Subspace Structure-aware Spectral Clustering for Robust Subspace Clustering

Masataka Yamaguchi, Go Irie, Takahito Kawanishi, Kunio Kashino
NTT Communication Science Laboratories, NTT Corporation, Japan
{yamaguchi.masataka.up, go.irie.nv, takahito.kawanishi.fx, kunio.kashino.me}@hco.ntt.co.jp

Abstract

Subspace clustering is the problem of partitioning data drawn from a union of multiple subspaces. The most popular subspace clustering framework in recent years is the graph clustering-based approach, which performs subspace clustering in two steps: graph construction and graph clustering. Although both steps are equally important for accurate clustering, the vast majority of work has focused on improving the graph construction step rather than the graph clustering step. In this paper, we propose a novel graph clustering framework for robust subspace clustering. By incorporating a geometry-aware term with the spectral clustering objective, we encourage our framework to be robust to noise and outliers in given affinity matrices. We also develop an efficient expectation-maximization-based algorithm for optimization. Through extensive experiments on four real-world datasets, we demonstrate that the proposed method outperforms existing methods.

1. Introduction

In many practical scenarios, high dimensional data often live in a union of low-dimensional linear subspaces. The problem of partitioning such data so that each cluster consists of all the data belonging to one subspace is called *Subspace Clustering*. Subspace clustering has greatly attracted attention as it has important and wide-ranging applications in various fields such as computer vision [44, 28], data mining [2, 35], network analysis [12, 8], system identification [46, 3], and biology [21, 29].

Over the past few decades, many subspace clustering methods have been proposed, including algebraic methods [5, 9, 14, 20, 45, 30, 19], iterative methods [6, 41, 1, 52], statistical methods [38, 37, 50, 36], and graph clustering-based methods [49, 7, 10, 11, 25, 47, 27, 26, 43, 15, 22]. Ever since the celebrated self-representation based approach [10, 11] was proposed, many recent efforts have focused on graph clustering-based methods, because they often outperform the other methods in practical settings. Most graph clustering-based methods are performed in two steps. The first step, graph construction, is to compute an affinity matrix wherein a pair of data points belonging to the same subspace has higher affinity than those in different subspaces. The second step, graph clustering, is to cluster data by applying a graph clustering method (*e.g.*, spectral clustering) to that affinity matrix.

Although both graph construction and graph clustering are important for achieving accurate clustering, most previous work has focused on improving graph construction. In fact, data can be correctly clustered by standard graph clustering methods if one can obtain an affinity matrix M that satisfies certain conditions (*e.g.*, $M_{ij} > 0$ if the ith and jth data points belong to the same subspace and otherwise $M_{ij} = 0$), hence pursuing a better method for computing an affinity matrix is important for correctly clustering data. However, in practical settings, data contain noise and outliers, so no method will not always provide an affinity matrix that satisfies such conditions. To accurately cluster data, it is equally important to improve the graph clustering step.

In the past decade, spectral clustering [32] has been the de facto standard method for the graph clustering step. Although its effectiveness in the graph clustering-based subspace clustering methods has been empirically validated, its performance deteriorates quickly as the intensity of noise in the affinity matrix increases, since spectral clustering clusters data based on only a given affinity matrix. One way to mitigate this problem is to use not only an affinity matrix but also the data's geometric structure in the ambient space for graph clustering. However, to the best of our knowledge, no graph clustering method has considered the data's geometric structure in the ambient space for subspace clustering.

To accurately conduct subspace clustering, we propose a geometry-aware graph clustering method for graph clustering-based subspace clustering. More specifically, we propose a novel graph clustering objective that consists of the spectral clustering objective and a new geometry-aware term that encourages data in each cluster to lie in the same low-dimensional subspace. We note that maximization of the proposed objective can be interpreted as maximum a posteriori (MAP) estimation problem of a variant of the

Gaussian mixture model (GMM), and hence we employ the expectation-maximization (EM) algorithm for solving the objective. Through extensive experiments on four real-world datasets, we demonstrate that the proposed method outperforms existing methods. Moreover, we also demonstrate that our method is also effective for frameworks that unify graph construction and clustering steps [24].

The contributions of this paper are as follows:

- We propose a novel graph clustering approach for graph clustering-based subspace clustering, which clusters data based not only on a given affinity matrix but also on the data's geometric structure in the ambient space.
- We provide results of experiments on four real-world datasets, which show the effectiveness of our approach.

2. Preliminary

2.1. Problem Formulation

Subspace clustering is the problem of clustering N D-dimensional data points $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^T \in \mathbb{R}^{N \times D}$ that live in (or near) a union of K low-dimensional linear subspaces $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2 \cup ... \cup \mathcal{U}_K$ so that each cluster consists of all the data belonging to one subspace. Specifically, the problem is to find an assignment matrix $G = [\boldsymbol{g}_1, ..., \boldsymbol{g}_K] \in \{0,1\}^{N \times K}$ that satisfies $G\mathbf{1}_K = \mathbf{1}_N$ and $\boldsymbol{x}_i \in \mathcal{U}_j$ if $G_{ij} = 1$.

2.2. K-subspaces

K-subspaces [41, 1] minimizes the following objective by updating an assignment matrix G and a set of orthonormal basis $\{U_i\}_{i=1}^K$ alternately:

$$\min_{\{\boldsymbol{y}_i\}_i^N, \{U_i\}_i^K} \sum_k^K \sum_i^N G_{ij} ||\boldsymbol{x}_i - U_k^T U_k \boldsymbol{y}_i||_2^2 \qquad (1)$$

Furthermore, by replacing the squared distance terms in Eq. (1) with the negative log-likelihood terms of the probabilistic principle component analysis (PPCA), K-subspaces can be naturally extended to be its probabilistic version, called mixtures of PPCA (MPPCA) [39]. Although both K-subspaces and MPPCA are practical, they have some disadvantages such as that they tend to converge to a local minimum [42].

2.3. Graph Clustering-based Approach

The graph clustering-based subspace clustering approach first computes an affinity matrix wherein a pair of data points belonging in the same subspace has higher affinity than those in different subspaces, and then applies a graph clustering method to that affinity matrix.

