SEMI-SUPERVISED SUBSPACE SEGMENTATION

Dong Wang, Qiyue Yin, Ran He, Liang Wang, Tieniu Tan

Center for Research on Intelligent Perception and Computing (CRIPAC)
National Laboratory of Pattern Recognition (NLPR)
Institute of Automation, Chinese Academy of Sciences, Beijing, China
{dwang, qyyin, rhe, wangliang, tnt}@nlpr.ia.ac.cn

ABSTRACT

Subspace segmentation methods usually rely on the raw explicit feature vectors in an unsupervised manner. In many applications, it is cheap to obtain some pairwise link information that tells whether two data points are in the same subspace or not. Though partially available, such link information serves as some kind of high-level semantics, which can be further used as a constraint to improve the segmentation accuracy. By constructing a link matrix and using it as a regularizer, we propose a semi-supervised subspace segmentation model where the partially observed subspace membership prior can be encoded. Specificly, under the common linear representation assumption, we enforce the representational coefficient to be consistent with the link matrix. Thus the low-level and high-level information about the data can be integrated to produce more precise segmentation results. We then develop an effective algorithm to optimize our model in an alternating minimization way. Experimental results for both motion segmentation and face clustering validate that incorporating such link information is helpful to assist and bias the unsupervised subspace segmentation methods.

Index Terms— subspace, clustering, semi-supervised, sparse, link matrix

1. INTRODUCTION

On many occasions, it is desirable to find a low dimensional representation for the original high dimensional data so that the information redundancy and computational complexity can be diminished. Very often, the data we collect come from a union of different subspaces. Then we not only care about the low dimensional representation, but meanwhile expect the data points to be assigned into their corresponding subspaces as well. To this end, many subspace segmentation (a.k.a subspace clustering) methods have been proposed.

In subspace segmentation, many models [1] follow a self-expression assumption, which means a data point can be written as a linear superposition of the other points. Among these models, one notable one is Sparse Subspace Clustering (SSC) [2], which seeks a sparse representation of the target data point using the other points. It also imposes the sparsity constraint on the reconstruction error. Low-Rank Representation (LRR) [3] tends to discover the global structure in the data using the nuclear norm regularization. The Least Squares Regression (LSR) [4] method for subspace clustering has the property to group correlated data together through the Frobenius norm penalty. All these models use spectral clustering methods such as [5] to implement the final cluster segmentation after the representational coefficients are obtained. There are also methods paying attention to other aspects like robustness [6][7] and alignment [8] in subspace clustering. For a more comprehensive review on subspace clustering, we refer the readers to the survey paper by René Vidal [1].

For many subspace segmentation methods in a complete unsupervised setting, the pre/post-processing steps are somehow the tricks for them to achieve promising results. In many cases, it is cheap and convenient to obtain some partial, if not all, label information via crowdsourcing platforms. Thus some semi-supervised clustering or constrained clustering methods [9][10][11][12][13][14][15] have been proposed to utilize such label information. By imposing additional constraints like "must-link" and "cannot link" [16], traditional clustering methods are enabled to perform better. Inspired by this line of research, we propose a Semi-Supervised Subspace Segmentation (S^4) model with instance pairwise link information embedded. So the dependence on the pre/post-processing steps can be to some extent alleviated. It should be noted that our focus in this paper is to test how helpful such partial link information is for those unsupervised subspace segmentation methods which adopt a self-expression assumption, so we will not compare with the above semi-supervised clustering methods which do not share the same assumption.

In the semi-supervised setting, we obtain a partially observed pairwise link matrix which indicates the incomplete membership information. Then in our model we enforce the corresponding entries of the linear representation matrix to be consistent with the observed entries of the link matrix. The intuition is, at least those data points that are already known to be in the same subspace as the target data point should be used as part of the representation components. Although we

only know some partial link information, they are high-level semantics that are more precise and reliable than the raw explicit feature vectors. Incorporating such high-level semantic information can improve the final subspace clustering results.

Finally, our contributions in this paper are summarized as follows:

• The proposed method is one of the early solutions to extend the subspace segmentation problem to a semi-supervised setting under the self-expression assumption. Concretely, by enforcing proper constraints, we propose a flexible model that takes advantage of the partially available link information.

• We develop an effective optimizing algorithm to solve the proposed model. And experiments on the benchmark datasets show that incorporating such link information is useful to aid and calibrate the unsupervised subspace segmentation methods.

