Clustering Exploiting Sparse Representation Learner For Robust Classification

Sudarshan Babu
ECE department, SVCE,
Anna university,
Chennai, India.
sudarshan.warft@gmail.com

***Abstract*—We propose a learning algorithm that gives a significant and consistent improvement in accuracy over the existing sparse representation learner. The proposed work is built on two essential properties of data. The first is that data points belonging to the same class span the same subspace (here onward referred as subspace property), and the second is that data points belonging to the same class are a part of the same cluster (here onward referred as clustering property). This paper proceeds by discussing a mode of breakdown for the sparse representation learner. Then, we introduce clustering exploiting sparse representation learner, which exploits both the subspace and clustering property to overcome the issues faced by sparse representation learner. The paper provides a strong geometric perspective of the classification scene involved with the different optimization frameworks discussed in the paper. To support our claims empirically, experiments where conducted comparing the sparse representation learner with the clustering exploiting sparse representation learner with a set of five diverse datasets. The final finding of this work is that clustering exploiting sparse learners could be safely assumed to give a improvement of 5% to 10% over the ordinary sparse representation learner.**

***Keywords*-Sparse Representation Learner; Clustering Exploited Sparse Representation Learner; A Novel Optimization Framework for Data Mining; Regularized Bicriteria Optimization Problem; l1-norm minimization;**

I. Introduction

Traditionally machine learning algorithms have exploited different properties exhibited by data. For instance, algorithms like SVM and kmeans exploit clustering property, while algorithms like sparse representation learner and its variants exploit subspace property [2], [3], [1]. While the two properties are being exploited separately, there is no unifying learning framework that exploits both the subspace and the clustering property simultaneously. Towards this end, clustering exploiting sparse representation learner has been developed.

The optimization framework of the sparse representation learner has gained a lot of popularity in recent times and has found applications in several classification problems [1],[4] and [5]. It has grown in popularity owing to its effectiveness and computational tractability. The optimization problem here is cast a l1-norm minimization problem. By making a few variations to this problem formulation, the sparse representation learner originally designed to exploit only the subspace property is now made to exploit both, the subspace and the clustering property.

The l1-norm minimization problem of the sparse learner, is altered to a regularized bicriterion optimization problem with three constraints. These changes enable the learning algorithm to exploit both the subspace and the clustering property simultaneously. It is important to note that the new optimization formulation is still convex, thereby making the proposed work computationally tractable.

The paper discusses a mode of break down for the sparse representation learner. This point of break down is overcome via the clustering exploiting sparse representation learner. The objective is to show that the proposed method of clustering exploiting sparse representation learner is an improvement over the standard sparse representation learner. This is done empirically by comparing the performance of both the learners on a diverse set of datasets. Our findings are that the cluster exploiting sparse representation learner gives a 5 to 10% improvement over the standard sparse representation learner.

A. Contributions by the paper

The paper introduces a novel unifying framework capable of exploiting both the subspace and the clustering property called the clustering exploiting sparse representation learner. The paper proposes a change in the optimization framework of the sparse representation learner for robust classification. In terms of accuracy this work is an improvement over the existing sparse representation learner.

B. Topology of the Paper

This paper is broken down into the following sections. Section II discusses other variations in sparse representation learners. This is followed by describing the working of the sparse representation learner III. Then in section IV the point of breakdown of the sparse representation learner is discussed. Subsequently, section V introduces the cluster exploiting sparse representation learner. Then to provide empirical backing, the performance upgrade brought about by the proposed learner is discussed in section VI. Finally, in section VII, the paper ends with a conclusion describing methods of extending the proposed work.

II. Related Work

Since our work is a variation on sparse representation learner, we discuss earlier works that are other variations of the sparse representation learner.

[6] proposes kernel sparse representation, which uses kernel tricks on the sparse representation framework to exploit non-linear and linear information, as opposed to the standard sparse representation framework that only exploits linear information.

[7] proposes an alternative to the scheme that is used to determine the class of the test vector from the solution vector obtained from the l1-norm minimization problem. Here they state that the dictionary matrix has a block structure where, the training vectors from each class form blocks. They argue saying that, by finding the representation that has a minimum number of blocks, classification results are improved.

[8] proposes a scheme where classification is done based on the residual that is obtained from the optimization problem involved in the classification scheme.

III. Supervised learning via Sparse representation learner

This section starts by giving a general overview of the supervised learning problem. It then proceeds by describing how the supervised learning problem is framed in the sparse representation learner setting. Then, it provides a geometrical picture of the sparse representation learning algorithm. Finally, the section ends with the required mathematical theory, translating the geometrical interpretation into a succinct set of equations.