Graph construction: Various approaches [49, 7, 10, 11, 25, 47, 27, 26, 43, 15, 22, 17] have been proposed for computing an affinity matrix. For example, the self-representation-based approach, the most representative one among them, first builds a self-representation matrix Z^* that is computed by representing each data point by a linear combination of the others and then computes an affinity matrix M using Z^* (*e.g.*, $M_{ij} = |Z^*_{ij}| + |Z^{*\mathrm{T}}_{ij}|$). To compute the self-representation matrix Z^*, most methods first solve the following problem:

$$\min_{Z \in \mathcal{C}} \mathrm{h}(E) + \eta \mathrm{r}(Z), \ s.t. \ X = XZ + E, \qquad (2)$$

where $\mathrm{h}(E)$ is the loss function for reconstruction error E (*e.g.*, $\mathrm{h}(E) = ||E||_F^2$), $\mathrm{r}(Z)$ is a regularizer for a self-representation matrix Z (*e.g.*, $\mathrm{r}(Z) = ||E||_1$), and $\mathcal{C}$ is a constraint set for Z (*e.g.*, $\mathcal{C} = \{Z | Z \in \mathbb{R}^{N \times N}, Z_{ii} = 0\}$). It has been proven that affinity matrices computed by some methods, *e.g.*, sparse subspace clustering (SSC) [10, 11], satisfy the self-expressiveness property (i.e., $M_{ij} > 0$ if ith and jth data points belong to the same subspace and otherwise $M_{ij} = 0$) under certain conditions. In such a case, we can get correct clustering results by simply applying spectral clustering, or much simpler methods such as depth-first search. However, in practical settings, those conditions do not usually hold due to noise and outliers, and the clustering performance highly depends on the robustness of the subsequent graph clustering method.

Graph clustering: After the graph construction step, data are clustered by applying a graph clustering method to a computed affinity matrix. Typical graph clustering methods cluster data by finding an assignment matrix $\mathcal{G}$ that maximizes some objectives, $\epsilon(G, M)$, as follows:

$$\max_G \epsilon(G, M) \text{ subject to } G \in \mathcal{G}, \qquad (3)$$

where $\mathcal{G} = \{G | G \in \{0,1\}^{N \times K}, G\mathbf{1}_K = \mathbf{1}_N\}$.

In subspace clustering, spectral clustering is used for graph clustering. Spectral clustering uses the following objective:

$$\epsilon(G, M) = \sum_{l=1}^K \frac{\boldsymbol{g}_l^T M \boldsymbol{g}_l}{\boldsymbol{g}_l^T D \boldsymbol{g}_l}, \qquad (4)$$

where D is a degree matrix, which satisfies $(D)_{ij} = 0$ if $i \neq j$; otherwise; $(D)_{ii} = \sum_j M_{ij}$. Since computing the optimal assignment matrix G is NP-hard, it is approximately solved via continuous relaxation. For more details, see [51].

One drawback of using typical graph clustering methods for subspace clustering is that their performance deteriorates quickly as the intensity of the noise in an affinity matrix increases, since they cluster data based on only a

given affinity matrix. One way to mitigate this problem is to use not only a given affinity matrix, but the data's geometric structure for graph clustering. Although Nie *et al*. [34] proposed a graph clustering method that considers data as well as an affinity matrix, their method is not specialized for subspace clustering. Some prior work [24, 48] proposed subspace clustering frameworks that unify both graph construction and graph clustering into a single optimization problem. However, their methods internally use spectral clustering as is; hence, their methods are still fragile to the noise in a given affinity matrix.

3. Subspace Structure-aware Spectral Clustering

To improve the robustness of spectral clustering in the subspace clustering problem, we consider using the data's geometric structure in the ambient space for graph clustering. More specifically, we propose to use the following objective, which consists of the spectral clustering objective $\epsilon(G, M)$ and a new geometry-aware term $\mathrm{r}(G, X)$ that encourages data in each cluster to lie in the same low-dimensional subspace:

$$\max_{G} (1-\eta)\epsilon(G, M) + \eta \mathrm{r}(G, X) \text{ subject to } G \in \mathcal{G}, \quad (5)$$

where η is a hyper-parameter.

The most important key is how to define the geometry-aware term $\mathrm{r}(G, X)$ in Eq. (5). A naïve idea is to use summation of rank of each cluster's data matrix as follows:

$$\begin{aligned} \mathrm{r}(G, X) &= -\sum_{k}^{K} \mathrm{rank}(\mathrm{Diag}(\boldsymbol{g}_k)X) \\ &= -\sum_{k}^{K}\sum_{d}^{D} |\lambda_{kd}|_0 \end{aligned} \quad (6)$$

where λ_{kd} is the kth singular value of $\mathrm{Diag}(\boldsymbol{g}_k)X$. The advantage of using Eq. (6) for the geometry-aware term is that it can directly encourage to decrease the number of intrinsic dimensions of the subspace spanned by each cluster's data. However, due to its discontinuity, solving Eq. (5) is intractable when using Eq. (6) for $\mathrm{r}(G, X)$. Moreover, Eq. (6) is extremely sensitive to noise in each cluster's data.

Another idea is to use the objective of K-subspaces for $\mathrm{r}(G, X)$ as follows:

$$\mathrm{r}(G, X) = -\min_{\{\boldsymbol{y}_i\}_i^N, \{U_i\}_i^K} \sum_{i}^{N}\sum_{k}^{K} G_{ij} ||\boldsymbol{x}_i - U_k^T U_k \boldsymbol{y}_i||_2^2, \quad (7)$$

where $\boldsymbol{y}_i \in \mathcal{R}^{d_i}$ and $U_i \in \mathcal{R}^{d_i \times D}$. However, when using Eq. (7), one has to set the number of dimensions d, which is often unknown in advance.

