Greedy Algorithm for Subspace Clustering from Corrupted and Incomplete Data

Alexander Petukhov, Inna Kozlov

Abstract—We describe the Fast Greedy Sparse Subspace Clustering (FGSSC) algorithm providing an efficient method for clustering data belonging to a few low-dimensional linear or affine subspaces. FGSSC is a modification of the SSC algorithm. The main difference of our algorithm from predecessors is its ability to work with noisy data having a high rate of erasures (missed entries at the known locations) and errors (corrupted entries at unknown locations).

The algorithm has significant advantage over predecessor on synthetic models as well as for the Extended Yale B dataset of facial images. In particular, the face recognition misclassification rate turned out to be 6–20 times lower than for the SSC algorithm.

Keywords—Subspace Clustering, Compressive Sampling, Face Recognition.

I. INTRODUCTION

We consider a greedy strategy based algorithm for preprocessing on vector database necessary for subspace clustering. The problem of subspace clustering consists in classification of the vector data belonging to a few linear or affine low-dimensional subspaces of a high dimensional ambient space when neither subspaces nor even their dimensions are known, i.e., they have to be identified from the same database. No dataset for algorithm learning is provided.

The problem has a long history and always was considered as difficult. In spite of its closeness to the problem of linear regression, the presence of multiple subspaces brings combinatorial non-polynomial complexity. Additional hardness of the settings considered in this paper is due to the presence of combined artifacts consisting of noise in the data, the errors at unknown entries of unknown vectors. In addition, a significant fraction may be missed.

There are many applied problems expecting the progress in related to processing of visual information. Among them the problems related to processing of visual information are especially attractive. Automatic sorting databases consisting of images taken from a finite set of objects like faces, characters, symbols are typical problems. A special class of applied problems is that of motion segmentation in video data which can be used for better motion estimation, video segmentation, and 2D- to 3D-video conversion. The last successes in those areas are connected with the recent Sparse Subspace Clustering algorithm developed in [6], [7] and thoroughly studied in [17], [18]. SSC algorithm has an error correction ability but has no special treating missed data. At the same time, a minor modification extends the area of its application to clustering from incomplete data. In [3] and [8], a different theoretically supported approach is considered. The algorithms designed in [3] and [8] reconstruct missed data but not intended for error processing. In addition, according to the data of numerical modeling available in [3] and [8], the number of the required samples is significantly larger and the total accumulated dimension of all subspaces is much less than SSC is able to afford for successful clustering.

The subspace clustering problem can be formalized as follows. We have $N$ vectors $\{Y_i\}_{i=1}^N$ in $K$ linear or affine subspaces $S := \{S^l\}_{l=1}^L$ of the $D$-dimensional Euclidean space $\mathbb{R}^D$. The dimensions of the subspaces are $\{d_l\}_{l=1}^L$. These subspaces may have non-trivial intersections. However, we do assume that any one of those spaces is not a subspace of other one. At the same time, the situation when one subspace is a subspace of a sum of two or more subspaces from $S$ is allowed. Such settings inspire the hope that when $N$ is significantly large and the points are randomly and independently distributed on the set of planes, some sophisticated algorithm is able to identify those planes (subspaces) and then classify the points by their association with (closeness to) the found subspaces. Then the problem consists in finding a permutation matrix $\Gamma$ such that

$$[Y_1, \ldots, Y_K] = Y \Gamma,$$

where $Y \in \mathbb{R}^{D \times N}$ is an input matrix whose columns are the given randomly permuted points. Whereas in $[Y_1, \ldots, Y_K]$ is the rearrangement of the matrix $Y$ in the accordance with the affiliation of the vectors with the subspaces $S^l$.

As it was discussed in [7], the problem of finding clusters $\{Y_k\}$ corresponding to linear/affine subspaces can be resolved by consecutive solving of the non-convex problem

$$\|e_i\|_0 \rightarrow \min, \text{ subject to } y_i = Y e_i, \; c_{i,l} = 0, \; i = 1, \ldots, N;$$

(1)

where $\|x\|_0$ is the Hamming weight of the vector $x$, $\|x\|_0 := \#\{x_j \neq 0\}$, and follow-up spectral clustering ([11], [9]) of the similarity graph defined by the matrix $W := |C| + |C^T|$, where $C$ is defined by its columns $e_i$.

