Randomly Sketched Sparse Subspace Clustering for Acoustic Scene Clustering

Shuoyang Li
Department of Electrical
and Electronic Engineering
University of Surrey
Guildford, UK
Email: shuoyang.li@surrey.ac.uk

Wenwu Wang
Department of Electrical
and Electronic Engineering
University of Surrey
Guildford, UK
Email: w.wang@surrey.ac.uk

***Abstract*—Acoustic scene classification has drawn much research attention where labeled data are often used for model training. However, in practice, acoustic data are often unlabeled, weakly labeled, or incorrectly labeled. To classify unlabeled data, or detect and correct wrongly labeled data, we present an unsupervised clustering method based on sparse subspace clustering. The computational cost of the sparse subspace clustering algorithm becomes prohibitively high when dealing with high dimensional acoustic features. To address this problem, we introduce a random sketching method to reduce the feature dimensionality for the sparse subspace clustering algorithm. Experimental results reveal that this method can reduce the computational cost significantly with a limited loss in clustering accuracy.**

I. Introduction

Much research attention has been given to acoustic scene classification recently [1], which has potentially wide applications such as audio surveillance. Audio signals from different scenes are recognized by classification approaches trained with tags. However, there are two difficulties in audio data classification. Firstly, acoustic databases are not always correctly labeled, which brings difficulties to model training procedures. Unsupervised clustering algorithms could help detect wrongly labeled data by comparing the clustering result with the labels. Secondly, labeled data are not always available, and supervised methods are no-longer useful. For example when a large quantity of raw acoustic data need to be labeled for the environments in which they are recorded, and manual labelling is not always reliable and feasible with massive data. Acoustic scene clustering, as an unsupervised method, is needed in such scenarios.

Subspace clustering is a powerful technique for learning class labels in high-dimensional data in an unsupervised manner. A high dimensional dataset can be viewed as lying in a low-dimensional intrinsic subspace [2], and subspace clustering methods aim at grouping them according to their intrinsic subspace distributions. Subspace clustering has drawn increasing research attention recently, and induced solutions towards problems with high-dimensional data [3], such as image segmentation [4] and motion segmentation [5]. A number of subspace clustering algorithms have been developed such as Generalised PCA [6], Agglomerative Lossy Compression [7] and Low-Rank Representation [8].

Recently, spectral clustering based subspace clustering methods attract increasing attention. Because they can be solved by standard linear algebraic methods and clustering results outperform other traditional clustering approaches [9]. Unlike clustering algorithms such as k-means [10] and fuzzy clustering [11] which rely on the distances between data points and centroids of the clusters, spectral clustering-based methods utilise distances or correlations among points, without regard to the centroids. In spectral clustering based algorithms, an affinity matrix is constructed to exploit the relativeness among data samples, then spectral clustering [9] is implemented by their Laplacian Graph, with segmentation strategies such as ratio cut [12] and normalised cut [13].

Sparse Subspace Clustering (SSC) [14], an approach based on spectral clustering, was shown to be a promising algorithm for subspace clustering, with good clustering accuracy and robustness to noise, and therefore is our focus here.

Features of audio sequences are usually of high dimension. To reduce the computational cost with the high dimensional datasets, a dimensionality reduction process can be performed. Conventional method like Principal Component Analysis [15] includes operations on covariance matrix and eigenvalue decomposition, and still has high computational complexity with high dimensional audio data.

In this paper, a random sketching (RS) method is proposed to address the high computational burden of SSC. The whole system is easy to implement and with low computational cost. Experimental results on unlabeled acoustic data reveal that with a proper sketching rate, the clustering accuracy of sparse subspace clustering can be retained while the computing-time is reduced significantly.

II. Background

The SSC algorithm and some complexity reduction methods for SSC are reviewed in this section.

A. SSC

SSC is a spectral clustering based algorithm, where an affinity matrix determining the similarities among data points is needed to build up the graph Laplacian [9].