2. PROPOSED MODEL

2.1. Notations

In this paper, the transpose of a matrix X is X^T. X_i means the i-th column of X. X_{ij} is the (i,j) entry of X. For two matrices X and Y of corresponding dimensions, $\langle X, Y\rangle$ is the trace of X^TY. $||\cdot||_F$ and $||\cdot||_1$ are used to represent the Frobenius norm and l_1 norm respectively. And we use "$\odot$" to represent the element-wise product of two matrices with the same size. $\forall a \in \mathbb{R}$, $(a)_+$ returns itself if it is non-negative and returns zero otherwise. diag(X) is a diagonal matrix whose non-zero entries are the diagonal of X.

2.2. Formulation

Inspired by SSC [2], we also assume that any data point can be expressed as a sparse linear representation of all the points excluding itself. Suppose d is the dimension of the original data space and N is the number of data points. Given the data matrix $X \in \mathbb{R}^{d\times N}$, the representation coefficient $Z \in \mathbb{R}^{N\times N}$ and the reconstruction error $E \in \mathbb{R}^{d\times N}$, we are going to optimize the following objective:

$$\begin{aligned} &\min_{Z,E} ||Z||_1 + \lambda||E||_1 + \alpha \sum_{i,j\in O} (Z_{ij} - A_{ij})^2 \\ &s.t \;\; X = XZ + E, \; \text{diag}(Z) = 0 \end{aligned} \tag{1}$$

λ and α are the tradeoff parameters. E is also required to be sparse so that the model can be to some extent insensitive to outliers. After solving for Z, we can manage to obtain the final clustering results.

The purpose of the third term in (1) is to enforce the representation coefficient Z to be consistent with the partially available label information.

Here A is a partially observed "link matrix" that has binary entries indicating which candidate data points can be used to represent a target point. We say points i and j have a link if they share the same label or come from the same subspace. Otherwise there is non-link between them. And O is the set of observed links. Concretely, for A_{ij} being 1, it tells that point i can be used to represent point j or vice versa. For A_{ij} being 0, points i and j cannot be used to express each other. For A_{ij} being "?", it means that whether points i and j can be used to express each other is unknown beforehand. See Fig.1 for a clear illustration.

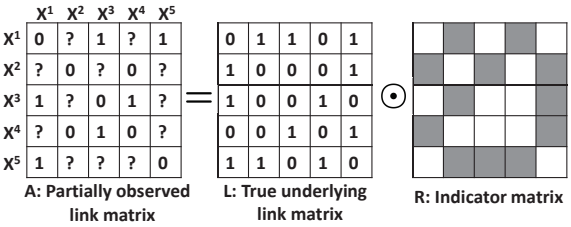

Fig. 1. Illustration of the link matrix.

It should be noted that A is binary, but the elements of Z may not be all binary. However, this can be solved by normalizing Z to lie in [0,1]. The more Z_{ij} approaches 1, the more possible it becomes for i and j to represent each other. On the contrary, if Z_{ij} approximates to 0, it becomes less possible for i and j to represent each other. Forcing Z to conform with A will lead to a more accurate Z and thus a more precise cluster segmentation.

Actually A can be seen as being generated by entry-wise multiplying the true underlying link matrix L with a binary indicator (or mask) matrix R. In Fig.1, the shaded entries of R indicate their corresponding entries in L will become unavailable. And the white ones mean their corresponding entries in L will be shown to us.

$$R_{ij} = \begin{cases} 1 & \text{if } A_{ij} \text{ is observed} \\ 0 & \text{if } A_{ij} \text{ is unobserved} \end{cases} \tag{2}$$

Therefore, R reflects how much label information we have and decides which corresponding entries of A and Z should be consistent with. Then the original formulation (1) can be rewritten as follows:

$$\begin{aligned} &\min_{Z,E} ||Z||_1 + \lambda||E||_1 + \alpha||Z\odot R - L\odot R||_F^2 \\ &s.t \;\; X = XZ + E, \; \text{diag}(Z) = 0 \end{aligned} \tag{3}$$

3. SOLUTION TO THE PROPOSED MODEL

In reality, we do not know the true underlying link matrix L, otherwise clustering results can be directly obtained. And L_i cannot be substituted into the computation if it has "?" being its entries. However we can generate a pseudo or artificial fully observed link matrix $\hat{L}$ by filling in the unobserved entries

of A with arbitrary values. Such operation will not influence the final solution since these added values will be ruled out by R during optimization. But the advantage here is that Z becomes easier to be computed once per column rather than once per element. Without loss of generality, we can obtain $\hat{L}$ by setting those unobserved entries in A to be all zero.

