ISIT'2013 1569696007

Nonlinear Approximations for Motion and Subspace Segmentation

Ali Sekmen
Dep. of Computer Science
Tennessee State University
Nashville, TN USA
Email: asekmen@tnstate.edu

Akram Aldroubi
Dep. of Mathematics
Vanderbilt University
Nashville, TN USA
Email: akram.aldroubi@vanderbilt.edu

***Abstract*—The motion segmentation problem is a special case of the general subspace segmentation problem that clusters data drawn from an unknown union of subspaces. This paper provides a nonlinear model for general subspace segmentation problem and presents an algorithm to compute the optimal solution for noiseless data. We also provide a combined algorithm that addresses issues with noise to some extent. Furthermore, a devised algorithm that specifically targets motion segmentation has been developed and applied to the Hopkins 155 Dataset. It generates the best segmentation rate to the date.**

I. Introduction

The motion segmentation problem can simply be defined as *identifying independently moving rigid objects in a video*. It can be formally stated as follows:

Problem 1. *Assume that F frames of a scene with k independently moving objects are given. Let $\{p_{i1}, \ldots, p_{iN}\}_{i=1}^{F}$ with $p_{ij} \in \mathbb{R}^3$ be the N feature points tracked across the F frames. Then,*

1) *determine the number of moving objects, k, and*
2) *determine clusters $\{\mathcal{C}_i\}_{i=1}^{k}$ of $\{p_{ij}\}_{i=1,j=1}^{i=F,j=N}$ so that $\mathcal{C}_i$ includes only the feature points that belong to the i^{th} moving object.*

The goal of subspace clustering is to identify all of the subspaces that a set of data $\mathbf{W} = \{w_1, ..., w_N\} \in \mathbb{R}^D$ is drawn from and assign each data point w_i to the subspace it belongs to. The number of subspaces, their dimensions, and a basis for each subspace are to be determined. The subspace segmentation problem can be stated as follows:

Problem 2. *Let $\mathcal{U} = \bigcup_{i=1}^{M} S_i$ where $\{S_i \subset \mathcal{B}\}_{i=1}^{M}$ is a set of subspaces of a Hilbert space or Banach space $\mathcal{B}$. Let $\boldsymbol{W} = \{w_j \in \mathcal{B}\}_{j=1}^{N}$ be a set of data points drawn from $\mathcal{U}$. Then,*

1) *determine the number of subspaces M,*
2) *find an orthonormal basis for each subspace S_i, and*
3) *group the data points belonging to the same subspace into the same cluster.*

The motion segmentation problem is a special case of the general subspace segmentation problem (see [1], [2], [3]). Note that often the data may be corrupted by noise, may have outliers or the data may not be complete, e.g., there may be missing data points. In some subspace clustering problems, the number M of subspaces or the dimensions of the subspaces $\{d_i\}_{i=1}^{M}$ are known. A number of approaches have been devised to solve the problem above or some of its special cases. They are based on sparsity methods [4], [5], [6], [7], algebraic methods [8], [9], iterative and statistical methods [10], [11], [12], [13], [14], [15], and spectral clustering methods [4], [5], [16], [17], [18], [19], [20], [21], [22].

A. Nonlinear Approximation

Let $\mathcal{B}$ be a Banach space, $\mathbf{W} = \{w_1, \ldots, w_m\}$ a finite set of vectors in $\mathcal{B}$. For $i = 1, \ldots, l$, let $\mathcal{C} = C_1 \times C_2 \times \ldots C_l$ be the cartesian product of l families C_i of closed subspaces of $\mathcal{B}$. Thus, an element $\mathbf{S} \in \mathcal{C}$ is a sequence $\{S_1, \ldots, S_l\}$ of l susbpaces of $\mathcal{B}$ with $S_i \in C_i$.

Problem 3. *Optimization Formulation for Subspace Segmentation*

1) *Given a finite set $\mathbf{W} \subset \mathcal{B}$, a fixed p with $0 < p \leq \infty$, and a fixed integer $l \geq 1$, find the infimum of the expression*

$$e(\mathbf{W}, \mathbf{S}) := \sum_{w \in \mathbf{W}} \min_{1 \leq j \leq l} d^p(w, S_j),$$

over $\mathbf{S} = \{S_1, \ldots, S_l\} \in \mathcal{C}$, and $d(x, y) := \|x - y\|_{\mathcal{B}}$.

