The Informative Vector Selection in Active Learning using Divisive Analysis

Zareen Sharf
Assistant Professor
Shaheed Zulfiqar Ali Bhutto Institute of Science & Technology

Maryam Razzak
International Islamic University
Islamabad

***Abstract*—Traditional supervised machine learning techniques require training on large volumes of data to acquire efficiency and accuracy. As opposed to traditional systems Active Learning *systems* minimizes the size of training data significantly because the selection of the data is done based on a strong mathematical model. This helps in achieving the same accuracy levels of the results as baseline techniques but with a considerably small training dataset. In this paper, the active learning approach has been implemented with a modification into the traditional system of active learning with version space algorithm. 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 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 have 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 results were more accurate.**

Keywords*—Active learning; machine learning; pre-clustering; semi-supervised learning

I. Introduction

Machine learning plays a vital role in the concepts and models 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 most difficult and time-consuming tasks, like decision-making, forecasting, pattern recognition etc. The most distinguishing feature of machine learning is that it empowers the machine to learn how to behave and react in a certain situation based on the rules and patterns it drew from the training dataset.

We are living in a world exploding with information. Data is everywhere in the form of, tweets on social networks, comments/reviews on popular blogs, threads on networks, daily publications and news feeds to name only a few sources. The need of the hour is to make our information systems so intelligent that they can extract, transform and reproduce this huge volume of data into a form useful for analysis and prediction. For example, consider a disease discovery system. A disease has certain sets of symptoms and after effects. This information can get updated if we are able to continuously extract new and unique information from the data which is being stored and updated in the patient's history.

The classification of text is the process of assigning a set of predefined categories to the document on the basis of content present in the document by Yang & Liu [16]. Classification could be on the basis of labeled data. The volume of which may vary according to the method used. One of the shortcomings of labeled data is that it is not readily available. The process of labeling data is an expensive task as it involves a lot of human effort. This problem brought new areas of research and most popular of them are Semi-Supervised Learning and Active Learning. They both aim at solving the problem of labeling unlabeled data by using a significantly small volume of labeled data [18].

Active Learning is a technique of semi-supervised machine learning which enables the learning algorithm to query a user interactively and be able to infer desired outputs for newly admitted data.

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 [14], natural language processing [10], information extraction [9], information retrieval [15] and many more.

II. A Review of Semi Supervise Learning

Semi supervise learning is the basis of most of the active learning techniques. Semi-supervised learning and active learning both tackle the problem of dealing with unlabeled data with only a small volume of labeled data [19].

Many studies have been undertaken for the purpose of comparison between supervise learning and semi-supervise learning like Zhu & Wu [18] proposed a technique for handling noisy datasets. The researchers mainly focused on improving cost sensitive classification. They started by applying a general classification strategy that integrated 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 were added to the training iteration by iteration and the accuracy of overall system in noise identification was improved. The major focus in their work was given to the cost of expensive classes, which was actually giving all the focus to some of the classes while the others were being neglected. This could cause inaccuracy in the calculation of the predicted value of the most important class; therefore, causing all the results to become unpredictable.

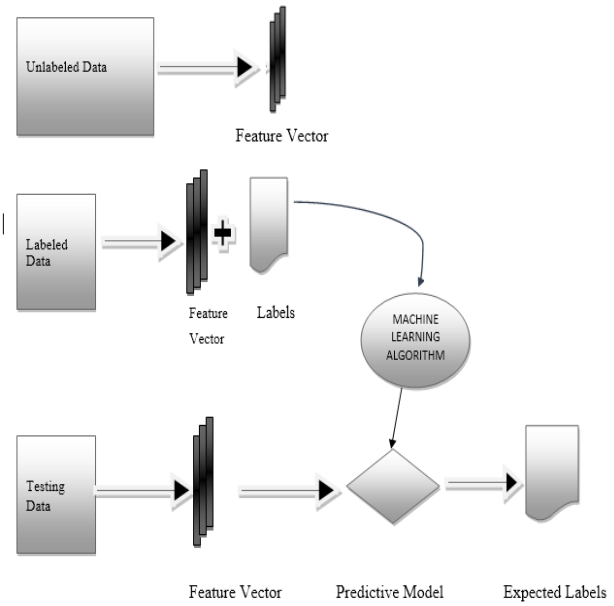

Fig. 1. A semi-supervise learning model.

