Low Rank Sequential Subspace Clustering

Yi Guo*, Junbin Gao†, Feng Li‡, Stephen Tierney†, Ming Yin§
*CSIRO Digital Productivity Flagship, North Ryde, NSW 1670, Australia
Email: yi.guo@csiro.au
†School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW 2795, Australia
Email: jbgao@csu.edu.au,stierney@csu.edu.au
‡Qian Xuesen Laboratory of Space Technology, Beijing, China
Email: feng_li@aliyun.com
§School of Automation, Guangdong University of Technology, Guangzhou, China
Email: yiming@gdut.edu.cn

***Abstract*—Sequential data are ubiquitous in data analysis. For example hyperspectral data taken from a drill hole in geology, high throughput X-ray diffraction measurements in materials research and EEG brain wave signals in neuroscience. The common feature of sequential data is that they are all acquired subject to one external variable such as location, time or temperature. The data evolve along the direction of that variable through several patterns and the "neighboring" data are very likely to share similar features. The purpose of the segmentation for sequential data is then to identify those sequentially continuous segments/patterns. We approach this problem by adopting the subspace clustering method and propose a novel algorithm called low rank sequential subspace clustering (LRSSC), inspired by another method called spatial subspace clustering (SpatSC). SpatSC finds the subspaces by data self-reconstruction with a sparsity constraint on reconstruction weights and promotes the spatial smoothness of the weights by fusion, the essential part in the fused LASSO. However, the subspace identification capability is limited due to the indeterminacy of the sparse regression in finding suitable samples to linearly reconstruct a given sample. This confuses the graph cut algorithm that produces the final clustering results on the weights. To overcome this drawback, we propose to use the low rank penalty instead of sparsity in learning phase to separate subspaces. This improves the subspace identification as well as the robustness to noise. To demonstrate its effectiveness, we test LRSSC on both simulated and real world data compared with SpatSC and other methods. The proposed algorithm is superior to others when noise level is very high.**

I. Introduction

We are concerned with the sequential data segmentation problem in this paper. Sequential data are ubiquitous in data analysis. For example hyperspectral data taken from scanning through rock samples from a drill hole by spectrometers[1], [2]. These data are very useful for exploration and mining. There are also plenty of examples from other areas such as high throughput X-ray diffraction (XRD) measurements, which are used to assist materials research [3] and electroencephalogram (EEG) brain wave signals in neuroscience to understand brain activities [4]. The common feature of sequential data is that they are all acquired subject to one external variable such as location, time or temperature. The data evolve with the external variable through several patterns. Moreover, "neighboring" data very likely share similar features. The purpose of the segmentation for sequential data is then to identify those sequentially continuous segments/patterns for the discovery of new knowledge, for example the formation of rocks, the crystallisation and nucleation phases of a new material and sleep cycles from EEG data. Figure 1 shows the first 20 TIR (Thermal Infrared) reflectance spectral samples (after background correction [1] [2]) from a real drill hole called DDH9, which will be detailed in Section IV. They are colored from dark red to dark blue according to their depths in the drill hole. They look fairly consistent if we ignore the brightness differences. Typically the mineralogy is stratified so that segments of similar minerals congregate together as shown in Figure 2.

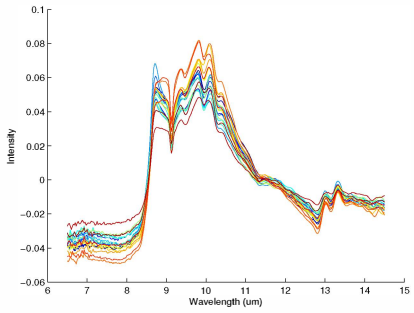


Fig. 1: First 20 background corrected spectra from a real drill hole called DDH9.

Fig. 2: An example of segmentation of a drill hole data set.

We consider this sequential segmentation problem through the subspace learning framework [5]. To facilitate further explanation, we introduce the following notations. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ be the matrix of the given sequential data set, and $\mathbf{x}_i \in \mathbb{R}^D$ the ith individual datum in the set. Note that the index $i \in \{1 \ldots N\}$ corresponds to the order of a particular sample acquired with the external variable such as physical location for spatial data e.g. drill hole spectral data. D is the dimensionality of the data. We write matrix $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_K]$,

[1]The background includes a temperature curve and a constant. Removing the constant from the spectra means dividing their means, which is standard in spectroscopy.

978-1-4799-1959-8/15/$31.00 ©2015 IEEE

$\mathbf{a}_i \in \mathbb{R}^D$, as the library of bases that generate all data, say the spectra library of pure materials [6], which may be absent. In subspace learning, the assumption is that the data $\mathbf{X}$ is contained in subspaces spanned by $\mathbf{A}$. We write $\mathbf{X}_{\mathcal{S}}$ as the submatrix of $\mathbf{X}$ where $\mathcal{S} \subseteq \{1 \dots N\}$ is some index subset. The basic model in subspace learning is

$$\mathbf{x}_i = \sum_{m \in \mathcal{M}} z_{im}\mathbf{a}_m + \boldsymbol{\epsilon}_i, \text{ s.t.} |\mathcal{M}| \leq M \tag{1}$$

for a subset $\mathcal{M} \subseteq \{1, \dots, K\}$, where $|\mathcal{M}|$ denotes the cardinality of set $\mathcal{M}$, M is the number of components contained in $\mathbf{x}_i$ which has to be estimated, and $\boldsymbol{\epsilon}_i$ is the error. When $\mathbf{A}$ is available, one can obtain the coefficients z_{im} in (1) by using least squares with subset selection [1], [6] or some sparse regularisation [7] to determine M. Therefore the model in (1) is often realised by solving

$$\min_{\mathbf{z}_i} \|\mathbf{x}_i - \mathbf{A}\mathbf{z}_i\|_p \text{ s.t. } \|\mathbf{z}_i\|_0 \leq M \tag{2}$$

where $\mathbf{z}_i$ is the vector of coefficients and $\|\mathbf{v}\|_p$ is the l_p-norm of $\mathbf{v}$ for example $p = 2$ corresponding to least squares.

