Hierarchical Multiple Kernel Clustering

Authors: Jiyuan Liu, Xinwang Liu*, Yuexiang Yang, Siwei Wang, Sihang Zhou

国防科技大学
National University of Defense Technology

Background

Current multiple kernel clustering (MKC) methods can be grouped into two categories, i.e.

- **Early-fusion** ones directly learn a consensus kernel or graph from multiple ones, afterwards generate the final partition. Fig. (i) - (ii)
- **Late-fusion** ones firstly obtain multiple partitions from each kernel independently with basic kernel clustering algorithms , then construct the final clustering assignments on them. Fig. (iii)

Problem and Solution

Problem. Both of them directly distill the clustering information from matrices of $\mathbb{R}^{n\times n}$ to $\mathbb{R}^{n\times k}$. This sudden drop of dimension would result in the loss of advantageous details for clustering.

Solution. we generate a sequence of intermediary matrices with size $\mathbb{R}^{n\times c_t}$, in which $n > c_1 > \cdots > c_s > k$. A consensus partition with size $\mathbb{R}^{n\times k}$ is simultaneously learned and conversely guides the construction of intermediary matrices. Fig. (iv)

Formulation

$$\max_{\mathbf{H},\{\mathbf{H}_p\}_{p=1}^{m},\boldsymbol{\beta},\boldsymbol{\gamma}} \sum_{p=1}^{m} \gamma_p^{(1)} Tr\left(\mathbf{K}_p\mathbf{H}_p^{(1)}\mathbf{H}_p^{(1)T}\right) + \sum_{p=1}^{m} \beta_p Tr(\mathbf{H}_p^{(s)}\mathbf{H}_p^{(s)\mathrm{T}}\mathbf{H}\mathbf{H}^{\mathrm{T}})$$

$$+\sum_{t=2}^{s}\sum_{p=1}^{m} \gamma_p^{(t)} Tr(\mathbf{H}_p^{(t-1)}\mathbf{H}_p^{(t-1)\mathrm{T}}\mathbf{H}_p^{(t)}\mathbf{H}_p^{(t)\mathrm{T}})$$

$$s.t.\ \mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}_k, \mathbf{H}\in\mathbb{R}^{n\times k}, \mathbf{H}_p^{(t)T}\mathbf{H}_p^{(t)} = \mathbf{I}_{c_t}, \mathbf{H}_p^{(t)}\in\mathbb{R}^{n\times c_t},$$

$$n > c_1 > \cdots > c_s > k, \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta} = 1, \beta_p \geq 0, \boldsymbol{\gamma}^{(t)T}\boldsymbol{\gamma}^{(t)} = 1, \gamma_p^{(t)} \geq 0.$$

Validation

Setting. According to the layer number of intermediary matrices, we instance the proposed model into HMKC-1 and HMKC-2.

Intermediary matrix. It can be seen that a more and more clear clustering structure is presented along with the clustering process. Fig. (a) – (d)

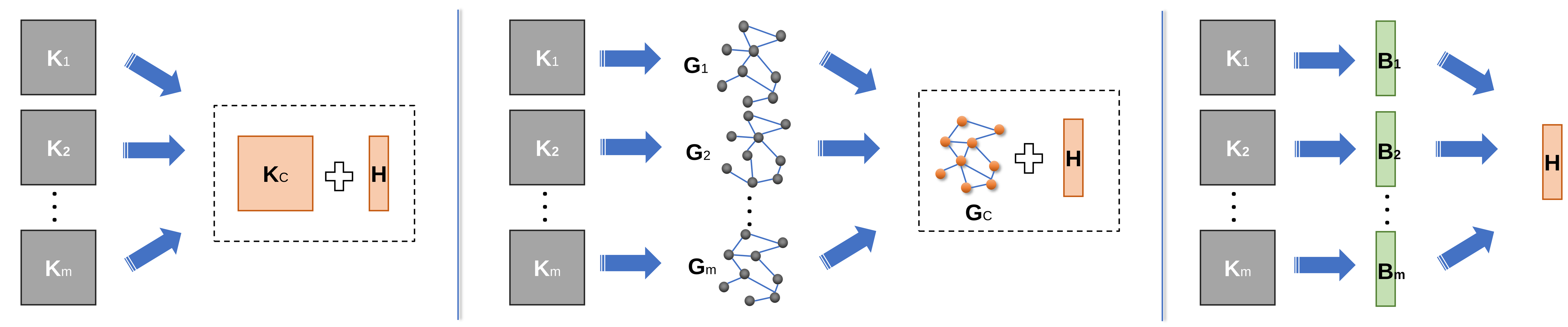
Performance. HMKC consistently and largely outperforms the other algorithms across all datasets. Especially, the proposed algorithm exhibits excellent performances on CCV, Flower17 and Flower102, where around 5/10% increases are obtained. Table

