Robust Estimation for Subspace Based Classifiers

Hemant Agrawal
IIIT Delhi
New Delhi, India
hemanta@iiitd.ac.in

Angshul Majumdar
IIITD
New Delhi, India
angshul@iiitd.ac.in

***Abstract*— The nearest subspace classifier (NSC) assumes that the samples of every class lie on a separate subspace and it is possible to classify a test sample by computing the distance between the test sample and the subspaces. The sparse representation based classification (SRC) generalizes the NSC – it assumes that the samples of any class can lie on a union of subspaces. By calculating the distance between the test sample and these subspaces, one can classify the test sample. Both NSC and SRC hinge on the assumption that the distance between the test sample and correct subspace will be small and approximately Normally distributed. Based on this assumption, these studies proposed using an l_2-norm measure. It is well known that l_2-norm is sensitive to outliers (large deviations at few locations). In order to make the NSC and SRC robust and improve their performance we propose to employ the l_1-norm based distance measure. Experiments on benchmark classification problems, face recognition and character recognition show that the proposed method indeed improves upon the basic versions of NSC and SRC; in fact our proposed robust NSC and robust SRC yield even better results than support vector machine and neural network.**

***Index Terms*— classification, robust estimation, face recognition, character recognition.**

I. Introduction

Perhaps one of the simplest classifiers is the nearest neighbour (NN). Here the distance between a test sample and all training samples are calculated and the test sample is assigned to the class of the training samples having the minimum distance. The nearest neighbour approach is generalized to KNN where instead of assigning the test sample to the class having the minimum distance, top-K minimum distances are considered and the test sample is assigned to the class where most of these top K-samples belongs to.

The sparse representation based classification (SRC) [1] has enjoyed a large popularity since its inception. Here it is assumed that the training samples of a particular class form a linear basis for a test sample belonging to that class. The test sample is expressed as a linear combination of all training samples with the underlying assumption that the combination weights will be mostly zeroes (for training samples belonging to the incorrect class) and will have non-zero values only corresponding to the correct class; in short the combination weights were supposed to be sparse. A sparsity promoting optimization was used to solve for the combination weights. A representative sample for each class was formed by linearly combining the training samples and the corresponding combination weights. The distance between the test sample and the representative sample for each class was computed; the test sample was assigned to the class having the minimum distance.

Geometrically speaking, somewhere between the NN and the SRC lies the nearest subspace classifier (NSC) [2, 3]. The NSC assumes that training samples for each class belongs to only one subspace. Therefore the classification task boils down to the problem of finding the subspace nearest to the test sample. At one extreme of NSC lies the simple NN classifier where each subspace is spanned by a single sample. On the other extreme is the SRC. SRC allows the training samples of each class to lie in a union of subspaces; the sparsity promoting optimization problem can effectively recover solutions lying in such a union of subspaces [4]. Thus SRC is a generalization of NSC since it allows the training samples to span multiple subspaces.

One must remember there is no free lunch. SRC achieves this generalization at significantly higher computational cost. The projection for NSC can be pre-computed (before the test sample is available); thus during testing one only needs to perform a few (same as the number of classes) matrix vector multiplications. The computational complexity is therefore O(n^3). On the other hand, SRC requires solving a sparse optimization problem after the test sample is made available. This needs to be solved iteratively; the number of iterations is about O($n^{.5}$). Every iteration consists of two matrix vector products whose complexity is O(n^3), therefore the overall complexity is O($n^{3.5}$). This is significantly larger than the complexity of NSC. In spite of computational advantages NSC is not very widely used; there are a handful of studies on this topic [6-9]. Some theoretical insights into this approach is also available [10].

Both SRC and NSC are basically subspace based approaches. In a nutshell, NSC assumes all the samples of a class to lie on a subspace whereas SRC allows the samples to lie on a union of subpsaces. Both SRC and NSC hinge on the assumption that the subspace based model is accurate, imperfections if any, are small. Both of them employ a Euclidean distance based cost function based on the said assumption.