Since the index of the sequential data is informative and plays an important role in understanding the dynamics, sequential data segmentation therefore requires clustering $\mathbf{x}_i$'s into several *continuous* segments so that each or several segments belong to a subspace spanned by several bases from $\mathbf{A}$. Precisely, $\forall i \in \mathcal{S}_j$, $\mathbf{x}_i \in \text{span}(\mathbf{A}_{\mathcal{U}_j})$ where $\mathcal{S}_j$ and $\mathcal{U}_j$ are the index sets for data and library corresponding to subspace j for $j \in \{1 \dots J\}$, $\text{span}(\mathbf{A})$ denotes the subspace spanned by $\mathbf{A}$, and the indices in $\mathcal{S}_j$ are continuous. Since neighbouring samples are very similar, we have

$$\mathbf{z}_i \approx \mathbf{z}_{i+1} \ \forall i, i+1 \in \mathcal{S}_j. \tag{3}$$

The obstacle in the segmentation problem discussed above is that the library $\mathbf{A}$ is not complete or is totally missing. This fact may drive researchers through the avenue of dictionary learning and sparse coding [8] to derive the library and regression coefficients at the same time. However for the purpose of segmentation, it is not necessary to obtain $\mathbf{A}$ at all. To solve this problem, spatial subspace clustering (SpatSC) [9], [3] integrates the fusion used in fused LASSO [10] into subspace learning. Note that SpatSC can be applied to sequential data without any modification. The mechanism of SpatSC is that the subspace identification is carried out by data self-reconstruction (use $\mathbf{X}$ as $\mathbf{A}$ in (1)) and sparsity constraint on the reconstruction weights, and the spatial smoothness is enforced by fusion. Without fusion, SpatSC is exactly the sparse subspace clustering [11], a typical subspace identification algorithm. It is shown in [11] that this data self-reconstruction plus sparsity constraint on weights guarantees the discovery of subspaces in noise free case and later extended to Gaussian noise case [12]. In SpatSC, the prior knowledge stated in (3) is utilised by minimising $\|\mathbf{z}_i - \mathbf{z}_{i+1}\|_1$, which is exactly the fusion. After the reconstruction weight matrix $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_N]$ is obtained, which is usually called the learning phase, a graph cut algorithm such as normalised cuts (NCUT) [13] is performed on some form of $\mathbf{Z}$ such as $\frac{|\mathbf{Z}|+|\mathbf{Z}^\top|}{2}$ for a final clustering solution.

It is clear that the success of graph based subspace learning algorithms such as SSC and SpatSC depends on the quality of $\mathbf{Z}$. However, the combination of self-reconstruction and sparsity constraint used in SSC and SpatSC for subspace identification suffers from the indeterminacy. The set of selected data from a subspace, called the reconstruction set, to linearly approximate a sample in the same subspace varies case by case because it is completely dependent on the noise present in the data. This may not be a problem for subspace clustering data without sequential structure. Nonetheless, when data have strong sequential correlation, it is desirable that the reconstruction sets for each continuous subspace are stable so that $\mathbf{Z}$ is clearly blocky meaning that $\mathbf{Z}$ contains column blocks of sub-matrices. As a result, the clustering accuracy could be improved. This problem is corrected to some extent by the fusion in SpatSC. However, it is not clear how to balance the effects of sparsity and fusion to achieve a stable reconstruction set for each subspace. This drawback motivates the use of the whole data set as the reconstruction set so that the reconstruction sets for all data are the same. Moreover the rank of a subspace should be low which gives rise to the use of the rank operator in the model. Finally, the sequential correlation among the data can be captured again by fusion. This idea is summarised as a novel algorithm called low rank sequential subspace clustering, or LRSSC for short.

In the rest of this paper we present the LRSSC model in Section II and its optimisation in Section III. To evaluate its effectiveness, we apply the proposed method to both synthetic and real world data in Section IV. Finally we conclude in Section V with a discussion.

II. Low Rank Sequential Subspace Clustering

The spatial subspace clustering has the following form

$$\begin{aligned} &\min_{\mathbf{Z}} \frac{1}{2}||\mathbf{E}||_F^2 + \lambda_1||\mathbf{Z}||_1 + \lambda_2||\mathbf{Z}\mathbf{R}||_1 \\ &\text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \ \text{diag}(\mathbf{Z}) = 0, \end{aligned} \tag{4}$$

where $\mathbf{Z} \in \mathbb{R}^{N \times N}$, $\mathbf{E} \in \mathbb{R}^{D \times N}$, and $||\mathbf{E}||_F$ is the Frobenius norm of matrix $\mathbf{E}$. $\mathbf{R}$ is a $N \times (N-1)$ matrix and

$$\mathbf{R} = \begin{bmatrix} -1 & & \\ 1 & \ddots & \\ & \ddots & -1 \\ & & 1 \end{bmatrix}.$$

We have the following remarks on the SpatSC model.

Remark II.1 (Subspace identification)**.** *Subspace identification is implemented by data self-reconstruction,* $\mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}$*, and sparsity regularisation on* $\mathbf{Z}$*, i.e.* $\lambda_1\|\mathbf{Z}\|_1$ *in the objective function* (4)*. The condition diag*$(\mathbf{Z}) = 0$ *is to avoid trivial solution although it is not essential in SpatSC. Note that when* $\lambda_2 = 0$*, SpatSC becomes exactly SSC.*

Remark II.2 (Spatial smoothness)**.** $\lambda_2||\mathbf{Z}\mathbf{R}||_1$ *is the fusion in the fused LASSO. The effect of this fusion is to smooth out the difference between neighboring columns in* $\mathbf{Z}$*, reflecting the prior knowledge in* (3)*.*

Remark II.3 (Reconstruction error $\mathbf{E}$)**.** *This is not the error* $\boldsymbol{\epsilon}_i$ *in* (1)*, but the difference between a sample and its estimation reconstructed linearly by others. In* (4)*,* $\mathbf{E}$ *is quantified by*

Frobenius norm. Other forms for this purpose can be considered. However, it is not very clear how this error is connected to the "native error" $\boldsymbol{\epsilon}_i$ that contaminates the data.

Let us focus on the subspace identification here. SpatSC shares the same features as SSC for this purpose. In the noise free case, i.e. $\boldsymbol{\epsilon}_i = 0$ in (1), the following procedure

$$\min_{\boldsymbol{\beta}_i} \|\boldsymbol{\beta}_i\|_1, \text{ s.t. } \mathbf{x}_i = \mathbf{X}_{-i}\boldsymbol{\beta}_i \tag{5}$$

guarantees that nonzero entries in $\boldsymbol{\beta}_i$ correspond to samples from the same subspaces as $\mathbf{x}_i$, where $\mathbf{X}_{-i}$ denotes matrix $\mathbf{X}$ with the ith column removed. In practice where noise is always present, (5) is relaxed to a linear regression problem with l_1 regularisation

$$\min_{\boldsymbol{\beta}_i} \|\mathbf{x}_i - \mathbf{X}_{-i}\boldsymbol{\beta}_i\|^2 + \lambda\|\boldsymbol{\beta}_i\|_1, \ (\forall i = 1 \sim N), \tag{6}$$

where $\lambda \geq 0$ controls the strength of the sparsity. $\boldsymbol{\beta}_i$'s are the translated feature vectors used by the graph cut algorithm for the final clustering solution. See [11] for details.

