Most Informative Vector Selection Using Active Learning

Submitted By

Maryam Razzaq
641/FBAS/MSCS-F10

Supervised by:

Ms. Zareen Sharf
Assistant Professor

Department of Computer Science
Faculty of Basic and Applied Sciences
International Islamic University Islamabad
2014

Size
-300dpi
-OCR
-Smallpdf.

Vector processing

Arry processing

Department of Computer Science & Software Engineering

International Islamic University Islamabad

Date : ________________

Final Approval

This is to certify that we have read the thesis submitted by **Maryam Razzaq, Reg # 641-FBAS/MSCS/F10**|. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree of **MSCS.**

Committee:

External Examiner:

Dr. Rafi Us Shan
Assistant Professor
Department of Computer Science
Comsats Institute of Information Technology
Abbotabad

Internal Examiner:

Umara Zahid
17/9/14

Ms. Umara Zahid
Lecturer
Department of Computer Science & Software Engineering
International Islamic University
Islamabad

Supervisor:

Ms. Zareen Sharf
Assistant Professor
Department of Computer Science & Software Engineering
International Islamic University
Islamabad

بسم الله الرحمن الرحيم

DEDICATION

I would like to dedicate my research work to the
Creator and most merciful
ALMIGHTY ALLAH

HOLIEST man Ever Born on Earth,
PROPHET MUHAMMAD (Peace Be Upon Him)
And

I also dedicate my work to my

PARENTS

Whose sincere love and prayers were a source of strength for me and enable me to accomplish this research work successfully

Maryam Razzaq

Reg # 641/FBAS/MSCS-F10

A dissertation Submitted To

Department of Computer Science,

Faculty of Basic and Applied Sciences,

International Islamic University, Islamabad

As a Partial Fulfillment of the Requirement for the Award of the Degree of *MSCS*.

Declaration

I hereby declare that this Thesis ***"Most Informative Vector Selection Using Active Learning"*** neither as a whole nor as a part has been copied out from any source. It is further declared that I have done this research with the accompanied report entirely on the basis of my personal efforts, under the proficient guidance of my teachers especially my supervisor ***Ms. Zareen Sharf.*** If any part of the thesis is proved to be copied out from any source or found to be reproduction of any work from any of the training institute or educational institutions, I shall stand by the consequences.

Maryam Razzaq
Reg # 641/FBAS/MSCS-F10

ACKNOWLEDGMENT

First of all I am Thankful to **Almighty Allah,** the most Merciful the most Beneficent, who has blessed me with the knowledge and courage to get my work done. This thesis is in present form due to the guidance and assistance of several people. I would like to offer my special thanks to all of them.

I express my sincere gratitude to my parents **Mr. & Mrs. Abdur Razzaq** for always praying for me and always being there for motivation whenever I get into problems. A special thanks to them for their efforts for providing me a good education. I am also grateful to my **siblings** for their motivation, support and prayers throughout the course of this work.

I would like to express my deepest gratitude to **Mrs. Zareen Sharf,** my supervisor, her thoughtful guidance and critical comments has shown me new perspectives of doing my work. Her utmost moral support has always been a source of encouragement for me.

Last but definitely not the least, I would like to say thanks to my **Colleagues, Class fellows, Seniors** and **friends** who have always inspired me to do better and prove myself.

Maryam Razzaq

Reg # 641/FBAS/MSCS-F10

Abstract

The machine learning algorithms tends to ease out the human effort for labelling huge volumes of data. The core idea is to train those algorithms so well that they can predict the labels of data that comes to them. These algorithms are the part of Supervised and semi-supervised learning. The traditional supervised learning techniques learn from the labeled examples but in all those cases the learning examples are in hundreds of number. Active Learning which is also called an extension to the semi-supervised learning systems minimizes this selection of training data in a very little amount of examples, the number of examples for training are although very small but they are the chosen ones which can count up for a better systems accuracy. In this thesis a new approach of active learning has been applied in which the most informative vectors have been selected with the help of pre-processing of data via divisive analysis (DIANA) algorithm. The comparison of proposed method has been made with the active learning approach that used version space's concept of general to specific ordering for pre-processing. The version space concept is replaced with the divisive analysis (DIANA) algorithm and the core idea is to pre-cluster the instances before distributing them into training and testing data. The results obtained by our system have justified our reasoning that a bit of pre-clustering instead of the traditional version space algorithm can bring a good impact on the accuracy of the overall system's classification. Two types of data has been tested, the binary class and multi-class. The proposed system worked well on the multi-class but in case of binary the version space algorithm maintained an edge.

TABLE OF CONTENTS

ACRONYMS

SL	Supervise Learning
SSL	Semi-Supervise Learning
AL	Active Learning
VS	Version Space
UKL	User Knowledge Level Dataset
ALVS	Active Learning with Version Space
ALDIANA	Active Learning with DIANA
MIV	Most Informative Vector
SVM	Support Vector Machine
MSVM	Multi Support Vector Machine
QBC	Query By Committee
EHR	Electronic Health Records

LIST OF FIGURES

LIST OF TABLES

CHAPTER # 01

INTRODUCTION

Machine Learning plays a vital rule in the disciplines that are related to artificial intelligence. It can be simply defined as a procedure which makes the computers so intelligent that they can assist the human in some of the much difficult and time taking tasks like i.e. decision making etc. the main distinguishing factor of machine learning from artificial intelligence is that it has a directed link with the data and the patterns that are drawn and learnt from that data. Now a days we are living in the live storms of information that keeps on exploding the systems on daily basis, the tweets on social networks, the documents on popular blogs, the threats on networks, the daily publications, news feeds etc. are some examples of data that keeps on getting stored on the information systems on daily basis.

The need of the hour is to make our information systems so intelligent that they can extract the best out of this newly coming information instead of throwing it into crash after some time period. Just take an example of disease discovery which might seems easy for some people but actually it is not. A particular disease has certain sets of symptoms and after effects but this information can get updated if we intelligently extract out the new and unique information from the data which is coming from the patient's history. A human expert can end up to nothing in many cases especially when he is confronted with a plethora of information provided to him as its beyond his capabilities to label and analyze all the data provided. The problem of disease diagnosis and many other such problems creates the need of an intelligent machine that can learn from the previous examples and on the basis of that learning it labels the new data no matter what is the volume of the new data.

Active Learning is the procedure or in simple words a way out for the intelligent information systems, it is that branch of machine learning algorithms which asks questions and then learn from those questions. The key notion of active learning is to decrease the human effort and reduce the time as well as cost for labeling the new data. The significant hypothesis behind the overall concept of active learning is that, if you allow the system to choose data from where it will learn, the performance of overall system will eventually rise.

The traditional supervised learning techniques also learn from the labeled examples but in all those cases the learning examples are in hundreds of number and was quite a task for the human experts especially when the chief concern is labeling of data. Active Learning systems minimizes this selection of training data in some few amount of examples, the no. of examples for training are although very small but they are the chosen ones which can count up for a better systems accuracy. In general form an active learner is categorized as a one which frequently keeps on asking the queries, receiving the responses and then increase its learning from those responses. The main difference between an active learner and a passive learners is that the active learner is the one which selectively takes the data for its classification.

The concept of both these learners can be best understood with the help of an example of two students, one is an active learner and the other one is an active learner. A passive learner is the student that keeps on gathering the information by just listening to the teacher and the active learner is the one that keeps on asking the questions and thus improving his learning via the queries in his mind.

In active learning the criteria is well defined on the basis of which the learner will ask queries and will get the answers. This concept of active learning can be applied into a variety of domains i.e. document classification, fraud detection, disease diagnosis etc. The most important to be taken care of this is the selection criteria. The pool based setting can be best applied for the selection criteria. The pool based active learning is basically a type of active learning in which we are given a large pool of unlabeled data and on the basis of some criteria the instances are selectively chosen from the pool and are presented to the human expert as queries which are then labeled.

Just take an example of disease discovery which might seems easy for some people but actually it is not. A particular disease has certain sets of symptoms and after effects but this information can get updated if we intelligently extract out the new and unique information from the data which is coming from the patient's history. A human expert can end up to nothing in many cases especially when he is confronted with a plethora of information provided to him as its beyond his capabilities to label and analyze all the data provided. The problem of disease diagnosis and many other such problems create the need of an intelligent machine that can learn from the previous examples and on the basis of that learning it labels the new data no matter what is the volume of the new data.

Active Learning is the procedure or in simple words a way out for the intelligent information systems, it is that branch of machine learning algorithms which asks questions and then learns from those questions. The key notion of active learning is to decrease the human effort and reduce the time as well as cost for labeling the new data. The significant hypothesis behind the overall concept of active learning is that, if you allow the system to choose data from where it will learn the performance of overall system will eventually rise.

The traditional supervised learning techniques also learn from the labeled examples but in all those cases the learning examples are in hundreds of number and were quite a task for the human experts especially when the chief concern is labeling of data. Active Learning systems minimizes this selection of training data in some few amount of examples, the no. of examples for training are although very small but they are the chosen ones which can count up for a better systems accuracy. In general form an active learner is categorized as a one which frequently keeps on asking the queries, receiving the responses and then increases its learning from those responses. The main difference between an active learner and a passive learner is that the active learner is the one which selectively takes the data for its classification.

The concept of both these learners can be best understood with the help of an example of two students, one is a passive learner and the other one is an active learner. A passive learner is the student that keeps on gathering the information by just listening to the teacher and the active learner is the one that keeps on asking the questions and thus improving his learning via the queries in his mind. Now the chief difference of learning would be the level of information received, both learners have received the information but the passive learner is getting whatever is coming to him while the active learner is frequently asking the question by which he is improving his level of information gained.

From the information provided above, this thing has got clear that in the active learning and passive learning phenomenon, the first step if to gather data/ information. The passive learner randomly perform the sampling of the gathered information by keeping in mind the population distribution and then it consults the classifier for the output which could come as a very time consuming task. The active learner gathers the information/data by asking the queries to the world then it goes to the classifier for carrying the task for which it is has been used.

Let's take an example of document classification, the learner may get a large collection of unlabeled documents, the passive learner will randomly pick up some documents for training its classifier and will ask the human expert to manually label it. Now the manual labeling of that learning data could be expensive not in terms of cost that the labeler will take but also in terms of the time that the labeler will take in labeling that data. The same task when given to the active learner, the learner will carefully choose the documents that are required to be labeled and then counted up as learning data.

Another approach of this thesis is going some levels beyond the active learning classification as the active learning technique has previously been applied in vast domains of classification as well as clustering. My point of focus in this thesis is the new approach into the version space which is implemented with the help of multilevel clustering and also classification. The traditional version space searching goes for two extreme hypothesis for training the classifier, one hypothesis is completely general and other is completely specific. In this thesis I am going to create a logical grouping of the whole pool of data rather than dividing it by the version space hypothesis.

1.1 Motivation

The main motivation of this work is to bring a new technique in to the querying process of the active learning system. As it has been stated in the previous section that active learning querying process can be carried out in a number of different approaches. It is actually the selection of training data which eventually brings a positive change into the performance of overall active learning system. The previous approach which I am going to focus mainly in this thesis is the concept of version space. The chief motivation is to change the selection hypothesis of version space from GS ordering into a more logical form of grouping which can be provided with the help of multilevel clustering of data.

In version space the learner first assume that all data is general/negative and this negative data goes for training the classifier, in the second step all the data is assumed to be positive/specific and this complete positive data is used for training the classifier. On the basis of these two classifiers the final labels of the testing data are decided. This training of classifier is actually done on the basis of two extreme assumptions and there is no logical reasoning behind the grouping of all this pool of data.

In the light of previous approach of active learning I am going to bring a technique of divisive analysis into it. The DIANA clustering algorithm has been used in this thesis for dividing the whole pool of data into as much groups as possible and then from each of that group/cluster we can get a member for labeling. The initial training data will now contain the members from all the possible groups of data provided and thus we'll get a member from the diverse division of data

1.2 Goals and Challenges

The Foremost goal of this thesis is to minimize the user burden by decreasing the user burden as much as possible without scarifying the performance or accuracy of the system. In order to achieve this goal I have introduced the concept of pre-clustering into the previous active learning system and with that I have also brought the validation technique which gives this assurance that the displayed accuracy is 100 percent correct.

The pool of data as the name suggests could be called as the ocean of information as it contains the members from all the information groups. It is quite a task to evaluate how many groups we are having in our data or to how much amount of groups should we stop our algorithm for further clustering. In the version space searching it pretty simple to divide the data under two hypothesis but when we start dividing the data we have to check for the centroid distribution of data and there are also chances that the random sampling result of traditional approach comes better than this new technique.

For example if we are working on disease diagnosis, we have to give labels i.e. the patients has the disease and the patient has not this disease. The random training data of version space might be containing the sufficient amount of data for training the classifier and the classifiers gives an output with good accuracy thus a praise able systems performance but the point of focus is that it's all based on a random selection and may be the next time this algorithm randomly chooses the data which decreases the whole systems performance. Thus the required goal here in this system is to create a system which gives the good performance in all the situations.

1.3 Contribution

In this thesis I am going to give the following contributions.

- The first contribution of this thesis is the development of new approach for cost sensitive active learning. I have shown that the labeling time taken by the human expert can be reduced if we give him as minimum amount of training data as possible. The minimum amount of training data is also enhanced by introducing the concept of pre-clustering before getting into the main classification.
- The second contribution is to bring the concept of multi-view learning into the previous binary learning system. In my work I have introduced a multi-view learner with the DIANA clustering algorithm. The benefit of using DIANA is that it first divide the data into two main clusters and then in the next levels it perform sub clustering of the primary two clusters and thus it can go to as much number of information groups as possible. The previous approaches were working on the two classes only and if some data containing more than two classes is introduced then the performance of overall system falls. Thus this work has provided the freedom of classes, it can work well for the binary class and also for more than two classes.
- The Third approach is the concept of validation, after all the classification of the system I have compared those labels which are provided by the classifier on the testing data with the original labels. This comparison of assigned labels and original labels thus validates the performance of overall system.

1.4 Thesis Outline

The remainder of this thesis is organized as follows.

Chapter2 will provide a brief introduction of the terms and techniques that I am going to use in this thesis. It will provide an overview of the active learning system, the scenarios of Active Learning system, Selection Strategies, the version space and the Divisive analysis (DIANA) algorithm.

Chapter 3 Presents the literature review of Active Learning using all the scenarios provided in literature, the version space algorithm and the classification algorithms.

Chapter 4 Introduces the proposed technique that is most informative vector selection using active Learning

Chapter 5 Reports performance measures, experimentation and results and comparison with other techniques.

Chapter 6 Concludes this work, summarize the work its application and future work.

Chapter 7 Contains references.

CHAPTER # 02

ACTIVE LEARNING STRATEGIES & APPROACHES

2.1 INTRODUCTION

Age of Information! It is not just a name but an era in which we all are living and actually we are contributing to make it strong day by day. In business, medicines, hospitals, agriculture in fact in every field of life a plethora of information comes on daily basis and it could be in the form of images, articles, research surveys etc. The information which is getting added to the repositories of different fields of life is in the form of unlabeled data which can't prove to be beneficial for the future. Now the point of focus is to dig out the important or beneficial work from this ocean of information (unlabeled data) and this can be only done if some categorization of information is performed on the records available.

The need of time is to make our machines so intelligent that they can handle whatever amount of information gets loaded on them. Machine learning, an offshoot of artificial intelligence (AI) is the only solution to the problem of huge volumes of data. In most of the machine learning tasks, we take some data and perform the labeling of it on the basis of features it acquires. The Labeled data helps to predict the labels of unlabeled data in future and this practice is performed in majority of the AI systems. Let's take an example of disease diagnosis, a machine learning system is developed to predict if some patient is going to have the chances of cancer or not, the system will take some previous records of patients from some hospital and will ask an expert (the physician) to label those records. Those labeled records will work as training data for the classifier of the system and on the basis of that learning the classifier will match the features of new unlabeled record with the previous labeled record and thus will tell if the patient is going to have the chances of cancer or not.

A machine learning system is said to be an ideal one if it carries a huge amount of labeled data and a very small amount of unlabeled data but in real world such systems are present at a very rare ratio and reason being the cost of collecting and labeling data. So now the main problem is the abundance of unlabeled data and small amount of labeled data, and the system has to be designed in such a way that it can get the maximum out of those limited training data. Some machine Learning phenomenon that can prove themselves the best especially in such cases are Supervise Learning, Semi Supervise Learning and Active Learning. The supervise machine learning and Active Learning are actually the main focus of the research that will be discussed in detail in the later chapters.

2.2 Supervise Learning

Gathering the labeled vector/examples is prone to many issues, like cost, error, time and for some cases it is not even possible to get labels. Supervise machine learning deals with the unlabeled vectors by actually labeling them with the help of a predictive function and that function is trained on some amount of labeled vectors. The set of unlabeled vectors X1,X2,X3,X4……Xn can't help in making the predictive function unless they get combined with some amount of labeled vectors. In supervise learning we start with the labeled instances/vectors and they are chosen randomly from the unlabelled vectors which are then labeled by the human expert. These vectors are in a pair (x, y). The x in the pair is the vector or instance and y being its label, in supervise learning a predictive function f: x → y is used which predict the labels of the new vectors on the basis of the training data provided. The training data actually helps the classifier to build a model or hypothesis and on the basis of that hypothesis, the classifier performs its future classification. In supervise learning we don't have some finite set of labels, any vector could have any set of label that are identified by the human annotator i.e. we could have following pair of vectors in the data (x1,yl) (x2,y2) (x3,y3)……. (xn, yn). In supervise Learning the predictive function totally relies on the amount of labeled vectors provided to it, if the labeled vectors are very small in number then the predictive performance would also be restricted and if the labeled vectors are good enough then a good predictive mapping can be seen.Supervise learning can be best understood from the figure given below.

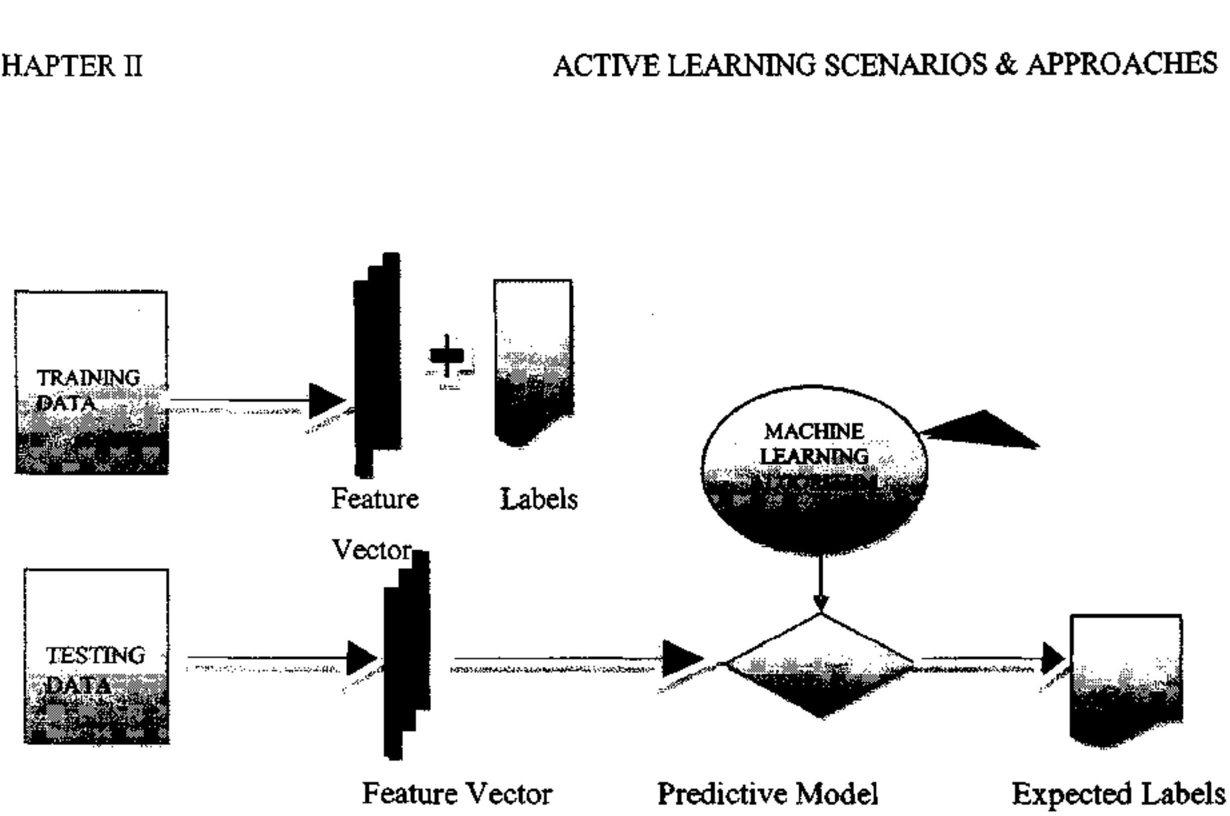


Figure: 2.1 Supervised Learning Mechanism

2.3 Semi-Supervised Learning (SSL)

This type of learning follows the beliefs that labeled data is not good enough for training a classifier, but we can get better results if we attach the large volume of unlabeled data with this small volume of labeled data [36].The term semi-supervised learning can go both for the classification and clustering domains in fact it can be called a center point between the supervised and unsupervised learning. It has also been analyzed that in some system the whole work revolves around extending the supervised or unsupervised learning into semi-supervised learning. The best quality of semi-supervised learning is that it is much faster than supervised learning and the major contributor to this are the time and cost. When we have little amount of training data then its labeling will definitely take less human effort, less time as well and because of these we call it faster.

Semi-Supervised learning can be better explained with the help of this figure.

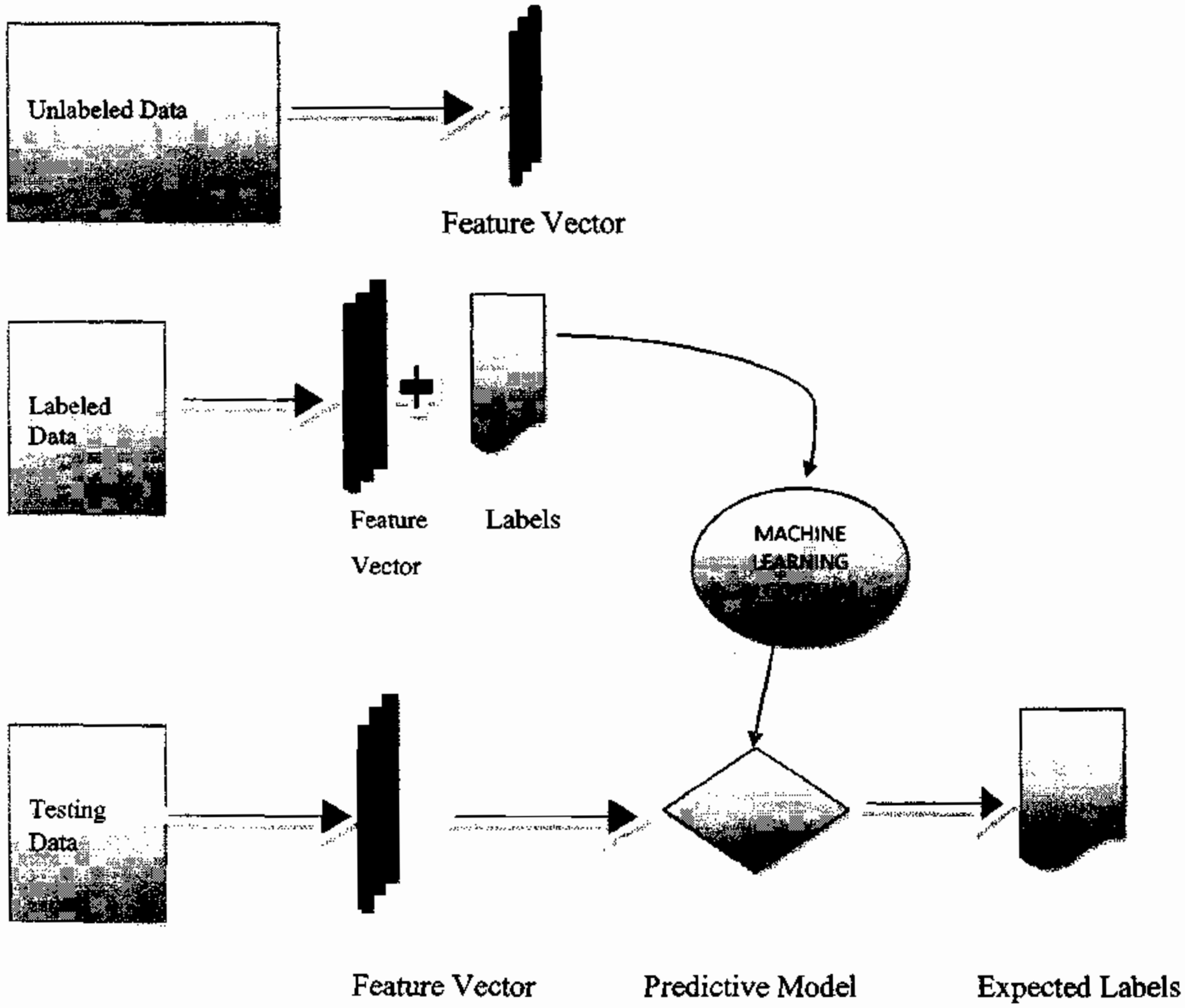


Figure 2.2: Semi-Supervised Learning Mechanism

As stated earlier, the semi-supervised technique can be applied to both supervised and unsupervised paradigms but much of the literature can be found in semi-supervised classification.

2.4 Active Learning (AL)

Active Learning is an offshoot of Semi-Supervised learning but a major difference among both these terms is that, in Semi-Supervised learning the algorithm randomly gets the data which it trains but in active learning we do the selective sampling of training data. Figure 2 is also a depiction for active learning but the selection of data to be labeled goes into two dilemmas as stated above. This term gained international attention in 1980's and since then it is proving itself to be a good area of research. The work of active learning gets completed in different iterations and in each iteration the classifier gets stronger.

Active learning falls into two main approaches: **Incremental Learning** and**selective learning.** In **Incremental Learning**a small fraction of unlabeled data is selected from the pool of unlabeled data and it is then labeled. Now this selected fraction of data will get permanently removed from that pool and inserted into the training set. Thus the training set increases after each iteration. In the **Selective Learning,** a selection criterion is chosen for the training set and according to that criterion we take a fraction of data from the pool of unlabeled data and then label it. The main point of difference of selective learning is that the selected sample for labeling don't get removed from the pool of unlabeled data while the training set's size keeps on increasing.

2.4.1 Issues Regarding designing of Active Learning Algorithm

The process of Active Learning starts with some preliminary decisions that are required for a successful implementation of an AL system. The tools and algorithm used for the active learning procedure will be discussed in the later section. First point of concern is to deal with some issues that are common for all learning algorithms.

1) How to select the Unlabeled Data for the first time? Mostly it is done randomly because at start we just predict a small sample to be informative and then after applying our technique we dig out where the good ones are located. This work can also be done by Pre-clustering, in that technique you don't start your whole process blindly but make some solid boundaries for picking the informative vectors.
2) What should be the amount of initial training set? The amount of training set is very important, as this set is going to train our classifier and if we take a small subset of data from a particular dimension then our classifier will be bound in its decisions. In incremental Learning this issue is not bothered as the training set incrementally gets added up with the new and informative examples. For selecting learning this issue

requires attention because on the basis of initial training set the classifier will recognize the patterns/features and will perform the later tasks.

3) What should be the stopping criterion? The stopping criteria can be pre-defined and post-defined. In most cases we see that stopping criterion is developed when observations have been made on the initial selection of data [26]. A very general stopping criterion of this type is the one which checks for the performance of trained classifierafter each iteration and then it stops the overall system when the classifiers performance ceases to improve.

