (TP + FN)$$

A high sensitivity ensures that the prediction recognizes all the true positives. In our prediction, the validity of the prediction mainly depends on the sensitivity, and hence it is necessary that we get that high sensitivity because the number of genes that are over-expressed is very few in number. We only have 384 under-expressed genes in the test set with 3090 genes. Hence, the comparison between the different predictions and the strength of the prediction lies in how close the classifier predicts the over-expressed genes because the number of such genes is very low, hence this prediction is tough.

Precision is the measure of how correct the prediction of over-expressed genes has been. This can be defined by using the above-mentioned values as follows.

$$\text{Precision} = TP / (TP + FP)$$

The success of a prediction lies in a high TP and low FP. In an ideal case, a low FP and a high TP would result in high precision.

The F1 measure, or F-measure, gauges the accuracy of the prediction. F1 measure can be defined by using the above mentioned values as follows.

$$\text{F1 measure} = (2*\text{Precision}*\text{Sensitivity}) / (\text{Precision} + \text{Sensitivity})$$

F1 measure can be also said to be the harmonic mean of precision and sensitivity [1]. The F1 measure will be discussed in Chapter 5.

4.2. Results

As discussed earlier, the predictions were first done for each set of data, alpha, cdc 15, cdc 28 and elu, separately and using a linear classifier. Next, the test was performed using a quadratic classifier on each of the data sets. After that, the log p value was taken for the data sets, and the test was performed once using a linear classifier and once using a quadratic classifier. Thereafter, the posterior value was taken for the data sets, and the test was performed using a linear classifier as well as a quadratic classifier. Finally, all the data sets were combined into one and the quadratic classifier was run with the combined data. Also the combined data was used to run support vector machine classifiers and results were generated. The confusion matrix presented for each of these experiments help us to compute the comparison metrics (mentioned in section 4.1 easily and ultimately find which experiment gave the best results.

4.2.1. Alpha Set Using Linear Classify

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions at different time intervals for the alpha cycle. After running the linear classify function in MATLAB 7.0, Table 4.2 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.2. Confusion matrix of cell-cycle regulation gene presence in yeast synchronized by the alpha method for a linear classifier

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 194**	**FN = 190**
Cell-Cycle Regulated Gene Absent	**FP = 483**	**TN = 2223**

From Table 4.2, we can compute the comparison metrics for prediction involving a linear classifier on the alpha data set as follows. The metric results will be used to compare with metric results from other experiments that explained in later sections.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 78.2\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 21.7\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.50$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.82$$

$$\text{Precision} = (TP) / (TP + FP) = 0.29$$

4.2.2. Cdc 15 Set Using Linear Classify

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions at different time intervals for the cdc 15 cycle. After running the classify function in MATLAB 7.0, Table 4.3 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.3. Confusion matrix of cell-cycle regulation gene presence in yeast synchronized by the cdc15 method for a linear classifier

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 207**	**FN = 177**
Cell-Cycle Regulated Gene Absent	**FP = 815**	**TN = 1891**

From Table 4.3, we can compute the comparison metrics for prediction involving a linear classifier on the cdc 15 data set as follows. The precision results are very low for this experiment. The other metrics values are average.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 67.89\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 32.11\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.54$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.70$$

$$\text{Precision} = (TP) / (TP + FP) = 0.20$$

4.2.3. Cdc 28 Set Using Linear Classify

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions at different time intervals for the cdc 28 cycle. After running the classify function in MATLAB 7.0, Table 4.4 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.4. Confusion matrix of cell cycle regulation gene presence in yeast synchronized by the cdc28 method for a linear classifier

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 225**	**FN = 159**
Cell-Cycle Regulated Gene Absent	**FP = 780**	**TN = 1926**

From Table 4.4, we can compute the comparison metrics for prediction involving a linear classifier on the cdc 28 data set as follows. The precision results are very low for this experiment. The other metrics values are average.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 69.6\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 30.4\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.58$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.71$$

$$\text{Precision} = (TP) / (TP + FP) = 0.22$$

4.2.4. Elu Set Using Linear Classify

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions at different time intervals for the elu cycle. After running the classify function in MATLAB 7.0, Table 4.5 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.5. Confusion matrix of cell-cycle regulation gene presence in yeast synchronized by the elu method for a linear classifier

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 228**	**FN = 156**
Cell-Cycle Regulated Gene Absent	**FP = 803**	**TN = 1903**

From Table 4.5, we can compute the comparison metrics for prediction involving a linear classifier on the elu data set as follows. The precision results are very low for this experiment. The other metrics values are average.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 68.96\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 31.04\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.59$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.70$$

$$\text{Precision} = (TP) / (TP + FP) = 0.22$$

4.2.5. Log p Using Linear Classify

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions which give the best results for log p at different time intervals. After running the classify function in MATLAB 7.0, Table 4.6 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.6. Confusion matrix of cell-cycle regulation gene presence in yeast synchronized using a linear classifier and log p analysis

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 144**	**FN = 240**
Cell-Cycle Regulated Gene Absent	**FP = 147**	**TN = 2559**

From Table 4.6, we can compute the comparison metrics for prediction involving a linear classifier and log p analysis as follows. The precision results are considerably better for this experiment as compared to those when not using log p analysis with a linear classifier. However, the sensitivity is low.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 87.47\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 12.53\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.38$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.94$$

$$\text{Precision} = (TP) / (TP + FP) = 0.49$$

4.2.6. Posterior Using Linear Classify

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions which give the best results for POST at different time intervals. After running the classify function in MATLAB 7.0, Table 4.7 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.7. Confusion matrix of cell-cycle regulation gene presence in yeast synchronized using a linear classifier and posterior analysis