The Semi-Supervise learning process can be well explained with a diagram. Fig. 1 depicts a structure of semi-supervise learning. On the top we have the unlabelled data having just one attribute and that is the feature vector which contains details of the features of the data other than class. A portion of the unlabelled data is selectively chosen and labeled by a human expert. This labeled data goes to the machine learning algorithm enabling it to learn a mathematical model from it. Next the remaining portion of unlabeled data is fed to that algorithm in the form of testing data. The algorithm then predicts the expected labels of the data according to its prediction model.

Semi supervised learning has received a lot of attention in the field of clinical research. Automated text analysis specifically in electronic health records (EHR) using natural language processing and machine learning have become very popular in the recent decade [13]. The paper proposed an automated system for selecting clinical records that could contain valuable information regarding disease diagnosis or regarding the symptoms of some disease. The classifier was trained on the records of numerous patients diagnosed with a disease. The major advantage of the proposed system was that it did not require any human effort. The proposed model could achieve good efficiency as it was trained based on careful selection of the training data.

III. Literature Review

The term active learning gained popularity among researchers in the 1980s [1] and since then it is a very rich area of research. The main idea proposed by Angluin [1] was that the learner may have the option to ask queries that might be related to its membership (For example, is this instance member of A class or not.), relevance (For example, 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.

Machine learning gains efficiency using active learning as it reduces the human effort required for labeling the vectors/instances. It gives the system liberty to choose the vectors/instances that will be labeled. This is achieved by defining a selection criterion for the vector, which would be labeled and then used as training set for the classifier. In this way, the input domain of the classifier can be defined efficiently.

Active Learning can be classified into two modes, single and batch mode. Therefore, the first step is selection of mode. Next is the selection of a suitable active learning technique. Two of the active learning techniques popularly used are: Pool Based Active Learning and Stream Based Active Learning. Pool based active learning as the name suggests, works for a large pool of unlabeled data while the stream based active learning works dynamically for a live stream of unlabeled data. Our proposed model will work on a large pool of unlabeled data. Finally, we are left with the selection of approach/strategy for active learning by which the sample of training data will be selected. The most common active learning strategies are 'error reduction strategy for sample selection', 'uncertainty based strategies for sample selection', 'uncertainty sampling with Bias', 'uncertainty sampling with prediction' and 'relevance based strategies for sample selection'. The strategy that we have followed in our work is the uncertainty based sample selection.

Biswas & Parikh [2] has proposed an active learning system following an attribute 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. This established an interactive connection between human and machine and this project was further applied for image classification. The human expert played the role of a supervisor to teach visual concepts to a machine. For example, for a certain image, the learner says, "This is a garden, what do you say?" the supervisor might respond saying, "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; thus, enhancing the active learning process.

Active Learning supports multiple instance learning [17] and this process is being commonly adopted because research is now shifting 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. To overcome this problem Yuan & Liu [17] proposed a model of pairwise similarity based instance reduction for Multiple Instance Learning (MIP). The process was dependent on the similarity among the instances within a bag, which was named as training bag. Better performance could be achieved if pair of instances without using the concept of bags was used.

Hu et al. [4] used a simple active learning process for selecting the most informative query that was created with the help of support vector machine. The overall process worked in the binary class domain and initially it started with two instances in the hyper plane, one positive and the other negative. As the process continued, the values of hyper plane kept changing and the instances were selected according to

their minimal distance from the hyper plane. The overall system's accuracy was above 90 percent and a major contribution of the new system was that, it was not working on an artificial dataset as the dataset was being assigned a proper location near or far from the hyper plane.