4) What should be the classification algorithm? This is the main question that should be answered before getting started. Active learning mostly doesn't have any particular classifier that is used for AL only and in most cases it uses the typical classifiers that are used for machine learning. There are many classifiers available for supervise learning tasks but according to the field in which we are working we should carefully take the decision of selecting the classifier i.e. If we have to do active learning for documents classification then I'll have to see which classifiers shows the best performance in this domain.

These were some issues that needed to be taken care of before getting started with the overall active learning system. The success of any system always relies on the primary steps taken as these are called the deciding steps towards success or failure. Active learning basically relates itself with the pre-processing, as before sending our data to the classifier we try to get the best out of all data available and then pass it on to the classifier for its improved accuracy.

2.5 Active Learning Scenarios

According to the fore mentioned concept of supervised learning, a random set of training data was always getting selected for the classification but it was actually stopping the performance of overall system at some point. To overcome this issue, the term active learning was developed which actually gave the freedom of selecting the most informative training data for some valid requirements. Two most general scenarios of active learning that are used in majority of the active learning systems are: Pool Based Active Learning and Stream Based Active Learning.

2.5.1 Membership Query Synthesis

This query synthesis phenomenon is one the pioneer concepts of active learning. In membership query synthesis the learner queries regarding the labels of instances which are selected DE Novo by the algorithm. The main idea of this scenario is that the hypothesis is generated by the learner (Algorithm) and human annotator performs the confirmation or denial of the hypothesis.

This scenario is found to be impractical for many scientific researches and the reason being the generalization of such a hypothesis which is not at all related to some practical problem domain. The example which I am going to give now will elaborate the problem of Membership query synthesis. In some image processing system a learner will generate a hypothesis containing some particular images. There are chances that the generate images have no real semantic meaning related the topic under discussion then the human annotator will find it hard or even impossible for him to label those images and thus the performance of overall system will fall to zero.

Despite of all the real time problems of this scenario, some positive outcomes of it have also being notified. In [4] a robotic system was developed which automatically generated a series of hypothesis and carried the scientific experiments based on those hypothesis. The surprising result of their proposed system showed that their artificial intelligence outperforms the human knowledge of generating intelligent experiments and checking the gene function with respect to deletion of mutants from yeast. The cost of overall system decreased up to 100 folds respectively.

Thus, the use of membership query synthesis could be a strongest decision for some system development but research has proved that in majority of cases it failed as the random amount of queries generated by this strategy can sometime come in an arbitrary amount. This thing can prove to a big burden for the human annotator who is supposed to label the queried instances.

2.5.2 Stream Based Active Learning

Much closer to the real life examples, Stream based active learning works for the filtering of a live stream of vectors rather than creating a self-developed artificial vectors. It is also called a form of sequential active learning in which the learners picks only one vector at a time and then this sequence continues until some stopping criteria is met. The learner keeps on filtering the live stream of data and then decides whether to label the selected vectors or not.

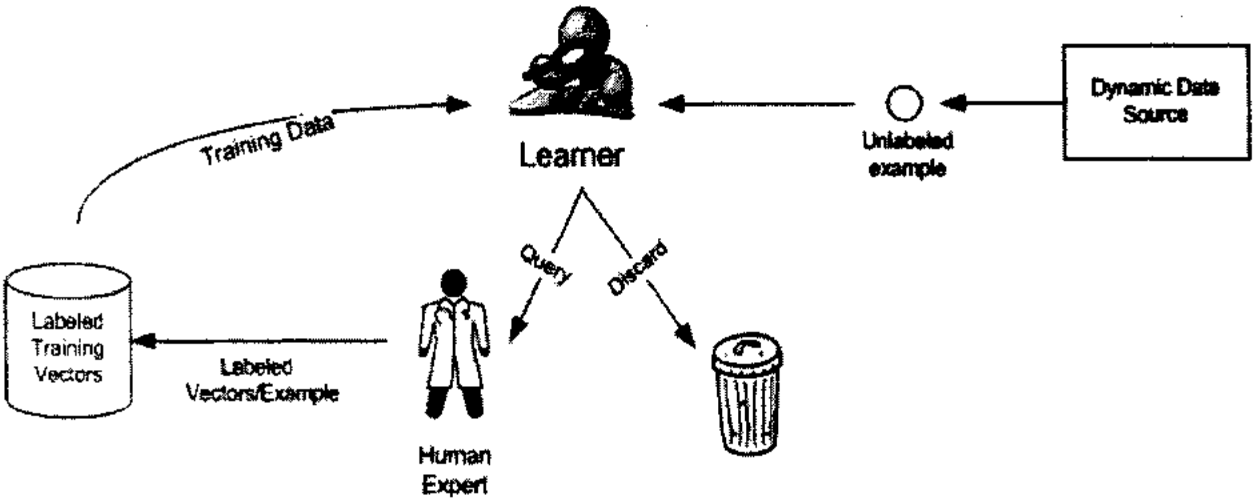


Figure 2.3 : An layout of Stream Based Active Learning

The use of Stream Based Active Learning has been found mostly in the projects where video sensor data is involved and the system is facing a live stream of unlabeled stream of videos. A major risk involved in the live streaming of data is the noise, and the advantage of stream based Active Learning is that it deals with the noise in data. Some other benefits being it efficiency in dealing with complex data, time frame data etc. The desirable use of stream based active learning systems is in the dynamic atmosphere especially in online classification systems.

A limitation of this technique in some systems is that the learner can't access all the unlabeled data all at a time in order to create the most informative vectors and because of this issue, this technique is used mostly for the dynamic systems.

2.5.3 Pool Based Active Learning

One of the easiest works nowadays is to collect large amount of unlabeled data which can be gathered at once. This gives rise to the concept of pool based active learning which works on the assumption that we have a large amount of unlabeled data and a very small amount of labeled data. In most of the pool based scenarios the queries or examples are selectively drawn out from the pool of unlabeled data and the selective procedure tends to be bound on some standards.

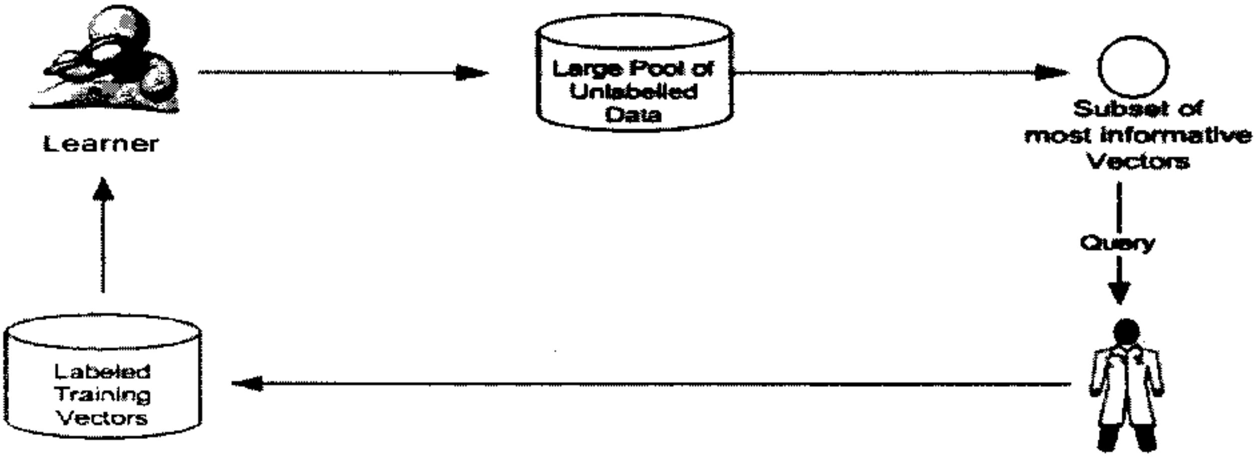
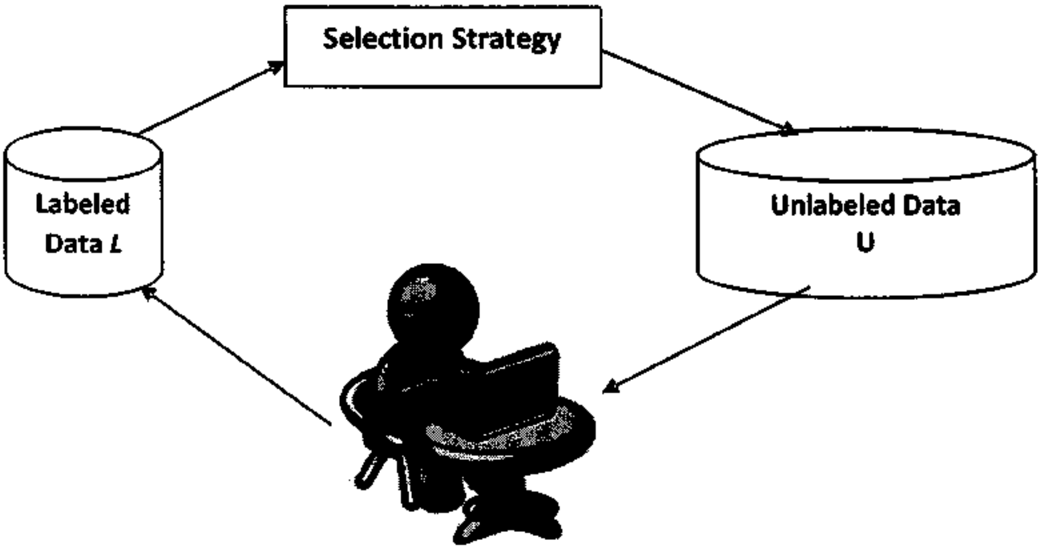

Learner
Large Pool of Unlabelled Data
Subset of most informative Vectors
Query
Labeled Training Vectors

Figure 2.4: A Layout of Pool Based Active Learning

The result of pool based active learning lead to the training of classifiers on manually trained instances and this perspective of human experts can prove to be more practical and efficient as the minimum amount of training data would be easy for any expert for better labelling. The resultant instances from the classifier can prove to be best for inferring the labels for more unlabeled data that require labels.

Another clear picture of the whole phenomenon of pool based active learning can be given on the behalf of human annotator. In the figure given below I am going to explain the pool based active learning as a model and its important components which should be given the due emphasis. The practical overview of the Pool based active learning comes as a quintuple model {*U, L, A, S, SC*} which is used in majority of active learning systems. The output of this model are the most informative learning examples which are very minimum in number and gives the best accuracy of the overall classification system. In the figure 2.5 three chief components of pool based active learning have been shown which are actually minimizing the effort of human annotator and are also maximizing the efficiency of the overall system.



Human Annotator

Figure 2.5 : The Pool Based Active Learning Scenario

U is the actual pool of unlabeled data from where the L (the learning examples will be abstracted and given to the A (the human annotator). S is the selection strategy by which the new instances will be taken from U and after getting labeled will be added to L. SC tends to be the stopping criteria by which the whole cycle will stop.

The Pool based active learning is based on two chief decisions, the first one is the decision for selecting the first and key labeled data *L* from *U* (either the learning data should be selected randomly or on the basis of some criteria). Second one is making the selection strategy which will pick the instances to be added in *L* after each iteration and this will actually decide the efficiency of active learning system.

The Stopping criteria SC is another important aspect of the pool based active learning system. In majority of the cases the stopping criterion is to run the algorithm until all instances of *U* will get added to *L* and no instances is left in *U*. The root of active learning process is the number of learning examples which can be called the batch size *b*. This thing must also be kept in mind that the batch size must be as minimum and efficient as possible because the aim of the overall system is the decrease of cost and time in terms of labeling.

2.6 Main Frameworks of Active Learning

Whatever the kind of scenario be applied in active learning but the main theme remains the same and that is evolving the in formativeness of unlabeled examples/vectors provided. We may find an abundance of literature showing the querying strategies for active learning but in this section I am going to describe the frameworks that are closely relevant to the work that I have performed. The process of active learning can be executed under various frameworks also called the sampling techniques but the two main which are required to be mentioned in this thesis are as under:

2.6.1 Uncertainty Sampling

This is one of the most famous sampling technique of active learning which works on the strategy of pure classification system, it actually works for measuring the confidence of any classifier on the unseen or new vectors. In uncertainty sampling a classifier is built on the basis of vectors which have got labeled by the oracle/ human expert. A ranking classifier like K Nearest Neighbor (KNN) classifier is used in the uncertainty sampling. The main procedure is that, after obtaining the classification results from the classifier, a measure of uncertainty is calculated from that output/result. The measure of uncertainty of the classifiers actually gives us the vector about which the classifier is least certain regarding labeling. The Least certain examples are then chalked out for the next step of training the classifier, actually these examples are given to the oracle/ human annotator and after the genuine labeling these are given back to the classifier as training data in order to pursue the next round of active learning process.

The main benefit of uncertainty sampling was that the classifier gets trained on the instances about which the classifier is most uncertain. This least assurance of classifier regarding the labels of some vectors make them more prominent as the examples which are hard to label and because of this reason these examples gets labelled from human annotator which label them according to his professional and real life experience and then these genuinely labelled examples goes again to the classifier as its training data. Thus the final result would be in the form of improved quality classification by the classifier.

2.6.2 Query by Committee

As the name depicts, this strategy works on the concept of making a committee of classifiers which takes decisions regarding the labels of new examples. The training of all those classifiers is performed on the set of training data which may or may not be same. After the training of all these classifiers an unseen data is presented to them. All the classifiers are given with same testing data and they label that data according to their training experience. The labels of the new data is compared for all the committee of classifiers and then the vectors/examples for which the classifiers are showing higher level of disagreement, are chosen for the next step of active learning. This strategy has announced those examples as unique for which the committee has higher level of disagreement and then these unique examples are given to the human annotator/oracle for labeling. The labeled examples by the Human expert are then presented to the final classifier as training data and the next phase of active learning proceeds.

The benefit of this strategy is that it has tried to chalk out the most difficult examples from the data. Then these difficult examples gets labelled by human expert who labels them according to his expertise and thus the classifier of the active learning system gets the training data for which it can trained at the best level. One drawback of this strategy is that it is benefited from the classifiers that have got trained in different domains and thus there are high chances that their results don't come same like each other. The purpose of this thesis is to work of the same kind of classifier that is trained in different data so the concept of Query By committee doesn't match the requirement of this thesis. The algorithms which are going to be used in this thesis are described in detail in the next section.

2.7 Support Vector Machine

The Support vector machine was proposed for the first time by [2*] and the main idea was to create an algorithm that minimizes the distance between the training patterns and decision boundaries. From the time of its development, this algorithm has attained a very strong empirical success. The main idea of support vector machine revolves around a strong separating hyperplane that is created between the data points of two classes.

The training data of Support Vector Machine are the real data points that are labeled by the human experts and on the basis of those labeled examples the SVM brings labels for the testing examples. The working behind this separating hyperplane and maximum margin hyperplane will be discussed in the later sections of this thesis.

In this thesis I am going to use a hybrid technique in which the overall classification is going to be followed by the clustering. Both Classification and Clustering will be done for the multiclass data as the main idea of the thesis counts on the dynamic classes for the testing and training data.

2.8 Hierarchical Cluster Analysis

Traditional clustering is a simple job in which the clusters are created by measuring the difference between the data points. A much known drawback of traditional clustering is that always requires a predefined number of clusters before starting the procedure. To overcome this problem the concept of hierarchical clustering was introduced. Hierarchical clustering is quite a different work which follows a step by step method for creating the clusters. As shown in the figure below, the hierarchical clustering is further subdivided into two main groups: The agglomerative clustering and divisive analysis or DIANA.

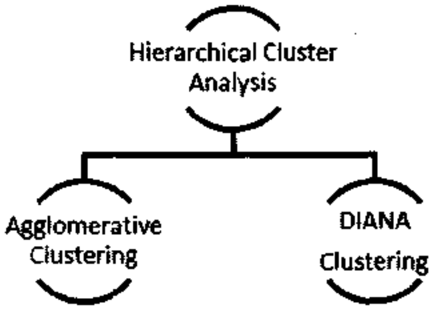


Figure 2.6: Hierarchical Cluster Analysis

2.8.1 Agglomerative Clustering

In this form of clustering we don't need to predefine the number of clusters, it's a bottom up approach of clustering in which the individual data is clustered into as much clusters as possible and then in a step by step procedure the related clusters are merged into a single clusters and this process continues until we get the minimum amount of clusters possible and these minimum comes as two in numbers in majority of the agglomerative procedures.

The agglomerative clustering is mostly shown in the form of a dendrogram and an exemplary dendrogram view of an agglomerative clustering is shown in the figure below.

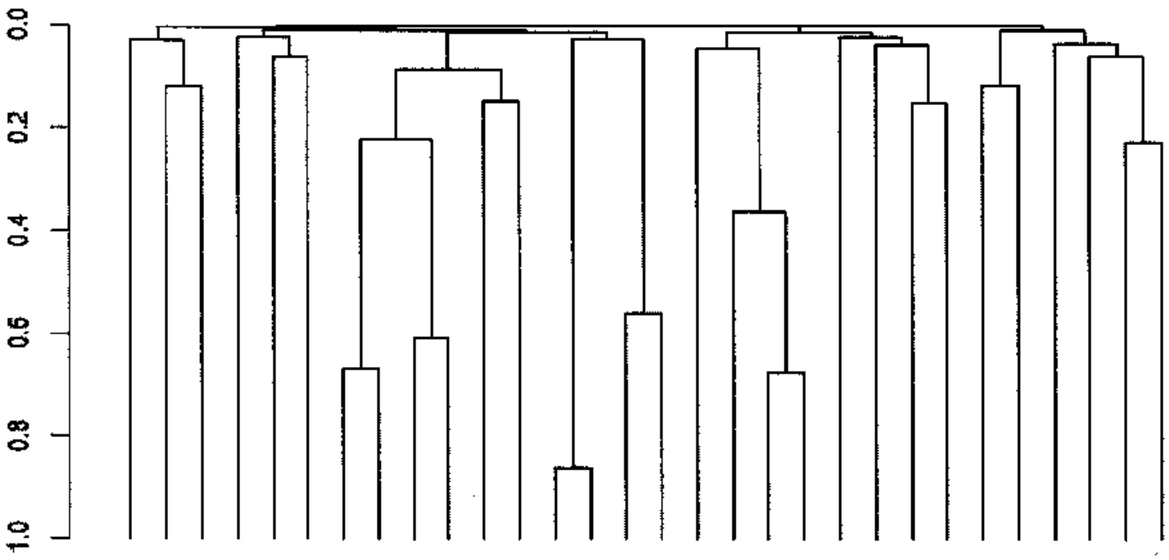


Figure 2.7 : Agglomerative Clustering Dendogram

This figure is showing that at the start of the procedure there are so many individual clusters which are later merged into groups and in the last iterations there are very few clusters present as the others are mixed up into them.

2.8.2 Divisive Analysis (DIANA)

Another important hierarchical clustering algorithm which I am also going to cover in my thesis implementation is the DIANA algorithm. The previous method of Agglomerative clustering was following the bottom up procedures for clustering but this algorithm goes in reverse. DIANA is a top down method for creating clusters, the method starts at the top in which we have all the data points assembled in a single cluster. The top down method is practically a bit more complex than the bottom up method as this requires a constant help from a simple clustering algorithm like K mean. Another benefit of this method over the bottom up method is that it goes for the recognition of globally distributed patterns right from its start while the bottom up method relies on the local distribution of data.

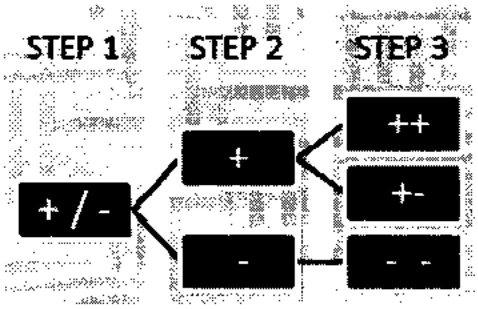


Figure 2.8: A simple layout of DIANA algorithm

The above mentioned figures shows an example of the divisive analysis, the steps are not necessarily be as minimum as shown but the overall procedure moves like this.

CHAPTER # 03

LITERATURE REVIEW

In this chapter a detailed literature review has been provided in order to get the required background knowledge for exploring the thesis subject. The review started with a little introduction to machine learning techniques and then a detailed summarization of the areas covered in this thesis are discussed. Detailed review has started with the semi-supervise learning which is somewhat a major area related to active learning, the key area to be covered in the thesis. Finally the two core techniques used in active learning have been discussed.

3.1 Preliminary

Dealing with a large amount of data is really getting a major problem for most of the industries either they are related to medicines, information technology, image processing or social networking etc. The machine learning which can also be called query learning is nowadays attracting a huge amount of researchers along the globe. The area of machine learning is basically an offshoot of artificial intelligence and it deals with the development and designing of such programs that can derive the rules from the data provided, and on the basis of those rules they can improve the performance of overall system.

With the gigantic emergence of the electronic data management system in the past decade, the machine learning has become crucial. The two main areas of machine learning are supervised learning and unsupervised learning. Supervised learning deals with the instances that are paired with their class labels and the predictive models are created from those instances while in unsupervised learning the system is provided with unlabeled data and the major goal is to organize the instances is some structured form i.e. clustering the data points etc. Another famous type of machine learning is the semi-supervise learning in which we have a large amount of unlabeled data and some amount of labeled data as well. Active learning could be named as semi-supervise learning as it also have the large amount of unlabeled data and a small set of labeled data but the main idea behind active learning is that if the learning algorithm is allowed to choose the data from which it will learn or generate the rules then it will perform better even with less amount of training data.

The main idea behind the approach of active learning is the evaluation of uncertainty of the instance; it is assumed that if the uncertainty of current sample is high then the model don't have enough knowledge in the classification of that sample and thus if we include this sample in the training data it can increase the overall performance of the model. A Large number of methods have been applied to check the uncertainty of the instances and the major two of them are [52]

(i) Active learning purely based on uncertainty of the instance

(ii) Active learning further considering the instance correlation.

This Section will provide the details of literature presented in the above mentioned areas of machine learning which I am also going to cover in my thesis.

3.2 Semi-Supervised Learning

The classification of text is the process of assigning set of predefined categories to the document on the basis of content present in the document [9]. Classification could be on the basis of labeled data, the amount may vary according to the method used. The labeled data is though a big interest for the text classification task but it's a fact that this kind of data is not easily available. The labeling of data may bring the blessings of cost, time wastage and this is all because of the involvement of human effort in it as no one labels loads of data for free and in the fraction of minutes. This problem brought new areas of research and most popular of them are semi-Supervise learning and active learning. Semi-supervised learning and active learning both hit the same problem of dealing with the unlabeled data by having just a small amount of labeled data [17]

Like many other field of information technology a lot of work has been done in image processing and object recognition based on semi-supervised learning, [44] has used this approach for the purpose of track classification for the dense 3D range data. The main idea revolves around training the classifier (EM learning algorithm was used) and extract useful examples by the exploitation of the tracking information. They speed up their algorithm with further addition of the incremental training of the classifier and concluded that this addition gave a good increase to the efficiency of the classifier. A major limitation to their system was the reliance on the segmenting object which is the background image and there are much chances of the arrival of some undistinguishable background image that may affect the whole process of classification.

Many studies have taken place for the purposes of comparison between supervise learning and semi-supervise learning i.e. [19] has checked that how they can deal with the noisy dataset problem. The main focus of researchers in this paper was to go for the cost sensitive classification; they started by applying a general classification strategy by integrating the misclassification of cost for noise handling. Then they boosted up their research by bringing a semi-supervise classification type strategy in which the noise detection results got added to the training iteration by iteration and the accuracy of overall system in noise identification got improved. A limitation to their work was that the major focus was given to the cost of expensive classes which was actually giving all the focus on some classes while the other were getting neglected and when it comes to unlabeled, your prediction of the most important class may be proved wrong.

Cost sensitive semi-supervise learning tries to achieve the optimal solution to the classification problem by minimizing the cost as much as possible. Another good example of the cost sensitive semi-supervise learning can be taken from [31], in their work they have proposed a cost sensitive semi supervised learning algorithm to deal with the problem of less labeled data. The algorithm first takes the label means and then it perform the classification, as the cost always increases when we have to label a very sparse data. In their work they have tried to minimize the distance travel by the labeler by giving a new framework for classification. Their work was basically a cost sensitive extension to the approach of [27] in which the label's mean are used to classify the unlabeled data with the help of support vector machine (SVM).The results have shown the significant improvement in terms of cost as well as accuracy. The work has focused on the binary class problem only while this approach can be applied for the multi-class problem as well.

In the field of clinical research the semi-supervise learning has attained quite a scope, automated text analysis specifically in electronic health records (EHR) using natural language processing and machine learning have got very popular in the recent decade [32]. The paper has emphasized on an automated system for the clinical records that may contain important information regarding disease diagnosis or regarding the symptoms of some particular disease. The work was not only focusing on the record of some particular patient as they were trying to train their classifier on that free text that was previously related to the diagnosis of some particular disease and data could be of any number of patients. The major advantage of their proposed system was that it was not requiring human effort; a good efficiency of the system was attained on the basis of previous record they were training their classifier. A limitation to their work was that, it was giving much attention to the clinical expressions that may be used for disease diagnosis so a linguistic model must be integrated with the machine learning model in order to get the accurate results in future.

Reference [40] can also be analyzed as a good example of disease classification with the help of semi-supervised learning. The work was focusing on the dimensionality reduction in semi-supervised learning and the area was medical image classification. They have combined their own work of [28] and the work of [33]. In the first one, a discriminative way for reducing the dimensionality was adopted without bringing any change on the semantics of the images while the second one was concerned with the incorporation of the information that was obtained from unlabeled data. It was concluded that when the semi-supervised dimensionality reduction was utilized for the reducing of image dimensions in the presence of unlabeled data, it outperforms all the simple supervised techniques being used for this work. A limitation to their work was that they have done a little comparison between the generative terms proposed by them and the laplacian regularization which was used previously. The future directions could be the implication of same strategy in other medical scenarios as well.

From the research it can be seen that the electronic health records (EHR) are always stored in an unstructured form while it is also true that much of the important medical information can be obtained from this free text or EHR. [45] has also focused their attention on this issue, they developed an automated semi-supervised classifier which can identify the useful free texts on the basis of some labeled examples that have been provided to it during classification phase. The main aim was to improve the diagnosis for coronary angiogram and ovarian cancer from the previous results of these diseases. Their algorithm outperformed all the previously utilized supervised learning algorithms and it was concluded that their algorithm can efficiently detect useful records in the EHR which can help in further disease diagnosis. The limitation to it was that the algorithm can get easily failed in front of linguistic expressions as no language aid has been utilized for interpreting the texts. In future the algorithm can be improved to get more detailed output rather than just getting the output as presence and absence of the disease.

3.3 Active Learning

The active learning strategy has got much resemblance with the semi-supervised techniques as both of them works for decreasing the amount of labeled data while improving the classification accuracy. The term active learning gathered attention by the researchers in 1980s [2*] and since then it is a very popular area of research. The main idea proposed by [2*] was that the learner may have the option to ask queries that might be of its membership (is this instance member of A class or not?), relevance (is this instance related to this class or not?) etc. The learner alters the value of instances by asking queries and thus after a specified number of iterations a model can be generated.

Active Learning can be implied on many domains where we have large amount of unlabeled data present and labeling tends to be a hard issue in terms of cost, time, and human effort. For example in **Drug Discovery** [15], **natural language processing** [18], **information extraction** [21], **information retrieval** [22] and many more.