A. Supervised Learning

In supervised learning, the algorithm is trained by providing it with labeled objects. These objects are presented to the learning algorithm by a set of attributes or features. From this training, the learning algorithm is expected to determine the label of unknown objects.

B. Intuitive Overview

The inductive bias of the sparse representation learner is that, the vectors representing the training objects of a particular class, will combine linearly to realize the test vector, if the test vector belongs to the same class as the training vectors. So for instance, if the test vector belonged to class 'A', then the training vectors belonging to class 'A' will combine linearly to realize the test vector. This assumption is framed from the subspace property according to which, data points belonging to the same class lie on the same low dimensional subspace.

So, in a k-class classification problem there exist k low dimensional subspaces, one for each class. In the training phase these k low dimensional subspaces are merely being sampled to get a spanning set for each of these k subspaces. During the testing phase our objective is to determine the subspace the test vector resides in. Thus by determining which of the k spanning sets can linearly realize the test vector, the subspace the test vector resides in can be determined, and equivalently the class of the test object. So, in our case the spanning set 'A' (formed by the training vectors of A) which spans the subspace 'A' will combine linearly to produce the test vector.

C. Theory

1) Data description: In a k-class classification problem let there be l training vectors per class and let each training vector have m attributes. So the total number of training vectors is k $\times$ l equal to n training vectors. These n observations are samples from the different k low dimensional subspaces.

2) Forming the spanning sets: A spanning set for a particular class' subspace is formed by horizontally concatenating the training vectors of a particular class. Since each training vector is in $\mathbb{R}^m$, by concatenating l such training observations, a spanning set for a subspace is formed as a matrix in $\mathbb{R}^{m \times l}$.

If v_{ij} denotes the the i_{th} training vector in the j_{th} class, the spanning matrix for the j_{th} class is given by equation 1.

$$D_j = [v_{1j}, v_{2j}, ..., v_{l1j}, v_{lj}] \quad (1)$$

Here D_j is a spanning set for the subspace j.

3) Forming the Dictionary Matrix: The dictionary matrix is a collection of spanning sets for all the classes involved in the classification problem. So the dictionary matrix D, is formed by horizontally concatenating all the spanning matrices from D_1 to D_k as given in equation 2

$$D = [D_1, D_2, ..., D_k] \quad (2)$$

Simply put, the dictionary matrix D is a collection of all the training vectors across all classes, arranged as column vectors existing in $\mathbb{R}^{m \times n}$. These column vectors are the n samples taken from the k different subspaces.

4) geometric interpretation: The figure 1 presents a geometrical picture of the learning problem. For now let us assume that the white data point is our test vector. Assume that the learning problem here involves four classes, hence four subspaces 'A', 'B', 'C', and 'D' are pictured in the figure 1. In this case the test vector can be realized, by only a linear combination of the training vectors of class 'A'. This is because the test vector lies on the subspace 'A' because it belongs to class 'A'.

This is an ideal case scenario because the test vector lies on the subspace 'A' and there is no noise component to move the test vector off the subspace 'A'. The training vectors can combine linearly to realize the test vector with zero residual. This is given in equation 3.

$$\alpha_1 v_{1A} + \alpha_2 v_{2A} + ... + \alpha_{l-1} v_{l-1A} + \alpha_l v_{lA} = Y \quad (3)$$

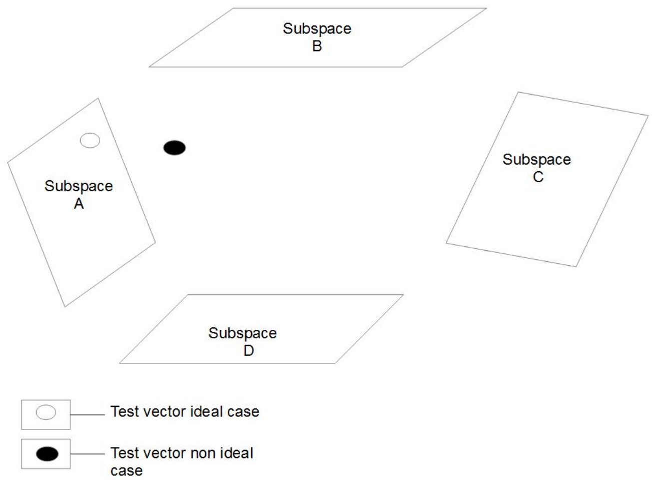

Figure 1. Figure providing a geometrical interpretation of a four class learning problem. The white data point is the test vector in the ideal case. The black data point is the test vector in the non ideal case.