Alternatively, inspired by the probabilistic version of K-subspaces, i.e., MPPCA [39], we consider replacing the squared norm of Eq. (7) with the likelihood of the zero-mean Gaussian distribution as follows:

$$\mathrm{r}(G, X) = \max_{\Sigma_1, \ldots, \Sigma_K \succeq 0} \sum_{i}^{N} \log \sum_{k}^{K} G_{ik} \mathcal{N}(\boldsymbol{x}_i; 0, \Sigma_k + \sigma I), \quad (8)$$

where σI is a term to avoid degeneration of the covariance matrix (we set $\sigma = 1e-6$ in this paper). When using Eq. (8) for $\mathrm{r}(G, X)$, one no longer has to set the number of dimensions d, unlike Eq. (7). In addition, more interestingly, it can be shown that Eq. (8) works as a smooth surrogate of the rank function.

3.1. Log-likelihood for Gaussian as Surrogate for Rank

In the following, we show that Eq. (8) works as a smooth approximation of the rank function. First, Eq. (8) can be represented as follows:

$$\mathrm{r}(G, X) = \max_{\Sigma_1, \ldots, \Sigma_K \in \mathcal{S}} \sum_{i}^{N} \log \sum_{k}^{K} G_{ik} \mathcal{N}(\boldsymbol{x}_i; 0, \Sigma_k), \quad (9)$$

where $\mathcal{S} = \{\Sigma | \Sigma \in \mathcal{R}^{D \times D} \text{ and } \Sigma \succeq \sigma I\}$. Let $\hat{\Sigma}_1, ..., \hat{\Sigma}_K$ be the maximizers of Eq. (9), which can be analytically solved as follows:

$$\hat{\Sigma}_k = U_k \mathrm{Diag}(\max(\boldsymbol{d}_k, \sigma \mathbf{1}_D)) U_k^T, \quad (10)$$

where $\frac{1}{\sum_i G_{ik}} \sum_i G_{ik} \boldsymbol{x}_i \boldsymbol{x}_i^T = U_k \mathrm{Diag}(\boldsymbol{d}_k) U_k^T$ (see supplementary material for derivation of this). Moreover, by substituting Eq. (10) into Eq. (9), $\mathrm{r}(G, X)$ can be represented as follows:

$$\begin{aligned} &\mathrm{r}(G, X) \\ &= -\sum_{k}^{K}\sum_{d}^{D} \frac{\rho_k^2}{\sigma} \log \max(1, \frac{\lambda_{kd}}{\rho_k}) + \frac{\rho_k^2}{2\sigma} \min(1, \frac{\lambda_{kd}^2}{\rho_k^2}) + const. \\ &= -\sum_{k}^{K}\sum_{d}^{D} \mathrm{f}_{\rho_k, \sigma}(\lambda_{kd}) + const. \end{aligned} \quad (11)$$

where $\rho_k = \sqrt{m_k \sigma}$, m_k is the number of data belonging to kth cluster and $\mathrm{f}_{\rho,\sigma}(\lambda) = \frac{\rho^2}{\sigma} \log \max(1, \frac{\lambda}{\rho}) + \frac{\rho^2}{2\sigma} \min(1, \frac{\lambda^2}{\rho^2})$ (see supplementary material for derivation of this). From Eq. (11), we can see that $\mathrm{r}(G, X)$ can be represented as summation of the function of each singular value.

[2]Interestingly, $\mathrm{f}_{\rho,\sigma}(\lambda)$ is also similar to the objective of agglomerative lossy compression (ALC) [28, 36]. It is worth noting that the objective of ALC is derived from the viewpoint of data compression, whereas Eq. (11)

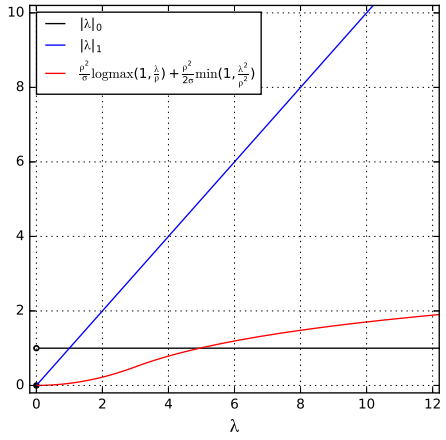

Figure 1. Comparison of $|\lambda|_0$, $|\lambda|_1$, and $\mathrm{f}_{\rho,\sigma}(\lambda) = \frac{\rho^2}{\sigma}\log\max(1,\frac{\lambda}{\rho}) + \frac{\rho^2}{2\sigma}\min(1,\frac{\lambda^2}{\rho^2})$ (in this figure we set $\rho = \sqrt{\sigma} = 3$). From this graph, we can observe that Eq. (11) is similar to $|\lambda|_0$ yet also smooth, and small input values are suppressed[2].

In Fig. 3, we show comparison of (1) the L_0 norm $|\lambda|_0$, which corresponds to the rank function $\mathrm{rank}(\lambda)$, (2) the nuclear norm $|\lambda|_1$, which can be considered convex relaxation of the rank function, and (3) the derived function $\mathrm{f}_{\rho,\sigma}(\lambda) = \frac{\rho^2}{\sigma}\log\max(1,\frac{\lambda}{\rho}) + \frac{\rho^2}{2\sigma}\min(1,\frac{\lambda^2}{\rho^2})$. From Fig. 3, we can observe that the rank function is extremely sensitive to noise at $\lambda = 0$, whereas nuclear norm overestimates a large input value, compared to the rank function. On the other hand, we can observe that the function $\mathrm{f}_{\rho,\sigma}(\lambda)$ is similar to the rank function, yet it can suppress small input values and also avoid overestimation of a large input value. Moreover, since the function $\mathrm{f}_{\rho,\sigma}(\lambda)$ is smooth, it is much easier to optimize than the rank function. For the above reasons, we employ Eq. (9) for the geometry-aware term $\mathrm{r}(G,X)$.