2) *Find a sequence of l-subspaces $\mathbf{S}^o = \{S_1^o, \ldots, S_l^o\} \in \mathcal{C}$ (if it exists) such that*

$$e(\mathbf{W}, \mathbf{S}^o) = \inf\{e(\mathbf{W}, \mathbf{S}) : \mathbf{S} \in \mathcal{C}\}. \quad (1)$$

[3] shows that if each family of subspaces C_i satisfies the Minimum Subspace Approximation Property p-(MSAP), Problem 3 has a minimizer. This solution can be obtained by Algorithm 1. This algorithm will work well if a good initial partition is chosen. Otherwise, the algorithm may terminate in a local optima instead of the global optima.

II. Algebraic Approach - RREF

This section provides an algebraic approach to solve the general subspace segmentation problem (Problem 2) with the following assumptions: (1) data is noise-free, (2) data is generic, and (3) the subspaces are independent. Generic data is defined as follows:

Algorithm 1 Optimal Solution $\mathbf{S}^o$

Pick any partition $P \in \mathcal{P}(\mathbf{W})$
For each subset $\mathbf{W}_i$ in the partition P find the subspace $S_i^o(P) \in C_i$ that minimizes the expression $e(\mathbf{W}_i, S) = \sum_{w \in \mathbf{W}_i} d^p(w, S)$
while $\sum_{i=1}^{l} e(\mathbf{W}_i, S_i^o(P)) > e(\mathbf{W}, \mathbf{S}^o(P))$ **do**
  for all i from 1 to l **do**
    Update $\mathbf{W}_i = \{w \in \mathbf{W} : d(w, S_i^o(P)) \leq d(w, S_k^o(P)),\ k = 1, \ldots, l\}$
    Update $S_i^o(P) = \underset{S \in C_i}{\operatorname{argmin}}\, e(\mathbf{W}_i, S)$
  end for
  Update $P = \{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$
end while
$\mathbf{S}^o = \{S_1^o(P), \ldots, S_l^o(P)\}$

Definition 1. Let S be a linear subspace of $\mathbb{R}^D$ with dimension d. A set of data $\mathbf{W}$ drawn from $S \subset \mathbb{R}^D$ with dimension d is said to be *generic* if (i) $|\mathbf{W}| > d$, and (ii) every d vectors from $\mathbf{W}$ form a basis for S.

Definition 2. Matrix R is said to be the *binary reduced row echelon form* of matrix A if all non-pivot column vectors of the reduced row echelon form of A are converted to binary vectors, i.e., non-zero entries are set to one.

Theorem II.1. *Let $\{S_i\}_{i=1}^k$ be a set of non-trivial linearly independent subspaces of $\mathbb{R}^D$ with corresponding dimensions $\{d_i\}_{i=1}^k$. Let $\mathbf{W} = [w_1 \cdots w_N] \in \mathbb{R}^{D \times N}$ be a matrix whose columns are drawn from $\bigcup_{i=1}^k S_i$. Assume the data is drawn from each subspace and that it is generic. Let Brref($\mathbf{W}$) be the binary reduced row echelon form of $\mathbf{W}$. Then*

1) *The inner product (e_i, b_j) of a pivot column e_i and a non-pivot column b_j in Brref($\mathbf{W}$) is one, if and only if the corresponding column vectors $\{w_i, w_j\}$ in $\mathbf{W}$ belong to the same subspace S_l for some $l = 1, \ldots, k$.*
2) *Moreover, $\dim(S_l) = \|b_j\|_1$, where $\|b_j\|_1$ is the ℓ_1-norm of b_j.*
3) *Finally, $w_p \in S_l$ if and only if $b_p = b_j$ or $(b_p, b_j) = 1$.*

This theorem suggests a very simple, yet effective, approach to cluster the data points. The data $\mathbf{W}$ can be partitioned into k clusters $\{\mathbf{W}_1, \ldots, \mathbf{W}_k\}$, such that span$\mathbf{W}_l = S_l$. The clusters can be formed as follows: Pick a non-pivot element b_j in Brref($\mathbf{W}$), and group together all columns b_p in Brref($\mathbf{W}$) such that $(b_j, b_p) > 0$. Repeat the process with a different non-pivot column until all columns are exhausted.