ISBN 978-90-827970-1-5 © EURASIP 2018

Based on the observation that a data point $\boldsymbol{x}_i$ lies in a d_i-dimensional subspace can be represented by a linear combination of d_i other points in general directions from the same subspace, SSC tries to represent each data point as a sparse combination of other data points, and assume that the sparse combination can guarantee that data points representing $\boldsymbol{x}_i$ lie in the same subspace as $\boldsymbol{x}_i$. In [16] and [17], the condition and bound of this assumption were provided. To generate the affinity matrix for spectral clustering, SSC seeks for the sparse representation of each data point, and estimate the correlation among points by the coefficients of the sparse representation.

In other words, let $\{\boldsymbol{x}_i\}_{i=1}^N$ denote data points embedded in a subspace with lower dimension than the ambient space, each data point $\boldsymbol{x_i}$ can be expressed as a linear combination of other points in the same subspace,

$$\boldsymbol{x}_i = \sum_{j=1}^{n} c_j \boldsymbol{x}_j \tag{1}$$

where $\boldsymbol{x}_j$ denotes a point in the same subspace as $\boldsymbol{x}_i$. Data matrix $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ is set as the dictionary matrix [14]. SSC seeks for the sparse representation of each point by the others in the set, for example, by the following objective function,

$$\arg\min_{\boldsymbol{C}} \|\boldsymbol{C}\|_0 \quad s.t. \quad \boldsymbol{X} = \boldsymbol{X}\boldsymbol{C}, \, diag(\boldsymbol{C}) = 0 \tag{2}$$

where $\boldsymbol{C}$ is the coefficient matrix for the sparse representation. Because solving the ℓ_0-norm based sparse optimization is NP-hard, the ℓ_0-norm is often relaxed to ℓ_1-norm, then $\boldsymbol{C}$ can be obtained by optimizing the following cost function,

$$\arg\min_{\boldsymbol{C}} \|\boldsymbol{C}\|_1 + \frac{\lambda}{2}\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_F^2 \tag{3}$$

This objective can be optimized with Alternating Direction Method of Multipliers (ADMM) method [18].

To guarantee the symmetry, the affinity matrix for spectral clustering algorithm which can reveal neighbourhood relationship among points is obtained by $\boldsymbol{E} = |\boldsymbol{C}| + |\boldsymbol{C}^T|$, where $|\cdot|$ takes the modulus of each element of the matrix. Afterwards, clustering results can be obtained by performing spectral clustering on the affinity matrix $\boldsymbol{E}$.

SSC is simple to implement, and has good clustering accuracy as well as robustness to noise and outliers [19], [16], [17]. However, its computational complexity increases sharply with the growth of data dimension and data size. Thus it is vital to perform a dimensionality reduction process before SSC is applied in this scenario.

B. Computational Simplification Methods for SSC

The time complexity of a typical SSC method [16] with ADMM solver is $\mathcal{O}(N^3 + N^2 D)$, in which N is the total number of data-points to be clustered and D is the dimensionality of data. In our case, the dimensionality of features of an audio clip is very high ($D \gg N$), so the time complexity is approximated to $\mathcal{O}(N^2 D)$.

Algorithms have been proposed to accelerate the computation procedure of SSC. Peng et al. [20] introduced a scalable-SSC framework to simplify the computation of SSC by decreasing the size of the affinity matrix. The algorithm randomly picks out some data out of the whole database as in-sample data. The in-sample data are clustered by SSC. Other out-of-sample data are classified by a classification method to existing clusters. The scalable-SSC is appropriate for data set of massive number of points. Nevertheless, this method includes a data classification procedure based on sparse representation, which introduces additional computational cost.

Traganitis and Giannakis [21] introduced a sketching method based on Johnson-Lindenstrauss transform (JLT, e.g., multiply the data matrix by a Gaussian random matrix). In the algorithm, the optimization equation becomes

$$\arg\min_{\boldsymbol{A}} \|\boldsymbol{A}\|_1 + \frac{\lambda}{2}\|\breve{\boldsymbol{X}} - \breve{\boldsymbol{B}}\boldsymbol{A}\|_F^2 \tag{4}$$

In which $\breve{\boldsymbol{X}}$ is the compressed matrix transformed from $\boldsymbol{X}$ by JLT, and has fewer rows than $\boldsymbol{X}$. $\breve{\boldsymbol{B}}$ is the dictionary matrix compressed from $\breve{\boldsymbol{X}}$, and has fewer columns than $\breve{\boldsymbol{X}}$. After solving (4), the affinity matrix for SSC is constructed according to the neighbourhood relationships of $\boldsymbol{A}$.

