ENHANCED STREAMING BASED SUBSPACE CLUSTERING APPLIED TO ACOUSTIC SCENE DATA CLUSTERING

Shuoyang Li[1], *Yuantao Gu*[2], *Yuhui Luo*[3], *Jonathon Chambers*[4], *Wenwu Wang*[1]
[1] Department of Electrical and Electronic Engineering, University of Surrey, UK
[2] Department of Electronic Engineering, Tsinghua University, China
[3] Data Science Department, National Physical Laboratory, UK
[4] Department of Engineering, University of Leicester, UK

ABSTRACT

Labelled data are often required to train an acoustic scene classification system. However, it is time-consuming and expensive to label the data manually. An unsupervised clustering algorithm can be used to facilitate the labelling process by dividing the acoustic data into different categories. Nevertheless, it can be problematic to run a clustering algorithm with growing data volume and dimension due to the sharp increase in the computational and memory costs. We propose a new streaming based subspace clustering algorithm which allows the data to be clustered on the fly, and also resolves data points in the overlapping regions of two subspaces by augmenting the learned low-rank representation with the original data samples. Experimental results show that our method can achieve the clustering objective for overwhelmingly high-volume data in an online fashion, while retaining good accuracy and reducing the memory cost significantly.

1. INTRODUCTION

Acoustic scene classification has attracted much research attention recently [1]. Labelled acoustic data, are often required to train the classification system. However, in practice, data labelling is an expensive and time-consuming task. A potential solution to the issue is to cluster the audio data by an unsupervised learning algorithm.

Subspace clustering is a powerful technique for learning class labels in high-dimensional data in an unsupervised manner. Compared to conventional clustering algorithms such as k-means [2] which rely on the spatial distances and centroids of the clusters, subspace clustering algorithms offer the potential to capture the clusters in different subsets of dimensions [3], and provide more reliable clustering results in scenarios with high dimensional datasets such as image clustering [4] and moving trajectory segmentation [5]. Recently, subspace clustering has also been used for acoustic scene [6] and vehicle sound classification [7].

Subspace clustering algorithms aim to divide a set of high dimensional data according to their intrinsic subspace distributions. Conventional subspace clustering algorithms include Generalized PCA [8], [9], K-subspace clustering [10], [11], and Agglomerative Lossy Compression [12]. Recently, the spectral clustering based subspace clustering methods, such as Sparse Subspace Clustering (SSC) [13] and Low-Rank Representation (LRR) [14], have drawn significant attention due to their outstanding performance. Nevertheless, these methods suffer from unaffordable computational and memory costs when dealing with overwhelmingly large datasets. The acoustic dataset considered in this paper, for example, has a large data volume, and is impractical to be clustered by batch methods.

To solve the computational problems, methods such as scalable-SSC [15] and sketched subspace clustering [16] have been proposed. These methods can simplify the computation, however, a large memory space is still needed. To address this problem, a method for streaming data has been introduced in [17, 18], based on LRR and matrix factorisation, and offers low memory cost. However, the clustering performance of such methods degrades when dealing with overlapping subspaces, and their convergence is relatively slow, often taking a large number of data points to converge.

In this paper, we propose two improvements to the streaming based subspace clustering algorithms, where the distance based clustering is used to augment the low-rank subspace clustering and to facilitate the resolution of the data points in the overlapping regions of the subspaces, and a new warming-up scheme is proposed to improve the convergence of the streaming algorithm.

2. BACKGROUND

The subspace clustering models and typical algorithms are reviewed in this section.

2.1. Subspace Clustering

For a set of high-dimensional data, which consist of data points from varied categories, the intrinsic dimension of each category is usually smaller than the ambient dimension [19]. Subspace clustering aims to determine the data segmenta-

This research was supported by EPSRC grant EP/N014111/1 Making Sense of Sounds.

tion according to subspace memberships. In the popular spectral clustering method, the pairwise similarities between data points are used to segment data into several groups, by maximizing the intra-group similarities while minimizing the inter-group similarities, based on an affinity matrix formed in terms of pairwise affinities.

