Subspace Clustering for Sequential Data

Stephen Tierney and Junbin Gao
School of Computing and Mathematics
Charles Sturt University
Bathurst, NSW 2795, Australia
{stierney, jbgao}@csu.edu.au

Yi Guo
Division of Computational Informatics
CSIRO
North Ryde, NSW 2113, Australia
yi.guo@csiro.au

Abstract

We propose Ordered Subspace Clustering (OSC) to segment data drawn from a sequentially ordered union of subspaces. Current subspace clustering techniques learn the relationships within a set of data and then use a separate clustering algorithm such as NCut for final segmentation. In contrast our technique, under certain conditions, is capable of segmenting clusters intrinsically without providing the number of clusters as a parameter. Similar to Sparse Subspace Clustering (SSC) we formulate the problem as one of finding a sparse representation but include a new penalty term to take care of sequential data. We test our method on data drawn from infrared hyper spectral data, video sequences and face images. Our experiments show that our method, OSC, outperforms the state of the art methods: Spatial Subspace Clustering (SpatSC), Low-Rank Representation (LRR) and SSC.

1. Introduction

In many applications such as machine learning and image processing high dimensional data are ubiquitous. This high dimensionality has adverse affects on the computation time and memory requirements of algorithms that want to extract information. Fortunately, it has been shown that high dimensional data often lie in a small number of much lower dimensional subspaces [1]. In most cases the assumption is that data lies in a union of subspaces. The goal of subspace clustering is to cluster the data according to their residing subspaces.

There are many tasks in machine learning that can be represented by the union of subspaces model. For example extracting feature trajectories of a rigidly moving object in video [2] and identifying face images of a subject under varying illumination [3]. In this work we apply the union of subspaces model to high dimensional sequential data. We assume that this data is sampled at specific points in time or space in uniform intervals. This data is also known as time series data. For example hyper spectral drill core data [1] is obtained by sampling the infrared reflectance along the length of the core. The mineralogy is typically stratified meaning segments of mineral compounds congregate together [4]. Another example is video data which as a function of time has a sequential structure [5] where we can assume most frames are similar to their neighbours until a scene change.

In this work we exploit the the sequential nature of sequential data by incorporating a neighbour penalty term into our model to enforce similarity. We show that the new model formulation improves subspace clustering results for sequential data and describe a method to extract the clusters without knowing the number of clusters a priori. Through experimental evaluation we show that our algorithm OSC outperforms state-of-the-art subspace clustering methods on real-world problems of video scene segmentation and face clustering (see Figures 1 and 2 respectively).

2. Preliminaries

Consider a matrix of column wise samples $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_N}] \in \mathbb{R}^{D \times N}$. Each sample or datum can be represented by a linear combination of some atoms in a dictionary $\mathbf{A} = [\mathbf{a_1}, \mathbf{a_2}, \dots, \mathbf{a_n}] \in \mathbb{R}^{D \times n}$:

$$\mathbf{X} = \mathbf{AZ} \tag{1}$$

where $\mathbf{Z} = [z_1, z_2, \dots, z_N] \in \mathbb{R}^{n \times N}$ is a coefficient matrix. Under this representation, each column $\mathbf{z}_i$ of the coefficient matrix $\mathbf{Z}$ can be interpreted as a new representation for each data sample $\mathbf{x}_i$ under the dictionary.

Let $\mathbf{X}$ be a set of data vectors drawn from a union of k subspaces $\{S_i\}_{i=1}^k$ of unknown dimensions $\{d_i\}_{i=1}^k$. Without knowing the dictionary $\mathbf{A}$ one can use the following self-expressiveness property of data [1] to find the subspaces:

> *each data point in a union of subspaces can be efficiently reconstructed by a combination of other points in the data*

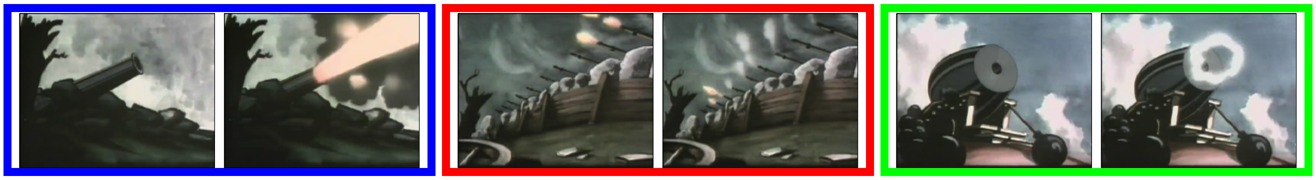


Figure 1: Video scene segmentation: given a sequence of video frames the goal is to cluster the frames that belong to the same scene. Clusters (scenes) highlighted by coloured borders.

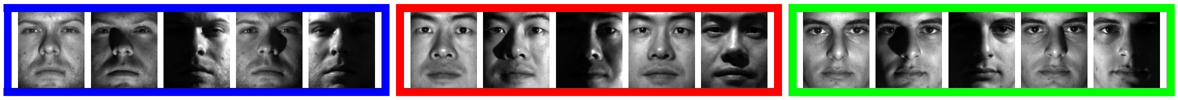


Figure 2: Face clustering: given an ordered set of face images the goal is to cluster images that belong to the same subject.