The Euclidean distance is optimal when the deviations are small – approximately Normally distributed; but fail when there are large outliers. In statistics there is a large body of literature on robust estimation. The Huber function [11] has been in use for more than half a century in this respect. The Huber function is an approximation of the more recent absolute distance based measures (l_1-norm). Recent studies in robust estimation prefer minimizing the l_1-norm instead of the Huber function [12]-[14]. The l_1-norm does not bloat the distance between the estimate and the outliers and hence is robust.

The problem with minimizing the l_1-norm is computational. However, over the years various techniques have been developed. The earliest known method is based on Simplex [15]; Iterative Reweighted Least Squares [16] used to be another simple yet approximate technique. Other approaches include descent based method introduced by [17] and Maximum Likelihood approach [18].

In this work we follow the studies in robust estimation and propose robust versions of SRC and NSC. We employ an l_1-norm instead of the standard Euclidean (l_2-norm) distance to handle outliers. This leads to a complex optimization problem, but yields considerably better results than their non-robust counterparts on many benchmark classification datasets.

II. Subspace Based Classification Approaches

A. Nearest Subspace Classifier

Nearest Subspace Classifier (NSC) assumes that the training samples of each class form a subspace, this is depicted in (1).

$$\left[\underset{subspace\ 1}{X_1} \ , \underset{subspace\ 2}{X_2} \ ,..., \underset{subspace\ C}{X_C} \right] \qquad (1)$$

If each of the classes form a subspace, the test sample belonging to that class can be represented as a linear combination of training samples from that class, i.e.

$$x_{test} = X_c \alpha_c + \eta, \eta \sim \qquad (2)$$

Here η depicts the modelling error.

The combination weights can be easily calculated assuming the modelling error to be Normally distributed:

$$\alpha_c = \min_{\alpha} \left\| x_{test} - X_c \alpha_c \right\|_2^2 \qquad (3)$$

This has an analytic closed form solution (assuming the dimensionality is larger than the number of samples – which is usually the case):

$$\alpha_c = \left(X_c^T X_c \right)^{-1} X_c^T x_{test} = X_c^{\dagger} x_{test} \qquad (4)$$

where $X_c^{\dagger}$ denotes the Moore-Penrose pseudoinverse.

To compute the magnitude of the difference vector between x_{test} and its projection onto the subspace spanned by X_c, one simply needs to calculate:

$$\varepsilon(c) = \left\| x_{test} - X_c X_c^{\dagger} x_{test} \right\|_2^2 = \left\| \left(I - X_c X_c^{\dagger} \right) x_{test} \right\|_2^2 \qquad (5)$$

The term within the parenthesis $\left(I - X_c X_c^{\dagger} \right)$ can be pre-computed. Thus during testing one just needs to compute the matrix vector product and calculate its norm.

For classification the distance between the test sample and every class is computed and the sample is assigned to the class having the minimum error.

We have discussed the scenario where the dimensionality of the training samples is larger than the number of samples in the class. This is the usual scenario but not a very conducive one. To get good results one would like to have larger number of samples that the dimensionality of the samples; in that case (2) would be under-determined – there are infinitely many solutions. The simplest one is the minimum energy solution –

$$\alpha_c = \min_{\alpha} \left\| x_{test} - X_c \alpha_c \right\|_2^2 + \lambda \left\| \alpha \right\|_2^2 \qquad (6)$$

This too has a closed form solution

$$\alpha_c = \left(X_c^T X_c + \sqrt{\lambda} I \right)^{-1} X_c^T x_{test} \qquad (7)$$

The rest can be computed as before.

B. Sparse Representation based Classification

The SRC assumes that the training samples of a particular class approximately form a linear basis for a new test sample belonging to the same class. One can write the aforesaid assumption formally. If x_{test} is the test sample belonging to the kth class then,

$$x_{test} = \alpha_{c,1} x_{c,1} + \alpha_{c,2} x_{c,2} + ... + \alpha_{c,n_k} x_{c,n_k} + \eta \qquad (8)$$

where $x_{c,i}$ are the training samples and η is the approximation error.

