Transformed Locally Linear Manifold Clustering

Jyoti Maggu
IIIT Delhi
jyotim@iiitd.ac.in

Angshul Majumdar
IIIT Delhi
angshul@iiitd.ac.in

Emilie Chouzenoux
LIGM, UMR CNRS 8049, UPEM
emilie.chouzenoux@univ-mlv.fr

***Abstract*— Transform learning is a relatively new analysis formulation for learning a basis to represent signals. This work incorporates the simplest subspace clustering formulation – Locally Linear Manifold Clustering, into the transform learning formulation. The core idea is to perform the clustering task in a transformed domain instead of processing directly the raw samples. The transform analysis step and the clustering are not done piecemeal but are performed jointly through the formulation of a coupled minimization problem. Comparison with state-of-the-art deep learning-based clustering methods and popular subspace clustering techniques shows that our formulation improves upon them.**

Keywords*— *subspace clustering, transform learning, alternating optimization

I. INTRODUCTION

The problem of clustering is well known. It studies how signals are naturally grouped together. One of the best-known application of clustering is image segmentation, where there is no labelled data available and one must distinguish between the background and foreground. Perhaps the simplest and most widely used clustering technique is the K-means [1]. It groups the samples such that the total distance of the data points within the cluster are minimized. The problem is NP hard, and hence is solved greedily.

One of the limitations of K-means is that it operates on the raw data and hence fails to capture non-linear relationships. This can be simply fixed by resorting to the kernel K-means [2], where the standard Euclidean distance between the samples typically used in K-means is replaced by a kernelized version of it.

Spectral clustering [2, 3] extends the kernel K-means strategy by replacing the kernelized data matrix by a so-called affinity matrix. This allows to generalize the kernel metric to any similarity measure.

Subspace clustering techniques [4], is a special class of spectral clustering approach which assumes that the samples from the same cluster lie in the same subspace. In practice, it requires to express each data point as a linear combination of the other data points. The associated linear weights then serve as inputs for creating the affinity matrix.

In the past, it has been shown [5] that instead of applying subspace clustering on the raw data, a projection space can be learnt such that the clustering is carried out in the projected domain. For instance, in [5] a tight-frame operator was learnt from the data along with the subspace clustering formulation.

In this work, we propose to adopt a similar concept as in [5], that is to perform subspace clustering in a transformed space, with the aim to obtain clusters with more useful features thanks to the transform step. Indeed, raw data have many irrelevant dimensions that could mask existing clusters in noisy data. Transform learning is thus expected to help in removing irrelevant and redundant dimensions of high-dimensional data. For improved versatility, we propose to replace the tight-frame transform from [5] by a more general linear transform operator, as it was done in [6]. A subspace clustering strategy based on Locally Linear Manifold Clustering (LLMC) is then incorporated in our transform learning framework and the ensuing estimation problem is solved jointly by means of an alternating minimization algorithm.

We have compared our technique with state-of-the-art deep sparse subspace clustering [7] and active orthogonal matching pursuit-based subspace clustering [8]. Although both studies are very recent and show the best state-of-the-art results, our method outperforms them by a considerable margin by all known clustering metrics in our experiments.

The paper is organized into five sections. A brief review of subspace clustering and transform learning will be given in the Section II. Our proposed formulation and minimization scheme are introduced in Section III. The experimental results will be described in Section IV. Conclusions of this work and future directions of research will be discussed in Section V.

II. LITERATURE REVIEW

A. Subspace clustering

Subspace clustering is an extension to the basic clustering technique which clusters high dimensional data that lie in union of several low-dimensional subspaces. Subspace clustering techniques such as locally linear manifold clustering (LLMC) [9], sparse subspace clustering (SSC) [10] and low rank representation (LRR) [11] express the samples as a linear combination of other samples. It tries to find clusters in different subspaces of the same dataset. Each data point is expressed as a linear combination of the other data points of the dataset. This is expressed as,

$$(\forall i \in \{1, \dots, n\}) \quad x_i = X_{i^c} c_i \tag{1}$$

Here above, for every $i \in \{1, \dots, n\}$, $x_i \in \mathbb{R}^m$ denotes the i^{th} sample, $X_{i^c} \in \mathbb{R}^{m \times n-1}$ gathers all the other samples column-

ISBN 978-90-827970-1-5 © EURASIP 2018

wise and $c_i \in \mathbb{R}^{n-1}$ states for the corresponding linear weight vector.

