CUR Decompositions, Similarity Matrices, and Subspace Clustering

Akram Aldroubi, Keaton Hamm, Ahmet Bugra Koku, and Ali Sekmen

Abstract

A general framework for solving the subspace clustering problem using the CUR decomposition is presented. The CUR decomposition provides a natural way to construct similarity matrices for data that come from a union of unknown subspaces $\mathscr{U} = \bigcup_{i=1}^{M} S_i$. The similarity matrices thus constructed give the exact clustering in the noise-free case. A simple adaptation of the technique also allows clustering of noisy data. Two known methods for subspace clustering can be derived from the CUR technique. Experiments on synthetic and real data are presented to test the method.

Index Terms

Subspace clustering, similarity matrix, CUR decomposition, union of subspaces, data clustering, skeleton decomposition, motion segmentation.

I. Introduction

We present here two tales: one about the so-called CUR decomposition (or sometimes skeleton decomposition), and another about the subspace clustering problem. It turns out that there is a strong connection between the two subjects in that the CUR decomposition provides a general framework for the similarity matrix methods used to solve the subspace clustering problem, while also giving a natural link between these methods and other minimization problems related to subspace clustering.

The CUR decomposition is remarkable in its simplicity as well as its beauty: one can decompose a given matrix A into the product of three matrices, $A = CU^{\dagger}R$, where C is a subset of columns of A and R is a subset of rows of A (see Theorem 1 for a precise statement). The primary uses of the CUR decomposition to date are in the field of scientific computing. In particular, it has been used as a low-rank approximation method that is more faithful to the data structure than other factorizations [1], [2], an approximation to the singular value decomposition [3], [4], [5], and also has provided efficient algorithms to compute with and store massive matrices in memory. In the sequel, it will be shown that this

decomposition is the source of some well-known methods for solving the subspace clustering problem, while also adding to the construction of many similarity matrices based on the data.

The subspace clustering problem may be stated as follows: suppose that some collected data vectors in $\mathbb{K}^m$ (with m large, and $\mathbb{K}$ being either $\mathbb{R}$ or $\mathbb{C}$) comes from a union of linear subspaces (often low-dimensional) of $\mathbb{K}^m$, which will be denoted by $\mathscr{U} = \bigcup_{i=1}^{M} S_i$. However, one does not know *a priori* what the subspaces are, or even how many of them there are. Consequently, one desires to determine the number of subspaces represented by the data, the dimension of each subspace, a basis for each subspace, and finally to cluster the data: the data $\{w_j\}_{j=1}^{n} \subset \mathscr{U}$ are not ordered in any particular way, and so clustering the data means to determine which data belong to the same subspace.

There are indeed physical systems which do fit into the model just described. Two particular examples are motion tracking and facial recognition. For example, the Yale Face Database B [6] contains images of faces, each taken with 64 different illumination patterns. Given a particular subject i, there are 64 images of their face illuminated differently, and each image represents a vector lying approximately in a low-dimensional linear subspace, S_i, of the higher dimensional space $\mathbb{R}^{307,200}$ (based on the size of the greyscale images). It has been experimentally shown that images of a given subject approximately lie in a subspace S_i having dimension 9 [7]. Consequently, a data matrix obtained from facial images under different illuminations has columns which lie in the union of low-dimensional subspaces, and one would desire an algorithm which can sort, or cluster, the data, thus recognizing which faces are the same.

There are many avenues of attack to the subspace clustering problem, including iterative and statistical methods [8], [9], [10], [11], [12], [13], algebraic methods [14], [15], [16], sparsity methods [17], [18], [19], [20], minimization problem methods inspired by compressed sensing [20], [21], and methods based on spectral clustering [18], [19], [22], [23], [24], [25], [26], [27]. For a thorough, though now incomplete, survey on the spectral clustering problem, the reader is invited to consult [28].

Many of the methods mentioned above begin by finding a *similarity matrix* for a given set of data, i.e. a square matrix whose entries are nonzero precisely when the corresponding data vectors lie in the same subspace, S_i, of $\mathscr{U}$. The present article is concerned with a certain matrix factorization method – the CUR decomposition – which provides a quite general framework for finding a similarity matrix for data that fits the subspace model above. It will be demonstrated that the CUR decomposition indeed produces similarity matrices for subspace data. Moreover, this decomposition provides a bridge between matrix factorization methods and the minimization problem methods such as Low-Rank Representation [20], [21].

A. Paper Contributions

- In this work, we show that the CUR decomposition gives rise to similarity matrices for clustering data that comes from a union of independent subspaces. Specifically, given the data matrix $\mathbf{W} = [w_1 \cdots w_n] \subset \mathbb{K}^m$ drawn from a union $\mathscr{U} = \bigcup_{i=1}^{M} S_i$ of independent subspaces $\{S_i\}_{i=1}^{M}$ of dimensions $\{d_i\}_{i=1}^{M}$, any CUR decomposition $\mathbf{W} = CU^{\dagger}R$ can be used to construct a similarity matrix for $\mathbf{W}$. In particular, if $Y = U^{\dagger}R$ and Q is the element-wise binary or absolute value version of Y^*Y, then $\Xi_{\mathbf{W}} = Q^{d_{max}}$ is a similarity matrix for $\mathbf{W}$; i.e. $\Xi_{\mathbf{W}}(i,j) \neq 0$ if the columns w_i and w_j of $\mathbf{W}$ come from the same subspace, and $\Xi_{\mathbf{W}}(i,j) = 0$ if the columns w_i and w_j of $\mathbf{W}$ come from different subspaces.
- This paper extends our previous framework for finding similarity matrices for clustering data that comes from the union of independent subspaces. In [29], we showed that any factorization $\mathbf{W} = BP$, where the columns of B come from $\mathscr{U}$ and form a basis for the column space of $\mathbf{W}$, can be used to produce a similarity matrix $\Xi_{\mathbf{W}}$. This work shows that we do not need to limit the factorization of $\mathbf{W}$ to bases, but may extend it to frames.
- Starting from the CUR decomposition framework, we demonstrate that some well-known methods utilized in subspace clustering follow as special cases, or are tied directly to the CUR decomposition; these methods include the *Shape Interaction Matrix* [30], [31] and *Low-Rank Representation [20], [21].*
- Experiments on synthetic and real data (specifically, the Hopkins155 motion dataset) are performed to justify the proposed theoretical framework. It is demonstrated that using an average of several CUR decompositions to find similarity matrices for a data matrix $\mathbf{W}$ outperforms some known methods in the literature.

B. Layout

The rest of the paper develops as follows: a brief section on preliminaries is followed by the statement and discussion of the most general exact CUR decomposition. Section IV contains the statements of the main results of the paper, while Section V contains the relation of the general framework that CUR gives for solving the subspace clustering problem. The proofs of the main theorems are enumerated in Section VI, whereupon the paper concludes with some numerical experiments and a discussion of future work.

II. Preliminaries

A. Definitions and Basic Facts

Throughout the sequel, $\mathbb{K}$ will refer to either the real or complex field ($\mathbb{R}$ or $\mathbb{C}$, respectively). For $A \in \mathbb{K}^{m\times n}$, its *Moore–Penrose psuedoinverse* is the unique matrix $A^\dagger \in \mathbb{K}^{n\times m}$ which satisfies the following conditions:

1) $AA^\dagger A = A$,
2) $A^\dagger AA^\dagger = A^\dagger$,
3) $(AA^\dagger)^* = AA^\dagger$, and
4) $(A^\dagger A)^* = A^\dagger A$.

Additionally, if $A = U\Sigma V^*$ is the Singular Value Decomposition of A, then $A^\dagger = V\Sigma^\dagger U^*$, where the pseudoinverse of the $m\times n$ matrix $\Sigma = \mathrm{diag}(\sigma_1,\dots,\sigma_r,0,\dots,0)$ is the $n\times m$ matrix $\Sigma^\dagger = \mathrm{diag}(1/\sigma_1,\dots,1/\sigma_r,0\dots,0)$. For these and other notions, see Section 5 of [32].