Our proposed model is very similar to this work and to summarize the entire process we begin with a large pool of unlabeled data and follow a batch mode of active learning technique by which we selected a certain number of vectors as training data and applied pool based active learning on it. Finally, the uncertainty based sample selection procedure was used on the tested vectors that our classifier labeled according to its model.

SVM is one of the most popular and frequently used classification Models. According to a statistical learning theory it is the best classification technique for binary classification. Apart from performing just binary classification, this classifier if merged with some other active learning approaches could give better results in multi-class systems as well.

IV. 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.

The first issue is related to defining some selection criteria for choosing unlabeled data. 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, which requires some solid boundaries for picking the informative vectors.

Second issue is to decide the size of the initial training set. The size of the training set is very important, as the performance of the classifier depends on how well it is trained. If we take a small subset of data from a particular dimension, then our classifier will be bounded in its decisions. This problem does not arise in incremental Learning as the training set incrementally gets appended by new and informative examples. For selective learning, this issue requires attention because based on initial training set the classifier will recognize the patterns/features and will perform the later tasks.

Third issue is to define 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. A very general stopping criterion of this type is the one which checks for the performance of trained classifier after each iteration and then it stops the overall system when the classifiers performance ceases to improve.

Fourth issue is the selection of classification algorithm. 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 supervising learning tasks but the selection of a problem specific classifier is very critical. For example, if we have to do active learning for document classification then we will have to explore which classifiers perform well in that domain.

V. Active Learning Scenarios

According to the fore mentioned concept of supervised learning, a random set of training data was always being 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.

A. Stream based Selective Sampling

The stream based selective sampling is utilized when we do not have static data and the learner has to process a continuous stream of data.

From Fig. 2, it can be seen that the learner, which is any algorithm is getting a dynamic stream of unlabeled data. At first it gives the data directly to the human expert for labelling but once it gets trained on a model then it decides by itself whether to give it to human expert or discard it being unimportant.

The work done by Kapoor & Horvitz [5] is based on discarding, caching and then recalling the samples in active learning. 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.

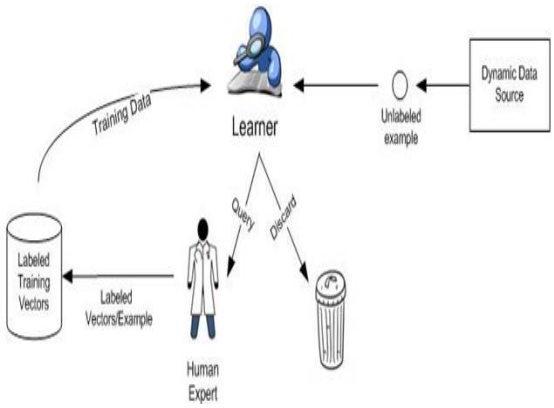

Fig. 2. Stream based active learning model.

Stream based sampling often faces the challenge of deciding whether to label a given instance or not. A possible solution to this issue could be the informativeness of the instance, which happens to be our central research idea. This could be done through selective sampling algorithm a discussion on these algorithms will be done the following section.

Another very interesting observation in the domain of stream based selective sampling was that some researchers were using this to enhance the pool based active learning. Moskovitch & Nissim [8] have done 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.

The term active learning has been utilized in the context of **exploration** and **exploitation** by Loy & Hospedales [6]. They developed a system that worked opposite to the previously used heuristics method in Bayesian classification. According to them if the process of classifying the image and videos was extensively used it could produce better results. A learner would be fed ambiguous instances constantly while a human expert would keep on labeling. Both activities would take place simultaneously. This process continues until the entire posterior distribution of classes have been utilized. During this process a committee of previous hypothesis was made. Thus, two hypotheses were created for each instance. If the classifier showed disagreement on both hypothesis, then that instance would be directed to the human expert for labeling. Next the instance that got labeled by human annotator would be sent to the classifier as training data. This model outperformed all the previous stream based active learning systems but a limitation in the system was that, it did not handle noisy data well. Thus, in case of any noise in the data the hypothesis might get affected and cause the overall system's accuracy to deteriorate. The system can be further enhanced by tackling 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 the paper by Zliobait & Bifet [20] 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.