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 270**	**FN = 115**
Cell-Cycle Regulated Gene Absent	**FP = 328**	**TN = 2378**

From Table 4.7, we can compute the comparison metrics for prediction involving a linear classifier and posterior analysis as follows. The precision results are considerably better for this experiment as compared to those when not using posterior analysis with a linear classifier. The sensitivity is better here too.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 85.69\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 14.31\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.70$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.88$$

$$\text{Precision} = (TP) / (TP + FP) = 0.45$$

4.2.7. Alpha Set Using Quadratic Classify

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions at different time intervals for the alpha cycle. After running the classify function with the type as quadratic in MATLAB 7.0, Table 4.8 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.8. Confusion matrix of cell-cycle regulation gene presence in yeast synchronized by the alpha method for a quadratic classifier

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 250**	**FN = 134**
Cell-Cycle Regulated Gene Absent	**FP = 248**	**TN = 2458**

From Table 4.8, we can compute the comparison metrics for prediction involving a quadratic classifier on the alpha data set as follows. The metrics are better with the quadratic classifiers as compared to when using linear classifiers.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 87.63\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 12.37\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.65$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.91$$

$$\text{Precision} = (TP) / (TP + FP) = 0.50$$

4.2.8. Cdc 15 Set Using Quadratic Classify

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions at different time intervals for the cdc 15 cycle. After running the classify function with the type as quadratic in MATLAB 7.0, Table 4.9 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.9. Confusion matrix of cell-cycle regulation gene presence in yeast synchronized by the cdc 15 method for a quadratic classifier

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 206**	**FN = 178**
Cell-Cycle Regulated Gene Absent	**FP = 141**	**TN = 2565**

From Table 4.9, we can compute the comparison metrics for prediction involving a quadratic classifier on the cdc 15 data set as follows. We get high accuracy and specificity, average precision and low sensitivity.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 89.67\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 10.33\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.54$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.95$$

$$\text{Precision} = (TP) / (TP + FP) = 0.59$$

4.2.9. Cdc 28 Set Using Quadratic Classify

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions at different time intervals for the cdc 28 cycle. After running the classify function with the type as quadratic in MATLAB 7.0, Table 4.10 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.10. Confusion matrix of cell-cycle regulation gene presence in yeast synchronized by the cdc-28 method for a quadratic classifier

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 243**	**FN = 141**
Cell-Cycle Regulated Gene Absent	**FP = 248**	**TN = 2458**

From Table 4.10, we can compute the comparison metrics for prediction involving a quadratic classifier on the cdc 28 data set as follows. We get high accuracy and specificity, lower precision and better sensitivity than before.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 87.40\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 12.60\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.63$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.91$$

$$\text{Precision} = (TP) / (TP + FP) = 0.49$$

4.2.10. Elu Set Using Quadratic Classify

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions at different time intervals for the elu cycle. After running the classify function with the type as quadratic in MATLAB 7.0, Table 4.11 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.11. Confusion matrix of cell-cycle regulation gene presence in yeast synchronized by the elu method for a quadratic classifier

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 150**	**FN = 234**
Cell-Cycle Regulated Gene Absent	**FP = 298**	**TN = 2408**

From Table 4.11, we can compute the comparison metrics for prediction involving a quadratic classifier on the elu data set as follows. We get high accuracy and specificity, but very low precision and sensitivity.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 82.78\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 17.22\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.39$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.89$$

$$\text{Precision} = (TP) / (TP + FP) = 0.33$$

4.2.11. Log p Using Quadratic Classify

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions which give the best results for log p at different time intervals. After running the classify function with the type as quadratic in MATLAB 7.0, Table 4.12 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.12. Confusion matrix of cell-cycle regulation gene presence in yeast by running a quadratic classifier and using log p analysis

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 147**	**FN = 237**
Cell Cycle Regulated Gene Absent	**FP = 145**	**TN = 2561**

From Table 4.12, we can compute the comparison metrics for prediction involving a quadratic classifier and log p analysis as follows. We get high accuracy, very high specificity, average precision and low sensitivity.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 87.63\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 12.37\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.38$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.95$$

$$\text{Precision} = (TP) / (TP + FP) = 0.50$$

4.2.12. Posterior Using Quadratic Classify

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions which give the best results for POST at different time intervals. After running the classify function with the type as quadratic in MATLAB 7.0, Table 4.13 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.13. Confusion matrix of cell-cycle regulation gene presence in yeast by running a quadratic classifier and using posterior analysis

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 269**	**FN = 116**
Cell-Cycle Regulated Gene Absent	**FP = 164**	**TN = 2542**

From Table 4.13, we can compute the comparison metrics for prediction involving a quadratic classifier and posterior analysis as follows. We get high accuracy, specificity, comparatively higher precision and sensitivity. We get the best results for sensitivity and precision here.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 90.97\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 9.03\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.70$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.94$$

$$\text{Precision} = (TP) / (TP + FP) = 0.62$$

4.2.13. Combined

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions for all the data sets combined together. After running the classify function with the type as quadratic in MATLAB 7.0, Table 4.14 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.14. Confusion matrix of cell-cycle regulation gene presence in yeast synchronized by the data combined together

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 183**	**FN = 73**
Cell-Cycle Regulated Gene Absent	**FP = 201**	**TN = 2633**

From Table 4.14, we can compute the comparison metrics for prediction involving a quadratic classifier on the data combined together as follows. We get better results here as compared to most experiments presented in above sections. However, we have a lower precision here.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 91.13\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 8.87\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.71$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.93$$

$$\text{Precision} = (TP) / (TP + FP) = 0.48$$

4.2.14. Support Vector Machine

In this experiment, the training set and the testing set contain microarray analysis results of gene expressions for all the data sets combined together. After running the support vector machine (svm) classifier function in MATLAB 7.0, Table 4.15 gives us an idea about the number of TP, TN, FP and FN that occurred.