4. Optimization

As with the original spectral clustering problem, it is hard to directly solve Eq. (5). Therefore, following the standard spectral clustering algorithms, we relax the assignment matrix Z from the discrete domain to the continuous domain. More specifically, we relax Eq. (5) as follows:

$$\max_{G}\ (1-\eta)\epsilon(G,M)+\eta\mathrm{r}(G,X)\ \text{subject to}\ G\in\mathcal{H}, \tag{12}$$

is derived from K-subspaces and its probabilistic interpretation, which is more intuitive for subspace clustering than data compression. Moreover, their algorithm is based on the greedy algorithm, whereas our algorithm is based on continuous optimization problem, which tends to result in better solutions than the greedy algorithm.

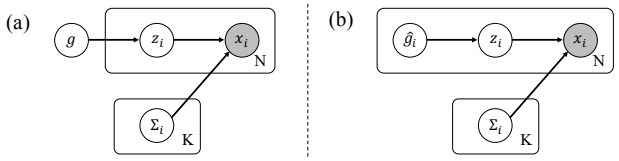

Figure 2. Comparison of two mixture models. (a) The directed graphical model corresponding to the zero-mean Gaussian mixture model. (b) The directed graphical model corresponding to the probabilistic model $p(\boldsymbol{x}_i) = \sum_k^K G_{ik}\mathcal{N}(\boldsymbol{x}_i;0,\Sigma_k)$.

where $\mathcal{H} = \{G|G\in\mathcal{R}^{N\times K}, 0\le G_{ij}\le 1, G\mathbf{1}_K = \mathbf{1}_N\}$[3]. Since standard algorithms for spectral clustering can no longer be used to solve Eq. (12) due to the geometry-aware term $\mathrm{r}(G,X)$, we propose a new algorithm for this problem.

First, we rewrite Eq. (12) as follows:

$$\begin{aligned}
&\max_{G\in\mathcal{H}}\ (1-\eta)\epsilon(G,M)+\eta\mathrm{r}(G,X)\\
=&\max_{G\in\mathcal{H}}\ (1-\eta)\epsilon(G,M)+\\
&\qquad\max_{\Sigma_1,\ldots,\Sigma_K\in\mathcal{S}}\ \eta\sum_i^N\log\sum_k^K G_{ik}\mathcal{N}(\boldsymbol{x}_i;0,\Sigma_k)\\
=&\max_{\substack{G\in\mathcal{H}\\ \Sigma_1,\ldots,\Sigma_K\in\mathcal{S}}}\ (1-\eta)\epsilon(G,M)+\eta\sum_i^N\log\sum_k^K G_{ik}\mathcal{N}(\boldsymbol{x}_i;0,\Sigma_k),
\end{aligned} \tag{13}$$

Note that the second term in Eq. (13) can be interpreted as a log-likelihood of a variant of the GMM, in which the mixing coefficient vector is independently defined for each sample (i.e., the ith sample's coefficient vector $\hat{\boldsymbol{g}}_i$ is $\hat{\boldsymbol{g}}_i = [G_{i1},...,G_{iK}]^T$). To clarify the difference between the GMM and that mixture model, we show comparison of those directed graphical models in Fig. 2. In addition, the first term can be interpreted as a regularizer for the mixing coefficient matrix, i.e., G. From this observation, solving Eq. (13) can be interpreted as solving the maximum a posteriori (MAP) estimation problem of G (and $\Sigma_1,...,\Sigma_K$) on this mixture model. To solve this MAP estimation problem, we propose a novel EM-based algorithm.

4.1. EM algorithm

Before introducing the proposed algorithm, we first review the EM algorithm. Given a mixture model $p(x|\theta) = \sum_z p(x,z|\theta)$, where x is a data point and z is a hard assignment, we have:

[3]Note that we restrict each sample's assignment vector $[G_{\cdot 1},\ldots,G_{\cdot K}]$ in the $(K-1)$-simplex. This corresponds to the problem of computing the samples' soft assignment vectors, which are also useful in some situations (*e.g.*, semi-supervised learning), though our main focus is to obtain the discrete solution.

$$\begin{aligned}\log p(x|\theta) &= \log \sum_z p(x,z|\theta) \\ &= \log \sum_z q(z) \frac{p(x,z|\theta)}{q(z)} \\ &\geq \sum_z q(z) \log \frac{p(x,z|\theta)}{q(z)} = \mathrm{L}(q,\theta),\end{aligned} \tag{14}$$

where $\mathrm{L}(q,\theta)$ and $q(Z)$ are known as the evidence lower bound (ELBO) and the variational distribution, respectively. The EM algorithm solves the maximum likelihood (or MAP) estimation problem for $p(x|\theta)$ by maximizing ELBO (plus log-likelihood of the parameters θ on their prior $p(\theta)$, i.e., $\log p(\theta)$) via alternately iterating the (**E**)xpectation step and (**M**)aximization step, which updates $q(z)$ and θ, respectively. In the E-step, given the parameters $\hat{\theta}$, the optimal variational distribution $q(z)$ is $p(z|X,\hat{\theta})$. In the M-step, given the variational distribution $\hat{q}(z)$, the optimal parameters θ is computed by solving $\max_{\hat{\theta}} \sum_z \hat{q}(z) \log p(x,z|\hat{\theta})$ (or $\max_{\hat{\theta}} \sum_z \hat{q}(z) \log p(x,z|\hat{\theta}) + \log p(\hat{\theta})$ if the parameters' prior $p(\theta)$ is given). For more detail, see [4].

4.2. Proposed algorithm

We next introduce our algorithm for solving Eq. (13). The overall algorithm is shown in Algorithm 1. Since we follow the standard EM algorithm, which alternately iterates the E-step and M-step, we explain those two steps in the following.