In a classification problem, the training samples and their class labels are provided. The task is to assign the given test sample with the correct class label. This requires finding the coefficients $\alpha_{c,i}$ in equation (8). Equation (8) expresses the assumption in terms of the training samples of a single class. Alternately, it can be expressed in terms of all the training samples so that

$$x_{test} = X\alpha + \eta \qquad (9)$$

where $X = [x_{1,1} \,|\, ... \,|\, x_{n,1} \,|\, ... \,|\, x_{c,1} \,|\, ... \,|\, x_{c,n_c} \,|\, ... x_{C,1} \,|\, ... \,|\, x_{C,n_C}]$

and $\alpha = [\alpha_{1,1} ... \alpha_{1,n_1} ... \alpha_{c,1} ... \alpha_{c,n_c} ... \alpha_{C,1} ... \alpha_{C,n_C}]^T$.

According to the SRC assumption, only those α's corresponding to the correct class will be non-zeroes. The rest are all zeroes. In other words, α will be sparse. Therefore, one needs to solve the inverse problem (9) with sparsity constraints on the solution. This is formulated as:

$$\min_{\alpha} \left\| x_{test} - X\alpha \right\|_2^2 + \lambda \left\| x \right\|_1 \qquad (10)$$

Once (10) is solved, the representative sample for every class is computed: $x_{rep}(c) = \sum_{j=1}^{n_c} \alpha_{c,j} v_{c,j}$. It is assumed that the test sample will look very similar to the representative sample of the correct class and will look very similar, hence the residual $\varepsilon(c) = \left\| x_{test} - x_{rep}(c) \right\|_2^2$, will be the least for the correct class. Therefore once the residual for every class is obtained, the test sample is assigned to the class having the minimum residual.

C. Iterative Re-weighted Least Squares

Solutions to linear inverse problems in the presence of Gaussian noise is well known.

$$b = Az + \eta \qquad (11)$$

It can be solved by minimizing the least squares.

$$\hat{z} = \min_z \|b - Az\|_2^2 \qquad (12)$$

It has a closed form solution (13) –

$$\hat{z} = (A^T A)^{-1} A^T b \qquad (13)$$

Solution of linear inverse problems where the solution is sparse has also been studied widely in signal processing and machine learning, especially after the advent of Compressed Sensing [19, 20]. Usually it is formulated as follows,

$$\hat{z} = \min_z \|b - Az\|_2^2 + \lambda \|z\|_1 \qquad (14)$$

It (14) does not have a closed form solution but can be solved iteratively using proximal methods.

Studies [11-18] have explored various techniques for robust regression, where the error (ε) in (11) is not Normally distributed; but is sparse and large in magnitude. In such cases, for robust estimation, the mean absolute distance is minimized instead –

$$\min_z \|b - Az\|_1 \qquad (15)$$

One of the methods for solving (15) is based on the iterative re-weighted least squares approach (IRLS) [16]. In IRLS, the l_1-norm is approximated as a weighted l_2-norm, i.e.

$$\|b - Az\|_1 \approx \|W(b - Az)\|_2^2 \qquad (16)$$

where $W = diag\left(b_{(i)} - (Az)_{(i)}\right)^{-1/2}$; the subscript denotes the ith component.

IRLS, as the name suggests is an iterative technique. It starts with the least squares solution. Then it computes the diagonal weight matrix based on the z from the first iteration. With the newly computed weight matrix it solves (16) in the second iteration. This is easy to solve since it is just a least squares problem.

IRLS is a simple algorithm; the heursitic is easy to comprehend. But the approximation of l_1-norm by a weighted l_2-norm only holds at convergence. Thus the IRLS reaches the actual solution asymptotically, i.e. after a large number of iterations.

In the initial days of Compressed Sensing, IRLS was used to solve (14) as well [21]. It recast (14) as,

$$\min_z \|b - Az\|_2^2 + \lambda \|Wz\|_2^2 \qquad (17)$$

As before, $W = diag\left(z_{(i)}\right)^{-1/2}$.

