Subspace Clustering with A Hybrid Adaptive Graph Filter

Lai Wei* Mingyuan Xi

College of information engineering, Shanghai Maritime University, Shanghai, China

*weilai@shmut.edu.cn

Abstract

Subspace clustering is a powerful tool for grouping data samples into their underlying subspaces. We propose an advanced subspace clustering algorithm called SCHAGF (Subspace Clustering with A Hybrid Adaptive Graph Filter). SCHAGF leverages the obtained reconstruction coefficient matrix to design a low-pass graph filter and a high-pass graph filter simultaneously. These graph filters are then integrated into a hybrid graph filter, which is used for designing a feature extraction function and a constraint for the reconstruction coefficient matrix. Then the hybrid graph filter and the coefficient matrix are iteratively updated to achieve optimal values. O

Subspace clustering

The generalized framework of subspace clustering could be expressed as follows:

$$\min_{\mathbf{C}} \|\phi(\mathbf{X}) - \mathbf{C}\phi(\mathbf{X})\|_s + \beta\psi(\mathbf{C})$$

Where $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{C} \in \mathbb{R}^{n\times n}$ is the feature matrix and the required reconstruction coefficient matrix. $\phi(\mathbf{X})$ is a function that is used to extract latent features from original data samples. $\psi(\mathbf{C})$ is usually some kind of constraint of $\mathbf{C}$.

After $\mathbf{C}$ is obtained, $\mathbf{W}$ is defined as $\mathbf{W} = (\mathbf{C} + \mathbf{C}^{\top})/2$. Then, normalized cuts (Ncuts) is performed on W to produce the final clustering results

Graph filter

- Suppose an undirected graph $\mathbb{G}$ has n vertices with $\mathbf{X}$ being the feature matrix corresponding to these vertices and $\mathbf{W} \in \mathbb{R}^{n\times n}$ represents the edge weights matrix. The normalized adjacency matrix and the normalized Laplacian matrix of this graph are defined as $\tilde{\mathbf{A}} = \mathbf{W} + \mathbf{I}$ and $\mathbf{L} = \mathbf{I} - \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ respectively. $\tilde{\mathbf{D}}$ is a diagonal matrix satisfying $[\tilde{\mathbf{D}}]_{ii} = \sum_{i=1}^{n}[\tilde{\mathbf{A}}]_{ij}$.
- Then the low-pass graph filter $\mathbf{F}_l = \mathbf{I} - \frac{\mathbf{L}}{2}$, the high-pass graph filter $\mathbf{F}_h = \frac{\mathbf{L}}{2}$