To construct the affinity matrix, a coefficient matrix $\boldsymbol{C}$ is often obtained by optimising the following cost function

$$\min_{\boldsymbol{C}} \|\boldsymbol{C}\|_\ell + \frac{\lambda}{2}\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{C}\|_F^2 \tag{1}$$

where $\boldsymbol{X}$ and $\boldsymbol{A}$ stand for the data matrix and a set of basis respectively. Entries of $\boldsymbol{C}$ are the representation coefficients of $\boldsymbol{X}$ in terms of $\boldsymbol{A}$. The regularizer $\|\cdot\|_\ell$ is a matrix norm, e.g. the ℓ_1 norm for SSC, and the nuclear norm for LRR. A symmetric affinity matrix $\boldsymbol{W}$, can be constructed as $\boldsymbol{W} = |\boldsymbol{C}| + |\boldsymbol{C}^\top|$, where $|\cdot|$ takes the element-wise absolute value.

The LRR algorithm utilises the low-rank representation coefficients to obtain the data affinity, where the nuclear-norm is often used to enforce the low-rank property. Since the nuclear-norm can be converted to a minimization of the sum of two Frobenius-norms of matrices [20], it offers the potential for the LRR to be solved in a streaming manner, thus reducing the memory cost for their implementation.

2.2. Online Low-Rank Subspace Clustering

The online implementation for subspace clustering has been introduced to mitigate the high memory cost of an overwhlmingly large input dataset [18, 21]. In terms of [22], the coefficient matrix $\boldsymbol{C} \in \mathbb{R}^{N\times N}$ can be represented by two low-rank matrices, exploiting the property

$$\|\boldsymbol{C}\|_* = \min_{\boldsymbol{C}=\boldsymbol{U}\boldsymbol{V}^\top} \frac{1}{2}(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2) \tag{2}$$

where $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{N\times r}$, and r denotes the supremum of the rank of $\boldsymbol{C}$. For the LRR case, applying this property to (1), the loss function optimisation becomes

$$\min_{\boldsymbol{U},\boldsymbol{V}} \frac{1}{2}\|\boldsymbol{U}\|_F^2 + \frac{1}{2}\|\boldsymbol{V}\|_F^2 + \frac{\lambda}{2}\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{U}\boldsymbol{V}^\top\|_F^2 \tag{3}$$

The matrix $\boldsymbol{C}$ can be obtained by solving this cost.

Traganitis and Giannakis [17] proposed a streaming method to solve the above loss function, by using a smaller basis matrix $\boldsymbol{A} \in \mathbb{R}^{D\times n}$ ($n \ll N$), so that $\boldsymbol{U} \in \mathbb{R}^{n\times r}$ with a concomitant decrease in the number of rows. It is assumed that the tth column of the data matrix $\boldsymbol{x}_t$ is known by the system at time $t \in \{1, 2, \cdots, N\}$. Letting $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ denote the ith column of $\boldsymbol{U}$ and $\boldsymbol{V}^\top$, and $\boldsymbol{U}_t$ and $\boldsymbol{V}_t$ to denote $\boldsymbol{U}$ and $\boldsymbol{V}$ at time t, the algorithm updates $\boldsymbol{v}_t$ and $\boldsymbol{U}_t$ alternatively with stochastic gradient descent [23]. However, the matrix $\boldsymbol{A}$ needs to be representative of the whole dataset $\boldsymbol{X}$, and the selection of $\boldsymbol{A}$ requires a uniformly sampling or sketching of $\boldsymbol{X}$, which needs prior knowledge about the whole dataset before running the streaming algorithm. Thus, this algorithm is not easy to be implemented in a full online manner.