The problem (1) is tightly associated with the problem of Compressive Sampling ([4], [5], [16]). In particular, it can be replaced with the convex $\ell^1$-problem

$$\|C\|_1 \rightarrow \min, \text{ subject to } Y = Y C, \; \text{diag}(C) = 0,$$

(2)

having the same solution when the columns of $C$ are sparse enough.
Provided that the data matrix $Y$ has a low level of noise, the problem (2) can be tractable as a standard problem of Compressed Sensing. However, when some entries are corrupted or missed the special treatment of such input is necessary.

The algorithm considered below does not require clean input data. We assume that the data may be corrupted simultaneously by three sources of distortion. Random noise is distributed over all vector entries. A quite significant fraction of the data is missing. Some unknown entries of $Y$ may be corrupted with errors. The problem of subspace clustering can be split into 2 stages:

1) preprocessing (the graph matrix $W$ composing);
2) search for clusters in the graphs;

In this paper, we develop a first stage algorithm helping to perform the second stage much more efficiently than the state-of-the-art algorithms. In [15], we designed the Greedy Sparse Subspace Clustering (GSSC) algorithm relying on main principles of SSC algorithm from [7] but having increased clustering capabilities due to implementation of greedy ideas at cost of the higher (about 5 times) computational complexity. Here we present an accelerated version of GSSC which is not slower than SSC. As for its capability to separate subspaces, we will show that it outperforms (sometimes significantly) not only SSC but even GSSC. We call this algorithm the Fast GSSC or FGSSC.

We do not discuss any aspects of improvements of stage 2. We just take one of such algorithms of the spectral clustering presented Sensing. However, when some entries are corrupted or missing, the problem (2) can be tractable as a standard problem of Compressed Sensing. However, when some entries are corrupted or missed the special treatment of such input is necessary.

In Section II, we discuss formal settings of the optimization problem to be solved. In Section III, we introduce the FGSSC algorithm which is the main topic of this paper. The results of numerical experiments showing the consistency of the proposed approach for both synthetic and real world data are given in Section IV.

II. PROBLEM SETTINGS

Following [7], we introduce two (unknown for the problem solver) $D \times N$ matrices $E$ and $Z$. The matrix $E$ contains a sparse (i.e., $\# \{E_{ij} \neq 0\} < DN$) set of errors with relatively large magnitudes. The matrix $Z$ defines the noise having a relatively low magnitude but distributed over all entries of $Z$. Thus, the clean data are representable as $Y - E - Z$. Therefore, when the data are corrupted with sparse errors and noise, the equation $Y = YC$ has to be replaced by

$$Y = YC + E(I - C) + Z(I - C). \quad (3)$$

The authors of [7] applied a reasonable simplification of the problem by replacing two last terms of (3) with some (unknown) sparse matrix $E := E(I - C)$ and the matrix with the distorted noise $Z = Z(I - C)$. Provided that the sparse $C$ exists, the matrix $E$ still has to be sparse. This transformation leads to some simplification of the optimization procedure.

Taking that simplification into account, we are able to formulate the constrained optimization problem

$$\min \|C\|_1 + \lambda_1 \|E\|_1 + \frac{\lambda_2}{2} \|Z\|_F^2,$$

$$\text{s.t. } Y = YC + E + Z, \ diag(C) = 0. \quad (4)$$

where $\| \cdot \|_F$ is the Frobenius matrix norm. If the clustering into affine subspaces is required, the additional constrain $C^T 1 = 1$ is added.

