Dynamic Late Fusion Multiple Kernel Clustering with Robust Tensor Learning via Min-Max Optimization

Yi Zhang, Siwei Wang, Jiyuan Liu, Shengju Yu, Zhibin Dong, Suyuan Liu, Liu Xinwang Liu* and En Zhu

E-mail: {zhangy,xinwangliu}@nudt.edu.cn

Code: github.com/ethan-yizhang/DLEFT-MKC/

xinwangliu.github.io/

April 8, 2025

Content

Content

Observation

Existing late fusion methods encounter three significant limitations:

1. reliance on fixed base partition matrices that do not adaptively optimize during the clustering process, thereby constraining their performance to the inherent representational capabilities of these matrices;
2. a focus on adjusting kernel weights to explore inter-view consistency and complementarity, which often neglects the intrinsic high-order correlations among views, thereby limiting the extraction of comprehensive multiple kernel information;
3. a lack of adaptive mechanisms to accommodate varying distributions within the data, which limits robustness and generalization.

contribution

The primary contributions of this paper are summarized as follows,

1. This study is the first to incorporate a min-max optimization paradigm into tensor-based MKC, which represents a pioneering exploration of min-max optimization aimed at enhancing both performance and robustness in clustering.
2. We propose a groundbreaking approach for the dynamical reconstruction and calibration of base partition matrices from LFMVC, effectively overcoming their representational bottleneck and enhancing clustering performance.
3. We stack the reconstructed representations into tensors and optimize dynamic partitions using tensor techniques, significantly enhancing our ability to learn high-order correlations and uncover latent structures across views.
4. To solve the resultant optimization problem, we design an innovative and efficient strategy to combine the RGDM with ADMM. Extensive experimental results across various benchmarks validate both the effectiveness and efficiency of our proposed algorithm.

Content

Late Fusion Multi-view Clustering

Given n samples in k clusters among m views, its optimization goal can be mathematically expressed as

$$\begin{aligned} \max_{\mathbf{H},\{\mathbf{W}_p\}_{p=1}^m,\boldsymbol{\beta}} \quad & \mathrm{Tr}(\mathbf{H}^\top \textstyle\sum_{p=1}^m \beta_p \mathbf{H}_p \mathbf{W}_p) \\ s.t. \quad & \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \mathbf{W}_p^\top \mathbf{W}_p = \mathbf{I}_k, \textstyle\sum_{p=1}^m \beta_p^2 = 1, \beta_p \geq 0, \forall p, \end{aligned} \tag{1}$$

where the objective denotes the alignment between the consensus partition matrix $\mathbf{H} \in \mathbb{R}^{n\times k}$ and a group of pre-calculated base partition matrices $\{\mathbf{H}_p\}_{p=1}^m$, and $\mathbf{W}_p \in \mathbb{R}^{k\times k}$ is the p-th transformation matrix. After obtaining the consensus partition matrix $\mathbf{H}$, a standard k-means algorithm is applied to compute the discrete cluster assignments.

Preliminaries of 3-Order Tensor

tensor singular value decomposition (t-SVD): For a tensor $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its t-SVD can be factorized as $\mathbf{A} = \mathbf{U} * \mathbf{S} * \mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathbf{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ are orthogonal tensors, and $\mathbf{V} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor, whose each frontal slices is a diagonal matrix. According to the literature kilmer2013third,kilmer2011factorization, the above t-SVD problem can be efficiently settled by matrix SVD in the Fourier domain, i.e., $\overline{\boldsymbol{A}}_k = \overline{\boldsymbol{U}}_k \overline{\boldsymbol{S}}_k \overline{\boldsymbol{V}}_k^\top, k = 1, 2, \cdots, n_3$.

t-SVD based tensor nuclear norm (t-TNN): For a tensor $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its t-TNN can be expressed as,$||\mathbf{A}||_{\circledast} = \sum_{k=1}^{n_3} ||\overline{\boldsymbol{A}}_k||_* = \sum_{k=1}^{n_3} \sum_{i=1}^{min(n_1, n_2)} \sigma_i(\overline{\boldsymbol{A}}_k)$, where $\sigma_i(\overline{\boldsymbol{A}}_k)$ denotes the i-th largest singular value of $\overline{\boldsymbol{A}}_k$.

Note that according to zhang2014novel,semerci2014tensor, t-TNN is proven to be valid and the tightest convex relaxation to l_1-norm of the tensor multi-rank.