Shen et al. [18] proposed an online low-rank subspace clustering (OLRSC) method. Different from the previous method, they use a basis matrix $\boldsymbol{A} \in \mathbb{R}^{D\times N}$. Thus, $\boldsymbol{U} \in \mathbb{R}^{N\times r}$ cannot be updated in a batch mode due to the unknown dimension N and the requirement for a large memory space. To address this, an auxiliary matrix $\boldsymbol{G} \in \mathbb{R}^{D\times r}$ is introduced, and $\boldsymbol{G} = \boldsymbol{A}\boldsymbol{U}$, which is no longer related to N. Then, $\boldsymbol{G}$ can be derived in a batch mode at each time point. In addition, a sparse corruption of data, denoted by $\boldsymbol{E}$, is taken into account, with a sparse constraint promoted by the ℓ_1 norm. Consequently, the loss function becomes

$$\begin{aligned}\min_{\boldsymbol{G},\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}} &\frac{\lambda_1}{2}\|\boldsymbol{X} - \boldsymbol{G}\boldsymbol{V}^\top - \boldsymbol{E}\|_F^2 + \lambda_2\|\boldsymbol{E}\|_1 \\ &+ \frac{\lambda_3}{2}\|\boldsymbol{G} - \boldsymbol{A}\boldsymbol{U}\|_F^2 + \frac{1}{2}\|\boldsymbol{U}\|_F^2 + \frac{1}{2}\|\boldsymbol{V}\|_F^2\end{aligned} \tag{4}$$

To solve it in an online manner, $\boldsymbol{x}_t$ and $\boldsymbol{a}_t$ are used to denote the tth column of $\boldsymbol{X}$ and $\boldsymbol{A}$, i.e. the input at time point t. Meanwhile, $\boldsymbol{e}_i$, $\boldsymbol{u}_i$, and $\boldsymbol{v}_i$ denote the ith columns of $\boldsymbol{E}, \boldsymbol{U}^\top$, and $\boldsymbol{V}^\top$ ($\boldsymbol{e}_i, \boldsymbol{u}_i, \boldsymbol{v}_i \in \mathbb{R}^D, i \in [1, ..., N]$), respectively, and $\boldsymbol{G}_t$ denotes $\boldsymbol{G}$ generated at time t. Then the loss at time t can be rewritten as

$$\begin{aligned}\min_{\boldsymbol{G},\boldsymbol{u},\boldsymbol{v},\boldsymbol{e}} &\frac{1}{2}\|\boldsymbol{u}_t\|_2^2 + \frac{1}{2}\|\boldsymbol{v}_t\|_2^2 + \frac{\lambda_1}{2}\|\boldsymbol{x}_t - \boldsymbol{G}_t\boldsymbol{v}_t - \boldsymbol{e}_t\|_2^2 \\ &+ \lambda_2\|\boldsymbol{e_t}\|_1 + \frac{\lambda_3}{2}\|\boldsymbol{G}_t - \sum_{i=1}^{t}\boldsymbol{a}_i\boldsymbol{u}_i^\top\|_F^2\end{aligned} \tag{5}$$

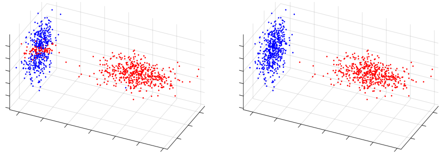
By fixing other variables in (5), $\boldsymbol{v}_t$, $\boldsymbol{e}_t$, $\boldsymbol{u}_t$ and $\boldsymbol{G}_t$ can be solved iteratively, by a coordinate descent algorithm [24]. Only three $D \times r$ accumulators are used and as a result, the memory cost is saved. Afterwards, $\boldsymbol{C}$ can be generated by $\boldsymbol{C} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_N]^\top \times [\boldsymbol{v}_1, ..., \boldsymbol{v}_N]$. The spectral clustering can be applied as post-processing to obtain the clusters.

Alternatively, the k-means algorithm [25] could be applied on $[\boldsymbol{v}_1, ..., \boldsymbol{v}_N]$ as post-processing, which has the loss function optimisation abstracted as

$$\min_{S}\sum_{k=1}^{K}\sum_{\boldsymbol{v}\in S_k}\|\boldsymbol{v} - \boldsymbol{p}_k\|_2^2 \tag{6}$$

where S_k denotes the kth cluster, and $\boldsymbol{p}_k$ is the centroid of the kth cluster. Using an online k-means algorithm [26] with $\boldsymbol{v}_i$, a fully online implementation can be achieved. In [18], it is claimed that $\boldsymbol{v}_i$ is a robust feature for the ith sample.