Table 4.15. Confusion matrix of cell-cycle regulation gene presence in yeast synchronized by the svm classifier using all data together

Actual Result	Predicted Result	
	Cell-Cycle Regulated Gene Present	Cell-Cycle Regulated Gene Absent
Cell-Cycle Regulated Gene Present	**TP = 7**	**FN = 14**
Cell-Cycle Regulated Gene Absent	**FP = 7**	**TN = 165**

From Table 4.14, we can compute the comparison metrics for prediction involving a quadratic classifier on the data combined together as follows.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FN + FP) = 86\%$$

$$\text{Error Rate} = (FP + FN) / (TP + TN + FN + FP) = 14\%$$

$$\text{Sensitivity} = (TP) / (TP + FN) = 0.33$$

$$\text{Specificity} = (TN) / (TN + FP) = 0.92$$

$$\text{Precision} = (TP) / (TP + FP) = 0.33$$

To classify data using support vector machines, MATLAB provides two functions: svmtrain, which prepares the svm model by the training data, and svmclassify, which does

classification based on this training on test data. However, an issue with the svmtrain function in MATLAB is that it cannot handle a very large data set and that the memory runs out in such cases. Our data are also very huge, and the svmtrain function could not be run by using the entire data source. Therefore, the first 200 records were taken from the training data set used for other classifiers, and these 200 records were used for training with the two mentioned functions. The results provided in Table 4.15 are based on the 200 records. We see a very high specificity, but low sensitivity and precision. The prediction results that we obtained by using posterior analysis for classification have given the best results for the experiments conducted in this thesis.

4.3. Plots

In this experiment, the main plot that has been drawn for comparison purposes is the sensitivity vs specificity Plot. We discuss the plots below.

4.3.1. Sensitivity vs Specificity

The sensitivity vs specificity plot can also be called true positive rate (TPR) vs true negative rate (TNR). Higher specificity and sensitivity indicate better prediction. Hence, we have plotted the TPR vs TNR for four sets of tests to compare them. The four sets of tests and their corresponding plots are given in the next sections. We compare the plots in such a way that it is easier to see that using the statistical relevance measure and the proposed algorithm gives us a higher specificity and sensitivity than when the classifiers are used without them. The goal is to have the data points on the graph to farthest right corner of the first quadrant. That would mean aiming for a specificity and sensitivity close to 1.

4.3.1.1. Linear Classify with Data Sets Treated Separately

Figure 4.1 shows TPR vs TNR for the test results when a linear classifier is used on each data set separately.

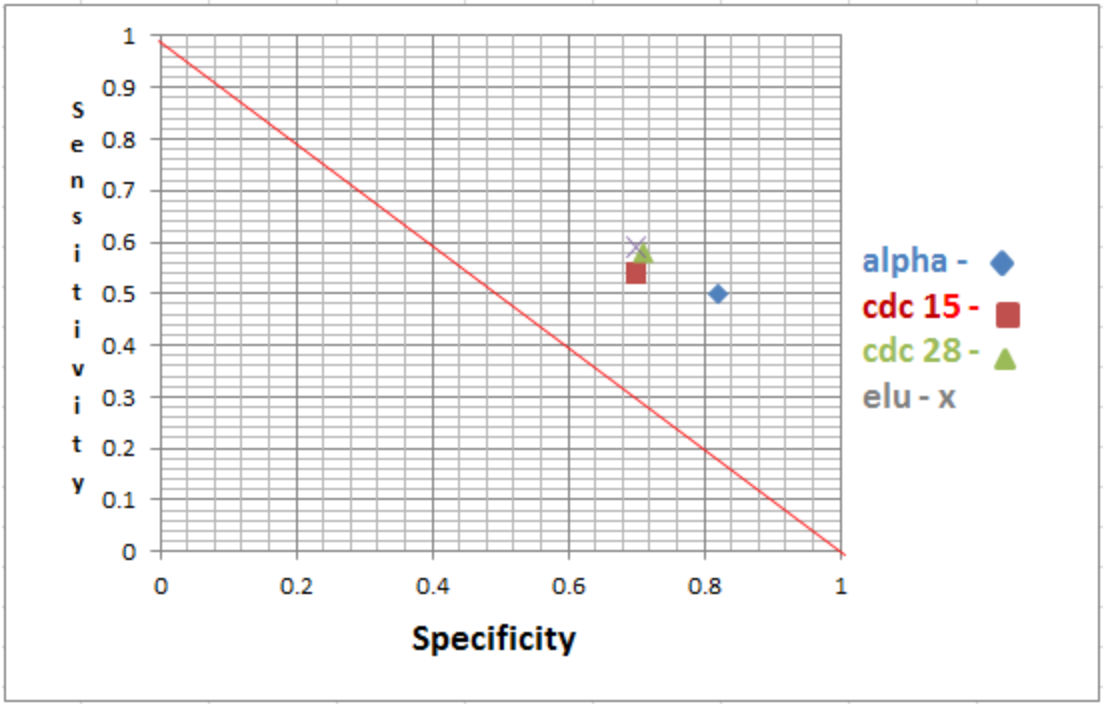

Figure 4.1. Specificity vs sensitivity graph for linear classification results. The X-axis represents the specificity measure of our predictions, and the Y-axis represents the sensitivity measure of our predictions.