Also of utility to our analysis is that a rank r matrix has a so-called *skinny SVD* of the form $A = U_r\Sigma_r V_r^*$, where U_r comprises the first r left singular vectors of A, V_r comprises the first r right singular vectors, and $\Sigma_r = \mathrm{diag}(\sigma_1,\dots,\sigma_r) \in \mathbb{K}^{r\times r}$. Note that in the case that $\mathrm{rank}(A) > r$, the skinny SVD is simply a low-rank approximation of A.

Definition 1 (Independent Subspaces)**.** *Non-trivial subspaces $\{S_i \subset \mathbb{K}^m\}_{i=1}^M$ are called* independent *if their dimensions satisfy the following relationship:*

$$dim(S_1 + \cdots + S_M) = dim(S_1) + \cdots + dim(S_M) \leq m.$$

The definition above is equivalent to the property that any set of non-zero vectors $\{w_1,\cdots,w_M\}$ such that $w_i \in S_i$, $i = 1,\dots,M$ is linearly independent.

Definition 2 (Generic Data)**.** *Let S be a linear subspace of $\mathbb{K}^m$ with dimension d. A set of data $\mathbf{W}$ drawn from S is said to be* generic *if (i) $|\mathbf{W}| > d$, and (ii) every d vectors from $\mathbf{W}$ form a basis for S.*

Note this definition is equivalent to the frame-theoretic description that the columns of $\mathbf{W}$ are a frame for S with spark $d+1$ (see [33], [34]). It is also sometimes said that the data $\mathbf{W}$ is in *general position*.

Definition 3 (Similarity Matrix)**.** *Suppose $\mathbf{W} = [w_1\cdots w_n] \subset \mathbb{K}^m$ has columns drawn from a union of subspaces $\mathscr{U} = \bigcup_{i=1}^M S_i$. We say $\Xi_{\mathbf{W}}$ is a* similarity matrix for $\mathbf{W}$ *if and only if (i) $\Xi_{\mathbf{W}}$ is symmetric, and (ii) $\Xi_{\mathbf{W}}(i,j) \neq 0$ if and only if w_i and w_j come from the same subspace.*

Finally, if $A \in \mathbb{K}^{m\times n}$, we define its *absolute value version* via $\text{abs}(A)(i,j) = |A(i,j)|$, and its *binary version* via $\text{bin}(A)(i,j) = 1$ if $A(i,j) \neq 0$ and $\text{bin}(A)(i,j) = 0$ if $A(i,j) = 0$.

B. Assumptions

In the rest of what follows, we will assume that $\mathscr{U} = \bigcup_{i=1}^{M} S_i$ is a nonlinear set consisting of the union of non-trivial, independent, linear subspaces $\{S_i\}_{i=1}^{M}$ of $\mathbb{K}^m$, with corresponding dimensions $\{d_i\}_{i=1}^{M}$, with $d_{max} := \max_{1\leq i\leq M} d_i$. We will assume that the data matrix $\mathbf{W} = [w_1 \cdots w_n] \in \mathbb{K}^{m\times n}$ has column vectors that are drawn from $\mathscr{U}$, and that the data drawn from each subspace S_i is generic for that subspace.

III. CUR DECOMPOSITION

Our first tale is the remarkable CUR matrix decomposition, also known as the skeleton decomposition [35], [36] whose proof can be obtained by basic linear algebra.

Theorem 1. *Suppose $A \in \mathbb{K}^{m\times n}$ has rank r. Let $I \subset \{1,\ldots,m\}$, $J \subset \{1,\ldots,n\}$ with $|I| = s$ and $|J| = k$, and let C be the $m\times k$ matrix whose columns are the columns of A indexed by J. Let R be the $s\times n$ matrix whose rows are the rows of A indexed by I. Let U be the $s\times k$ sub-matrix of A whose entries are indexed by $I\times J$. If $\text{rank}(U) = r$, then $A = CU^{\dagger}R$.*

Proof. Since U has rank r, $\text{rank}(C) = r$. Thus the columns of C form a frame for the column space of A, and we have $A = CX$ for some (not necessarily unique) $k\times n$ matrix X. Let P_I be an $s\times m$ row selection matrix such that $R = P_I A$; then we have $R = P_I A = P_I CX$. Note also that $U = P_I C$, so that the last equation can then be written as $R = UX$. Since $\text{rank}(R) = r$, any solution to $R = UX$ is also a solution to $A = CX$. Thus the conclusion of the theorem follows upon observing that $Y = U^{\dagger}R$ is a solution to $R = UX$. Thus, $A = CY = CU^{\dagger}R$. ∎

Note that the assumption on the rank of U implies that $k,s \geq r$ in the theorem above. While this theorem is quite general, it should be noted that in some special cases, it reduces to a much simpler decomposition, a fact that is recorded in the following corollaries. The proof of each corollary follows from the fact that the pseudoinverse $U^{\dagger}$ takes those particular forms whenever the columns or rows are linearly independent ([32, p. 257], for example).

Corollary 1. *Let A,C,U, and R be as in Theorem 1 with $C \in \mathbb{K}^{m\times r}$; in particular, the columns of C are linearly independent. Then $A = C(U^*U)^{-1}U^*R$.*

Corollary 2. *Let A,C,U, and R be as in Theorem 1 with $R \in \mathbb{K}^{r\times n}$; in particular, the rows of R are linearly independent. Then $A = CU^*(UU^*)^{-1}R$.*

Corollary 3. *Let A,C,U, and R be as in Theorem 1 with $U \in \mathbb{K}^{r\times r}$; in particular, U is invertible. Then $A = CU^{-1}R$.*

In most sources, the decomposition of Corollary 3 is what is called the skeleton or CUR decomposition, [37], though the case when $k = s > r$ has been treated in [38]. The statement of Theorem 1 is the most general version of the exact CUR decomposition.

The precise history of the CUR decomposition is somewhat difficult to discern. Many articles cite Gantmacher [39], though the authors have been unable to find the term skeleton decomposition therein. Perhaps the modern starting point of interest in this decomposition is the work of Goreinov, Tyrtyshnikov, and Zamarashkin [35], [37]. They begin with the CUR decomposition as in Corollary 3, and study the error $\|A - CU^{-1}R\|_2$ in the case that A has rank larger than r, whereby the decomposition $CU^{-1}R$ is only approximate. Additionally, they allow more flexibility in the choice of U since computing the inverse directly may be computationally difficult. See also [40], [41], [42].

More recently, there has been renewed interest in this decomposition. In particular, Drineas, Kannan, and Mahoney [5] provide two randomized algorithms which compute an approximate CUR factorization of a given matrix A. Moreover, they provide error bounds based upon the probabilistic method for choosing C and R from A. It should also be noted that their middle matrix U is not $U^\dagger$ as in Theorem 1. Moreover, Mahoney and Drineas [1] give another CUR algorithm based on a way of selecting columns which provides nearly optimal error bounds for $\|A - CUR\|_F$ (in the sense that the optimal rank r approximation to any matrix A in the Frobenius norm is its skinny SVD of rank r, and they obtain error bounds of the form $\|A - CUR\|_F \leq (2+\varepsilon)\|A - U_r\Sigma_r V_r^T\|_F$). They also note that the CUR decomposition should be favored in analyzing real data that is low dimensional because the matrices C and R maintain the structure of the data, and the CUR decomposition actually admits an viable interpretation of the data as opposed to attempting to interpret singular vectors of the data matrix, which are generally linear combinations of the data. See also [43].

Subsequently, others have considered algorithms for computing CUR decompositions which still provide approximately optimal error bounds in the sense described above; see, for example, [2], [36], [44], [45], [46]. For applications of the CUR decomposition in various aspects of data analysis across scientific disciplines, consult [47], [48], [49], [50].