On the next step, using the representation $Z = Y - YC - E$ and introducing an auxiliary matrix $A \in \mathbb{R}^{N \times N}$, constrained optimization problem (4) is transformed into

$$\min \|C\|_1 + \lambda_1 \|E\|_1 + \frac{\lambda_2}{2} \|Y - YA - E\|_F^2$$

$$\text{s.t. } A^T 1 = 1, A = C - \text{diag}(C). \quad (5)$$

At last, the quadratic penalty functions with the weight $\rho/2$ corresponding to the constrained are added to the functional in (5) and the Lagrangian functional is composed. The final Lagrangian functional is as follows

$$L(C, A, E, \delta, \Delta) = \min \|C\|_1 + \lambda_1 \|E\|_1 + \frac{\lambda_2}{2} \|Y - YA - E\|_F^2$$

$$+ \frac{\rho}{2} \|A^T 1 - 1\|_2^2 + \frac{\rho}{2} \|A - C + \text{diag}(C)\|_F^2$$

$$+ \delta^T (A^T 1 - 1) + \Delta (A^T (A - C + \text{diag}(C))),$$

where the vector $\delta$ and the matrix $\Delta$ are Lagrangian multipliers.

The first terms in lines 3 and 4 of (6) have to be removed when only linear subspace clusters are considered.

III. OUR FGSSC ALGORITHM

For finding the stationary point of functional (6) an Alternating Direction Method of Multipliers (ADMM, [1]) is used.

The parameters $\lambda_1$ and $\lambda_2$ in (6) are selected in advance. They define the compromise between the better approximation of $Y$ with $YC$ and the higher sparsity of $C$. The general rule is to set the larger values of the parameters for the less level of the noise or errors. In [7], the selection of parameters by formulas

$$\lambda_1 = \alpha_1/\mu_1, \quad \lambda_2 = \alpha_2/\mu_2, \quad (7)$$

where $\alpha_1$, $\alpha_2 > 1$ and

$$\mu_1 := \min_{i,j \neq i} \|y_j\|_1, \quad \mu_2 := \min_{i,j \neq i} \|y_j^T y_j\|_1,$$

is recommended. The initial parameter $\rho = \rho^0$ is also set in advance. It is updated as $\rho := \rho^{k+1} = \rho^{k} \mu$ for consecutive iterations of SSC algorithm.

We will need the following notation

$$S_{\epsilon}[x] := \left\{ \begin{array}{ll}
  x - \epsilon, & x > \epsilon, \\
  x + \epsilon, & x < -\epsilon, \\
  0, & \text{otherwise};
\end{array} \right.$$

where $x$ can be either a number or a vector or a matrix. The operator $S_{\epsilon}[-]$ is called the shrinkage operator.

In Algorithm 1, we assume that the data (matrix $Y$) is available only at entries with indexes on the set $\Omega = \{1, \ldots, D\} \times \{1, \ldots, N\}$. $\chi(\cdot)$ is a characteristic function of the set $\Omega$. The symbol $\circ$ is used for the entry-wise product of matrices.
Algorithm 1 FGSSC

Input: $Y \in \mathbb{R}^{D \times N}$, $\Omega \in \mathbb{R}^{D \times N}$.

1. Initialization: $C^0 := 0$, $A^0 := 0$, $E^0 := 0$, $\delta^0 := 0$, $\Delta^0 := 0$, $k := 0$, $M = \infty$, STOP:=false, $\epsilon > 0$, $\rho^0 > 0$, $\mu \geq 1$.
2. mxMed=median($\chi_{\Omega} \odot Y$);
3. while STOP=false do
4. \quad if $k = k_0$ then $M := a_0 \|E_{ij}\|_{\infty}$;
5. \quad else
6. \quad \quad if $k$ is even then
7. \quad \quad \quad $M := \max\{a_1M, a_2\text{mxMed}\}$;
8. \quad \quad end if
9. \quad end if
10. $\Omega := \Omega \\setminus \{(i,j) \mid |E_{ij}| > M\}$;
11. update $A^{k+1}$ by solving the system of linear equations
12. \quad $(\lambda Y^T Y + \rho I + \rho^k 11^T)A^{k+1}$
13. \quad \quad $= \lambda Y^T (Y - E^k) + \rho^k(11^T + C^{k}) - \Delta^{k}$.
14. update $E^{k+1} := \chi_{\Omega} \odot S_{\frac{Y}{\alpha}} [Y - YA^{k+1}]$
15. \quad $+ (1 - \chi_{\Omega}) \odot (Y - YA^{k+1})$.
16. update $\delta^{k+1} := \delta^k + \rho^k(A^{k+1} - 1)$;
17. update $\Delta^{k+1} := \Delta^k + \rho^k(A^{k+1} - C^{k+1})$.
18. \quad if $\|A^{k+1} - 1\|_{\infty} < \epsilon$, \& $\|A^{k+1} - C^{k+1}\|_{\infty} < \epsilon$, \& $\|E^{k+1} - E^k\|_{\infty} < \epsilon$ then STOP:=true;
19. \quad if $k$ is even then
20. \quad \quad $Y := Y - \chi_{\Omega} E^k$;
21. \quad \quad $E^{k+1} := 0$;
22. \quad \end if
23. \quad update $\mu_{\chi}(\Omega)$ and $\mu_{\chi}(\Omega)$;
24. end if
25. update $\rho_{\nu} := \rho_{\nu}$
26. update $k := k + 1$
return $C^* := C^k$