Traganitis et al. [3] proposed a sketching and validation algorithm. The algorithm randomly picks out a subset $\tilde{\boldsymbol{X}}$ with n columns from the original dataset $\boldsymbol{X}$. SSC is then performed on $\tilde{\boldsymbol{X}}$. Performance of SSC is then estimated by kernel density estimation [22]. This is based on the intuition that the underlying probability density function (pdf) of $\tilde{\boldsymbol{X}}$ is expected to be multi-modal. Thus the discrepancy of this pdf is compared with a uni-modal pdf. The random picking is performed repeatedly and the one with the largest discrepancy is used for SSC. The complementary set of $\tilde{\boldsymbol{X}}$ is associated with clusters by the residual minimization method [20] or by finding the closest subspace.

Mao and Gu [23] introduced a random compression method to reduce the data dimensionality by a random projection procedure. The data matrix $\boldsymbol{X}$ is compressed after being multiplied by a Gaussian random matrix.

Because the main challenge of the audio dataset here is the overwhelmingly high dimensionality D, the algorithms proposed by [20] [3] focus on reducing the size of N in computation and are inappropriate for the audio dataset. The algorithms in [21] [23] reduce the dimensionality D by Gaussian random projection, and are tested in our experiment. The experimental result shows that SSC with our random sketching algorithm outperforms the Gaussian random projection method in terms of accuracy.

III. A Randomly Sketched SSC

Different from the above mentioned works, we introduce a new method for dimensionality reduction with a random sketching strategy. To explain the idea, we show an example of data distribution in Fig. 1.

In Fig. 1 (a), two clusters of points (red and blue) lie in a 3-dimensional space spanned by three axes ($\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$).

ISBN 978-90-827970-1-5 © EURASIP 2018

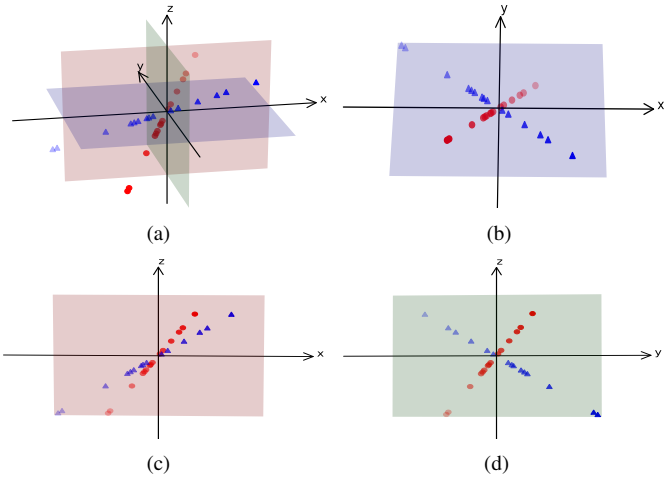

Fig. 1. (a) Points of two groups in a 3-D space. (b)(c)(d) Two dimensions of the ambient space are retained by abandoning information of the third dimension. It can be observed that these points are still splittable.

In (b)(c)(d), by projecting all points into three 2-dimensional planes, which are spanned by axes $\boldsymbol{xy}$, $\boldsymbol{xz}$, and $\boldsymbol{yz}$, we can see that points are still splittable on each plane. It indicates that in this case, by retaining 2 dimensions and abandoning the $3rd$ dimension, points can still be clustered by the SSC. This observation gives us the intuition that, high dimensional SSC can be accelerated by taking a random sketching of the dimensions. By sketching a large enough size of original dimensions, SSC can be implemented with a good performance, and computational time cost can be saved simultaneously.