Here, vectors v_{1A} to v_{lA}, which are the training vectors for the class A, form a spanning set for the subspace A. The constants α_1 to α_l are the coefficients of the linear combination. Y is the test vector. The equation 3 can be written in matrix form as given in equation 4.

$$\begin{aligned} D_A\alpha_A &= Y \\ D_A\alpha_A - Y &= 0 \end{aligned} \tag{4}$$

Here D_A is a spanning set for the subspace A. The residual is equal to zero because the test vector lies on the subspace 'A'.

Introducing noise into the system the test vector is now moved away from the subspace 'A'. Now the black data point in figure 1 is our new test vector. It can be seen that the test vector is no longer lying on the subspace 'A' though it actually belongs to class 'A'. Here the training vectors of class 'A' can no longer combine linearly to realize the test vector. Hence this scenario is regarded as the non ideal case. For such an event class is assigned to the test vector depending on the subspace the test vector is nearest to. So, if the test vector is nearest to the subspace 'A', then the test vector is classified as class 'A'.

5) Need for a Sparse Solution: This section discusses the need for a sparse solution in the setting of the k class classification problem that was discussed earlier in subsection III-C1. Algebraically, the subspace the test vector is lying on (ideal case scenario) or is nearest to (non-ideal scenario) can be found by casting the problem as an $Ax = Y$ problem as given in equation 5.

$$Dx = Y \tag{5}$$

By searching for a sparse solution to the problem in equation 5, the subspace the test vector is nearest to can be determined. If the x vector in equation 5 is sparse, then the x vector will have non-zero coefficients only in the entries that multiply with the vectors of the dictionary matrix D belonging to the subspace the test vector is the nearest to. This is made more clear with the example discussed in the following paragraph.

The columns of the Dictionary matrix D are the training vectors associated with different classes. Let the columns from p to z of the Dictionary matrix be training vectors belonging to class i. So, if the test vector belonged class i then the solution vector to equation 5 will be given by equation 6

$$x = [0, 0...0....\alpha_p, \alpha_q...\alpha z, 0, 0....0] \tag{6}$$

Here, α_p is a constant denoting the magnitude of the vector along its p_{th} component. In the next section an optimization framework is discussed to determine the best possible sparse linear combination.

6) Optimization Framework: The optimization problem in equation 7 can be used to determine the subspace the given test vector is residing in. The objective function is minimizing the one norm of the solution vector. One norm minimization is done in order to achieve a sparse x vector. The reason why one norm minimization results in a sparse solution is because the norm ball is a diamond, thereby forces the constraint $Dx = y$ to meet the norm ball only along any one of the axes. This leads to sparsity in the solution vector.

$$\begin{aligned} &\underset{x}{\text{minimize}} \|x\|_1 \\ &\text{such that } Dx = Y \end{aligned} \tag{7}$$

The equation 7 is effective only in the ideal case. In the non ideal case the optimization problem becomes infeasible as the constraint is almost never satisfied. This is because the test vector is moved off the subspace it is supposed to lie in. This problem is overcome in the formulation given by equation 8.

$$\begin{aligned} &\underset{x}{\text{minimize}} \|x\|_1 \\ &\text{such that } \|Dx - Y\|_2 \leq \epsilon \end{aligned} \tag{8}$$

Here the constraint is relaxed and allows the residual to be lesser than or equal to the regularization constant ϵ. Hence in this optimization formulation even if the test vector is moved off the subspace, the optimization problem is still solvable as the residual is allowed to be lesser than or equal to ϵ.

7) Classification: In the case of both equations 7 and 8, the solution vector owing to the forced sparsity, will have non zero coefficients only at entries that will multiply with the training vectors of a particular class in the dictionary matrix. This is given by equation 6. So by seeing the positions of the non zero coefficients in the test vector, the