B. Pool based Active Learning

Real time data is available in huge volumes. The task of labeling this data belonging to various domains gave rise to the concept of Pool Based Active Learning. Usually queries are drawn from the pool which is non-static in their nature, but this is not always the case as there can be presence of dynamic data at some instances. The major difference between stream based and pool based active learning is that the first one sequentially goes through all the data while the latter deals with multiple instances at a time in the form of a huge pool of data.

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, Maccallum & Migham [7] have utilized it for reducing the cost of labeling for a huge set of unlabeled data. The model of (Query by Committee) QBC was extended with the key aim of getting the density of the document explicitly at the time of selecting the examples for labeling. They proposed a probabilistic framework that was based on the EM algorithm in addition to the typical active learning framework. The combination of density weighing methods and EM with active learning methods proved that the accuracy of the system could be improved by having a small training dataset. A limitation to this system was the density estimation which is difficult to calculate especially when dealing with high dimensional data. The work could be further improved by combining the concept of poor probability with the density weight scheme. The techniques for interleaving EM and active learning could also be explored to improve performance.

Ganti & Gray [3] used pool based active learning to overcome the problem 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 assuming there was no noise in the data and it only worked towards unbiased sampling of the labeled data. This made the model quite rigid as such requirements were hardly ever satisfied by real time data.

Pool based active learning can be carried out for a single instance selection mode or multiple instance selection mode. Wang & Kwong [11] performed the Pool Based Active Learning for Multiple Instance selection criteria. The work was performed on MNIST handwritten data and almost 100 bags were created from the whole pool of unlabeled data. The Multi-criteria decision-making procedures were applied for the selection of bags with the help of active ranking. It was seen that the bag margin based active learning outperformed the random sampling as well as the simple SVM active Learning procedures.

VI. INSTANCE/ VECTOR SELECTION

Once the active learning technique, whether Pool Based or Stream Based Active Learning, is selected the next stage is selection of the most informative instances. For the Selection of instances, many strategies have been applied in active learning some of which are 'Error reduction/estimation based

strategies', 'uncertainty Based strategies', 'Uncertainty sampling with prediction', 'Uncertainty sampling with bias', 'Inconsistency based uncertainty sampling', 'Relevance based selection strategies' etc.

In our work, we have focused on uncertainty based sample selection strategies. According to the concept of Query by Committee (QBC) algorithm, a committee of classifiers is developed and trained on different training data. Then a Test data is provided to all classifiers for the sake of predicting class labels. After that the labels of all classifiers are compared and the instances which carry highest value of uncertainty are selected for first querying the human expert and then being added into the training data of main active learning classifier.

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

A. Query Construction

In query construction an arbitrary value is given to a query which is then forwarded to the expert for labeling. The arbitrary value chosen is mostly the extreme possibility of any situation and is well suited for the purpose of training the learner system. For example, if we are required to classify a document we will either add the exact keyword (required to keep the document as a member of certain class) in the arbitrary query or we will give it extreme negative keywords that might be slightly related to the keyword of that class. Query construction is not applicable in most of the classification problems as it is based on the system's/ expert's knowledge rather than being based on patterns found in the data.

B. 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. In in this approach the learners' select queries from the dataset provided and then forward it to the expert for labeling. Thus, labeling is done on real time data rather than arbitrary data making the training process more authentic.

This research also focused on the selective sampling technique and but before that we applied a pre-clustering technique. Pre- Clustering technique divides the data into as many clusters as possible and then the query is selected by taking one, two or more members from each cluster. Pre-clustering assists the active learner to get trained on data coming from each data distribution therefore, various types of data get labelled and classified making the learning process very robust.