The new problem (17) is just a Tikhonov regularized least squares problem; it has a straightforward closed form solution.

$$\hat{z} = (A^T A + W^T W)^{-1} A^T b \qquad (18)$$

Starting with W=Identity in the first iteration, (17) is iterated until convergence. The problem remains the same as before; the equivalence between l_1-norm and l_2-norm happens only asymptotically [22].

The IRLS method has also been used to solve the l_1-regularized l_1-minimization problem [23] of the form,

$$\min_z \|b - Az\|_1 + \lambda \|z\|_1 \qquad (19)$$

The trick is to replace both the cost function and the penalty with the corresponding weighted l_2-norms.

$$\min_z \|W_1(b - Az)\|_2^2 + \lambda \|W_2 z\|_2^2 \qquad (20)$$

As before, (20) is solved iteratively by updating W_1 and W_2. The algorithm starts with both of them being Identity. Robust Estimation of Subspaces.

III. Robust Subspace Based Classification

A. Robust Nearest Subspace Classifier

The NSC assumes that the subspace based model holds well in practice and that the modelling error if any is small and approximately Normally distributed (2). This may not be the case, especially when there are outliers. In such a case, the deviation from the actual model is large but sparse. The modelling error follows a more heavy tailed distribution. To robustly estimate in such a scenario, the l_1-norm is a more appropriate choice [12-18]. Thus, in this work we propose to estimate the coefficient α_c by minimizing the following:

$$\alpha_c = \min_\alpha \|x_{test} - X_c \alpha_c\|_1 \qquad (21)$$

As mentioned before [15-18], there are several techniques to solve (11). The most practical way being the reweighted least squares method [16]. We have discussed it in the previous section. The issues associated with such heuristics methods has also been discussed. In this work we follow a more elegant approach to solve (21) exactly (as opposed to approximate solutions like IRLS) based on the Augmented Lagrangian formulation.

For simplicity of notation, we drop the subscripts from (21) and introduce a proxy variable: *p=x-Xα*. The problem (21) is therefore expressed as,

$$\min_{D,Z,P} \|x\|_1 \ s.t. \ p = x - X\alpha \qquad (22)$$

The unconstrained Lagrangian for (22) is,

$$L = \|p\|_1 + \mu^T (p - x + X\alpha) \qquad (23)$$

The Lagrangian enforces strict equality; this is not required. One only needs to enforce strict equality at convergence. Therefore one can relax the equality constraint and use the Augmented Lagrangian instead.

$$AL = \|p\|_1 + \mu \|p - x + X\alpha\|_F^2 \qquad (24)$$

The value of μ controls the relaxation; for small values the equality constraint between *p* and *x-Xα* is relaxed, and for high values it is enforced. One way to achieve this is to start with a small value of μ, solve (24); increase the value, solve (24) again and so on.

A more elegant solution is to introduce a Bregman relaxation variable (*B*) [24, 25] –

$$\min_{\alpha,p} \|p\|_1 + \mu \|p - x + X\alpha - b\|_F^2 \qquad (25)$$

Instead of tinkering with μ, one can update b iteratively. The update is based on simple gradient descent and hence is very efficient. We only need to solve (25) once – for a fixed value of μ. Hence solving (25) is much less time consuming compared to (24). This approach is the so called Split Bregman technique.

One can segregate (25) into the alternating minimization of the following sub-problems:

$$\mathrm{P1}: \min_{\alpha} \|p - x + X\alpha - b\|_F^2 \qquad (26)$$

$$\mathrm{P2}: \min_{p} \|p\|_1 + \mu\|p - x + X\alpha - b\|_F^2 \qquad (27)$$

Solving P1 is straightforward – it is a least squares problem and have closed form updates. They can also be solved using conjugate gradient based methods. Also P2 has a closed form update – soft thresholding [26]. This is given by:

$$p \leftarrow signum(x - X\alpha + b)\max(0, |x - X\alpha + b| - \mu)$$

The last step is to update the Bregman relaxation variable.

$$b \leftarrow p - x + X\alpha - b$$

There are two stopping criteria for the Split Bregman algorithm. Iterations continue till the objective function converges (to a local minima). The other stopping criterion is a limit on the maximum number of iterations. We have kept it to be 200.