Inspired by [17], we introduce an auxiliary variable C so that the objective function becomes separable and easy to optimize. Replacing L with $\hat{L}$, (3) can be further refomulated as follows:

$$\begin{aligned} &\min_{C,Z,E} ||C||_1 + \lambda||E||_1 + \alpha||Z \odot R - \hat{L} \odot R||_F^2 \\ &s.t \;\; X = XZ + E, \;\; Z = C - \text{diag}(C) \end{aligned} \tag{4}$$

The above objective will share the same solution with (3) if the convergence condition is to let C and Z become as close as possible during optimization.

Suppose matrix P and Q are two Lagrange multipliers, then we have the following augmented Lagrangian function :

$$\begin{aligned} \mathcal{L} = &||C||_1 + \lambda||E||_1 + \alpha||Z \odot R - \hat{L} \odot R||_F^2 \\ &+ < P, X - XZ - E > + < Q, Z - (C - \text{diag}(C)) > \\ &+ \frac{\mu}{2}(||X - XZ - E||_F^2 + ||Z - (C - \text{diag}(C))||_F^2) \end{aligned} \tag{5}$$

According to the Alternating Direction Method of Multipliers (ADMM) [18], we can alternatively solve one of the variables C, Z, E and meanwhile keep the other two fixed.

• Solve for C with Z and E fixed.

$$C^{k+1} = \hat{C} - \text{diag}(\hat{C}) \tag{6}$$

$$\hat{C} = \mathcal{T}_{\frac{1}{\mu}}(Z^k + Q^k/\mu) \tag{7}$$

where $\hat{C}$ is the solution without considering $\text{diag}(C)$. Here $\mathcal{T}_\eta(\cdot)$ is the soft-thresholding operator for every matrix entry and is defined as: $\mathcal{T}_\eta(v) = (|v| - \eta)_+ \text{sgn}(v)$.

• Solve for E with C and Z fixed:

$$E^{k+1} = \mathcal{T}_{\frac{\lambda}{\mu}}(X - XZ^k + P^k/\mu) \tag{8}$$

• Solve for Z with C and E fixed: To drop "$\odot$", we solve for Z once per column. By optimizing (5) *w.r.t* Z, we have:

$$\begin{aligned} \min_{Z_i} \;\; &\alpha||S_{(i)}Z_i - S_{(i)}\hat{L}_i||_2^2 \\ &+ P_i^T(X_i - XZ_i - E_i) + Q_i^T[Z_i - (C - \text{diag}(C))_i] \\ &+ \frac{\mu}{2}(||X_i - XZ_i - E_i||_2^2 + ||Z_i - (C - \text{diag}(C))_i||_2^2) \end{aligned} \tag{9}$$

where $S(i)$ is a diagonal matrix with the i-th column of R as its diagonal entries. This leads to a closed form solution:

$$\begin{aligned} Z_i^{k+1} = &(2\alpha S_{(i)}^T S_{(i)} + \mu X^T X + \mu I) \backslash [2\alpha S_{(i)}^T S_{(i)} \hat{L}_i + X^T P_i^k \\ &- Q_i^k + \mu(C^{k+1} - \text{diag}(C^{k+1}))_i + \mu X^T X_i - \mu X^T E_i^{k+1})] \end{aligned} \tag{10}$$

• Update the Lagrange multipliers with C, Z and E fixed.

$$\begin{aligned} P^{k+1} &= P^k + \mu(X - XZ^{k+1} - E^{k+1}) \\ Q^{k+1} &= Q^k + \mu(Z^{k+1} - C^{k+1}) \end{aligned} \tag{11}$$

The above steps are repeated until $||C^k - Z^k||_\infty$, $||Z^k - Z^{k-1}||_\infty$, $||E^k - E^{k-1}||_\infty$ and $||X - XZ^k - E^k||_\infty$ are all below the converging tolerance $\varepsilon = 10^{-4}$.

After we get the representation matrix Z and construct the affinity matrix $W = |Z| + |Z|^T$, the spectral clustering method of [5] is used to obtain the final clustering results. The whole procedure of our method can be summarized in Algorithm 1.