Experiments have revealed that OLRSC obtained good clustering accuracy [18], with a streaming implementation and low memory costs, hence OLRSC is our focus here. However, we observed that OLRSC is degraded when dealing with intersecting (overlapping) subspaces, and the convergence of this algorithm is relatively slow with our acoustic dataset. To address these problems, we propose new ideas for improvements, as discussed next.



(a) (b)

Fig. 1. Clustering results by (a) OLRSC and (b) the proposed JOLRSC for points in two 2-D subspace with low noise.

3. THE PROPOSED METHOD

3.1. Dealing With Overlaps Between Subspaces

The OLRSC method is based on the LRR and matrix factorisation algorithm. Typically, batch LRR comes from the fact that each point in a subspace can be represented by a linear combination of several other points within the same subspace, but the spatial distances between points are not considered for clustering. In [27], Hu et al. considered the effect of spatial distance for clustering. They introduced a penalty related to the spatial distance to the loss function for clustering, and obtained improved clustering results. This gives us the intuition that, the performance of OLRSC could be improved by considering the spatial distance.

In addition, the effectiveness of batch LRR is based on an assumption that subspaces are independent [14], i.e. the sum of dimensions of subspaces is equal to the dimension of the space spanned by all the subspaces. Since OLRSC is an online implementation of the batch LRR, the effectiveness of OLRSC also relies on the subspace independence. Once there are intersections between subspaces, the low-rank representation may not be reliable. Clustering performance of OLRSC can be adversely affected by this. As shown in Fig. 1 (a), when the data points in two subspaces are clustered by OLRSC, some points within one subspace (blue points) are clustered into another (red points). This is because subspace clustering is only based on subspace distributions, regardless of the spatial distance.

To address this problem, we introduce the spatial distance to further improve subspace clustering. In [27], a constraint based on the spatial closeness of points is used, in order to enforce neighbouring points to be partitioned into the same cluster. In such method, the k nearest neighbour graph is generated, representing the pairwise closeness. Their experiments show that considering the spatial closeness can be beneficial for improving the clustering performance. However, this method cannot be easily employed with the OLRSC algorithm, because the k nearest neighbour can hardly be extracted in an online manner. Nevertheless, the traditional clustering algorithm, such as k-means, utilises the spatial distances to partition each data point into the cluster that contains the nearest centroid to the point. As mentioned in Section 2.2, the OLRSC method employed the online k-means on $\boldsymbol{v}$. Accordingly, the k-means can be easily implemented on the data $\boldsymbol{x} \in \mathbb{R}^D$. As the k-means can be applied with both $\boldsymbol{x}$ and $\boldsymbol{v}$, once a point is attached to S_k, both $\boldsymbol{x}$ and $\boldsymbol{v}$ are attached to S_k. Therefore, we propose the following joint objective

$$\min_{S} \sum_{k=1}^{K} \sum_{\boldsymbol{x},\boldsymbol{v}\in S_k} \|\boldsymbol{v}-\boldsymbol{p}_k\|_2^2 + \lambda_o \sum_{k=1}^{K} \sum_{\boldsymbol{x},\boldsymbol{v}\in S_k} \|\boldsymbol{x}-\boldsymbol{q}_k\|_2^2 \quad (7)$$

where λ_o considers the trade-off between k-means for $\boldsymbol{v}$ and $\boldsymbol{x}$. The associated cost can be re-written as follows

$$\begin{aligned} f(\boldsymbol{v},\boldsymbol{x}) &= \sum_{k=1}^{K} \sum_{\boldsymbol{x},\boldsymbol{v}\in S_k} \|\boldsymbol{v}-\boldsymbol{p}_k\|_2^2 + \lambda_o \sum_{k=1}^{K} \sum_{\boldsymbol{x},\boldsymbol{v}\in S_k} \|\boldsymbol{x}-\boldsymbol{q}_k\|_2^2 \\ &= \sum_{k=1}^{K} \sum_{\boldsymbol{x},\boldsymbol{v}\in S_k} \left\| \left[\boldsymbol{v}^\top \quad \sqrt{\lambda_o}\boldsymbol{x}^\top\right]^\top - \left[\boldsymbol{p}_k^\top \quad \sqrt{\lambda_o}\boldsymbol{q}_k^\top\right]^\top \right\|_2^2 \end{aligned} \quad (8)$$