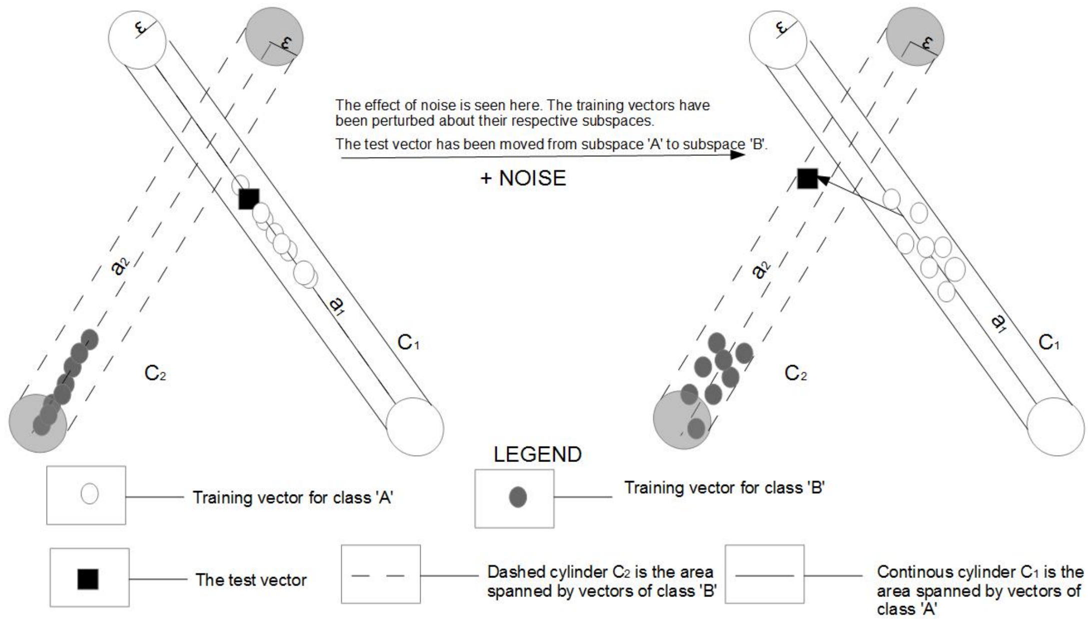

Figure 2. Figure describing the effect of noise on the data points.

training samples that are involved in the linear combination can be found. Typically, most training samples will belong to the correct class, with a few training samples belonging to other classes. The test vector is assigned the class that is the most dominant.

IV. Point of breakdown of the sparse representation learner

Previously the paper discussed a couple of scenarios the sparse representation learner can handle. Now, we discuss a more general scenario which causes sparse representation leaner to mis-classify. In this scenario noise component could push the test vector to a subspace, that is not the subspace the test vector actually belongs to. That is the test vector could belong to class ’A’ but could be pushed to the subspace spanned by the training vectors of class ’B’. In this event the sparse representation learner mis-classifies the test vector as class ’B’. This scenario is where the algorithm breaks down. In the next section a novel optimization framework is discussed to make the learning algorithm more robust to such events.

V. Clustering exploiting sparse representation learner

This section introduces a learning framework that is an improvement over the existing sparse representation learner. The optimization framework given by equation 8 is modified to add a few extra attributes to the sparse representation learner. These new attributes help the learning framework tackle the issues faced by the sparse representation learner and overcome the point of break down that is described in section IV.

A. Motivation

As mentioned in the introduction the proposed work is designed to exploit both the clustering and the subspace property. These two properties are used as two way checks to determine if a test vector belongs to a particular class. To give a deeper insight of how the exploitation of these two properties work, let us go to the example that is discussed in the ensuing paragraph.

Let data points be positioned as shown in the figure 2. It can be seen in the left side of the figure 2 that, the axes a_1 and a_2 of the cylinders C_1 and C_2 are the true subspaces for the classes ’A’ and ’B’. The training vectors can be seen lying on the axes of the respective cylinders C_1 and C_2. The constraint $\|Dx - Y\|_2 \leq \epsilon$ enables the training vectors to be able to realize the vectors that are ϵ units away from the axis. This is the reason why the space realizable by the training vectors is a cylinder with a radius of ϵ. There are two observations to make here, one is that the data points of the same class lie on the same subspace, and two is that data points of the same class, cluster near one portion of the subspace. Thus both the properties have been held. The test

vector as it belongs to class 'A', is seen lying in the space that is cylinder C_1 and also is seen in the cluster formed by the training vectors of class 'A'. Here, the proposed learning algorithm will rightly adjudge the test vector as class 'A', as the test vector is a part of the subspace and the cluster. It is to be noted that the sparse representation learner will also predict the right class.

The scenario discussed in the previous section is an ideal scenario, let us see what happens when noise is introduced into the system. The effect of noise is shown in the right hand side of the figure 2. Here the test vector and the training vectors have been moved. It is to be noted that the test vector is no longer lying within the cylinder C_1, it has been moved to C_2. Here only the training vectors of class 'B' can combine linearly to realize the test vector. Sparse representation learner will wrongly adjudge the test vector as class 'B'. This is the case where sparse representation learner breaks down. Now one important observation to make is that, even though noise has moved the test vector to the cylinder C_2, the test vector is actually closer to the cluster formed by the the training vectors of 'A', rather than the cluster formed by training vectors of 'B'. Here only the subspace property is violated, the clustering property is still held. This observation is what the proposed algorithm is designed to exploit. This algorithm classifies test vectors not only by determining the subspace the test vector is the nearest to, but also the cluster the test vector is the nearest to. This is the motivation behind the formulation of the clustering exploiting sparse representation learner. In the next subsection a mathematical framework is discussed to exploit this observation.