The typical settings of the active learning framework can be seen from the work of [11] in which active learning is utilized for parameter estimation in the **Bayesian network** which has used the uncertainty of the model as its primary object for the computation of the loss function. The algorithm developed by them involved an iterative function by which they can actively select those instances that make their performance much better in learning. A limitation to their work is that they have worked for the structures that tend to be known while this thing can bound the classification in future when new structures will arise. The future direction could be that, the work should get extended to the continuous variables.

The general overview of active learning can easily be taken by **Model and Model loss phenomenon** [12]. First of all the model for active learning is chosen and then a model loss function that is according to the learning task. At start of the process we built up a model on the basis of amount of information we have, then when the process continues it asks more queries (might give more data which the learner is supposed to learn) and on the basis of that query the previous model may vary and this is by the model loss function.

The term actively learning can fit into the settings of both **supervised learning** and **unsupervised learning** but majority of work is performed on the supervised learning side.

Another term which has been mentioned before is **Semi-Supervised learning**, the field in which there is plenty of literature available regarding active learning and the reason being that their working resembles to each other. Another type of active learning is the reinforcement learning which can be simply called as a process which starts at one domain but once it start working it keep on entering into other domains as well.

Active learning further gets divided into many scenarios and study of literature shows that the first ever scenario to be proposed in the field of active learning is the **Membership Query Synthesis** [2]. In this phenomenon the learner is allowed to ask for the labels of any unlabeled data in the input space and this technique often gets fail when we have the arbitrary data and the labeler is some human expert. A very reasonable approach to the Query synthesis process was given by [16]. They proposed a **robotic system** that was going to be implemented physically and the main idea was to integrate the **artificial intelligence** with the robotic technology. The closed-loop method of **active learning** was being utilized in their work in order to get the implementation of serial experiments in a very intelligent manner. The robot developed by them was capable of carrying a series of **biological experiments** in order to discover the metabolic pathways in yeast and this was all possible because of an active learning approach in the background that was based on **inductive logic programming**. The results came out to be pretty much positive in that regard as because of active learning, little effort by the human annotators was required and this was eventually good for the cost of overall system. A limitation was still present in their system and it was related to the random generation of experiments, according to them they have decreased the cost almost 10 folds during the randomly generated experiments by the robot. This random generation may get false if the robot starts working on the expressions that are not familiar to it or that were not included in the predefined model.

The active learning phenomenon is getting much new advancement just like some other active fields of research. [53] Has proposed an active learning system of **attributes based feedback process** in which the learner not only queries for the labels of the instances but the human expert also gives his feedback about the query. A more communicative way was tried to be created among man and machine and this project was carried for the image classification. Basically they have represented the human expert as a supervisor which is teaching visual concepts to a machine. For example on some image, the learner will say "This is a garden, what do you say?" the supervisor answer like this "No, this is too open to be a garden". After getting the feedback of the supervisor they also introduced a **weighing schema** for checking the likelihood of any image and thus enhance the **active learning process**.

Active Learning spreads itself to **multiple instance learning** [56] and this process has been continued because now the researcher have tried to move from working on a single instance to a bag of multiple instances. Moving on to the bag instead of single instance can be risky in terms of computational cost but [56] has tried to overcome this problem by introducing a new phenomenon of **pairwise similarity based instance reduction for Multiple Instance Learning(MIP).** The whole process was dependent on the similarity among the instances within a bag which is named as training bag. The work could have been much improved if they would have worked on the pair of instances without using the process of bags.

A very similar work to that of the proposed system of this thesis is done by [54]. The simple active learning process for selecting the most informative query has been created with the help of support vector machine. The overall process works in the binary class domain and initially they carry just 2 instances in the hyper plane, one positive and the other negative. As the process continues the values of hyper plane keep changing and the instances are selected according to their minimal distance from the hyper plane. The overall system's accuracy was above 90% and a major contribution of the new system was that, it was not working on a artificial dataset as the dataset was now getting a proper location near or far the hyper plane.

The Query synthesis Scenarios carries a little literature work specifically in the field of active learning and this might be because of its problem with arbitrary data labeling. The main limitation to the membership query synthesis problem was that it fails in some natural language processing systems specifically when the system generates streams or huge amount of data to be labeled. To overcome this problem the two new scenarios were developed [34] and these are the two most popular scenarios of active learning in which one may find plenty of work done by the researcher. They are shown as under:

- Stream Based Selective Sampling
- Pool Based Active Learning

In the section below I am going to give the little details about the research work that has been performed in the following fields.

3.3.1 Stream Based selective sampling

As mentioned above, this scenario came to overcome the problems created due to membership query synthesis. The concept of **selective sampling** was first proposed by [6] and the main idea was that, if the unlabeled data is freely obtainable then why don't we sample it first so that the learner may get the right either to label it or not. The outcome of it was that the result of overall system got improved. As in this scenario each instance comes to the learner once and then it gets removed from the overall data, it is called stream based sampling.

The stream based sampling acquired a great interest from the researchers and it got popularized in to many other machine learning domains. Mainly this technology is utilized when we don't have static data and the learner has to cope with a regular stream of data. The work done by [23] is based on discarding, caching and then recalling the samples in active learning and they have performed the classification in stream based environment. The main idea of the paper was based on the observation that dynamic data like handwriting recognition data may vary over time so instead of discarding data after labeling we must have some recall function that may ask for the label of same data after some iteration. Their stream based setting was repeatedly based on decisions of removing data from active stream, then caching those decisions and then recalling that data later in future. It was found that the proposed setup was very beneficial for learning especially when we have to update our model for the new coming data. The man limitation in their work was that, it was getting complex iteration after iteration especially in terms of the cost of labeling.

Now an issue with stream based sampling is that, on what basis the learner should decide whether or not to label any instance? The possible solution to this issue could be the informativeness of the instance (which I am going to cover in detail in my work). There could be any selective sampling algorithm and I am going to discuss the popular among them in the later section.

Another very interesting thing seen in the domain of stream based selective sampling was that some researchers were using this to enhance the pool based active learning which I am going to discuss in the next section. In [24] the researchers have done the selective sampling in order to enhance their **SVM** classifier which was working mainly on the pool of unlabeled data. An online detection system for the unknown computer worms was developed and the data was taken by monitoring 323 computer features which were later reduced to 20 after feature selection. The stream based sampling was actually utilized to get the real time records and the performance was observed to be considerably improved after the addition of active learning with the simple **SVM** classification. A limitation in their system was that, it was giving its best performance with the typical computer worms but still for the detection of sophisticated worms they were using same classification which was again bringing the cost and human effort.

As mentioned above, the stream based learning strategy is largely implied for the dynamic data and such data is mostly available for online systems. [41] has shown a very important work performed in this domain for the purpose of online active learning in data streams. The main study was performed on **selective labeling** in data streams. The **Bayesian networks** have been consulted for getting the posteriors distribution of the initial instances in terms of their weights. Further a procedure has been adopted for checking the likelihood of the weights. The work was inspired by the spam detection system that are used for the online stream of data and on the basis of that the Bayesian algorithm was added up with the weighting of sample and forgetting of sample. The results gave a manifest improvement on the accuracy of overall system. The work can be further improved by taking into some other domains through active learning, the domain could be fraud detection, sentiment analysis etc.

Another work performed in stream based active learning is performed by [46]. In their work the term active learning has been utilized in the context of **exploration** and **exploitation**. The have developed the system that works on the contrary to the previously used heuristics method in Bayesian classification. According to them if they use the process of extensively classifying the image and videos they can get good results, a learner will get the ambiguous instances constantly and the human expert will keep on labeling them and this process will continue. They have utilized the posterior distribution of classes while making a committee of previous hypothesis, the two hypothesis were created for each instance and if the classifier shows disagreement on both hypothesis then that instance goes to the human expert for labeling. Now the instance that has got labeled by human annotator will now go to the classifier as training data. The system developed by them outperformed all the previous stream based active learning systems but a limitation in their system was that, it was not well prepared for the noisy data and in case of any noise in the data the hypothesis may get affected and thus the overall system's accuracy will be affected. So in future they must bring such changes in their system that can cope the noise in online streams of data.

By the time, the advancement in areas of research is bringing new concepts into the domain of active learning. The area of **concept drifting** was introduced under the stream based active learning range. As we know that the data in streams carries the requirement of getting predictions in real time and here the main issue that can arise is concept drifting. So the learning should be so strong and adaptive that instances don't get wasted from memory without getting labeled. In [57] three active learning strategies had been adopted to overcome the above mentioned problem. The three concepts are based on **uncertainty, randomization** and **dynamic allocation** of data. The results proved that the proposed strategies of splitting data according to concept drift performed very well especially when the labeling resources are very small.

3.3.2 Pool Based Active Learning

In real world, we can gather a large pool of unlabeled data in any domain, this easy availability of unlabeled data gave rise to the concept of Pool Based Active Learning. Usually we draw queries from the pool which are non-static in their nature but this is not a strict rule of this technique they can be dynamic for some examples as well. The major point of discrimination among stream based and pool based active learning is that the first one sequentially goes through all the data while the other one deal with too many instances at a time in the form of huge pool.

The pool based active learning has been performed in many real world scenarios, like Text classification, image classification, disease diagnosis, speech recognition etc. Much of the work in active learning is done by the technique of pool based, [7] has utilized it for reducing the cost of labeling for a huge set of unlabeled data. The concept of **(Query by Committee) QBC** was modified in their work and the key aim was to get the density of document explicitly at time of selecting the examples for labeling. Their probabilistic framework was utilizing the **EM** algorithm in addition to the typical active learning frame work and their density weighing methods and EM with active learning methods proved that the accuracy of the system can be improved by having as less training data as possible. A limitation to their system was that, the density estimation is known to be a hard problem especially when we are concerned with high dimensional data. The work can be further improved if they utilize the concept of poor probability with the density weight scheme and further techniques for interleaving EM and active learning can also be explored in future.

In [29] the **pool based active learning** has been performed in order to deal with **linear regression**. In their work, a very clear comparison was made between **passive learning** and **active learning**, the solution they provided for passive learning problems was a good change for the accuracy of the system. For linear regression, two pool based active learning criteria were developed and they were actually extensions to the work of [10] & [20]. Their first method was creating a close form of the best resampling function while the second one was based on the conditional expectation analysis. A final Method of active learning was also proposed which was actually joining the methods given above. The results showed that their proposed method outperformed the previous population based active learning system but a little reservation regarding their system was that it was doing well for approximately correct model and if the model gets a little unspecified their system may fail. So in future a model based approach must be generated which can improve the efficiency.

In [47] the pool based active learning has been applied for the problems of **binary classification**. The proposed system was named as **UPAL (unbiased pool based active learning)** which tries to minimize the unbiased estimator of risk. The proposed system was developed for the noise free scenario and it always works for the unbiased sampling of the labeled data whereas this thing is not always efficient in the practical way. A limitation in this work (as already mentioned) is that it gives the best results when data is completely noise free but when you think of the real world scenarios, noise do comes with the data and if not taken care of it can cause serious problems with the overall system.

3.4 Approaches in Active Learning

The active learning is mainly concerned with the selection of most informative instances and majority of its approaches are working on this regard. In most cases we have seen that the most informative vector is selected by analyzing and selecting the instance about which our model is most uncertain. A similar technique to this is **co-training** [8], in which we have very small amount of labeled instances and large amount of unlabeled instances, we train a classifier to get tags for the unlabeled data and we'll get the decision values of that classifier for all these unlabeled instances. Now among the labeled instances we select the ones for which classifier has maximum confidence and these instances makes up to the training data for the next classifier and the iteration continues until a stopping criteria is met. In [8] a limitation to co-training is also given and according to that, in co-training the process is completely independent, there is no interaction of human expert directly with the instances which may cause problem in accurate future classification. To cope with this problem the term active learning is used, where the classifier queries for the labels of instances and a human expert perform the labeling according to his practical expertise.

Many sample selection strategies have been involved in active learning and the major among them are Error reduction, uncertainty and relevance. In the section below I am going to discuss a little literature on the above mentioned selection strategies.

3.4.1 Error Reduction Strategies for Sample Selection

The aim of error reduction strategy is to minimize the expected error of the system and this is done by the **estimation of error.** [13] has proposed the error reduction strategy for the classification of text and according to them they were the first to be using the Naïve Bayes algorithm for this purpose. The classifier was further supported by two strategies which are **LogLoss** and **0-1 LOSS** which were mainly concerned with **entropy** of the **posterior class.** Another big contribution of this paper is subset reduction with the help of which the **Error Reduction Strategy (ERS)** can be applied to small subsets of data. A limitation to their method was that their proposed method of version space gets fail when we have a model that has complex parameter structure and not all parameters of the data goes for a single data set. In future this work can be further extended for other classifiers like **SVM** and others.

3.4.2 Uncertainty Based Strategies For Sample Selection

Mostly the practical work in the field of active learning has been performed using the domain of uncertainty, a sample or an instance uncertainty is measured which actually tells if it is informative or not. After checking the uncertainty value, the sample goes to the model and trains it for the future data. In [14] the uncertainty based sampling has been applied by measuring the distance of the instances from the classification boundary and the work is one of the most popular works in this field which is utilizing the **SVM** for the whole classification. A detailed theoretical baseline has been discussed in the paper using the notion of **vector space** and this thing also counts up to a limitation in their work. The **version space** concept goes well for the binary classes but even for 3 classes the results get drastically changed and even sometimes the classification process fails completely. The future work could be that the SVM can prove to be a good replacement for version space and future instances if comes directly to SVM will get the classification accuracy improved.

In active learning a big challenge is that, how well we can make our **uncertainty metric**? The uncertainty of an instance can be calculated by bringing a bias in our model or by predicting the value on the basis of some rules. In [35] a clear comparison among different techniques used for creating uncertainty metric has been performed. A detailed description of different algorithm in querying models and feature selection was provided and it was stated that on what type of data which model can perform at its best. The query strategy can be further improved by the addition of some advance features. The uncertainty sampling can be further divide into two categories: Based on Bias and Based on Prediction

3.4.3 Uncertainty sampling with Bias

In machine learning we are not always provided with balanced data, i.e. there could be any proportion of data available, more positive less negative or more negative less positive. The second phase is mostly applicable for the real world problems as in most of the cases we have very little proportion of positive data available. In all such cases we have to bring a bias in our algorithm which eventually digs out the positive examples. In [48] a bias was introduced in the version space in order to get the most informative examples with the help of **SVM classification**. The work was based on the **inconsistency based active learning** and the concept of **QBC** was utilized to get the desired training data for our classifier. The algorithm was proposed to get those results which are inconsistent with the rest of data and about which we can say that the classifier is most uncertain about them. According to them, they were the first one to introduce the concept of version space in this domain and using this concept, they actually extended the work of **QBC** by taking two extreme hypotheses instead of many hypotheses. The experimental results showed an improved generalization performance but a major limitation to their work was the increase in time complexity. Apart from just time complexity, the proposed algorithm was also blindly going for two extreme classes while there may be some small classes in the data as well.

3.4.4 Uncertainty sampling with Prediction

The concept of uncertainty sampling can be applied to the whole data, in the previous approach we were taking out the most uncertain examples by learning from a single model but in prediction based sampling we consider the uncertainty values of the present model as well as from the previous models as well. In [30] concept of uncertainty sampling with prediction was proposed in the domain of **Meta learning.** In Meta Learning we usually associate the feature of learning problems with the performance of learning algorithm. Each of the Meta example stores features of the problem plus the prediction results performed by previous algorithms and on the basis of that it perform predictions for the future examples. A combination of **active Meta learning** techniques was used in their work and it was concluded that the combined approach gave significant increase in the performance of system but the complexity of the system got increased. The proposed work was done for the ranking and finding the weight of an example while it can also be used for other domains of active learning like error reduction etc.

In [42] the work of **probabilistic sampling** is performed. In their work they proposed a **multi-labeler model** which allows them to learn from the experiences of all the previous labelers and then sort out the most informative example. It was basically an iterative task in which they choose the best labeler and also the most inconsistent data set. They first select the most inconsistent point and then the labeler was selected for that by taking out the labeler with maximum confidence. Experimental results showed that the performance improves over the rate of learning. A limitation to their work was that they focused only on modeling and the empirical analysis while the problem of actively getting the feature labels can still be focused with this basis.

One of the latest work has been done by [55] has taken use of this approach in **spam filtering.** They have combined the concept of **active learning** with **incremental clustering**. The classification has been done by the **Naïve Based classifier** and the **K-Mean clustering** has been performed on the incremental level. The classifier creates a filter that receives input data as unlabeled instances and then for the first time the user labels them into clusters afterwards the labeling is done automatically until a level where the user is again consulted for labeling a new and unique instance. It was concluded that the new system of incremental clustering outperformed the base method which was labeling the email messages on a default set of training data.

3.4.5 Relevance Based Strategies For Sample Selection

The relevance based active learning is another term on which a lot of work has been performed. For the classification of Email, [25] has given a **relevance feedback** based active learning system in which the Labels are obtained for the limited amount of Emails and on the basis of that feedback, future Emails gets classified. The chief goal was to minimize the chances of an important Email to get dropped from the overflowed mailbox. The benefit of active learning in all this scenario is that the spam filters are being implemented dynamically thus reducing the chances of spam to almost none.

In [49] the relevance feedback was used for the purpose of query expansion. Basically an active learning approach was utilized for the query expansion which would be actually based on the user's **relevance feedback**. The documents were sampled according to the uncertainty values and then the documents with maximum uncertainty values were taken out for user's relevance feedback. On the basis of that feedback further classification was performed. The experimental results showed that their proposed system of informative documents on the basis of user feedback outperformed all the previous systems of digging out the informative documents. A limitation to their work was that, they were not focusing on the noise in their data which is in fact a major threat to the performance of overall system.

3.5 Summary

In this Section I have tried to briefly discuss the major domains in active learning in which research is performed and even continued. A lot of work has been conducted in the settings of **semi-supervise learning** as well as **supervise learning** and majority of the algorithm have been development for the purpose of **classification** and **regression.** As stated above, the main theme of active learning is to select the **most informative instance** for training the classifier. I have categorized the selection strategies for the sample (training data) selection under three major domains which are: **Error Reduction**, **Uncertainty Sampling** and **Relevance Based Selection Sampling**. For all these approaches the major part goes for the classifier which eventually shows either the approach was successful or not. The classifier which has captured the interest of majority of researchers is **SVM** and according to a statistical learning theory it is the best classification technique for **binary classification**. Apart from just binary classification, this classifier if merged with some other active learning approaches can give the best results in multi-class systems as well.

My area of research lies under the domain of **uncertainty sampling** with the help of **SVM.** Many sub-fields may come under the domain of **uncertainty sampling** like **Query by Committee, Margin Sampling,** and **Random Sampling etc.** Some of the recent work in the same domain has been listed below and from this table I'll try to summarize how my work is different and unique when compared to the previous approaches.

Author & Year	**Year**	**Approach**	**Results / Classifier's Accuracy**
S. J. Huang et. al. [37]	2010	QUIRE (Querying Informative and Representative example)	99%
S. J. Huang et. al. [37]	2010	Margin sampling	97%
S. J. Huang et. al. [37]	2010	Clustering Based Active Learning	79%
S. J. Huang et. al. [37]	2010	IDE (Informative and Diverse example selection)	89%
S. J. Huang et. al. [37]	2010	DUAL (Exploits Informativeness And Representativeness Of Query)	78%
P. Lindstrom et. al. [38]	2010	EAGL (Classifier Independent Sampling Method)	Area under curve decreases when replaced with APC
P. Lindstrom et. al. [38]	2010	PosBias (Postive Biased) Sampling	Performance curve rise
P. Lindstrom et. al. [38]	2010	Most representative	Performance decreases when classifier's parameter changed
P. Lindstrom et. al. [38]	2010	APC (Classifier Independent Sampling Method)	Area under the curve increased when replaced with EAGL
J. Zhu et. al. [39]	2010	Margin Sampling	Poor performance as no. the iteration increases
J. Zhu et. al. [39]	2010	(SUD) Sampling uncertainty + density	Performance increases when no. of training data increases
J. Zhu et. al. [39]	2010	Density based re-ranking	Overall performance was normal and its increased when number of

			instances were increased
J. Zhu et. al. [39]	2010	Uncertainty + density based re-ranking	Performance increases when no. of training data increases Percentage improved from SUD
P. Rashidi & D. J. Cook [43]	2011	Uncertainty Sampling	81% (for numeric + nominal data)
P. Rashidi & D. J. Cook [43]	2011	RIQY (Rule Induced Query method)	89% (for numeric + nominal data)
K. Liu & X. Qian [51]	2012	Random Sampling	76.1 %
K. Liu & X. Qian [51]	2012	Margin Sampling	81.2%
K. Liu & X. Qian [51]	2012	Uncertainty + Diversity Sampling	83.7 %
K. Liu & X. Qian [51]	2012	(TED) Transductive Experimental Design	72.6 %
F. Fukumoto et. al. [50]	2012	Error Correction based Uncertainty Sampling	59.7 %
F. Fukumoto et. al. [50]	2012	Positive Examples based Learning (PEBL)	58.4 %
F. Fukumoto et. al. [50]	2012	Random sampling	35.3 %
F. Fukumoto et. al. [50]	2012	PEBL + Error Correction	62.4 %
F. Fukumoto et. al. [50]	2012	Boosting	60 %
R. Wang et. al. [48]	2012	QBC (Query by Committee)	75.63 %
R. Wang et. al. [48]	2012	Co-SVM (Active Learning with 2 views)	76.13 %
R. Wang et. al. [48]	2012	Random Sampling	78 %
R. Wang et. al. [48]	2012	ALSVM (SVM Based Active Learning)	79 %
R. Wang et. al. [48]	2012	I-ALSVM (Inconsistency Based)	78.17%
R. Wang et. al. [48]	2012	QBC + I-ALSVM	80.41%
L. Hu et. al [54]	2013	Random Sampling	Low Performance on the curve
L. Hu et. al [54]	2013	Near Optimal	Moderate Performance shown on the curve
L. Hu et. al [54]	2013	Adaptive active learning	The performance curve shows best performance for object recognition
S. Sivaraman et. al. [58]	2014	Random Sampling	59% Precision
S. Sivaraman et. al. [58]	2014	Query By Confidence	60% Precision
S. Sivaraman et. al. [58]	2014	Query By Misclassification	81% Precision

TABLE 3.1: A Comparison of Previous Approaches in Active Learning

All the above mentioned literature review gets summed up to a point that active learning is all about selecting the best ever sample which is usually the training sample for our classifier. Some people have chosen this sample randomly from the unlabeled data; some have used measures like probability, entropy, density etc. Some have given some measures to get the most uncertain sample for training the classifier. So the most accurate your sample is, the more accurate your whole system would be. The most recent achievement of literature is the making of ensemble, a technique in which you add up different small sub techniques especially different classification system which adds up to the overall classification accuracy of the system.

CHAPTER IV

PROPOSED ARCHITECTURE

In this chapter I am going to give the description of the proposed method of this thesis which is used for getting the most informative Vector (MIV). The comparison of proposed method and previous method will also be made. The theoretical as well as mathematical description of all the classification and clustering algorithm used in the thesis will be given in detail. At the end, the summary of this chapter will also be given.

Before going into details of the proposed system, I'll discuss the main concepts of the base work of this thesis on which I have worked and brought some changes. The prior work done by R. Wang, S. Kwong [48] has been used as the baseline model for this thesis. In their work the process of active learning was taking place with the help of version space algorithm and the whole model was based on the binary classification which was performed by the SVM. I'll go into the details of SVM in the later sections of this chapter, first let me discuss the salient features of the version space algorithm which was the key component of my baseline work.

4.1 The Active Learning with Version Space

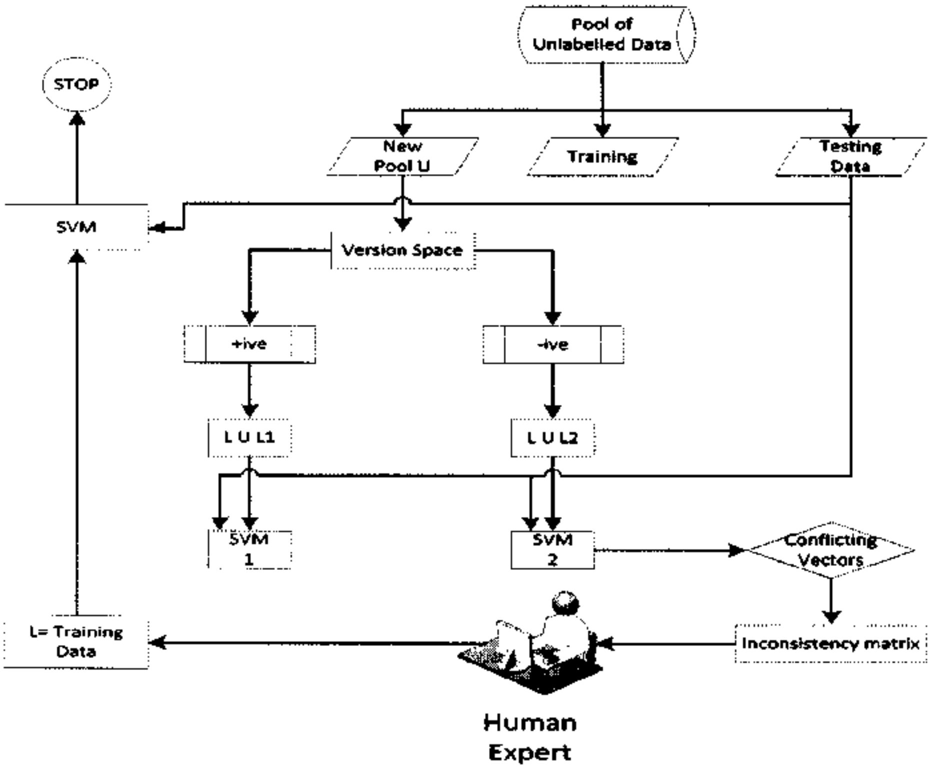


Fig 4.1: An Overview of Active learning with version space's architecture

An overall architecture of the baseline method has been showed in the above diagram. The first step of the preprocessing has been performed with the help of version space algorithm. The details of the version space will be given below but here the concept of version space has been applied for making the stages of general and specific. The whole data is divided into two stages as version space always works at the binary level. At one stage the whole pool of data is considered to be good/positive which is the general class of it and at the other stage the whole pool of data is considered to be negative. Now two separate SVMs have got trained from each of the two stages. After each SVM getting trained the same amount of test data is provided separately to both the SVMs and the results have been stored separately into matrices. The classification results of both SVMs are compared then and the instances which received conflicting results are stored in a separated matrix and the others are discarded. Now according to the formula of inconsistency the instances in this new matrix are allotted a new feature and this is their inconsistency value. The instances are then aligned in descending order according to their inconsistency value so that the instances with higher inconsistency value can come on the top. Now the number of training data is defined and given to the human expert for labeling. These labeled instances go as the training data for the final SVM classifier of the system.