Our FGSSC is the result of 3-way modification of the basic SSC algorithm from [7]. The ideas of those modifications have come from our papers [12]-[15].

The original SSC algorithm does not take an advantage of the knowledge of coordinates of erasures (missing data). Our first modification (see [15]) consists in taking this information into account by means of replacing the update formula $E^{k+1} := S_{\frac{Y}{\alpha}} [Y - YA^{k+1}]$ with our version given at Line 13 of Algorithm 1.

The second modification is also due to [15]. It is based on iterating SSC with updates of $\Omega$ based on relocation of the ”reliable” high magnitude errors to the set of erasures (see Line 10 above). This is a ”greedy” step.

At last, the third modification leading to the fast algorithm given above consists in incorporation of the ”greedy” erasure update steps directly into the basic SSC algorithm.

IV. NUMERICAL EXPERIMENTS

In the beginning, we will present the comparison of the FGSSC and SSC algorithms on two types of synthetic data. The third part of this section is devoted to the problem of the face recognition.

A. Synthetic Input I

The input data for the first experiment are composed in accordance with the model given in [7]. 105 data vectors of dimension $D = 50$ are equally split between three 4-dimensional linear spaces $\{S^i\}_{i=1}^3$. To make the problem more complicated each of those 3 spaces belongs to sum of two others. The smallest angles between spaces $S^i$ and $S^j$ are denoted by $\theta_{ij}$. We construct the data sets using vectors generated by decompositions with random coefficients in orthonormal bases $e^i_j$ of the spaces $S^i$. Three vectors $e^i_1$ belong to the same 2D-plane with angles $\theta_{12} = \theta_{13} = \theta_{23} = 20$. The vectors $e^1_1, e^2_1, e^3_1, e^1_2, e^2_2, e^3_2$ are mutually orthogonal and orthogonal to $e^1_3, e^2_3, e^3_3$. The standard normal distribution is used to generate data decomposition coefficients. After the generation, a random unitary matrix is applied to the result to avoid zeros in some regions of the matrix $Y$.

We use the notation $P_{\text{err}}$ for probabilities of erasures and errors correspondingly.

When we generate erasures, we set the value 0 to the erased entries of the matrix $Y$ since no a priori information about those values is known.

The coordinates of samples with errors are generated randomly with probability $P_{\text{err}}$. We use the additive model of errors, adding values of errors to the correct entries of $Y$. The magnitudes of errors are taken from standard normal distribution.

We run 50 trials of FGSSC and SSC algorithms for each combination of $(\theta, P_{\text{err}}, P_{\text{ras}})$, $0 \leq P_{\text{err}} \leq 0.5$, $0 \leq P_{\text{ras}} \leq 0.7$. Then we output average rates of misclassification.