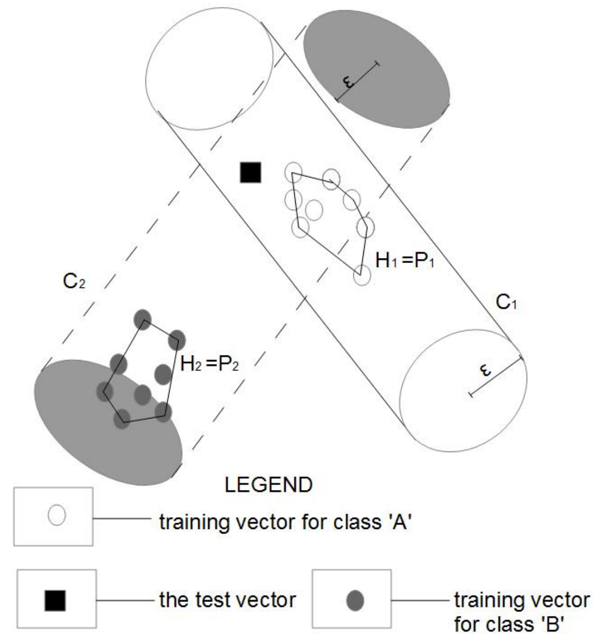

Figure 3. The intersection of the continuous line cylinder C_1 and the continuous line polygon P_1 is the region R_1, which is the space spannable by the training vectors of class 'A'. The intersection of the dashed line cylinder C_2 and the dashed line polygon P_1 is the region R_2, is the space spannable by the training vectors of class 'B'. Here H_1 and H_2 are convex hulls and are the same as P_1 and P_2. The regions R_1 and R_2 are not marked in the figure.

B. Optimization framework

This section describes the construction of a novel optimization framework that is capable of exploiting both the subspace and the clustering property simultaneously. The novel optimization framework proposed in this paper is built upon by effecting a set of changes to the optimization framework discussed in equation 8. The different changes brought about are discussed as this section proceeds.

1) Subspace and Clustering Exploiting Model: In order to exploit both the subspace and the clustering property, the optimization problem framed in 8 is changed to the optimization framework as given in 9.

$$\begin{aligned} &\underset{x}{\text{minimize}} \|x\|_1 \\ &\text{such that,} \\ &\|Dx - Y\|_2 \leq \epsilon \\ &\|x\|_1 = 1 \\ &x_i \geq 0 \\ &\text{where } 1 \leq i \leq n \end{aligned} \tag{9}$$

It can be seen that the optimization framework in equation 9 is the same as that in equation 8, only difference is that two extra constraints, $\|x\|_1 = 1$, and $x_i \geq 0$, have been added. These two constraints force the columns of the dictionary matrix D to combine convexly. The combination of the columns of the dictionary matrix is given by equation 10.

$$= x_1 v_1 + x_2 v_2 + ... + x_n v_n \tag{10}$$

Here v_1 to v_n are the columns of the dictionary matrix D. For a combination of vectors to be convex, two conditions are to be satisfied, one is that sum of the individual components of the x vector has to be one and the other is that the individual components should be non-negative. So because of the constraints $\|x\|_1 = 1$ and $x_i \geq 0$, the equation 10 is a convex combination.

The objective function in equation 9 has the same purpose as in equation 8, it is used to introduce sparsity in x, thereby forcing the learner to use only a very few columns of the dictionary matrix to realize the test vector Y. The purpose of the $\|Dx - Y\|_2 \leq \epsilon$ constraint is the same as in equation 8, helps the learner to deal with test vectors that do not lie on any of the subspaces. Now that the purpose of the different parts of the optimization problem is discussed, let us see how these parts work in tandem to solve the problem

scenario discussed in figure 3.