In subspace clustering techniques, the general learning formulation is expressed as follows,

$$(\forall i \in \{1, \dots, n\}) \quad \min_{c_i \in \mathbb{R}^{n-1}} \|x_i - X_{i^c} c_i\|_2^2 + R(c_i) \qquad (2)$$

where R is a regularization function. Depending on its nature, several formulations can be obtained. For LLMC, there is no regularization, i.e. $R=0$. For sparse subspace clustering, R is a sparsity promoting penalty, such as l_1-norm [10] or l_0 pseudo-norm [8]. For LRR, R is a low-rank penalty usually taking the form of a nuclear norm.

Let us define the coefficient matrix $C = [\check{c}_1 \ |\dots|\ \check{c}_n] \in \mathbb{R}^{n \times n}$, where, for every $i \in \{1, \dots, n\}$, $\check{c}_i \in \mathbb{R}^n$ is a vector with its i-th entry equals to 0, and the remaining $n-1$ entries equals to c_i. Once C is obtained for all the n samples by resolution of (2), an affinity matrix $A \in \mathbb{R}^{n \times n}$ needs to be computed. Such matrix defines the similarity (inverse distance) between the samples and hence by applying some sort of cut (eg, N-Cut), allows to segment the clusters. Several variants have been proposed for the definition of the affinity matrix [4]. For example, one option can be:

$$A = |C| + |C^T| \qquad (3)$$

This is usually used in SSC.

Another option, retained in LRR, is to form the affinity matrix from the scaled left singular values of C. Since C is low rank, its skinny SVD reads $C = USV^T$. The affinity matrix is generated from scaling the left singular vectors by the corresponding square rooted singular values, such that for every $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, n\}$,

$$A_{ij} = \left(\left[\tilde{U} \tilde{U}^T \right]_{ij} \right)^2 \qquad (4)$$

with $\tilde{U} = US^{1/2}$.

Yet another way to generate the affinity matrix (usually for LLMC) is by:

$$A = C + C^T - C^T C \qquad (5)$$

Once the affinity matrix is defined (by using any suitable formula), one needs to segment the clusters. Usually spectral clustering algorithm (Normalized-Cuts) is used for this purpose.

In [5] a variant of subspace clustering was proposed. They learnt a projector $P \in \mathbb{R}^{s \times m}$, $s < m$, jointly with the sparse subspace clustering step. The estimates of P and C are obtained by solving the following joint minimization problem:

$$\min_{P,C} \|PX - PXC\|_F^2 + \alpha \|C\|_1 + \beta \left\|X - P^T PX\right\|_F^2$$
$$s.t.\ diag(C) = 0 \text{ and } P^T P = I \qquad (6)$$

with I the identity matrix. Note that the equality constraint $diag(C)=0$ ensures that the samples are not represented by themselves. The weights α and β are positive regularization parameters.

B. Transform Learning

Dictionary learning is a well-studied topic, but transform learning is relatively new. Hence, we discuss it briefly for the ease of the reader. In short, it can be viewed as the analysis equivalent of dictionary learning. In dictionary learning, a basis is learnt such that it synthesizes the data from the learnt coefficients. Transform learning analyses the data by learning a basis to produce sparse coefficients. Mathematically this is expressed as,

$$TX = Z \qquad (7)$$

Here $T \in \mathbb{R}^{m \times m}$, represents the transform (i.e. analysis) basis, $X \in \mathbb{R}^{m \times n}$ gathers the data and $Z \in \mathbb{R}^{m \times n}$ are the corresponding sparse coefficients within the transformed domain. Transform learning has been largely used for solving inverse problems in signal processing. There are only a handful of studies where it has been used for machine learning problems. In [14] supervised versions of transform learning have been proposed. In [15], a kernelized version of transform learning has been proposed for unsupervised feature extraction.