The plot shows that the specificity and sensitivity are pretty average for predictions made using the linear classifier. While the alpha data set gives better specificity, it has lower sensitivity than the other three data sets. Cdc 15, cdc 28 and elu data-set prediction results are very close to each other and can been seen in the plot.

4.3.1.2. Quadratic Classify with Data Sets Treated Separately

Figure 4.2 shows TPR vs TNR for the test results when a quadratic classifier is used on each data set separately. It can be interpreted that the test results are better in this case compared to using the linear classifier for the same data sets. The data points appear to have shifted towards more right as compared to the data points on Figure 4.1 implying better specificity.

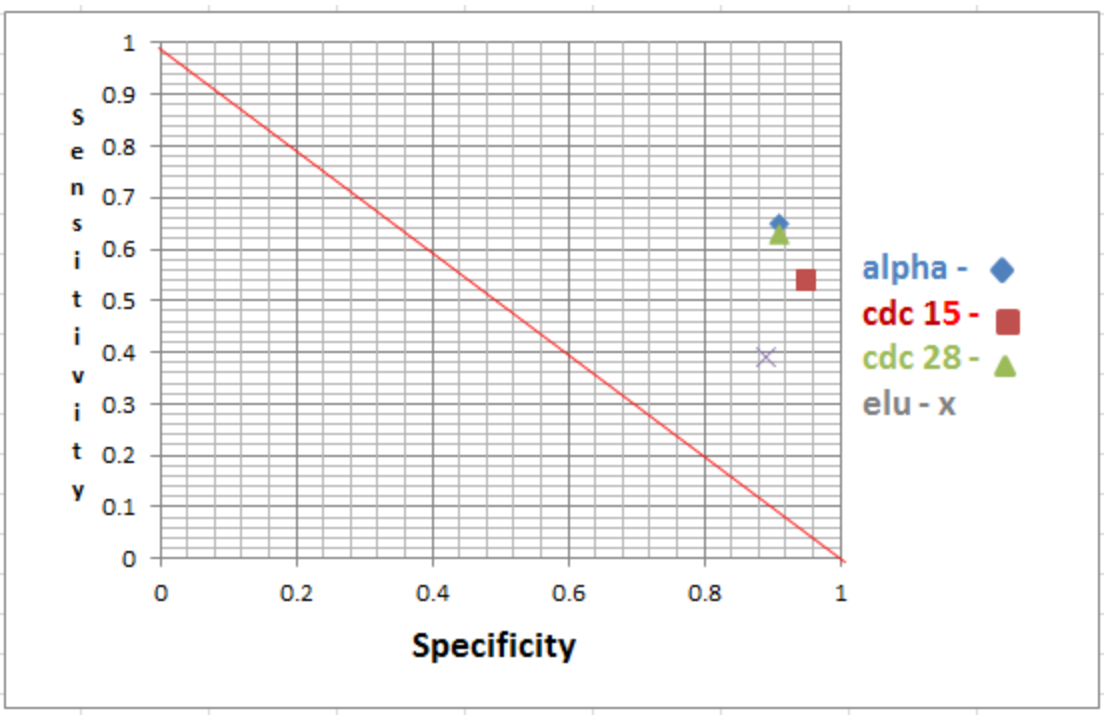

Figure 4.2. Specificity vs sensitivity graph for quadratic classification results. The X-axis represents the specificity measure of our predictions, and the Y-axis represents the sensitivity measure of our predictions.

Figure 4.2 shows better plots as compared to Figure 4.1 which means that, in general, we obtain higher specificity and sensitivity while using a quadratic classifier. All data sets show considerable improvement for the prediction specificity in this plot. While the alpha data set gives the highest sensitivity, the cdc 15 data set gives the highest specificity, and the elu data set has the lowest prediction results.

4.3.1.3. Linear Classify Using Log p and Posterior

Figure 4.3 shows TPR vs TNR for the test results when a linear classifier is used after finding the best classifier for the log p and posterior predictions. It can be interpreted from the plots that the test results are better using log p and posterior (POST) despite using a linear classifier.