which can be plugged into (7). Based on this, the online k-means will be applied to $\left[\boldsymbol{v}^\top \quad \sqrt{\lambda_o}\boldsymbol{x}^\top\right]^\top$ in our system, rather than the $\boldsymbol{v}_t$ of (5) in OLRSC. We call this approach as Joint OLRSC (JOLRSC). Fig. 1 (b) shows an example of the clustering result obtained by JOLRSC, with a significant improvement over the baseline OLRSC.

3.2. A Warm-Up Strategy to Speed up Convergence

In practice, the matrix $\boldsymbol{G}$ is initialised randomly, the clustering accuracy for the early input data points can be quite poor, since the matrix $\boldsymbol{G}_{t+1}$ at time $t+1$ can be very different from $\boldsymbol{G}_t$ at time t. As $\boldsymbol{v}_t$ is derived from $\boldsymbol{G}_t$, $\boldsymbol{v}_t$ is not a reliable feature for clustering if $\boldsymbol{G}$ has still to converge. When the data volume is large, the average accuracy will not be much affected. However, if the early input data are important, vital information may be lost because of the poor performance with the early data.

One solution would be to pass sufficient number of data points through the OLRSC system, to obtain a converged $\boldsymbol{G}$ as the initial value. However, we observed that for the acoustic dataset the convergence could be slow, and as a result, this incurs a large extra memory cost. Otherwise, we can pass a smaller subset of data through the OLRSC system, and predict the converged value based on the trend of $\boldsymbol{G}$. In our experiment, we initialise the i,jth element of $\boldsymbol{G}$ as

$$\boldsymbol{g}_{ij} = \boldsymbol{g}_{ij}(M)^{L+1} / \boldsymbol{g}_{ij}(0)^{L} \quad (9)$$

where $\boldsymbol{g}_{ij}(0)$ is a random matrix, and then update it with a small subset of data, which have M inputs. The updated $\boldsymbol{g}_{ij}$ is denoted by $\boldsymbol{g}_{ij}(M)$. The constant L is relevant to the number of clusters K. The predicted $\boldsymbol{G}$ is close to the converged $\boldsymbol{G}$ in terms of magnitude, and can converge quickly. In this way, $\mathcal{O}(MD)$ memory is needed to store the subset of data, much less than $\mathcal{O}(ND)$ to store the whole dataset ($M \ll N$). In practice, $M < r$, thus the memory cost remains to be $\mathcal{O}(Dr)$ in total. Apart from the memory cost, the extra computational cost is only slightly increased. Our experiments show that this approach is effective in improving the convergence.

4. EXPERIMENTAL EVALUATIONS

4.1. Dataset and Performance Index

Since our objective is to cluster a set of unlabelled audio data, in our experiments, we employed the TUT Acoustic Scenes 2017 dataset [28, 29] to evaluate our algorithm. The acoustic signals were recorded from 15 different urban scenes with two channels. Recording environments include: lakeside beach, bus, cafe/restaurant, car, city center, forest path, grocery store, home, library, metro station, office, urban park, residential area, train, and tram. Each scene has 312 audio clips of 10 seconds. The original sampling rate is 44100Hz. The performance of each algorithm is estimated by the clustering accuracy, defined as the ratio of the number of correctly clustered data to the number of data points in the whole set.

4.2. Feature Extraction and Pre-processing

Mel-Frequency Cepstral Coefficients [30] are derived. Audio clips are re-sampled to $16kHz$. Then they are filtered by a mel-filter bank after taking a Fast Fourier Transform. The frame length and hop size are set to 256 and 160, respectively. Twelve DCT coefficients of the logarithm are retained. We took the 1^{st} and 2^{nd} order time differential derivatives and obtained 36 features per frame. For each clip, 995 frames are obtained. Therefore, for each audio clip, we have a 36×995 feature matrix, which is then reshaped into a vector of 35820 elements and used as the data vector.

