Matrix Reconstruction: Skeleton Decomposition versus Singular Value Decomposition

Ali Sekmen*, Akram Aldroubi†, Ahmet Bugra Koku*‡, Keaton Hamm†

*Department of Computer Science, Tennessee State University

†Department of Mathematics, Vanderbilt University

‡Department of Mechanical Engineering, Middle East Technical University

***Abstract*—In this work, Skeleton Decomposition (SD) and Singular Value Decomposition (SVD) are compared and evaluated for reconstruction of data matrices whose columns come from a union of subspaces. Specifically, an original data matrix is reconstructed from noise-contaminated version of it. First, matrix reconstruction using SD iteratively is introduced and alternative methods for forming SD-based reconstruction are discussed. Then, through exhaustive simulations, effects of process parameters such as noise level, data size, number of subspaces and their dimensions are evaluated for reconstruction performance. It is also shown that SD-based reconstruction is more effective when data is drawn from a union of low dimensional subspaces compared to a single space of the same dimension.**

***Index Terms*—Skeleton decomposition, SVD, matrix reconstruction, low-tail-noise, high-tail-noise**

I. Introduction

In this paper, it is assumed that the columns of the data matrix $\mathbf{W} = [w_1 \cdots w_N] \in \mathbb{R}^{D\times N}$ are drawn from a union $\mathscr{U} = \bigcup_{i=1}^{M} S_i$ of linearly independent subspaces $\{S_i\}_{i=1}^{M}$ of dimensions $\{d_i\}_{i=1}^{M}$. In many real world applications, data lives in a union of low dimensional subspaces [1], [2], [3], [4], [5]. For instance, the set of all two dimensional images of a given face i, obtained under different illuminations and facial positions, can be modeled as a set of vectors belonging to a low dimensional subspace S_i living in a higher dimensional space $\mathbb{R}^D$ [4], [6], [7]. It is further assumed that $\mathbf{W}$ is contaminated with noise $\mathbf{N} \in \mathbb{R}^{D\times N}$ whose entries are i.i.d $\mathscr{N}(0,\sigma^2)$ random variables (i.e. zero-mean with variance σ^2). Given $\mathbf{W}_n = \mathbf{W} + \mathbf{N}$, SD and SVD based techniques are evaluated and compared for reconstruction of $\mathbf{W}$. Performance of each method is assessed based on the Frobenius norm of the estimation error matrix (i.e. $\|\mathbf{W} - \widetilde{\mathbf{W}}\|_F$), where $\widetilde{\mathbf{W}}$ is the reconstructed matrix. It is assumed that the rank of the original data matrix is available.

II. Skeleton Decomposition

The skeleton decomposition (also known as CUR factorization [8]) of a matrix $\mathbf{W} \in \mathbb{R}^{D\times N}$ can be written as:

$$\mathbf{W} = CA^{-1}R \tag{II.1}$$

where $C \in \mathbb{R}^{D\times r}$, $A \in \mathbb{R}^{r\times r}$, and $R \in \mathbb{R}^{r\times N}$ with $r = rank(\mathbf{W})$ are formed by any r columns and rows selected from $\mathbf{W}$ such that they are full rank, and A is the matrix that is formed by intersecting C and R. The construction of C, A, R is as in Algorithm 1. Fig. 1 illustrates a sample case where row and column indices are selected as $\{2,4,5\}$ and $\{2,5,7\}$, respectively.

Algorithm 1: Matrix Skeleton Decomposition

Data: A data matrix $\mathbf{W} = [w_1 \cdots w_N] \in \mathbb{R}^{D\times N}$.
Result: Skeleton Decomposition of $\mathbf{W}$

```
r = rank(W)
do
    Pick two index vectors v_row and v_col of size r from
    {1, ..., N} (randomly or systematically)
    Construct A such that A(i, j) = W(v_row(i), v_col(j)),
    where i and j go from 1 to r
while rank(A) ≠ r;
Construct a matrix C such that C(i, j) = W(i, v_col(j))
with j ∈ {1, ..., r} and i ∈ {1, ..., D}
Construct a matrix R such that R(i, j) = W(v_row(i), j)
with i ∈ {1, ..., r} and j ∈ {1, ..., N}
```

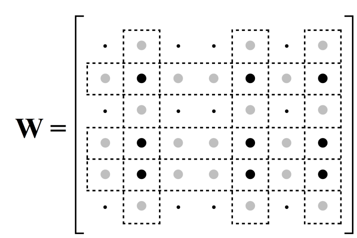


Fig. 1: Demonstration of skeleton decomposition of a matrix.

In general, there are numerous alternatives for choosing a sub-matrix A [9], [10], [11], [12], [13]. In case there is no noise, selecting A (i.e. the row and column indices) is not much of a concern except for numerical stability. However, in the presence of noise, the choice of A heavily affects the performance of the reconstruction of the original data matrix.

III. Matrix Reconstruction

This section explains SD-based, SVD-based, and hybrid SD-SVD methods to reconstruct $\mathbf{W}$ using the noisy data matrix $\mathbf{W}_n$. SVD-based reconstruction is a single step process whereas SD-based reconstruction is an iterative process as explained below.