[E-step] Updating $q(Z)$: Given the parameters $\Sigma_1, ..., \Sigma_K$ and G, the optimal variational distribution $q(Z)$ can be obtained as follows:

$$q(Z_{ij}=1) = \frac{G_{ij}\mathcal{N}(\boldsymbol{x}_i;0,\Sigma_j)}{\sum_k^K G_{ik}\mathcal{N}(\boldsymbol{x}_i;0,\Sigma_k)} \tag{15}$$

[M-step] Updating $\Sigma_1, ..., \Sigma_K$ and G: Given the variational distribution $q(Z)$, we want to solve the following problem:

$$\begin{aligned}&\max_{\substack{G\in\mathcal{H}\\ \Sigma_1,...,\Sigma_K\in\mathcal{S}}} \log p(G) + \sum_Z q(Z) \log p(X,Z|G,\Sigma_1,...,\Sigma_K) \\ =& \max_{\substack{G\in\mathcal{H}\\ \Sigma_1,...,\Sigma_K\in\mathcal{S}}} \frac{1-\eta}{\eta}\epsilon(G,M) + \\ & \sum_Z q(Z) \sum_i^N \log \sum_k^K Z_{ik} G_{ik} \mathcal{N}(\boldsymbol{x}_i;0,\Sigma_k) \\ =& \max_{\substack{G\in\mathcal{H}\\ \Sigma_1,...,\Sigma_K\in\mathcal{S}}} \frac{1-\eta}{\eta}\epsilon(G,M) + \\ & \sum_i^N \sum_k^K q(Z_{ik}=1)(\log G_{ik} + \log \mathcal{N}(\boldsymbol{x}_i;0,\Sigma_k))\end{aligned} \tag{16}$$

Following the standard EM algorithm for updating the Gaussian mixture model's parameters, we update covariance matrices $\Sigma_1, ..., \Sigma_K$, then update mixing coefficients G, and proceed to the E-step. When G is fixed, the optimal $\Sigma_1, ..., \Sigma_K$ can be obtained as (see the supplementary for the derivation of this):

$$\hat{\Sigma}_k = U_k \mathrm{Diag}(\max(\boldsymbol{d}_k, \sigma \mathbf{1}_D)) U_k^T, \tag{17}$$

where $\frac{1}{\sum_i^N q(Z_{ik}=1)} \sum_i^N q(Z_{ik}=1)\boldsymbol{x}_i\boldsymbol{x}_i^T = U_k \mathrm{Diag}(\boldsymbol{d}_k) U_k^T$. When $\Sigma_1, ..., \Sigma_K$ are fixed, we obtain the optimal parameters G by solving the following problem:

$$\begin{aligned}&\max_{G\in\mathcal{H}} \frac{1-\eta}{\eta}\epsilon(G,M) + \sum_i^N \sum_k^K q(Z_{ik}=1)\log G_{ik} \\ =& \max_{G\in\mathcal{R}^{N\times D}} \frac{1-\eta}{\eta}\epsilon(G,M) + \\ & \sum_i^N \sum_k^K q(Z_{ik}=1)\log G_{ik} - \iota_{\mathcal{H}}(G)\end{aligned} \tag{18}$$

where $\iota_{\mathcal{H}}(G)$ is an indicator function defined as $\iota_{\mathcal{H}}(G) = 0$ if $G \in \mathcal{H}$; otherwise, $\iota_{\mathcal{H}}(G) = \infty$. To solve this problem, we employ the proximal gradient descent (PGD) method, which iteratively updates G as follows:

$$G_{new} = \mathrm{prox}_{\gamma,\iota_{\mathcal{H}}}(G + \gamma(\frac{1-\eta}{\eta}\frac{\partial\epsilon(G,M)}{\partial G} + Q_Z \oslash G)), \tag{19}$$

where Q_Z is a matrix such that $(Q_Z)_{ij} = q(Z_{ij}=1)$, $\oslash$ is the element-wise division operator and

$$\mathrm{prox}_{\gamma,\iota_{\mathcal{H}}}(G^\dagger) = \arg\min_{\hat{G}} \frac{1}{2\gamma}||G^\dagger - \hat{G}||_F^2 + \iota_{\mathcal{H}}(\hat{G}). \tag{20}$$

Eq. (20) corresponds to the operator that projects each sample's assignment vector $[G_{\cdot 1}, \ldots, G_{\cdot K}]$ to its closest point in the $(K-1)$-simplex. Eq. (20) can be efficiently computed by some existing algorithms, *e.g.*, [33].

Initialization: Similar to the standard EM algorithms, our algorithm tends to lead a poor local minimum with random initialization. Fortunately, we empirically found that our algorithm tends to find a better local minimum by initializing $q(Z)$ with the assignment matrix obtained by the standard spectral clustering algorithm (i.e., $q(Z_{ij}=1) = G_{ij}$). From this observation, we initialize $q(Z)$ by spectral clustering.

4.3. Discretization

We then discretize the result obtained via the relaxed problem Eq. (12). Prior to explaining our discretization approach, we introduce the following proposition (see the supplementary for proof of this):

Algorithm 1 Subspace structure-aware spectral clustering

Input: Data matrix $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^T$ and initial variational distribution $q(Z)$.

repeat
  [M-step]
  Update the parameters $\Sigma_1, ..., \Sigma_K$ as follows:
  $\Sigma_k \leftarrow U_k \mathrm{Diag}(\max(\boldsymbol{d}_k, \sigma \mathbf{1}_D)) U_k^T$
  Update the parameters G by solving the following problem with PGD:
  $\max_{G \in \mathcal{R}^{N \times D}} \frac{1-\eta}{\eta} \epsilon(G, M) + \sum_i^N \sum_k^K q(Z_{ik} = 1) \log G_{ik} - \iota_{\mathcal{H}}(G)$
  [E-step]
  Update the variational distribution $q(Z)$ as follows:
  $q(Z_{ik} = 1) = \frac{G_{ik} \mathcal{N}(\boldsymbol{x}_i; 0, \Sigma_k)}{\sum_j^K G_{ij} \mathcal{N}(\boldsymbol{x}_i; 0, \Sigma_j)}$
until convergence
Discretize G as follows:
$G^*_{ik} =< k = \underset{k' \in \{1,...,K\}}{\arg\max} \hat{G}_{ik'} >$

Output: G^*

Proposition 1. *Suppose that $\eta = 1$, and $\Sigma_1, ..., \Sigma_K$ are fixed. If $\hat{G} \in \mathcal{H}$ is (one of) the optimal solution(s) for Eq. (13), $G^\dagger \in \mathcal{H}$, defined as follows, is also (one of) its optimal solution(s):*

$$G^\dagger_{ij} =< j = \underset{j' \in \{1,...,K\}}{\arg\max} \hat{G}_{ij'} >, \tag{21}$$

where $< \cdot >$ is 1 if the argument is true and 0 otherwise.