Both sparse subspace clustering (SSC) [1] and low-rank representation (LRR) [6] take this strategy by using the data samples themselves as the dictionary. That is, $\mathbf{A} = \mathbf{X}$. In this case, the coefficient matrix $\mathbf{Z}$ becomes a square matrix of size $N \times N$. In fact, we have the secondary interpretation for $\mathbf{Z}$ for which the element z_{ij} is the similarity of data points $\mathbf{x}_i$ to $\mathbf{x}_j$, because

$$x_i = \mathbf{X} z_i. \tag{2}$$

In other words each point can be written as a linear combination of other points. In general $N > D$, in this unrestricted case there are near infinite possibilities for the coefficient matrix $\mathbf{Z}$. The choice of $\mathbf{Z}$ is the main point of difference among subspace clustering techniques, which we discuss in section 3.

3. Related Work

Little prior work exists on the matter of subspace clustering on time series data. As such we provide a brief overview of the recent developments in field of subspace clustering. We refer readers to more comprehensive reviews of this field found in [7] and [1].

In recent years compressed sensing techniques have been applied to subspace clustering. Sparse Subspace Clustering (SSC or L1-Graph) by Elhamifar & Vidal [8, 1] aims to find the sparsest representation using ℓ_1 approximation. More specifically every point in the data as a set of sparse linear combinations of other points from the same subspace. Mathematically we write this sparse formulation as

$$\begin{aligned} \min_{\mathbf{E},\mathbf{S},\mathbf{Z}} \ & \frac{\lambda_1}{2}\|\mathbf{E}\|_F^2 + \lambda_2\|\mathbf{S}\|_1 + \|\mathbf{Z}\|_1 \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{XZ} + \mathbf{E} + \mathbf{S}, \operatorname{diag}(\mathbf{Z}) = \mathbf{0} \end{aligned} \tag{3}$$

where $\mathbf{E}$ is Gaussian noise and $\mathbf{S}$ is high magnitude sparse noise. From these sparse representations an affinity matrix $\mathbf{Z}$ is compiled. This affinity matrix is interpreted as a a graph upon which a clustering algorithm such as Normalized Cuts (NCut) [9] is applied for the final segmentation. This is the typical approach of modern subspace clustering techniques.

Rather than compute the sparsest representation of each data point individually Low-Rank Representation (LRR) by Liu et al. [6] attempts to incorporate global structure of the data by computing the lowest-rank representation of a set of data points. This low rank representation is achieved by approximating rank with the nuclear norm as follows

$$\begin{aligned} \min_{\mathbf{E},\mathbf{Z}} \ & \lambda\|\mathbf{E}\|_{1,2} + \|\mathbf{Z}\|_* \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{XZ} + \mathbf{E} \end{aligned} \tag{4}$$

where $\|\mathbf{E}\|_{1,2} = \sum_{i=1}^{n} \|\mathbf{e_i}\|_2$ is called the $\ell_{1,2}$ norm and penalises column specific error, meaning some columns of the reconstruction error $\mathbf{E}$ can be large but most will be small. In many cases LRR can outperform SSC but its computational complexity is higher [7, 10] and it is unable to be separated and computed in parallel.

Of most relation to our work is Spatial Subspace Clustering (SpatSC) from Guo et al. [4], which extends SSC by incorporating an extra penalty term to model the sequential ordering of data. The penalty term indirectly forces similarity for neighbouring data points in the affinity matrix $\mathbf{Z}$. In contrast our method OSC takes a more direct approach, which enforces sequential similarity in a more holistic manner. We provide further comparison in section 4.

4. Our Contribution

Most subspace clustering techniques do not take into account the information that is implicitly encoded into ordered data. That is that data samples or columns are either neighbours in a space or time domain. One example of time ordered data is video, each frame can be vectorised into a single column $\mathbf{x}_i$ and placed side by side with neighbouring frames to form $\mathbf{X}$. This sequential ordering should be reflected in the coefficients of $\mathbf{Z}$ so that neighbours are similar i.e. $\mathbf{z}_i \approx \mathbf{z}_{i+1}$.

In [4] the following spatial subspace clustering (SpatSC) algorithm was proposed

$$\begin{aligned}\min_{\mathbf{Z},\mathbf{E}} \frac{1}{2}\|\mathbf{E}|_F^2 + \lambda_1\|\mathbf{Z}\|_1 + \lambda_2\|\mathbf{ZR}\|_1 \\ \text{s.t.} \quad \mathbf{X} = \mathbf{XZ} + \mathbf{E}, \text{diag}(\mathbf{Z}) = \mathbf{0}\end{aligned} \tag{5}$$

where $\mathbf{R}$ is a lower triangular matrix with -1 on the diagonal and 1 on the second diagonal

$$\mathbf{R} \in \mathbb{Z}^{N\times N-1} = \begin{bmatrix} -1 & & & & \\ 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix}.$$