A. SVD-Based Reconstruction

It is assumed that the original data matrix $\mathbf{W}$ is noise free and has rank r. However, due to presence of noise, $\mathbf{W}_n$ should be full rank (i.e. $rank(\mathbf{W}_n) = min(D,N)$). SVD-based estimation of the original data matrix $\mathbf{W}$ is a single step calculation using the first r singular values of $\mathbf{W}_n$:

1) Find the SVD of noisy data matrix $\mathbf{W}_n = U\Sigma V^T$
2) Estimate of the data matrix is calculated by using the first r singular values found in step 1 as:

$$\widetilde{\mathbf{W}}_{svd} = \hat{U}\hat{\Sigma}\hat{V}^T \tag{III.1}$$

where $\hat{U} \in \mathbb{R}^{D\times r}$, $\hat{\Sigma} \in \mathbb{R}^{r\times r}$, and $\hat{V} \in \mathbb{R}^{N\times r}$.

In the presence of light noise, SVD based reconstruction works well, yet as noise contamination becomes more severe this method falls short of the following two methods.

B. SD-Based Reconstruction

Skeleton decomposition based estimation is similar to the SVD based estimation in that a rank-r skeleton decomposition of $\mathbf{W}_n$ is used to reconstruct the original data matrix (as in Eq. II.1). However, unlike SVD, there is not a unique way of forming this decomposition; hence, every decomposition yields a different reconstruction of the original matrix. In this work, numerous such reconstructions are found and their average is used as the estimation of the original matrix. The procedure is as in Algorithm 2.

Algorithm 2: Matrix Skeleton Decomposition

Data: A noisy data matrix $\mathbf{W}_n \in \mathbb{R}^{D\times N}$.
Result: $\widetilde{\mathbf{W}}_{sd}$: Average of m SD reconstructions of $\mathbf{W}$

$r = \text{rank}(\mathbf{W})$
for $i = 1$ *to* m **do**
  do
    Select a random $A \in \mathbb{R}^{r\times r}$ submatrix from $\mathbf{W}_n$
  while $rank(A) \neq r$ *and* A *is not good*;
  Construct C and R as in Algorithm 1
  Calculate the skeleton decomposition based estimate of original data matrix: $\widetilde{\mathbf{W}}^i_{sd} = C\,A^{-1}R$
end
Calculate $\widetilde{\mathbf{W}}_{sd} = \frac{\sum_{i=1}^{m}\widetilde{\mathbf{W}}^i_{sd}}{m}$

C. SD-SVD-Based Reconstruction

Both SVD-based and SD-based methods yield reconstructions of the original data matrix using a noisy version of it. Yet, both methods yield different matrices. Therefore, at this stage it is quite natural to ask what happens if the estimation obtained by the SD-based method is fed to the SVD-based method for one more reconstruction. It is seen that feeding the output of the SD-based method (i.e. $\widetilde{\mathbf{W}}_{sd}$) to SVD-based method as input, improves the performance of the SD-based method even further.

D. Assessment

Reconstruction performance of each method is calculated as a scalar using the Frobenius norm of the estimation error as follows:

$$e = \frac{\left\|\mathbf{W} - \widetilde{\mathbf{W}}\right\|_F}{\left\|\mathbf{W} - \mathbf{W}_n\right\|_F} \tag{III.2}$$

IV. Numerical Studies

To investigate the effect of different process parameters, various experiments are conducted using synthetic data. An experiment involves the calculation of one or more estimations of the original data matrix using a set of fixed process parameters. Each set of estimations of the original data matrix calculated using the same parameter set is referred to as an ***experiment*** throughout the text. At the end of each experiment, error statistics are calculated for each of the methods being compared. Evidently, even if the process parameters are same, if the original data matrix is changed, this becomes a new experiment. To get statistically meaningful results about the effect of process parameters, numerous experiments with the same process parameters are conducted.

Before moving on with the experiments, some nomenclature is presented followed by implementation details.

A. Nomenclature

In this work, experiments are conducted using synthetic data. Data comes from a single subspace or a union of subspaces. Parameters used in the generation of test data are described below:

- D : The ambient space dimension
- n : The set of dimensions of subspaces from which data comes
- N : The set of the number of data coming from each subspace
- σ : Standard deviation of noise added to the noise-free data matrix
- *n_rep* : Number of estimations calculated for an experiment
- *n_exp* : Number of different experiments executed

Throughout the paper, these parameters as a set will be referred to as ***process parameters***.

A derived term, the so called *occupancy factor* is defined as follows:

$$\kappa = min\left(\frac{\sum_{i=1}^{|n|} n_i}{D}, 1\right) \tag{IV.1}$$

where n is the set containing subspace dimensions, and D is the dimension of the ambient space. Observe that as the sum of subspace dimensions equals or surpasses the ambient space dimension D, the occupancy parameter κ saturates at 1.