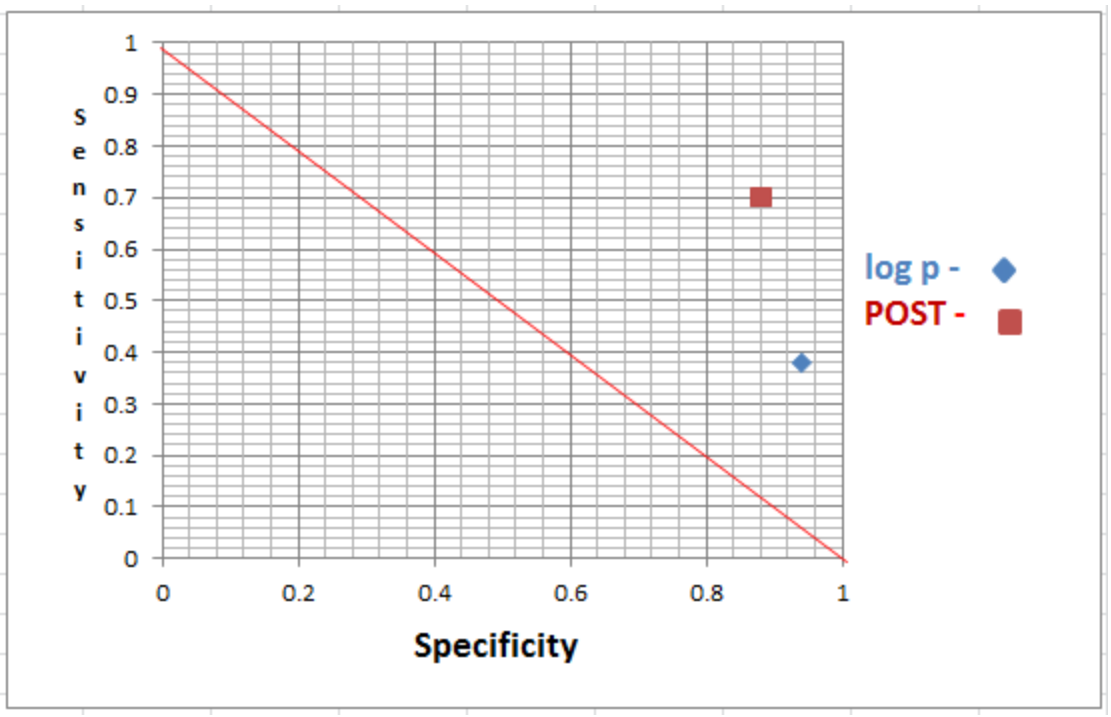

Figure 4.3. Specificity vs sensitivity graph for linear classification results using log p and POST. The X-axis represents the specificity measure of our predictions, and the Y-axis represents the sensitivity measure of our predictions.

We have used two types of statistical relevance measure, introduced in Chapter 1, in this thesis, and those measures are log p and posterior analysis. Both measures are used for a linear classifier as well as a quadratic classifier. We see very good improvement in the specificity for both log p and posterior analysis. However, sensitivity is lower for log p analysis as opposed to when using the data sets separately. From Figure 4.3, we could, however, conclude that for a linear classifier, when using posterior analysis, we get better specificity as well as sensitivity. Using our data sets separately for both the classifiers, we have noticed that the sensitivity is always much lower. Figure 4.3, however, shows a very good (0.70) sensitivity when the most predictive data source is utilized by using the posterior analysis measure.

4.3.1.4. Quadratic Classify Using Log p and Post

The following plot shows TPR vs TNR for the test results when a quadratic classifier is used after finding the best classifier for log p and posterior predictions. It can be interpreted that the test results are much better in this case compared to a using linear classifier or a quadratic classifier separately for each data set.

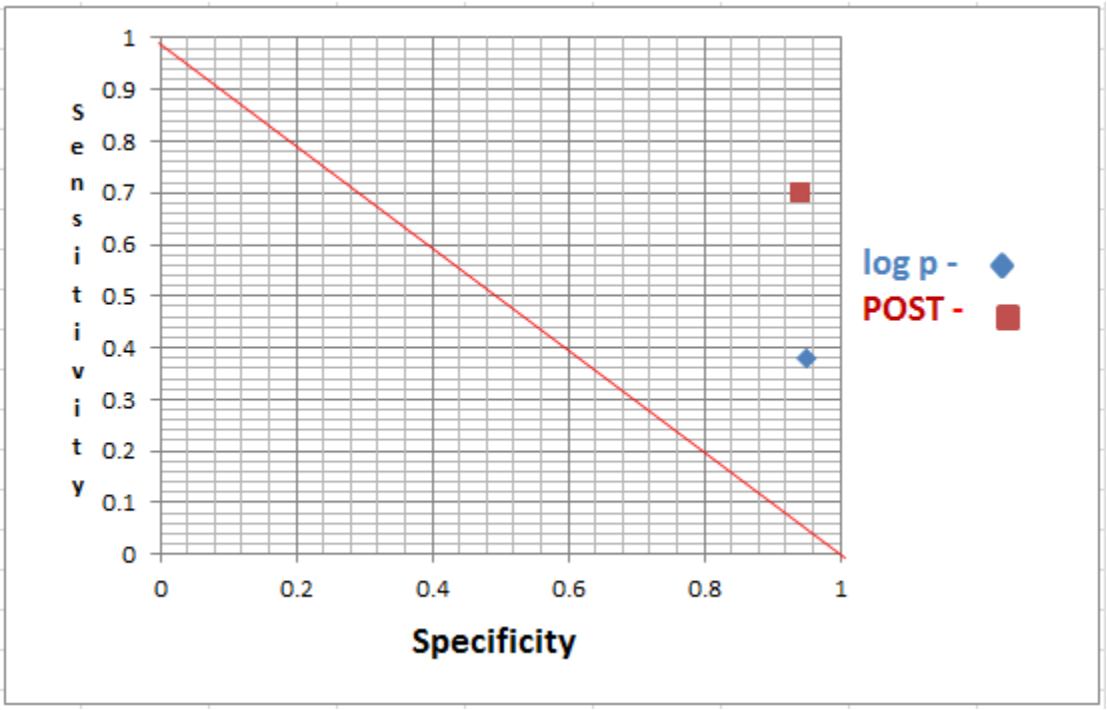

Figure 4.4. Specificity vs sensitivity graph for quadratic classification results using log p and POST. The X-axis represents the specificity measure of our predictions and the Y-axis represents the sensitivity measure of our predictions.