This proposition suggests that the quality of the solution is not significantly deteriorated by converting it to a new discrete solution by Eq. (21) when η is sufficiently close to one. Based on this observation, we convert the continuous solution to the final assignment by Eq. (21).

5. Experiments

To validate the effectiveness of our method, we conducted experiments on four real world applications: face clustering, object image clustering, hand-written digit clustering and motion clustering. To show the versatility of our method, we adopted four methods for computing self-representation matrices: sparse subspace clustering (SSC), low-rank representation (LRR), least square regression (LSR), and correlation-adaptive subspace segmentation (CASS). We also adopted the heuristics used by Elhamifar and Vidal [11] and Ji *et al*. [18] for converting self-representation matrices into affinity matrices. To evaluate each method's performance, we use two metrics: the clustering accuracy and normalized mutual information (NMI). We compare our method with multiclass spectral clustering (MSC) and spectral embedding clustering (SEC). MSC approximately solves the maximization problem of Eq. (4). SEC, as with our method, considers the geometrical information in the ambient space as well as Eq. (4), but its objective is not specialized for subspace clustering. For image datasets, we found that graph clustering methods result in chance-level results with some affinity computation methods. To avoid this phenomenon, following Hu *et al*. [15], we applied PCA to data and use top 20 components for experiments. Hyper-parameters for SSC, LRR, LSR, and CASS were chosen so that the best accuracy was achieved when MSC was used. MSC has no hyper-parameters, whereas SEC has two hyper-parameters, which are denoted as γ and μ in their paper. In our experiments, following the experiments by Nie *et al*., we set $\gamma = 1$. With regard to μ, we examined the cases of setting $\mu = 0, 0.01, 0.1, 1$, and show the best score among them for *each combination of datasets, methods, and metrics*[4]. Note that this experimental protocol leads SEC to overfitting of each setting. For our method, we set $\frac{1-\eta}{\eta} = 100.0$ unless specified otherwise.

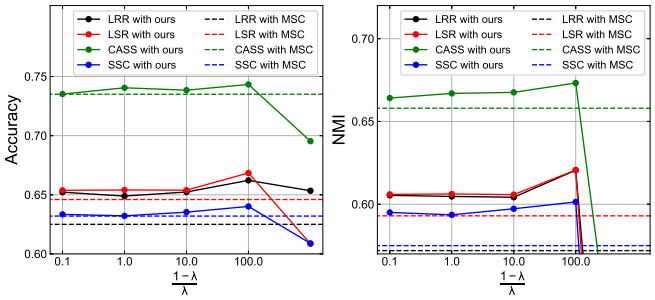

Figure 3. Experimental results on the EYaleB dataset. Since NMI computed with our method by setting $\frac{1-\eta}{\eta} = 1000.0$ are much less than the others, they are not shown in this figure. Best viewed in color.

5.1. Face Image Clustering

Settings: We first conducted an experiment on the Extended Yale Face Database B (EYaleB) [23]. EYaleB is a face image dataset, which consists of 2,432 facial images of 38 subjects under various illumination conditions. Each subject has 64 images. We used images resized to 48×42, which are provided by Elhamifar and Vidal [11]. We generated ten subsets by randomly choosing ten subjects and used them to evaluate each method's performance.

Results: Table 1 shows the clustering results on EYaleB. It can be seen that, in all cases, our method outperforms the other two methods. This suggests the effectiveness of our method.

Parameter sensitivity: We also tested the performance of our method by varying the parameter η. The results are

[4]Since the case of setting $\mu = 0$ corresponds to MSC, its scores are never less than MSC.

Method	SSC		LRR		LSR		CASS	
	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI
MSC	0.625	0.572	0.632	0.575	0.646	0.593	0.735	0.658
SEB	0.631	0.583	0.645	0.598	0.646	0.595	0.735	0.658
ours	**0.643**	**0.600**	**0.662**	**0.621**	**0.669**	**0.616**	**0.743**	**0.667**

Table 1. Experimental results on the EYaleB dataset.

Method	SSC		LRR		LSR		CASS	
	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI
MSC	0.781	0.873	0.763	0.838	0.792	0.882	**0.904**	**0.958**
SEB	0.781	0.873	0.763	**0.862**	0.792	0.882	**0.904**	**0.958**
ours	**0.788**	**0.885**	**0.781**	0.861	**0.797**	**0.891**	**0.904**	**0.958**

Table 2. Experimental results on the COIL100 database.

shown in Fig. 3. In all cases, the peak of the metrics is when $\frac{1-\eta}{\eta} = 100.0$, and the result gets worse when η is larger or smaller than that. This suggests that both the graph clustering term $\epsilon(G, M)$ and the geometry-aware term $\mathrm{r}(G, X)$, rather than one or the other, should be considered for achieving better clustering results. In addition, in all cases, we can observe that the clustering result is much worse than its peak when $\frac{1-\eta}{\eta}$ is too large. This is because, when η is close to zero, a continuous solution of Eq. (12) is likely to be approximately factorized in the form of $G = \mathbf{1}_N \boldsymbol{v}^T$[5], where $\boldsymbol{v}$ is a vector in the $(K-1)$-simplex, and hence all the data points tend to be assigned to a single cluster. Furthermore, our method tends to outperform the MSC baseline even when $\frac{1-\eta}{\eta}$ is much smaller than its peak. Based on this observation, we suggest selecting small η when it cannot be tuned (*e.g.*, when a validation dataset is not available).

5.2. Object Image Clustering

Settings: We next conducted an experiment on the COIL100 database [31], which consists of 7200 images of 100 object categories such as a duck and a car. Each class has 72 images, and each image has 32×32 pixels. As with the experiments on EYaleB, we generated ten subsets by randomly choosing ten categories and used them to evaluate each method's performance.