Therefore $\mathbf{ZR} = [\mathbf{z}_2 - \mathbf{z}_1, \mathbf{z}_3 - \mathbf{z}_2, ..., \mathbf{z}_N - \mathbf{z}_{N-1}]$. The aim of this formulation is to force consecutive columns of $\mathbf{Z}$ to be similar. The third penalty term introduced in (5) is for this purpose. Unfortunately $\|\mathbf{ZR}\|_1$ only imposes sparsity at the element level in the column differences $\mathbf{z}_i - \mathbf{z}_{i-1}$ and does not directly penalise whole column similarity. In effect this allows some values in consecutive columns to be greatly different. On the contrary we wish to directly penalise the similarity between consecutive columns and maintain sparsity. To achieve this goal we replace the ℓ_1 norm of the third term with the $\ell_{1,2}$ norm. That is, we will consider the following problem instead

$$\begin{aligned}\min_{\mathbf{Z},\mathbf{E}} \frac{1}{2}\|\mathbf{E}\|_F^2 + \lambda_1\|\mathbf{Z}\|_1 + \lambda_2\|\mathbf{ZR}\|_{1,2} \\ \text{s.t.} \quad \mathbf{X} = \mathbf{XZ} + \mathbf{E}\end{aligned} \tag{6}$$

Thus in (6), we replace the weaker penalty $\|\mathbf{ZR}\|_1$ with the stronger penalty $\|\mathbf{ZR}\|_{1,2}$ to strictly enforce column similarity. In summary the contributions of this work include:

1. We introduce the $\ell_{1,2}$ norm over $\mathbf{ZR}$ to enforce column similarity, which outperforms state of the art methods on sequential data.

2. We propose a new segmentation algorithm that exploits information encoded in our mostly block diagonal $\mathbf{Z}$ so that the number of clusters is no longer a required parameter.

5. Solving the Objective Function

To solve (5), [4] uses the property of separation of the ℓ_1-norm and adopts a row-wise formation to decouple the columns in the third term. However the similar trick cannot be used for problem (6) because of the new $\ell_{1,2}$ norm over $\mathbf{ZR}$. Instead we will use the alternating direction method of multipliers (ADMM) [11] to find a solution. As discussed in [12] we cannot guarantee convergence when using this approach. However in our experiments the algorithm always converged. First we remove the variable $\mathbf{E}$ by using the constraint and thus the objective (6) can be re-written as follows,

$$\min_{\mathbf{Z}} \frac{1}{2}\|\mathbf{X} - \mathbf{XZ}\|_F^2 + \lambda_1\|\mathbf{Z}\|_1 + \lambda_2\|\mathbf{ZR}\|_{1,2} \tag{7}$$

To further separate the terms of variable $\mathbf{Z}$, let $\mathbf{S} = \mathbf{Z}$ and $\mathbf{U} = \mathbf{SR}$, then the Augmented Lagrangian for the two introduced constraints is

$$\begin{aligned}\mathcal{L}(\mathbf{Z},\mathbf{S},\mathbf{U}) =& \frac{1}{2}\|\mathbf{X} - \mathbf{XS}\|_F^2 + \lambda_1\|\mathbf{Z}\|_1 + \lambda_2\|\mathbf{U}\|_{1,2} \\ &+ \langle \mathbf{G}, \mathbf{Z} - \mathbf{S}\rangle + \frac{\gamma_1}{2}\|\mathbf{Z} - \mathbf{S}\|_F^2 \\ &+ \langle \mathbf{F}, \mathbf{U} - \mathbf{SR}\rangle + \frac{\gamma_2}{2}\|\mathbf{U} - \mathbf{SR}\|_F^2\end{aligned} \tag{8}$$

We can solve (8) for $\mathbf{Z}, \mathbf{S}, \mathbf{U}$ in an alternative manner when fixing the others, respectively.

1. Fixing $\mathbf{S}$ and $\mathbf{U}$, solve for $\mathbf{Z}$ by

$$\min_{\mathbf{Z}} \lambda_1\|\mathbf{Z}\|_1 + \langle \mathbf{G}, \mathbf{Z} - \mathbf{S}\rangle + \frac{\gamma_1}{2}\|\mathbf{Z} - \mathbf{S}\|_F^2$$

which is equivalent to

$$\min_{\mathbf{Z}} \lambda_1\|\mathbf{Z}\|_1 + \frac{\gamma_1}{2}\|\mathbf{Z} - (\mathbf{S} - \frac{\mathbf{G}}{\gamma_1})\|_F^2 \tag{9}$$

Problem (9) is separable at element level and each has a closed-form solution defined by the soft thresholding operator as follows, see [13, 14],

$$Z = \text{sign}\left(\mathbf{S} - \frac{\mathbf{G}}{\gamma_1}\right)\max\left(\left|\mathbf{S} - \frac{\mathbf{G}}{\gamma_1}\right| - \frac{\lambda_1}{\gamma_1}\right). \tag{10}$$