4.3. Experimental setting of JOLRSC

For the TUT Acoustic Scenes 2017 dataset, we have $K = 15$ different categories, and $N = 4680$ audio clips. Each data vector has $D = 35820$ dimensions. We set the basis matrix $\boldsymbol{A}$ equal to the data matrix $\boldsymbol{X}$, $r = 20K$, $M = 10K$, and $L = 20K$. Experiments have been carried out for 50 trials.

4.4. Initialisation of OLRSC

As mentioned in Section 3.2, the clustering accuracy with data before convergence is not good. We evaluate how the performance of the OLRSC algorithm changes with respect to the streaming acoustic data given. As shown in Fig. 2, the red line denotes the accuracy of OLRSC with the first 300 data points, and the green line denotes the accuracy after employing our proposed initialisation method. It can be seen that after approximately 100 data points, the two lines in the figure twisted. However, the clustering accuracy of the initial 100 data points is significantly improved by our approach.

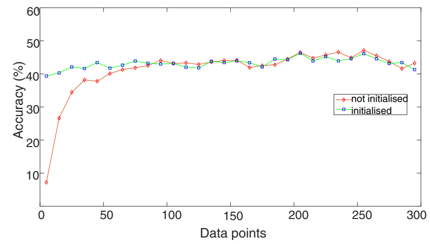


Fig. 2. Accuracy of first 300 data points, with the proposed initialisation method (green) and randomly initialised (red)

Table 1. Average accuracy (%) of subspace clustering algorithms compared on the TUT Acoustic Scenes 2017 dataset.

Methods	OLRSC-S	OLRSC-K	JOLRSC
Accuracy(%)	22.36	43.64	**45.84**
Methods	SSSC	SLRR	k-means
Accuracy(%)	25.31	31.60	38.58

4.5. Performance of approaches

We compare the proposed algorithm with two baselines OLRSC and k-means. As typical batch subspace clustering algorithms, LRR and SSC obtained good performance. However, both LRR and SSC have unaffordable computational complexity with our acoustic dataset because of the large volume. Thus, we use two scalable version of such methods, scalable SSC (SSSC) [31] and sketched LRR (SLRR) [16] instead, for the comparison with our method. For the OLRSC and JOLRSC methods, to reduce running time caused by the high dimensionality of data, we utilise the random projection algorithm [32] to reduce the data dimension from 35820 to 2000 as pre-processing. The results are presented in Table 1, where the OLRSC-S denotes the OLRSC followed by the spectral clustering to obtain the subspace memberships, while OLRSC-K denotes the OLRSC with k-means. Results of both OLRSC-S and OLRSC-K are obtained after algorithm's convergence. It is clear that our JOLRSC method outperforms other compared methods. Importantly, by introducing a joint clustering method with k-means as well as a new initialisation scheme, our method improved the performance of OLRSC. Additionally, as mentioned in Section 2.2 and Section 3.2, our method reduces the memory cost from $\mathcal{O}(N(N + D))$ of the batch LRR to $\mathcal{O}(Dr)$.

5. CONCLUSION

We have introduced a JOLRSC algorithm for clustering streaming data, by considering the closeness of data points in online spectral clustering, based on the OLRSC algorithm. We also improved the convergence of the algorithm with a new warm up scheme. The algorithm can deal with datasets of large data volume, which cannot be handled by the batch methods. Effectiveness of our approach has been confirmed by the experiments with an acoustic dataset.

6. REFERENCES

[1] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, “Acoustic scene classification: Classifying environments from the sounds they produce,” *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[2] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[3] L. Parsons, E. Haque, and H. Liu, “Subspace clustering for high dimensional data: a review,” *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[4] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, “Unsupervised segmentation of natural images via lossy data compression,” *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

[5] R. Vidal, R. Tron, and R. Hartley, “Multiframe motion segmentation with missing data using powerfactorization and GPCA,” *International Journal of Computer Vision*, vol. 79, no. 1, pp. 85–105, 2008.