Comparing Figure 4.2 where we use a quadratic classifier with the data sets separately and Figure 4.4 where we utilize the most predictive data set by using first the log p analysis measure and then the posterior analysis measure, we can say that we, again, have comparatively better specificity for both log p and posterior analysis. We, however, get lower sensitivity while using log p analysis but the best sensitivity (in comparison to Figure 4.2) when we use posterior

analysis. From what we concluded in Section 4.3.1.3 and from what we see in Figures 4.2 and 4.4, we can say that, irrespective of the two classifiers (linear and quadratic), we can find the most predictive data source when we use the posterior analysis measure. Using the corresponding data set for our predictions, we obtain very high specificity and sensitivity, which is, on average, higher than what we get when we obtain the data sets separately. We have more discussions about the results and plots presented in this chapter in the next chapter.

CHAPTER 5. DISCUSSIONS AND FURTHER WORK

In this chapter, we go in depth with the comparison between the various results that we obtained. This project's main objective was to find the best predictive data source. For that, various prediction techniques were used so that comparisons could be made on more than one level. In this thesis, we predicted using linear and quadratic classifiers with different data sets separately once without using log p or posterior analysis and once using log p and posterior analysis. The comparison metrics helps us to understand whether, using log p or posterior analysis, we are in better shape to find the most predictive data source.

5.1. Result Analysis

In this experiment, it was pretty much expected that the quadratic classifier would produce better prediction results than the linear classifier. However, the real challenge was to see how we could use the log p and posterior analysis results in our predictions and to find the most predictive data source for each prediction. After finding the most predictive data source by virtue of the proposed algorithm, we used those results against the ones we already had. The results are given in Tables 5.1 and 5.2.

Table 5.1. Results at a glance for linear classification using the data sets separately

Linear	Error rate	Accuracy	Specificity	Sensitivity	Precision
Alpha	21.70%	78.20%	0.82	0.50	0.29
Cdc 15	32.11%	67.89%	0.70	0.54	0.20
Cdc 28	30.04%	69.60%	0.71	0.58	0.22
Elu	31.04%	68.96%	0.70	0.59	0.22
Log p	12.53%	87.47%	0.94	0.38	0.49
POST	14.31%	85.69%	0.88	0.70	0.45

Table 5.2. Results at a glance for quadratic classification using data sets separately

Quadratic	Error rate	Accuracy	Specificity	Sensitivity	Precision
Alpha	12.37%	87.68%	0.91	0.65	0.50
cdc15	10.33%	89.67%	0.95	0.54	0.59
cdc28	12.60%	87.40%	0.91	0.63	0.49
Elu	17.22%	82.78%	0.89	0.39	0.33
Log p	12.37%	87.63%	0.95	0.38	0.50
POST	9.03%	90.97%	0.94	0.70	0.62

As we see, the accuracy, the specificity, the sensitivity and the precision are higher when using posterior analysis for both set of experiments: quadratic classification and linear classification. Using log p analysis, we obtain better results for accuracy, specificity and precision but not for sensitivity.

5.2. Log p and Posterior

The final comparison in results to find the most predictive data source would be using log p and posterior values in the classifiers. Tables 5.3 and 5.4 would give us the comparisons.

Table 5.3. Results at a glance for linear classification using log p and posterior analysis

Linear	Error rate	Accuracy	Specificity	Sensitivity	Precision
Using log p	12.53%	87.47%	0.94	0.38	0.49
Average without using log p or POST	28.81%	71.16%	0.73	0.55	0.23
Using posterior	14.31%	85.69%	0.88	0.70	0.45

Table 5.4. Results at a glance for quadratic classification using log p and posterior

Quadratic	Error rate	Accuracy	Specificity	Sensitivity	Precision
Using log p	12.37%	87.63%	0.95	0.38	0.50
Average without using log p or POST	13.13%	86.88%	0.91	0.55	0.48
Using posterior	9.03%	90.97%	0.94	0.70	0.62

Tables 5.3 and 5.4 compare results of three predictive techniques: (i) using log p; (ii) using a classifier without using log p or posterior; this value is an average of the values used in tables 5.1 and 5.2 for linear and quadratic classifiers respectively; and (iii) using posterior. In case of both linear and quadratic classifiers, using log p has better specificity and precision, but lower sensitivity. When using log p and posterior, predicting True Negative has been better than predicting True Positives for both sets of experiments. Using log p along with the classifier has given a lot of False Negatives, i.e., this technique has not been able to predict the genes with class label 1 as compared to when not using this technique.

However, using posterior analysis, we get better results with every comparison metric. While using log p gives high accuracy, specificity and precision, using posterior gives us the highest sensitivity. While the specificity has been above 70% for all of the tests with the best being a very high 94% (using log p analysis), predicting high sensitivity has been a challenge throughout with the average being 55% (results without using log p or posterior). While predicting the True Negatives is important, the real challenge is to correctly predict the True Positives. The reason for this challenge is that the total number of genes used in the test set is 3090, from which 388 genes have class label 1. Only 12.5% of the total genes are a class label 1,

which is a small number. This reason is why predicting the True Positives is much more challenging than predicting the True Negatives. Hence, considering the kind of data set that we used, a high sensitivity indicates a very good prediction. In the case of using posterior analysis on a quadratic classifier, the sensitivity is 70%, the highest that has been seen until now. This technique correctly predicts 269 class label 1 genes from 384 genes, the best we got. Also using posterior analysis with quadratic gives very high accuracy, specificity and precision even though they were not the best.