Obviously, (6) is a data driven regression. The sparse pattern of $\boldsymbol{\beta}_i$ depends on the noise defined in the basic model in (1), which are not the same across the samples. To be precise, we define reconstruction set $\mathcal{I}_i = \{k|\beta_{ik} \neq 0\}$, where β_{ik} is the kth element in $\boldsymbol{\beta}_i$. Then $\mathcal{I}_i$ is not necessarily equal to $\mathcal{I}_j$ even if $\mathbf{x}_i$ and $\mathbf{x}_j$ are from the same subspace. Furthermore, it is possible that $\mathcal{I}_i$ contains indices of samples from other subspaces that do not include $\mathbf{x}_i$. In the sequential data case, this lead to $\mathcal{I}_i \neq \mathcal{I}_{i+1}$ with some probability although $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ belong to the same subspace. This phenomenon propagates in sequence along the indices of the sequential data, which we call "the stability problem of reconstruction set". This is one of the major sources that compromises the performance of graph cut applied to $\boldsymbol{\beta}_i$'s.

For sequential data, it is desirable that the reconstruction set is stable along neighboring data when they are from the same subspace. It is not clear how one can achieve this goal with the objective in (6). This motivates the idea of using the whole data set as the only reconstruction set, i.e. $\mathcal{I}_i = \{1, \ldots, N\}$ $\forall i$. Then the stability of reconstruction set is ensured. As such the sparsity has to be taken away from the formula. But how to identify subspaces? Fortunately, the subspace model in (1) reveals that the subspaces are low rank, so are the reconstruction weights. Exploiting this observation, we propose the following low rank sequential subspace clustering (LRSSC) objective

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E}} \operatorname{rank}(\mathbf{Z}) \\ &\text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \ \|\mathbf{Z}\|_{TV} \leq t_1, \ \|\mathbf{E}\|_p \leq t_2, \end{aligned} \tag{7}$$

where t_1 and t_2 are pre-specified positive thresholds to control smoothness and reconstruction error and $\|\mathbf{X}\|_{TV}$ is total-variation seminorm e.g. $\|\mathbf{Z}\mathbf{R}\|_1$ as used in SpatSC.

The interpretation of (7) is straightforward, similar to that of SpatSC. The observations on spatial smoothness and reconstruction error remain the same. The only difference is from the subspace identification, which is now carried out by rank operator on $\mathbf{Z}$. The outcome from minimisation of (7) is a full matrix $\mathbf{Z}$ with smoothed patterns corresponding to sequentially located subspaces. This will be shown clearly in Section IV. Meanwhile, we have the following additional remarks.

Remark II.4. *When $t_1 = \infty$ and $p = 1, 2$ (l_1/l_2 norm [6]), the LRSSC degenerates to the low rank representation (LRR) model [14], which claims superior performance to SSC. We believe the superiority of LRR comes from the stability of reconstruction sets so that the learnt features, i.e. the reconstruction weights, contain consistent patterns from which the following graph cut algorithm will benefit. As LRSSC shares the same subspace separation mechanism as LRR, we expect that LRSSC outperforms SpatSC for the same reason.*

Remark II.5. *If a sparse $\mathbf{Z}$ is desirable, one can add another constraint to the model such as $\|\mathbf{Z}\|_1 \leq t_3$. The negative effect of sparsity to the stability of construction set may be eased by the low rank in* (7)*. However, we will not delve into this issue and leave it for our future work.*

Direct minimisation of (7) is NP-hard. Therefore we use the convex relaxation of rank operator, namely nuclear norm instead. Furthermore we relax the inequality constraints to regularisation. The revised objective is as follows.

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E}} \frac{1}{2}\|\mathbf{E}\|_F^2 + \lambda_1\|\mathbf{Z}\|_* + \lambda_2\|\mathbf{Z}\mathbf{R}\|_1 \\ &\text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \end{aligned} \tag{8}$$

where we choose Frobenius norm for the reconstruction error. The above problem is convex as summation of proper norms. The following section is about how to solve it.

III. Optimisation

The objective (8) is a multiple-block function with a linear constraint and therefore we use a variant of ADMM called LADMPSAP (linearized alternating direction method with parallel splitting and adaptive penalty) [15] for its simple structure and the convergence guarantee. We introduce variable substitution as

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E},\mathbf{J}} \frac{1}{2}\|\mathbf{E}\|_F^2 + \lambda_1\|\mathbf{Z}\|_* + \lambda_2\|\mathbf{J}\|_1 \\ &\text{s.t.} \quad \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \mathbf{J} = \mathbf{Z}\mathbf{R}. \end{aligned} \tag{9}$$

Then we have the Augmented Lagrangian

$$\begin{aligned} \mathcal{L}(\mathbf{E},\mathbf{Z},\mathbf{J},\mathbf{Y}_1,\mathbf{Y}_2|\mu) = &\frac{1}{2}\|\mathbf{E}\|_F^2 + \lambda_1\|\mathbf{Z}\|_* + \lambda_2\|\mathbf{J}\|_1 \\ &+ \langle\mathbf{Y}_1, \mathbf{X}\mathbf{Z} - \mathbf{X} + \mathbf{E}\rangle + \frac{\mu}{2}\|\mathbf{X}\mathbf{Z} - \mathbf{X} + \mathbf{E}\|_F^2 \\ &+ \langle\mathbf{Y}_2, \mathbf{J} - \mathbf{Z}\mathbf{R}\rangle + \frac{\mu}{2}\|\mathbf{J} - \mathbf{Z}\mathbf{R}\|_F^2, \end{aligned} \tag{10}$$

where $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are Lagrangian multipliers and $\mu > 0$ is the proximal parameter to be optimised. The update rules are explained in the following based on (10).