Before going into the details of the proposed system, we will discuss the base work by Wang & Kwong [12] which has been implemented in the experiments of this research. Wang & Kwong [12] followed the same procedure for finding the inconsistency based active learning but they worked on the version space concept of general to specific ordering. The GS ordering is always performed on binary class data. We on the other hand have tried to extend this model to work with multi-class data.

The version space algorithm processes a given pool of data based on two hypotheses. According to hypothesis one: it labeled all the data instances as positive while the second hypothesis labeled all the data instances as negative. Two separate SVMs were trained on the dataset from both of these hypotheses and then the testing of those SVMs was performed on the same data. According to concept of QBC both the classifiers give their classification results for testing data. The conflicting data was collected in separate metric and an inconsistency value was added as a feature to all that data. For training, the member with higher inconsistency value was selected for being labeled by the human expert. Finally, the labelled instances were provided to the final classifier as training data.

The modifications made to the base work has been discussed in the next section.

VII. Proposed Architecture

The Active Learning task is always initiated with a random training data which later gets updated with each iteration. The initial data was selected so as to represent data from all the areas of the pool and for that purpose equal volume of data had been provided from each of the cluster that were generated by the DIANA algorithm. A working model of DIANA is shown in Fig. 3.

The Learning of the classifiers continues until it has added a pre-defined number of training vectors after which the process of learning terminates. The testing was done iteratively by assigning different sizes of training data L. Since the data gets selected randomly on each iteration that's why the results generated were different from each other on each run. We checked for the consistency of the results and they were found to be quite consistent for most of the instances.

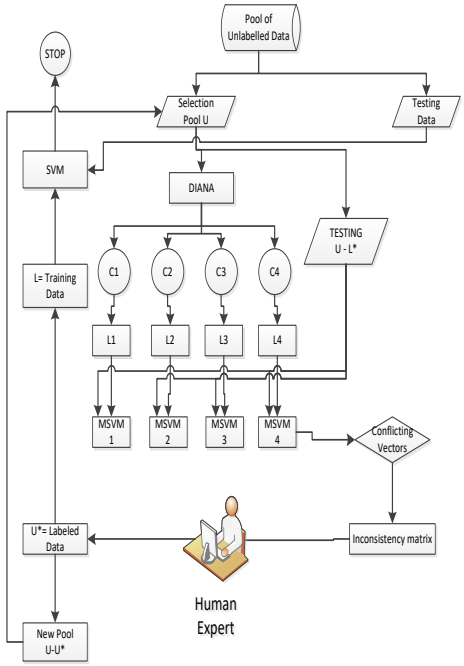

Fig. 3. System architecture for active learning with divisive analysis.

Since we started with the idea of keep our training dataset as small as possible we initially restricted the value of L to be 90. However, since the number of vectors in binary data are usually high therefore the value of L was raised to become 120.

The first and foremost task of this proposed architecture is the handling of noisy data. We have applied the procedure of Mean Average for tackling the missing values in the data. This allowed us to fill in all the missing values with the mean value of the remaining parameters.

To validate the performance of our proposed system, we did a comparison of our classifier with existing real time labeled data and after that the accuracy of the system was calculated.

VIII. Divisive Analysis (Diana)

We selected a hierarchical clustering algorithm for our model. In simple clustering data is divided into a number of groups that are based on the similarity between objects but in hierarchical clustering a proper hierarchy of objects is built. The unique feature of our work is the pre-clustering in which the divisive analysis-s algorithm is used. In this algorithm we create a hierarchy of clusters. Traditionally used clustering is not applied here because we want to get as many clusters as the level of resolution among the data allows us. Another reason for not using simple clustering methodology is that the initial number of cluster are unknown at the start of the process. This problem can be rectified by hierarchical clustering which gives us the freedom of choosing N number of steps to produce a suitable number of clusters for our analysis.

The DIANA algorithm is applied here to check for inter cluster similarity among 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 two clusters and then from those clusters we move on to as many clusters as possible. DIANA initially starts with one cluster which contains all the data instances. On every iteration the larger clusters split up into two clusters and this process continues until every object belongs to 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 objects having highest level of dissimilarity with all other objects and this becomes the splinter group.