2. Fixing $\mathbf{Z}$ and $\mathbf{U}$, solve for $\mathbf{S}$ by

$$\begin{aligned}\min_{\mathbf{S}} \frac{1}{2}\|\mathbf{X} - \mathbf{XS}\|_F^2 + \langle \mathbf{G}, \mathbf{Z} - \mathbf{S}\rangle + \frac{\gamma_1}{2}\|\mathbf{Z} - \mathbf{S}\|_F^2 \\ + \langle \mathbf{F}, \mathbf{U} - \mathbf{SR}\rangle + \frac{\gamma_2}{2}\|\mathbf{U} - \mathbf{SR}\|_F^2\end{aligned}$$

Setting the derivative of the above objective with respect to $\mathbf{S}$ to zero gives

$$\begin{aligned}&(\mathbf{X}^T\mathbf{X} + \gamma_1 I)\mathbf{S} + \gamma_2\mathbf{SRR}^T \\ =&\mathbf{X}^T\mathbf{X} + \gamma_2\mathbf{UR}^T + \gamma_1\mathbf{Z} + \mathbf{G} + \mathbf{FR}^T\end{aligned}$$

We can vectorize the above linear matrix equation into

$$\begin{aligned}&[I \otimes (\mathbf{X}^T\mathbf{X} + \gamma_1 I) + \gamma_2\mathbf{RR}^T \otimes I]\text{vec}(\mathbf{S}) \\ =& \text{vec}(\mathbf{X}^T\mathbf{X} + \gamma_2\mathbf{UR}^T + \gamma_1\mathbf{Z} + \mathbf{G} + \mathbf{FR}^T)\end{aligned}$$

where $\otimes$ is the tensor product.

3. Fixing $\mathbf{Z}$ and $\mathbf{S}$, solve for $\mathbf{U}$ by

$$\min_{\mathbf{U}} \lambda_2 \|\mathbf{U}\|_{1,2} + \langle \mathbf{F}, \mathbf{U} - \mathbf{SR} \rangle + \frac{\gamma_2}{2} \|\mathbf{U} - \mathbf{SR}\|_F^2$$

which is equivalent to

$$\min_{\mathbf{U}} \lambda_2 \|\mathbf{U}\|_{1,2} + \frac{\gamma_2}{2} ||\mathbf{U} - (\mathbf{SR} - \frac{1}{\gamma_2}\mathbf{F})||_F^2$$

Denote by $\mathbf{M} = \mathbf{SR} - \frac{1}{\gamma_2}\mathbf{F}$, then the above problem has a closed-form solution defined as follows,

$$\mathbf{U}(:,i) = \begin{cases} \frac{\|\mathbf{M}(:,i)\| - \frac{\lambda_2}{\gamma_2}}{\|\mathbf{M}(:,i)\|} \mathbf{M}(:,i) & \text{if } \|\mathbf{M}(:,i)\| > \frac{\lambda_2}{\gamma_2} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where $\mathbf{U}(:,i)$ and $\mathbf{M}(:,i)$ are the i-th columns of $\mathbf{U}$ and $\mathbf{M}$, respectively. Please refer to [6].

4. Update $\mathbf{G}$ by

$$\mathbf{G} = \mathbf{G}^{old} + \gamma_1(\mathbf{Z} - \mathbf{S})$$

5. Update $\mathbf{F}$ by

$$\mathbf{F} = \mathbf{F}^{old} + \gamma_2(\mathbf{U} - \mathbf{SR})$$

6. Also update γ_1 and γ_2 by

$$\gamma_1 = \rho\gamma_1^{old}; \;\; \gamma_2 = \rho\gamma_2^{old}$$

6. Segmentation

Once a solution to (6) has been found the next step is to segment the coefficient matrix $\mathbf{Z}$ to find the subspace clusters. We discuss methods for segmentation depending on amount and type of prior knowledge about the original data. Unlike prior subspace clustering methods we do not always require the number of clusters k to be known before hand.

1. If we assume that the data is drawn from a set of disconnected subspaces i.e. $\mathbf{Z}$ is block diagonal we can use information encoded by $\mathbf{ZR}$ to find the cluster boundaries. Ideally columns of $\mathbf{ZR}$, i.e. $\mathbf{z}_i - \mathbf{z}_{i-1}$, that are within a segment should be the zero vector or very close to it because columns from the same subspace share similarity. Columns of $\mathbf{ZR}$ that greatly deviate away from the zero vector indicate the boundary of a segment as the similarity is low. First let $\mathbf{B} = (|\mathbf{ZR}_{ij}|)$ be the absolute value matrix of $\mathbf{ZR}$. Then let $\mu^{\mathbf{B}}$ be the vector of column-wise means of $\mathbf{B}$. Then we employ a peak finding algorithm over $\mu^{\mathbf{B}}$ to find the segment boundaries. We call this method "intrinsic segmentation".

2. Alternatively if $\mathbf{Z}$ is block diagonal and noiseless we can analyse the eigenspectrum of $\mathbf{Z}$ to find the number and size of each cluster [7]. Using the eigengap heuristic we find a set of explanatory eigenvalues, the number of eigenvalues indicates the number of clusters and the magnitude of each indicates the cluster size. If $\mathbf{Z}$ contains noise then the eigengap heuristic fails to provide accurate cluster size but the number of clusters will still be accurate [10].