The random sketching algorithm aims to pick out a union of rows from a data matrix $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ randomly, and these rows are then ranked. Assume d number of rows are to be sketched out, and $\boldsymbol{F} \in \mathbb{R}^{d\times D}$ is the random sketching matrix, then

$$\boldsymbol{F} = \boldsymbol{Q}\boldsymbol{P} = \begin{bmatrix} 1 & \dots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & 0 & \cdots & 0 \end{bmatrix} \boldsymbol{P} \tag{5}$$

where the left part of $\boldsymbol{Q}$ is an identity matrix of $d \times d$ and the right part of $\boldsymbol{Q}$ is a zero matrix of $d \times (D-d)$. $\boldsymbol{P}$ is a random permutation matrix of order D.

The procedure used to sketch a data point $\{\boldsymbol{x}_i\}_{i=1}^N$ is denoted by

$$\boldsymbol{w}_i = \boldsymbol{F}\boldsymbol{x}_i \tag{6}$$

where $\{\boldsymbol{w}_i\}_{i=1}^N$ is the sketched data vector. Equivalently, the procedure of random sketching on dataset $\boldsymbol{X}$ is denoted by

$$\boldsymbol{W} = \boldsymbol{F}\boldsymbol{X} \tag{7}$$

where $\boldsymbol{W}$ is the sketched data matrix, and $\boldsymbol{W}$ is then used as the input of the SSC.

With random sketching, a union of rows are selected from the data matrix $\boldsymbol{X}$ randomly, to generate a new data matrix $\boldsymbol{W}$, which retains some features of $\boldsymbol{X}$ with a lower dimension. With an appropriate rate of sketching, the preserved features could be sufficient for subspace clustering algorithms to detect different subspaces. It is easy to imagine that sketching higher-dimensional features obtains a higher clustering accuracy with higher time cost, or vice-versa.

Unlike algorithms mentioned in Section II-B, our algorithm focuses only on dimensional reduction because the overwhelmingly high dimension of the acoustic data is the main challenge here. Moreover, the RS-SSC algorithm takes a random sketch of the original data without the formation of a Gaussian random matrix. The biggest advantage of the proposed random sketching algorithm is that it is rather simple to implement, has low computational cost, and enjoys good performance with the acoustic scene data. To minimise the time cost and guarantee accuracy simultaneously, we need to study the feature-preserving ability of random sketching. In this paper, we study the performance of random sketching algorithm for acoustic scene clustering to be presented later.

A. Practical implementation

By applying the SSC on an acoustic scene dataset, we observed that the subspace distribution changes from scene to scene and the performance of SSC is unstable with different scenes. To address this, a scheme which can adjust automatically the parameters in SSC is proposed.

According to [16], the λ in (3), which serves as the trade off between the sparsity constraint and the fitting error, can be obtained from

$$\lambda = \frac{\alpha}{\min\limits_i \max\limits_{j\neq i} |\boldsymbol{x}_i^T \boldsymbol{x}_j|} \tag{8}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are arbitrary column vectors from the data matrix $\boldsymbol{X}$, and α is a constant larger than 1, which is used to balance the optimization result.

Empirically, abandoning some trivial elements in the self-representing coefficient matrix $\boldsymbol{C}$ can improve the performance of SSC. Hence, a threshold value $\rho \in (0, 1]$ can be set to filter out some small elements in $\boldsymbol{C}$. Assume that the sum of modulus of each element in a column vector $\boldsymbol{c}_i$ is sum_i. We can retain the elements of the largest modulus that sum up to $\rho \cdot sum_i$ while discarding the remaining elements that are regarded as trivial based on the choice of ρ.