We note that for the angle $\theta = 0$ the spaces $\{S^i\}$ have a common line and $\dim(\bigoplus_{i=1}^3 S^i) = 7$. Nevertheless, we see on Fig. 1 that SSC, and especially FGSSC, shows a high clustering capability even for those hard settings. SSC processing was performed with $\alpha_c$ linearly changing from 5 (no erasures) to 24 ($P_{\text{err}} = 0.7$). $\alpha_z = 7$, $\rho^0 = 10$, $\mu = 1.05$, $\epsilon = 0.001$.

The results for the FGSSC algorithm presented on Fig. 1 were obtained with parameters $\alpha_0 = 0.6$, $\alpha_1 = 0.95$, $\alpha_2 = 1.0$; the parameter $\alpha_c$ linearly changed from 11 for $P_{\text{err}} = 0.0$ to 22 for $P_{\text{err}} = 0.7$; $\alpha_z = 20$. 
All points of the images are obtained as an average value over 50 trials. The matrix Y, the sets of erasures and errors, as well as the values of errors, are randomly drawn, according to the models described above. The intensity of each point takes a range from the "white" level, when there is no errors in the results clustering, to the "black" level, when more than 50% of vectors were misclassified.

We also introduced independent Gaussian noise with the signal-to-noise-ratio 15dB. Our algorithm showed practically same reliability of clustering as for the noise free input.

We emphasize that all results of this section were obtained with the same algorithm settings.

### B. Synthetic Input II

We put our data into the ambient space of the same dimension $D = 50$ as in Section IV-A. Now, we use the model with random selection of subspace orientations. We fix the number of subspaces $K = 7$ while the dimensions of subspaces are selected randomly within the range $3 \div 10$. In particular, those settings mean that the expectation of subspace dimensions is 6.5. Therefore, the total average dimension (the sum of all dimensions) of all subspace is close to the dimension of the entire ambient space. The number of data points is set as $N = 200$. Those points are randomly distributed between planes. The number of points in each subspace does not depend on their dimensions. However, to avoid the shortage of points defining a linear subspace we make the number of those points larger than the dimension of that subspace.

The parameters $\alpha_1, \alpha_2$ has the following dependency on $P_{ers}$: $\alpha_c := \beta_1 + \beta_2 P_{ers}$, $\alpha_s = \gamma_1 + \gamma_2 P_{ers}^2$, where $\beta_i$ and $\gamma_i$ are defined in Table 1. The parameters, as well as the dependencies above, were found empirically. The results of modeling are presented on Fig. 2.

On Fig.3 and Fig.4, we present results of two experiments with the lower dimension of the ambient space.

<table>
<thead>
<tr>
<th>$D = 15$</th>
<th>$D = 25$</th>
<th>$D = 50$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_1$</td>
<td>36</td>
<td>5</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>22</td>
<td>18</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>$\gamma_2$</td>
<td>29</td>
<td>10.8</td>
</tr>
</tbody>
</table>

The results on Fig.3 and Fig.4 are given for the cases $D = 25$ and $D = 15$ when the mean value of sum of dimensions of the subspaces $S^i$ is several times greater than the dimension of the ambient space. Say, $E[\sum_{i=1}^2 \dim S^i] = 7 \cdot 6.5 = 45.5 > 3 \cdot 15 = 45$. Therefore, the sum of the subspace dimensions is more than 3 times greater than $D = 15$. Even for these hard settings FGSSC demonstrates its high performance.

### C. Face Recognition

It was shown in [2] that the set of all Lambertian reflectance functions (the mapping from surface normals to intensities) obtained with arbitrary distant light sources lies close to a 9D linear subspace. One of possible ways to use this result in combination with subspace clustering algorithms is the problem of face images databases clustering.

Provided that face images of multiple subjects are acquired with a fixed pose and varying light conditions, the problem of sorting images according to their subjects is the obvious object for trying the designed FGSSC. To our knowledge the state-of-the-art benchmarks are reached in [7]. Therefore, we organize our numerical experiments according to settings accepted in [7].