The proposed method

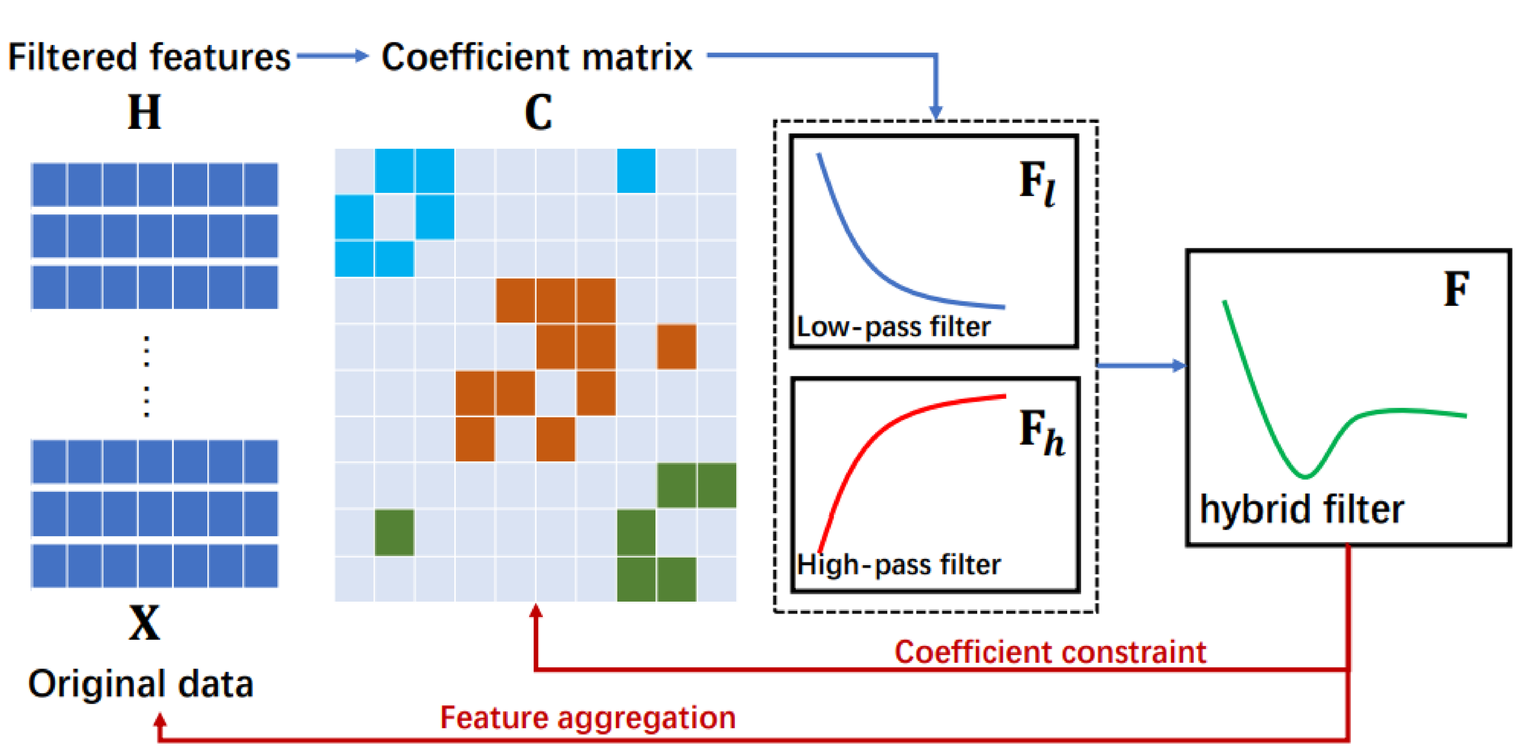

Figure 1: The overview of the proposed method. The low-pass graph filter $\mathbf{F}_l$ and high-pass graph filter $\mathbf{F}_h$ are constructed by using the required reconstruction coefficient matrix $\mathbf{C}$. The hybrid filter $\mathbf{F}$ is integrated by $\mathbf{F}_l$ and $\mathbf{F}_h$. Then the updated $\mathbf{F}$ will be used to obtain the aggregated features and construct a constraint of the coefficient matrix $\mathbf{C}$.

The proposed method contains three main steps:

A. Designing a hybrid graph filter. We first assume that a coefficient matrix $\mathbf{C}$ has been gotten. Then a low-pass graph filter and a high-pass graph filter could be defined as $\mathbf{F}_l = \mathbf{I} - \frac{\mathbf{L}}{2} = \frac{3\mathbf{I}}{4} + \frac{3\mathbf{C}}{4}$ and $\mathbf{F}_h = \frac{\mathbf{L}}{2} = \frac{\mathbf{I}}{4} - \frac{\mathbf{C}}{4}$

Here, we require $\mathbf{C} \geq \mathbf{0}, \mathbf{C} = \mathbf{C}^{\top}, \mathbf{C1} = \mathbf{C}, \mathrm{diag}(\mathbf{C}) = \mathbf{0}$. Then the affinity graph $\mathbf{W} = \mathbf{C}$, hence the graph filter will have the above formulation.

B. Smoothing the data by the defined hybrid graph filter. Namely,

$$\begin{aligned}\mathbf{H} &= \phi(\mathbf{X}) = \mathbf{FX} = \eta\mathbf{F}_l\mathbf{X} + (1-\eta)\mathbf{F}_h\mathbf{X} \\ &= (2\eta + 1)/4\mathbf{X} + (2\eta - 1)/4\mathbf{CX}\end{aligned}$$

C. Defining a regularizer of C. Namely,

$$\psi(\mathbf{C}) = \|\mathbf{C} - \mathbf{FC}\|_F^2 = \|(3-2\eta)\mathbf{C} - (2\eta-1)\mathbf{C}^2\|_F^2$$

Finally, the proposed problem could be obtained:

$$\begin{aligned}\min_{\mathbf{C}^i} \quad & \|4\mathbf{H} - (2\eta+1)\mathbf{X} - (2\eta-1)\mathbf{CX}\|_F^2 + \alpha\|\mathbf{H} - \mathbf{CH}\|_F^2 \\ & + \beta\|(3-2\eta)\mathbf{C} - (2\eta-1)\mathbf{C}^2\|_F^2 \\ s.t. \quad & \mathbf{C} \geq \mathbf{0}, \mathbf{C} = \mathbf{C}^{\top}, \mathbf{C1} = \mathbf{C}, \mathrm{diag}(\mathbf{C}) = \mathbf{0}\end{aligned}$$

Experiments

Dataset	Samples	Classes	Size
ORL	400	40	32×32
EYALEB	2432	38	48×42
Umist	480	20	32×32
COIL20	1440	20	32×32
COIL40	2880	40	32×32
MNIST	1000	10	28×28