3. If the number of clusters is known beforehand or estimated via eigenspectrum analysis we suggest using Ncut [9] to segment the data. Ncut has been shown to be robust in subspace segmentation tasks and is considered state of the art [1, 6]. In cases where $\mathbf{Z}$ is not block diagonal or contains significant noise NCut will provide better segmentation than previous methods.

7. Experimental Results and Applications

In this section we first evaluate the performance of OSC on a synthetic experiment using data from real hyper spectral mineral data. We then evaluate OSC on real world data with video scene segmentation and face clustering experiments. Parameters were fixed for each experiment. We used parameters suggested by original authors of competing methods where applicable (e.g. face clustering) and tuned parameters for best performance in other experiments. In order to evaluate performance consistently we used NCut for final segmentation for every method in every experiment.

We use the subspace clustering error metric from [1] to compare results. The subspace clustering error (SCE) is as follows

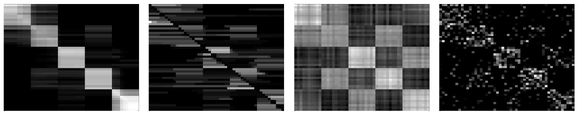
$$\text{SCE} = \frac{\text{num. misclassified points}}{\text{total num. of points}} \tag{12}$$

In contrast to other works [1, 6] we provide minimum, maximum, median and mean data on clustering error for each experiment. It is important to consider these ranges holistically when evaluating these methods.

Additionally we evaluate performance when extra noise is added to each dataset, which to the best of our knowledge has been avoided by others. In all experiments we used Guassian noise with zero mean and unit variance. We modify the noise by varying magnitudes and provide results for each magnitude of noise. We report the level of noise using Peak Signal-to-Noise Ratio (PSNR) which is defined as

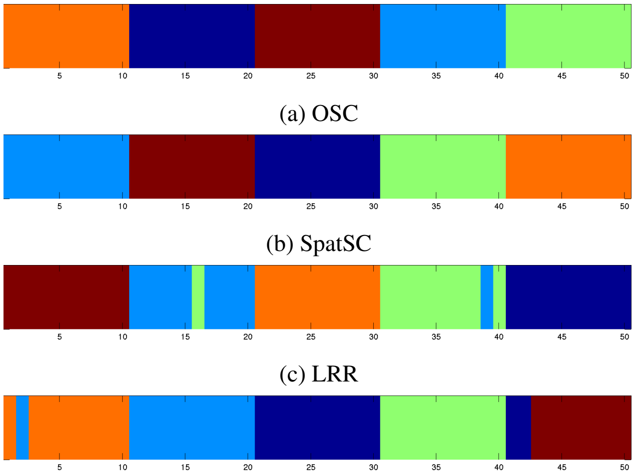
$$\text{PSNR} = 10 \log_{10} \left(\frac{s^2}{\frac{1}{mn} \sum_i^m \sum_j^n (\mathbf{I}_{ij} - \mathbf{K}_{ij})^2} \right) \tag{13}$$

where $\mathbf{I}$ is the noise free data, $\mathbf{K}$ is a noisy approximation and s is the maximum possible value of an element of $\mathbf{I}$.



(a) OSC (b) SpatSC (c) LRR (d) SSC

Figure 3: Examples of affinity matrices $\mathbf{Z}$ from the synthetic experiment using hyper spectral mineral data.



(a) OSC

(b) SpatSC

(c) LRR

(d) SSC

Figure 4: Clustering results from affinity matrices in Figure 3. OSC and SpatSC achieve perfect segmentation, while LRR and SSC suffer from misclassification.

Decreasing values of PSNR indicate increasing amounts of noise. Since the denominator of (13) will be 0 in noise free cases we mark the PSNR as “Max”. PSNR values reported are rounded averages.

7.1. Synthetic Experiment

We assemble synthetic data from a library of pure infrared hyper spectral mineral data. We randomly take 5 pure spectra samples from the library such that $\mathbf{A}_i = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_5] \in \mathbb{R}^{321\times 5}$. Next we combine these samples into a single synthetic sample using uniform random weights $\mathbf{w}_i \in \mathbb{R}^5$ such that $\mathbf{x}_i \in \mathbb{R}^{321} = \mathbf{A}_i\mathbf{w}_i$. We then repeat $\mathbf{x}_i$ 10 times column-wise giving us $\mathbf{X}_i \in \mathbb{R}^{321\times 10}$. We repeat this process 5 times and combine all $\mathbf{X}_i$ to create our artificial data $\mathbf{X} \in \mathbb{R}^{321\times 50} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_5]$. The aim is to correctly segment the 5 synthetic spectra.

We then corrupt data with various levels of Gaussian noise and evaluate clustering performance of OSC against SpatSC, LRR and SSC. The experiment is repeated for 50 variations of $\mathbf{X}$. In the noise free case SSC performed marginally better than OSC. At all other levels of noise OSC outperforms every competitor. This experiment highlights OSC’s robustness to noise. Since parameters were the same across all noise magnitudes the experiment demonstrated that OSC is less sensitive to parameter values than other methods. Results can be found in Table 1.