1) Update $\mathbf{Z}^{k+1}$

$$\mathbf{Z}^{k+1} = \arg\min_{\mathbf{Z}} \lambda_1\|\mathbf{Z}\|_* + F(\mathbf{Z}) \tag{11}$$

where

$$\begin{aligned} F(\mathbf{Z}) = &\langle\mathbf{Y}_1^k, \mathbf{X}\mathbf{Z} - \mathbf{X} + \mathbf{E}^k\rangle + \frac{\mu^k}{2}\|\mathbf{X}\mathbf{Z} - \mathbf{X} + \mathbf{E}^k\|_F^2 \\ &+ \langle\mathbf{Y}_2^k, \mathbf{J}^k - \mathbf{Z}\mathbf{R}\rangle + \frac{\mu^k}{2}\|\mathbf{J}^k - \mathbf{Z}\mathbf{R}\|_F^2. \end{aligned}$$

We approximate (11) by linearising $F(\mathbf{Z})$ with proximal term so that

$$\mathbf{Z}^{k+1} \approx \arg\min_{\mathbf{Z}} \lambda_1 \|\mathbf{Z}\|_* + \frac{\sigma_z^k}{2}\|\mathbf{Z} - (\mathbf{Z}^k - \frac{1}{\sigma_z^k}\nabla F(\mathbf{Z}^k))\|_F^2,$$

where $\sigma_z^k = \mu^k\rho$ (ρ is an appropriate constant) and

$$\begin{aligned}\nabla F(\mathbf{Z}^k) =& \mathbf{X}^T(\mathbf{Y}_1^k + \mu^k(\mathbf{X}\mathbf{Z}_k - \mathbf{X} + \mathbf{E}^k)) \\ &- (\mathbf{Y}_2^k + \mu^k(\mathbf{J}^k - \mathbf{Z}^k\mathbf{R}))\mathbf{R}^T.\end{aligned}$$

The above problem has closed-form solution

$$\mathbf{Z}^{k+1} = \mathbf{U}S_{\lambda_1/\sigma_z^k}(\mathbf{\Sigma})\mathbf{V}^\top, \tag{12}$$

where $\mathbf{U}$,$\mathbf{\Sigma}$ and $\mathbf{V}$ are from the singular decomposition of $\mathbf{Z}^k - \frac{1}{\sigma_z^k}\nabla F(\mathbf{Z}^k)$ and $S_{\lambda_1/\sigma_z^k}(\mathbf{\Sigma})$ is the singular value thresholding [16] defined as

$$S_\tau(\mathbf{\Sigma}) = \text{diag}(\max\{\sigma_i - \tau, 0\}),$$

where σ_i is the ith singular value of $\mathbf{\Sigma}$.

2) Update $\mathbf{E}^{k+1}$

$$\begin{aligned}\mathbf{E}^{k+1} = \arg\min_{\mathbf{E}} &\frac{1}{2}\|\mathbf{E}\|_F^2 + \langle \mathbf{Y}_1^k, \mathbf{X}\mathbf{Z}^k - \mathbf{X} + \mathbf{E}\rangle \\ &+ \frac{\mu^k}{2}\|\mathbf{X}\mathbf{Z}^k - \mathbf{X} + \mathbf{E}\|_F^2.\end{aligned} \tag{13}$$

This is a quadratic problem and we have

$$\mathbf{E}^{k+1} = \frac{\mathbf{X}\mathbf{Z}^k - \mathbf{X} + \frac{1}{\mu^k}\mathbf{Y}_1^k}{\frac{1}{\mu^k} + 1}. \tag{14}$$

If we take different norm to bound $\mathbf{E}$, e.g. the l_1 norm, the extra cost is to linearise the smooth part in (13) and use the same strategy as for updating $\mathbf{Z}^{k+1}$. The optimisation will still converge thanks to the general structure of LADMPSAP.

3) Update $\mathbf{J}^{k+1}$ by

$$\min_{\mathbf{J}} \lambda_2\|\mathbf{J}\|_1 + \langle \mathbf{Y}_2^k, \mathbf{J} - \mathbf{Z}^k\mathbf{R}\rangle + \frac{\mu^k}{2}\|\mathbf{J} - \mathbf{Z}^k\mathbf{R}\|_F^2.$$

The above problem has a closed-form solution defined by the soft thresholding operator as follows [17]

$$\mathbf{J}^{k+1} = \text{sign}\left(\mathbf{U}^k\right)\max\left(\left|\mathbf{U}^k\right| - \frac{\lambda_2}{\mu^k}\right), \tag{15}$$

where $\mathbf{U}^k = \mathbf{Z}^k\mathbf{R} - \mathbf{Y}_2^k/\mu_k$.

4) Update Lagrange multipliers by

$$\begin{aligned}\mathbf{Y}_1^{k+1} =& \mathbf{Y}_1^k + \mu^k(\mathbf{X}\mathbf{Z}^{k+1} - \mathbf{X} + \mathbf{E}^{k+1}) \\ \mathbf{Y}_2^{k+1} =& \mathbf{Y}_2^k + \mu^k(\mathbf{J}^{k+1} - \mathbf{Z}^{k+1}\mathbf{R})\end{aligned}$$

5) Update μ^{k+1} by $\mu^{k+1} = \min(\mu_{\max}, \gamma\mu^k)$.

$\mu_{\max}$, γ and ρ are chosen according to LADMPSAP requirements to ensure convergence. The above updating steps are summariesd as Algorithm 1.

Once $\mathbf{Z}$ has been obtained, we derive an affinity matrix $\mathbf{W}$ by $\mathbf{W} = \frac{|\mathbf{Z}|+|\mathbf{Z}^\top|}{2}$, which is input to NCUT to obtain the final clustering solution. Nevertheless, NCUT requires the number of clusters. As the determination of this number is beyond this paper, we assume that it is provided by some heuristics such as trial-and-error or cross validation.

Algorithm 1 LRSSC: solving (9)

Require: $\mathbf{X}^{D\times N}$, λ_1, λ_2, μ^0 (> 0), $\mu^{\max}$ ($>> \mu^0$), ρ ($> \|\mathbf{X}\|_F^2$), γ^0, and $\epsilon_1, \epsilon_2 > 0$.

Initialise $\mathbf{S} = \mathbf{0}$, $\mathbf{U} = \mathbf{S}\mathbf{R}$, $\mathbf{Y}_1 = \mathbf{1}$, $\mathbf{Y}_2 = \mathbf{1}$, $\mathbf{Z} = \mathbf{0}$
while not converged **do**
  Find $\mathbf{Z}^{k+1}$ by using (12)
  Find $\mathbf{E}^{k+1}$ by using (14)
  Find $\mathbf{J}^{k+1}$ by using (15)
  Check stopping criteria

$$\frac{\|\mathbf{X}\mathbf{Z}^{k+1} - \mathbf{X} + \mathbf{E}^{k+1}\|_F}{\|\mathbf{X}\|_F} < \epsilon_1;$$

$$\frac{\|\mathbf{J}^{k+1} - \mathbf{Z}^{k+1}\mathbf{R}\|_F}{\|\mathbf{X}\|_F} < \epsilon_1;$$

$$\frac{\mu^k\sqrt{\rho}}{\|\mathbf{X}\|_F}\max\{\|\mathbf{Z}^{k+1} - \mathbf{Z}^k\|_F, \|\mathbf{E}^{k+1} - \mathbf{E}^k\|, \|\mathbf{J}^{k+1} - \mathbf{J}^k\|_F, \|\mathbf{Z}^{k+1}\mathbf{R} - \mathbf{Z}^k\mathbf{R}\|_F)\} < \epsilon_2$$