IV. Subspace Clustering via CUR Decomposition

Our second tale is one of the utility of the CUR decomposition in the similarity matrix framework for solving the subspace segmentation problem discussed above. Prior works have typically focused on CUR as a low-rank matrix approximation method which has a low cost, and also remains more faithful to the initial data than the singular value decomposition. This perspective is quite useful, but here we demonstrate what appears to be the first application in which CUR is responsible for an overarching framework, namely subspace clustering.

As mentioned in the introduction, one approach to clustering subspace data is to find a similarity matrix from which one can simply read off the clusters, at least when the data exactly fits the model and is considered to have no noise. The following theorem provides a way to find many similarity matrices for a given data matrix $\mathbf{W}$, all stemming from different CUR decompositions (recall that a matrix has very many CUR decompositions depending on which columns and rows are selected).

Theorem 2. *Let $\mathbf{W} = [w_1 \cdots w_n] \in \mathbb{K}^{m\times n}$ be a rank r matrix whose columns are drawn from $\mathscr{U}$. Let $\mathbf{W}$ be factorized as $\mathbf{W} = CU^{\dagger}R$ where $C \in \mathbb{K}^{m\times k}$, $R \in \mathbb{K}^{s\times n}$, and $U \in \mathbb{K}^{s\times k}$ are as in Theorem 1, and let $Y = U^{\dagger}R$ and Q be either the binary or absolute value version of Y^*Y. Then, $\Xi_{\mathbf{W}} = Q^{d_{max}}$ is a similarity matrix for $\mathbf{W}$.*

The key ingredient in the proof of Theorem 2 is the fact that the matrix $Y = U^{\dagger}R$, which generates the similarity matrix, has a block diagonal structure due to the independent subspace structure of $\mathscr{U}$; this fact is captured in the following theorem.

Theorem 3. *Let $\mathbf{W}, C, U$, and R be as in Theorem 2. If $Y = U^{\dagger}R$, then there exists a permutation matrix P such that*

$$YP = \begin{bmatrix} Y_1 & 0 & \cdots & 0 \\ 0 & Y_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & Y_M \end{bmatrix},$$

where each Y_i is a matrix of size $k_i \times n_i$, where n_i is the number of columns in $\mathbf{W}$ from subspace S_i, and k_i is the number of columns in C from S_i. Moreover, $\mathbf{W}P$ has the form $[\mathbf{W}_1 \ldots \mathbf{W}_M]$ where the columns of $\mathbf{W}_i$ come from the subspace S_i.

Perhaps the most important consequence of Theorem 2 is that the shape interaction matrix is a special case of the general framework of the CUR decomposition.

Corollary 4. *Let* $\mathbf{W} = [w_1 \cdots w_n] \in \mathbb{K}^{m\times n}$ *be a rank r matrix whose columns are drawn from* $\mathscr{U}$*. Let* $\mathbf{W}$ *be factorized as* $\mathbf{W} = \mathbf{W}\mathbf{W}^\dagger\mathbf{W}$*. Let Q be either the binary or absolute value version of* $\mathbf{W}^\dagger\mathbf{W}$*. Then* $\Xi_{\mathbf{W}} = Q^{d_{max}}$ *is a similarity matrix for* $\mathbf{W}$*. Moreover, if the skinny singular value decomposition of* $\mathbf{W}$ *is* $\mathbf{W} = U_r\Sigma_r V_r^*$*, then* $\mathbf{W}^\dagger\mathbf{W} = V_r V_r^*$*.*

Corollary 5. *Let* $\mathbf{W} = [w_1 \cdots w_n] \in \mathbb{K}^{m\times n}$ *be a rank r matrix whose columns are drawn from* $\mathscr{U}$*. Choose* $C = \mathbf{W}$*, and R to be any rank r row restriction of* $\mathbf{W}$*. Then* $\mathbf{W} = \mathbf{W}R^\dagger R$ *by Theorem 1. Moreover,* $R^\dagger R = \mathbf{W}^\dagger\mathbf{W} = V_r V_r^*$*, where* V_r *is as in Corollary 4.*

The proofs of the above facts will be related in a subsequent section.

V. Special Cases

A. Shape Interaction Matrix

In their pioneering work on factorization methods for motion tracking [30], Costeira and Kanade introduced the *Shape Interaction Matrix*, or SIM. Given a data matrix $\mathbf{W}$ whose skinny SVD is $U_r\Sigma_r V_r^*$, SIM($\mathbf{W}$) is defined to be $V_r V_r^*$. Following their work, this has found wide utility in theory and in practice. Their observation was that the SIM often provides a similarity matrix for data coming from independent subspaces. It should be noted that in [29], it was shown that examples of data matrices $\mathbf{W}$ can be found such that $V_r V_r^*$ is not a similarity matrix for $\mathbf{W}$; however, it was noted there that SIM($\mathbf{W}$) is almost surely a similarity matrix.

Note though that there is an easy way around this due to Theorem 2: if $Q = \mathrm{abs}(V_r V_r^*)$ or $\mathrm{bin}(V_r V_r^*)$, then $Q^{d_{\max}}$ is a similarity matrix for $\mathbf{W}$. This also provides some theoretical justification for the idea of the *Robust Shape Interaction Matrix* for $\mathbf{W}$ in [31], where the authors took powers (albeit entrywise) of SIM($\mathbf{W}$) to get a better similarity matrix for the (noisy) data.

B. Low-Rank Representation Algorithm

Another class of methods for solving the subspace clustering problem arises from solving some sort of minimization problem. It has been noted that in many cases such methods are intimately related to some matrix factorization methods [28], [51].

One particular instance of a minimization based algorithm is the *Low Rank Representation* (LRR) algorithm of Liu, Lin, and Yu [20], [21]. As a starting point, the authors consider the following rank minimization problem:

$$\min_Z \mathrm{rank}(Z) \quad \text{s.t.} \quad \mathbf{W} = AZ, \tag{V.1}$$

where A is a dictionary that linearly spans $\mathbf{W}$.

Note that there is indeed something to minimize over here since if $A = \mathbf{W}$, $Z = I_{n\times n}$ satisfies the constraint, and evidently $\text{rank}(Z) = n$; however, if $\text{rank}(\mathbf{W}) = r$, then $Z = \mathbf{W}^\dagger\mathbf{W}$ is a solution to $\mathbf{W} = \mathbf{W}Z$, and it can be easily shown that $\text{rank}(\mathbf{W}^\dagger\mathbf{W}) = r$. Note further that any Z satisfying $\mathbf{W} = \mathbf{W}Z$ must have rank at least r, and so we have the following.

Proposition 1. *Let* $\mathbf{W}$ *be a rank* r *data matrix whose columns are drawn from* $\mathscr{U}$*, then* $\mathbf{W}^\dagger\mathbf{W}$ *is a solution to the minimization problem*

$$\min_Z \text{rank}(Z) \quad s.t. \quad \mathbf{W} = \mathbf{W}Z.$$

Note that in general, solving this minimization problem V.1 is NP–hard since it is equivalent to minimizing $\|\sigma(Z)\|_0$ where $\sigma(Z)$ is the vector of singular values of Z, and $\|\cdot\|_0$ is the number of nonzero entries of a vector. Additionally, the solution to Equation V.1 is generally not unique, so typically the rank function is replaced with some norm to produce a convex optimization problem. Based upon intuition from the compressed sensing literature, it is natural to consider replacing $\|\sigma(Z)\|_0$ by $\|\sigma(Z)\|_1$, which is the definition of the nuclear norm, denoted by $\|Z\|_*$ (also called the trace norm, Ky–Fan norm, or Shatten 1–norm). In particular, in [20], the following was considered:

$$\min_Z \|Z\|_* \quad \text{s.t.} \quad \mathbf{W} = AZ. \tag{V.2}$$

Solving this minimization problem applied to subspace clustering is dubbed *Low-Rank Representation* by the authors in [20].