2) For every object '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 using 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) The 2nd step is repeated until we get all the values of D_h to be negative. At this point the whole pool will get divided into two groups.

5) The cluster having the largest diameter will get selected as the largest dissimilarity between any of the two objects. Next, this cluster will get further divided.

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

IX. Multilevel Classification via Support Vector Machine

Support Vector Machine commonly known as simply SVM is basically a binary classifier but it can be turned into a multiclass classifier by combining some of its' approaches. Following is a discussion on some of the variants of SVM:

A. Multi-SVM

The main classifier of SVM was developed in 1995 by Cortes and Vapnik and since then it has proven to be one of the best classifiers for binary classification of data. The main concept behind SVM is to plot the whole data on a high dimension space and try to bring a maximum margin hyperplane among the sets of data.

Some of the common approaches for the multi-SVM are:

- One Against one
- One against All
- DAGSVM

In this study, we have followed the one against all approach which is described in the next section.

B. One against All

SVM was primarily designed for binary classification problem but then it was extended to work with multiple classes. One Against All is an extension of SVM in which we construct *k* SVM models that deal with *k* number of classes. Suppose we have an i[th] SVM which is trained on i[th] class that has all instances of 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 i[th] SVM will now solve the following problem.

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

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

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

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

Where, 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. The main aim behind the SVM classification was to search for a balance between the regularization term

$$\frac{1}{2}(w^i)^T w^i \quad .$$

And the errors obtained while training the data.

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

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

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

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

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

X. Experimental Setup

For the demonstration of the results achieved by the proposed method, two datasets have been used. The comparison has been made with the study conducted by Wang & Kwong [12] who applied inconsistency based active learning with the help of version space's general to specific ordering. The Key objective of this work is actually the comparison of version space with divisive analysis (DIANA Algorithm). Some of the features of our proposed system are the handling of missing values with the help of average mean formula and the comparison of final classifier's results with the actual values of the data and then calculating the overall accuracy. The datasets were taken from the UCI machine repository and the details of both datasets are represented in Tables 1 and 2.

TABLE I. User Knowledge Modeling 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 II. PIMA Indian Diabetes Dataset

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

XI. Training and Testing Data

Gendat function is used to generate data randomly for testing and training; according to that 70 per cent data is used randomly for training and 30 per cent for testing. Random selection of the instances follows the prior probability of the class. So the estimation of the sample that would be selected from the particular class is equal to P*N, where P is the prior probability of the class. And N, is the percentage of the data for training.

XII. Results

The Classification work is performed under MATLAB R2010a and the 'SVMtrain' and 'SVMPredict' functions have been used from LibSVM. The Algorithm has been executed on a computer with 2.13GHz Intel Core i3 processor with 2 GB memory and Windows 7, 64-bit Operating System.

As we have worked for the betterment of accuracy and decrease in the computational cost so the results have been shown in terms of time consumed on each iteration and then the accuracies attained in each iteration with different volumes of 'L' which is the training data.

The results have been obtained from the classifier in different iterations, as on each iteration, the classifier chooses random available data and thus the accuracy may vary. Although random results are generated after each run but accuracy still remains better than the base method. This leads us to believe that the base system (Active Learning with Version Space) is not iterative in nature as the classifier always gives the same accuracy and performance regardless of the length of execution. Fig. 4 shows a graph for all the values of L for which we have tested both systems.

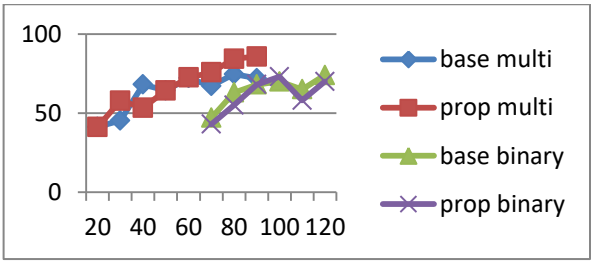

Fig. 4. Accuracies of both systems with binary and multi-class data.