4.2 The Active Learning with DIANA (ALDIANA)

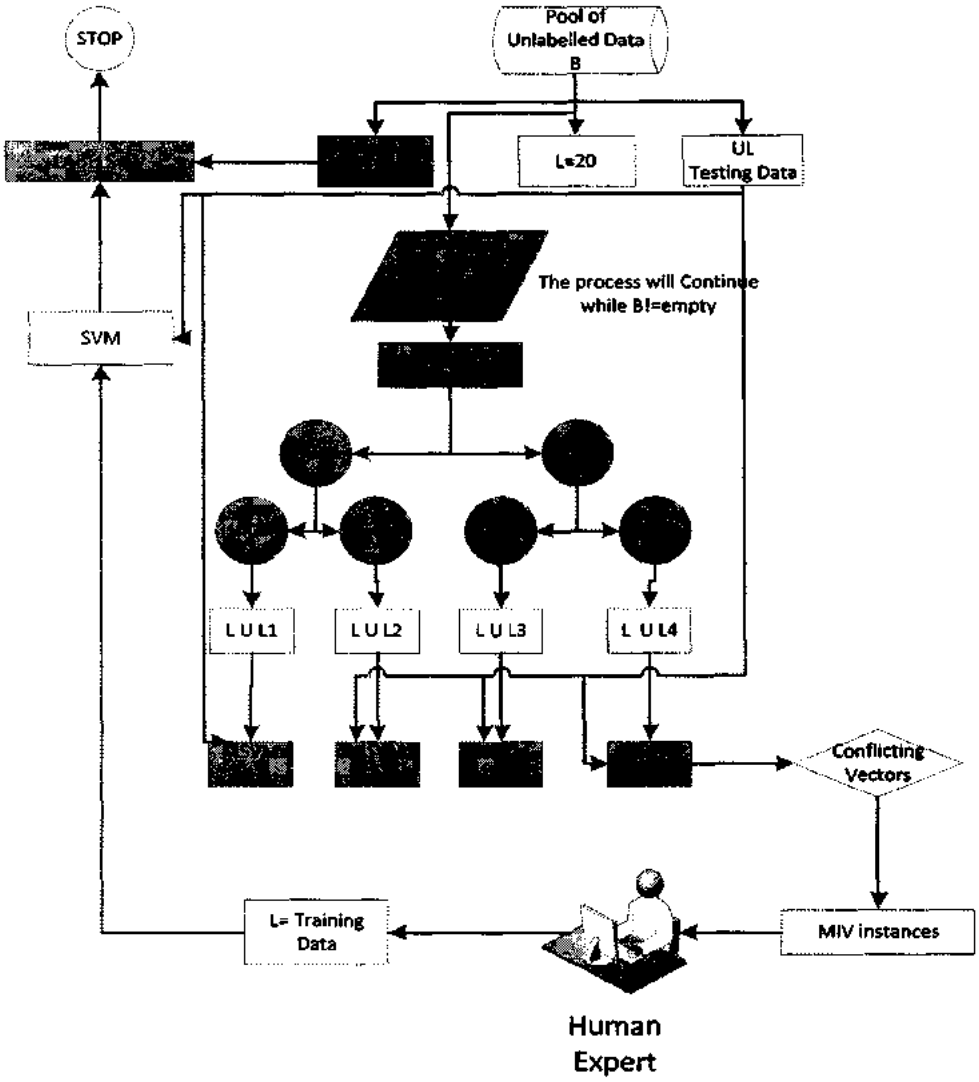


Fig 4.2: An Overview of the Active Learning with DIANA's architecture

The proposed architecture of my system has been displayed in detailed in the figure. The main difference between the proposed architecture and the base architecture can be clearly seen now especially in the preprocessing stage. The divisive analysis is applied for the preprocessing and according to the problem statement of my thesis, if the data is distributed on some logical basis rather than just randomly then the better results can be achieved. The Diana algorithm divides the whole pool of data into four clusters and then we take separate training data from all those clusters and train four separate multi-SVM classifiers. Now comes the turn of testing. For testing 30% of whole data was already taken out and this data is given to all multi-SVMs. The classification results are now compared and the instance which gets 50% of the conflict between the SVMs is taken out into a new matrix which is also called inconsistency matrix. The inconsistency matrix gets a new feature of inconsistency and then the instances gets aligned in the descending order and according to the manually decided amount of training data the instances go to the final SVM training and thus showing the overall classification results of the system.

In the later section I am going to give the details of those entire algorithms that I have used in my work.

4.3 The Version Space Algorithm

Version space is related to the field of concept learning which could be defined as the automatic inference of the general definition of the concepts; the concepts could be any objects, instances or examples on which the required work is to be done. We can simply call a concept as the set of all positive examples. The concept of version space was introduced by Mitchell [1*] and according to that we divide the whole dataset into two hypothesis. One is completely positive which is also termed as general category and the other one is completely negative also known as specific category. The version space search also precedes General to the specific concept. In version space we assume that if we are discussing some particular hypothesis then it must be related to that one group specifically, any member from the other group or relating to other group must not be present in that group.

The overall algorithm of version space works by the steps that I have given below.

1. In version space we first of all create a complete lattice of an overall concept and from that lattice we further search inside according to the mentioned terms. i.e if we have made 2 groups of flower (lilies and Tulips). At the beginning of process there will be just two lattices that will be having flowers of their own type only now further the searching will proceed for the types of both these flowers.

2. In the second step the process goes for search related to the fore mentioned terms. i.e. In our example we have seen that there are 3 types of lilies present (Calla lilies, Blue grass

lilies, tufted blue lilies) and 3 types of tulip (fringed hybrid tulip, Darwin hybrid tulip, parrot hybrid lilies).

3. Now the overall data will get divided into 2 hypothesis and all the other data will either go for first hypothesis (TULIP) or for the second hypothesis (LILLY).
4. From the diagram given below we have seen that a hierarchy from the two main hypothesis have been created but this can also create a finer grained generalization/specialization lattice for this version space.

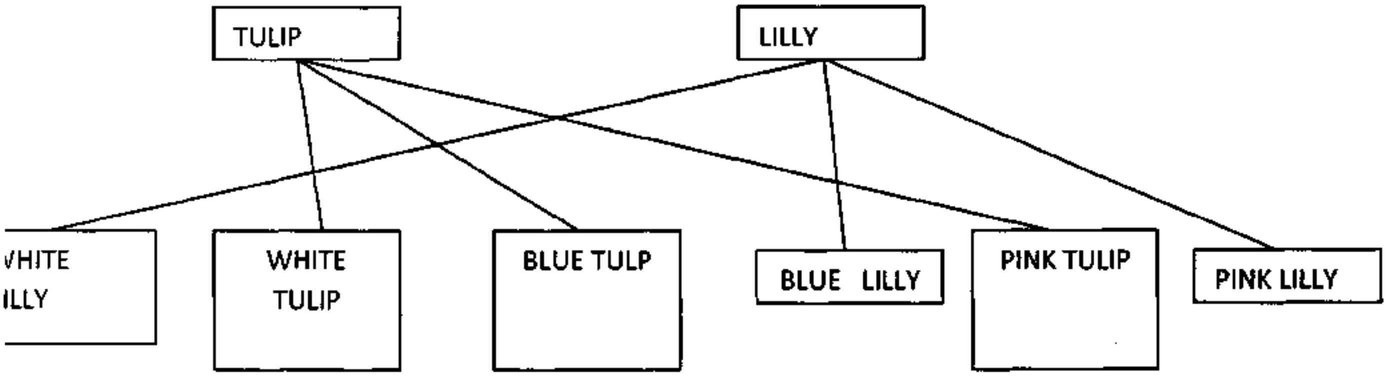


Figure 4.3: The Version Space Example

Now from the above mentioned diagram of version space, the overall process for the new and unlabeled instances falls into two main rules for the version space

- If the new instance matches all the details of the TULIP class then it must be stored totally in the TULIP hypothesis.
- If the new instance matches all the details of the LILLY class then it must be stored totally in the LILLY hypothesis.
- Any other case must be either a LILLY or a TULIP.

A slightly different approach of version space has been applied in the base work of this thesis, according to that the classifier has got trained on a single dataset but two assumption have been made for training two different SVM classifiers. For one training data they assumed that all the data is positive (according to version space's specific hypothesis) and trained an SVM classifier on that data. For the 2nd SVM classifier they trained it on that data in which they assumed that all data is negative (according to version space's general hypothesis). One limitation in that version space concept was that, the training of the classifiers was made on totally assumed concepts and there was no practical or factual correctness in the data on which those SVMs were trained.

In comparison to that afore mentioned version space concept I have proposed the concept of pre-clustering with divisive analysis clustering algorithm which was a more factual division of data rather than the conceptual assumptions.

Now coming back onto the main focus of this thesis, active learning which was helping us to get the most informative vectors. As discussed in Chapter 3 a lot of work has been performed in the domain of active learning, this approach of using the minimum amount of training data got its attention from the researcher in 1980's and [2] was one of the pioneer work of this domain. According to [2] the learner was given the freedom to ask queries regarding labeling of its data. This freedom further leads to many fruitful results and the major one among them was the improved accuracy of the system. The advent of active learning gave rise to a new terminology 'Active Learner' which is discussed in the section below.

4.4 Active Learner

In the introduction of this thesis I have called the active learner as a student who continuously increases the level of his information gain by asking question and participating in the learning process rather than collecting whatever amount of data comes to him. Actually this quality of active learner differentiates him from the passive learner as the questioning is actually the response to the knowledge gained by the teacher and by questing the learner can excel in the field in which he wants to excel. The basic algorithm for an active learner would be as under:

```
Input:  An initial training set L, an unlabelled pool U, a selection strategy S,
Output: A labeled set or a classifier
        Selected = $\emptyset$;
        Choose b most informative examples using S;
        Add the b examples to Selected;
        Label each example $x^i \in$ Selected;
        L = L $\cup$ Selected, U = U/Selected;
        End
```

This is the simplest form of an active learner which is choosing the most informative examples and then asking for the labels of those examples and adding them to the training data. The initial training data is denoted by L, the pool of unlabeled data is denoted by U, the selection strategy by S. 'b' are the number of examples which the learner is going to take from U on the basis of S which could be any selection strategy used in the process of active learning.

Active learner's distinguishing quality is that it always detects the most informative instances from the data which are also as minimum in amount as possible and then asks the users/human expert to label that data thus from such efforts it actually tries to minimize the users effort for the labelling of data.

The selection strategy of an active learner mostly revolves around two concepts: one is Query Construction and the other is Selective Sampling

4.4.1 Query Construction

In the query construction an arbitrary value is given to a query which is then forwarded to the expert for labeling. The arbitrary value is mostly the extreme possibility of any situation and on such query the learner gets trained. For example if we have to classify some document then we'll add either exact keyword in the arbitrary query that are required to keep that document as a member of certain class or we'll give it extreme negative keywords that might be little bit related to the keyword of certain class but not exactly. Query construction is not applicable in most of the classification problems as it is based on the system's/ experts knowledge rather than being based on the practical facts found in the data.

4.4.2 Selective Sampling

A relatively more practical approach than query construction is the selection sampling. This approach proceeds by selecting the query from the large pool of unlabeled data. I have called it more practical because in this approach the learners' choses the queries from the dataset provided and then give it to the expert for labeling. Thus the labeler is now working on a practical data and it can be more helpful than some arbitrary data.

In this thesis I have also focused on the selective sampling technique and I have applied a pre-clustering technique before coming to the selective sampling. The Clustering technique which I am going to discuss in the next section actually divides the data into as much clusters as possible and then the query will be selected by taking one two or more members from each cluster. The benefit of using pre-clustering is that it has made my active learner so capable that now it will take member from each data distribution thus member of every variety present in the data will now get labeled and the classifier will get trained on it.

4.5 Divisive Analysis (DIANA)

In this thesis I am going to work on a hierarchical clustering algorithms, the major difference between simple clustering and hierarchical clustering is that in the prior one we just divide the data into some number of groups that are based on the similarity between objects but in hierarchical clustering we build a proper hierarchy of objects. The unique feature of this thesis is the pre-clustering in which the divisive analysis-s algorithm is used. In this algorithm we create a hierarchy of clusters, the traditionally used clustering is not applied here because we want to get as much clusters as the level of resolution among the data allows us. Another reason for not using simple clustering methodology is the initialization of number of clusters at the start of overall process (About which we can't be sure at the start of the process). In simple word we can say that hierarchical clustering gives us the freedom of choosing the N number of steps that can give us the convenient number of clusters for our analysis.

The DIANA algorithm is applied here to check for the inter cluster similarity among the two are more chief clusters. In DIANA the hierarchy is created in the inverse order, we start from the most general form in which we have 2 clusters and then from those clusters we move on to as much amount of clusters as possible. The initial step of DIANA is that we have one clusters which contains all the instances or technically we can say that it has n number of objects. On every posterior step the larger clusters gets split up into two clusters and this process continues until every object come sin its own cluster. The whole hierarchy in DIANA is built up in **N-1** steps.

The Overall algorithm of DIANA proceeds as follows.

1. Get the object having highest level of dissimilarity with all other objects and this will be our splinter group.

2. For every object name as 'i' compute the following formula

   $$D_i = [average\ \ d(i,j)\, j \notin R_{splintergroup}] - [average\ \ d(i,j)\, j \in R_{splintergroup}]$$

3. Let's suppose we have an object h for which we have to calculate the distance D_h by the above mentioned formula. If the value of D_h is largest and also if it is positive then we can say that h is close to the splinter group but on an average.

4. Now we'll have to repeat the 2nd step until we get all the values of D_h to be negative. Now this whole pool will get divided into two groups.

5. The cluster which will be having the largest diameter will get selected now this is our largest dissimilarity between any of the two objects. Now it's the turn of this largest cluster to get further divided.

6. All the above steps will kept on being repeated until we get one object in each cluster.

The Diana algorithm can be plotted with the help of a dendrogram and an example of such diagram is given below

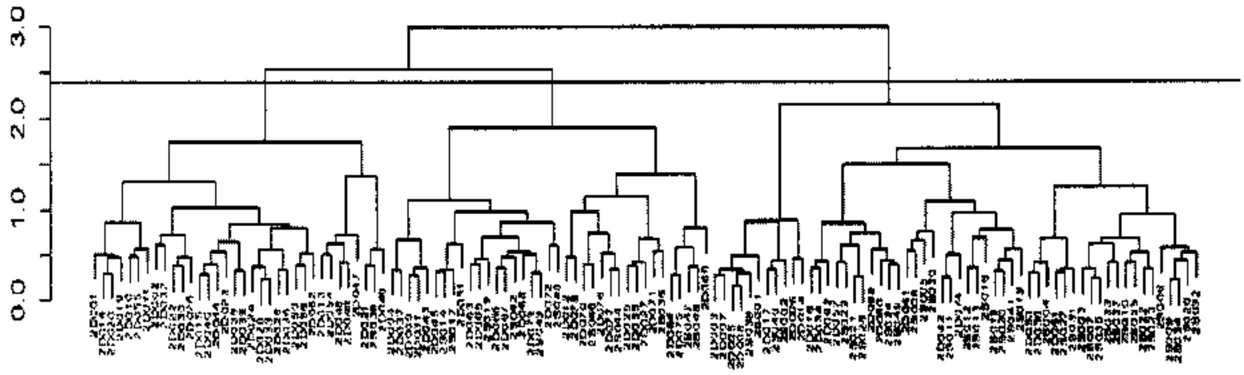


Figure 4.4 : A Dendogram Example of DIANA

After the pre-clustering phase the algorithm moves on to the classification of all those clusters that were create by DIANA. For this thesis I have also brought a change into the previous base system and that is the usage of support vector machine for more than two classes. The multiSVM has been discussed in detail in the section given below.

4.6 Multilevel classification via support vector machine

The SVM by its definition is the binary classifier but it is not much difficult to extend it, we just need to combine its approaches into a multiclass classifier. It has been seen in most of the cases that SVM is being used for the binary classes but it's also a fact that the binary classification tends to be very limited. A positive aspect in this case is that many researches have been proposed in the history to tackle the problem of multiclass classification with SVM. First of all I will try to give the brief introduction of the working of SVM and after that I will discuss the approaches of multiSVM

The support vector machine was basically designed for the binary classification of data but it is also possible to change the level of classification from binary to multiclass. In the past many researches have taken place regarding this domain and even today it is a very hot topic for the research.

In this thesis I have also used the multiclass SVM for the purpose of changing the overall classification system from binary to multiclass. In this section I'll give the description of main classifier that I have used for the classification but before that let me discuss the root of it which is the main SVM classifier.

4.6.1 Support Vector Machine

The classifier was developed in 1995 by cortes and vapnik and since then it has got the fame as the best classifier for the binary classification of data. The main conception behind the SVM is to stick out the whole data on a high dimension space and try to bring a maximum margin hyperplane among the both sets of data.

The Hyper Plane

The hyperplane of SVM is not specified in terms of number, there could be any number of hyperplane created for the data points and they are classified into thin and thick hyperplanes. In the SVM classification the key aim is to find the hyperplane with following characteristics.

- One which is linearly separable
- One whose margin is the largest.

For calculating the hyperplane we need following constraints.

- x i which is the vectors that contains the attribute values of all the instances i
- W is the vector containing weights of all attributes
- For representing the y intercept we create a real number b
- We set the decision boundary on he points such that the following equation proves true

$$x\,.\,w + b = 0$$

Let's say we have two points that lie on the decision boundary then

$$x\text{a}\,.\,w + b = x\text{b}\,.\,w + b = 0$$

Thus we can also say that

$$w\ (x\text{b} - x\text{a}) = 0$$

Where we know that both these points are parallel to the decision boundary

The main formula for the hyperplane as generated by w and b is as under:

$$f(x_i) = x_i.w + b$$

Now if can say that for some point if x i . w + b> 0 then it will lie above the hyperplane and if

x i . w + b <0 then it will lie below the hyperplane. We can represent the classes as 1,0 and they can written in the form

$$Y = \{0,\ \text{if } x\text{i}\,.\,w + b > 0\}$$

$$\text{Y} = \{1, \text{if } x\text{i}\,.\,w + b < 0\}$$

Now we can name these points as the support vectors.

Distance

The distance between the margins and the decision boundary could be given by the following rule

$$D = \frac{2}{\|\vec{w}\|}$$

Throughout the learning of our SVM we need to estimate the parameters w and b.

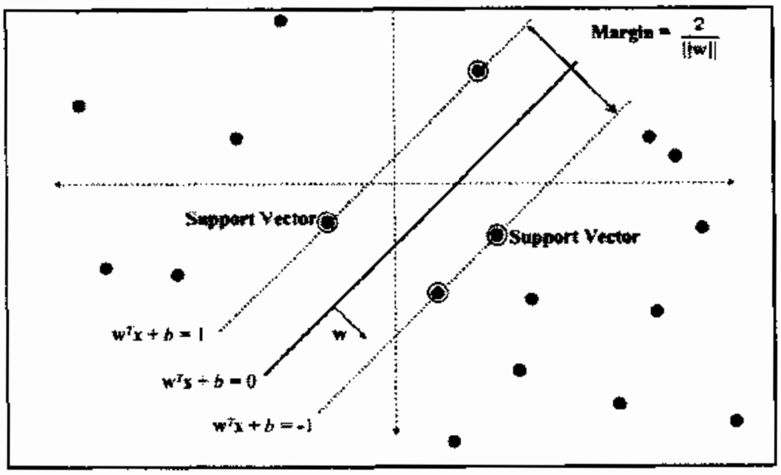


Figure 4.5 : The SVM Hyperplane

As mentioned above the key criterion behind SVM classification is first see that all points are classified correctly

$$x i . w + b \geq 1 \text{ if } y = 1$$

$$x i . w + b \geq -1 \text{ if } y = 0$$

Now as mentioned above, SVM states another criteria, according to which the margin must be as large as possible and this thing can be achieved by minimizing the following formula

$$f(w) = \frac{1}{2} \|\vec{w}\|^2$$

The minimization is carried out by fulfilling the following constraint

$$Y_i(\vec{w}.\ x_i + b \geq 1) \ \ for \ \ 1 \leq i \leq N$$

The optimization or the constraint minimization is solved with the help of Lagrange multipliers

The formula for Lagrange multipliers is as under:

$$L(\vec{x},\lambda) = f(\vec{x}) + \sum_{i=1}^{m} \lambda_i g_i(\vec{x})$$

Two steps are required to solve the Lagrange multiplier

- $\frac{\partial L}{\partial xi} = 0 \quad for \quad 1 \le i \le n$
- $\frac{\partial L}{\partial xi} = 0 \quad for \quad 1 \le i \le m$

4.6.2 MULTISVM

The multisvm dilemma follows the above mentioned conceptions of the support vector machine but it is further extended to certain approaches. Some of the common approaches for the multisvm are

- One Against one
- One against All
- DAGSVM

1. **ONE AGAINST ONE**

This is one of the pioneer approach for the multisvm that was introduced in [3]. According to this approach we build *k (k*-1)/2 classifiers and we train each classifier with the data from two classes. The classifiers are the simple SVM classifiers which are build up by the same process as I have mentioned in the previous section. Now for training the data from both classes i.e. ith and jth class, we are supposed to solve the following binary classification problem.

$$\min_{w^{i,j}, b^{i,j}, \xi^{i,j}} \quad \frac{1}{2}(w^{ij})^T w^{ij} + C\sum \xi_t^{ij}$$

$$(w^{ij})^T \phi(x_t) + b^{ij} \ge 1 - \xi_t^{ij}, if \quad y_t = i$$

$$(w^{ij})^T \phi(x_t) + b^{ij} \le 1 - \xi_t^{ij}, if \quad y_t = j$$

$$\xi_t^{ij} \geq 0$$

There could be many methods for the future testing as we know the fact that k (k-1)/2 classifiers are constructed. The best and most common one is the voting strategy, according to that if the equation says $((w^{ij})^T \phi(x) + b^{ij}))$ x is the ith class then one is added to the votes of ith class otherwise the jth class is increased by one. This voting strategy is called the 'Max Wins' strategy. For the situation where we get identical number of votes for the two classes we select the one with the smaller index although we can't call this an accurate strategy.

2. One Against All

The previous approach for the SVM was dealing with binary classification problem but this is probably the earliest approach of multisvm that was dealing with multiple classes. In this approach we construct k SVM models which deals with k number of classes. Let's say we have an ith SVM which is trained on ith class that has all number of examples positive labels and all other examples with negative labels. Now if we are provided with the training data l which is in the form $(x_1, y_1)........(x_i, y_i)$ where $x_i \in R^n$, $i = 1......l$ and $y_i \in \{1,......,k\}$ which is actually the class of x_i. The ith SVM will now solve the following problem.

$$\min_{w^i, b^i, \xi^i} \quad \frac{1}{2}(w^i)^T w^i + C\sum_{j=1}^{l} \xi_j^i$$

$$(w^i)^T \phi(x_j) + b^i \geq 1 - \xi_j^i, if \ \ y_j = i$$

$$(w^i)^T \phi(x_j) + b^i \leq -1 + \xi_j^i, if \ \ y_j \neq i$$

$$\xi_j^i \geq 0, j = 1,.....,l$$

C is the penalty parameter in the above mentioned equations and the function ϕ is actually mapping the data x_i on the higher dimensional space. In the above equation the main part is minimizing the $\frac{1}{2}(w^i)^T w^i$ which actually shows that we should maximize the $2/\|w^i\|$ margin between the two groups of data. Now the main aim behind the SVM classification is to search for a balance between the regularization term $\frac{1}{2}(w^i)^T w^i$ and the errors obtained while training the data.

After finishing with the above mentioned problem the *k* decision functions are checked.

$$(w^1)^T \phi(x) + b^1$$

$$\vdots$$

$$(w^k)^T \phi(x) + b^k$$

Now we say that *z* is the class which has the largest number of decision functions

$$class\ of\ z \equiv \arg\max_{i=1,\ldots,k}((w^i)^T \phi(z) + b^i)$$

3. DAGSVM

Another approach for multisvm which I am going to discuss in this thesis is the Directed Acyclic Graph Support Vector Machine (DAGSVM). The SVM training in this approach is exactly same as one to one approach which uses *k (k*+1)/2 binary SVMs but in the testing phase it completely differs itself from the previous approach by utilizing a rooted binary directed acyclic graph which contains *k (k*+1)/2 binary nodes and *k* number of leaves.

We can call each node as the binary SVM for the ith and jth class respectively. As this approach is more theoretical than the previous approaches and because of this, it has an advantage that it can establish some sort of analysis for the sake of generalization.

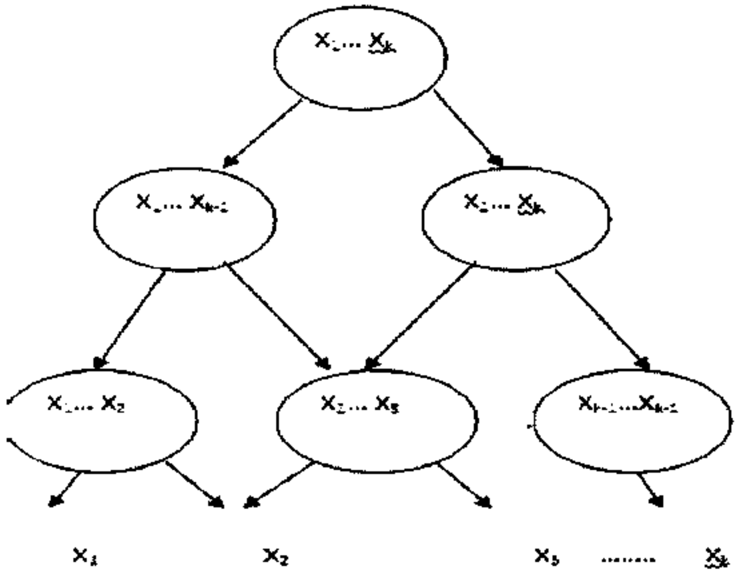


Figure 4.6: The Decision Tree Structure of DAGSVM

These were the three most common approaches of multisvm, for my thesis I have utilized the approach of one against all multisvm. The one against one and DAGSVM were both dealing with the binary classification problems, the DAGSVM can also be used for multiple classes but the graph creation at times can get complex. Because of these reasons I have chosen to use the One against All approach as my problem area is related to multiple classes.

CHAPTER V

EXPERIMENTAL RESULTS

In this chapter I am going to discuss the results of my thesis and the detailed comparison of proposed technique (Active Learning with DIANA) and previous technique (Active Learning with Version Space) and both are used under the domain of active learning. The experimental results of my work have shown that the active learning approach of most informative vector selection can outperform the traditional approach of passive learning. The main point of focus is the concept of version space which is replaced with a more logical divisive analysis (DIANA) approach.

In the start of this chapter I am going to discuss the performance measures being utilized in this thesis. The use of performance measures is an important part of any information retrieval system as they tell us in detail about the useful of the system which has been developed in order to check the performance of the classifier being used.

5.1 Performance Measures Used

5.1.1 Confusion Matrix

This type of matrix has many names like contingency table, errors matrix, matching matrix etc and it is applicable for both supervised learning systems and the unsupervised systems. The main work is to give a visual overview of the overall performance of any algorithm. The performance is measured by comparing two classes, the actual class and the predicted class. This matrix very beautifully compliments its own name because in this matrix we can actually visualize if our algorithm's results are confusing between two classes or not.

In confusion matrix the predicted values of a classifier are compared with the ground truth values and this comparison gives us a clear vision about the performance of our classifier's prediction. A confusion matrix is drawn in the form of the table that is shown below:

		[illegible]	
		POSITIVE	NEGATIVE
[illegible]	POSITIVE	TP	FN
	NEGATIVE	FP	TN

Figure 5.1: Confusion Matrix

In this matrix the two classes are supposed to be positive and negative. The data is distributed in the 4 boxes which are shown by the terms TP, FN, FP and TN.

- TP refers to the term true positive and it shows the amount of values which are actually positive and our classifier has also predicted them as positive.
- FN refers to the term false negative and it shows the amount of values which are actually negative but classifier has not predicted them to be negative.
- FP refers to the term false positive and it shows the amount of values which are actually positive but the classifier has not predicted them as positive.
- TN refers to the term true negative and it shows the amount of values which are actually negative and the classifier has also predicted them as negative.

The same idea for confusion matrix can also be used for more than 2 classes i.e. in the table given below the confusion matrix for three classes has been shown.

		[illegible]		
		A	B	C
[illegible]	A	TP_A	E_{AB}	E_{AC}
	B	E_{BA}	TP_B	E_{BC}
	C	E_{CA}	E_{CB}	TP_C

Figure 5.3: The confusion matrix of 3 classes

This confusion is also conceptually similar to the previous one. On the diagonal of it the actual classification performance of the classifier can be seen which is shown in the form of true positive. The false predicted value of any class can be analyzed by adding all the E values for that class i.e. for the F predicted values of A we'll have to add the E_{AB} and E_{AC}. The performance measure which I am going to discuss in the later sections will be calculated with the help of true predicted and false predicted values of the confusion matrix.