Proof: The reduced row echelon form of $\mathbf{W}$ is of the form

$$\text{rref}(\mathbf{W}) = \begin{bmatrix} R \\ 0 \end{bmatrix}. \tag{2}$$

Let P be an $N \times N$ permutation matrix such that $\mathbf{W}P = \begin{bmatrix} U & V \end{bmatrix}$, where the columns of U are the columns associated with the pivots rref($\mathbf{W}$) and preserving their left to right order. Thus, U forms a basis for $\bigcup_{i=1}^k S_i$. This can be done, since the data is drawn from each subspace and it is generic, and that the $\{S_i\}$ are independent. In particular, U includes exactly d_i points from each S_i, and $U \subset \mathbb{R}^{D \times r}$ with rank $r = \sum_{i=1}^k d_i$. Moreover, because of the generic assumption of the data, $|V| \geq k$. In addition, every column of V is a linear combination of the columns of U, that is, there exists an $r \times (N - r)$ matrix Q with $V = UQ$. Therefore

$$\mathbf{W}P = \begin{bmatrix} U & V \end{bmatrix} = U \begin{bmatrix} I_r & Q \end{bmatrix}, \tag{3}$$

where I_r is $r \times r$ identity matrix. Let $E := E_l \cdots E_1$ be the product of elementary row operation matrices such that $E\mathbf{W}P = \text{rref}(\mathbf{W}P)$. Then,

$$E\mathbf{W}P = EU \begin{bmatrix} I_r & Q \end{bmatrix} = \begin{bmatrix} I_r & X \\ 0 & 0 \end{bmatrix}. \tag{4}$$

Thus $EU = \begin{bmatrix} I_r \\ 0 \end{bmatrix}$, and $X = Q$. By the choice of U above, we get that $\begin{bmatrix} I_r & Q \end{bmatrix} = RP$. It follows that, $\mathbf{W}P = U \begin{bmatrix} I_r & Q \end{bmatrix} = URP$, and since P is invertible, $\mathbf{W} = UR$.

$(e_i, b_j) = 1$ if and only if $(e_i, r_j) \neq 0$ where r_j is the column in R that corresponds to the column b_j in Brref($\mathbf{W}$). Now $r_j = \sum_{i=1}^{r} c_i e_i$. If $(e_i, r_j) \neq 0$, then $c_i \neq 0$. Thus $w_j = Ur_j = c_i w_i + \sum_{k \neq i} c_l U e_l$. If $w_i \in S_l$, then $w_i = Ue_i$ is one of the basis vectors of S_l, and since $c_i \neq 0$, independence of the subspaces implies that $w_j \in S_l$. Conversely, if $w_j = Ur_j$ and $w_i = Ue_i$ belong to the same subspace S_l, then $w_j = c_i w_i + \sum_{Ue_k \in S_l} c_k U e_k$, due to independence of the subspaces. This, together with the assumption that the data is generic implies that $c_i \neq 0$. Hence $r_j = c_i e_i + \sum_k c_k e_k$, and we get $(e_i, r_j) = c_i \neq 0$. This proves part (1).

Now let us assume that $w_j \in S_l$. Since the data is generic and subspaces are independent, w_j can be written as a linear combination of exactly d_l columns of U. This means there are d_l nonzero entries in the corresponding column r_j in R. Since all the nonzero entries are set to 1 for Brref($\mathbf{W}$), the ℓ_1-norm of the corresponding non-pivot columns must be d_l. This proves part (2).

Finally consider part (3). If w_p and w_j belong to S_l, then if $w_p = Ue_p$ then part (1) implies $(e_p, b_j) = 1$. Otherwise the fact the subspaces are independent and the data generic imply that $b_p = b_j$.