Results: Table 2 shows the clustering results on COIL100. It can be seen that, in most cases, our method outperforms the other two methods. This also suggests the effectiveness of our method.

[5]In particular, we can show that, when $\eta = 0$, an assignment matrix G is always one of the optimal solutions of Eq. (12) if it can be factorized in the form of $G = \mathbf{1}_N \boldsymbol{v}^T$, where $\boldsymbol{v}$ is a vector in the $(K-1)$-simplex and satisfies $\boldsymbol{v}_i > 0$ for all i. For proof, see supplementary material.

5.3. Hand-written Digit Clustering

We next conducted an experiment on the USPS handwritten digit dataset [16], which consists of ten digit categories. Each image has 16×16 pixels. We generated ten subsets by randomly choosing 100 images from each category and used them to evaluate each method's performance.

Results: Table 3 shows the clustering results on COIL100. It can be seen that, in most cases, our method outperforms the other two methods. This also suggests the effectiveness of our method.

5.4. Trajectory Clustering

We next conduct an experiment on the Hopkins 155 motion segmentation database (Hopkins155) [40]. Hopkins155 is a video dataset, which consists of 155 video sequences with multiple 2D trajectories. Each sequence contains two or three motions. We used all 155 video sequences to evaluate each method's performance.

Results: Table 4 shows the clustering results on Hopkins155. It can be seen that, in most cases, our method outperforms the other two methods. This also suggests the effectiveness of our method.

5.5. Introducing Our Method into Structured Sparse Subspace Clustering

Recently, subspace clustering frameworks have been proposed that unify graph construction and graph clustering into a single optimization problem [13, 24, 48]. To investigate whether our method is also effective for such frameworks, we conducted experiments with a combination of our method and structured sparse subspace clustering (S3C) [24], one of the representative unified framework, on the EYaleB dataset.

Method	SSC		LRR		LSR		CASS	
	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI
MSC	0.749	0.617	0.765	0.773	0.766	0.768	0.784	0.789
SEB	0.749	0.677	0.771	0.784	0.766	0.768	0.784	**0.799**
ours	**0.756**	**0.787**	**0.777**	**0.797**	**0.786**	**0.797**	**0.787**	0.798

Table 3. Experimental results on the USPS dataset.

Method	SSC		LRR		LSR		CASS	
	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI	Accuracy	NMI
MSC	0.939	0.840	0.937	0.830	**0.980**	**0.936**	0.881	0.733
SEB	0.939	0.840	0.937	0.830	**0.980**	**0.936**	0.881	0.733
ours	**0.947**	**0.860**	**0.947**	**0.861**	0.979	**0.936**	**0.894**	**0.767**

Table 4. Experimental results on the Hopkins dataset.

S3C: S3C simultaneously optimizes a self-representation matrix Z and an assignment matrix G by minimizing the following objective:

$$\min_{Z\in\{Z|Z\in\mathrm{R}^{N\times N},Z_{ii}=0\},G} ||Z||_{1,G} + \tau||X - XZ||_F^2, \quad (22)$$

where $||Z||_{1,G} = \sum_{i,j}|Z_{ij}|(1+\frac{\alpha}{2}||\boldsymbol{g}^i - \boldsymbol{g}^j||_2^2)$, and α is a hyper-parameter. S3C solves this objective by alternately updating Z and G while fixing the other. More specifically, alternating direction method of multipliers (ADMM) is used for updating Z, whereas the two following methods have been proposed for updating G:

- G is updated with a binary assignment matrix computed by applying spectral clustering to the affinity matrix $M = |Z| + |Z^T|$. This method is called Hard-S3C.

- G is updated with a real-valued assignment matrix computed by concatenating each data point's normalized K-dimensional embedding, produced by applying spectral clustering to the affinity matrix $M = |Z| + |Z^T|$ (i.e., before the k-means step). The final (binary) assignment matrix G is computed by simply applying spectral clustering to a final affinity matrix M. This method is called Soft-S3C.

Experinmental settings: In this experiment, we investigated whether the clustering results are consistently improved by replacing spectral clustering in both Hard-S3C and Soft-S3C with our method. More specifically, we compared four methods: Hard-S3C, Hard-S3C with our method, Soft-S3C and Soft-S3C with our method. Following Li *et al*. [24], we set $\alpha = 0.1$ and $\alpha = 1.0$ for Hard-S3C and Soft-S3C, respectively. We also set $\tau = 1.0$. We iterated updating Z and G ten times and compute performance at the end of each iteration. Other experimental settings were same as in the EYaleB experiments in the section 5.1.

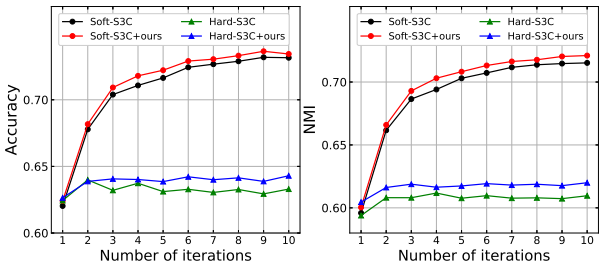

Figure 4. Experimental results on the EYaleB dataset. For each method, accuracy and NMI after each iteration are shown in this figure. Best viewed in color.

Experimental results: The experimental results are shown in Fig. 4. From these results, we can observe that the clustering results are improved by introducing our method into both Hard-S3C and Soft-S3C. These results indicate the effectiveness of using our method even for frameworks that unify graph construction and graph clustering.

6. Conclusion

We proposed a novel graph clustering framework for robust subspace clustering. By incorporating a novel geometry-aware term with the spectral clustering objective, we encourage our framework to be robust to noise and outliers in given affinity matrices. We also developed a novel EM algorithm for optimization. Through extensive experiments on four real-world datasets, we demonstrated that the proposed method outperforms existing methods. Moreover, we also demonstrated that our method is also effective for frameworks that unify graph construction and clustering steps. In the future, we will aim at providing a further theoretical analysis of the proposed method.