  $\mathbf{Y}_1^{k+1} = \mathbf{Y}_1^k + \mu_1^k(\mathbf{X}\mathbf{Z}^{k+1} - \mathbf{X} + \mathbf{E}^{k+1})$
  $\mathbf{Y}_2^{k+1} = \mathbf{Y}_2^k + \mu_2^k(\mathbf{J}^{k+1} - \mathbf{S}^{k+1}\mathbf{R})$
  Update γ

$$\gamma = \begin{cases}\gamma^0 & \text{if } \frac{\mu^k\sqrt{\rho}}{\|\mathbf{X}\|_F}\max\{\|\mathbf{Z}^{k+1} - \mathbf{Z}^k\|_F, \\ & \|\mathbf{E}^{k+1} - \mathbf{E}^k\|, \|\mathbf{J}^{k+1} - \mathbf{J}^k\|_F, \\ & \|\mathbf{Z}^{k+1}\mathbf{R} - \mathbf{Z}^k\mathbf{R}\|_F\} < \epsilon_2 \\ 1 & \text{otherwise,}\end{cases}$$

  $\mu^{k+1} = \min(\mu_{\max}, \gamma\mu^k)$
end while
return $\mathbf{Z}$

IV. Experimental Evaluation

We tested LRSSC on a semi-simulation used in [9], [3] for the availability of *completely accurate* ground truth, and real data sets, including XRD measurement for material science and EEG data for sleep cycles.

We normalised each input datum to have unitary l_2 norm. We used $\lambda_1 = 0.01, \lambda_2 = 0.04$ throughout all experiments. Note that we did *not* optimise λ_1 and λ_2 in LRSSC but tested several combinations manually and fixed the ones that worked reasonably well. For other methods, we set the recommended values to parameters if any. We fixed the following optimisation parameters in LRSSC algorithm to ensure convergence. $\epsilon_1 = 10^{-2}$, $\epsilon_2 = 10^{-4}$, $\gamma^0 = 1.1$, $\mu_{\max} = 10$, $\rho = 1.1N$, and $\mu^0 = 0.1$. We allowed LRSSC to update no more than 200 iterations.

A. Semi-simulation

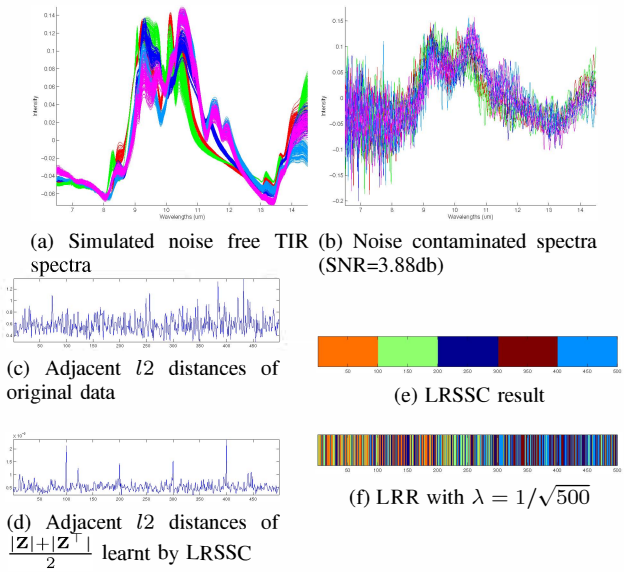
The simulation was carried out by using a thermal infrared (TIR) spectral library with 120 typical TIR spectra of pure materials and a real TIR spectral data set called DDH9. DDH9 was sampled by a hyperspectral data acquisition system from a mining site in Australia. It contains more than 9,000 valid spectra at 321 thermal infrared wavelengths (6um - 14um). We randomly picked 2 to 6 from the first 10 library spectra to form 5 spatially continuous subspaces with 100 samples (spectra) from each subspace. It is possible that some typical

spectra appear in multiple subspaces, which is very challenging indeed. Next we estimated the error structure by fitting the whole library to all DDH9 spectra. The mean $\boldsymbol{\epsilon}$ and covariance matrix $\boldsymbol{\Sigma}$ of the fitting error were then used to contaminate the samples so that they are close to reality. The simulated noise was drawn from a Gaussian distribution $\mathcal{N}(\boldsymbol{\epsilon}, m\boldsymbol{\Sigma})$ and added to the noise free samples from the subspaces, where $m > 0$ controls the noise level.

Figure 3 (a) shows the simulated TIR data without noise (different colors for spectra from different subspaces). There is little trouble for some subspace clustering algorithms to identify these subspaces. However, when we add noise, the situation changes drastically, especially when the noise level is as high as the case shown in Fig 3 (b), where the signal to noise ratio (SNR) of the spectra is very low (3.88db). The spectral features are not distinguishable and the wavelengths from 6um to 9um are inundated with noise. Note that we define

$$\text{SNR} = -10\log_{10}(\frac{||\mathbf{S}-\mathbf{T}||_F^2}{||\mathbf{S}||_F^2}), \tag{16}$$

where $\mathbf{S}$ is the noise free signal and $\mathbf{T}$ is the noise contaminated signal. This boils down to SNR$= -10\log_{10}[m\text{tr}(\boldsymbol{\Sigma})]$ in our case. When $m = 10^4$, the SNR is 3.88db. The l_2 distances between adjacent spectra shown in Figure 3 (c) have no distinct pattern for easy segmentation.



(a) Simulated noise free TIR spectra

(b) Noise contaminated spectra (SNR=3.88db)

(c) Adjacent $l2$ distances of original data

(e) LRSSC result

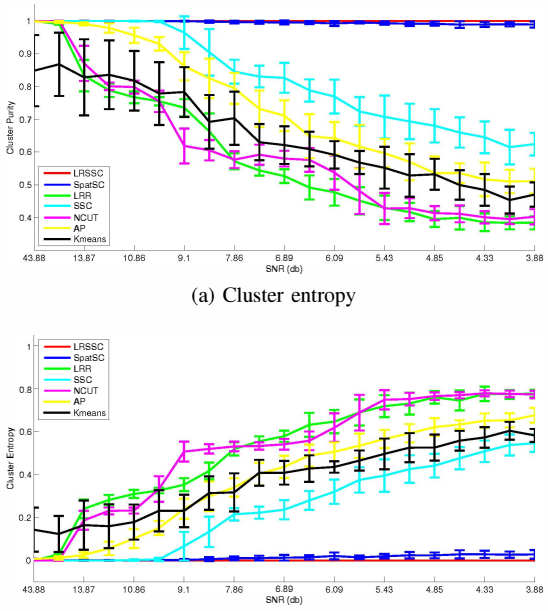
(d) Adjacent $l2$ distances of $\frac{|\mathbf{Z}|+|\mathbf{Z}^\top|}{2}$ learnt by LRSSC

(f) LRR with $\lambda = 1/\sqrt{500}$

Fig. 3: Semi-simulated TIR spectra and subspace clustering results.