Once the coefficient for each class is obtained, the representative sample for every class is calculated:

$$x_{rep} = X_c\alpha_c \qquad (28)$$

We assume that the representative sample for the correct class will be close to the test sample, and hence we compute the residue between the representative sample and the test sample: $\varepsilon(c) = \|x_{test} - x_{rep}(c)\|_2^2$. The test sample, as before, is assigned to the class with the minimum residue.

B. Robust Sparse Representation based Classification

The original formulation for SRC assumed that the modelling error is small and Normally distributed, therefore an l_2-norm data fidelity term was used (10). As discussed before, the Euclidean norm is not robust to outliers. In order to have a robust estimation, we propose to replace the l_2-norm by an l_1-norm. We formulate a robust solution via:

$$\min_{\alpha} \|x_{test} - X\alpha\|_1 + \lambda\|x\|_1 \qquad (29)$$

We follow a similar approach as before. The first step in a Split Bregman formulation is to introduce a proxy variable – $p = x_{test} - X\alpha$. We add terms relaxing the equality constraints of this quantity and its proxy, and in order to enforce equality at convergence, we introduce Bregman relaxation variables *b*. The new objective function is:

$$\min_{x,p} \|p\|_1 + \lambda\|\alpha\|_1 + \mu\|p - x_{test} + X\alpha - b\|_2^2 \qquad (30)$$

This allows the problem (30) to be split into an alternating minimization of the following subproblems:

$$P_1: \min_{x} \lambda\|\alpha\|_1 + \mu\|p - x_{test} + X\alpha - b\|_2^2 \qquad (31)$$

$$P_2: \min_{p} \|p\|_1 + \mu\|p - x_{test} + X\alpha - b\|_2^2 \qquad (32)$$

The second subproblem is the same as (27) and has a closed form update – one step of soft thresholding. This has already been discussed in the previous subsection. The first subproblem needs to be solved iteratively using Iterative Soft Thresholding Algorithm [16]. The update (at kth iteration):

$$z = \alpha_k + \frac{1}{a}X^T\left(p - x_{test} + X\alpha_k - b\right)$$

$$\alpha_{k+1} = signum(z)\max\left(0, |z| - \frac{\lambda}{2\mu a}\right)$$

where a is the maximum eigenvalue of X^{T}X.

The final step of the Split Bregman technique is to update the relaxation variable:

$$b \leftarrow p - x_{test} + X\alpha - b \qquad (33)$$

There are two stopping criteria for the Split Bregman algorithm. Iterations continue till the objective function converges (to a local minima). The other stopping criterion is a limit on the maximum number of iterations. We have kept it to be 200.

Once the *α* is obtained, the classification algorithm proceeds as before, i.e. we calculate the representative sample for each class using (28) and compute the Euclidean distance between the representative sample test sample. The test sample is assigned to the class having the minimum distance.

IV. Experimental Evaluation

A. Results on Benchmark Classification Datasets

Our experiments were carried out on some well known databases from the UCI Machine Learning repository [27]. From all the databases around 10% of the samples are selected for tuning. The remaining samples are used for actual testing. Leave-one-out cross validation is used for avoiding variance due to random splits.

In the Table 1 shows the results for NSC – the original NSC (with l_2-norm) [2] and our proposed robust NSC (with l_1-norm). We compare our results against two standard classifiers: the nearest neighbor (NN) and the support vector machine (SVM).