To evaluate the SSC performance with a certain pair of α and ρ, we compare the clustering result with a "ground truth". The "ground truth" is obtained from the clustering result of the k-means algorithm, which is suitable with low-dimensional data hence it was implemented with frame-based features. Features for the whole chunk of audio data is high-dimensional and k-means does not perform well on them. Whether an audio clip belongs to a cluster is decided by "majority voting". That is, the cluster-belonging is determined based on the cluster which occupies the largest proportion of frames. Intuitively, an audio clip with a greater majority of frames of one cluster is more likely to belong to this cluster than another with a relatively lower proportion of the majority frames. Experimental results support this intuition. So we pick out several frames with the highest majority proportion as "ground truth", then adjust parameters accordingly.

ISBN 978-90-827970-1-5

IV. Evaluations

In this section we present the simulation results of RS-SSC for audio scene clustering. The performance of RS-SSC is compared with the Gaussian Random Projection SSC method.

A. Database and performance index

TUT Acoustic Scenes 2017 data set [24], [25], which was used as the dataset for task1 of DCASE 2017 challenge is chosen for the evaluation of our algorithm. The acoustic signal was 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. Original sampling rate is 44100Hz.

The performance of each algorithm is estimated by error rate. Additionally, the time cost of the sketching algorithm is estimated by *Relative time cost*, which is defined as

$$\frac{\text{time cost of RS-SSC}}{\text{time cost with full dimensional data}} \tag{9}$$

B. Feature extraction

Mel-Frequency Cepstral Coefficients (MFCC) [26] are derived from the original database. Audio clips are re-sampled to $16kHz$. Then they are filtered by mel-filter bank of 24 filters after taking a Fast Fourier Transform (FFT) of length 256. Frame length is set as 256 samples and the hop size is set as 160 samples. Twelve DCT coefficients of the logarithm are kept. As differential and acceleration coefficients could improve the recognition performance of MFCC [27], we took the 1^{st} and 2^{nd} order time differential derivatives. As a result, 36 features per frame are obtained.

The features for original data clips are frame-based. 995 frames are obtained and each frame has 36 features. Those 36×995 features are the input of the SSC system.

C. SSC and dimensionality reduction

For each audio clip, the input data for SSC has a dimension of 36×995. Conventionally, this is transformed into a vector of 35820 elements, which however brings heavy computational cost for SSC because of its high dimension. To simplify the computation, we used the random sketching based SSC (RS-SSC), which is compared with the Gaussian random projection based SSC (GRP-SSC) [23]. In this section, fixed parameters are used (α=300, ρ=0.7) to save the computational time.

The obtained 35820 dimensional features are used as input data points. In total, there are 15 subjects (scenes). Each subject has 312 data points which are the audio clips in the raw data. We take out 200 feature points from 4 subjects (50 points from each subject), then test the performance in terms of the error rate.

Data samples are randomly picked out from the original database, and the whole process is run for 200 trials. We take the average of the error-rate. The reduced dimensionality changes from 1000 to 12000.

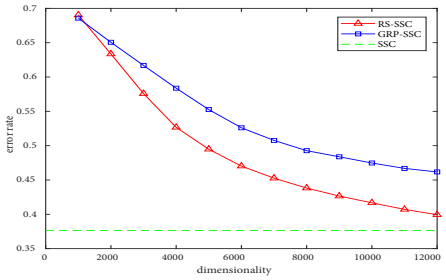

Fig. 2. Error-rate (%) of RS-SSC, GRP-SSC, and SSC with database of 4 subspaces. Dimensionality of RS-SSC and GRP-SSC changes from 1000 to 12000.

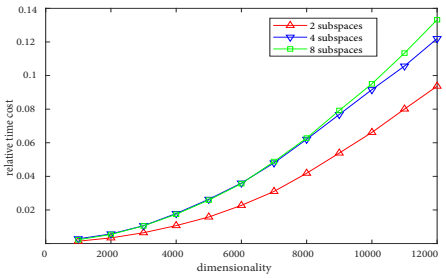

Fig. 3. Relative time cost of RS-SSC with database of 2 subspaces, 4 subspaces and 8 subspaces. Dimensionality changes from 1000 to 12000.

As presented in Fig. 2, RS-SSC obtains lower error rate than GRP-SSC with this dataset. Their computational complexities are the same. When the sketched dimensionality comes to 12000, the accuracy of RS-SSC is close to SSC with full-dimensional data, while saving a lot computational time simultaneously.