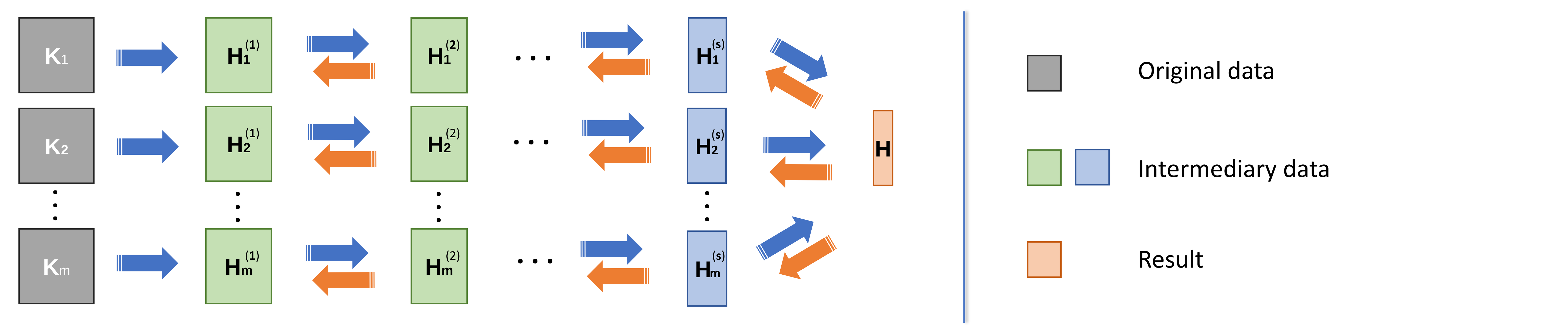
Note

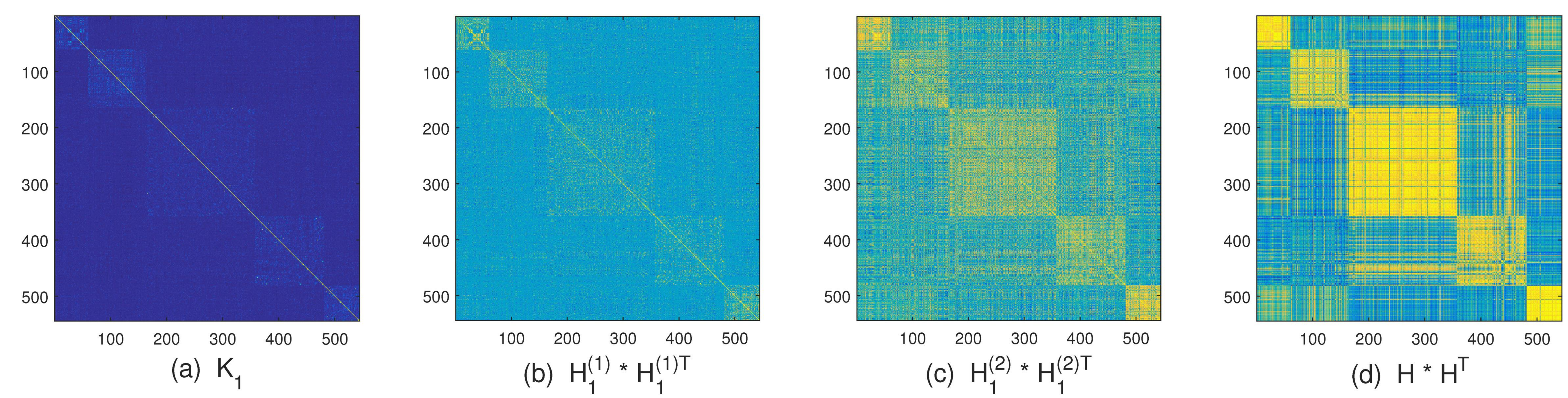
Email: liujiyuan13@nudt.edu.cn

Paper: https://liujiyuan13.github.io/pubs/HMKC.pdf

Code: https://github.com/liujiyuan13/HMKC-code_release







ACC	AR10P	BBCSport	CCV	Flower17	Flower102	Heart	Ionosphere	Plant
AMKKM	38.46	65.99	19.74	51.03	27.29	82.22	61.25	61.70
SBKKM	49.23	76.65	20.08	42.06	33.13	76.3	70.09	51.60
CSRC	36.92	67.65	23.4	51.69	35.19	80.37	75.78	54.57
MKKM	37.69	66.36	18.01	45.37	21.96	53.33	63.82	56.38
RMSC	34.62	86.03	16.29	52.57	32.97	83.33	84.62	55.53
RMKC	43.85	66.36	19.74	52.35	33.55	82.22	66.10	61.70
RMKKM	32.31	53.13	17.11	53.16	29.61	76.30	65.81	55.32
MKCMR	43.08	66.36	22.47	60.00	40.27	83.33	61.25	62.87
ONKC	48.46	67.65	23.12	59.85	41.26	83.7	63.82	*64.89*
LFAM	46.92	72.61	26.66	59.63	44.61	82.22	68.09	62.77
SPC	38.46	83.09	-	57.50	-	75.93	71.23	60.43
MVCC	43.85	74.26	22.10	51.47	37.23	82.96	55.56	55.64
SPMKC	54.62	40.81	13.54	35.81	-	57.04	72.93	53.94
HMKC-1	*56.15*	*89.52*	*36.57*	*66.91*	*46.84*	**86.67**	*86.32*	64.68
HMKC-2	**60.00**	**90.99**	**37.37**	**71.18**	**50.32**	*86.30*	**86.89**	**67.02**