Table 1: Detailed information of the benchmark datasets

Dataset	Metric	Model												
		Classical Methods					GF-related methods			T-related methods			Ours	
		SSC	LRR	LSR	LRSC	BDR	FLSR	GCSC	AGCSC	TRR	FTRR	TAGCSC	SCHAGF	TSCHAGF
ORL	ACC	72.50	72.75	75.50	77.00	78.25	73.50	73.75	80.50	85.75	79.75	**86.25**	83.50	**88.75**
	NMI	84.52	83.26	84.97	85.02	88.46	83.84	83.98	88.51	91.49	87.60	**92.84**	90.10	**94.04**
EYALEB	ACC	55.15	73.48	74.05	75.64	76.56	72.94	62.50	84.79	91.65	91.90	**92.31**	86.51	**95.60**
	NMI	55.71	77.11	78.13	78.32	80.34	76.57	68.04	87.37	93.03	93.18	**94.56**	88.81	**94.16**
Umist	ACC	52.92	64.79	64.17	63.33	64.92	60.62	79.58	81.04	74.38	69.37	**90.83**	**82.29**	**90.83**
	NMI	75.38	73.41	73.17	72.02	75.13	70.72	86.44	87.46	80.63	78.49	**94.99**	87.78	**94.03**
COIL20	ACC	68.61	70.14	69.17	71.81	71.71	69.93	79.79	88.75	85.97	86.53	**98.96**	89.79	**99.31**
	NMI	66.85	76.43	74.17	77.27	80.51	77.19	79.79	93.38	90.23	91.17	**99.11**	94.64	**99.31**
COIL40	ACC	63.13	60.42	56.88	58.23	57.25	62.88	73.72	78.12	65.00	71.39	**92.60**	78.96	**93.13**
	NMI	82.28	76.29	75.87	74.48	76.73	76.26	84.32	89.21	79.83	82.46	**97.32**	89.42	**97.86**
MNIST	ACC	63.70	64.60	62.80	64.30	61.30	65.10	67.70	71.40	67.70	66.40	**72.80**	71.80	**77.00**
	NMI	59.75	60.67	57.18	58.91	54.76	61.10	61.99	65.84	64.43	63.21	**67.54**	65.82	**72.95**

Table 2: Clustering results (in %) of evaluated methods on the used benchmark data sets. The best results are emphasized in red and bold and the second-best results are denoted in bold.