By taking out 50 audio clips from each scene as input data points, the relative time costs of RS-SSC of 2 subspaces, 4 subspaces and 8 subspaces are observed and presented by Fig. 3. When the sketched dimensionality is below 10000, more than 90% of the computational time is saved.

We evaluate the performance of RS-SSC on this dataset with 2, 4, and 8 subjects. Each subject has 50 data points. The reduced dimensionality is set as 10000. Simulation results are presented in Table I.

The error-rate of SSC for full dimensionality on 2 subspaces is 14.9%. Using the RS-SSC algorithm, with the sketched dimensionality set as 10000, 93% of the computational time can be saved with only a low loss in clustering accuracy.

D. Improve the performance by parameter-tunning

In the implementation process of SSC, many parameters need to be adjusted to achieve a fine accuracy. However, attributes of acoustic data vary from scene to scene, which

TABLE I
Average error-rate (%) of RS-SSC for 2, 4, and 8 subjects.

subjects number	2	4	8
clustering error-rate	15.67	40.58	69.78

ISBN 978-90-827970-1-5 © EURASIP 2018

TABLE II
AVERAGE ERROR-RATE (%) OF RS-SSC WITH DIFFERENT SCENES FOR 2 SUBJECTS. WITH PARAMETERS FIXED OR ADJUSTED.

Scenes	beach	bus	cafe	car	city
Fixed	16.76	12.54	20.64	5.10	15.01
Adjusted	14.74	11.82	19.11	4.99	13.92
Scenes	forest	grocery	home	library	metro
Fixed	10.52	13.53	17.37	17.26	17.54
Adjusted	10.38	12.02	14.95	16.92	16.55
Scenes	office	park	residential	train	tram
Fixed	20.72	16.04	16.95	21.55	13.57
Adjusted	18.87	15.80	13.66	21.04	12.50

means proper values of parameters change significantly with different scenes.

We utilize the parameter updating method discussed in Section III-A to improve the performance of RS-SSC, as shown in Table II where the results obtained by using fixed parameters (α=300, ρ=0.7) are also given. In our experiment, 50 points from each subspace are taken. The result reveals that the parameters could be optimized to improve the performance of RS-SSC considerably.

V. CONCLUSION

We have presented a random sketching algorithm to reduce the dimensionality of audio features for scene classification. This has significantly reduced the computational cost of SSC while maintaining its performance. The RS-SSC has the same time cost as the GRP-SSC, while performing better in terms of clustering results.

Our future work will include the development of a theoretical frame of the proposed random sketching algorithm, and the evaluation of the algorithm for other datasets, such as AudioSet [28].

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

[3] P. A. Traganitis, K. Slavakis, and G. B. Giannakis, "Large-scale subspace clustering using sketching and validation," *arXiv preprint arXiv:1510.01628*, 2015.

[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] 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.

[7] 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. 65080H–65080H, 2007.

[8] 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.

[9] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[10] 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.

[11] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[12] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, vol. 11, no. 9, pp. 1074–1085, 1992.

[13] 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.

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

[15] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[16] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[17] C. You and R. Vidal, "Geometric conditions for subspace-sparse recovery," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1585–1593.

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., "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.

[19] M. Soltanolkotabi, E. J. Candes, et al., "A geometric analysis of subspace clustering with outliers," *The Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2012.

[20] 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.

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

[22] M. P. Wand and M. C. Jones, *Kernel Smoothing*, Crc Press, 1994.

[23] X. Mao and Y. Gu, "Compressed subspace clustering: A case study," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2014, pp. 453–457.

[24] 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.

[25] 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.

[26] 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.

[27] X. Huang, A. Acero, F. Alleva, M. Hwang, L. Jiang, and M. Mahajan, "From sphinx-ii to whispermaking speech recognition usable," in *Automatic Speech and Speaker Recognition*, pp. 481–508. Springer, 1996.

[28] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

ISBN 978-90-827970-1-5 © EURASIP 2018