5.1.2 Precision

As I have mentioned in the previous sections that my work is related to the multi class classification, precision by itself is a performance measure for binary classifier but it can also give its best result for the multi class classification. First let me discuss the main idea for the precision then I'll describe the way by which I have used it for multi class SVM. Just like its meaning, precision tells us the fraction of retrieved instances that shows relevance with the positive class.

By the definition we call precision as the ratio of all those instances which are correctly predicted by the classifier as the positive ones. The formula for precision is as under:

$$\frac{TP}{TP+FP}$$

As I have discussed the confusion matrix of multi class classifier, according to that the precision for a particular class i.e. A would be

$$\frac{TP_A}{TP_A+e_{BA}+e_{CA}}$$

5.1.3 Recall

Recall which is also known as sensitivity checks for the strength of classifiers probability to select instances of a particular class from the whole dataset. Recall mainly corresponds to the true positive rate. The formula is as under:

$$\mathrm{Re}\,call = Sensitivity = \frac{TP}{(TP+FN)}$$

The above formula is used when we have to deal with only 2 classes, for any example when we have to find the recall rate for some class then the formula will be as under:

$$\mathrm{Re}\,call_A = Sensitivity_A \frac{TP_A}{(TP_A+e_{AB}+e_{AC})}$$

Here we have calculated the number of correct results obtained divided by the total number of records that should have been calculated.

5.1.4 Specificity

This is the performance measure which is opposite to sensitivity and it is also referred as true negative rate. It calculates the total number of correct instances that are found to be negative. The formula is as under:

$$Specificity = \frac{TN}{(TN + FP)}$$

For more than 2 classes let's suppose we have to calculate the specificity then the formula would be as under:

$$Specificity_A = \frac{TN_A}{TN_A + e_{BA} + e_{CA}}$$

5.1.5 Kappa Coefficient

This is a very important and useful performance measure which compares the actual accuracy of the system with a random accuracy. According to R. Landis and G. Koch, "Total accuracy is an observational probability of agreement and (random accuracy) is a hypothetical expected probability of agreement under an appropriate set of baseline constraints."

The Formula for Kappa is as under

$$Kappa = \frac{Total\ Accuracy - Random\ Accuracy}{1 - Total\ Accuracy}$$

The formula to calculate the total accuracy is as under:

$$Total\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The Formula for Random Accuracy is as under:

$$Random\ Accuracy = \frac{(TP + FP)^{*}(TN + FN) + (FN + TP)^{*}(FP + TP)}{Total * Total}$$

5.2 Experimental Setup

For the demonstration of the results achieved by the proposed method, two datasets have been used. The datasets were taken from the UCI machine repository and the details of both datasets are as under:

5.2.1 User Knowledge Level Dataset

Dataset Details	
No. of attributes	5 Plus class
No. of instances /vectors	259
Attribute Characteristics	Integer
Missing Values	No
Variables to be predicted	Knowledge level of use (very low, low, middle, high)

TABLE 5.1: Details of User Knowledge Level Data

The user Knowledge data was taken from undergraduate students of Department of Electrical Education of Gazi University in the 2009 semester and the algorithm to find the user knowledge class was developed by Hamdi Tolga Kahraman, Ilhami Colak and Seref Sagiroglu of the same university.

The Attributes values are as under:

1. STG (The degree of study time for goal object materials), (input value)
2. SCG (The degree of repetition number of user for goal object materials) (input value)
3. STR (The degree of study time of user for related objects with goal object) (input value)
4. LPR (The exam performance of user for related objects with goal object) (input value)
5. PEG (The exam performance of user for goal objects) (input value)
6. UNS (The knowledge level of user) (target value)

Very Low:	50
Low:	129
Middle:	122
High:	130

5.2.2 Pima Indian Diabetes

Dataset Details	
Total Attributes	8 Plus class
Missing values	Yes
Instances/Vectors	768
Attribute types	Numeric
Variables to be detected	Presence /Absence of Disease

TABLE 5.2: Details of Pima Indian Diabetes Dataset

This database has been obtained from the UCI Machine Repository and the original owner of the data is the National Institute of Diabetes and Digestive and Kidney Diseases.

The Class values are set to be binary i.e. 0 and 1 only. The indicative, '0' or '1' valued variable indicate either patients shows signs of diabetes as per to world health organization criteria (that is as a minimum 200 mg/plasma post load in two hour at any survey examination). The population lives near Arizona , USA and Prediction made by algorithm was in the range of 0-1. This was transformed into a binary decision using a cutoff of 0.448. All patients includes in database are women of minimum 21 year old.

Attributes label of diabetes database are frequency of pregnancy, concentration of the plasma glucose is tested in 2 hour using oral glucose tolerance test. 3. blood pressure (mm Hg)(Diastolic),skin fold thickness of the triceps, serum insulin ,index f the body mass, and pedigree function of diabetes ,age in years of patient, and last is the class variable for diagnosing sign of diabetes.

5.3 Results and Discussion

In this section I am going to elaborate the outcomes of the whole system, the results have been obtained by implementing different iterations of the whole algorithm. The data has been specified into proportions of testing and training data in each iteration. The comparisons in the results has been made by comparing the results of same data on both base algorithm which is using version space and the proposed algorithm which has used Divisive analysis(DIANA).

5.3.1 Training and Testing Data

The training and testing data selection has been done manually and for both of the datasets, I have taken 20 random instances for training the classifier for the first time and 100 instances have been selected for testing from UKL dataset and 168 have been randomly selected as testing data from Pima Indian diabetes dataset. The training data continuously keeps on getting updated by the new instances which have been considered most informative and labeled by the human expert.

5.4 Classification Results

The classification results of the system have been obtained in different iterations with different volumes of the training data 'L'. The DIANA algorithm always selects the centroids by some selection criteria or randomly. In this work I have given a manual selection criteria by giving the centroids of two extreme classes, High and Low. After that the second level of clustering runs from the knowledge of first one. The Classification of Active Learning with DIANA gives different accuracies when run for the same amount of data for more than one time. I have calculated the results of my classifier by following ten iteration for each volume of 'L' and then the results in table 5.1 are given as a mean of the results from those 10 iterations. A positive thing about the results is that the accuracy of overall system is still better than the previous system. Here comes the difference of AL with DIANA and AL with Version Space, the DIANA algorithm gives different values in each iteration (as centroids are selected differently in each iteration) but the AL with version space gives same result no matter how many iterations are performed. From this a limitation into the base system it can be seen the Active Learning with DIANA gives a very smart technique to deal with the sparseness of data. I have demonstrated by classification results in 3 different forms and their details have been listed below.

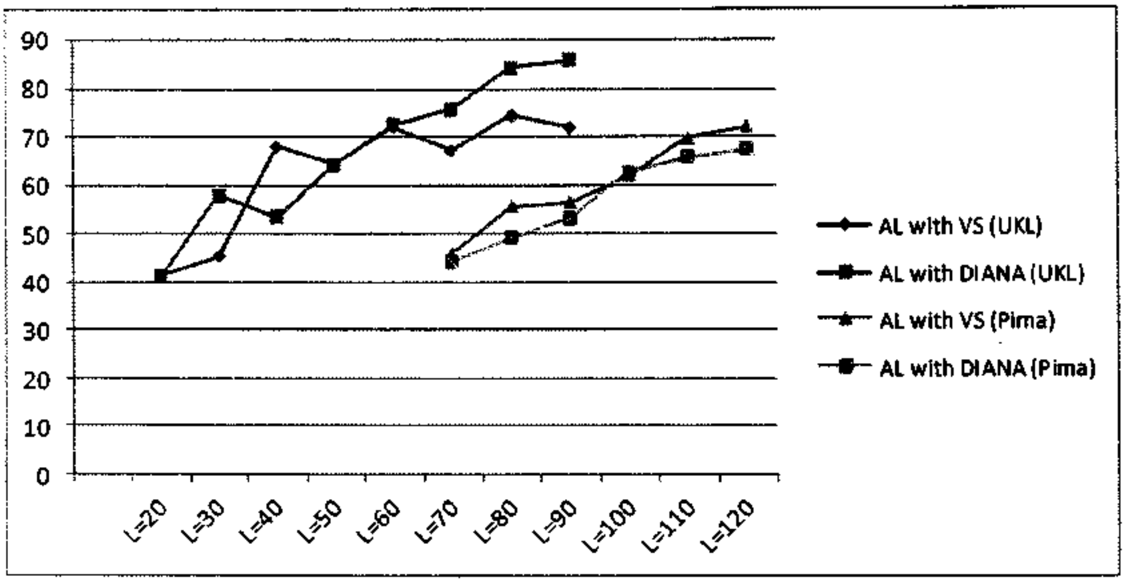


Figure 5.3: Overall performance curves on both types of data

5.5 Over All Classification Accuracies of Both Systems on UKL Data

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Active Learning With Version Space	41.25 %	45.33 %	68.06 %	64.29 %	72.06 %	67.16 %	74.60 %	72 %
Active Learning with DIANA	41.25 %	57.87 %	53.37 %	64.25 %	72.5 %	75.75 %	84.25 %	85.60 %

TABLE 5.1: Classification Accuracy on UKL data

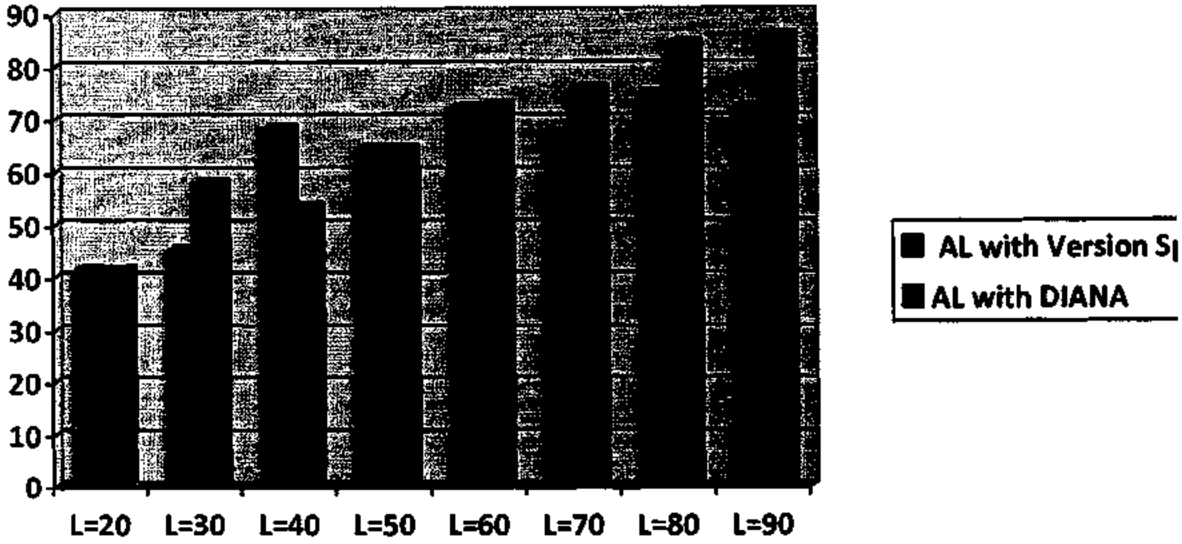


FIGURE 5.4: Bar Chart representation of both systems on UKL dataset

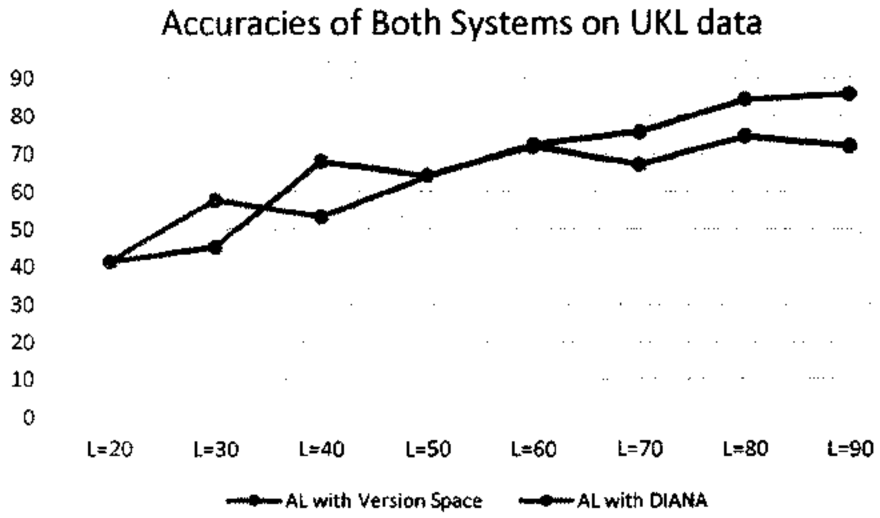


FIGURE 5.5: Performance curve of both systems on UKL dataset

From the above results it is very noticeable that the amount of accuracy is increasing with the amount of 'L' and this behavior is pretty obvious as the more labeled data you'll give to the system the more accurate the classification would take place. The point to be notice here is that the amount of classifier's accuracy of the active learning with DIANA (ALDIANA) is more than the active learning with version space classifier (ALVS). This counts up to the success of our proposed method

(ALDIANA) which proposed that the accuracy will increase if we'll go for the logical discrimination between the data rather than just following a blind assumption.

From the results another thing is also very noticeable, at the amount of L=20 the accuracies of both the systems were same and the obtained accuracies after following 10 iterations also came out to be same. The reason behind it is that the amount of initial data given to the classifier is 20 and when we have defines 'L' to be just 20 then no space left for the most informative vectors and thus no model has been learnt and that's why the results of both systems are same. After L=20 the accuracies of both systems start to vary from each other, for ALDIANA it can be observed that the accuracies are pretty consistent and getting increased.

The sections given below will be showing the details of the actual values and predicted values for each volume of 'L' and from there the performance of both systems can be observed in depth.

5.6 The Classification Details of UKL Data

In this section I am going to give the detailed description of the results obtained for different volumes of L. The actual Values of the classes present in the data have been shown and after that the predicted values have been mentioned along with the percentage of precision and accuracy. The Kappa rates have also been mentioned.

5.6.1 When L=90

CLASS	**Actual Value**	**Predicted Value**	**Precision**	**Recall**
VERY LOW	16	15	100%	93.75
LOW	23	26	76.92%	86.95%
MIDDLE	34	29	87%	76.47%
HIGH	7	10	70%	100%
Overall Accuracy		85%		
Kappa		0.787		

TABLE 5.2: ALDIANA values for L=90 (UKL)

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	19	13	100%	68.41%
LOW	5	19	26.31%	100%
MIDDLE	32	21	100%	65.62%
HIGH	5	8	62.6%	100%
Overall Accuracy		72.13%		
Kappa		0.611		

TABLE 5.3: ALVS values for L=90 (UKL)

5.6.2 When L= 80

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	17	15	100%	88.25%
LOW	25	26	84.61%	88%
MIDDLE	31	29	89.65%	83.87%
HIGH	7	10	70%	100%
Overall Accuracy		87.5%		
Kappa		0.823		

TABLE 5.4: ALDIANA values for L=80 (UKL)

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	20	14	100%	70%
LOW	7	19	36.84%	100%
MIDDLE	32	22	100%	68.75%
HIGH	4	8	50%	100%
Overall Accuracy		74.60%		
Kappa		0.643		

TABLE 5.5: ALVS values for L=80 (UKL)

5.6.3 When L=70

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	20	15	100%	75%
LOW	25	26	73.07%	76%
MIDDLE	27	29	75.86%	81.48%
HIGH	8	10	70%	87.5%
Overall Accuracy		78.75%		
Kappa		0.704		

TABLE 5.6: ALDIANA values for L=70 (UKL)

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	22	14	100%	63.63%
LOW	5	22	22.72%	100%
MIDDLE	37	23	100%	62.16%
HIGH	3	8	37.5%	100%
Overall Accuracy		67.16%		
Kappa		0.539		

TABLE 5.7: ALVS values for L=70 (UKL)

5.6.4 When L= 60

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	19	15	100%	78.94%
LOW	25	26	73.07%	78.57%
MIDDLE	28	29	75.86%	78.57%
HIGH	8	10	70%	87.5%
Overall Accuracy		78.75		
Kappa		0.703		

TABLE 5.8: ALDIANA values for L=60 (UKL)

L=60 (Base Method)				
CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	21	14	100%	66.66%
LOW	7	22	31.81%	100%
MIDDLE	36	24	100%	66.66%
HIGH	4	8	60%	100%
Overall Accuracy		72.05%		
Kappa		0.606		

TABLE 5.9: ALVS values for L=60 (UKL)

5.6.5 When L= 50

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	20	16	93.33%	70%
LOW	16	26	60%	66.66%
MIDDLE	37	29	89.66%	70.27%
HIGH	8	10	70%	87.6%
Overall Accuracy		75%		
Kappa		0.649		

TABLE 5.10: ALDIANA values for L=50 (UKL)

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	29	14	100%	48.27%
LOW	2	23	8.69%	100%
MIDDLE	35	25	100%	71.42%
HIGH	4	8	50%	100%
Overall Accuracy		64.28 %		
Kappa		0.506		

TABLE 5.11: ALVS values for L=50 (UKL)

5.6.6 When L= 40

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	28	15	93.33%	60%
LOW	24	26	46.16%	50%
MIDDLE	11	29	34.48%	90.99%
HIGH	17	10	90%	62.94%
Overall Accuracy		56.25 %		
Kappa		0.42		

TABLE 5.12: ALDIANA values for L=40 (UKL)

L=40 (Base Method)				
CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	28	14	100%	50%
LOW	4	23	13.04%	75%
MIDDLE	32	27	92.59%	78.12%
HIGH	8	8	87.5%	87.5%
Overall Accuracy		68.05%		
Kappa		0.561		

TABLE 5.13: ALVS values for L=40 (UKL)

5.6.7 When L=30

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	19	15	93.33%	73.68%
LOW	13	26	43.15%	92.30%
MIDDLE	34	29	75.86%	64.70%
HIGH	14	10	80%	57.14%
Overall Accuracy		70%		
Kappa		0.587		

TABLE 5.14: ALDIANA values for L=30 (UKL)

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	0	14	0%	0
LOW	0	25	0%	0
MIDDLE	69	28	100%	40.59%
HIGH	6	8	75%	100%
Overall Accuracy		45.33		
Kappa		0.156		

TABLE 5.15: ALVS values for L=30 (UKL)

5.6.8 When L=20

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	30	15	93.33%	46.67%
LOW	29	26	38.46%	34.83%
MIDDLE	0	29	0	0
HIGH	21	10	90%	42.85%
Overall Accuracy		41.25		
Kappa		0.246		

TABLE 5.16: ALDIANA values for L=20 (UKL)

CLASS	**Actual Value**	**Predicted Value**	**Precision**	**Recall**
VERY LOW	**30**	**15**	**93.33%**	**46.67%**
LOW	**29**	**26**	**38.46%**	**34.83%**
MIDDLE	**0**	**29**	**0**	**0**
HIGH	**21**	**10**	**90%**	**42.85%**
Overall Accuracy		**41.25**		
Kappa		**0.246**		

TABLE 5.17: ALVS values for L=20 (UKL)

5.7 Over All Classification Accuracies of Both Systems on Pima Indian Diabetes Data

In the previous section the classification accuracies have been mentioned for the user knowledge modeling database which is the multiclass data and now I am going to show the results that I have obtained for Pima Indian Diabetes data that is the binary data. From the results a clear comparison of my technique can be seen for both the binary data and the multiclass data.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
AL with version space	46.03 %	55.55 %	56.37%	61.90 %	[illegible]	[illegible]
AL with Divisive Analysis	44.26 %	49.20 %	53.47 %	62.69 %	[illegible]	[illegible]

TABLE 5.18: Overall Performance of Both System on Pima Database

This table has clearly showed the performance of both methods of different vloumes of training vector i.e. L. The accuracies of ALDIANA can noticed to be below the ALVS. A reason behind this might be the dealing of missing values in my algorithm. It is possible that the mean average value taken to fill the missing values has disturbed the criteria of that feature and thus not so good results have been obtained.

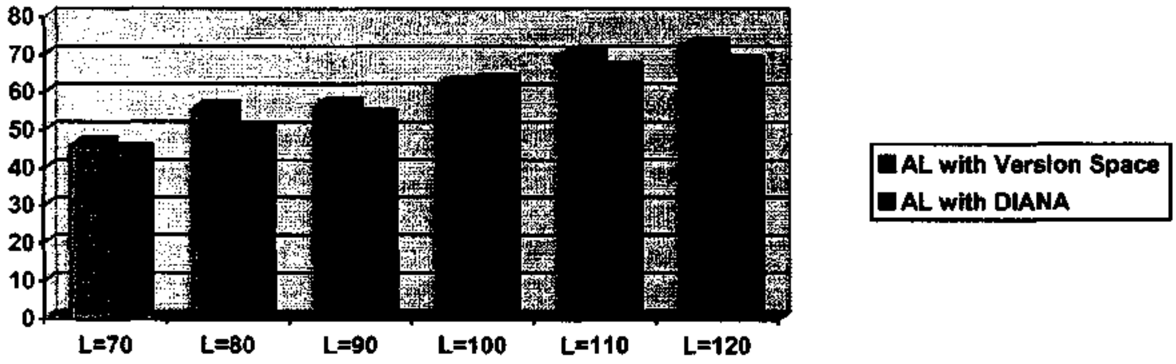


FIGURE 5.6: Bar Chart representation of both systems on Pima Indian Diabetes dataset

The results obtained on the binary data are not up to the mark as in most of the cases the proposed method is lagging behind the base method. The dataset here consists of 768 vectors and the sample amount of training data that I have chosen to represent the results of my technique, falls between 70 – 120. It can be clearly viewed that in majority of cases the accuracy of proposed method is below the accuracy of the base method.

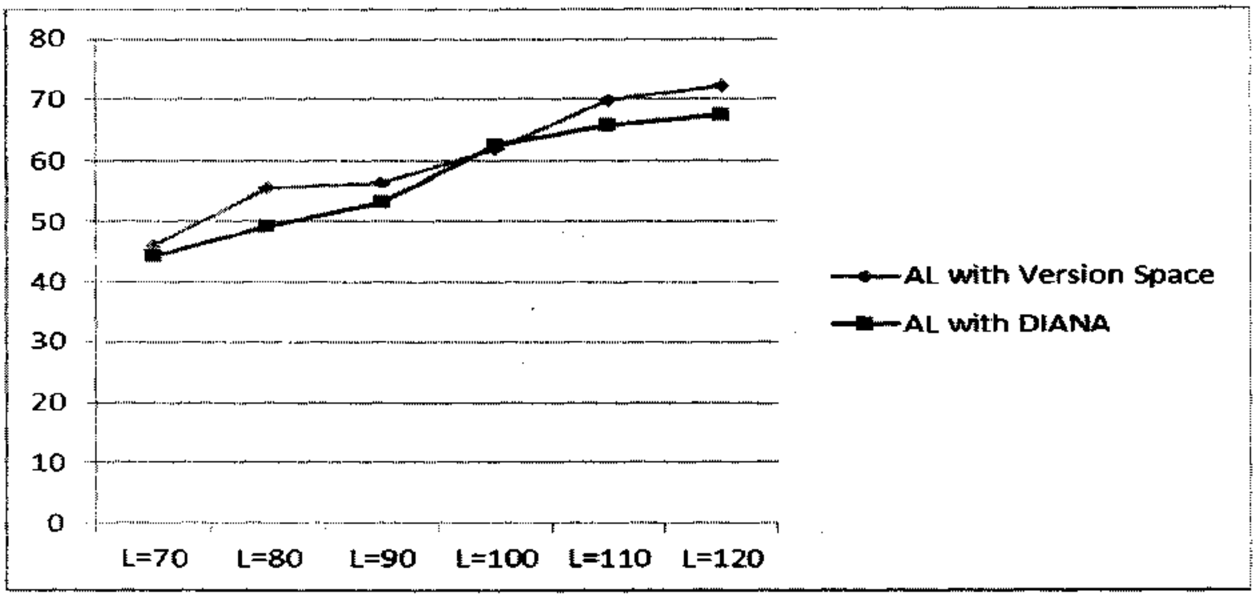


FIGURE 5.7: Performance curve of both systems on Pima dataset

On this line graph the accuracies have been mentioned on the y-axis and the values of Training data L have been mentioned on the X –axis. The one on the x-axis correspond to L=70, 2 for L=80 and this series continues till L=120. When L=70 the accuracy received from the base methods is 47 while that of proposed method is 43. At L=90 both the models showed same accuracy and L=100 is the only point where proposed technique has taken a lead in terms of accuracy. By summarizing this whole result I can say that although the selection of training vectors is random but still the base method has showed a better performance as compared to the proposed method.

5.8 The Classification Details of Pima Data

5.8.1 When L=120

CLASS	**Actual Value**	**Predicted Value**	**Precision**	**Recall**
YES	**42**	**28**	**78.57**	**52.38**
NO	**58**	**72**	**72.22**	**89.65**

Overall Accuracy	74
Kappa	0.441

TABLE 5.19: ALVS values for L=120 (Pima)

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	50	60	66.66	80
LOW	50	40	75	50
Overall Accuracy		70		
Kappa		*0.4*		

TABLE 5.20: ALDIANA values for L=120(Pima)

5.8.2 When L=110

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	57	56	69.64	68.42
LOW	43	44	59.09	60.45
Overall Accuracy		65		
Kappa		0.288		

TABLE 5.21: ALVS values for L=110 (Pima)

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	53	35	66.71	43.39
LOW	47	65	53.84	74.46
Overall Accuracy		58		
Kappa		0.175		

TABLE 5.22: ALDIANA values for L=110(Pima)

5.8.3 When L=100

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	46	44	68.18	65.21
LOW	54	56	71.42	74.01
Overall Accuracy		70		
Kappa		0.394		

TABLE 5.23: ALVS values for L=100 (Pima)

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	54	61	72.13	81.48
LOW	46	39	74.35	63.04
Overall Accuracy		73		
Kappa		0.45		

TABLE 5.24: ALDIANA values for L=100(Pima)

5.8.4 When L=90

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	62	70	71.42	80.64
LOW	38	30	60	47.36
Overall Accuracy		68		
Kappa		0.292		

TABLE 5.25: ALVS values for L=90 (Pima)

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	36	46	69.44	54.34
LOW	64	54	67.18	79.63
Overall Accuracy		68		
Kappa		0.345		

TABLE 5.26: ALDIANA values for L=90(Pima)

5.8.5 When L=80

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	45	62	56.45	77.77
LOW	55	38	73.68	50.09
Overall Accuracy		63		
Kappa		0.277		

TABLE 5.27: ALVS values for L=80 (Pima)

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	46	51	50.98	56.52
LOW	54	49	59.18	53.70
Overall Accuracy		55		
Kappa		0.101		

TABLE 5.28: ALDIANA values for L=80(Pima)

5.8.6 When L=70

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	35	64	35.93	65.71
LOW	65	36	66.66	36.92
Overall Accuracy		47		
Kappa		0.022		

TABLE 5.29: ALVS values for L=70 (Pima)

CLASS	Actual Value	Predicted Value	Precision	Recall
VERY LOW	50	41	41.46	34
LOW	50	59	44.06	52
Overall Accuracy		43		
Kappa		-0.14		

TABLE 5.30: ALDIANA values for L=70(Pima)

5.9 Results on the Calculated Time

Apart from the accuracies another major challenge of this work was the computational cost in terms of time. It has been noticed that although the proposed system with DIANA clustering has not attained a good consistency in the accuracy for the binary data but it is worth mentioning that on both types of data the calculated time of the proposed system was below the base system and this was even more consistent as compared to the accuracies.