The effect of occupancy factor κ is investigated through experiments as well. However, since it is a derived parameter, it is not mentioned among the process parameters.

B. Number of Iterations in SD-Based Methods

The main contribution of this work is the use of SD obtained from a noisy data matrix in order to reconstruct an estimate to the non-noisy original data matrix. This involves numerous reconstructions being averaged as explained in Section III-B. However, the first question to be answered is: How many reconstructions should be averaged? In our implementations a conservative high number of iterations are set as the maximum number (i.e. $10,000$). Iterations are terminated either when the maximum number of iterations is reached, or when the last n error terms (where $n = 100$) have a standard deviation less then a certain threshold (here the threshold is selected as 0.05). Note that this approach is for assessment purposes only and is not usable in practice since the original data matrix is not known and hence actual error cannot be calculated. In other words, this error based method is preferred here in order to have a realistic performance assessment of different process parameters.

C. Choosing **Good** Sub-Matrices

The skeleton decomposition method introduced in this section uses the noisy data matrix to extract numerous sub-matrices (that are referred to as matrix A). However, any chosen sub-matrix A is not necessarily going to be helpful in the estimation process. Two different approaches are tested in order to select better sub-matrices. The first method is based on the condition number of the chosen sub-matrix A. A threshold is empirically selected (as 50), and any sub-matrix A with a condition number larger than this threshold is not used in the estimation process.

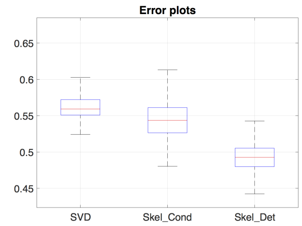
The second method is based on the determinant of the sub-matrix A. Again a threshold is used, and this time sub-matrices that have a determinant smaller than this threshold value are discarded from the process [10]. The determinant threshold is chosen as the average of the determinants of 100 randomly chosen sub-matrices.

In order to decide on which method for selecting sub-matrices should be used, a set of experiments are conducted with the following parameter values:

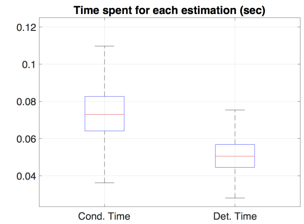
- D = 20
- n = $\{2,2\}$
- N = $\{40, 40\}$
- σ = 0.25
- n_rep = 25
- n_exp = 100

In other words, 100 different data matrices are randomly generated using the same D, N, n and σ values. And for each data matrix, 25 different estimations are found. Results as shown in Fig. 2 suggest that for this particular case, where the ambient space is not highly occupied ($\kappa = 0.2$), using a determinant based threshold seems to perform better.

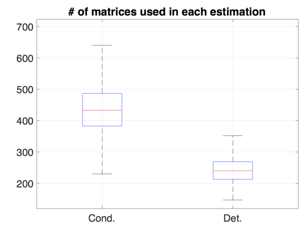
Fig. 2a illustrates the average error of 100 experiments for 3 different methods: (1) SVD-based reconstruction, (2) SD-based reconstruction selecting A based on condition number, (3) SD-based reconstruction selecting A based on determinant.



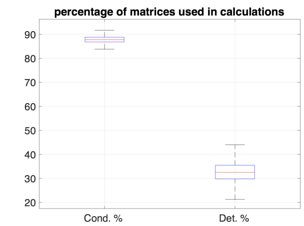
(a) Estimation error



(b) Time spent for each experiment



(c) Number of matrices used in reconstructions



(d) Percentage of matrices used

Fig. 2: Comparison of condition number and determinant based threshold in choosing sub-matrix A where $\kappa = 0.2$

It is seen that use of determinant results in a better approximation under similar termination conditions.

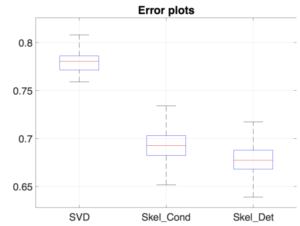
Fig. 2b shows that the determinant based method does not only yield less error, but is also computationally fast for this case. This is not a surprise since sub-matrices are very small in size in comparison to the data matrix. Additionally, Fig. 2c somewhat explains the fast computation since the determinant based method uses roughly half the number of matrices to compute the estimate. Fig. 2d on the other hand depicts a fact to be aware of: the determinant based method discards almost 70% of all the sub-matrices randomly found whereas the condition number based method uses most of them (i.e. more than 85%). Therefore, it should be expected that, as the occupancy factor increases, this inefficiency of the determinant based method will be reflected in execution times.

Therefore, it is worth checking what happens when occupancy increases. It is observed that as occupancy is increased, the condition number based method catches up with the determinant based method, and eventually performs even better. In the next sets of experiments, subspace dimensions are increased from $n = \{4,5\}$ to $n = \{9,9\}$, and occupancy values become $\kappa = 0.45$ and $\kappa = 0.9$, respectively. All other parameters are kept the same. Results are illustrated in Figures 3 and 4.