Let us now specialize these problems to the case when the dictionary is chosen to be the whole data matrix, in which case we have

$$\min_Z \|Z\|_* \quad \text{s.t.} \quad \mathbf{W} = \mathbf{W}Z. \tag{V.3}$$

It was shown in [21], [51] that the SIM defined in Section V-A, is the unique solution to problem (V.3):

Theorem 4 ([51], Theorem 3.1)**.** *Let* $\mathbf{W} \in \mathbb{K}^{m\times n}$ *be a rank* r *matrix whose columns are drawn from* $\mathscr{U}$*, and let* $\mathbf{W} = U_r\Sigma_r V_r^*$ *be its skinny SVD. Then* $V_rV_r^*$ *is the unique solution to* (V.3)*.*

For clarity and completeness of exposition, we supply a simpler proof of Theorem 4 here than appears in [51].

Proof. Suppose that $\mathbf{W} = U\Sigma V^*$ is the full SVD of $\mathbf{W}$. Then since $\mathbf{W}$ has rank r, we can write

$$\mathbf{W} = U\Sigma V^* = \begin{bmatrix} U_r & \tilde{U}_r \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_r^* \\ \tilde{V}_r^* \end{bmatrix},$$

where $U_r\Sigma_r V_r^*$ is the skinny SVD of $\mathbf{W}$. Then if $\mathbf{W} = \mathbf{W}Z$, we have that $I - Z$ is in the null space of $\mathbf{W}$. Consequently, $I - Z = \tilde{V}_r X$ for some X. Thus we find that

$$\begin{aligned} Z &= I + \tilde{V}_r X \\ &= VV^* + \tilde{V}_r X \\ &= V_r V_r^* + \tilde{V}_r \tilde{V}_r^* + \tilde{V}_r X \\ &= V_r V_r^* + \tilde{V}_r(\tilde{V}_r^* + X) \\ &=: A + B. \end{aligned}$$

Recall that the nuclear norm is unchanged by multiplication on the left or right by unitary matrices, whereby we see that $\|Z\|_* = \|V^*Z\|_* = \|V^*A + V^*B\|_*$. However,

$$V^*A + V^*B = \begin{bmatrix} V_r^* \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{V}_r^* + X \end{bmatrix}.$$

Due to the above structure, we have that $\|V^*A + V^*B\|_* \geq \|V^*A\|_*$, with equality if and only if $V^*B = 0$ (for example, via the same proof as [52, Lemma 1], or also Lemma 7.2 of [21]).

It follows that

$$\|Z\|_* > \|A\|_* = \|V_r V_r^*\|_*,$$

for any $B \neq 0$. Hence $Z = V_r V_r^*$ is the unique minimizer of problem (V.3).

■

Corollary 6. *Let* $\mathbf{W}$ *be as in Theorem 4, and let* $\mathbf{W} = \mathbf{W}R^\dagger R$ *be a factorization of* $\mathbf{W}$ *as in Theorem 1. Then* $R^\dagger R = W^\dagger W = V_r V_r^*$ *is the unique solution to the minimization problem* (V.3).

Proof. Combine Corollary 5 and Theorem 4. ■

Let us note that the unique minimizer of problem (V.2) is known for general dictionaries as the following result of Liu, Lin, Yan, Sun, Yu, and Ma demonstrates.

Theorem 5 ([21], Theorem 4.1)**.** *Suppose that A is a dictionary that linearly spans* $\mathbf{W}$*. Then the unique minimizer of problem* (V.2) *is*

$$Z = A^\dagger \mathbf{W}.$$

The following corollary is thus immediate from the CUR decomposition.

Corollary 7. *If* $\mathbf{W}$ *has CUR decomposition* $\mathbf{W} = CC^{\dagger}\mathbf{W}$ *(where* $R = \mathbf{W}$*, hence* $U = C$*, in Theorem 1), then* $C^{\dagger}\mathbf{W}$ *is the unique solution to*

$$\min_{Z} \|Z\|_{*} \quad s.t. \quad \mathbf{W} = CZ.$$

The above theorems and corollaries provide a theoretical bridge between the shape interaction matrix, CUR decomposition, and Low-Rank Representation. In particular, in [51], the authors observe that of primary interest is that while LRR stems from sparsity considerations à la compressed sensing, its solution in the noise free case in fact comes from matrix factorization, which is quite interesting.

C. Basis Framework of [29]

As a final note, the CUR framework proposed here gives a broader vantage point for obtaining similarity matrices than that of [29]. Consider the following, which is the main result therein:

Theorem 6 ([29], Theorem 2)**.** *Let* $\mathbf{W} \in \mathbb{K}^{m\times n}$ *be a rank r matrix whose columns are drawn from* $\mathscr{U}$*. Suppose* $\mathbf{W} = BP$ *where the columns of B form a basis for the column space of* $\mathbf{W}$ *and the columns of B lie in* $\mathscr{U}$ *(but are not necessarily columns of* $\mathbf{W}$*). If* $Q = abs(P^{*}P)$ *or* $Q = bin(P^{*}P)$*, then* $\Xi_{\mathbf{W}} = Q^{d_{\max}}$ *is a similarity matrix for* $\mathbf{W}$*.*

Let us pause to note that at first glance, Theorem 6 is on the one hand more specific than Theorem 2 since the matrix B must be a basis for the span of $\mathbf{W}$, whereas C may have some redundancy. On the other hand, Theorem 6 seems more general in that the columns of B need only come from $\mathscr{U}$, but are not forced to be columns of $\mathbf{W}$ as are the columns of C. However, one need only notice that if $\mathbf{W} = BP$ as in the theorem above, then defining $\tilde{\mathbf{W}} = [\mathbf{W}\ B]$ gives rise to the CUR decomposition $\tilde{\mathbf{W}} = BB^{\dagger}\tilde{\mathbf{W}}$. But clustering the columns of $\tilde{\mathbf{W}}$ via Theorem 2 automatically gives the clusters of $\mathbf{W}$.

VI. Proofs

Here we enumerate the proofs of the results in Section IV, beginning with some lemmata.

A. Some Useful Lemmata

The first lemma follows immediately from the definition of independent subspaces.

Lemma 1. *Suppose $U = [U_1 \dots U_M]$ where each U_i is a submatrix whose columns come from independent subspaces of $\mathbb{K}^m$. Then we may write*

$$U = [B_1 \ \dots \ B_M] \begin{bmatrix} V_1 & 0 & \dots & 0 \\ 0 & V_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & V_M \end{bmatrix}.$$

where the columns of B_i form a basis for the column space of U_i.

The next lemma is a special case of [29, Lemma 1].

Lemma 2. *Suppose that $U \in \mathbb{K}^{m\times n}$ has columns which are generic for the subspace S of $\mathbb{K}^m$ from which they are drawn. Suppose $P \in \mathbb{K}^{r\times m}$ is a row selection matrix such that* $\operatorname{rank}(PU) = \operatorname{rank}(U)$. *Then the columns of PU are generic.*

Lemma 3. *Suppose that $U = [U_1 \ \dots \ U_M]$, and that each U_i is a submatrix whose columns come from independent subspaces S_i, $i = 1,\dots,M$ of $\mathbb{K}^m$, and that the columns of U_i are generic for S_i. Suppose that $P \in \mathbb{K}^{r\times m}$ with $r \geq \operatorname{rank}(U)$ is a row selection matrix such that* $\operatorname{rank}(PU) = \operatorname{rank}(U)$. *Then the subspaces $P(S_i)$, $i = 1,\dots,M$ are independent.*

Proof. Let $S = S_1 + \dots + S_M$. Let $d_i = \dim(S_i)$, and $d = \dim(S)$. From the hypothesis, we have that $\operatorname{rank}(PU_i) = d_i$, and $\operatorname{rank}(PU) = d = \sum_{i=1}^{M} d_i$. Therefore, there are d linearly independent vectors for $P(S)$ in the columns of PU. Moreover, since $PU = [PU_1,\dots,PU_M]$, there exist d_i linearly independent vectors from the columns of PU_i for $P(S_i)$. Thus, $\dim P(S) = d = \sum_i d_i = \sum_i \dim P(S_i)$, and the conclusion of the lemma follows. ■

The next few facts which will be needed come from basic graph theory. Suppose $G = (V,E)$ is a finite, undirected, weighted graph with vertices in the set $V = \{v_1,\dots,v_k\}$ and edges E. The geodesic distance between two vertices $v_i, v_j \in V$ is the length (i.e. number of edges) of the shortest path connecting v_i and v_j, and the *diameter* of the graph G is the maximum of the pairwise geodesic distances of the vertices. To each weighted graph is associated an adjacency matrix, A, such that $A(i,j)$ is nonzero if there is an edge between the vertices v_i and v_j, and 0 if not. We call a graph *positively weighted* if $A(i,j) \geq 0$ for all i and j. From these facts, we have the following lemma.