TABLE I. NSC Classification Accuracy

Dataset	# of classes	Recognition Accuracy (%)			
		NSC	Robust NSC	NN	SVM
Page Block	5	94.78	95.32	93.34	**96.86**
Abalone	29	27.17	**26.99**	26.67	24.22
Segmentation	7	96.31	**96.88**	96.31	95.87
Yeast	10	**57.75**	**57.75**	57.71	54.32
German Credit	2	69.32	75.40	74.54	**75.88**
Tic-Tac-Toe	2	78.89	85.31	83.28	**86.72**
Vehicle	4	65.58	**74.16**	73.86	72.97
Australian Cr	2	85.94	**87.52**	86.66	87.15
Balance Scale	3	93.33	93.33	93.33	85.52

Ionosphere	2	86.94	**91.67**	90.32	**91.67**
Liver	2	66.68	**69.04**	**69.04**	**69.04**
Ecoli	8	**81.53**	81.26	80.98	80.26
Glass	7	68.43	**69.23**	68.43	**69.23**
Wine	3	**85.62**	85.51	82.21	74.69
Iris	3	**96.00**	**96.00**	**96.00**	92.00
Lymphography	4	85.81	**89.32**	85.32	88.64
Hayes Roth	3	40.23	**41.01**	33.33	34.85
Satellite	6	80.3	80.3	77.00	**89.73**
Haberman	2	40.52	**67.28**	57.40	65.22

The results show that in most cases our robust NSC yields better results than the original formulation. Only for the Ecoli and the Wine dataset, the original formulation beats us by a small margin. In most cases, our method is even better than SVM; only for German Credit, Tic Tac Toe and Satellite, does SVM beat our robust NSC. From these results, we can conclude that the proposed method improves considerably over the original NSC; it also yields better results than sophisticated classifiers like SVM and the well known NN.

Next we discuss the results for SRC classification. We compare our proposed robust SRC with the original formulation (l_2-norm) [1]; as benchmark we use the Artificial Neural Network (ANN) and SVM. The results are tabulated in the following table.

TABLE II. SRC CLASSIFICATION ACCURACY

Dataset	# of classes	Recognition Accuracy (%)			
		SRC	Robust SRC	ANN	SVM
Page Block	5	95.78	96.33	95.32	**96.86**
Abalone	29	28.39	**28.98**	26.49	24.22
Segmentation	7	**97.22**	**97.22**	96.31	95.87
Yeast	10	57.75	**58.00**	57.71	54.32
German Credit	2	77.16	**77.43**	75.40	75.88
Tic-Tac-Toe	2	85.31	**86.88**	85.31	86.72
Vehicle	4	75.51	**75.88**	73.46	72.97
Australian Cr	2	**87.64**	**87.64**	86.52	87.15
Balance Scale	3	**94.33**	**94.33**	93.33	85.52
Ionosphere	2	92.20	**94.12**	91.67	91.67
Liver	2	69.04	**70.21**	69.04	69.04
Ecoli	8	**83.45**	82.86	81.26	80.26
Glass	7	**70.19**	**70.19**	69.23	69.23
Wine	3	**95.45**	**95.45**	85.51	74.69
Iris	3	**98.67**	**98.67**	96.00	92.00
Lymphography	4	86.32	**86.81**	86.32	88.64
Hayes Roth	3	43.94	**45.38**	41.01	34.85
Satellite	6	83.15	**86.22**	80.30	89.73
Haberman	2	73.20	**77.78**	43.28	65.22

The table shows that except for one case (Ecoli) our method always gives at par or better results than the original SRC formulation. Our proposed method always better than ANN and is also better than SVM (except for a single instance – Page Block). In short, our method yields better results than existing algorithms like SRC, ANN and SVM.

B. Experiments on Face Recognition

Fig. 1. Samples from Extended Yale B

We follow the experimental protocol outlined in [1]. The experiments are carried on the Extended Yale B Face Database. For each subject, we randomly select half of the images for training and the other half for testing. Table 3 contains the results for face recognition. The features are selected using the simple Eigenface method. Although more sophisticated feature extraction techniques exist, our goal is to investigate that given the feature set how different classifiers perform. To compare our results with [1], we select the same number of Eigenfaces as proposed in [1]. We do not compare the results with SVM and ANN, since it has already shown in [1] that the SRC outperforms them for face recognition problems.