The reason why using posterior probability and log p calculation gave us better results was explained briefly in section 2.4.5. Using these two measures helped us to narrow down the best predictive data set. In this thesis, it has been repeated that a big challenge was handling multiple data sets. While, in case of certain genes, using one data set may have given a correct prediction, utilizing another may have given an incorrect one. By using the two measures, the data set that had a higher probability of a better prediction was identified and was used for final prediction results. While, earlier, we would predict say, gene x and gene y, using the same data set, say alpha, now as per the matrices given by the posterior probability calculations, we could be using alpha for gene x and elu for gene y simply because alpha gave highest posterior probability for gene x and elu gave the highest for gene y.

In Chapter 4, we discussed the F1 measure as a measure of quality of the prediction. In Table 5.5 we compare the F1 measures of the three main prediction techniques that we tested in this thesis. Even in this comparison, we see that using the posterior analysis on the quadratic classifier gives the highest F1 measure. As mentioned before, our data have very low number of genes which participate in cell-cycle regulation or have class label 1. So predicting the sensitivity for this thesis has been a greater challenge than predicting the specificity. F1 measure involves

precision and sensitivity and posterior analysis results giving the best F1 measure as compared to when not using it proves that we have, indeed, been able to find the most predictive data source and have used it for better results.

Table 5.5. Comparison between various prediction techniques with respect to the F1 measure

Using Quadratic	F1 Measure
All test samples using log p	0.43
Average without using log p and posterior	0.51
All test samples using posterior	0.66

5.3. Conclusions and Further Work

From the objectives that we summarized at the end of Section 1.1, we can say that we have been able to complete each of them. The results presented in Chapter 4 talk about how we went about performing tasks for the objectives (i), (ii) and (iii) where we classified and predicted class label for genes present in test data by using linear and quadratic classifiers; then computed a statistical relevance measure and by using the proposed algorithm predicted the most reliable data source for each gene. The comparison of results between the two methods mentioned as objective (iv) in Section 1.1 have been listed in Tables 5.1 through 5.5.

After completing all the tasks with respect to the objectives of the thesis, we can conclude the following: (i) using posterior analysis we are more successful in predicting the most reliable data source when there are multiple datasets involved, (ii) real challenge is to correctly predict the True Positives: the total number of genes used in the test set is 3090, from which 384 genes have class label 1. In the case of using posterior analysis on a quadratic classifier, the sensitivity is 70%, the highest that has been seen until now, (iii) this technique, i.e., using posterior analysis,

correctly predicts 269 class label 1 genes from 384 genes, the best we got whereas on an average using the data sets separately correctly predicts about 211 genes, and (iv) F1 measure is 0.62 when using posterior analysis as opposed to when using the data sets separately, the average F1 measure is 0.5.

Referring back to the end of Section 1.1, where we talk about the significance of our finding, which is being able to find the most predictive data source in the given biological data and subsequently find the biological significance of this finding which is the phase in which the regulation has most likely happened for a gene. The algorithm that is proposed in this thesis can now be taken back to the biologists and applied in such kinds of experimental data where multiple subsets of data have a biological significance. The algorithm in such a case is successful in finding the most reliable data source for each record and its corresponding significance for the record.

The thesis demonstrated the benefit of using multiple data sets in the prediction of protein function. Analysis on further data sets and comparison with other alternative approaches would further establish it as an important data mining approach for addressing classification based on multiple data sets.

REFERENCES

[1] Tan, P. N., Steinbach, M., and Kumar, V. 2006. Classification. *Introduction To Data Mining*. Pearson Addison Wesley. Boston, MA.

[2] Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein D., and Futcher, B. 1998. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9(12):3273-3297.

[3] Demeter, J., Beauheim, C., Gollub, J., Hernandez-Boussard, T., Jin, H., Maier, D., Matese, J. C., Nitzberg, M., Wymore, F., Zachariah, Z. K., Brown, P. O., Sherlock, G., and Ball, C. A. 2007. The Stanford Microarray Database: Implementation of new analysis tools and open source release of software. *Nucleic Acids Research.* 35(Database Issue):D766-770.

[4] Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467-470.

[5] Fisher, R. A. The statistical utilization of multiple measurements. 1938. *Annals of Eugenics*, 8, 376–386.

[6] Cover, T., Hart, P. 1967. Nearest neighbor pattern classification. Information Technology, *IEEE Transactions*, 13(1):21-27.

[7] Kuncheva, L. I.. 2004. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience. Hoboken, NJ.

[8] Kuncheva, L. I., Bezdek, J. C., and Duin, R. P. W. 2001. Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition*, 34:299-314.

[9] Mangai, U., Samanta, S., Das, S., and Chowdhury, P. 2010. A survey of decision fusion and feature fusion strategies for pattern classification. *IETE Technical Review*, 27:293-307.

[10] Kumar, T. P., Hasegawa, Y., and Iba, H. 2006. Classification of gene expression data by majority voting genetic programming classifier. In *IEEE World Congress on Computational Intelligence*, pages 8690-8697.

[11] Lam, L., and Suen, C. 1997. Application of majority voting pattern recognition: An analysis of its behavior and performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):553-568.

[12] Xu, L., Krzyzak, A., and Suen, C. 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):418-435.

[13] Wang, Y., Dunham, M. H., Waddle, J. A., and Mcgee, M. 2006. Classifier fusion for poorly differentiated tumor classification using both messenger rna and microrna expression profiles. *Proceedings of the 2006 Computational Systems Bioinformatics Conference* (CSB 2006), Stanford, California.