Now let b_p be a column of Brref($\mathbf{W}$) with $b_p = b_j$. Let r_p, r_j be the corresponding columns in R. Then, $w_p = Ur_p$ and $w_j = Ur_j$. Since $w_j \in S_l$, and w_p and w_j are in the span of the same column vectors of U corresponding to S_l, it follows, $w_p \in S_l$. Finally if $b_p \neq b_j$ and $(b_p, b_j) = 1$, then r_p is a pivot column of R. Part (1) then implies that $\{w_p, w_j\}$ belong to the same subspace S_l. ■

III. Improved Nonlinear Approximation

Algorithm 1 starts with a partition of the data matrix $\mathbf{W}$ and it may not find the global optimum solution if this initial partition is not good. Theorem II.1 works perfectly for *noiseless* data (it determines a basis for each subspace

and it correctly clusters all of the data points). However, it is likely to fail for noisy data due to difficulty of finding an appropriate threshold to set the small values of rref($\mathbf{W}$) to zero. rref($\mathbf{W}$) does not have the properties of those in Theorem II.1, in general, and therefore cannot be used to determine the subspaces, their dimensions, or the clusters. However, the thresholded reduced echelon form can be used to determine a set of clusters that can in turn be used to determine a good initial set of subspaces in Algorithm 1.

This is achieved as follows if the number of subspaces, l, is known and each subspace is d-dimensional: First, the reduced row echelon form rref($\mathbf{W}$) of $\mathbf{W}$ is computed. Since the data is noisy, the non-pivot columns of rref($\mathbf{W}$) will most likely have all non-zero entries. Since each subspace is d-dimensional, the highest d entries of each non-pivot column is set to 1 and the all other entries are set to 0 to determine the binary reduced row echelon form Brref($\mathbf{W}$) of $\mathbf{W}$ (note that, according to Theorem II.1, each non-pivot column of Brref($\mathbf{W}$) is supposed to have d entries). The next step is to have an l groups of the equivalent columns of Brref($\mathbf{W}$). Those l groups is then used as the initial partition for Algorithm 1. This process is described in Algorithm 2. Note that a dimensionality reduction is also performed to speed up the process as described in [3].

Algorithm 2 Combined Algorithm - Optimal Solution $\mathbf{S}^o$

Require: Normalized data matrix $\mathbf{W}$.

1: Set $r = l \times d$.
2: Compute the SVD of $\mathbf{W}$ and find $(V_r)^t$.
3: Replace the data matrix $\mathbf{W}$ with $(V_r)^t$.
4: Compute rref($\mathbf{W}$)
5: Compute Brref($\mathbf{W}$) by setting the highest d entries of each non-pivot column to 1 and all the others to 0.
6: Group the non-pivot equivalent columns of Brref($\mathbf{W}$) into l largest clusters $\{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$ and set the initial partition $P = \{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$.
7: For each subset $\mathbf{W}_i$ in the partition P find the subspace $S_i^o(P)$ that minimizes the expression $e(\mathbf{W}_i, S) = \sum_{w \in \mathbf{W}_i} d^p(w, S)$.
8: **while** $\sum_{i=1}^{l} e(\mathbf{W}_i, S_i^o(P)) > e(\mathbf{W}, \mathbf{S}^o(P))$ **do**
9: **for all** i from 1 to l **do**
10: Update $\mathbf{W}_i = \{w \in \mathbf{W} : d(w, S_i^o(P)) \leq d(w, S_k^o(P)),\ k = 1, \ldots, l\}$
11: Update $S_i^o(P) = \underset{S}{\operatorname{argmin}}\ e(\mathbf{W}_i, S)$
12: **end for**
13: Update $P = \{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$
14: **end while**
15: $\mathbf{S}^o = \{S_1^o(P), \ldots, S_l^o(P)\}$

In Step-7 of Algroithm 2,, we need to determine a subspace $S_i^o(P) \in C_i$ that minimizes the expression $e(\mathbf{W}_i, S) = \sum_{w \in \mathbf{W}_i} d^p(w, S)$. If the data is contaminated with a light-tailed noise distribution (such as Gaussian distributed noise), we set $p = 2$ and minimize $||\mathbf{W}_i - U_i V_i^t||_2$, where the columns of U_i form a basis for $S_i^o(P)$. It is known that SVD (which is an ℓ_2-based approach) can achieve this. However, SVD is not a very effective subspace matching approach if the data is contaminated with a heavy-tailed noise distribution (such as Laplacian distributed noise). In this case, a better way is to estimate a subspace that minimizes $e(\mathbf{W}_i, S) = \sum_{w \in \mathbf{W}_i} d(w, S)$ (p is set to 1). This is the ℓ_1-based approximation of $S_i^o(P)$, in which the minimization $||\mathbf{W}_i - U_i V_i^t||_1$ generally leads to a non-convex optimization problem. However, it can be recast as convex optimization with an iterative reformulation of the problem [23].