The following analysis sparse coding formulation was proposed in [6]:

$$\min_{T,Z} \|TX - Z\|_F^2 + \lambda(\|T\|_F^2 - \text{logdet}T) + \mu\|Z\|_1 \qquad (8)$$

The term $-\text{logdet}T$ imposes a full rank on the learned transform to prevent the degenerate solution ($T=0$, $Z=0$). The additional quadratic penalty aims at controlling scale, otherwise the log-determinant term could keep on increasing producing degenerate results in the other extreme.

In [6], an alternating minimization approach was proposed to solve Problem (8). The following two steps are alternatively performed until convergence:

$$Z \leftarrow \min_Z \|TX - Z\|_F^2 + \mu\|Z\|_1 \qquad (9)$$

$$T \leftarrow \min_T \|TX - Z\|_F^2 + \lambda(\varepsilon\|T\|_F^2 - \text{logdet}T) \qquad (10)$$

Let us remark that updating the coefficients in (9) is straightforward. It can be updated via one step of soft thresholding, expressed as:

$$Z \leftarrow \text{sign}(TX) \odot \max(0, \text{abs}(TX) - \mu) \qquad (11)$$

Here $\odot$ indicates the element-wise product.

In the seminal paper on transform learning [6], a non-linear conjugate gradient-based technique was proposed to solve the transform update in (10). In a more refined version [12], using linear algebra properties, the authors show that a closed form actually exists for the transform update, that is given below:

$$XX^T + \lambda\varepsilon I = LL^T \qquad (12)$$

$$L^{-1} XZ^T = USV^T \qquad (13)$$

ISBN 978-90-827970-1-5

$$T = \frac{1}{2}R(S + (S^2 + \lambda I)^{\frac{1}{2}})Q^T L^{-1} \quad (14)$$

An analysis for the convergence guarantees of such an alternating update algorithm can be found for instance in [13].

III. Proposed Formulation

In this work we propose to embed the simplest subspace clustering formulation (LLMC) into the transform learning formulation. The core idea is to learn the affinity matrix on the coefficient space. A naïve solution would be to learn the transform on the data and then use the coefficients as inputs for LLMC. But such a piecemeal formulation may not yield the best results. Therefore, we propose to formulate a joint solution. Mathematically, our formulation is expressed as,

$$\min_{T,Z,C} \|TX - Z\|_F^2 + \lambda\left(\|T\|_F^2 - \log\det T\right) + \mu\|Z\|_1 + \gamma\sum_i \left\|z_i - Z_{i^c}c_i\right\|_2^2 \quad (15)$$

Alternating minimization approach is used for solving (15). It can be segregated into the following sub-problems.

$$\text{P1}: \min_T \|TX - Z\|_F^2 + \lambda\left(\|T\|_F^2 - \log\det T\right)$$

$$\text{P2}: \min_Z \|TX - Z\|_F^2 + \mu\|Z\|_1 + \gamma\sum_i \left\|z_i - Z_{i^c}c_i\right\|_2^2$$

$$\text{P3}: \min_C \sum_i \left\|z_i - Z_{i^c}c_i\right\|_2^2$$

The update for P1 is the standard transform update as given by (12) – (14). We do not repeat it here.

P2 can be alternately expressed as follows:

$$\text{P2}: \min_Z \|TX - Z\|_F^2 + \mu\|Z\|_1 + \gamma\|Z(I - C)\|_F^2$$

$$\equiv \min_Z \left\|X^T T^T - Z^T\right\|_F^2 + \mu\left\|Z^T\right\|_1 + \gamma\left\|(I - C)^T Z^T\right\|_F^2$$

$$\equiv \min_Z \left\|\begin{pmatrix} X^T T^T \\ 0 \end{pmatrix} - \begin{pmatrix} I \\ \sqrt{\gamma}\,(I - C)^T \end{pmatrix} Z^T\right\|_F^2 + \mu\left\|Z^T\right\|_1$$

This is a standard l_1-norm minimization problem which can be solved efficiently, for instance using the spectral projected gradient solver from [16].

For sub-problem P3, each of the c_i's can be obtained via a pseudo-inverse operation. Once all the c_i's are obtained, they are stacked as off-diagonal column terms of a matrix to form C.