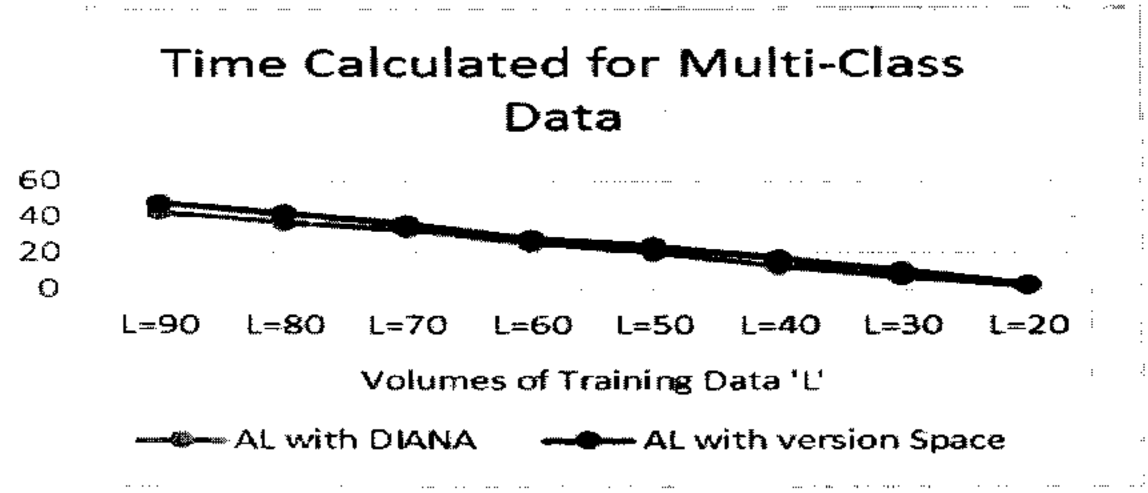


FIGURE 5.8: Performance curve showing the time taken by both systems on UKL dataset

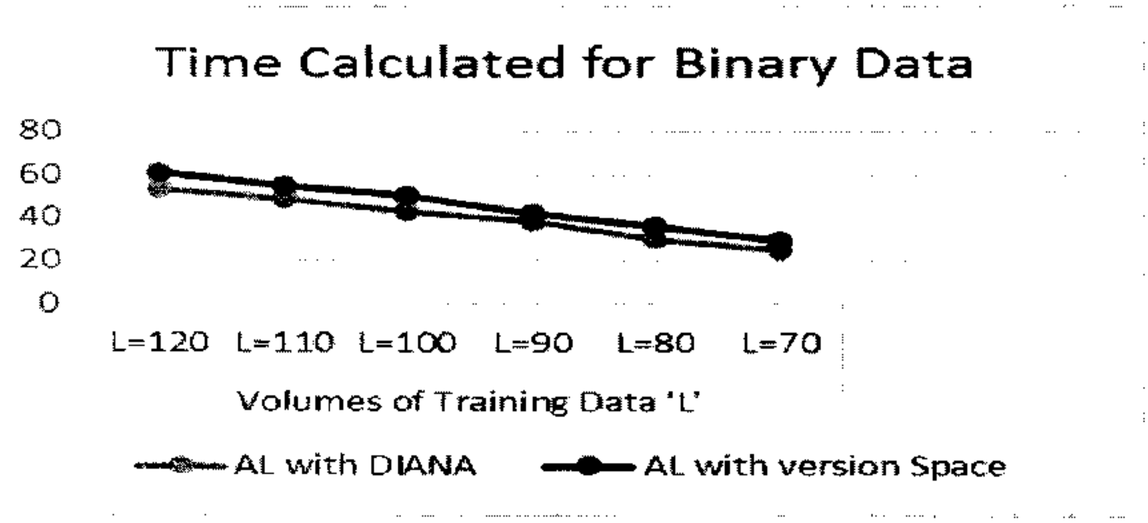


FIGURE 5.9: Performance curve showing the time taken by both systems on Pima dataset

CHAPTER VI

CONCLUSION

6. Conclusion & Future Work

In this chapter I am going to conclude all the findings of this dissertation, the new techniques which can be used for the selection of most informative vector will be discussed. This chapter will try to portray the whole status of this thesis, how much effective the work is? How it can be used as a replacement for the previously used classification systems. The discussion on the previous algorithm and the proposed algorithm will be made and it will be tried to make a clear comparison scenario that will be helpful in showing the effectiveness of this technique.

6.1 Conclusion

From the experimental setup it has been observed that the proposed technique which is basically the implementation of pre-clustering approach in active learning brings an observable change in the performance of the overall classification of the system. The main idea behind any active learning system is to go for the cost & time reduction, there is no fault in the traditional classifiers but the proposed idea is to make those classifiers so efficient that they can give us the best efficiencies in the minimum amount of cost and time. The Proposed system has basically tried to change the active learning system which was based on version space's concept. According to the version space the whole data once gets labeled as one class i.e. positive and then the whole data gets labeled as the other class i.e. negative. My hypothesis was that, if the classifier gets trained on a logical group of data rather than on a supposed data then its accuracy can be improved and the results have proved this correct for the multi-class data.

For the binary data the results have gone quite disappointing and the reason being the good efficiency of version space algorithm on the binary data. A disadvantage of version space algorithm as stated by [5] the version space gets into trouble when the data carries noise in it and also in the case when the learning concept tends to be disjunctive in nature. So in binary when we have only two choices of Yes and No the version space can perform at its best. Thus a major contribution of this dissertation is the comparison of version space with multiclass clustering and as above the results has shown that the multiclass clustering gets its best efficiency for multiple classes and incase of binary data the version space still carries an edge. .

The aim behind the usage of version space or DIANA is to minimize the cost of classification system and in my work I have pre-clustered the data according to divisive analysis clustering (DIANA) procedure and then train the classifier on a fixed ratio of vectors from each cluster. This approach brings a training data that carries member from every area of the provided pool of data and thus the classifier trained on this diverse data shows better performance than the classifier that gets trained on a supposed group of data. The number of training data has also decreased as well.

6.2 Future Work

In my research I have tried to take some very important steps for labeling the textual data but its also true that every work open new areas. In this section I will try to introduce some of the research directions that can take place after the proposed solution.

- The traditional concept of active learning follows the selection of instances and asks the user to label those instances but with the same technique and with the same proposed method one can extend this work for the feature selection. The feature selection phenomenon can be used individually for any research and it can also get summed up with the instance selection as well.
- I have worked on the pool based active learning scenario but the work can be extended in almost the same way for the stream based active learning scenario which works for the dynamically coming data streams.

REFERENCES

[1] Mitchell, T. Michael "Version spaces: a candidate elimination approach to rule learning". In proceedings of 5th international joint conference on artificial intelligence, pp: 305-310, vol 1, 1977

[2] D. Angluin "Queries and Concept Learning" In Machine Learning, Vol. 2, Issue. 4, pp. 319-342, April 1988.

[3] S.Knerr, L. Personnaz, G. Dreyfus "Single layer Learning revisted : a stepwise procedure for building and training a neural network" In Neurocomputing, vol: 68, pp:41-50, 1990.

[4] B. E, Boser, I. M. Guyon and V. N. Vapnik "A Training Algorithm for Optimal Margin Classifiers" In Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory (COLT), 1992.

[5] P. Winston, "Learning by Managing Multiple Models", in P. Winston, Artificial Intelligence, Addison-Wesley Publishing Company, pp. 411-422, 1992.

[6] D. Cohn, L. Atlas, R. Ladner. "Improving Generalization with Active Learning" In Machine Learning - Special issue on structured connectionist systems, Vol. 15, issue. 2, pp. 201- 221, May 1994.

[7] A. Maccallum & K. Migham. " Employing EM in pool-based active learning for text classification" In Proceedings of Fifteenth International Conference on Machine Learning, pp. 350-358, 1998.

[8] A. Blum & T. Mitchell. "Combining labeled and unlabeled data with co-training." In Proceedings of the 11th Annual Conference on Computational Learning Theory, pp. 92–100, ACM, 1998.

[9] Y. Yang & X. Liu."A re-examination of text categorization methods". In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 42–49, 1999.

[10] D.P Weins. "Robust weights and designs for biased regression models: Least squares and generalized M-estimation" In Journal of Statistical planning and Inference, Vol. 83, Issue. 1, February 2000.

[11] S. Tong & D.Koller. "Active Learning for Structure in Bayesian Networks" In International Joint Conference on Artificial Intelligence, pp. 863-869, 2001.

[12] S.Tong "Active Learning: Theory and Applications" P.hd Thesis, Stanford University, California, August 2001.

[13] N. Roy & A. McCallum. "Toward optimal active learning through sampling estimation of error reduction." In Proceedings of the International Conference on Machine Learning (ICML), pages 441–448, 2001.

[14] S. Tong, D. Koller. "Support Vector Machine Active Learning with Applications to Text Classification" In Journal of Machine Learning Research, pp. 45-66, 2001.

[15] M. K. Warmuth, J. Liao, G. Ratsch, M. Matheisom, S. Putta and C. Lemmen, "Active Learning with Support Vector Machines in the Drug Discovery Process" In journal of chemical information and computer science, Vol. 43, Issue. 2, pp. 667–673, Feb 12, 2003.

[16] R. D. King, K. E. Whelan, F. M. Jones, P. G. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell and S. G. Oliver. "Functional genomic hypothesis generation and experimentation by a robot scientist" In Nature, Volume 427, pp. 247-252, January 15, 2004.

[17] X. Zhu. "Semi-Supervised learning literature survey". Technical report, Computer Sciences, University of Wisconsin-Madison, 2005.

[18] G. Tur, D. Hakkani-tur and R. E. Schapire "Combining active and semi-supervised learning for spoken language understanding" In Speech Communication 45, pp. 171–186, 2005.

[19] X. Zhu & X. Wu "Class Noise Handling for Effective Cost-Sensitive Learning by Cost-Guided Iterative Classification Filtering" IEEE Transactions on Knowledge and Data Engineering. Vol 18, issue 10, pp 1435-1440, October 2006.

[20] M. Sugiyama. "Active Learning in Approximately Linear Regression Based on Conditional Expectation of Generalization error" In the Journal of Machine Learning Research, Vol. 12, pp. 141-166, January 12, 2006.

[21] K. Probst, R. Ghani. "Towards `Interactive' Active Learning in Multi-view Feature Sets for Information Extraction" In Proceedings of the 18th European conference on Machine Learning, pp. 683-690, 2007

[22] Z. Xu, R. Akella, Y. Zhang. "Incorporating diversity and density in active learning for relevance feedback" In Proceedings of the 29th European conference on IR research, pp. 246-257, 2007.

[23] A. Kapoor, E. Horvitz, "On Discarding, Caching, and Recalling Samples in Active Learning" In Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, pp. 209-216, 2007

[24] R. Moskovitch, N. Nissim, D. Stopel, C. Feher, R. Englert, and Y. Elovici. "Improving the detection of unknown computer worms' activity using active learning." In Proceedings of the German Conference on AI, pages 489–493. Springer, 2007.

[25] M. S. Islam & M. I. Amin. "An architecture of active learning SVMs with relevance feedback for classifying E-mail." In Journal of Computer Science, Vol. 1, Issue. 1, pp. 15-18, June 2007.

[26] A. Vlachos. "A stopping criterion for active learning"In Computer Speech and Language, Vol. 22, Issue. 3, pp. 295–312, 2008.

[27] Y. F. Li, J. T. Kwok and Z. H. Zhou. "Semi-Supervised Learning Using Label Mean". In Proceedings of the 26th Annual International Conference on Machine Learning(ICML 09), pp. 633-640, 2009.

[28] K. N. Batmanghelich & B. Taskar, and C. Davatzikos. ""A general and unifying framework for feature construction," In image-based pattern classification.," Inf Process Med Imaging, vol. 21, pp. 423–434, 2009.

[29] M. Sugiyama, S. Nakajima. "Pool-based active learning in approximate linear regression" In Machine Learning, Vol. 75, Issue. 3, pp. 249-274, June 2009.

[30] R.B.C. Prud˜encio & T. B. Ludermir. "Combining Uncertainty Sampling Methods for Active Meta-Learning," In Proceedings of Ninth International Conference on Intelligent Systems Design and Applications, ISDA '09, pp.220,225, Dec 2009.

[31] Y. F. Li, J. T. Kwok and Z. H. Zhou "Cost-Sensitive Semi-Supervised Support Vector Machine". In Proceedings 24th AAAI on Artificial Intelligence Conference on Data Mining, July 07, 2010..

[32] Z. Wang, S. Taylor, A. Shah. "Semi-Supervised Feature Learning from Clinical Text" In proceedings of IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 462-466, Dec 2010.

[33] J. Hamm, D. H. Ye, R. Verma, and C. Davatzikos."Gram: A framework for geodesic registration on anatomical manifolds." In Med Image Anal, vol. 14, issue. 5, pp. 633–642, Oct 2010

[34] B. Settles. "Active learning literature survey", Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2010.

[35] Y. Chen, S. Mani."Study of active learning in the challenge," In the International Joint Conference on Neural Networks, PP. 1-7, July 23, 2010

[36] S. J. Pan and Q. Yang. "A Survey on Transfer Learning" In IEEE Transactions onKnowledge and Data Engineering, Vol. 22, Issue No. 10, October 2010.

[37] S. J. Huang, R. Jin & Z. H. Zhou. "Active learning by querying informative and representative examples" In NIPS, pp. 892-900, 2010.

[38] P. Lindstrom, R. Hu, S. J. Delany, B.M. Namee. "SVM Based Active Learning with Exploration" In AISTATS Workshop on Active Learning and Experimental Design, May 16, 2010.

[39] J. Zhu, H. Wang, K. Tsou and M. Ma. "Active Learning with Sampling by Uncertainty and Density for Data Annotations" In Journal of Audio, Speech, And Language Processing, Vol. 18, Issue No. 6, pp. 1323-1331, August 2010.

[40] K. N. Batmanghelich, D. H. Ye, K. M. Pohl, B. Tasker, C. Davatzikos and ADNI "Disease classification and prediction via semi-supervised dimensionality reduction." In Proceedings of Biomedical Imaging: From Nano to Macro,In IEEE International Symposium, pp.1086-1090, March 30 - April 2, 2011.

[41] W. Chu, M. Zinkevich, L. Li, A. Thomas and B. Tseng, "In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 195-203, 2011.

[42] Y. Yan, R. Rosales, G. Fung, J. G. Dy. "Active Learning from Crowds" In proceedings of 28th International Conference on Machine Learning, ICML, July 2011.

[43] P. Rashidi, D. J. Cook. "Ask Me Better Questions: Active Learning Queries Based on Rule Induction" In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 904-912, 2011.

[44] A. Teichman & S.Thrun "Tracking-Based Semi-Supervised Learning" In International Journal of Robotics Research, Vol 31, Issue 7,pp 804-818, June 2012.

[45] Z. Wang, A.D. Shah, A. R. Tate, S. Denaxas, J. S. Taylor, H. Hemingway "Extracting diagnoses and investigation results from unstructured text in electronic health records by semi-supervised machine" in PLOS One, Vol. 7,issue.1,2012

[46] C. C. Loy, T. M. Hospedales, T. Xiang, S. Gong, "Stream-based Joint Exploration-Exploitation Active Learning". In Proceedings of Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference, pp. 1560-1567, June 2012.

[47] R. Ganti, A. Gray. "UPAL: Unbiased Pool Based Active Learning" In In Proceedings of the 15th International Conference on Artificial Intelligence and Statistics, 2012

[48] R. Wang, S. Kwong, D. Chen. "Inconsistency-based active learning for support vector machines." In Pattern Recognition Vol. 45, pp. 3751–3767, 2013.

[49] S. M. Bidoki & S. M. R Moosavi."IDUF: An active learning based scenario for relevance feedback query expansion," In proceedings of International Conference on Information Retrieval & Knowledge Management (CAMP), pp. 244,248, 13-15 March 2012.

[50] F. Fukumoto, T. Yamamoto, S. Matsuyoshi, Y. Suzuki. "Text Classification with Relatively Small Positive Documents and Unlabeled Data" In Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM), pp. 2315-2318, 2012.

[51] K. Liu & X. Qian. "A Novel Batch-mode Active Learning Method for SVM Classifier" In Journal of Information & Computational Science, vol. 9, issue. 16, pp. 5077–5084. 2012.

[52] Y. Fu, X. Q. Zhu and B. Li. "A survey on instance selection for active learning". Knowledge and Information Systems. Vol 35, Issue 2, pp.249-283, May 01,2013.

[53] A. Biswas, D. Parikh "Simultaneous Active Learning of Classifiers & Attributes via Relative Feedback." In proceedings of IEEE conference on computer vision and pattern recognition, pp: 644-651, 2013.

[54] L.Hu, S. Lu and X.Wang "A new and informative active learning approach for support vector machine"In Information Science, Vol 244, pp142-160, 2013.

[55] K. Georgala, A. Kosmopoulos and G. Paliouras "Spam Filtering: An active Learning Approach using incremental clustering". In proceedings of 4th International Conference on Web intelligence, mining and semantics, article No. 23, 2014.

[56] L. Yuan, J. Liu, X. Tang, D. Shi and L. Zhao "Pairwise-similarity based instance reduction for efficient instance selection in multiple instance learning." In International journal of machine learning and cybernetics, March 2014.

[57] I. Žliobait˙e, A. Bifet, B. pfahringer and G. Holmes "Active Learning with drifting streaming data" In IEEE Transactions on Neural Networks and Learning Systems, Vol. 25, January 2014.

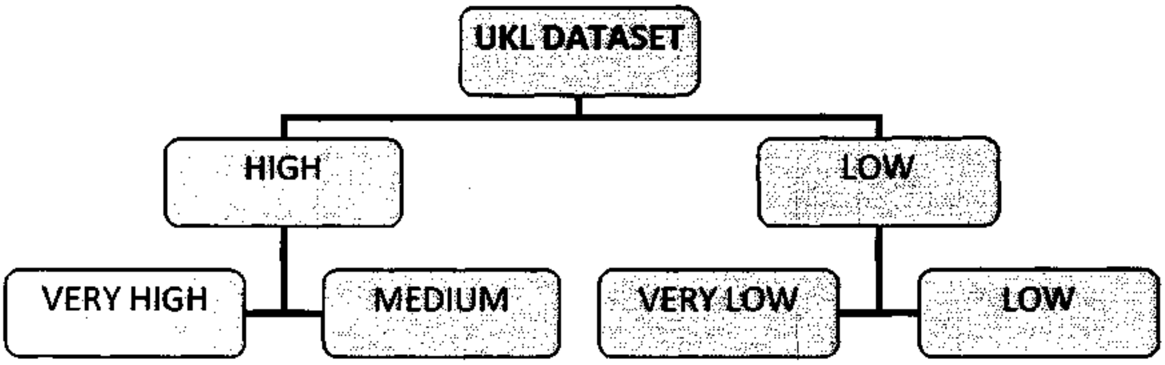
[58] S. Sivaraman, M. M. Trivedi "Active learning for on-road vehicle detection: a comparative study" In proceedings of Machine Vision and Applications, Vol. 25, pp. 599–611, 2014

APPENDIX A

1. THE DATASET DETAILS

1.1 USER KNOWLEDGE LEVEL DATA

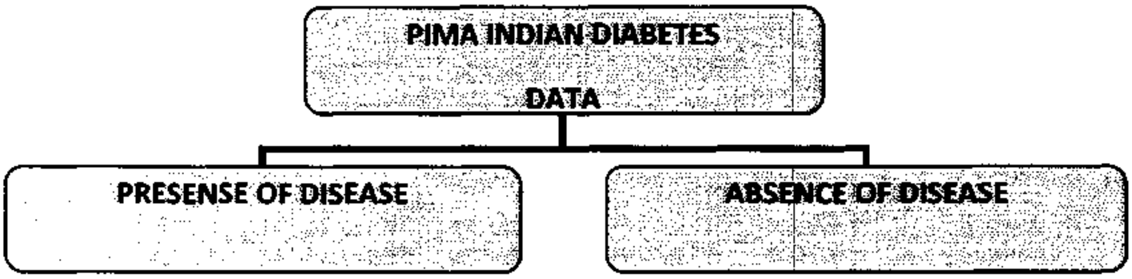
The Database of user knowledge Level has been taken from the UCI Machine Repository and the main theme of the data was to give the knowledge level to students according to their study time, exam performances in a particular objects and other subjects etc. The data is of Karadeniz Technical University, Trabzon, Turkiye and the author of data has used K nearest neigbour approach to generate this data. The ranking of the students in ALDIANA has been performed in layers and apart from clustering the intelligent selection of data for the base clusters has been done manually. At the first Layer the whole pool of data has been divided into two clusters where I have given the extreme cases as centroids i.e. student having 0, 0 ,0 values in the feature space has been given as centroid for low class and students depicting highest values in the feature space has been given as centroid for high class. The Whole procedure has been performed like this.



The split point criteria for the above mentioned grouping has been provided manually as stated earlier but this creates a research area for future as this work can be done atomically via Block Plot and in future I intend to extend this work like this. Another good feature of ALDIANA is that, a comparison of classification results have been made with actual results and thus it is assured the grouping of students according to their knowledge level has been made correctly.

1.2 Pima Indian Diabetes Data

This data is a disease diagnosis data and source is UCI machine repository. The data has been provided by National Institute of Diabetes and Digestive and Kidney Diseases. The instances are containing data of females age 21 and above.

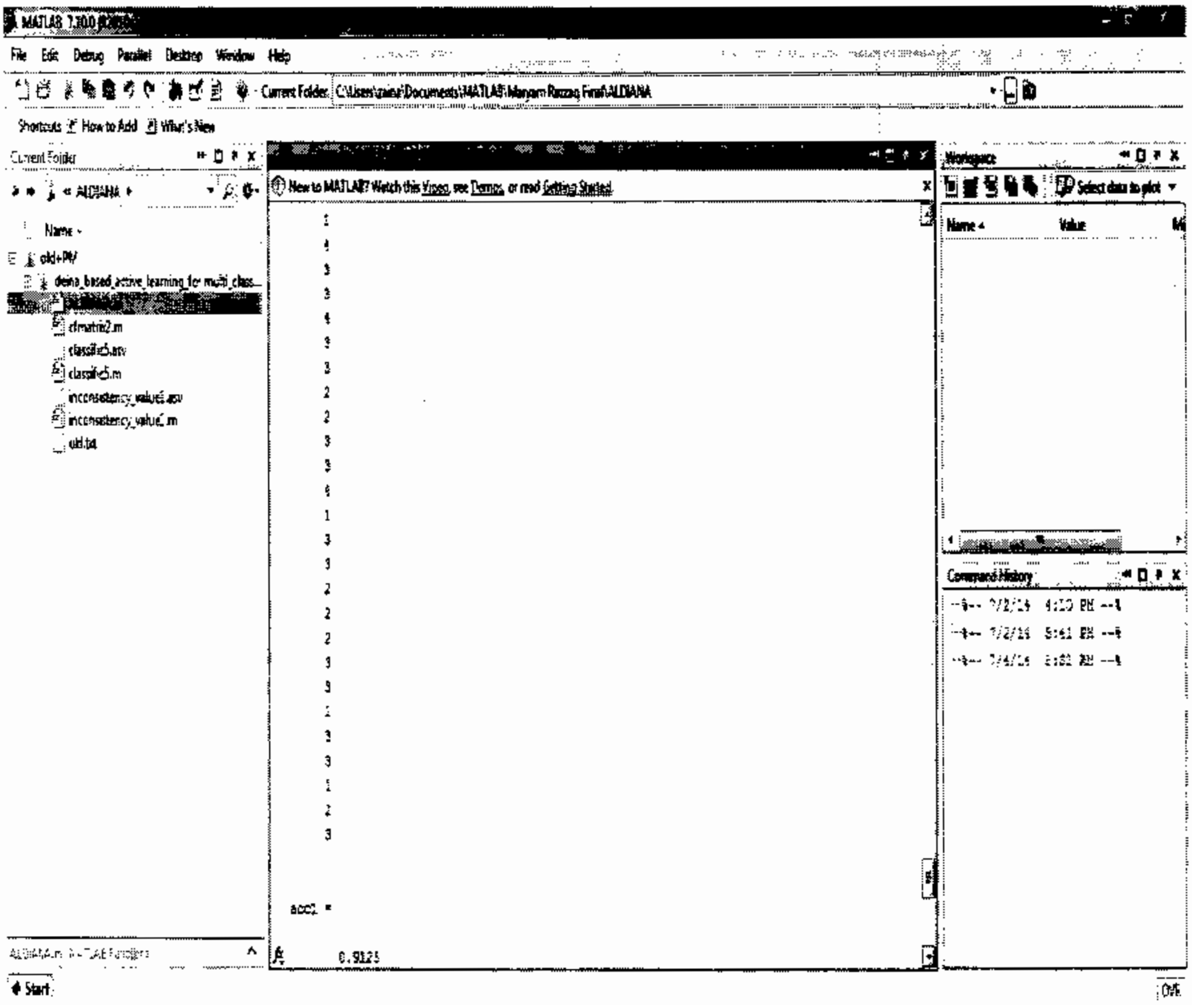


The data is binary in nature as seen from the above figure and the features are related to Plasma glucose concentration, blood pressure, body mass etc. For the Classification of this Database our algorithm ALDIANA has not performed well and a reason of that could be my procedure of dealing with the missing values. I have applied the mean average formula for filling the missing values and it is possible that the filled value is totally opposite the criteria of that feature. Thus a wrong info might have affected the accuracy of the whole system.

APPENDIX B

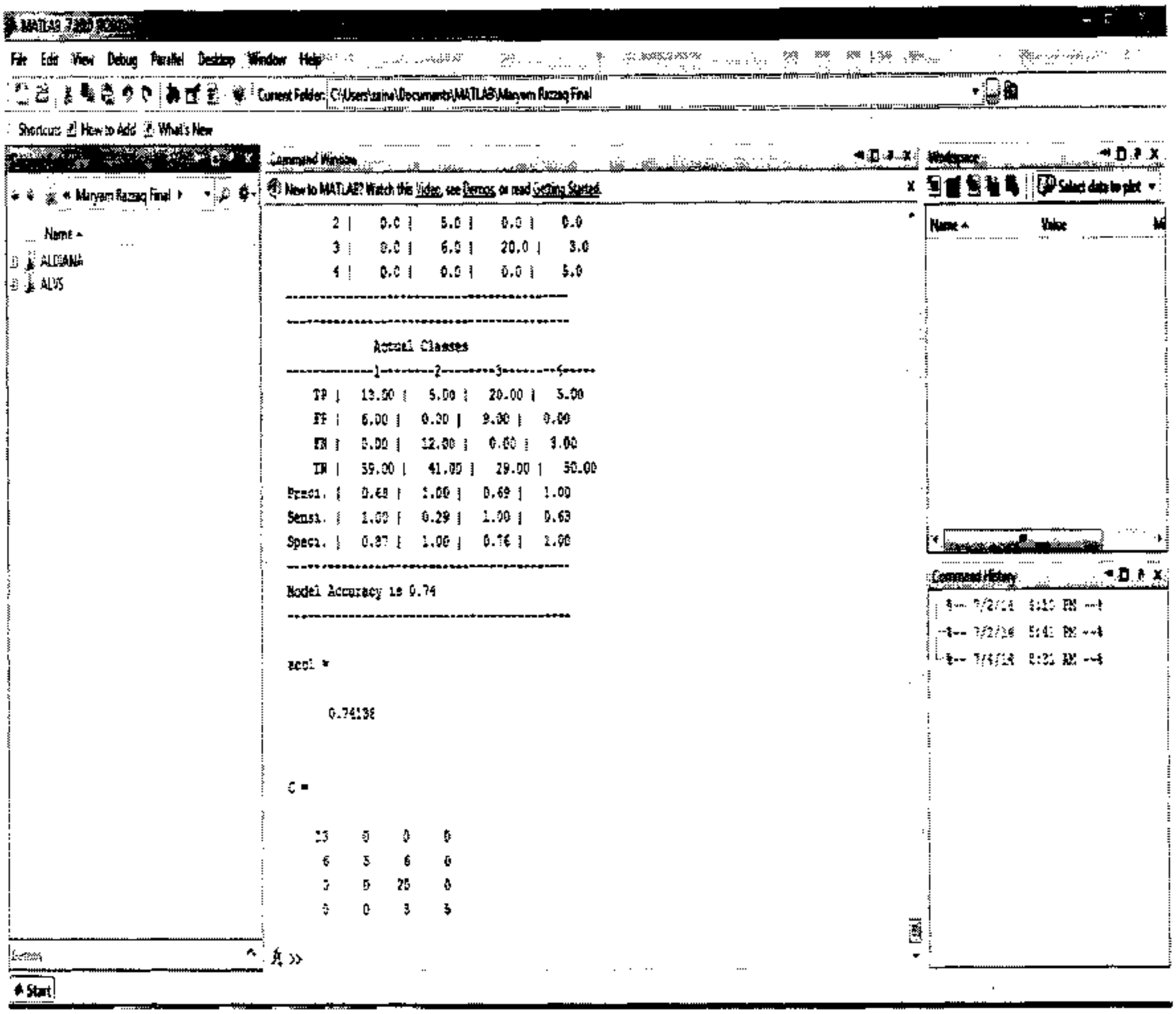
SCREENSHOTS OF THE RESULTS

Matlab R2010 has been used for the experimentation and the results of the algorithm has been displayed in the form of labels as well as in the form of confusion matrix.