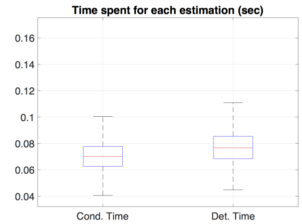
As shown in Figs. 2a, 3a and 4a, performance of condition number based method gradually catches up and gets even marginally better.

Figs. 2b, 3b and 4b, suggest that as occupancy increases, both approaches perform similar in time.

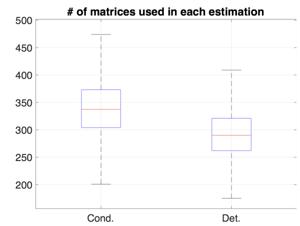
As is seen in Figs. 2c, 3c and 4c, as occupancy increases, the determinant based method uses more matrices to compute an estimation, whereas Figs. 2d, 3d and 4d suggest that as



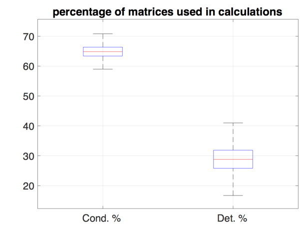
(a) Estimation error



(b) Time spent for each experiment

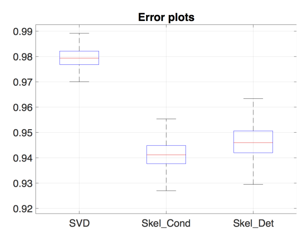


(c) Number of matrices used in reconstructions

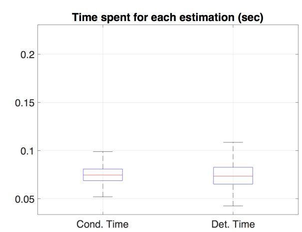


(d) Percentage of matrices used

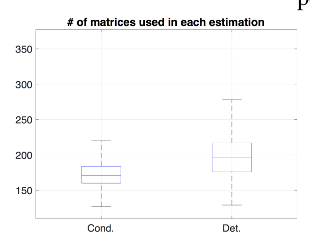
Fig. 3: Comparison of condition number or determinant based threshold in choosing sub-matrix A where $\kappa = 0.45$



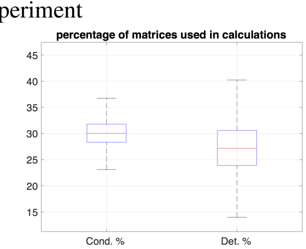
(a) Estimation error



(b) Time spent for each experiment



(c) Number of matrices used in reconstructions



(d) Percentage of matrices used

Fig. 4: Comparison of condition number or determinant based threshold in choosing sub-matrix A where $\kappa = 0.9$

occupancy increases, efficiency of the condition number based method gradually drops down to the level of the determinant based method.

Figs. 2a, 3a and 4a illustrate the performance of SVD as well. However, SVD-based comparison will be presented in the experiments that are to follow.

Finally, it is observed that changing the level of noise does not change the relative performance of the two methods. It is also noted that, as long as the structure of the data is not changed (i.e. ambient space dimension, number of subspaces and their dimensions), changing the number of data in subspaces directly affects computation time but does not favor one method over the other.

As a result, we can conclude that, for cases of low occupancy, the determinant based matrix selection method should be preferable over condition number based selection. Given that most of the subspace segmentation problems present data matrices with low occupation, SD-based reconstruction in subspace segmentation problems should be done using determinant based filtering.

V. Experiments

Experiments presented in this section compare the performance of SVD-based versus SD-based data matrix reconstruction performances based on the Frobenious norm of the estimation matrix. Given the fact that most subspace clustering problems are not highly occupied and using the results presented in Section IV, determinant based filtering is adopted in the calculation of SD-based reconstructions.

In these experiments, effects of the following parameters are investigated systematically:

- Level of noise present on data
- Occupancy of ambient space
- Nature of occupancy
- Number of data in subspace(s)
- Noise type
- Rank estimation

It should also be noted that data is randomly generated in each subspace separately with uniform probability. To be more specific, first an orthonormal basis for a subspace is found, and coordinates that are randomly generated in $[-1,1]$ with uniform probability are projected to this subspace.

The first parameter that is investigated is the level of noise. For the same data matrix, level of added noise $\sim \mathcal{N}(0,\sigma^2)$ is gradually increased, and estimation performance of SVD-based method is observed to fall short compared to SD-based methods. The condition number selection threshold is also studied for the random sub-matrix A of skeleton decomposition. Estimation performance is evaluated based on the scalar value computed as given in Eq. III.2.

The second parameter that is investigated relates to how much the subspace is occupied. Experiments suggest that, as the occupancy parameter κ gets closer to 1, performance of all estimation methods become similar, and when the occupancy parameter is 1, all methods yield exactly the same result.

Even though it is shown that occupancy is a parameter that affects the relative performance of the methods in question, the nature of the occupancy is also found to impact the results. In particular, since data with the same occupancy can come from different number of subspaces with different dimensions, performance is studied when the occupancy factor κ is the same, but the number of subspaces and their dimensions varies.