Lemma 4. *Let G be a finite, undirected, positively weighted graph with diameter r, and let A be its adjacency matrix. Then $A^r(i,j) > 0$ for all i, j. In particular, A^r is the adjacency matrix of a fully connected*

graph.

Proof. See [29, Corollary 1]. ■

The following lemma connects the graph theoretic considerations with the subspace model described in the opening.

Lemma 5. *Let $V = \{p_1, \dots, p_N\}$ be a set of generic vectors that represent data from a subspace S of dimension r and $N > r \geq 1$. Let Q be as in Theorem 2 for the case $\mathscr{U} = S$. Finally, let G be the graph whose vertices are p_i and whose edges are those $p_i p_j$ such that $Q(i,j) > 0$. Then G is connected, and has diameter at most r. Moreover, Q^r is the adjacency matrix of a fully connected graph.*

Proof. See [29, Lemmas 2 and 3] for the proof of the first part. The moreover statement follows directly from Lemma 4. ■

B. Proof of Theorem 3

Without loss of generality, we assume that $\mathbf{W}$ is such that $\mathbf{W} = [\mathbf{W}_1 \dots \mathbf{W}_M]$ where $\mathbf{W}_i$ is an $m \times n_i$ matrix for each $i = 1, \dots, M$ and $\sum_i^M n_i = n$, and the vector columns of $\mathbf{W}_i$ come from the subspace S_i. Under this assumption, we will show that Y is a block diagonal matrix.

Let P be the row selection matrix such that $P\mathbf{W} = R$; in particular, note that P maps $\mathbb{R}^m$ to $\mathbb{R}^s$, and that because of the structure of $\mathbf{W}$, we may write $R = [R_1 \dots R_M]$ where the columns of R_i belong to the subspace $\tilde{S}_i := P(S_i)$. Note also that the columns of each R_i are generic for the subspace $\tilde{S}_i$ on account of Lemma 2, and that the subspaces $\tilde{S}_i$ are independent by Lemma 3. Additionally, since U consists of certain columns of R, and $\operatorname{rank}(U) = \operatorname{rank}(R) = \operatorname{rank}(\mathbf{W})$ by assumption, we have that $U = [U_1 \ \dots \ U_M]$ where the columns of U_i are in $\tilde{S}_i$.

Next, recall from the proof of the CUR decomposition that $Y = U^\dagger R$ is a solution to $R = UX$; thus $R = UY$. Suppose that r is a column of R_1, and let $y = [y_1 \ y_2 \ \dots \ y_M]^*$ be the corresponding column of Y so that $r = Uy$. Then we have that $r = \sum_{j=1}^M U_j y_j$, and in particular, since r is in R_1,

$$(U_1 y_1 - r) + U_2 y_2 + \cdots + U_M y_M = 0,$$

whereupon the independence of the subspaces $\tilde{S}_i$ implies that $U_1 y_1 = r$ and $U_i y_i = 0$ for every $i = 2, \dots, M$. On the other hand, note that $\tilde{y} = [y_1 \ 0 \ \dots \ 0]^*$ is another solution; i.e. $r = U\tilde{y}$. Recalling that $U^\dagger y$ is the minimal-norm solution to the problem $r = Uy$, we must have that

$$\|y\|_2^2 = \sum_{i=1}^M \|y_i\|_2^2 \leq \|\tilde{y}\|_2^2 = \|y_1\|_2^2.$$

Consequently, $y = \tilde{y}$, and it follows that Y is block diagonal by applying the same argument for columns of R_i, $i = 2, \ldots, M$.

■

C. Proof of Theorem 2

Without loss of generality, on account of Theorem 3, we may assume that Y is block diagonal as above. Then we first demonstrate that each block Y_i has rank $d_i = \dim(S_i)$ and has columns which are generic. Since $R_i = U_i Y_i$, and $\operatorname{rank}(R_i) = \operatorname{rank}(U_i) = d_i$, we have $\operatorname{rank}(Y_i) \geq d_i$ since the rank of a product is at most the minimum of the ranks. On the other hand, since $Y_i = U^\dagger R_i$, $\operatorname{rank}(Y_i) \leq \operatorname{rank}(R_i) = d_i$, whence $\operatorname{rank}(Y_i) = d_i$. To see that the columns of each Y_i are generic, let $y_1, \ldots, y_{d_i}$ be d_i columns in Y_i and suppose there are constants $c_1, \ldots, c_{d_i}$ such that $\sum_{j=1}^{d_i} c_j y_j = 0$. It follows from the block diagonal structure of Y that

$$0 = U_i \left(\sum_{j=1}^{d_i} c_j y_j \right) = \sum_{j=1}^{d_i} c_j U_i y_j = \sum_{j=1}^{d_i} c_j r_j,$$

where r_j, $j = 1, \ldots, d_i$ are the columns of R_i. Since the columns of R_i are generic by Lemma 2, it follows that $c_i = 0$ for all i, whence the columns of Y_i are generic.

Now $Q = Y^* Y$ is an $n \times n$ block diagonal matrix whose blocks are given by $Q_i = Y_i^* Y_i$, $i = 1, \ldots, M$, and we may consider each block as the adjacency matrix of a graph as prescribed in Lemma 4. Thus from the conclusion there, each block gives a connected graph with diameter d_i, and so $Q^{d_{\max}}$ will give rise to a graph with M distinct fully connected components, where the graph arising from Q_i corresponds to the data in $\mathbf{W}$ drawn from S_i. Thus $Q^{d_{\max}}$ is indeed a similarity matrix for $\mathbf{W}$.

■

D. Proofs of Corollaries

Proof of Corollary 4. That Ξ_W is a similarity matrix follows directly from Theorem 2. To see the moreover statement, recall that $\mathbf{W}^\dagger = V_r \Sigma_r^\dagger U_r^*$, whence $\mathbf{W}^\dagger \mathbf{W} = V_r \Sigma_r^\dagger U_r U_r^* \Sigma_r V_r^* = V_r \Sigma_r^\dagger \Sigma_r V_r^* = V_r V_r^*$. ■

Proof of Corollary 5. By Lemma 1, we may write $\mathbf{W} = BZ$, where Z is block diagonal, and $B = [B_1 \ \ldots \ B_M]$ with the columns of B_i being a basis for S_i. Let P be the row-selection matrix which gives R, i.e. $R = P\mathbf{W}$. Then $R = PBZ$. The columns of B are linearly independent (and likewise for the columns of PB by Lemma 3), whence $\mathbf{W}^\dagger = Z^\dagger B^\dagger$, and $R^\dagger = Z^\dagger (PB)^\dagger$. Moreover, linear independence of the columns also implies that $B^\dagger B$ and $(PB)^\dagger PB$ are both identity matrices of the appropriate size, whereby

$$R^\dagger R = Z^\dagger (PB)^\dagger PBZ = Z^\dagger Z = Z^\dagger B^\dagger BZ = \mathbf{W}^\dagger \mathbf{W},$$

which completes the proof (note that the final identity $\mathbf{W}^{\dagger}\mathbf{W} = V_r V_r^*$ follows immediately from Corollary 4). ■

VII. Experimental Results

In this section, the use of the CUR decomposition in practice is investigated by first considering its performance on synthetic data, whereupon the initial findings are subsequently verified using the motion segmentation dataset known as Hopkins155 [53]. In motion segmentation, one begins with a video, and some sort of algorithm which extracts features on moving objects and tracks the positions of those features over time. At the end of the video, one obtains a data matrix of size $2F \times N$ where F is the number of frames in the video and N is the number of features tracked. Each vector corresponds to the trajectory of a fixed feature, i.e. is of the form $(x_1, y_1, \ldots, x_F, y_F)$, where (x_i, y_i) is the position of the feature at frame $1 \leq i \leq F$. Even though these trajectories are vectors in a high dimensional ambient space, it is known that the trajectories of all feature belonging to the same rigid body lie in a subspace of dimension 4 [54]. Consequently, motion segmentation is a suitable practical problem to tackle in order to verify the validity of the proposed approach.