PSNR		OSC	SpatSC	LRR	SSC
Max	Min	0% (49)	0% (43)	0% (12)	**0% (50)**
	Max	22%	22%	56%	**0%**
	Med	**0%**	**0%**	24%	**0%**
	Mean	0.44%	1.16%	23.68%	**0%**
46	Min	**0% (49)**	0% (31)	0% (27)	0% (40)
	Max	**6%**	28%	42%	30%
	Med	**0%**	0%	0%	0%
	Mean	**0.12%**	3.6%	8.8%	1.52%
24	Min	**0% (26)**	0% (1)	4%	16%
	Max	**20%**	46%	52%	62%
	Med	**0%**	10%	30%	40%
	Mean	**2.64%**	15.88%	29.72%	41.48%
10	Min	**0% (1)**	14%	38%	50%
	Max	**24%**	62%	70%	70%
	Med	**12%**	42%	70%	70%
	Mean	**12%**	42%	58%	62%

Table 1: Misclassification results for the synthetic hyper spectral mineral data set with various magnitudes of Gaussian noise, lower is better. Numbers in brackets indicate how many times clustering was perfect, i.e. zero error.

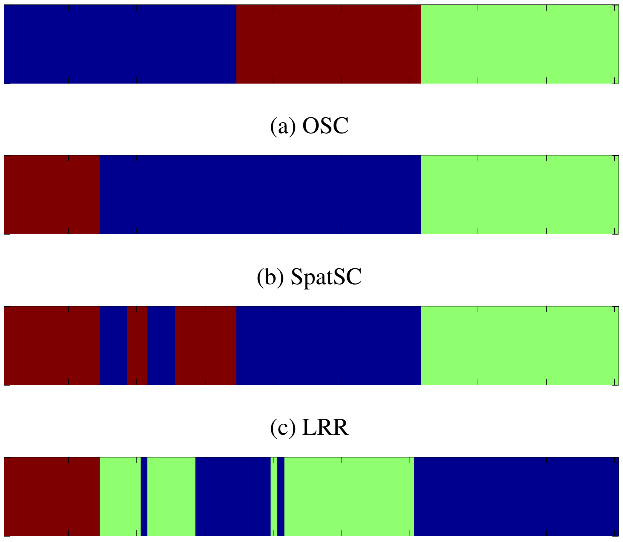
We provide a visual comparison of affinity matrices $\mathbf{Z}$ for this experiment in Figure 3. In this experiment we added Gaussian noise at 20% magnitude. Visually we observe that OSC provides affinity matrices which are more block diagonal and contain stronger and more numerous within block weights than other methods. LRR produces the next best affinity matrices, in terms of being block diagonal, but lacks sparsity which allows for easier segmentation and more efficient storage. Visual results of clustering accuracy for these affinity matrices is provided in Figure 4.

7.2. Video Scene Segmentation

The aim of this experiment is to segment individual scenes from a video sequence. The video sequences are drawn from two short animations freely available from the Internet Archive[1]. See Figure 1 for an example of a sequence to be segmented. The sequences are around 10 seconds in length (approximately 300 frames) containing three scenes each. There are 19 and 24 sequences from videos 1 and 2 respectively. The scenes to be segmented can contain significant translation and morphing of objects within the scene and sometimes camera or perspective changes. Scene changes (or keyframes) were collected manually to form ground truth data.

The pre processing of a sequence consisted of converting

[1] http://archive.org/



(a) OSC

(b) SpatSC

(c) LRR

(d) SSC

Figure 5: Clustering results from the video scene segmentation experiment. OSC achieves perfect segmentation while SpatSC, LRR and SSC suffer from significant misclassification.

colour video to grayscale and down sampling to a resolution of 129 × 96. Each frame in the sequence was vectorised to $\mathbf{x}_i \in \mathbb{R}^{12384}$ and concatenated with consecutive frames to form $\mathbf{X} \in \mathbb{R}^{12384 \times 300}$.

The video sequences were then corrupted with various magnitudes of Gaussian noise to evaluate clustering performance. Results can be found in Table 2. Generally OSC outperforms other methods and the error rates are consistently low when compared to other methods which greatly increase as the magnitude of the noise is increased. In Figure 5 we provide a visual example of OSC's robustness where we set the magnitude of noise at 30%. A sample of frames from this sequence without noise can be found in Figure 1

We performed a second set of experiments with the same video sequences as before, however we interpolated the first and last six frames of scenes boundaries to form a fading transition. We provide an illustration of this in Figure 6. We observed that as before OSC generally outperforms the other tested methods and is most robust to noise. Results can be found in Table 3.