Experiments are also designed to understand if the number of data points in each subspace causes any performance change among these algorithms. It is observed that SVD-based methods performs better than SD-based methods as the number of data increases. Further experiments are conducted to verify this observation.

The first set of experiments solely focus on the case of additive Gaussian noise as prescribed above, and results are compared for various σ values. In order to see what happens for other noise types, where noise has relatively lighter or heavier tail with respect to the Gaussian noise, an additional set of experiments are conducted.

To conclude our analysis, validity of one of our assumptions, i.e. we can accurately estimate the rank of the original data matrix, is tested. The last set of experiments studies the performance of SVD-based and SD-based methods under poor rank estimation.

A. Effect of Noise Level

This section focuses on how two proposed methods compare with the SVD-based method under various levels of noise. An analogous case to the motion segmentation problem [14], [15] is constructed, where 5 different features on 3 different rigid bodies are followed for 150 frames. This is roughly a 5 sec. video at 30 frames per second. For this motion segmentation scenario (Experiment-1), process parameters are set as follows:

- $D = 300$
- $n = \{4, 4, 4\}$
- $N = \{5, 5, 5\}$
- $\sigma = \{.010, .015, .020, .025, .030, .035, .040, .045, .050\}$
- $n\_rep = 20$
- $n\_exp = 100$

where the test parameter is the noise level σ. Results are presented in Fig. 5. It is seen that as noise level increases, SD-based reconstruction yields less error, and as expected, SD-SVD outperforms SD.

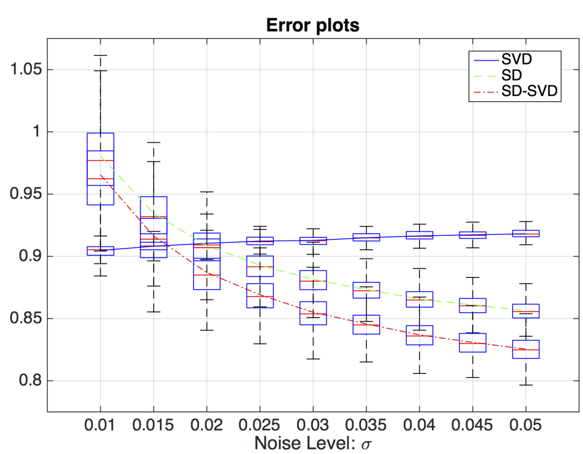


Fig. 5: SVD, SD, and SD-SVD results for Experiment-1

Another set of experiments (Experiment-2) with different process parameters are executed as follows:

- $D = 50$
- $n = \{2, 4, 6\}$
- $N = \{5, 10, 15\}$
- $\sigma = \{.025, .050, .080, .120, .200, .300\}$
- $n\_rep = 20$
- $n\_exp = 100$

Results are shown in Fig. 6.

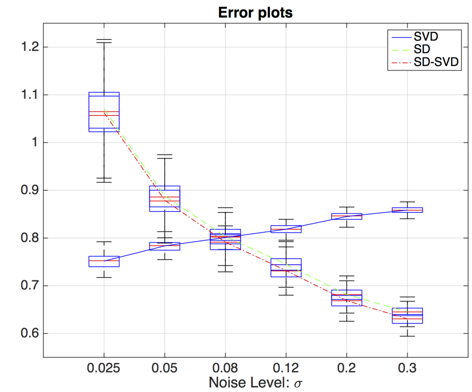


Fig. 6: SVD, SD, and SD-SVD results for Experiment-2

It is seen that as noise level increases, the SD-based methods perform better. In other words, as the noise grows, SVD-based reconstruction performance degrades. The noise level over which the SD-based method performs better depends on the process parameters; however, extensive simulations point to the fact that, above a certain noise level, the SD-based method surpasses the performance of SVD-based method. It should also be noted that, for very low noise levels, SD-based methods yield errors that are larger than 1.0 which suggests that SD-based methods are not suitable for these cases.

B. Effect of Occupancy

Occupancy factor turns out to affect the relative performances of SVD-based and SD-based methods. As occupancy factor κ increases, both methods start to behave similar in terms of their estimation performance. When $\kappa = 1.0$, both methods yield exactly the same result independent of the remaining process parameters. This is simply due to the fact that both methods are reconstructing the noisy data matrix exactly. Experiments with the following process parameters are carried out to study how these methods behave as occupancy gradually increases.

- $D = 35$
- $n = \{\{2, 2\}, \{4, 4\}, \{8, 8\}, \{15, 15\}, \{18, 17\}\}$
- $N = \{50, 50\}$
- $\sigma = 0.100$
- $n\_rep = 20$
- $n\_exp = 100$

The corresponding occupancy values are 0.11, 0.22, 0.46, 0.86 and 1.00 respectively for each value of n. Fig. 7 shows the simulation results.

It is observed that at lower occupancy values SD-based methods perform better than the SVD-based method for this process, and as a matter of fact for many other sets of process parameters that are not included in this text. It is possible to find a case where the SVD-based method performs better at lower occupancies, yet the trend is the same as shown in Fig. 7, i.e. performance of all methods gradually becomes similar as the level of occupancy increases.