We try the algorithms on the Extended Yale B dataset [10]. The images of that dataset have resolution $192 \times 168$ pixels. We downsample those images to the resolution $48 \times 42$ by simple subsampling. Then we normalize their $l^2$-norm. Thus, each image is represented by a vector of dimension $D = 2016$ with the unit norm. The data set consists of images corresponding to 38 individuals. Frontal face images of each individual are acquired under 64 different lighting conditions. We split those images into 4 groups corresponding to 1–10, 11–20, 21–30, 31–38 individuals. We conduct our experiments inside those groups and then collect those results into total estimates for all groups. We estimate the efficiency of our algorithm on all possibles subgroups of $K = 2, 3, 5, 8, 10$ individuals.

While, varying algorithm parameters for different input data (say for different $K$) the results can be significantly improved, we set the fixed parameters in all our trials.

We use the following set of parameters for FGSSC. $\epsilon = 10^{-3}$, $\alpha = 9.7$, $\alpha_2 = 81$, $\alpha_0 = 0.6$, $\alpha_1 = 1.0$, $\rho_0 = 1$, $\mu = 1.02$. 

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon$</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>9.7</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>81</td>
</tr>
<tr>
<td>$\alpha_0$</td>
<td>0.6</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>1.0</td>
</tr>
<tr>
<td>$\rho_0$</td>
<td>1</td>
</tr>
<tr>
<td>$\mu$</td>
<td>1.02</td>
</tr>
</tbody>
</table>
The results of processing are presented in Table 2. The leading 4 columns of Table 2 contain the average percent-

table.

Table II. Misclassification Rate (%) 

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>FGSSC</th>
<th>SSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 subjects</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>FGSSC</td>
<td>SSC</td>
</tr>
<tr>
<td>Mean</td>
<td>0.087</td>
<td>0.071</td>
<td>0.122</td>
<td>0.140</td>
<td>0.098</td>
<td>1.86</td>
</tr>
<tr>
<td>Median</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3 subjects</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>FGSSC</td>
<td>SSC</td>
</tr>
<tr>
<td>Mean</td>
<td>0.234</td>
<td>0.142</td>
<td>0.191</td>
<td>0.995</td>
<td>0.297</td>
<td>3.10</td>
</tr>
<tr>
<td>Median</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.04</td>
</tr>
<tr>
<td>5 subjects</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>FGSSC</td>
<td>SSC</td>
</tr>
<tr>
<td>Mean</td>
<td>0.642</td>
<td>0.248</td>
<td>0.270</td>
<td>4.840</td>
<td>0.694</td>
<td>4.31</td>
</tr>
<tr>
<td>Median</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.31</td>
<td>0.00</td>
<td>2.50</td>
</tr>
<tr>
<td>8 subjects</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>FGSSC</td>
<td>SSC</td>
</tr>
<tr>
<td>Mean</td>
<td>1.280</td>
<td>0.460</td>
<td>0.768</td>
<td>16.2</td>
<td>0.949</td>
<td>5.85</td>
</tr>
<tr>
<td>Median</td>
<td>1.17</td>
<td>0.40</td>
<td>0.39</td>
<td>n/a</td>
<td>0.40</td>
<td>4.49</td>
</tr>
<tr>
<td>10 subjects</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>FGSSC</td>
<td>SSC</td>
</tr>
<tr>
<td>Mean</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>0.84</td>
<td>10.94</td>
</tr>
<tr>
<td>Median</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>0.64</td>
<td>5.63</td>
</tr>
</tbody>
</table>

age rate of misclassification for FGSSC applied on group-wise data. Whereas the 5th column gives the processing results for consolidated data from all groups. The 6th column provides the results for the SSC algorithm reported in [7]. Comparison of the results in columns 5 and 6 of Table 2 shows that FGSSC outperforms SSC in 6–20 times in average misclassification.

V. CONCLUSIONS AND FUTURE WORK

We presented the Fast Greedy Sparse Subspace Clustering algorithm which is a greedy approach-based modification of the SSC algorithm. FGSSC has significantly increased resilience to the corruption of entries on sparse set, to the data incompleteness, and to noise. On the real database of face images it provides 6–20 times lower rate of misclassification in face recognition than the SSC algorithm. It also significantly outperforms SSC on a few presented models of synthetic data.

REFERENCES