The formulation (15) is non-convex, which makes the convergence analysis of the alternating scheme quite challenging. Using [17], one can establish that the iterates of our algorithm are well defined, and that every cluster point is a stationary point of the cost function in (15). The convergence of the iterates to a stationary point could be obtained by adding a proximal relaxation to the updates [18]. More sophisticated schemes could also have been used, such as [19]. However, these changes may come at the price of an increase of the computational complexity. In practice, the method appears to have a stable behavior, and very fast convergence rate.

Once we have obtained the solution, the affinity matrix is created using (3). Spectral clustering is applied to (3) for segmentation.

IV. Experimental Results

We have compared our work with two recent studies in clustering. The first one is deep subspace clustering (DSC) [7] and the second one is orthogonal matching pursuit based sparse subspace clustering (OMP) [8]. We also compare with the piecemeal technique where features are first extracted by transform learning and then subjected to LLMC.

Fig. 1. Samples from COIL20

Fig. 2. Samples from YaleB

Experiments were carried out on the COIL20 (object recognition) [20] and Extended YaleB (face recognition) [21] datasets. For our proposed method, we do not require any feature extraction technique. However, when we applied the OMP, DSC and TL-LLMC algorithms on the raw data, very poor results were obtained. We thus chose to feed them with extracted features, based on DSIFT (dense scale invariant feature transform) and HOG (histogram of oriented gradients), reduced by PCA to a dimensionality of 300.

ISBN 978-90-827970-1-5 © EURASIP 2018

TABLE I: RESULTS ON COIL20

Method	OMP		DSC		TL-LLMC	Proposed
	DSIFT	HOG	DSIFT	HOG	Raw	Raw
Accuracy	65.36	74.93	85.76	85.50	95.83	**97.01**
NMI	.7709	.8926	**.9119**	**.9119**	.8817	.9045
ARI	.5659	.7425	.8480	.8192	.8674	**.8999**
Precision	.5147	.6665	.8245	.7912	.9222	**.9550**
F-measure	.5907	.7570	.8558	.8286	.9551	**.9744**

TABLE II: RESULTS ON YALEB

Method	OMP		DSC		TL-LLMC	Proposed
	DSIFT	HOG	DSIFT	HOG	Raw	Raw
Accuracy	82.30	84.78	88.55	92.08	92.96	**97.02**
NMI	.8754	.9343	.9085	**.9691**	.7310	.8553
ARI	.7582	.8257	.8300	.9025	.7508	**.9174**
Precision	.7090	.8586	.7952	.8507	.9634	**.9565**
F-measure	.7650	.8307	.8345	.8946	.9008	**.9060**

Since the ground truth (class labels) for both datasets is available, clustering accuracy was measured in terms of Accuracy, NMI (normalized mutual information), ARI (adjusted rand index), Precision and F-score. The results are shown in Table I (COIL20) and Table II (YaleB). The parameters are selected using grid search. All the results are averaged over ten runs.

From the above tables (I – II), one can see that our proposed method is considerably better (on average) than the rest in terms of every clustering metric. The main reason may be that the use of transform learning strategy allows to select relevant features from high dimensional dataset jointly with searching the appropriate cluster in some low-dimensional subspace of the dataset. The joint formulation reduces the chances of masking relevant clusters by noisy features and avoids the need for a preliminary step of feature extraction.

V. CONCLUSION

This work incorporates subspace clustering – specifically locally linear manifold clustering into the transform learning formulation. Results on benchmark problems show that our proposed method outperforms state-of-the-art clustering techniques.

There are several ways we plan to proceed further. First, we would like to incorporate other subspace clustering formulations like sparse subspace clustering and low rank representation into the transform learning framework. Our preliminary results seem to show that adding extra regularization terms such as sparsity and low-rank penalties indeed improves performance.

We also plan to apply the developed techniques to solve real world problems such as the analysis of gene expression datasets [22,23].

ACKNOWLEDGEMENT

This work is partially supported by DST-CNRS-2016-02, CNRS-CEFIPRA project under grant NextGenBP PRC2017 and Infosys Center for Artificial Intelligence @ IIIT Delhi.