MATLAB 7.10.0 (R2010a)
File Edit Debug Parallel Desktop Window Help
Current Folder: C:\Users\zaina\Documents\MATLAB\Maryam Razzaq Final\ALDIANA
Shortcuts How to Add What's New
Current Folder
« ALDIANA »
Name
old+PW
deina_based_active_learning_for multi_class...
cfmatrix2.m
classifier5.asv
classifier5.m
inconsistency_value5.asv
inconsistency_value5.m
old.txt
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Workspace
Select data to plot
Name
Value
Command History
Start
OVR

```
     1
     4
     3
     3
     4
     3
     3
     2
     2
     3
     3
     4
     1
     3
     3
     2
     2
     2
     3
     3
     1
     3
     3
     1
     2
     3


acc2 =

     0.9125
```

The confusion matrix has been generated from the results and with that the accuracy of the overall process has been displayed.

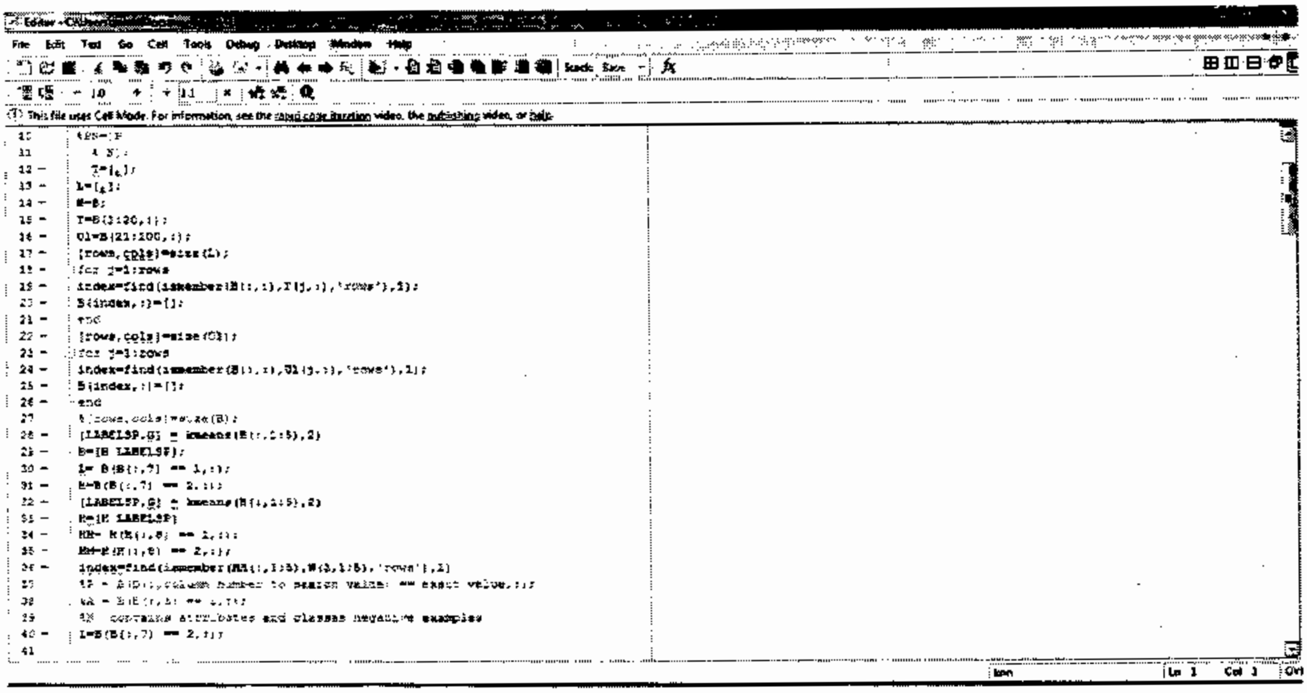

MATLAB 7.10.0 (R2010a)
File Edit View Debug Parallel Desktop Window Help
Current Folder: C:\Users\zaina\Documents\MATLAB\Maryam Razzaq Final
Shortcuts How to Add What's New
Maryam Razzaq Final
Name
ALDIANA
ALVS
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Workspace
Select data to plot
Name
Value
Command History
Start

```
        2 |   0.0 |   5.0 |   0.0 |   0.0
        3 |   0.0 |   6.0 |  20.0 |   3.0
        4 |   0.0 |   0.0 |   0.0 |   5.0
   ------------------------------------------
   ------------------------------------------
              Actual Classes
   -----------1--------2--------3--------4-----
      TP |  13.00 |   5.00 |  20.00 |   5.00
      FP |   6.00 |   0.00 |   9.00 |   0.00
      FN |   0.00 |  12.00 |   0.00 |   3.00
      TN |  39.00 |  41.00 |  29.00 |  50.00
   Preci. |   0.68 |   1.00 |   0.69 |   1.00
   Sensi. |   1.00 |   0.29 |   1.00 |   0.63
   Speci. |   0.87 |   1.00 |   0.76 |   1.00
   ------------------------------------------
   Model Accuracy is 0.74
   ------------------------------------------


   acc1 =

       0.74138


   C =

       13     0     0     0
        6     5     6     0
        0     0    20     0
        0     0     3     5

   fx >>
```

APPENDIX C

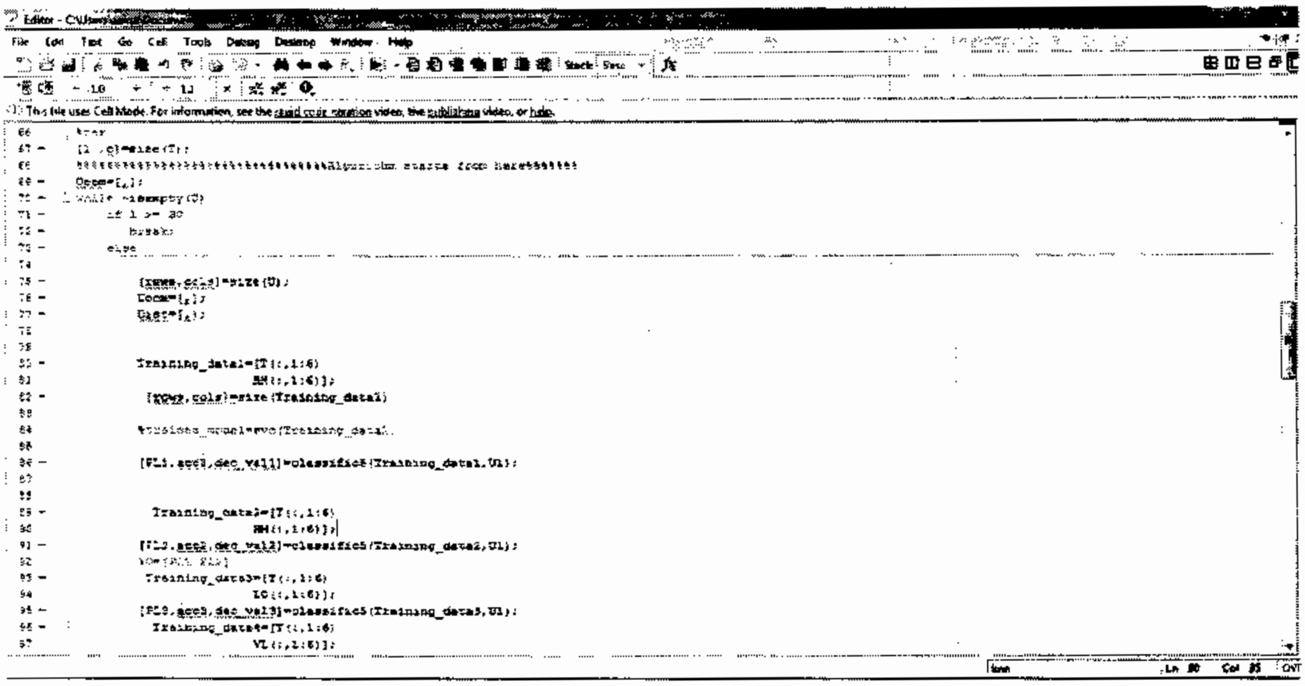
SCREEN SHOTS OF THE CODE

1. ACTIVE LEARNING WITH DIANA (UKL)



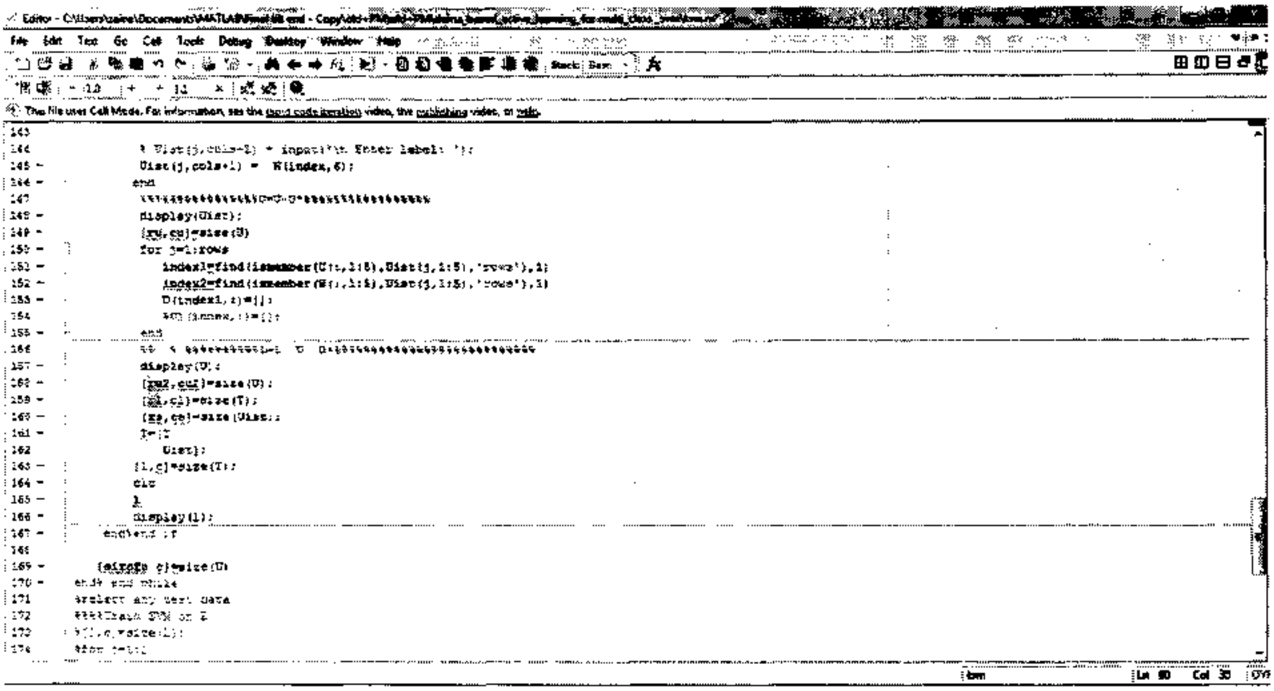
Editor - C:\Users\
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.
Ln 1 Col 1 OVR

```
%PN=[P
  % N];
  T=[];
L=[];
M=B;
T=B(1:20,:);
U1=B(21:200,:);
[rows,cols]=size(L);
for j=1:rows
index=find(ismember(B(:,:),T(j,:),'rows'),1);
B(index,:)=[];
end
[rows,cols]=size(U1);
for j=1:rows
index=find(ismember(B(:,:),U1(j,:),'rows'),1);
B(index,:)=[];
end
%[rows,cols]=size(B);
[LABELSP,G] = kmeans(B(:,1:5),2)
B=[B LABELSP];
L= B(B(:,7) == 1,:);
R=B(B(:,7) == 2,:);
[LABELSP,G] = kmeans(R(:,1:5),2)
R=[R LABELSP]
RR= R(R(:,8) == 1,:);
RH=R(R(:,8) == 2,:);
index=find(ismember(RR(:,1:5),M(1,1:5),'rows'),1)
%P = A(A(:,column number to search value) == exact value,:);
%A = B(B(:,5) == 1,:);
%N contains attributes and classes negative examples
I=B(B(:,7) == 2,:);
```

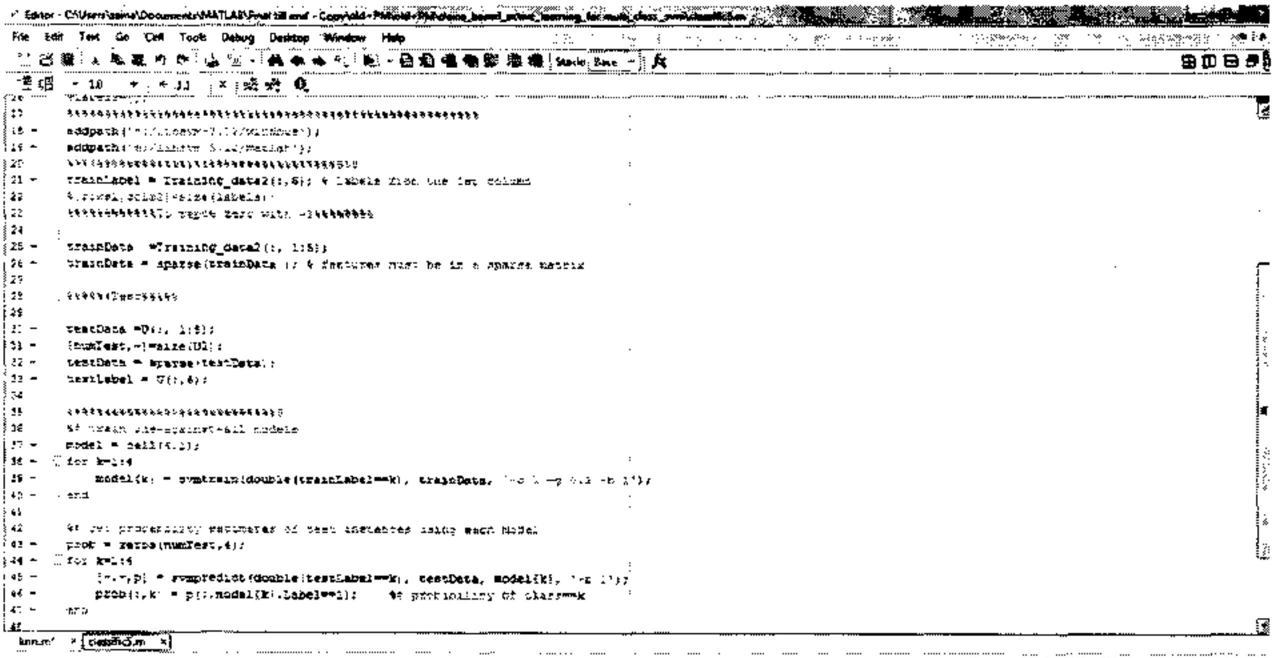


Editor - C:\Users\
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.
Ln 90 Col 35 OVR

```
%rer
[l ,c]=size(T);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Algorithm starts from here%%%%%%%
Open=[];
while ~isempty(U)
    if l >= 30
        break;
    else

        [rows,cols]=size(U);
        Locm=[];
        Uset=[];


        Training_data1=[T(:,1:6)
                        RR(:,1:6)];
         [rows,cols]=size(Training_data1)

        %svmstruct_model=svm(Training_data1.

        [FL1,acc1,dec_val1]=classific5(Training_data1,U1);


          Training_data2=[T(:,1:6)
                          RH(:,1:6)];
        [FL2,acc2,dec_val2]=classific5(Training_data2,U1);
        %O=[FL1 FL2]
         Training_data3=[T(:,1:6)
                         LC(:,1:6)];
        [FL3,acc3,dec_val3]=classifac5(Training_data3,U1);
          Training_data4=[T(:,1:6)
                          VL(:,1:6)];
```

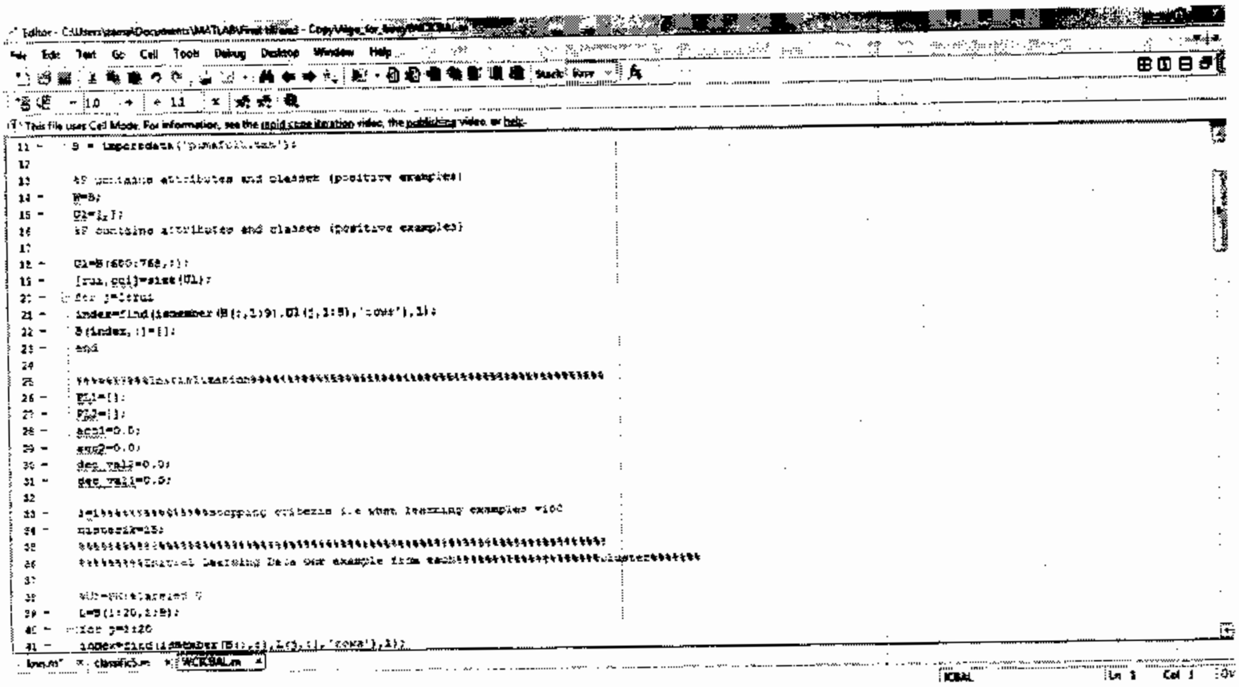

File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.
knn
Ln 90 Col 35

```
            % Dist(j,cols+1) = input('\n Enter label: ');
            Dist(j,cols+1) =  W(index,6);
           end
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            display(Dist);
            [rw,cu]=size(U)
            for j=1:rows
               index1=find(ismember(U(:,1:5),Dist(j,1:5),'rows'),1);
               index2=find(ismember(W(:,1:5),Dist(j,1:5),'rows'),1)
               U(index1,:)=[];
               %W(index2,:)=[];
            end
            %% % %%%%%%%%%%%% U %%%%%%%%%%%%%%%%%%%%%%%%%%%
            display(U);
            [rw2,cu2]=size(U);
            [r1,c1]=size(T);
            [rs,cs]=size(Dist);
            T=[T
               Dist];
           [l,c]=size(T);
           clc
           l
           display(l);
       end%end if

     [mirow c]=size(U)
  end% end while
  %select any test data
  %%%Train SVM on T
  %[l,c]=size(L);
  %for i=1:l
```



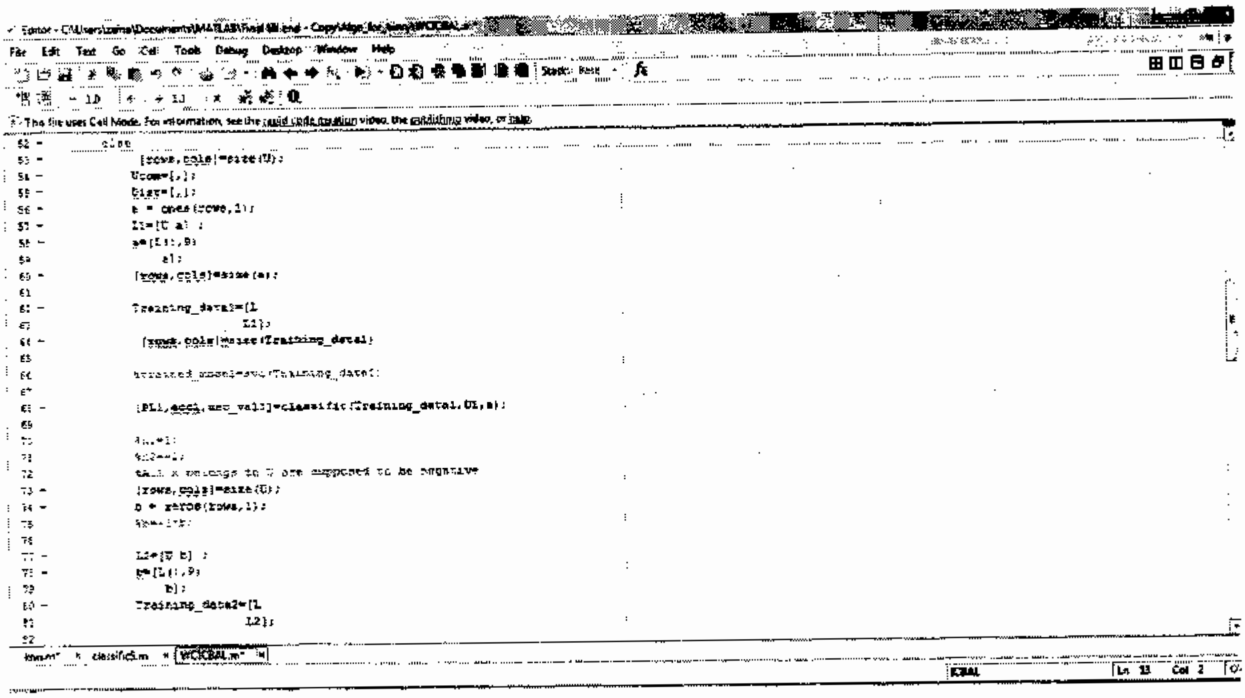
```
  %latest=...;
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  addpath('D:/libsvm-3.12/windows');
  addpath('D:/libsvm-3.12/matlab');
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  trainLabel = Training_data2(:,6); % labels from the 1st column
  %[rows2,cols2]=size(labels);
  %%%%%%%%%%%%%Input data with +1%%%%%%%%

  trainData  =Training_data2(:, 1:5);
  trainData = sparse(trainData ); % features must be in a sparse matrix

  %%%%%Test%%%%%

  testData =U(:, 1:5);
  [numTest,~]=size(U1);
  testData = sparse(testData);
  testLabel = U(:,6);

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  %% train one-against-all models
  model = cell(4,1);
  for k=1:4
      model{k} = svmtrain(double(trainLabel==k), trainData, '-c 1 -g 0.2 -b 1');
  end

  %% get probability estimates of test instances using each model
  prob = zeros(numTest,4);
  for k=1:4
      [~,~,p] = svmpredict(double(testLabel==k), testData, model{k}, '-b 1');
      prob(:,k) = p(:,model{k}.Label==1);    % probability of class==k
  end
```

2. ACTIVE LEARNING WITH DIANA (PIMA)



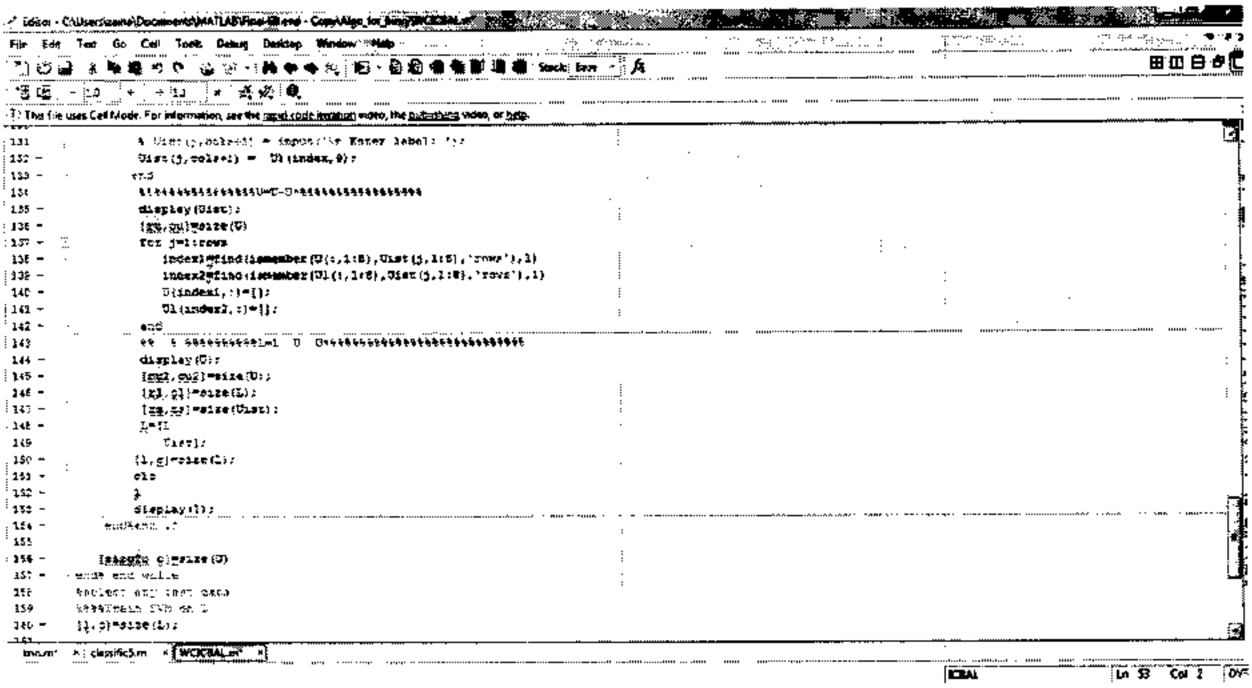
Editor - C:\Users\zaina\Documents\MATLAB\Final till end - Copy\Algo_for_knn\WCICBAL.m
File Edit Text Go Cell Tools Debug Desktop Window Help
This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.
knn.m*
classific5.m
WCICBAL.m
WCICBAL
Ln 1 Col 1 OV

```
B = importdata('pimafull.txt');

%P contains attributes and classes (positive examples)
N=8;
U2=[,];
%P contains attributes and classes (positive examples)

U1=B(600:768,:);
[ru1,cu1]=size(U1);
for j=1:ru1
index=find(ismember(B(:,1:9),U1(j,1:9),'rows'),1);
B(index,:)=[];
end

%%%%%%%%%classification%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PL1=[];
PL2=[];
acc1=0.0;
acc2=0.0;
dec_val2=0.0;
dec_val1=0.0;

i=1%%%%%%%%%%%%%stopping criteria i.e when learning examples =100
clustersize=15;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%Initial Learning Data one example from each%%%%%%%%%%%%%%%%%%%%%cluster%%%%%%%%

%U=U1:classified U
L=B(1:20,1:9);
for j=1:20
index=find(ismember(B(:,1:9),L(j,:),'rows'),1);
```