Algorithm 1 Semi-Supervised Subspace Segmentation (S^4)

Input: X, A, k=0, Set C^0=Z^0=E^0=P^0=Q^0=0, $\mu = 1$
Output: $W = |Z| + |Z|^T$
  while "not converged" **do**
    Fix Z and E, update C using (7) and (6)
    Fix Z and C, update E using (8)
    Fix C and E, update Z using (10)
    Fix C, Z and E, update P and Q using (11)
    $\mu = \min(10^{30}, 1.1\mu)$
    $k = k + 1$
  end while

4. EXPERIMENTS

In this section, we test the usefulness of our method for motion segmentation and face clustering. In particular, we are going to compare with Spectral Curvature Clustering (SCC)[19], LRR [20], LSR [4], Low-Rank Subspace Clustering (LRSC) [21] and SSC [17]. As explained in introduction, we are not going to compare with the aforementioned semi-supervised clustering methods since we only concentrate on verifying how helpful the link information is for unsupervised subspace segmentation methods which follow a self-expression assumption rather than compare all the semi-supervised clustering methods.

4.1. Settings

For the methods whose source codes are available, we reproduce their results using the same settings therein. Both two versions of LSR, LSR1 and LSR2, are tested in the experiments. For LRR, we report the results with and without (LRR-H) post-processing on the coefficient matrix. We test all the models directly on the original data without preprocessing steps like PCA.

4.2. Motion Segmentation

For motion segmentation, the Hopkins 155 dataset [22] is chosen for evaluation. It includes 120 two-motion and 35 three-motion video sequences. One motion corresponds to

Table 1. Clustering errors (%) on Hopkins 155 dataset.

Model	2 motions		3 motions		All motions	
	mean	median	mean	median	mean	median
SCC	2.24	**0.00**	6.69	0.40	3.25	**0.00**
LRR	3.30	0.34	7.39	2.80	4.22	0.53
LRR-H	1.33	**0.00**	2.51	**0.00**	1.60	**0.00**
LSR1	1.80	0.11	4.14	1.60	2.33	0.31
LSR2	2.08	0.10	4.85	1.83	2.72	0.31
LRSC	2.46	**0.00**	6.03	2.20	3.27	**0.00**
SSC	1.52	**0.00**	4.40	0.56	2.18	**0.00**
S^4	**0.61**	**0.00**	**2.09**	0.64	**0.94**	**0.00**

a single subspace. We report results in terms of both mean and median error rates over a corresponding number of video sequences. See Section 7.1 in [17] for more details.

In this experiment, we unveil the subspace membership for a random 30% subset of the data points and therefore construct R as described in Sections 2 and 3. This also amounts to having a link matrix A with 9 percent of its entries being observed. The subspace segmentation results of different models are shown in Table 1. As can be seen, our proposed method S^4 performs the best for the 2-motion case. For 3-motion, S^4 is also superior to other models in terms of mean and is comparable in terms of median. With an average clustering error of 0.94%, the overall performance of S^4 on all motions beats all the other compared methods. Such empirical results prove that encoding the partially available link information is helpful to the subspace segmentation task.

4.3. Face Clustering

For face clustering, we choose the Extended Yale B [23] dataset. This dataset consists of frontal facial images of 38 subjects with varying lighting conditions. And the first 10 subjects are used to evaluate the compared models. In this experiment, we resize each face image to 48×42 pixels and stack its column into a 2016D vector.

Similarly, we disclose the subspace membership for a random 20% selection of the data points and construct R as mentioned before. To observe the effect of the number of subjects in face clustering, we assess all the models on the first 5, 8 and 10 subjects respectively. As shown in Table 2, when the number of subjects increases, the clustering errors of different models all increase to various degrees. In the case of 8 and 10 subjects, S^4 consistently outperforms the other methods. For 5 subjects, S^4 and SSC both produce a perfect clustering.

To check the performance change as the amount of available label information increases, we test our model on the first 10 subjects when 0%, 10%, 20% and 30% membership information are given respectively. As shown in Table 3, when more such high level semantic information is fed to our model, it performs better and better.

Table 2. Clustering errors (%) on Extended Yale B dataset.

Methods	5 subjects	8 subjects	10 subjects
SCC	60.56	65.63	73.59
LRR	14.69	21.09	33.75
LRR-H	1.88	2.34	8.91
LSR1	13.75	36.91	37.81
LSR2	5.31	17.38	34.38
LRSC	3.75	8.79	10.47
SSC	**0.00**	4.88	9.38
S^4	**0.00**	**0.98**	**1.41**

Table 3. Error rate change of S^4 under different amount of link information

Link information (%)	0	10	20	30
Error rate (%)	9.38	2.03	1.41	0.63

4.4. Discussion

About the fairness of model comparison: We are aware that all the baseline models are unsupervised without using any supervised information. However, even though we use such partial link information to somehow correct the representational matrix in the optimizing process, we still simply rely on the solved Z to realize an unsupervised clustering in the end without using the label information to classify the data directly. Therefore the fairness is guaranteed. Using a small amount of link information to complement the raw feature based unsupervised subspace segmentation is where our model differs and stands out.