7.3. Face Clustering

The aim of this experiment is to segment or cluster unique subjects from a set of face images. Although this task is not exactly suited to our assumption of clustering sequential data we ensure that in our tests the faces are kept contiguous i.e. unique subjects do not mix. We can exploit the spatial information since we know neighbours of each data vector are likely to belong to the same subject. We draw our data from the Exteded Yale Face Database B [15]. The dataset consists of approximately 64 photos of 38 subjects under varying illumination. See Figure 2 for an example. For each test we randomly pick 3 subjects from the dataset then randomly order the images within each subject's set. The images are resampled to 42×48 to form data vectors $\mathbf{x}_i \in \mathbb{R}^{2016}$ and concatenated together in order to ensure that subjects do not mix.

We repeated these tests 50 times for each level of corruption by Gaussian noise. Since the original data is already corrupted by shadows (see Figure 2) the maximum magnitude of extra noise that we apply is lower than previous experiments. Additionally we impose an additional constraint $\text{diag}(\mathbf{Z}) = \mathbf{0}$ on OSC (6) to avoid the trivial identity solution.

Results can be found in Table 4. We observed that OSC outperforms all other methods in most cases. The power of OSC is most noticeable with larger magnitudes of noise. There was only a negligible difference between LRR and SSC. We observed that SpatSC performed incosistently in this experiment, for example the mean error rate is lower at 10% magnitude noise than without any noise. In contrast our method OSC was more stable and behaved more predictably.

8. Conclusion and Future Work

We have presented and evaluated a novel subspace clustering method, Ordered Subspace Clustering, that exploits the ordered nature of data. OSC produces more interpretable and accurate affinity matrices than other methods. In ideal cases it is able to provide clustering without knowing the number of clusters, which other methods are not capable of. We showed that this method generally outperforms existing state of the art methods in quantitive accuracy, particularly when the data is heavily corrupted with noise.

While OSC outperforms other methods there are areas of improvement remaining:

- It has been shown that LRR is much better than SSC at capturing the global structure of data. As such we wish to replace our core structure of sparse representation with low-rank representation.

- As the computation of the $\ell_{1,2}$ penalty is expensive it would be better to have a faster alternative.

- The **ZR** structure limits us to sequentially structured or 1D data. We hope to develop a new penalty which can be defined by the user to suit other geometric structures such as images, which are 2D.

		Video 1				Video 2			
PSNR		OSC	SpatSC	LRR	SSC	OSC	SpatSC	LRR	SSC
Max	Min	**0% (12)**	0% (11)	0% (4)	0% (5)	**0% (3)**	**0% (3)**	0.69%	**0% (3)**
	Max	**28.4%**	49.33%	65.6%	44.4%	**2.96%**	28.29%	45.37%	44.39%
	Med	**0%**	0%	19.9%	24.5%	0.96%	**0.92%**	2.72%	1.11%
	Mean	**6.58%**	8.38%	21.29%	20.56%	**1.02%**	2.11%	13.42%	10.59%
46	Min	0% (12)	**0% (14)**	0% (5)	0% (5)	0% (3)	**0% (3)**	0% (3)	0% (3)
	Max	**32.1%**	40.07%	44%	45.89%	**2.96%**	28.29%	35.6%	43.9%
	Med	**0%**	0%	12.5%	21.54%	0.96%	**0.92%**	1.01%	1.11%
	Mean	6.54%	**5.74%**	18.21%	21.5%	**1.02%**	2.11%	5.78%	10.79%
24	Min	**0% (14)**	0% (11)	0% (6)	0% (3)	0% (3)	0% (3)	0% (3)	0% (3)
	Max	**26.24%**	50.62%	49.67%	59.33%	**2.96%**	28.29%	28.29%	100%
	Med	**0%**	0%	8.89%	31.82%	0.96%	**0.92%**	0.96%	1.2%
	Mean	**2.22%**	9.63%	14.05%	27.4%	**1.08%**	2.68%	2.89%	13.78%
10	Min	**0% (11)**	0% (4)	0% (6)	3.03% (0)	**0% (1)**	**0% (1)**	0% (3)	1.14%
	Max	**24.11%**	53.45%	48%	100%	**11.6%**	32.07%	40.8%	63.79%
	Med	**0%**	25.33%	8.66%	57.78%	1.05%	1.38%	**0.96%**	40.2%
	Mean	**3.38%**	23.91%	12.94%	52.7%	**2.01%**	9.2%	4.21%	39.74%

Table 2: Misclassification results for the interpolated video data set with various magnitudes of Gaussian noise, lower is better. In this test we interpolated the end and beginning of each scene together to form fading transitions. Numbers in brackets indicate how many times clustering was perfect, i.e. zero error.

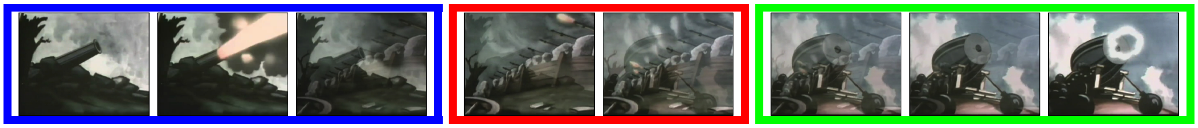


Figure 6: A sample of frames from the interpolated video test. We modified the existing video sequences to make a fading transition between each scene. Clusters (scenes) highlighted by coloured borders.