Figure 3 (e) plots the clusters obtained by LRSSC for these very noisy spectra, which recover the true subspace perfectly. LRR clusters shown in Figure 3 (f) are nowhere close to the truth. Figure 3 (d) shows the l_2 distance between columns of $\mathbf{Z}$ from LRSSC, the learnt features from the original data. It is quite clear that the borders of a subspace are marked by much larger l_2 distances, or in other words, the LRSSC model successfully extracted the distinct features that are more suitable for sequential clustering. SpatSC has very close performance to LRSSC, which we cannot tell visually. To better assess the performance, we used cluster purity and cluster entropy [18] to quantify the clustering results. Cluster entropy of the rth cluster of size N_r is defined as $E_r = -\frac{1}{\log q}\sum_{i=1}^{q}\frac{N_r^i}{N_r}\log\frac{N_r^i}{N_r}$ where q is the number of classes in the dataset and N_r^i is the number of samples of the ith class that were assigned to the rth cluster. Classes mean true clusters. The entropy of the entire clustering solution of total K clusters is given by $E = \sum_{r=1}^{K}\frac{N_r}{N}E_r$. The cluster purity of a clustering result is defined as $P = \sum_{r=1}^{K}\frac{N_r}{N}P_r$, where $P_r = \frac{1}{N_r}\max_i N_r^i$ is the purity of the rth cluster. The cluster entropy describes how the various classes of samples are distributed within each cluster, while the purity measures the extent to which each cluster contains samples from primarily one class. Using these two measures in conjunction enables us to evaluate the clustering quality comprehensively. The ideal clustering solution should have high purity and low entropy. Furthermore, we also extended our comparison to other clustering methods to include NCUT, SSC, affinity propagation (AP) [19] and Kmeans [20] with their recommended parameters.

We repeated the semi-simulation with very low SNR (3.88db) 20 times and collected the mean entropy and purity results for both individual clusters and overall solution of each method in Table I. The clusters produced by LRSSC are very pure. Other solutions have significant confusion of classes indicated by high entropy and low purity values.



(a) Cluster entropy

(b) Cluster purity

Fig. 4: Clustering quality comparison.

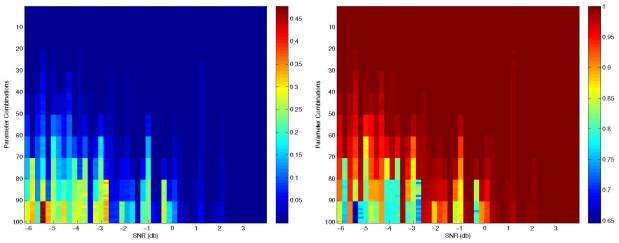
We did a more comprehensive comparison on this semi-simulated data. We varied SNR from about 44db to 4db to test the clustering quality of different methods. For each noise level, we repeated 20 times to obtain the mean and variance of the clustering quality. The final results are presented in Figure 4, where the horizontal axis is the SNR which is sampled to equal spacing for better visual presentation. LRSSC works quite well and the clustering quality drops little as SNR

Methods	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	Overall
LRSSC	0.0000/1.0000	0.0000/1.0000	0.0000/1.0000	0.0000/1.0000	0.0000/1.0000	0.0000/1.0000
SpatSC	0.0247/0.9902	0.0233/0.9911	0.0217/0.9908	0.0221/0.9931	0.0411/0.9833	0.0281/0.9890
LRR	0.7842/0.3812	0.8205/0.3544	0.7111/0.4356	0.8176/0.3578	0.8309/0.3575	0.7784/0.3839
SSC	0.5086/0.6783	0.5299/0.6308	0.5329/0.6347	0.5533/0.6276	0.5934/0.5804	0.5466/0.6240
NCUT	0.7398/0.4269	0.7788/0.3964	0.7725/0.3942	0.7388/0.4329	0.8890/0.3313	0.7734/0.4023
AP	0.6233/0.5696	0.5333/0.6241	0.7811/0.4752	0.7241/0.4581	0.6926/0.4716	0.6769/0.5110
Kmeans	0.5760/0.4742	0.5486/0.5401	0.5531/0.4858	0.5862/0.4593	0.6187/0.4454	0.5828/0.4701

TABLE I: Evaluation of clustering results (shown as entropy/purity) when SNR is 3.88db in semi-simulation.

decreases. In contrast, other methods' performance deteriorates to some extent.

We provide some empirical analysis on regularisation parameters in LRSSC, i.e. λ_1 and λ_2, which are trade-off between reconstruction quality, low rank and sequential smoothness. We tested the values of λ_1 and λ_2 within the range from 0.1 and 1 with step size 0.1 (100 combinations) under the noise level from 3.88db to -6.12db (the negative SNR means the power of noise is essentially stronger than signal, see (16) for SNR). The reason we chose such strong noise is that we found in our experiments that LRSSC was extremely robust to noise and insensitive to these parameters. In other words, when SNR is 3.88db, which is challenging already to other methods including SpatSC, LRSSC has constantly perfect clustering solution across a large range of values of parameters. Therefore we have to turn up the noise level to observe its variance of performance. Figure 5 provides the evidence of the robustness of LRSSC. Even when the noise level is incredibly high, LRSSC still gets almost perfect clustering with various values of parameters.

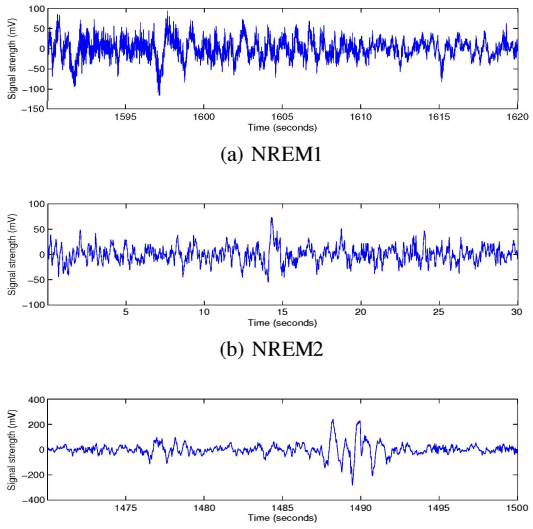


(a) Entropy (b) Purity

Fig. 5: The effect of the parameters in LRSSC. Vertical axis is for the combination of parameters and horizontal axis for SNR. Color corresponds to entropy or purity value.