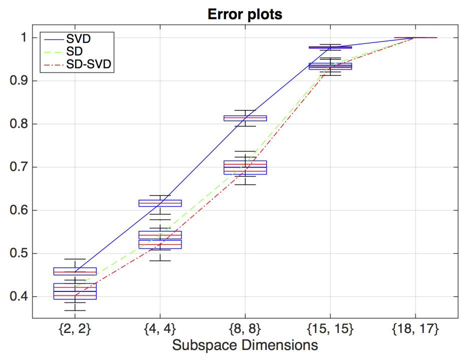


Fig. 7: Effect of occupancy levels: $[0.11 - 1.00]$

C. Effect of Occupancy Structure

After understanding the effect of occupancy, a natural question that comes to mind is whether the structure of occupancy matters. Observe that a particular occupancy value can be obtained as a result of various subspace configurations. For example, every other process parameter being the same, two cases with $n = \{2,2,2\}$ and $n = \{3,3\}$ yield the same occupancy factor. To further understand the effects of occupancy structure, a set of experiments with 5 different cases are conducted where the process parameters are as follows:

- D = 35
- σ = 0.150
- n_rep = 20
- n_exp = 100

Values of n and N are set as follows for different cases:

- Case 1:
  $n = \{1,1,1,1,1,1,1,1\}$,
  $N = \{10,10,10,10,10,10,10,10\}$
- Case 2:
  $n = \{2,2,2,2\}$,
  $N = \{20,20,20,20\}$
- Case 3:
  $n = \{2,3,3\}$,
  $N = \{20,30,30\}$
- Case 4:
  $n = \{4,4\}$,
  $N = \{40,40\}$
- Case 5:
  $n = \{8\}$,
  $N = \{80\}$

Observe that the number of subspaces are gradually decreased while their dimension is increasing to keep the occupancy factor constant. Also note that the number of data points is kept constant in these experiments.

Results shown in Fig. 8 suggest that, for cases where the data comes from the union of a higher number of lower dimensional subspaces, SD-based methods will perform better. Many other tests, not included here, suggest the same conclusion (with similar plots) that SD-based methods should be preferred if data comes from the union of many low-dimensional subspaces.

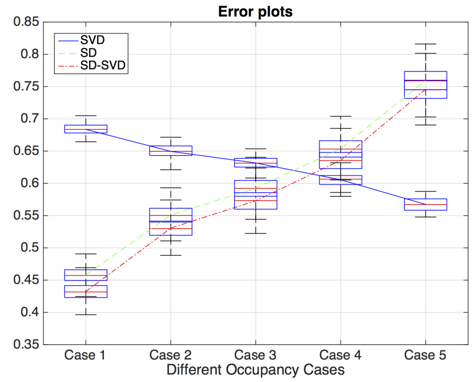


Fig. 8: Effect of occupancy structure

D. Effect of Number of Data Points

The final process parameter studied is the number of data points. In a set of experiments, while keeping all other parameters constant, the number of data points is increased. Note that the occupancy factor is still constant in this analysis.

Process parameters are set as follows:

- D = 50
- $n = \{1,2,4\}$
- N = { {10, 10, 10}, {20, 20, 20}, {40, 40, 40}, {100, 100, 100} }
- σ = 0.150
- n_rep = 20
- n_exp = 100

where the number of data in the three subspaces is gradually increased from 10 to 100.

Results shown in Fig. 9 are representative of the results from many other test runs with completely different process parameters. As suggested by this figure, increase in the number of data points favor SVD-based reconstruction over SD-based methods at some point. What changes from one parameter set to another is this point beyond which SD-based methods exhibit subpar performance.

E. Effect of Noise Type

In all of the previous experiments, noise was zero mean Gaussian with various σ values. The following probability distribution function (PDF) is used to obtain PDFs with different characteristics by changing a parameter ϕ:

$$f(x|\sigma,\phi) = \frac{1}{K}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{|x|^{\phi}}{2\sigma^2}}. \tag{V.1}$$

Observe that the zero-mean Gaussian distribution is obtained when $\phi = 2$ (i.e. $\mathcal{N}(0,\sigma^2)$). When $\phi > 2$ the PDF becomes lighter tailed and when $\phi < 2$, the corresponding PDF becomes

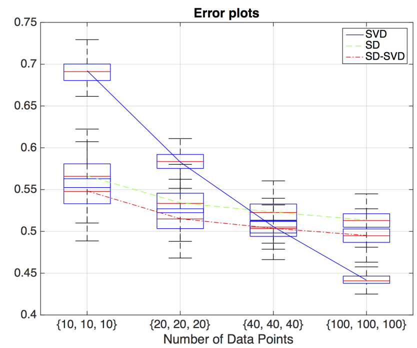


Fig. 9: Effect of number of data points

heavier tailed with respect to the normal distribution as illustrated in Fig. 10. Here, K is a normalizing term so that as ϕ changes, $\int_{-\infty}^{\infty} f(x|\sigma,\phi)$ remains to be 1.