TABLE III. FACE RECOGNITION

Method	Number of Eigenfaces			
	30	56	120	504
NSC	86.49	91.71	93.87	96.77
Robust NSC	86.96	92.05	94.26	97.13
SRC	89.40	93.37	95.14	97.79
Robust SRC	**91.11**	**94.56**	**96.08**	**98.25**
NN	74.48	81.85	86.08	89.47

The results are as expected. The robust version of NSC yields better results than the original NSC formulation; the robust version of SRC yields better results than SRC [1]. In general, SRC yields better results than NSC. The nearest neighbor results are shown in Table 3 as a benchmark.

C. Experiments on Character Recognition

The MNIST digit classification task is composed of 28x28 images of the 10 handwritten digits. There are 60,000 training images with 10,000 test images in this benchmark. The images are scaled to [0,1] and we do not perform any other pre-processing.

Experiments are also carried out on the more challenging variations of the MNIST dataset [29]. These were introduced as benchmark deep learning datasets. All these datasets have 10,000 training, 2000 validation and 50,000 test samples. The size of the image as before is 28x28 and the number of classes are 10.

Dataset	Description
basic	Smaller subset of MNIST.
basic-rot	Smaller subset of MNIST with random rotations.
bg-rand	Smaller subset of MNIST with uniformly distributed random noise background.
bg-img	Smaller subset of MNIST with random image background.
bg-img-rot	Smaller subset of MNIST digits with random background image and rotation.

We compare our proposed robust versions on NSC and SRC with the original algorithms. We also show the results from SVM. These are tabulated in Table 4.

TABLE IV. CHARACTER RECOGNITION

Dataset	**SRC**	**NSC**	**SVM**	**Robust SRC**	**Robust NSC**
MNIST	98.33	87.19	88.43	**98.42**	97.16
basic	96.91	85.03	87.49	**97.03**	95.43
basic-rot	90.04	68.63	79.47	**90.19**	87.76
bg-rand	91.03	72.25	79.67	**91.69**	76.17
bg-img	84.14	65.68	75.09	**85.11**	85.84
bg-img-rot	62.46	34.01	49.68	**62.61**	47.75

Our robust SRC always yields the best results. NSC yields the worst. The disparsity between the NSC and its robust version is marked; the robust NSC yields better results than the SVM (and also NSC). The original SRC yields better results than the robust NSC.

V. CONCLUSION

In this work we improve two subspace based classification techniques. The first one is the Nearest Subspace Classifier (NSC) and the second one is the Sparse Representation based Classifier (SRC). NSC assumes that the samples of every class lie on a different subspace. In SRC the assumption is more generalized – it assumes the samples to lie on a union of subspaces. The original formulations of NSC and SRC assumed that the modelling error, if any, is small and approximately Normally distributed and hence the cost function should be the Eucidean distance. It is well known that such a cost function is sensitive to outliers. In this work we propose robust versions of NSC and SRC – we replace the Euclidean distance based cost functions with the more robust l_1-norm. One could argue that having an l_p-norm ($0<p<1$) would be more robust, but such l_p-norms are non-convex.

We verify the improvement empirically on benchmark datasets. In the first set of experiments, we compare these techniques on classification datasets from the UCI Machine Learning Repository. The second set of experiments are on the Extended YaleB face recognition database. The third database is the MNIST (and its variations) character recognition set. In all these experiments, we have seen that the robust versions of NSC and SRC perform better than their original versions; in fact in most cases our proposed techniques outperform sophisticated classifiers like support vector machine and neural network.

REFERENCES

[1] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” IEEE Trans. On Pattern Analysis and Machine Intell., vol. 31, no. 2, 2009.

[2] F. Porikli and Y. Chi, “Connecting the dots in multi-class classification: From nearest subspace to collaborative representation”, IEEE CVPR 2012.

[3] W. Zhao, “Subspace methods in object/face recognition”, IEEE IJCNN 1999.

[4] Y. C. Eldar M. Mishali, “Robust Recovery of Signals From a Structured Union of Subspaces”, IEEE Transactions on Information Theory, Vol. 55 (11), pp. 5302-5316, 2009.