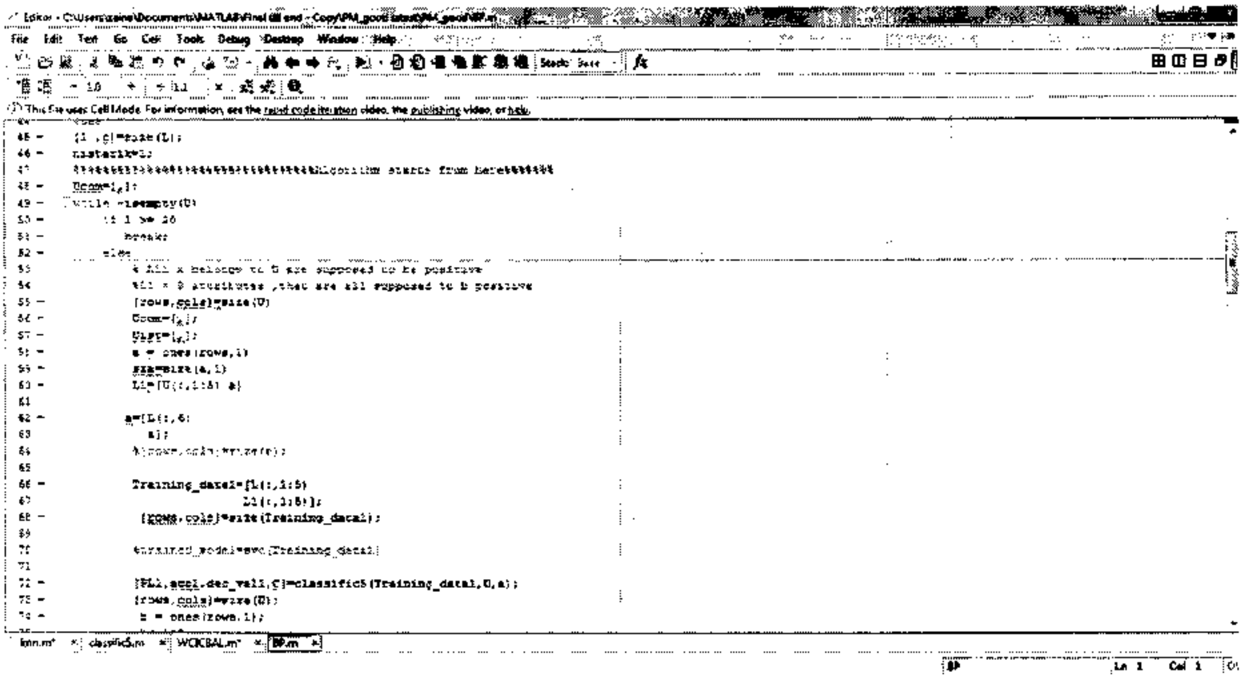
Editor - C:\Users\zaina\Documents\MATLAB\Final till end - Copy\Algo_for_knn\WCICBAL.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.
knn.m*
classific5.m
WCICBAL.m*
WCICBAL
Ln 13 Col 2 OV

```
        else
            [rows,cols]=size(U);
           Ucom=[,];
           Dist=[,];
           a = ones(rows,1);
           L1=[U a] ;
           a=[L(:,9)
               a];
           [rows,cols]=size(a);

           Training_data1=[L
                          L1];
            [rows,cols]=size(Training_data1)

           %trained_model=svm(Training_data);

           [PL1,acc1,dec_val1]=classific(Training_data1,U1,a);

           %a1=1;
           %a2=-1;
           %all x belongs to U are supposed to be negative
           [rows,cols]=size(U);
           b = zeros(rows,1);
           %b=-1*b;

           L2=[U b] ;
           b=[L(:,9)
               b];
           Training_data2=[L
                          L2];
```

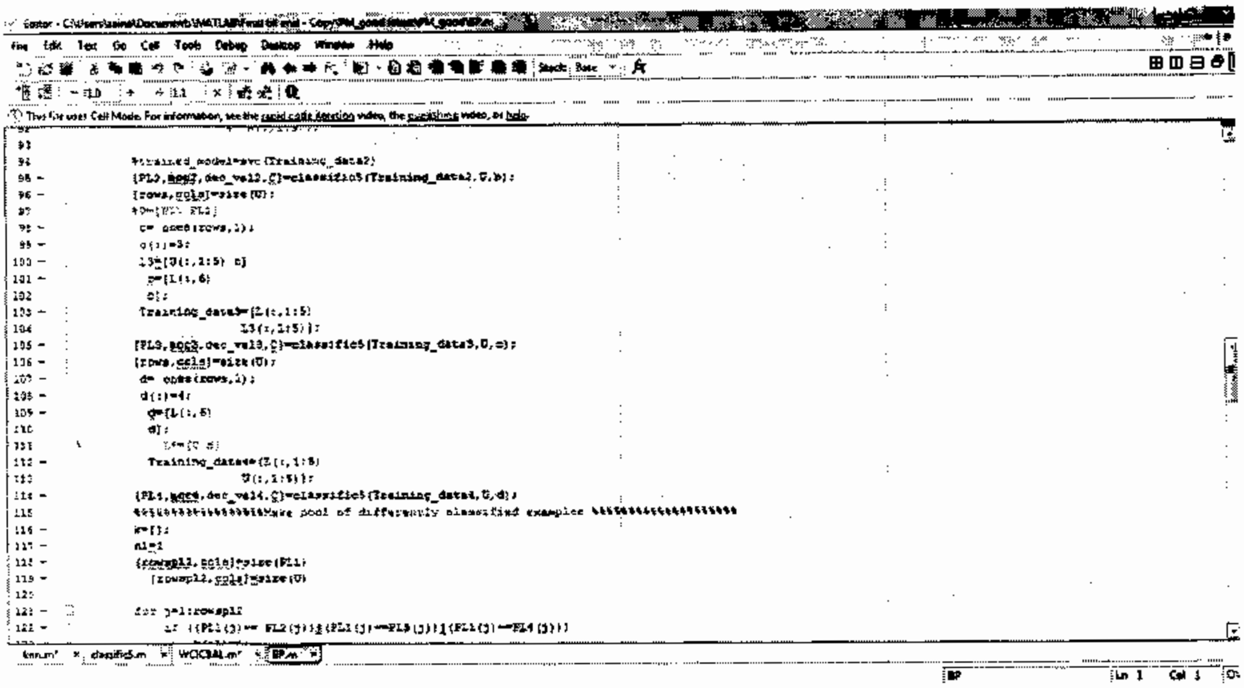

Editor - C:\Users\zaina\Documents\MATLAB\Final-till end - Copy\Algo_for_blog\WCKBAL.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.
knn.m* classific5.m WCKBAL.m*
WCKBAL
Ln 53 Col 2 OVR

```
            % Uist(j,cols+1) = input('\n Enter label: ');
            Uist(j,cols+1) =  U1(index,9);
        end
        %%%%%%%%%%%%%%%%%%U=U-Uist%%%%%%%%%%%%%%%%%%
        display(Uist);
        [rr,cc]=size(U)
        for j=1:rows
            index1=find(ismember(U(:,1:8),Uist(j,1:8),'rows'),1)
            index2=find(ismember(U1(:,1:8),Uist(j,1:8),'rows'),1)
            U(index1,:)=[];
            U1(index2,:)=[];
        end
        %%  % %%%%%%%%%%L=L U  U%%%%%%%%%%%%%%%%%%%%%%%%%%%
        display(U);
        [rr2,cc2]=size(U);
        [r1,c1]=size(L);
        [rs,cs]=size(Uist);
        L=[L
            Uist];
        [l,g]=size(L);
        clc
        j
        display(j);
    end%end if

    [sizeU c]=size(U)
end% end while
%select any test data
%%%%train SVM on L
[l,g]=size(L);
```

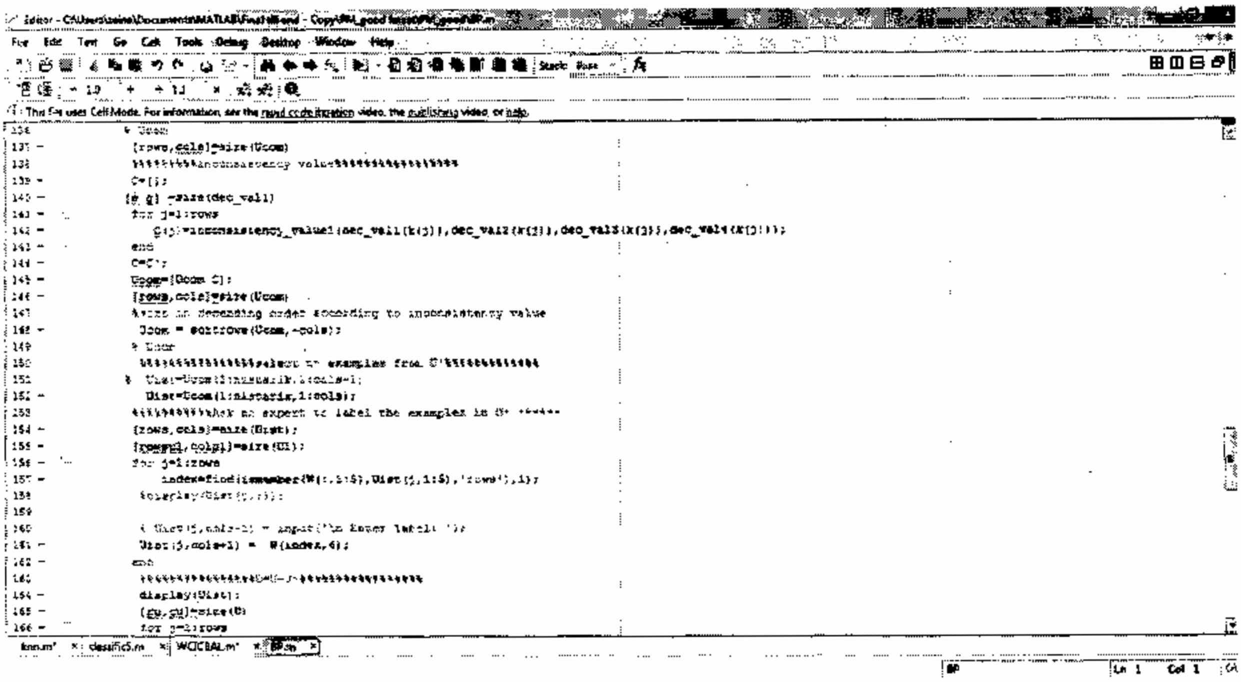
3. ACTIVE LEARNING WITH VERSION SPACE (UKL)


Editor - C:\Users\zaina\Documents\MATLAB\Final till end - Copy\PM_good latest\PM_good\BP.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.
knn.m* classific5.m WCKBAL.m* BP.m
BP
Ln 1 Col 1

```
[l ,g]=size(L);
misterik=1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Algorithm starts from here%%%%%%%%
Ucom=[,];
while ~isempty(U)
    if l >= 20
        break;
    else
        % All x belongs to U are supposed to be positive
        %ll x 8 attributes ,that are all supposed to b positive
        [rows,cols]=size(U)
        Ucom=[,];
        Uist=[,];
        a = ones(rows,1)
        sza=size(a,1)
        L1=[U(:,1:8) a]

       a=[L(:,9)
           a];
        %[rows,cols]=size(U);

        Training_data1=[L(:,1:8)
                        L1(:,1:8)];
         [rows,cols]=size(Training_data1);

        %trained_model=svm(Training_data1)

        [PL1,acc1,dec_val1,C]=classific5(Training_data1,U,a);
        [rows,cols]=size(U);
         b = ones(rows,1);
```

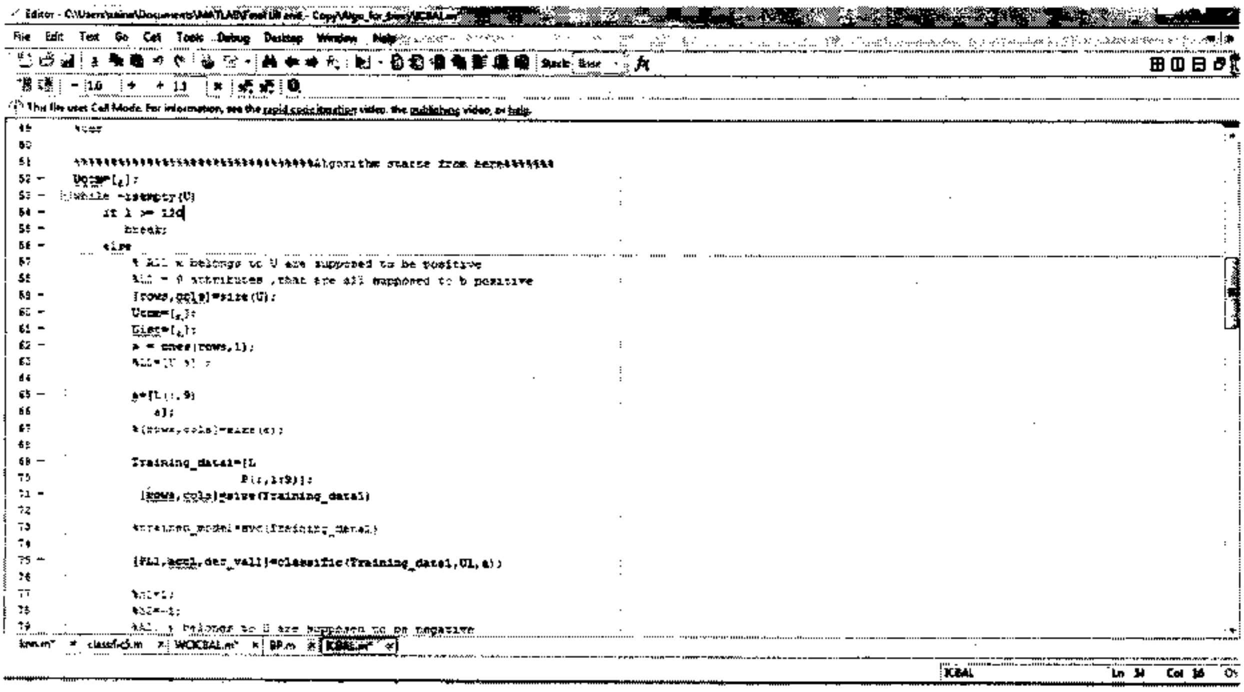

Editor - C:\Users\saine\Documents\MATLAB\Final till end - Copy\M_good\BP.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.
knn.m* classific5.m WCICBAL.m* BP.m
BP Ln 1 Col 1

```
%trained_model=svc(Training_data2)
[FL2,acc2,dec_val2,C]=classific5(Training_data2,U,b);
[rows,cols]=size(U);
%D=[U2 FL2]
 c= ones(rows,1);
 c(:)=3;
 L3=[U(:,1:5) c]
  c=[L(:,6)
  c];
 Training_data3=[L(:,1:5)
                 L3(:,1:5)];
[FL3,acc3,dec_val3,C]=classific5(Training_data3,U,c);
[rows,cols]=size(U);
 d= ones(rows,1);
 d(:)=4;
  d=[L(:,6)
  d];
   %L4=[U d]
  Training_data4=[L(:,1:5)
                 U(:,1:5)];
[FL4,acc4,dec_val4,C]=classific5(Training_data4,U,d);
%%%%%%%%%%%%%%%%make pool of differently classified examples %%%%%%%%%%%%%%%%%%%
k=[];
n1=1
[rowspl1,cols]=size(FL1)
  [rowspl2,cols]=size(U)

for j=1:rowspl2
    if ((FL1(j)~= FL2(j))&(FL1(j)~=FL3(j))|(FL1(j)~=FL4(j)))
```

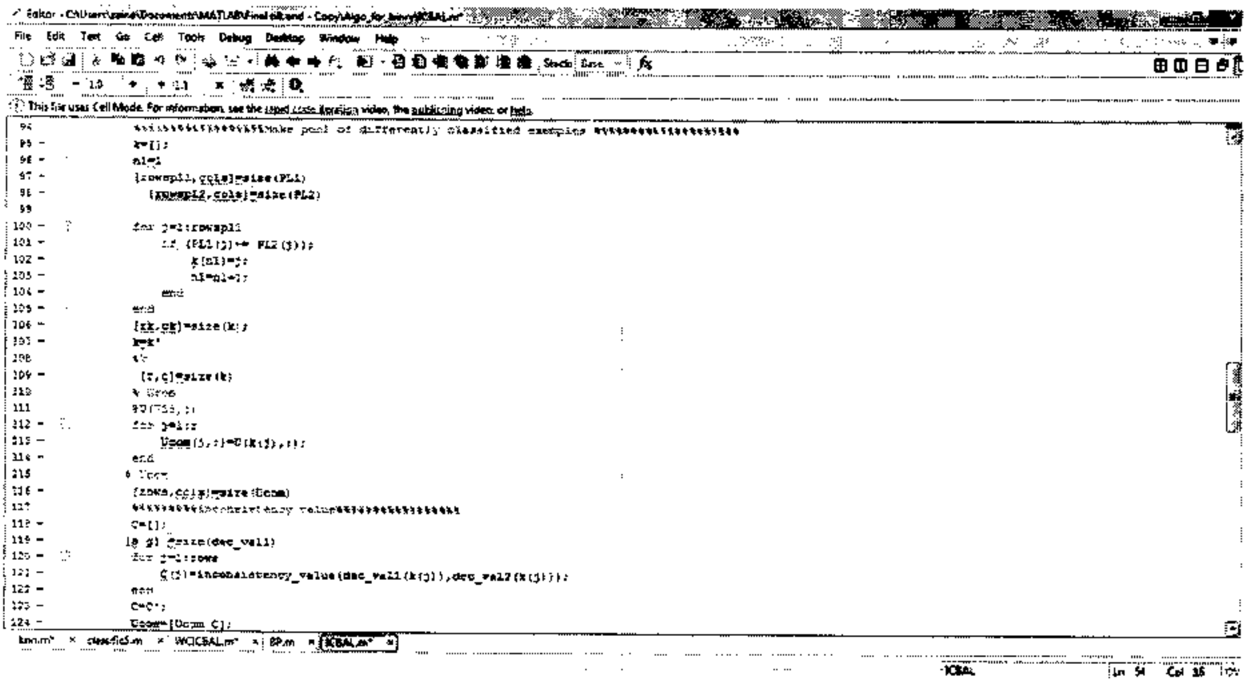

Editor - C:\Users\saine\Documents\MATLAB\Final till end - Copy\M_good\BP.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.
knn.m* classific5.m WCICBAL.m* BP.m
BP Ln 1 Col 1

```
% Ucom
[rows,cols]=size(Ucom)
%%%%%%%%%inconsistency value%%%%%%%%%%%%%%%%%
C=[];
[m n] =size(dec_val1)
for j=1:rows
    C(j)=inconsistency_value1(dec_val1(k(j)),dec_val2(k(j)),dec_val3(k(j)),dec_val4(k(j)));
end
C=C';
Ucom=[Ucom C];
[rows,cols]=size(Ucom)
%sort in descending order according to inconsistency value
 Ucom = sortrows(Ucom,-cols);
% Ucom
 %%%%%%%%%%%%%%%%select n* examples from U'%%%%%%%%%%%%
%  Uist=Ucom(1:nistaris,1:cols-1);
  Uist=Ucom(1:nistaris,1:cols);
%%%%%%%%%%Ask an expert to label the examples in U* %%%%%%
[rows,cols]=size(Uist);
[rowsu1,colsu1]=size(U1);
for j=1:rows
      index=find(ismember(W(:,1:5),Uist(j,1:5),'rows'),1);
 %display(Uist(j,:));

 % Uist(j,cols-1) = input('\n Enter label: ');
 Uist(j,cols+1) =  W(index,6);
end
 %%%%%%%%%%%%%%%%%%U=U-U*%%%%%%%%%%%%%%%%%%%
 display(Uist);
 [ru,cu]=size(U)
 for j=1:rows
```

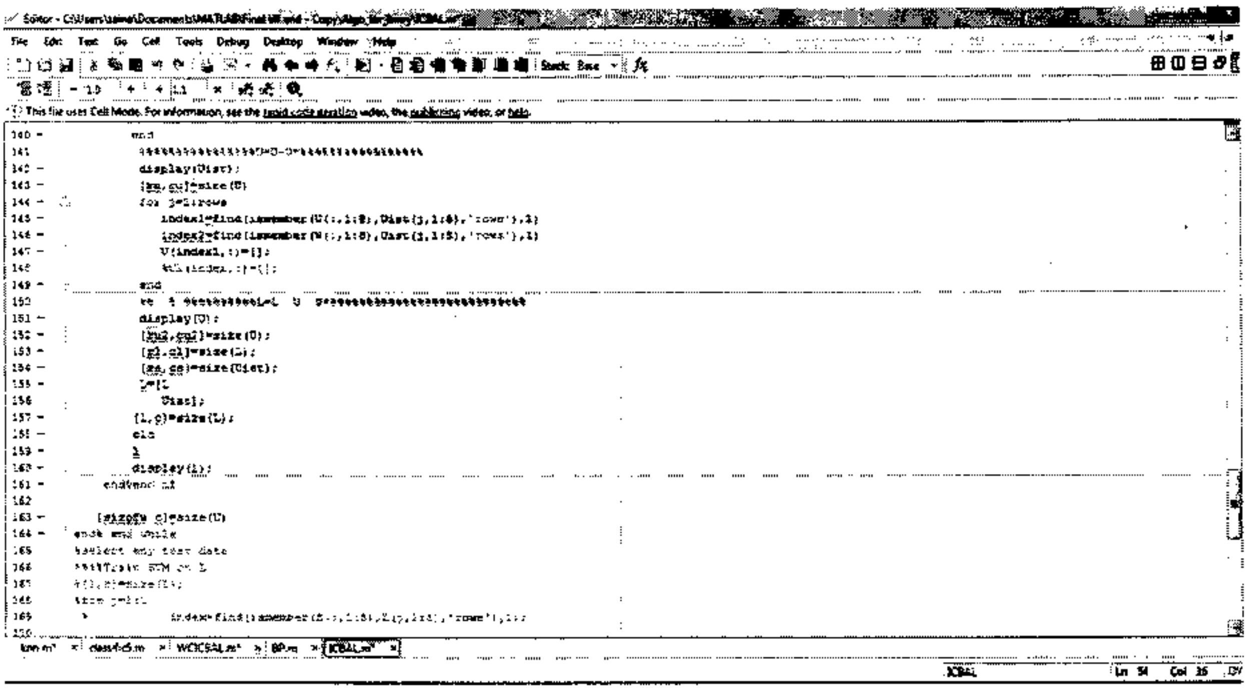
4. ACTIVE LEARNING WITH VERSION SPACE (Pima)


Editor - C:\Users\saina\Documents\MATLAB\Final till end - Copy\Algo_for_binary\ICBAL.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
- 1.0 + ÷ 1.1 ×
This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.
knn.m* × classific5.m × WCICBAL.m* × BP.m × ICBAL.m* ×
ICBAL Ln 54 Col 16 OVR

```
%end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Algorithm starts from here%%%%%%%%
Ucom=[];
while ~isempty(U)
    if i >= 120
        break;
    else
        % All x belongs to U are supposed to be positive
        %U1 = 9 attributes ,that are all supposed to b positive
        [rows,cols]=size(U);
        Ucom=[];
        Uict=[];
        a = ones(rows,1);
        %U1=[U a] ;

        a=[L(:,9)
           a];
        %[rows,cols]=size(a);

        Training_data1=[L
                        P(:,1:9)];
         [rows,cols]=size(Training_data1)

        %trained_model=svc(Training_data1)

        [PL1,acc1,dec_val1]=classific(Training_data1,U1,a))

        %c1=1;
        %c2=-1;
        %All x belongs to U are supposed to be negative
```


Editor - C:\Users\saina\Documents\MATLAB\Final till end - Copy\Algo_for_binary\ICBAL.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
- 1.0 + ÷ 1.1 ×
This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.
knn.m* × classific5.m × WCICBAL.m* × BP.m × ICBAL.m* ×
ICBAL Ln 54 Col 16 OVR

```
        %%%%%%%%%%%%%%%%Make pool of differently classified examples %%%%%%%%%%%%%%%%%%%
        k=[];
        n1=1
        [rowspl1,cols]=size(PL1)
          [rowspl2,cols]=size(PL2)

        for j=1:rowspl1
            if (PL1(j)~= PL2(j));
                k(n1)=j;
                n1=n1+1;
            end
        end
        [rk,ck]=size(k);
        k=k'
        %k
         [r,c]=size(k)
        % Ucom
        %U(755,:)
        for j=1:r
            Ucom(j,:)=U(k(j),:);
        end
       % Ucom
        [rows,cols]=size(Ucom)
        %%%%%%%%%%inconsistency value%%%%%%%%%%%%%%%%%%
        C=[];
       [a b] =size(dec_val1)
        for j=1:rows
            C(j)=inconsistency_value(dec_val1(k(j)),dec_val2(k(j)));
        end
        C=C';
        Ucom=[Ucom C];
```



Editor - C:\Users\[illegible]\Documents\MATLAB\Final [illegible] - Copy\Algo_for_[illegible]\ICBAL.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
fx
- 1.0 + ÷ 1.1 ×
knn.m* × classifyG.m × WCICBAL.m* × BP.m × ICBAL.m* ×
ICBAL
Ln 54 Col 36 OV

This file uses Cell Mode. For information, see the rapid code iteration video, the publishing video, or help.

```
            end
             %%%%%%%%%%%%%%%%D-D%%%%%%%%%%%%%%%%%
             display(Dist);
             [ru,cu]=size(U)
             for j=1:rows
                 index1=find(ismember(U(:,1:8),Dist(j,1:8),'rows'),1)
                 index2=find(ismember(U(:,1:8),Dist(j,1:8),'rows'),1)
                 U(index1,:)=[];
                 %U(index,:)=[];
             end
             %% % %%%%%%%%%L=L U %%%%%%%%%%%%%%%%%%%%%%%%%%%
             display(U);
             [ru2,cu2]=size(U);
             [rl,cl]=size(L);
             [rs,cs]=size(Dist);
             L=[L
                 Dist];
            [l,c]=size(L);
            clc
            L
            display(L);
        end%end it

      [sizeU c]=size(U)
 end end while
 %select any test data
 %%%Train SVM on L
 %[l,c]=size(L);
 %for j=1:l
   %              index=find(ismember(L(:,1:8),L(j,1:8),'rows'),1);
```