The Hopkins155 motion dataset contains 155 videos which can be broken down into several categories: checkerboard sequences where moving objects are overlaid with a checkerboard pattern to obtain many feature points on each moving object, traffic sequences, and articulated motion (such as people walking) where the moving body contains joints of some kind making the 4-dimensional subspace assumption on the trajectories incorrect. Associated with each video is a data matrix giving the trajectories of all features on the moving objects (these features are fixed in advance for easy comparison). Consequently, the data matrix is unique for each video, and the ground truth for clustering is known *a priori*, thus allowing calculation of the clustering error, which is simply the percentage of incorrectly clustered feature points.

In the synthetic experiments, similarity matrices that are generated using CUR and SIM are clustered using the Spectral Clustering algorithm [22], and both methods are compared in terms of clustering performance. In the case of clustering real motion segmentation from the Hopkins155 dataset (where the ground truth of clusters is known), our clustering performance is compared to known results in the literature [55].

As expected, in the case where the synthetic data contains no noise, SIM and CUR yield the same result for clustering. In the presence of noise, it makes sense to use the flexibility of the CUR decomposition and take an average of several different such decompositions (for further explanation, see the next subsection). In this case, we will see that an averaging of several CUR-derived similarity matrices yields better or

similar results to the SIM.

A. Simulations using Synthetic Data

A set of simulations are designed using synthetic data. In order for the results to be comparable to that of the motion segmentation case presented in the following section, the data is constructed in a similar fashion. Particularly, in the synthetic experiments, data comes from the union of independent 4 dimensional subspaces of $\mathbb{R}^{300}$. This corresponds to a feature being tracked for 5 seconds in a 30 fps video stream. Two cases similar to the ones in Hopkins155 dataset are investigated for increasing levels of noise. In both cases, the data is randomly drawn from the unit ball of a 4 dimensional subspace. In the first case, data comes from the union of 2 subspaces, whereas in the second case, the number of subspaces is 3. In both cases, the level of noise on $\mathbf{W}$ is gradually increased from the initial noiseless state to the maximum noise level. The entries of the noise are i.i.d. $\mathcal{N}(0,\sigma^2)$ random variables (i.e. with zero-mean and variance σ^2), where the variance increases as $\sigma = [0.000,\ 0.001,\ 0.010,\ 0.030,\ 0.050]$.

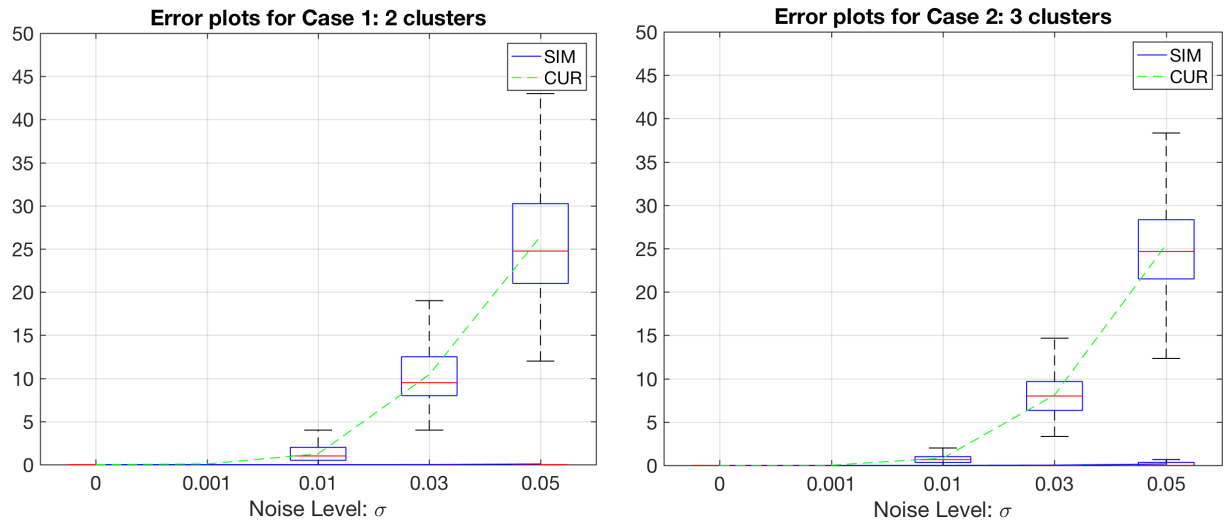
In each case, for each noise level, 100 data matrices are randomly generated. Once each data matrix $\mathbf{W}$ is formed, a similarity matrix $\Xi_{\mathbf{W}}$ is generated using CUR and SIM. These similarity matrices are fed to Spectral Clustering and the results are evaluated based on clustering error percentages.

$\Xi_{\mathbf{W}}$ for SIM is found using the skinny SVD of $\mathbf{W}$ with the expected rank of the data matrix (i.e. 8 and 12 for Case 1 and Case 2, respectively), while that for CUR is found by using all columns and the expected rank number of rows of $\mathbf{W}$. Therefore, the matrix Y of Theorem 2 is $R^{\dagger}R$. Rows are chosen uniformly at random from $\mathbf{W}$, and it is ensured that R has the expected rank before proceeding.

Results for Case 1 and Case 2 are given in Figure 1. Results indicate that CUR performs exactly as SIM for the noiseless case (as expected via Corollary 5) and degrades gradually as the level of noise increases.

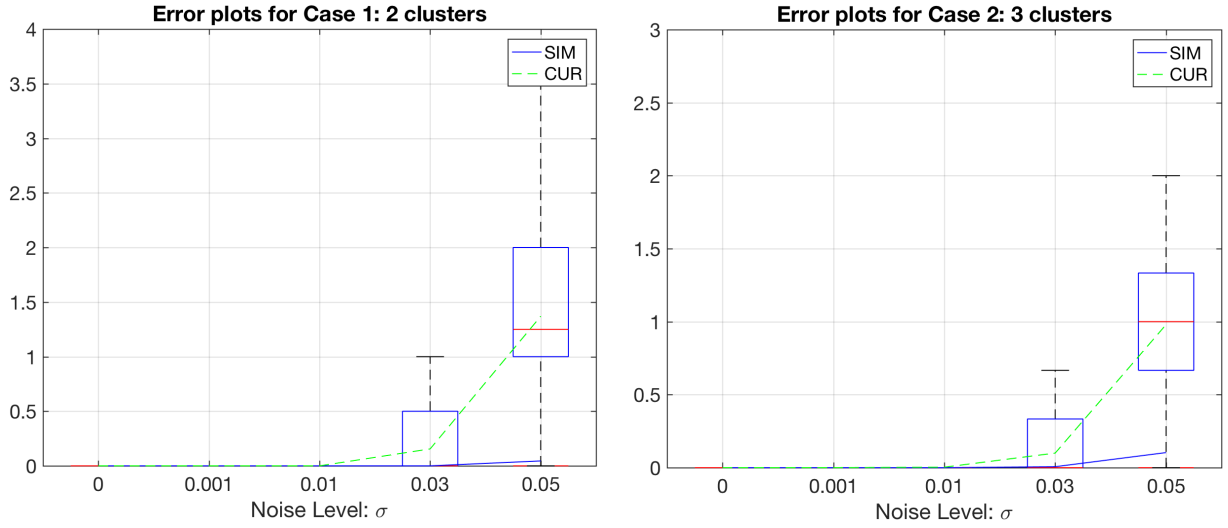
Given the random nature of the CUR decomposition, a simple improvement on CUR performance is achieved by calculating the similarity matrix $\Xi_{\mathbf{W}}$ not by using a single CUR decomposition, but by calculating n different CUR decompositions from the same data matrix. For each CUR decomposition, a similarity matrix $\Xi^i_{\mathbf{W}}$ is found, and the final similarity matrix for $\mathbf{W}$ is given by taking the median entrywise of the family $\{\Xi^1_{\mathbf{W}},\ldots,\Xi^n_{\mathbf{W}}\}$.