A. Experimental Results

Table I displays the results when Algorithm 2 (with SVD-based subspace approximation) is applied to the two-motion data from the Hopkins 155 Dataset [9]. The RREF-based algorithm is extremely fast and works well with two-motion video sequences. The average error for all two-motion sequences is 11.45%. However, the error is very high for three-motion sequences and obviously it does not work well with such video sequences. We believe that this is due to unknown nature of the noise in data.

IV. Approach for Motion Segmentation

This section discusses a devised algorithm (Algorithm 3) that clusters subspaces when the dimensions of subspaces are equal and known. For rigid motion segmentation, we assume that each subspace is 4-dimensional. First, a local subspace is estimated for each data point. Then, the distances between the local subspaces and points are computed and a distance matrix is generated. This is followed by construction of a binary similarity matrix by applying a data-driven threshold to the distance matrix. Finally, the segmentation problem is converted to a one-dimensional data clustering problem.

Let W be an $m \times N$ data matrix whose columns are drawn from a union of subspaces of dimensions at most d, possibly perturbed by noise. In order to reduce the dimensionality of the problem, we compute the SVD of W

$$W = U \Sigma V^t \tag{5}$$

where $U = \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix}$ is an $m \times m$ matrix, $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_N \end{bmatrix}$ is an $N \times N$ matrix, and Σ is an $m \times N$ diagonal matrix with diagonal entries $\sigma_1, \ldots, \sigma_l$, where $l = \min\{m, N\}$. To estimate the effective rank of W, one can use the modal selection algorithm [19], [24] to estimate the rank r if it is not known. We can now replace the data matrix W with the matrix $(V_r)^t$ that consists of the first r rows of V^t (thereby reducing the dimensionality of data). This step is justified by the following proposition which is used to validate that a data matrix W whose columns represent data points can be replaced with a lower rank matrix after computing its SVD (i.e. $W = U \Sigma V^t$):

Proposition IV.1. *Let A and B be $m \times n$ and $n \times k$ matrices. Let $C = AB$. Assume $J \subset \{1, 2, \cdots, k\}$.*

1) *If $b_i \in$ span $\{b_j : j \in J\}$ then $c_i \in$ span $\{c_j : j \in J\}$.*

Checker (78)	RREF-Based Approach
Average	8.81%
Median	5.44%
Traffic (31)	RREF-Based Approach
Average	16.04%
Median	11.94%
Articulated (11)	RREF-Based Approach
Average	17.25%
Median	12.69%
All (120 seq)	RREF-Based Approach
Average	11.45%
Median	6.78%

TABLE I
% SEGMENTATION ERRORS FOR SEQUENCES WITH TWO MOTIONS.

Average	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS
Checker-2 (78)	6.09%	2.57%	6.52%	4.46%	1.55%	0.83%	1.12%	0.23%
Traffic-2 (31)	1.41%	5.43%	2.55%	2.23%	1.59%	0.23%	0.02%	1.40%
Articulated-2 (11)	2.88%	4.10%	7.25%	7.23%	10.70%	1.63%	0.62%	1.77%
All-2 (120 seq)	4.59%	3.45%	5.56%	4.14%	2.40%	0.75%	0.82%	**0.57%**
Checker-3 (26)	31.95%	5.80%	25.78%	10.38%	5.20%	4.49%	2.97%	0.87%
Traffic-3 (7)	19.83%	25.07%	12.83%	1.80%	7.75%	0.61%	0.58%	1.86%
Articulated-3 (2)	16.85%	7.25%	21.38%	2.71%	21.08%	1.60%	1.60%	5.12%
All-3 (35 seq)	28.66%	9.73%	22.94%	8.23%	6.69%	3.55%	2.45%	**1.31%**
All (155 seq)	10.34%	4.94%	9.76%	5.03%	3.56%	1.45%	1.24%	**0.76%**

TABLE II
% SEGMENTATION ERRORS FOR SEQUENCES WITH TWO AND THREE MOTIONS.