The scenario in figure 3, is the same as in figure 2, just that a convex hull is drawn around each of the clusters. For the constraints given in the optimization framework by equation 9, the figure 3 gives the space that can be spanned by the training vectors of classes A and B. The space spanned by the training vectors owing to the convexity constraints $\|x\|_1 = 1$, and $x_i \geq 0$ are the polygons P_1 and P_2. These constraints force the training vectors of a particular class to combine convexly, thereby making only vectors within the convex hull realizable. The infinitely long extending cylinders C_1 and C_2 are formed because of the $\|Dx - Y\|_2 \leq \epsilon$ constraint. The intersection of the polygons P_1 and P_2 with the infinitely extending cylinders C_1 and C_2 are the regions R_1 and R_2 (R_1 and R_2 are not denoted in the figure 3). The regions R_1 and R_2 are the spannable regions for the vectors of classes 'A' and 'B' respectively. Now, if the test vector is in region R_1, it is classified as class 'A' and if it is in region R_2, it is classified as class 'B'. But in the figure 3, the test vector does not lie in both the regions R_1 and R_2. Hence in this case the optimization problem is infeasible, and the learner outputs a symbol saying that it is uncertain about the class the test vector belongs to. This is kind of an improvement over the learning framework given in equation 8, where the learner wrongly classifies the test vector as class 'B'. It is to be noted that for this problem formulation in equation 9, the polygons P_1 and P_2 may seem, just as same as the convex hulls H_1 and H_2, but in the next formulation these polygons have more significance and this is made clearer as the paper proceeds.

Now, let us discuss the issues with the optimization framework given by equation 9. Here the issue with the optimization framework is the strict convex closure introduced by constraints $\|x\|_1 = 1$, and $x_i \geq 0$. Now, if the polygon P_1 were a little larger than the convex hull H_1, the test vector would have been in the region R_1 and would have been classified correctly. So this problem could be tackled by having control over the size of the polygons. This can be done by the optimization framework given in equation 11

2) Regularized Subspace and Clustering Exploiting Model:

$$
\begin{aligned}
&\underset{x,\hat{x}}{\text{minimize}} \|x + \hat{x}\|_1 \\
&\text{such that,} \\
&\|Dx - Y\|_2 \leq \epsilon \\
&\|D\hat{x} - Y\|_2 \leq \hat{\epsilon} \\
&\|\hat{x}\|_1 = 1 \\
&\hat{x_i} \geq 0 \\
&\text{where } 1 \leq i \leq n
\end{aligned}
\tag{11}
$$

The equation 11 is built from equation 9 by altering the objective and adding an extra, $\|D\hat{x} - Y\|_2 \leq \hat{\epsilon}$ constraint. As discussed in the previous paragraph controlling the size would have enabled the learning algorithm to predict the test vector correctly as label 'A'. The new constraint $\|D\hat{x} - Y\|_2 \leq \hat{\epsilon}$ provides control over the size of the polygons P_1 and P_2. Now by varying $\hat{\epsilon}$, the size can be varied. It is to be noted that the convex hulls are still the same, only that the region spanned by the training vectors by convex combination has increased to polygons P_1 and P_2. This is possible because the $\hat{\epsilon}$ term allows the test vectors to be $\hat{\epsilon}$ units away from the hulls. The $\hat{\epsilon}$ is another regularization constant by increasing its value the variance is lowered and by lowering its value the bias is lowered.

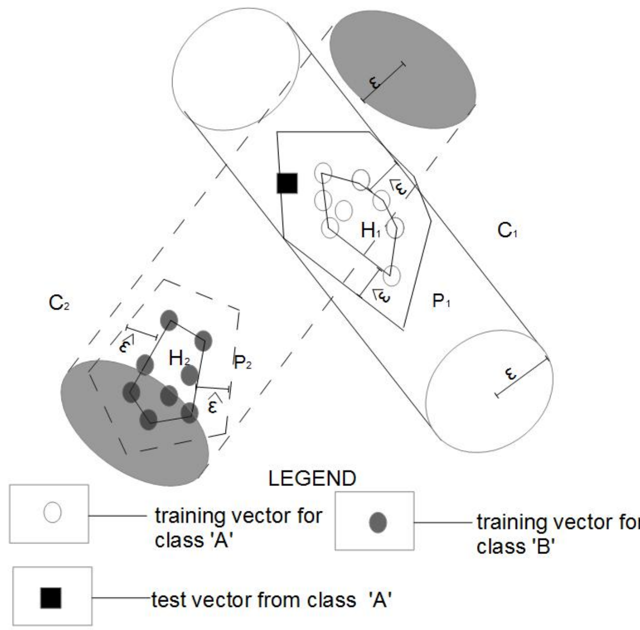

Figure 4. This figure shows that by varying the $\hat{\epsilon}$ parameter the size of the polygons can be varied. It can be seen that the polygons in this figure are larger than the ones in the previous figure 3. The figure shows both the polygons P_1 and P_2 growing by ϵ units in all directions

In figure 4 shows how $\hat{\epsilon}$ can be varied to make the learner output the right label. Now that $\hat{\epsilon}$ is increased, the region R_1 grows to the extent that the test vector is now a part of the region R_1. Since the test vector is a part of R_1 it is correctly classified as label 'A'.