If the same experiments are repeated, for $n = 25$, significant improvements are achieved as shown in Figure 2. Extensive testing shows no real improvement for larger values of n, so this value was settled on empirically.



(a) Case 1 (b) Case 2

Fig. 1: Synthetic Cases 1 and 2 for $\Xi_{\mathbf{W}}$ calculated using a single CUR decomposition



(a) Case 1 (b) Case 2

Fig. 2: Synthetic Cases 1 and 2 for $\Xi_{\mathbf{W}}$ calculated using the median of 25 CUR decompositions

B. Motion Segmentation Dataset: Hopkins155

As mentioned, clustering performance using CUR decompositions is tested using the Hopkins155 dataset. Here, $\Xi_{\mathbf{W}}$ is calculated as in Case 2 of the synthetic data section; i.e. $\Xi_{\mathbf{W}}$ for CUR is still the aggregate result for $n = 25$ different similarity matrices as explained above. It turns out that for real

motion data, CUR yields better overall results than SIM. This is reasonable given the flexibility of the CUR decomposition. Finding several similarity matrices and averaging them has the effect of averaging out some of the inherent noise in the data. The purpose of this work is not to fine tune CUR's performance on the Hopkins155 dataset; nonetheless, the results using this very simple method are already better than some of those in the literature.

TABLE I: % classification errors for sequences with two motions.

Checker (78)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS	SIM	CUR
Average	6.09%	2.57%	6.52%	4.46%	1.55%	0.83%	1.12%	0.23%	2.49%	1.81%
Median	1.03%	0.27%	1.75%	0.00%	0.29%	0.00%	0.00%	0.00%	0.50%	0.00%
Traffic (31)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS	SIM	CUR
Average	1.41%	5.43%	2.55%	2.23%	1.59%	0.23%	0.02%	1.40%	10.76%	7.88%
Median	0.00%	1.48%	0.21%	0.00%	1.17%	0.00%	0.00%	0.00%	2.44%	1.30%
Articulated (11)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS	SIM	CUR
Average	2.88%	4.10%	7.25%	7.23%	10.70%	1.63%	0.62%	1.77%	8.75%	19.38%
Median	0.00%	1.22%	2.64%	0.00%	0.95%	0.00%	0.00%	0.88%	1.37%	11.11%
All (120 seq)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS	SIM	CUR
Average	4.59%	3.45%	5.56%	4.14%	2.40%	0.75%	0.82%	0.57%	5.20%	4.99%
Median	0.38%	0.59%	1.18%	0.00%	0.43%	0.00%	0.00%	0.00%	1.06%	0.27%

TABLE II: % classification errors for sequences with three motions.

Checker (26)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS	SIM	CUR
Average	31.95%	5.80%	25.78%	10.38%	5.20%	4.49%	2.97%	0.87%	8.82%	7.66%
Median	32.93%	1.77%	26.00%	4.61%	0.67%	0.54%	0.27%	0.35%	4.10%	1.56%
Traffic (7)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS	SIM	CUR
Average	19.83%	25.07%	12.83%	1.80%	7.75%	0.61%	0.58%	1.86%	29.02%	31.27%
Median	19.55%	23.79%	11.45%	0.00%	0.49%	0.00%	0.00%	1.53%	32.27%	33.58%
Articulated (2)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS	SIM	CUR
Average	16.85%	7.25%	21.38%	2.71%	21.08%	1.60%	1.60%	5.12%	20.55%	19.48%
Median	16.85%	7.25%	21.38%	2.71%	21.08%	1.60%	1.60%	5.12%	20.55%	19.48%
All (35 seq)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS	SIM	CUR
Average	28.66%	9.73%	22.94%	8.23%	6.69%	3.55%	2.45%	1.31%	13.53%	13.06%
Median	28.26%	2.33%	22.03%	1.76%	0.67%	0.25%	0.20%	0.45%	6.51%	4.05%

It should be noted that after thorough experimentation on noisy data, using a CUR decomposition which takes all columns and exactly the expected rank number of rows exhibits the best performance.

TABLE III: % classification errors for all sequences.

All (155 seq)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS	SIM	CUR
Average	10.34%	4.94%	9.76%	5.03%	3.56%	1.45%	1.24%	0.76%	7.08%	6.81%
Median	2.54%	0.90%	3.21%	0.00%	0.50%	0.00%	0.00%	0.20%	1.37%	0.52%

That is, a decomposition of the form $\mathbf{W} = \mathbf{W}R^{\dagger}R$ performs better on average than one of the form $\mathbf{W} = CU^{\dagger}R$. The fact that choosing more columns performs better when the matrix $\mathbf{W}$ is noisy makes sense in that any representation of the form $\mathbf{W} = CX$ is a representation of $\mathbf{W}$ in terms of the frame vectors of C. Consequently, choosing more columns in the matrix C means that we are adding redundancy to the frame, and it is well-known to be one of the advantages of frame representations that redundancy provides greater robustness to noise. Additionally, we noticed experimentally that choosing exactly r rows in the decomposition $\mathbf{W}R^{\dagger}R$ exhibits the best performance. It is not clear as of yet why this is the case.

As a final remark, let us note that the CUR algorithm for finding similarity matrices for subspace data discussed here can be quite fast. Indeed, it took only 57 seconds to cluster all 155 data matrices in the Hopkins155 dataset. This means that it takes approximately 0.37 seconds per sequence to perform the clustering via averaged CUR similarity matrices plus spectral clustering. This is faster than most of the algorithms used [53], which is another advantage given its good performance.

VIII. Concluding Remarks

The motivation of this work was truly the realization that the exact CUR decomposition of Theorem 1 can be used for the subspace clustering problem. We demonstrated that, on top of its utility in randomized linear algebra, CUR enjoys a prominent place atop the landscape of solutions to the subspace clustering problem. CUR provides a theoretical umbrella under which sits the known shape interaction matrix, but it also provides a bridge to other solution methods inspired by compressed sensing, i.e. those involving the solution of a minimization problem. Moreover, we believe that the utility of CUR for clustering and other applications will only increase in the future. Below, we provide some reasons for the practical utility of CUR decompositions, particularly related to data analysis and clustering, as well as some future directions.

Benefits of CUR:

- From a theoretical standpoint, the CUR decomposition of a matrix is utilizing a frame structure rather than a basis structure to factorize the matrix, and therefore enjoys a level of flexibility beyond something like the SVD. This fact should provide utility for applications.

- Additionally, a CUR decomposition remains faithful to the structure of the data. For example, if the given data is sparse, then both C and R will be sparse, even if $U^\dagger$ is not in general. In contrast, taking the SVD of a sparse matrix yields full matrices U and V, in general.
- Often, in obtaining real data, many entries may be missing or extremely corrupted. In motion tracking, for example, it could be that some of the features are obscured from view for several frames. Consequently, some form of matrix completion may be necessary. On the other hand, a look at the CUR decomposition reveals that whole rows of a data matrix can be missing as long as we can still choose enough rows such that the resulting matrix R has the same rank as $\mathbf{W}$.