[14] Zheng,N., Ching, P. C., Wang, N., and Lee, T. 2006. Integrating complementary features with a confidence measure for speaker identification. *International Symposium on Chinese Spoken Language Processing*, pages 549-557.

[15] Huenupán, F., Yoma, N. B., Molina, C., and Garretón, C. 2008. Confidence based multiple classifier fusion in speaker verification. *Pattern Recognition Letters*, 29(7):957-966.

[16] Edgard, N. 1996. Evaluation of pattern classifiers: testing the significance of classification efficiency using exact probability technique. *Pattern Recognition Letters*, 17(11):1125-1129.

[17] Edgard, N. 1998. Evaluation of pattern classifiers: applying a monte carlo significance test to the classification efficiency. *Pattern Recognition Leters.*, 19(1):1-6.

[18] Lee, G.N., and Bottema, M. J. 2006. Significance of classification scores subsequent to feature selection. *Pattern Recognition Letters*, 27(14):1702-1709.

[19] Klecka, W. R. *Discriminant Analysis*. 1980. Wiley-Interscience Publication. Hoboken, NJ.

[20] Li, T., Zhu, S., and Ogihara, M. 2006. Using discriminant analysis for multi-class classification: an experimental investigation. *Knowledge Information System*, 10(4):453-472.

[21] Marti'nez, A. M., and Kak, A. C. 2001. Pca versus Ida. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228-233.

[22] Maclachlan, G. J. 2004. *Discriminant Analysis and Statistical Pattern Recognition (Wiley Series in Probability and Statistics).* Wiley-Interscience. Hoboken, NJ.

[23] Fauvel, M., Chanussot, J., and Benediktsson, J. 2006. A combined support vector machines classification based on decision fusion. *IEEE International Geoscience and Remote Sensing Symposium*, pages 2494-2497.

[24] Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S. et al. 2001. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America*, 98:15149-54.

[25] Nalbantov, G., Bioch, J., and Groenen, P. 2010. Classification with support hyperplanes. Econometric Institute Report EI 2006-42, Erasmus University Rotterdam, Econometric Institute.

[26] Joachims, T. 2005. A support vector method for multivariate performance measures. In ICML *'.05:Proceedings of 22nd international conference on Machine learning*, pages 377-384, ACM, New York, NY, USA.

[27] Shafer, G and Vohk, V. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371-421, 2008.

[28] Moed, M., Smirnov, E. N. 2009. Efficient adaboost region classification. *MLDM '09: Proceedings of the 6th International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 123-136, Berlin, Heidelberg, Springer-Verlag.

[29] Schleif, F. M., Villmann, T., Kostrzewa, M., Hammer, B., and Gammerman, A. 2009. Cancer informatics by prototype networks in mass spectrometry. *Artificial Intelligence Medicine*, 45(2-3):215-228.

[30] Diaz-Uriarte, R., and Alvarez de Andres, S. 2006. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*. 7:3.

[31] Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., and Davis, R. W. 1998. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2:65-73.

[32] Krzanowski, W. J. 1988. *Principles of Multivariate Analysis: A User's Perspective*. New York: Oxford University Press.

[33] Seber, G. A. F. 1984. *Multivariate Observations*. John Wiley & Sons, Inc., Hoboken, NJ.

[34] Hsiao, A., Worrall, D. S., Olefsky, J. M., Subramanium, S. 2004. Variance-modeled posterior inference of microarray data: detecting gene-expression changes in 3T3-L1 adipocytes. *Bioinformatics*. Vol. 20 no 17 2004 3108-3127.

[35] Long, A. D., Mangalam, H. J., Chan, B. Y. P., Tolleri, L., Hatfield, G.W. and Baldi, P. 2001. Improved statistical inference from DNA microarray data using analysis of variance and a Bayesian statistical framework. Analysis of global gene expression in Escherichia coli K12. *Journal of Biological Chemistry*, 276, 19937-19944.

[36] Sherlock, G., Hernandez-Boussard, T., Kasarskis, A., Binkley, G., Matese, J. C., Dwight, S. S., Kaloper, M., Weng, S., Jin, H., Ball, C. A., Eisen, M. B., Spellman, P. T., Brown, P. O., Botstein, D., and Cherry, J. M. 2001. The Stanford Microarray Database. *Nucleic Acids Research.* 29(1):152-155.

[37] Chen, K. C., Calzone, L., Csikasz-Nagy, A., Cross, F. R., Novak, B., and Tyson, J. J. 2004. Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell*, 15:3841-3862.

[38] Amon, A., Irniger, S., and Nasmyth, K. 1994. Closing the cell cycle circle in yeast: G2 cyclin proteolysis initiated at mitosis persists until the activation of G1 cyclins in the next cycle. *Cell*, 77:1037-1050.

[39] Yeung, K. Y., and Bumgarner, R. E. 2003. Multiclass classification of microarray data with repeated measurements: application to cancer. *Genome Biology*, 4:R83.

[40] Bevilacqua, V., Mastronardi, G., Menolascina, F., Paradiso, A., and Tommasi, S. 2006. Genetic algorithms and artificial neural networks in microarray data analysis: A distributed approach. *Engineering Letters* 13:3.

[41] Brown, M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares~Jr, M., and Haussler, D. 2000. Knowledge-based analysis of microarray gene expression data by using Support Vector Machines. *Proceedings of National Academy of Sciences, USA*. 97(1):262-267.

[42] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. 1999. Molecular classification of cancer: Class discovery and class prediction by gene epression monitoring. *Science*. 286:531-537.