B. LRSSC on EEG Data

We now apply LRSSC to EEG signals for sleep research. Neurologists often classify sleep into several stages based on some physiological characteristics including rapid eye movement (REM) stages and non-REM (NREM) stages. The EEG data we tested on is a 65 minute recording of a single channel brain wave signal containing three NREM sleep stages. According to the standards set out in [21], researchers classify sleep in 30 second epochs. Figure 6 shows 30 second epochs of three NREM stages. The data have been labelled with three NREM stages as ground truth. The purpose of applying LRSSC is to segment the signal to clusters corresponding to NREM stages exploring the time correlation among signals. Spectrogram is widely used in EEG signal processing [22], [4] which is adopted here as a preprocessing. We applied FFT (fast Fourier transformation) [23] of 256 points on each of 30 second epochs without overlap and obtained the power spectra. We discarded the frequencies beyond 30Hz containing no interesting information. This generated a 30×130 matrix as the observed data matrix for clustering algorithms.



(a) NREM1

(b) NREM2

(c) NREM3

Fig. 6: EEG signals

The hypnogram [21] (a combination of spectrogram of EEG signal and NREM stages) with ground truth and aligned LRSSC clusters is shown in Figure 7, where the lines indicate the stages labelled on right hand side of the figure. To avoid cluttered lines, we present only LRSSC clusters in the figure. The detailed cluster quality comparison is presented in Table II. LRSSC has slightly better results than SpatSC while both of them are better than others. Note that at around 1400 seconds towards 1800 seconds, there are four rapid NREM sleep stage shifts, especially the one from NREM 3 to NREM 2 and from NREM 2 to NREM 1 at around 1500 second. This short stay at NREM 2 for about 30 seconds is almost devastating to sequential clustering because the sequential smoothness penalty imposed in both LRSSC and SpatSC is "fixed" so that this short stay is screened out as noise. Although LRSSC identifies the majority of the true stages, these rapid stage changes are totally missed. This observation brings up the desire of introducing adaptive smoothness into

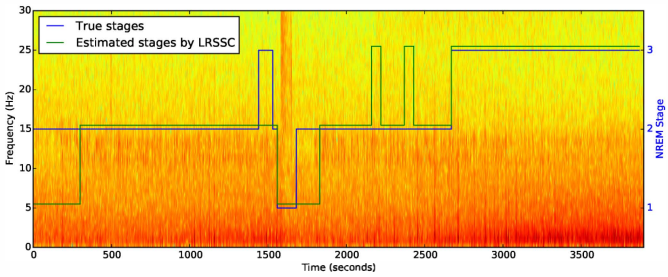


Fig. 7: Hypnogram of EEG with stages estimated by LRSSC (green lines) and ground truth (blue lines).

Methods	NREM1	NREM2	NREM3	Overall
LRSSC	0.277/0.909	0.168/0.955	0.455/0.800	0.249/0.915
SpatSC	0.234/0.929	0.181/0.950	0.455/0.800	0.252/0.915
LRR	0.552/0.705	0.541/0.783	0.479/0.850	0.526/0.777
SSC	0.564/0.760	0.388/0.889	0.644/0.682	0.542/0.769
AP	0.548/0.778	0.420/0.875	0.637/0.689	0.539/0.777
NCUT	0.664/0.646	0.534/0.787	0.461/0.857	0.562/0.754
Kmeans	0.644/0.667	0.380/0.884	0.480/0.848	0.515/0.785

TABLE II: The cluster entropy/purity of different algorithms on EEG data.

these algorithms.

C. *LRSSC on* in situ *XRD Material Data*

Finally we apply LRSSC to *in situ* XRD measurements for material science research. As materials research has embraced high throughput, it is necessary to have some tools available for automated data analysis in this area. Figure 8 shows one XRD data set of natural arsenian plumbojarosite measured at different time and temperatures [24]. Each curve is a series of X-Ray counts at different angles (2θ in degrees). The plumbojarosite sample was placed in an XRD instrument and the measurements were taken at a range of controlled and stepwise increasing temperatures from $20 \sim 900°C$ along time. So each curve in Fig. 8 corresponds to an XRD measurement of the sample at a certain temperature and we can also treat them as time or temperature evolving spectra. As temperature increases, the material goes through a complex breakdown and recrystallisation sequence including the formation of amorphous intermediate phases. These phase changes are more obvious in Figure 9, where the data matrix is presented as an image. We show only a subset of angles of the data where the most distinct features are prominent. A human interpreter can easily pick up the main parts of the phases (clusters) while the boundaries may seem questionable. The purpose of this exercise is to identity these time and temperature evolving phases to understand the dynamics of the material formation [24].

The authors of [24] derived 7 continuous clusters indicating 7 phases using a hierarchical clustering procedure with some manual tweaking. Within those clusters, the XRD curves share similar patterns. We take these manually tuned clusters as ground truth and compare the performance of different methods here. The boundaries of the ground truth clusters are shown as white bars on the left of the image shown in Figure 9. The clustering solutions were quantified by using cluster entropy and cluster purity again. Table III shows the results of different methods on each cluster and their overall performance. LRSSC has the lowest overall entropy and the second highest purity (next to SSC) respectively, although the leading entropy reading from LRSSC is only marginal. However, checking Figure 9 where the segments identified by different methods are plotted, one will find that LRSSC results are actually preferable as they more or less include some true cluster members in each segment while SSC splits the first segment into two. Note that LRSSC, SpatSC, SSC and kmeans produce continuous clusters along time and temperature. In contrast, AP and LRR have disjoint clusters and therefore their results cannot be used for this task, although they produce pretty good results on several individual clusters. LRSSC found several cluster boundaries very close to the ground truth such as the boundaries between cluster 5 and 6 and cluster 3 and 4. But it overestimated cluster 5 and 3. The solutions from other methods follow the same pattern but they miss or split true clusters to some extent.

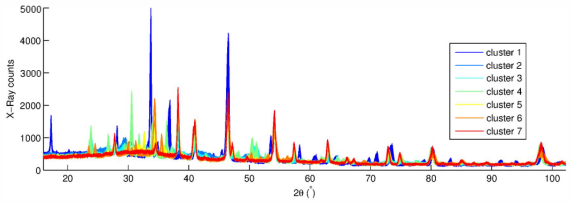


Fig. 8: *In situ* XRD data of plumbojarosite.