[6] S. Li and W. Wang, “Randomly sketched sparse subspace clustering for acoustic scene clustering,” *European Signal Processing Conference, EUSIPCO*, 2018.

[7] G. Sally, P. Ashkan, K. Hamid, A. K. Ryan, and M. John, “Information subspace-based fusion for vehicle classification,” *European Signal Processing Conference, EUSIPCO*, 2018.

[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, 2005.

[9] Y. Ma, A. Y. Yang, H. Derksen, and R. Fossum, “Estimation of subspace arrangements with applications in modeling and segmenting mixed data,” *SIAM Review*, vol. 50, no. 3, pp. 413–458, 2008.

[10] P. Tseng, “Nearest q-flat to m points,” *Journal of Optimization Theory and Applications*, vol. 105, no. 1, pp. 249–252, 2000.

[11] P. K. Agarwal and N. H. Mustafa, “K-means projective clustering,” in *Proceedings of the Twenty-Third ACM Symposium on Principles of Database Systems*. ACM, 2004, pp. 155–165.

[12] H. Derksen, Y. Ma, W. Hong, and J. Wright, “Segmentation of multivariate mixed data via lossy coding and compression,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 6508, pp. 65 080H–65 080H, 2007.

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

[14] G. Liu, Z. Lin, and Y. Yu, “Robust subspace segmentation by low-rank representation,” in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010, pp. 663–670.

[15] C. You, D. Robinson, and R. Vidal, “Scalable sparse subspace clustering by orthogonal matching pursuit,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3918–3927.

[16] P. A. Traganitis and G. B. Giannakis, “Sketched subspace clustering,” *arXiv preprint arXiv:1707.07196*, 2017.

[17] ——, “Efficient subspace clustering of large-scale data streams with misses,” in *2016 Annual Conference on Information Science and Systems (CISS)*. IEEE, 2016, pp. 590–595.

[18] J. Shen, P. Li, and H. Xu, “Online low-rank subspace clustering by basis dictionary pursuit,” in *International Conference on Machine Learning*, 2016, pp. 622–631.

[19] R. Vidal, “Subspace clustering,” *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[20] M. Fazel, H. Hindi, and S. P. Boyd, “A rank minimization heuristic with application to minimum order system approximation,” in *Proceedings of the 2001 American Control Conference*, vol. 6, 2001, pp. 4734–4739.

[21] J. Shen, P. Li, and H. Xu, “Efficient online minimization for low-rank subspace clustering,” *arXiv preprint arXiv:1503.08356*, 2017.

[22] N. Srebro and A. Shraibman, “Rank, trace-norm and max-norm,” in *International Conference on Computational Learning Theory*. Springer, 2005, pp. 545–560.

[23] D. P. Bertsekas, *Nonlinear Programming*. Athena scientific Belmont, 1999.

[24] P. Richtárik and M. Takáč, “Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function,” *Mathematical Programming*, vol. 144, no. 1-2, pp. 1–38, 2014.

[25] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[26] S. Zhong, T. M. Khoshgoftaar, and N. Seliya, “Clustering-based network intrusion detection,” *International Journal of Reliability, Quality and Safety Engineering*, vol. 14, no. 02, pp. 169–187, 2007.

[27] H. Hu, Z. Lin, J. Feng, and J. Zhou, “Smooth representation clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3834–3841.

[28] A. Mesaros, T. Heittola, and T. Virtanen, “TUT database for acoustic scene classification and sound event detection,” in *24th European Signal Processing Conference 2016 (EUSIPCO 2016)*, Budapest, Hungary, 2016.

[29] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, “Dcase 2017 challenge setup: Tasks, datasets and baseline system,” in *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

[30] M. Sahidullah and G. Saha, “Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition,” *Speech Communication*, vol. 54, no. 4, pp. 543–565, 2012.

[31] X. Peng, L. Zhang, and Z. Yi, “Scalable sparse subspace clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 430–437.

[32] G. Li and Y. Gu, “Distance-preserving property of random projection for subspaces,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 3959–3963.