[5] A. Yang, A. Ganesh, S. Sastry and Y. Ma, “Fast L1-Minimization Algorithms and An Application in Robust Face Recognition: A Review”, Technical Report No. UCB/EECS-2010-13.

[6] M. Cohen and K. K. Paliwal, “Classifying microarray cancer datasets using nearest subspace classification”, PRIB 2008.

[7] L. Wei, S. Prasad, J. E. Fowler, “Nearest Regularized Subspace for Hyperspectral Classification”, IEEE Transactions on Geoscience and Remote Sensing, Vol. 52 (1) pp. 477 – 489, 2013

[8] Y. Chi, Nearest Subspace Classification with Missing Data, Asilomar 2013.

[9] A. Majumdar, "Classification by Linearity Assumption", ICAPR, pp. 255-258, 2009.

[10] Yi Wang, Consistency Analysis of Nearest Subspace Classifier, arXiv:1501.06060.

[11] P. J. Huber, "Robust Estimation of a Location Parameter", The Annals of Mathematical Statistics, Vol. 35 (1), pp. 73-101, 1964.

[12] R. L. Branham Jr., "Alternatives to least squares", Astronomical Journal 87, pp. 928-937, 1982.

[13] M. Shi and M. A. Lukas, "An L1 estimation algorithm with degeneracy and linear constraints". Computational Statistics & Data Analysis, Vol. 39 (1), pp. 35-55, 2002.

[14] L. Wang, M. D. Gordon and J. Zhu, "Regularized Least Absolute Deviations Regression and an Efficient Algorithm for Parameter Tuning". IEEE ICDM. pp. 690-700, 2006.

[15] I. Barrodale and F. D. K. Roberts, "An improved algorithm for discrete L1 linear approximation". SIAM Journal on Numerical Analysis, Vol. 10 (5), pp. 839-848, 1973.

[16] E. J. Schlossmacher, "An Iterative Technique for Absolute Deviations Curve Fitting". Journal of the American Statistical Association, Vol. 68 (344), pp. 857-859, 1973.

[17] G. O. Wesolowsky, "A new descent algorithm for the least absolute value regression problem". Communications in Statistics - Simulation and Computation, Vol. B10 (5), pp. 479-491, 1981.

[18] Y. Li and G. R. Arce, "A Maximum Likelihood Approach to Least Absolute Deviation Regression". EURASIP Journal on Applied Signal Processing, Vol. (12), pp. 1762-1769, 2004.

[19] R. Baraniuk, "Compressive sensing", IEEE Signal Processing Magazine, Vol. 24(4), pp. 118-121, 2007.

[20] E. Candès and M. Wakin, "An introduction to compressive sampling", IEEE Signal Processing Magazine, 25(2), pp. 21 - 30, 2008.

[21] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing", IEEE ICASSP 2008.

[22] G. Hennenfent, E. van den Berg, M. P. Friedlander, and F. Herrmann, "New insights into one-norm solvers from the Pareto curve", Geophysics, Vol. 73(4), pp. A23-A26, 2008.

[23] B. Wohlberg and P. Rodríguez, "An Iteratively Reweighted Norm Algorithm for Minimization of Total Variation Functionals", IEEE Signal Processing Letters, Vol. 14 (12), pp. 948-951, 2007.

[24] T. Goldstein and S. Osher. "The Split Bregman Method for L1 Regularized Problems", SIAM Journal on Imaging Sciences, Vol. 2 (2), pp. 323-343, 2009.

[25] H. Nien and J. A. Fessler "A convergence proof of the split Bregman method for regularized least-squares problems", arXiv:1402.4371

[26] http://cnx.org/contents/c9c730be-10b7-4d19-b1be-22f77682c902@3/Sparse_Signal_Restoration

[27] http://archive.ics.uci.edu/ml/

[28] http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

[29] http://www.iro.umontreal.ca/~lisa/twiki/bin/view.cgi/Public/MnistVariations