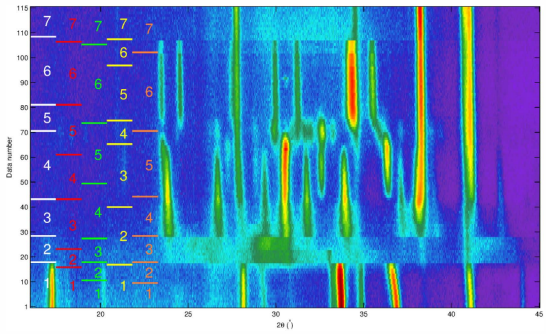


Fig. 9: The image of the *in situ* XRD data of plumbojarosite and the temporal clusters. The white, red, green and yellow bars on the left are the ground truth, LRSSC, SpatSC, kmeans and SSC clusters respectively. AP and LRR did not produce continuous clusters so they are excluded here.

V. Discussion

We proposed the low rank sequential subspace clustering (LRSSC) algorithm to address the clustering problem for sequential data such as spatial data (drill hole hyperspectral data) and time series data (EEG). It is based on subspace learning framework utilising data self-reconstruction and low rank prior for subspace identification and incorporates the

Methods	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6	cluster 7	Overall
LRSSC	0.0000/1.0000	0.3074/0.7143	0.2962/0.7368	0.0000/1.0000	0.3555/0.5263	0.0000/1.0000	0.2108/0.8571	0.1520/0.8435
SpatSC	0.0000/1.0000	0.0000/1.0000	0.0000/1.0000	0.3974/0.6667	0.1990/0.8696	0.2792/0.7667	0.2572/0.8000	0.2187/0.8261
LRR	0.8041/0.3636	0.7056/0.2727	0.0000/1.0000	0.6928/0.4688	0.0000/1.0000	0.0000/1.0000	0.0000/1.0000	0.6354/0.4696
SSC	0.0000/1.0000	0.0000/1.0000	0.3271/0.6667	0.3271/0.6667	0.0000/1.0000	0.0000/1.0000	0.1259/0.9333	0.1529/0.8522
AP	0.0000/1.0000	0.0000/1.0000	0.0000/1.0000	0.6501/0.5000	0.6359/0.5294	0.5459/0.6667	0.7058/0.2857	0.5522/0.5652
NCUT	0.2700/0.7813	0.0000/1.0000	0.3562/0.5000	0.2172/0.8500	0.0000/1.0000	0.3271/0.6667	0.1394/0.9231	0.2271/0.7826
Kmeans	0.0000/1.0000	0.4345/0.5000	0.1936/0.8750	0.3530/0.5556	0.3074/0.7143	0.0000/1.0000	0.1394/0.9231	0.2231/0.7826

TABLE III: The cluster entropy/purity of different algorithms on *in situ* XRD data.

fusion used in fused LASSO to exploit sequential smoothness. Its effectiveness and robustness against noise have been demonstrated by its application to semi-simulation and real world data sets.

LRSSC has shown powerful clustering capability for sequential data, however, there are still questions yet to be answered. The first is the understanding of the effect of the parameters. We know conceptually that they control subspace identification and sequential smoothness respectively. However their specific interactions are not clearly understood. Furthermore, it is very interesting that LRSSC seems very tolerant to the choice of the values of these two parameters which is demonstrated by the simulation experiments. This may indicate that an interval of values of parameters for successful recovery of the sequentially continuous subspace could be derived. Another question is the smallest size of the segment that is recognisable by LRSSC. In the EEG experiment section, we showed this problem. We also pointed out that an adaptive smoothness penalty may be desirable. We believe this is connected with the regularisation parameters and we will explore these issues in near future.

ACKNOWLEDGEMENTS

The first author is supported by the CSIRO AMTCP high throughput research project. The work of the second author is supported by Australian Research Council (ARC) under grant DP130100364. The third author appreciates the support of National Natural Science Foundation of China under grant no. 41371415.

REFERENCES

[1] Y. Guo and M. Berman, "A comparison between subset selection and L1 regularisation with an application in spectroscopy," *Chemometrics and Intelligent Laboratory Systems*, vol. 118, pp. 127–138, 2012.

[2] M. Berman, L. Bischof, R. Lagerstrom, Y. Guo, J. Huntington, and P. Mason, "An unmixing algorithm based on a large library of shortwave infrared spectra," CSIRO Mathematics, Informatics and Statistics, Tech. Rep. EP117468, 2011.

[3] Y. Guo, J. Gao, and F. Li, "Spatial subspace clustering for drill hole spectral data," *Journal of Applied Remote Sensing*, vol. 8, no. 1, p. 083644, 2014.

[4] M. L. Scheuer, "Continuous EEG monitoring in the intensive care unit," *Epilepsia*, vol. 43, no. s3, pp. 114–127, 2002.

[5] R. Vidal, "A tutorial on subspace clustering," in *Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[6] Y. Guo, M. Berman, and J. Gao, "Group subset selection for linear regression," *Computational Statistics And Data Analysis*, vol. 75, pp. 39–52, 2014.

[7] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301–320, 2005.

[8] M. Harandi, C. Sanderson, C. Shen, and B. Lovell, "Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution," in *International Conference on Computer Vision (ICCV)*, Sydney, 2013.

[9] Y. Guo, J. Gao, and F. Li, "Spatial subspace clustering for hyperspectral data segmentation," in *Proc of the Conference of The Society of Digital Information and Wireless Communications (SDIWC)*, 2013.

[10] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[11] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[12] M. Soltanolkotabi, E. Elhamifar, and E. J. Cands, "Robust subspace clustering," *The Annals of Statistics*, vol. 42, no. 2, pp. 669–699, 04 2014.

[13] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transaction on Pattern Analysis And Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Augest 2000.

[14] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[15] R. Liu, Z. Lin, and Z. Su, "Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning," *JMLR: Workshop and Conference Proceedings*, vol. 29, pp. 1–16, 2013.

[16] J. F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2008.

[17] I. Daubechies, M. Defrise, and C. DeMol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[18] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Machine Learning*, vol. 55, pp. 311–331, 2004.

[19] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, 2007.

[20] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster Analysis*, 5th ed., ser. Wiley Series in Probability and Statistics. Wiley, January 2011.

[21] C. Iber, S. Ancoli-Israel, A. L. C. JR, and S. F. Quan, *The AASM Manual for the Scoring of Sleep and Associated Events: RRule, Terminology and Technical Specifications*, Westchester, IL, 2007.

[22] R. Khosrowabadi, C. Quek, K. K. Ang, S. W. Tung, and M. Heijnen, "A brain-computer interface for classifying EEG correlates of chronic mental stress," in *The 2011 International Joint Conference on Neural Networks (IJCNN)*, July 2011, pp. 757–762.

[23] H. J. Nussbaumer, *Fast Fourier Transformation and Convolution Algor*, second edition ed. Springer, 1990.

[24] I. C. Madsen and I. E. G. S. Mills, "In situ diffraction studies: Thermal decomposition of a natural plumbojarosite and the development of rietveld-based data analysis," *Materials Science Forum*, vol. 651, pp. 37–64, 2010.