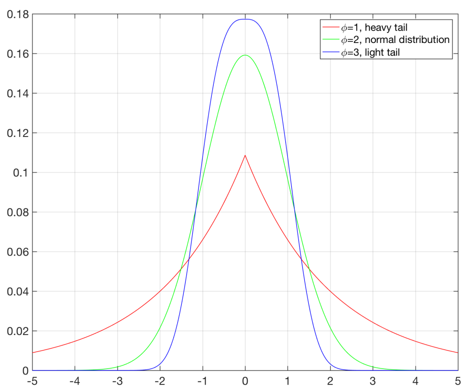


Fig. 10: 3 different PDFs for $\phi = \{1,2,3\}$ and $\sigma = 1$

A set of experiments are conducted where noise is generated using distributions with the following ϕ values: $\{4,3,2,1,0.75,0.5,0.25,0.10\}$. Results obtained using the following process parameters are shown in Fig. 11.

- D = 200
- n = $\{4,4\}$
- N = $\{6,6\}$
- σ = 0.050
- n_rep = 1
- n_exp = 100

It is seen that noise type definitely is a discriminative factor among SVD-based vs SD-based methods. For Gaussian or lighter tailed noise, SD-based method have a definitive advantage over SVD-based method, whereas, for heavier tailed noise, SVD-based method has marginal advantage over SD-based method.

F. Effect of Rank Estimation

Finally, our assumption that rank is properly estimated is tested. This section focuses on what happens when rank is poorly estimated. Many tests are run and most of the time

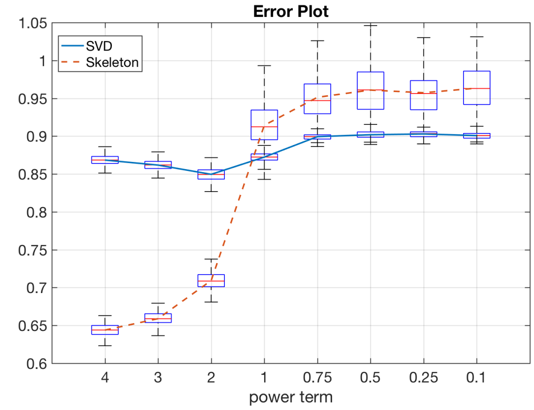


Fig. 11: Effect of noise type of a heavier tail

results are as expected, that is, reconstruction error is minimal when the rank is properly estimated. However, for certain conditions, wrong rank estimation has different impacts.

Consider the following process parameters, with relatively low levels of noise and small number of data in each subspace:

- D = 50
- n = $\{2,4,6\}$
- N = $\{5,10,15\}$
- σ = 0.020
- n_rep = 20
- n_exp = 100

where, rank estimation varied as $\{2,6,11,12,13,18,20,22,25,30\}$. Given that the correct data rank is 12, Fig. 12 illustrates that at and around 12, the algorithms perform best. It should be observed that in this case, estimating rank lower yields error values above 1, which means the original noisy matrix is better than the reconstruction, hence reconstruction does not help at all.

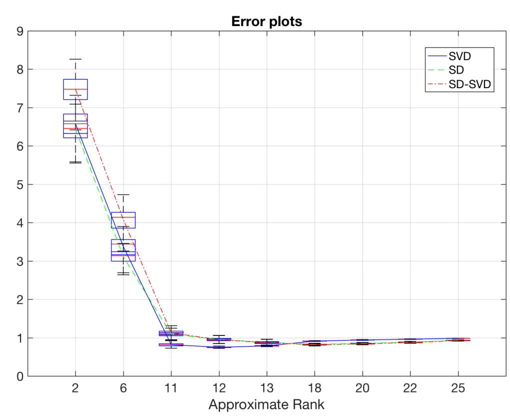


Fig. 12: Effect of rank approximation: Low noise, small number of data points

It should also be observed that as rank is estimated higher than that of the actual rank, error gradually increases to 1. In other words, as rank is over estimated, reconstruction starts to yield the original noisy matrix, defeating the purpose of reconstruction.

As the noise level is increased as shown in Fig. 13, where $\sigma = 0.200$, it is seen that, estimating the rank lower than the actual value indeed improves performance or reconstruction for SD-based methods. It should be noted that best reconstruction performance coincides with the dimension of the largest subspace. Increasing number of data points present in subspaces (10 folds in this case), results shown Fig. 14 suggest more intuitive results where reconstruction is best around the actual rank.