XIII. Overall Classification Accuracies of both Systems on Multiclass Data

From the results shown in Table 3 and Fig. 5, we can observe that the rate of accuracy is increasing with the increasing values of 'L'. It is also noticeable that the proposed classifier's accuracy is better than the base method's classifier.

TABLE III. Accuracies in Terms of Percentages of Both Systems with Different Values of L on Multi-Class Data

Method	**L=20**	**L=30**	**L=40**	**L=50**	**L=60**	**L=70**	**L=80**	**L=90**
AL with version space	41.25	45.33	68.06	64.29	72.06	67.16	74.60	72
AL with Divisive Analysis	41.25	57.87	53.37	64.25	72.5	75.75	84.25	85.69

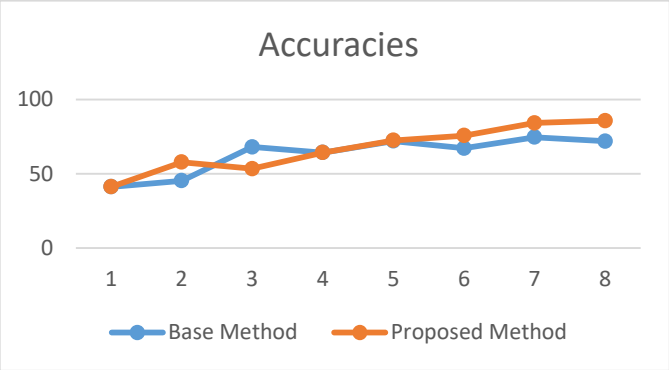

Fig. 5. Accuracies of both system on multiclass data.

Another significant observation is that for L=20 the accuracies of both the systems were same and the obtained accuracies after following 10 iterations also came out to be same. This may be because of the value of L=20 instances out of 258 instances proved to be an extremely small sample size and so the actual performance gain of the proposed algorithm could not be accurately measured. This is the reason both algorithms have run to a level that is similar to the normal classification and because of this the results of each iteration obtained from both of the algorithms stays the same.

XIV. Comparative Analysis of Baseline and Proposed Classifiers on Binary Data

In the previous section, we discussed the performance of our classifier with respect to a user knowledge database which happened to be multiclass database. In this section, we will discuss the results obtained for Pima Indian Diabetes database which is a binary class database. From the results depicted in Table 4 and Fig. 6, a clear comparison of our technique can be seen for both the binary data and the multiclass data.

TABLE IV. Accuracies in Terms of Percentages of Both Systems with Different Values of L on Binary Data

	L=70	L=80	L=90	L=100	L=110	L=120
AL with version space	46.03	55.55	56.37	61.90	69.84	72.22
AL with Divisive Analysis	44.26	49.20	53.17	62.69	65.87	67.46

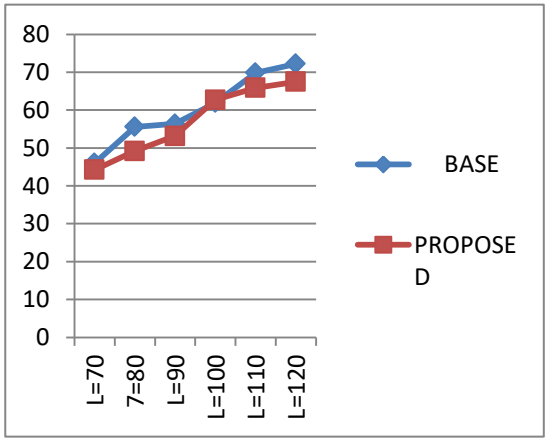

Fig. 6. Accuracies of both system on binarydata.