2) *If A is full row rank (thus, $m \geq n$) then $b_i \in$ span $\{b_j : j \in J\} \Longleftrightarrow c_i \in$ span $\{c_j : j \in J\}$*

Proof: It is a straightforward application of linear algebra. ■

Another type of data reduction is normalization. Specifically, the columns of $(V_r)^t$ are normalized to lie on the unit sphere $\mathbb{S}^{r-1}$. This is because by projecting the subspace on the unit sphere, we effectively reduce the dimensionality of the data by one. Moreover, the normalization gives equal contribution of the data matrix columns to the description of the subspaces. The trajectory vectors that are close to each other are likely to belong to the same subspace. For this reason, we estimate a local subspace for each data point using its closest neighbors. Once we associate a local subspace S_i to each point x_i, the points and only those points that belong to the same subspace as x_i should have zero distance from S_i. This suggests computing the distance of each point x_j to the local subspace S_i and forming a distance matrix H. The distance matrix H is generated as $H = (d_{ij}) = \left(||x_j - A_i A_i^t x_j||_p + ||x_i - A_j A_j^t x_i||_p\right)/2$. Since we are not in the ideal case, a point x_j that belongs to the same subspace as x_i may have non-zero distance to S_i. However, this distance is likely to be small compared to the distance between x_j and S_k if x_j and x_k do not belong to the same subspace. This suggests that we compute a threshold that will distinguish between these two cases and transform the distance matrix into a binary matrix in which a zero in the (i, j) entry means x_i and x_j are likely to belong to the same subspace, whereas (i, j) entry of one means x_i and x_j are not likely to belong to the same subspace. To do this, we convert the distance matrix $H = (d_{ij})_{N \times N}$ into a binary similarity matrix $S = (s_{ij})$. This is done by applying a data-driven thresholding. The last step is to use the similarity matrix S to segment the data. To do this, we first normalize the rows of S using l_1-norm, i.e., $\tilde{S} = D^{-1}S$, where D is a diagonal matrix $(d_{ij}) = \sum_{j=1}^{N} s_{ij}$. Observe that the initial data segmentation problem has now been converted to segmentation of n 1-dimensional subspaces from the rows of $\tilde{S}$. This is because, in the ideal case, from the construction of $\tilde{S}$, if x_i and x_j are in the same subspace, the i^{th} and j^{th} rows of $\tilde{S}$ are equal.

A. Experimental Results

Table II displays some of the experimental results for the Hopkins 155 Dataset [9]. Our Nearness to Local Subspace (NLS) approach have been compared with: (1) GPCA, (2) RANSAC , (3) Local Subspace Affinity (LSA), (4) MLS, (5) Agglomerative Lossy Compression, and (6) Sparse Subspace Clustering (SSC). Table II used the number of neighbors $k = 3$. Since each point is drawn from a 4-dimensional subspace, a minimum of 3 neighbors are needed to fit a local subspace for each point. Using the same assumption as the algorithms that we compare with, we take the rank of the data matrix to be 8 for two motions and 12 for three motions. NLS has 0.76% misclassification rate compared to 1.24% of the next best algorithm.

Algorithm 3 Subspace Segmentation

Require: The $m \times N$ data matrix W whose columns are drawn from subspaces of dimension d

Ensure: Clustering of the feature points.

1: Compute the SVD of W.
2: Estimate the rank of W (denoted by r) if it is not known.
3: Compute $(V_r)^t$ consisting of the first r rows of V^t.
4: Normalize the columns of $(V_r)^t$.
5: Replace the data matrix W with $(V_r)^t$.
6: Find the angle between the column vectors of W and represent it as a matrix. {i.e., $\arccos(W^tW)$.}
7: Sort the angles and find the closest neighbors of column vector.
8: **for all** Column vector x_i of W **do**
9: Find the local subspace for the set consisting of x_i and k neighbors. {Theoretically, k is at least $d-1$. We can use the least square approximation for the subspace (see the section *Local Subspace Estimation*). Let A_i denote the matrix whose columns form an orthonormal bases for the local subspace associated with x_i.}
10: **end for**
11: **for** $i = 1$ to N **do**
12: **for** $j = 1$ to N **do**
13: define $H = (d_{ij}) = \left(||x_j - A_iA_i^tx_j||_p + ||x_i - A_jA_j^tx_i||_p\right)/2$
14: **end for**
15: **end for**{Build the distance matrix}
16: Sort the entries of the $N \times N$ matrix H from smallest to highest values into the vector h and set the threshold η to the value of the T^{th} entry of the sorted and normalized vector h, where T is such that $\|\chi_{[T,N^2]} - h\|_2$ is minimized, and where $\chi_{[T,N^2]}$ is the characteristic function of the discrete set $[T, N^2]$.
17: Construct a similarity matrix S by setting all entries of H less than threshold η to 1 and by setting all other entries to 0. {Build the binary similarity matrix}
18: Normalize the rows of S using l_1-norm.
19: Perform SVD $S^t = U_n\Sigma_n(V_n)^t$.
20: Cluster the columns of $\Sigma_n(V_n)^t$ using k-means. $\Sigma_n(V_n)^t$ is the projection on to the span of U_n.