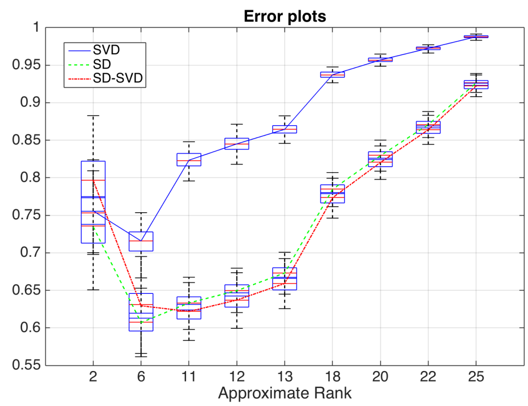


Fig. 13: Effect of rank approximation: High noise, small number of data points

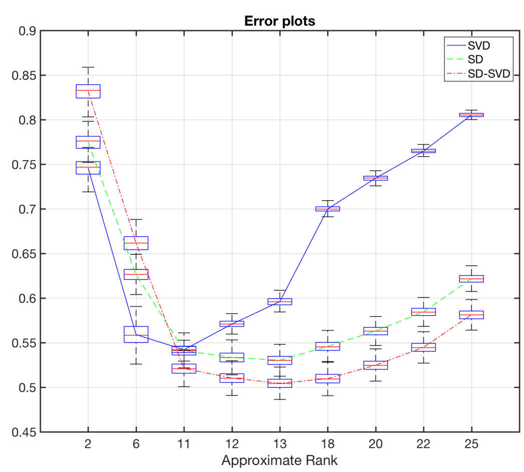


Fig. 14: Effect of rank approximation: High noise, higher number of data points

Unsurprisingly, proper rank estimation is important. It can be concluded that, for low levels of noise with only small number of data, over-estimation is safer, but reconstruction only provides very marginal advantage, whereas for high levels of noise with small number of data, a rank estimation between the dimension of the largest subspace and the actual rank yields better results. Even though only one instance with specific process parameters are presented here, many experiments not documented in this paper suggested similar results.

VI. Conclusion

This study shows that SD-based matrix reconstruction may perform better as data gets corrupted with higher level of noise. Additionally, as the occupancy factor gets lower, SD-based methods outperforms the SVD-based method in the presence of lower level of noise. However, as this factor gets higher, SVD-based and SD-based methods perform comparably. A major conclusion of this work is that SD-based reconstruction outperforms SVD-based reconstruction when data comes from a union of low dimensional subspaces. In other words, in subspace clustering problems, it is expected that SD-based denoising would be more effective than SVD-based denoising.

Acknowledgement

The research of A. Sekmen, and A. Koku is supported by DoD Grant W911NF-15-1-0495. The research of A. Aldroubi is supported by NSF Grant NSF/DMS 132099. The research of A. Koku is also supported by TUBITAK-2219-1059B191600150.

References

[1] Y. M. Lu, M. N. Do, A theory for sampling signals from a union of subspaces, IEEE Transactions on Signal Processing 56 (6) (2008) 2334–2345.

[2] K. Kanatani, Y. Sugaya, Multi-stage optimization for multi-body motion segmentation, in: IEICE Trans. Inf. and Syst, 2003, pp. 335–349.

[3] A. Aldroubi, K. Zaringhalam, Nonlinear least squares in $\mathbb{R}^n$, Acta Applicandae Mathematicae 107 (1-3) (2009) 325–337.

[4] R. Vidal, Y. Ma, S. Sastry, Generalized Principal Component Analysis, unpublished, 2006.

[5] G. Chen, A. V. Little, M. Maggioni, L. Rosasco, Some recent advances in multiscale geometric analysis of point clouds, in: J. Cohen, A. I. Zayed (Eds.), Wavelets and Multiscale Analysis, Applied and Numerical Harmonic Analysis, Birkhuser Boston, 2011, pp. 199–225.

[6] R. Basri, D. W. Jacobs, Lambertian reflectance and linear subspaces, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003) 218–233.

[7] J. Ho, M. Yang, J. Lim, D. Kriegman, Clustering appearances of objects under varying illumination conditions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 11–18.

[8] N. K. Kumar, J. Shneider, Literature survey on low rank approximation of matrices, CoRR abs/1606.06511.
URL `http://arxiv.org/abs/1606.06511`

[9] S. A. Goreinov, E. E. Tyrtyshnikov, N. L. Zamarashkin, A theory of pseudo–skeleton approximations, Linear Algebra Appl. 261 (1997) 1–21.

[10] S. A. Goreinov, N. L. Zamarashkin, E. E. Tyrtyshnikov, Pseudo–skeleton approximations by matrices of maximum volume, Mathematical Notes 62 (4) (1997) 515–519.

[11] M. W. Mahoney, P. Drineas, Cur matrix decompositions for improved data analysis, Proceedings of the National Academy of Sciences 106 (3) (2009) 697–702.

[12] V. Guruswami, A. K. Sinop, Optimal column-based low-rank matrix reconstruction, CoRR abs/1104.1732.
URL `http://arxiv.org/abs/1104.1732`

[13] P. Drineas, R. Kannan, Pass efficient algorithms for approximating large matrices, in: In Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, 2003, pp. 223–232.

[14] K. Kanatani, C. Matsunaga, Estimating the number of independent motions for multibody motion segmentation, in: 5th Asian Conference on Computer Vision, 2002, pp. 7–9.

[15] J. Yan, M. Pollefeys, A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and nondegenerate, in: 9th European Conference on Computer Vision, 2006, pp. 94–106.