Future Directions

- It remains to be seen the best way to use the CUR decomposition to cluster subspace data. The naive method tested here only gives a baseline, but further testing needs to be done. As commented above, its flexibility is a distinct advantage over SVD based methods. Consequently, some of the tools to make SIM more robust will no doubt lead to similar improvements for CUR.
- Another direction is to combine the CUR technique with sparse methods to construct algorithms that are strongly robust to noise and that allow clustering when the data points are not drawn from a union of independent subspaces.

ACKNOWLEDGEMENT

The research of A. Sekmen, and A.B. 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.B. Koku is also supported by TUBITAK-2219-1059B191600150.

REFERENCES

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

[2] C. Boutsidis, D. P. Woodruff, Optimal CUR matrix decompositions, SIAM Journal on Computing 46 (2) (2017) 543–589.

[3] P. Drineas, R. Kannan, M. W. Mahoney, Fast monte carlo algorithms for matrices I: Approximating matrix multiplication, SIAM Journal on Computing 36 (1) (2006) 132–157.

[4] P. Drineas, R. Kannan, M. W. Mahoney, Fast monte carlo algorithms for matrices II: Computing a low-rank approximation to a matrix, SIAM Journal on computing 36 (1) (2006) 158–183.

[5] P. Drineas, R. Kannan, M. W. Mahoney, Fast monte carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition, SIAM Journal on Computing 36 (1) (2006) 184–206.

[6] A. S. Georghiades, P. N. Belhumeur, D. J. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE transactions on pattern analysis and machine intelligence 23 (6) (2001) 643–660.

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

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

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

[10] A. Aldroubi, C. Cabrelli, U. Molter, Optimal non-linear models for sparsity and sampling, Journal of Fourier Analysis and Applications 14 (5) (2009) 793–812.

[11] P. Tseng, Nearest q-flat to m points, Journal of Optimization Theory and Applications 105 (1) (2000) 249–252.

[12] M. Fischler, R. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[13] N. Silva, J. Costeira, Subspace segmentation with outliers: a grassmannian approach to the maximum consensus subspace, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2008.

[14] R. Vidal, Y. Ma, S. Sastry, Generalized principal component analysis (GPCA), IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (12) (2005) 1945–1959.

[15] Y. Ma, A. Y. Yang, H. Derksen, R. Fossum, Estimation of subspace arrangements with applications in modeling and segmenting mixed data, SIAM Review 50 (1) (2008) 1–46.

[16] M. C. Tsakiris, R. Vidal, Filtrated spectral algebraic subspace clustering, SIAM Journal on Imaging Sciences 10 (1) (2017) 372–415.

[17] Y. C. Eldar, M. Mishali, Robust recovery of signals from a structured union of subspaces, IEEE Transactions on Information Theory 55 (11) (2009) 5302–5316.

[18] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2790–2797.

[19] E. Elhamifar, R. Vidal, Clustering disjoint subspaces via sparse representation, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2010.

[20] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, in: International Conference on Machine Learning, 2010, pp. 663–670.

[21] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2013) 171–184.

[22] U. V. Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (2007) 395–416.

[23] G. Chen, G. Lerman, Spectral curvature clustering (SCC), International Journal of Computer Vision 81 (2009) 317–330.

[24] F. Lauer, C. Schnorr, Spectral clustering of linear subspaces for motion segmentation, in: IEEE International Conference on Computer Vision, 2009.

[25] 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.

[26] A. Goh, R. Vidal, Segmenting motions of different types by unsupervised manifold clustering, in: Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, 2007, pp. 1 –6.

[27] G. Chen, G. Lerman, Foundations of a multi-way spectral clustering framework for hybrid linear modeling 9 (5) (2009) 517–558.

[28] R. Vidal, A tutorial on subspace clustering, IEEE Signal Processing Magazine 28 (2010) 52–68.

[29] A. Aldroubi, A. Sekmen, A. B. Koku, A. F. Cakmak, Similarity matrix framework for data from union of subspaces, Applied and Computational Harmonic Analysis, In Press.

[30] J. Costeira, T. Kanade, A multibody factorization method for independently moving objects, International Journal of Computer Vision 29 (3) (1998) 159–179.

[31] P. Ji, M. Salzmann, H. Li, Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4687–4695.

[32] G. H. Golub, C. F. Van Loan, Matrix computations, Vol. 3, JHU Press, 2012.

[33] B. Alexeev, J. Cahill, D. G. Mixon, Full spark frames, Journal of Fourier Analysis and Applications 18 (6) (2012) 1167–1194.

[34] D. L. Donoho, M. Elad, Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization, Proceedings of the National Academy of Sciences 100 (5) (2003) 2197–2202.

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

[36] J. Chiu, L. Demanet, Sublinear randomized algorithms for skeleton decompositions, SIAM Journal on Matrix Analysis and Applications 34 (3) (2013) 1361–1383.

[37] S. A. Goreinov, E. E. Tyrtyshnikov, N. L. Zamarashkin, A theory of pseudoskeleton approximations, Linear algebra and its applications 261 (1-3) (1997) 1–21.

[38] C. F. Caiafa, A. Cichocki, Generalizing the column–row matrix decomposition to multi-way arrays, Linear Algebra and its Applications 433 (3) (2010) 557–573.

[39] F. R. Gantmacher, Theory of Matrices. 2V., Chelsea publishing company, 1960.

[40] G. Stewart, Four algorithms for the the efficient computation of truncated pivoted qr approximations to a sparse matrix, Numerische Mathematik 83 (2) (1999) 313–323.

[41] M. W. Berry, S. A. Pulatova, G. Stewart, Algorithm 844: Computing sparse reduced-rank approximations to sparse matrices, ACM Transactions on Mathematical Software (TOMS) 31 (2) (2005) 252–269.

[42] S. Wang, Z. Zhang, Improving CUR matrix decomposition and the nyström approximation via adaptive sampling, The Journal of Machine Learning Research 14 (1) (2013) 2729–2769.

[43] P. Drineas, M. W. Mahoney, S. Muthukrishnan, Relative-error CUR matrix decompositions, SIAM Journal on Matrix Analysis and Applications 30 (2) (2008) 844–881.

[44] S. Voronin, P.-G. Martinsson, Efficient algorithms for cur and interpolative matrix decompositions, Advances in Computational Mathematics 43 (3) (2017) 495–516.

[45] S. Wang, Z. Zhang, T. Zhang, Towards more efficient SPSD matrix approximation and CUR matrix decomposition, Journal of Machine Learning Research 17 (210) (2016) 1–49.

[46] U. Oswal, S. Jain, K. S. Xu, B. Eriksson, Block CUR: Decomposing large distributed matrices, arXiv preprint arXiv:1703.06065.

[47] X. Li, Y. Pang, Deterministic column-based matrix decomposition, IEEE Transactions on Knowledge and Data Engineering 22 (1) (2010) 145–149.

[48] C.-W. Yip, M. W. Mahoney, A. S. Szalay, I. Csabai, T. Budavári, R. F. Wyse, L. Dobos, Objective identification of informative wavelength regions in galaxy spectra, The Astronomical Journal 147 (5) (2014) 110.

[49] J. Yang, O. Rubel, M. W. Mahoney, B. P. Bowen, Identifying important ions and positions in mass spectrometry imaging data using CUR matrix decompositions, Analytical chemistry 87 (9) (2015) 4658–4666.

[50] M. Xu, R. Jin, Z.-H. Zhou, CUR algorithm for partially observed matrices, in: International Conference on Machine Learning, 2015, pp. 1412–1421.

[51] S. Wei, Z. Lin, Analysis and improvement of low rank representation for subspace segmentation, arXiv.org.
URL http://arxiv.org/abs/1107.1561

[52] B. Recht, W. Xu, B. Hassibi, Null space conditions and thresholds for rank minimization, Mathematical programming 127 (1) (2011) 175–202.

[53] R. Tron, R. Vidal, A benchmark for the comparison of 3-d motion segmentation algorithms, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[54] 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.

[55] A. Aldroubi, A. Sekmen, Nearness to local subspace algorithm for subspace and motion segmentation, IEEE Signal Processing Letters 19 (10) (2012) 704–707.