3) Classification: Classification here is very similar to the classification process described in section III-C7, here the sum of x and $\hat{x}$ is taken as vector x_c as given in equation 12

$$x_c = x + \hat{x} \tag{12}$$

This vector x_c will be sparse and will have non zero coefficients only at entries that multiply with columns vectors that are a part of the subspace the test vector lies in, and

the cluster that the test vector is a part of. Hence by seeing the positions of the non zero coefficients the cluster and the subspace the test vector is can be determined. Typically both the cluster and the subspace are contain training vectors of the same class, but if they contain different classes, then classification is done by determining if the test vector is closer to the subspace or to the cluster.

C. Discussion

On seeing the figures 3, and 4, readers could feel that the constraint $\|Dx-Y\|_2 \leq \epsilon$ has little control over determining the regions R_1 and R_2 as the polygons are P_1 and P_2 are almost a subset of the cylinders C_1 and C_2. But the constraint $\|Dx-Y\|_2 \leq \epsilon$ is essential, the x-vector encodes the distance between the test vector and the nearest subspace. The $\hat{x}$ vector encodes the distance between the test vector and the nearest cluster. The proposed work uses both the details that have been encoded by vectors x and $\hat{x}$, as the sum of these two vectors is used in classification.

Now, equation 11 is the final optimization formulation for our learner. This formulation enables the sparse representation learner to exploit the clustering property, hence the name.

VI. RESULTS AND ANALYSIS

In this section, the experimentation methodology along with the environment in which the experiments were conducted is described. Then the background of the datasets that are being to used to compare the performance of the sparse representation and the sparse representation sparse representation learner is discussed. Then finally the section concludes after performing a head to head comparison with the original sparse representation learner and the proposed algorithm.

A. Experimenting Methodology

Standard machine learning practices are carried out to estimate the accuracy of an algorithm for a given dataset. Here first the algorithms are trained with samples whose classes are known. Then the test vector is provided to the trained learner for the classification process. The output labels from the learner are compared with the ground truth table to determine the number of test cases that have been rightly classified, from this, the different accuracies of the algorithm for different datasets are determined.

It is to be noted that this project along with all its associated experiments was run on a fourth generation i5 processor. The code for the project was written in MATLAB, and different optimization problems were solved during the course of the project with the help of cvx software[9].

B. About the Datasets

In order to support the idea, that the presence of subspace and clustering property is safe to assume without any loss of generality, a diverse set of datasets has been chosen. There are five datasets that have been chosen and a brief summary is given about each of these datasets.

1) Tagme Dataset: This dataset is a five class image classification problem. Here the learning algorithm is trained with a hundred images of buildings, shoes, cars, humans, and flowers, in total five hundred images. These are the five classes involved in the classification problem. Here classification is performed on fifty images, ten from each of the classes.

2) Ecoli Dataset: This is a dataset that is freely available in the UCI college repository for datasets. This is a 8 class classification problem, with 8 attributes for each observation. The dataset is typically used in protein localization sites determining projects. Based on a set of 8 attributes one of the 8 different protein localization sites such as, cp (cytoplasm) im(inner membrane without signal sequence), pp(perisplasm), imU (inner membrane, uncleavable signal sequence), om (outer membrane), omL (outer membrane lipoprotein), imL (inner membrane lipoprotein), imS (inner membrane, cleavable signal sequence) has to be predicted as the localization site for Ecoli bacteria. The 8 different protein localization sites are the 8 classes. There are 252 training and 84 testing samples in this dataset.

3) Tamil OCR Dataset: This dataset is obtained from hp's Tamil character dataset. The dataset in total has more than three hundred class, but only sixteen classes are chosen. The sampled dataset made from the original hp dataset consists of one hundred and twenty eight training vectors and thirty two test vectors across 16 classes.

4) English OCR Dataset: This is a self made dataset. The data set consists of images of English characters. This is a twenty six class problem. There are two hundred and eight training samples and fifty two testing samples.

5) Alabone Dataset: This dataset is again obtained from the UCI college repository for datasets. This is a 29 class problem, with each class having eight attributes. The learner is used to predict the age of an alabone using the eight attributes. Here four thousand training samples and hundred and seventy seven testing samples are used.