References

[1] Pankaj K Agarwal and Nabil H Mustafa. K-means projective clustering. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2004.

[2] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. *Automatic subspace clustering of high dimensional data for data mining applications*, volume 27. ACM, 1998.

[3] Laurent Bako. Identification of switched linear systems via sparse optimization. *Automatica*, 47(4):668–677, 2011.

[4] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[5] Terrance E Boult and L Gottesfeld Brown. Factorization-based segmentation of motions. In *IEEE Workshop on Visual Motion*, 1991.

[6] Paul S Bradley and Olvi L Mangasarian. K-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.

[7] Guangliang Chen and Gilad Lerman. Spectral curvature clustering (scc). *IJCV*, 81(3):317–330, 2009.

[8] Yudong Chen, Ali Jalali, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. *JMLR*, 15(1):2213–2238, 2014.

[9] João Paulo Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *IJCV*, 29(3):159–179, 1998.

[10] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *CVPR*, 2009.

[11] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *TPAMI*, 35(11):2765–2781, 2013.

[12] Brian Eriksson, Laura Balzano, and Robert Nowak. High-rank matrix completion. In *Artificial Intelligence and Statistics*, 2012.

[13] Jiashi Feng, Zhouchen Lin, Huan Xu, and Shuicheng Yan. Robust subspace segmentation with block-diagonal prior. In *CVPR*, 2014.

[14] C William Gear. Multibody grouping from motion images. *IJCV*, 29(2):133–150, 1998.

[15] Han Hu, Zhouchen Lin, Jianjiang Feng, and Jie Zhou. Smooth representation clustering. In *CVPR*, 2014.

[16] Jonathan J. Hull. A database for handwritten text recognition research. *TPAMI*, 16(5):550–554, 1994.

[17] Amin Jalali and Rebecca Willett. Subspace clustering via tangent cones. In *NIPS*, 2017.

[18] Pan Ji, Mathieu Salzmann, and Hongdong Li. Efficient dense subspace clustering. In *WACV*, 2014.

[19] Pan Ji, Mathieu Salzmann, and Hongdong Li. Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data. In *ICCV*, 2015.

[20] Ken-ichi Kanatani. Motion segmentation by subspace separation and model selection. In *CVPR*, 2001.

[21] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *TKDD*, 3(1):1, 2009.

[22] Hanjiang Lai, Yan Pan, Canyi Lu, Yong Tang, and Shuicheng Yan. Efficient k-support matrix pursuit. In *ECCV*, 2014.

[23] Kuang-Chih Lee, Jeffrey Ho, and David J Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *TPAMI*, 27(5):684–698, 2005.

[24] Chun-Guang Li, Chong You, and René Vidal. Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework. *IEEE Trans. Image Processing*, 26(6):2988–3001, 2017.

[25] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010.

[26] Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. Correlation adaptive subspace segmentation by trace lasso. In *ICCV*, 2013.

[27] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. In *ECCV*, 2012.

[28] Yi Ma, Harm Derksen, Wei Hong, and John Wright. Segmentation of multivariate mixed data via lossy data coding and compression. *TPAMI*, 29(9), 2007.

[29] Brian McWilliams and Giovanni Montana. Subspace clustering of high-dimensional data: a predictive approach. *Data Mining and Knowledge Discovery*, 28(3):736–772, 2014.

[30] Quanyi Mo and Bruce A Draper. Semi-nonnegative matrix factorization for motion segmentation with missing data. In *ECCV*, 2012.

[31] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-100). 1996.

[32] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002.

[33] Feiping Nie, Xiaoqian Wang, Michael I Jordan, and Heng Huang. The constrained laplacian rank algorithm for graph-based clustering. In *AAAI*, 2016.

[34] Feiping Nie, Dong Xu, Ivor Wai-Hung Tsang, and Changshui Zhang. Spectral embedded clustering. In *IJCAI*, 2009.

[35] Lance Parsons, Ehtesham Haque, and Huan Liu. Subspace clustering for high dimensional data: a review. *Acm Sigkdd Explorations Newsletter*, 6(1):90–105, 2004.

[36] Shankar Rao, Roberto Tron, Rene Vidal, and Yi Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *TPAMI*, 32(10):1832–1845, 2010.

[37] Yasuyuki Sugaya and Kenichi Kanatani. Geometric structure of degeneracy for multi-body motion segmentation. In *International Workshop on Statistical Methods in Video Processing*, 2004.

[38] Michael E Tipping and Christopher M Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482, 1999.

[39] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[40] Roberto Tron and René Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *CVPR*, 2007.

[41] Paul Tseng. Nearest q-flat to m points. *JOTA*, 105(1):249–252, 2000.

[42] René Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011.

[43] René Vidal and Paolo Favaro. Low rank subspace clustering (lrsc). *PRL*, 43:47–61, 2014.

[44] René Vidal and Richard Hartley. Motion segmentation with missing data using powerfactorization and gpca. In *CVPR*, 2004.

[45] Rene Vidal, Yi Ma, and Shankar Sastry. Generalized principal component analysis (gpca). *TPAMI*, 27(12):1945–1959, 2005.

[46] René Vidal, Stefano Soatto, Yi Ma, and Shankar Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *CDC*, 2003.

[47] Shusen Wang, Xiaotong Yuan, Tiansheng Yao, Shuicheng Yan, and Jialie Shen. Efficient subspace segmentation via quadratic programming. In *AAAI*, 2011.

[48] Xiaobo Wang, Xiaojie Guo, Zhen Lei, Changqing Zhang, and Stan Z Li. Exclusivity-consistency regularized multi-view subspace clustering. In *CVPR*, 2017.

[49] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, 2006.

[50] Allen Y Yang, Shankar R Rao, and Yi Ma. Robust statistical estimation and segmentation of multiple subspaces. In *CVPR Workshop*, 2006.

[51] Stella X Yu and Jianbo Shi. Multiclass spectral clustering. In *ICCV*, 2003.

[52] Teng Zhang, Arthur Szlam, and Gilad Lerman. Median k-flats for hybrid linear modeling with many outliers. In *ICCV Workshop*, 2009.