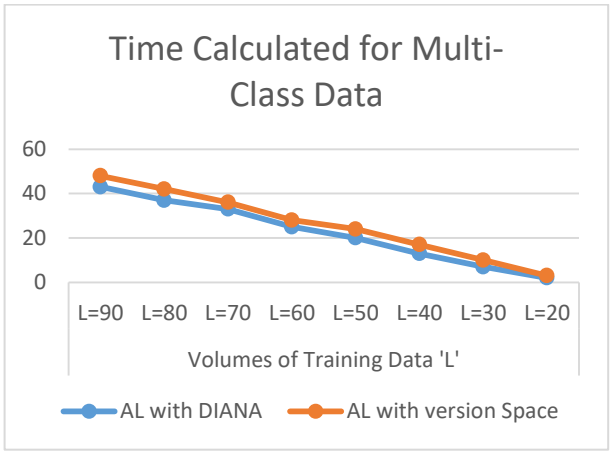

Fig. 7. Calculated time for both systems in multi-class enviornment.

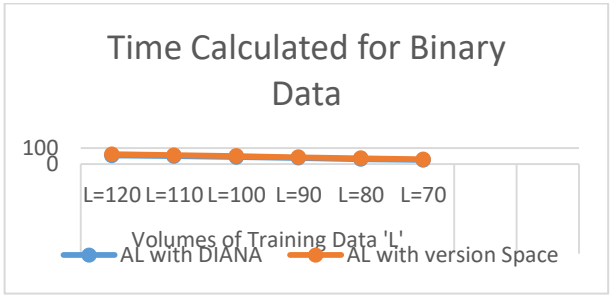

Fig. 8. Calculated time for both systems in binary enviornment.

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 consisted of 768 vectors and the sample amount of training data that we have chosen to represent the results of our technique, ranges between 70 and 120. It can be clearly viewed that in most of the cases the accuracy of proposed method is below the accuracy of the base method when working with binary data.

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 as shown in Fig. 7 and 8, on both types of data, the calculated time of the proposed system was better than the base system and this was even more consistent as compared to the accuracies.

XV. 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 reduce the computation cost & time. The proposed idea is an effort to improve the performance of the baseline classifier. We begin with the assumption that, if the classifier gets trained on a logical group of data rather than one based on random assumption then its accuracy can be improved. This hypothesis was further validated to be correct for the multi-class data.

On the other hand, the proposed model did not produce any performance gain for the binary data. It was observed that the version space algorithm works better on binary data. A disadvantage of version space algorithm as stated by Zhu [19] is that it does not work well in cases where there is noise found in the data and also in the case when the learning concept tends to be disjunctive in nature. The major contribution of this research is the comparison of version space with multiclass clustering and as stated above the results have shown that the multiclass clustering performs better in case of data with multiple classes but in the case of binary data the version space algorithm performed better.

The aim behind the usage of version space or DIANA is to minimize the cost of classification system and in our work. We 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 each group of the given 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 volume of training data was also reduced considerably.

XVI. Future Work

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.

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

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

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

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

[5] Kapoor A. & Horvitz E., “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

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

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

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

[9] Probst R. &Ghani R. “Towards Interactive Active Learning in Multi-View Feature Sets for Information Extraction”. In Proceedings of European Conference on Machine Learning (ECML), pp, 683-690, 2007

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

[11] Wang R. & Kwong S., “Active Learning with multi-criteria decision-making systems” In Pattern Recognition, vol 47, pp. 3106-3119, 2014.

[12] Wang R., Kwong S. & Chen D., “Inconsistency Based Active Learning for Support Vector Machines” In Pattern Recognition, vol 45, pp: 3571-3567, 2012.

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

[14] Warmuth M. K., Liao J., Ratsch G., Matheisom M., Putta S. & Lemmen C., “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.

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

[16] Yang Y. & Liu X. 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.

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

[18] Zhu X. & Wu X. “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.

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

[20] Žliobait˙e I., Bifet A., Pfahringer B. & Holmes G. “Active Learning with drifting streaming data” In IEEE Transactions on Neural Networks and Learning Systems, Vol. 25, January 2014.