C. Analyzing the Performance

In this section empirical backing is provided to the theory discussed in the paper. The main focus of the paper is to show the improvement that is brought about by the proposed algorithm, which is clearly captured by the last column of the table I. The last column which is the "The increase in accuracy" column, has comparable values for four of the five datasets. This shows the stability of the proposed algorithm. Owing to the improvements obtained across these diverse set of datasets, one can safely assume from the table that the proposed algorithm offers a increase in accuracy by five percent to ten percent. Purely on a head to head comparison between sparse representation and clustering

Name of the dataset	Number of classes	Total number of training samples	Total number of test cases	Accuracy of the sparse representation learner	Accuracy of the clustering exploiting sparse representation learner	Percentage of increase in accuracy
Tagme dataset [12]	5	500	50	29/50= 58%	33/100=%64	6%
Ecoli dataset [13]	8	252	84	58/84= 69.04%	64/84=76.1%	7%
Tamil OCR dataset [11]	16	128	32	20/32=62.5%	23/32= 71.8%	9.5%
English OCR	26	208	52	46/52= 88.46%	49/52= 94.23%	6%
Alabone dataset [10]	29	4000	150	39/150=26%	42/150=28%	2%

Table I
TABLE COMPARING THE PERFORMANCES OF THE SPARSE REPRESENTATION LEARNER AND THE CLUSTERING EXPLOITING SPARSE REPRESENTATION LEARNER ON DIFFERENT DATASETS.

exploiting sparse representation learner, the latter does better than the former. Here only the improvement brought about by the proposed learning algorithm is important and not the actual accuracy. This is is because the paper aims to show only the effectiveness of the clustering exploiting sparse representation learner over the original sparse representation learner, and the last column of the table I very clearly shows this.

VII. CONCLUSION AND DISCUSSION

The authors plan to extend this work by studying how the two regularization constants ϵ and $\hat{\epsilon}$ can be tuned to extract maximum efficiency from the proposed learning algorithm. We want to conduct experiments to study the effects of the two regularization constants on the variance and the bias. This is why the current work only outlines the effect of the two parameters.

Finally we conclude that, the paper brings to the field, clustering exploiting sparse representation learner, which is a novel framework designed to exploit both the clustering and the subspace property. The required mathematical framework to perform the above, is well defined and is supported both by theory and experimental results. Our results clearly suggest a 5% to 10% improvement in accuracy over the existing sparse representation framework.

REFERENCES

[1] 'Wright, John, et al. "Robust face recognition via sparse representation." Pattern Analysis and Machine Intelligence, IEEE Transactions on 31.2 (2009): 210-227.

[2] Hearst, Marti A., et al. "Support vector machines." Intelligent Systems and their Applications, IEEE 13.4 (1998): 18-28.

[3] Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." Applied statistics (1979): 100-108.

[4] Yang, Jianchao, et al. "Image super-resolution via sparse representation." Image Processing, IEEE Transactions on 19.11 (2010): 2861-2873.

[5] Agarwal, Shivani, and Dan Roth. "Learning a sparse representation for object detection." Computer VisionECCV 2002. Springer Berlin Heidelberg, 2002. 113-127.

[6] Gao, Shenghua, Ivor Wai-Hung Tsang, and Liang-Tien Chia. "Kernel sparse representation for image classification and face recognition." Computer VisionECCV 2010. Springer Berlin Heidelberg, 2010. 1-14.

[7] Elhamifar, Ehsan, and Ren Vidal. "Robust classification using structured sparse representation." Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011.

[8] Yang, Meng, et al. "Fisher discrimination dictionary learning for sparse representation." Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

[9] CVX Research, Inc. CVX: Matlab software for disciplined convex programming, version 2.0. http://cvxr.com/cvx, April 2011. M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs, Recent Advances in Learning and Control (a tribute to M.Vidyasagar), V. Blondel, S. Boyd, and H. Kimura, editors, pages 95-110, Lecture Notes in Control and Information Sciences, Springer, 2008. $[http://stanford.edu/\ boyd/graph_dcp.html.]$

[10] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml/datasets/Abalone]. Irvine, CA: University of California, School of Information and Computer Science.

[11] HPL Isolated Handwritten Tamil Character Dataset [http://lipitk.sourceforge.net/datasets/tamilchardata.htm].

[12] Presented by (TagMe!) $http://events.csa.iisc.ernet.in/opendays2014/events/MLEvent/$

[13] Lichman, M. (2013). UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/Ecoli]. Irvine, CA: University of California, School of Information and Computer Science.