REFERENCES

1. Hartigan, J.A. and Wong, M.A., Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), pp.100-108, 1979.
2. Dhillon, I.S., Guan, Y. and Kulis, B., Kernel k-means: spectral clustering and normalized cuts. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 551-556). ACM, August 2004.
3. Ng, A.Y., Jordan, M.I. and Weiss, Y., On spectral clustering: Analysis and an algorithm. In Advances in neural information processing systems, pp. 849-856, 2002.
4. Vidal, R., Subspace clustering. IEEE Signal Processing Magazine, 28(2), pp.52-68, 2011.
5. Patel, V.M., Van Nguyen, H. and Vidal, R., Latent space sparse subspace clustering. In Proceedings of the International Conference on Computer Vision (ICCV 2013), pp. 225-232, Sydney, Australia, 1-8 December 2013.
6. Ravishankar, S. and Bresler, Y., Learning sparsifying transforms. IEEE Transactions on Signal Processing, 61(5), pp.1072-1086, 2013.
7. Peng, X., Xiao, S., Feng, J., Yau, W.Y. and Yi, Z., July. Deep Subspace Clustering with Sparsity Prior. In IJCAI pp. 1925-1931, New-York City, New-York, 9-15 July 2016.
8. Chen, Y., Li, G. and Gu, Y., Active Orthogonal Matching Pursuit for Sparse Subspace Clustering. IEEE Signal Processing Letters, 25(2), pp.164-168, 2018?
9. Goh, A. and Vidal, R., Segmenting motions of different types by unsupervised manifold clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07), pp. 1-6, Minneapolis, Minnesota, 18-23 June 2007.
10. Elhamifar, E. and Vidal, R., Sparse subspace clustering: Algorithm, theory, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(11), pp.2765-2781, 2013.
11. Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y. and Ma, Y., Robust recovery of subspace structures by low-rank representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1), pp.171-184, 2013.
12. Ravishankar, S. and Bresler, Y., Closed-form solutions within sparsifying transform learning. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013), pp. 5378-5382, Vancouver, Canada, 26-30 May 2013.
13. Ravishankar, S. and Bresler, Y., Online sparsifying transform learning—Part II: Convergence analysis. IEEE Journal of Selected Topics in Signal Processing, 9(4), pp.637-646, 2015.
14. Guo, J., Guo, Y., Kong, X., Zhang, M. and He, R., Discriminative Analysis Dictionary Learning. In Proceedings on AAAI Conference on Artificial Intelligence, pp. 1617-1623, Phoenix, Arizona, 12-17 February 2016.
15. Maggu, J. and Majumdar, A., Kernel transform learning. Pattern Recognition Letters, 98, pp.117-122, 2017.
16. Van Den Berg, E. and Friedlander, M.P, Probing the Pareto frontier for basis pursuit solutions. SIAM Journal on Scientific Computing, 31(2), pp.890-912, 2008.
17. Tseng, P., Convergence of a block coordinate descent method for nondifferentiable minimization. Journal of Optimization Theory and Applications, 109(3), pp. 475-494, 2001.
18. Attouch, H., Bolte, J., Redont, P., Soubeyran, A., Proximal alternating minimization and projection methods for nonconvex problems. An approach based on the Kurdyka Lojasiewicz inequality. Mathematics of Operations Research, 35(2), pp. 438-47, 2010.
19. Chouzenoux, E. and Pesquet, J.-C. and Repetti, A., A Block Coordinate Variable Metric Forward-Backward Algorithm. Journal of Global Optimization, 66(3), pp. 457-485, 2016.
20. www.cs.columbia.edu/CAVE/software/softlib/coil-20.php
21. https://computervisiononline.com/dataset/1105138686
22. Sinha, D., Kumar, A., Kumar, H., Bandyopadhyay, S. and Sengupta, D., DropClust: Efficient clustering of ultra-large scRNA-seq data. Nucleic Acids Research, 2018.
23. Yeung, K.Y., Haynor, D.R. and Ruzzo, W.L., Validating clustering for gene expression data. Bioinformatics, 17(4), pp.309-318, 2001.

ISBN 978-90-827970-1-5 © EURASIP 2018