V. Conclusion

This paper describes a method that solves the general subspace segmentation problem in the absence of noise. The theoretical limitations of this method in the presence of noise is given in [25]. We also present a devised algorithm that specifically targets segmentation of subspaces of equal and known dimensions, as in the motion segmentation problem.

Acknowledgment

The research of A. Sekmen is supported in part by NASA Grant NNX12AI14A and the research of A. Aldroubi is supported in part by NSF Grant DMS-1108631.

References

[1] K. Kanatani and C. Matsunaga, "Estimating the number of independent motions for multibody motion segmentation," in *5th Asian Conference on Computer Vision*, 2002, pp. 7–9.
[2] J. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 159–179, 1998.
[3] A. Sekmen, "Subspace segmentation and high Dimensional data analysis," Dissertation, Vanderbilt University, 2010.
[4] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2790–2797.
[5] ——, "Clustering disjoint subspaces via sparse representation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010.
[6] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *International Conference on Machine Learning*, 2010, pp. 663–670.
[7] M. Soltanolkotabi and E. J. Cands, "A geometric analysis of subspace clustering with outlierse," *Annals of Statistics*, 2011. [Online]. Available: http://arxiv.org/pdf/1112.4258v4.pdf
[8] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1945–1959, 12 2005.
[9] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
[10] K. Kanatani and Y. Sugaya, "Multi-stage optimization for multi-body motion segmentation," in *IEICE Trans. Inf. and Syst*, 2003, pp. 335–349.
[11] A. Aldroubi and K. Zaringhalam, "Nonlinear least squares in $\mathbb{R}^n$," *Acta Applicandae Mathematicae*, vol. 107, no. 1-3, pp. 325–337, 7 2009.
[12] A. Aldroubi, C. Cabrelli, and U. Molter, "Optimal non-linear models for sparsity and sampling," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 793–812, 12 2009.
[13] P. Tseng, "Nearest q-flat to m points," *Journal of Optimization Theory and Applications*, vol. 105, no. 1, pp. 249–252, 2000.
[14] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 6 1981.
[15] N. Silva and 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.
[16] U. V. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, 2007.
[17] G. Chen and G. Lerman, "Spectral curvature clustering (SCC)," *International Journal of Computer Vision*, vol. 81, pp. 317–330, 2009.
[18] F. Lauer and C. Schnorr, "Spectral clustering of linear subspaces for motion segmentation," in *IEEE International Conference on Computer Vision*, 2009.
[19] J. Yan and 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.
[20] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 6 2007, pp. 1 – 6.
[21] R. Vidal, "A tutorial on subspace clustering," *IEEE Signal Processing Magazine*, 2010.
[22] G. Chen and G. Lerman, "Foundations of a multi-way spectral clustering framework for hybrid linear modeling," vol. 9, no. 5, pp. 517–558, 2009.
[23] R. Marks, G. Wise, D. D. Haldeman, and J. L. Whited, "Detection in laplace noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-14, no. 6, pp. 866–872, 1978.
[24] L. Zappella, X. Lladó, E. Provenzi, and J. Salvi, "Enhanced local subspace affinity for feature-based motion segmentation," *Pattern Recognition*, vol. 44, pp. 454–470, 2011.
[25] A. Aldroubi and A. Sekmen, "Reduced row echelon form and non-linear approximation for subspace segmentation and high-dimensional data clustering," *Preprint*.