5. CONCLUSION

Under the linear representation assumption, we have interpreted the mutual representational relation between any two data points in the subspace as the existence of a pairwise link. In terms of selecting proper representation components for a given target point, we have enforced the representational coefficient to be consistent with the partially observed link matrix. Therefore the resulting model is able to prefer points that are known to be in the same subspace while denying points known to come from other subspaces. Experimental results have proved that incorporating link information can aid the unsupervised subspace segmentation results without using any link information at all. One possible future work is to extend other methods in the subspace segmentation family such as LRR to the semi-supervised setting.

6. ACKNOWLEDGEMENT

This work is jointly supported by National Basic Research Program of China (No.2012CB316300) and National Science Foundation of China (No.61175003).We would like to thank Dr. Shu Wu for his very helpful discussions during this work.

7. REFERENCES

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

[2] Ehsan Elhamifar and René Vidal, "Sparse subspace clustering," in *CVPR*, 2009.

[3] Guangcan Liu, Zhouchen Lin, and Yong Yu, "Robust subspace segmentation by low-rank representation," in *ICML*, 2010, pp. 663–670.

[4] Canyi Lu, Hai Min, Zhongqiu Zhao, Lin Zhu, Deshuang Huang, and Shuicheng Yan, "Robust and efficient subspace segmentation via least squares regression," in *ECCV*, 2012, pp. 347–360.

[5] Andrew Y Ng, Michael I Jordan, and Yair Weiss, "On spectral clustering: analysis and an algorithm," *Advances in Neural Information Processing Systems*, vol. 2, pp. 849–856, 2002.

[6] Yingya Zhang, Zhenan Sun, Ran He, and Tieniu Tan, "Robust low-rank representation via correntropy," in *ACPR*, 2013, pp. 461–465.

[7] Yingya Zhang, Zhenan Sun, Ran He, and Tieniu Tan, "Robust subspace clustering via half-quadratic minimization," in *ICCV*, 2013, pp. 3096 – 3103.

[8] Qi Li, Zhenan Sun, Ran He, and Tieniu Tan, "Joint alignment and clustering via low-rank representation," in *ACPR*, 2013, pp. 591–595.

[9] Sugato Basu, Ian Davidson, and Kiri Wagstaff, *Constrained clustering: Advances in algorithms, theory, and applications*, CRC Press, 2008.

[10] Sugato Basu, Arindam Banerjee, and Raymond J Mooney, "Semi-supervised clustering by seeding," in *ICML*, 2002, vol. 2, pp. 27–34.

[11] Mikhail Bilenko, Sugato Basu, and Raymond J Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *ICML*, 2004, p. 11.

[12] Brian Kulis, Sugato Basu, Inderjit Dhillon, and Raymond Mooney, "Semi-supervised graph clustering: a kernel approach," *Machine Learning*, vol. 74, no. 1, pp. 1–22, 2009.

[13] Eric Bair, "Semi-supervised clustering methods," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 5, no. 5, pp. 349–361, 2013.

[14] Xiaotong Yuan, Baogang Hu, and Ran He, "Agglomerative mean-shift clustering via query set compression.," in *SDM*, 2009, pp. 221–232.

[15] Xiaotong Yuan, Baogang Hu, and Ran He, "Agglomerative mean-shift clustering," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 2, pp. 209–219, 2012.

[16] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl, "Constrained k-means clustering with background knowledge," in *ICML*, 2001, vol. 1, pp. 577–584.

[17] Ehsan Elhamifar and René Vidal, "Sparse subspace clustering: algorithm, theory, and applications," *IEEE Trans. PAMI*, vol. 34, no. 11, pp. 2765–2781, 2013.

[18] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[19] Guangliang Chen and Gilad Lerman, "Spectral curvature clustering (scc)," *IJCV*, vol. 81, no. 3, pp. 317–330, 2009.

[20] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. PAMI*, vol. 35, no. 1, pp. 171–184, 2013.

[21] Paolo Favaro, René Vidal, and Avinash Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *CVPR*, 2011, pp. 1801–1807.

[22] Roberto Tron and René Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *CVPR*, 2007, pp. 1–8.

[23] Kuang Chih Lee, Jeffrey Ho, and David J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. PAMI*, vol. 27, no. 5, pp. 684–698, 2005.