		Video 1				Video 2			
PSNR		OSC	SpatSC	LRR	SSC	OSC	SpatSC	LRR	SSC
Max	Min	0% (2)	**0% (3)**	0% (1)	0% (2)	0% (9)	**0% (11)**	0% (5)	**0% (11)**
	Max	**36.23%**	50.79%	66.3%	55.1%	**27.2%**	27.2%	41.9%	31.8%
	Med	1.44%	**1.19%**	16.85%	35.85%	0.76%	**0.37%**	4.45%	0.41%
	Mean	**10.46%**	12.38%	24.35%	27.82%	**1.74%**	1.58%	12.66%	5.29%
46	Min	0% (3)	**0% (6)**	0% (2)	0% (2)	0% (8)	**0% (10)**	0% (6)	**0% (10)**
	Max	**30.43%**	50.79%	47.83%	61.22%	**1.8%**	27.2%	34.63%	28.46%
	Med	1.28%	**1.19%**	12.24%	30.26%	0.76%	**0.37%**	0.84%	0.48%
	Mean	**8.85%**	10.92%	19.06%	27.93%	**0.68%**	1.56%	6.06%	5.38%
24	Min	0% (3)	**0% (8)**	0% (4)	0% (1)	0% (4)	0% (7)	**0% (8)**	0% (6)
	Max	**38.78%**	59.18%	49.65%	56.52%	**2.16%**	27.2%	27.2%	50%
	Med	1.44 %	**1.06%**	8.68%	31.34%	0.95%	**0.53%**	0.67%	0.83%
	Mean	**8.36%**	15.26%	14.61%	28.51%	**0.92%**	1.96%	2.63%	9.89%
10	Min	0% (1)	0.65%	**0% (4)**	12.5% (0)	0% (3)	0% (2)	**0% (8)**	0.61%
	Max	**46.94%**	58.73%	50.72%	64.93%	**12.61%**	31.65%	40.76%	65.11%
	Med	**3.19%**	28.21%	11.54%	51.32%	1.9%	1.05%	**0.73%**	33.72%
	Mean	**9.13%**	27.11%	15.4%	49.85%	**2.8%**	8.78%	4.01%	35.18%

Table 3: Misclassification results for the video data set with various magnitudes of Gaussian noise, lower is better. Numbers in brackets indicate how many times clustering was perfect, i.e. zero error.

PSNR		OSC	SpatSC	LRR	SSC
Max	Min	**0% (5)**	0% (7)	0% (1)	0% (1)
	Max	**54.69%**	56.25%	57.81%	57.81%
	Med	**3.66%**	7.3%	3.69%	3.73%
	Mean	10.56%	52.85%	**10%**	10.23%
46	Min	**0% (1)**	1.04%	0.52%	0.52%
	Max	**38.83%**	64.58%	41.71%	42.78%
	Med	**3.17%**	39.58%	4.17%	4.17%
	Mean	**8.22%**	34.95%	8.53%	8.53%
24	Min	**8.51%**	26.83%	47.97%	47.97%
	Max	**59.3%**	66.15%	66.15%	66.15%
	Med	**39%**	44.85%	65.52%	65.33%
	Mean	**36.56%**	49.42%	64.05%	63.97%

Table 4: Misclassification results for the face clustering dataset with various magnitudes of Gaussian noise, lower is better. Numbers in brackets indicate how many times clustering was perfect, i.e. zero error.

Acknowledgment

The research project is supported by the Australian Research Council (ARC) through the grant DP130100364.

References

[1] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. 1, 2, 4

[2] C. Tomasi and T. Kanade, “Shape and motion from image streams under orthography: a factorization method,” *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992. 1

[3] R. Basri and D. W. Jacobs, “Lambertian reflectance and linear subspaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003. 1

[4] Y. Guo, J. Gao, and F. Li, “Spatial subspace clustering for hyperspectral data segmentation,” in *Conference of The Society of Digital Information and Wireless Communications (SDIWC)*, 2013. 1, 2, 3

[5] 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, 2005. 1

[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. 2, 4

[7] R. Vidal, “A tutorial on subspace clustering,” *Signal Processing Magazine, IEEE*, vol. 28, no. 2, pp. 52–68, 2011. 2, 4

[8] E. Elhamifar and R. Vidal, “Sparse subspace clustering,” in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2790–2797. 2

[9] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000. 2, 4

[10] P. Xi, L. Zhang, and Z. Yi, “Constructing l2-graph for subspace learning and segmentation,” 2012. 2, 4

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011. 3

[12] R. Liu, Z. Lin, and Z. Su, “Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning,” in *ACML*, 2013, pp. 116–132. 3

[13] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, “Convex optimization with sparsity-inducing norms,” *Optimization for Machine Learning*, pp. 19–53, 2011. 3

[14] J. Liu and J. Ye, “Efficient l1/lq norm regularization,” *arXiv preprint arXiv:1009.4766*, 2010. 3

[15] A. S. Georghiades, P. N. Belhumeur, and 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*, vol. 23, no. 6, pp. 643–660, 2001. 6