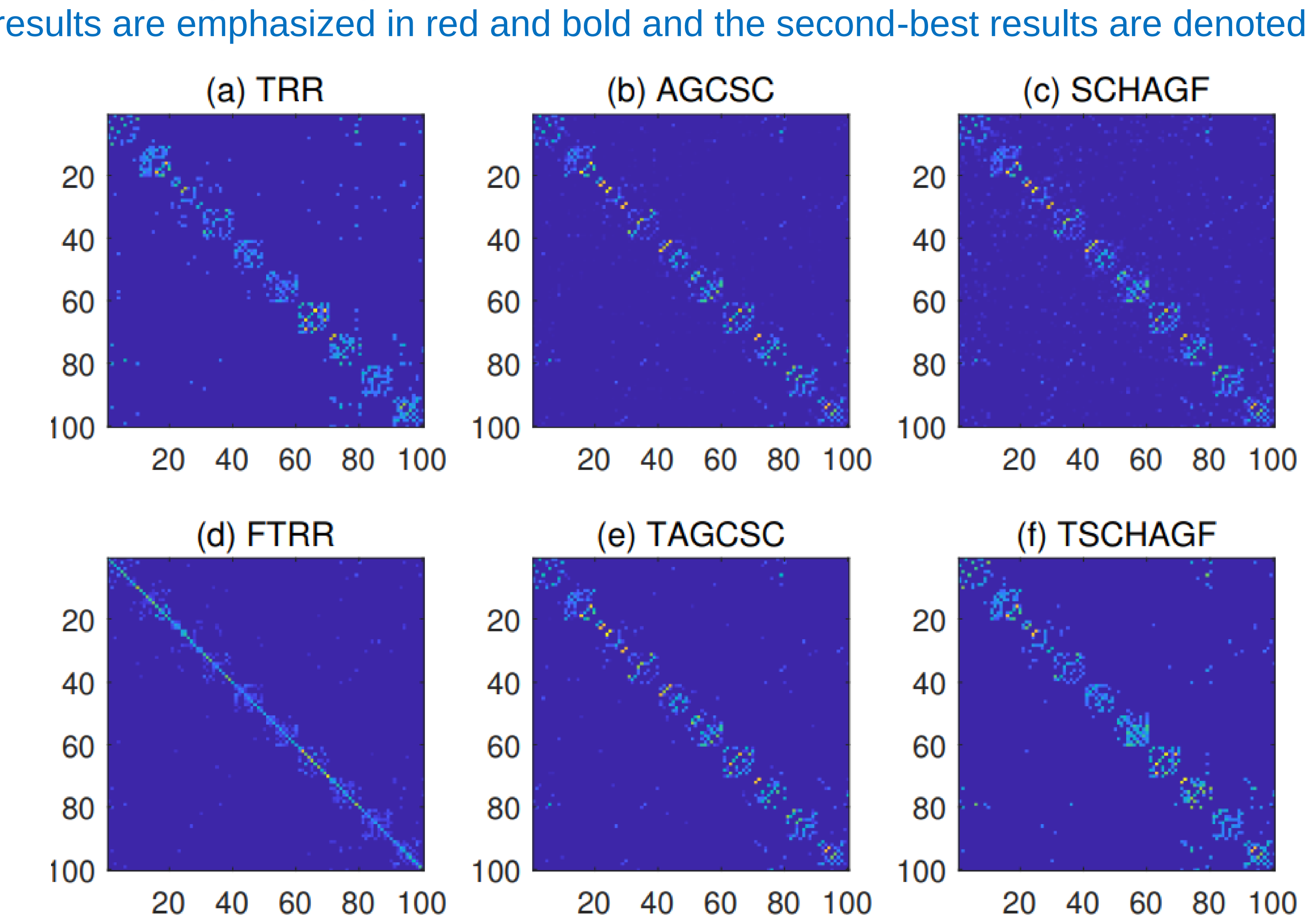

Figure 2: The obtained coefficient matrices obtained by (a) TRR, (b) AGCSC, (c) SCHAGF, (d) FTRR, (e) TAGCSC, and (f) TSCHAGF.

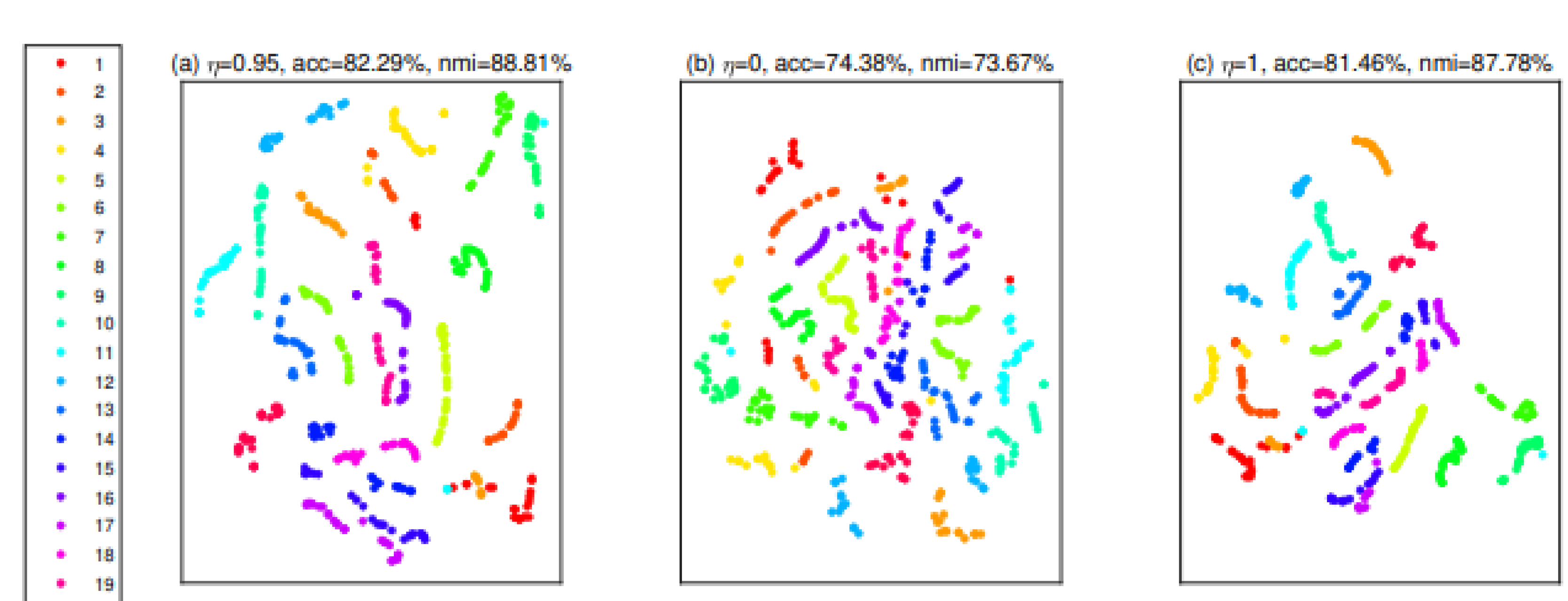

Figure 3: The aggregated features with different η.

Conclusions

Plenty of subspace clustering experiments have established that SCHAGF is superior to related algorithms, and by incorporating the thresholding technique, thresholding SCHAGF (TSCHAGF) is shown to outperform some deep models.