Content

Framework diagram of DLEFT-MKC

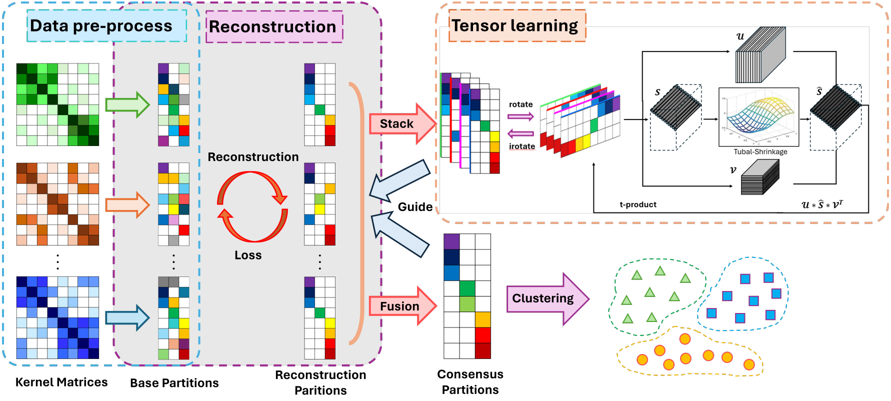

Figure 1: The framework diagram of the proposed DLEFT-MKC algorithm.

Formulation of DLEFT-MKC

We first introduce the dynamic partitions $\{\hat{\boldsymbol{F}}_p\}_{p=1}^m$ to reconstruct the base partition matrices $\{\boldsymbol{F}_p\}_{p=1}^m$ of late fusion strategy based MKC. Next, We maximize the alignment between the reconstructed and base partitions to ensure the quality of reconstruction, and dynamically optimize this alignment during the subsequent iterations. Furthermore, to explore and capture higher-order intrinsic correlations across views, we stack the dynamic reconstruction $\{\hat{\boldsymbol{F}}_p\}_{p=1}^m$ into a tensor $\hat{\mathbf{F}}$ and optimize it with t-TNN. Additionally, we impose an orthogonal constraint on it to preserve its capacity to reveal the clustering structure. Thus we can obtain the following expression:

$$\max_{\hat{\mathbf{F}},\,\hat{\boldsymbol{F}}_p} \sum_{p=1}^{m} \mathrm{Tr}(\hat{\boldsymbol{F}}_p^\top \boldsymbol{F}_p) - \rho||\hat{\mathbf{F}}||_{\circledast}, \ s.t. \ \ \hat{\boldsymbol{F}}_p^\top \hat{\boldsymbol{F}}_p = \boldsymbol{I}, \forall p. \tag{2}$$

Formulation of DLEFT-MKC

Next, we attempt to directly learn the consensus clustering partition by incorporating Eq.(2) and permutation matrices $\{\boldsymbol{T}_p\}_{p=1}^m$ with the paradigm of LFMVC. In addition, due to the different contributions of various views, we assign kernel weight coefficients γ to each view in order to sufficiently mine and learn each kernel view with particular emphasis. Finally, we pioneeringly introduce the min-max paradigm into the resultant objective function, which minimizes the function w.r.t. γ and maximizes it w.r.t. $\hat{\mathbf{F}}, \boldsymbol{F}^*, \hat{\boldsymbol{F}}_p$ and $\boldsymbol{T}_p$. Therefore, the final objective function can be expressed as follows,

$$\max_{\hat{\mathbf{F}}, \hat{\boldsymbol{F}}_p, \boldsymbol{T}_p} \min_{\gamma} \max_{\boldsymbol{F}^*} \mathrm{Tr}(\boldsymbol{F}^{*\top}(\sum_{p=1}^m \gamma_p^2 \hat{\boldsymbol{F}}_p \boldsymbol{T}_p)) + \lambda \sum_{p=1}^m \gamma_p^2 \mathrm{Tr}(\hat{\boldsymbol{F}}_p^\top \boldsymbol{F}_p) - \rho ||\hat{\mathbf{F}}||_{\circledast},$$
$$s.t.\ \hat{\boldsymbol{F}}_p^\top \hat{\boldsymbol{F}}_p = \boldsymbol{I}, \boldsymbol{T}_p^\top \boldsymbol{T}_p = \boldsymbol{I}, \forall p, \gamma \in \Delta, \boldsymbol{F}^{*\top} \boldsymbol{F}^* = \boldsymbol{I}. \tag{3}$$

Optimization of DLEFT-MKC

To solve the resultant max-min-max optimization problem of DLEFT-MKC in Eq.(3), we combine the optimization strategies of the reduced gradient descent method (RGDM) and Alternating Direction Method of Multipliers (ADMM), updating one specific variable while keeping others fixed. To facilitate the divisibility of $\mathbf{F}$, we introduce an auxiliary tensor variable $\mathbf{A}$ according to the principles of ADMM.

Detail optimization is given in the paper.

The Proposed Algorithm Pseudo Code

Algorithm 1 Min-Max Optimization For γ and $\boldsymbol{F}^*$

Input: $\boldsymbol{F}^*, \{\hat{\boldsymbol{F}}_p, \boldsymbol{F}_p, \boldsymbol{T}_p\}_{p=1}^m, \gamma, k, \lambda$.
Ouput: Weight coefficients γ and consensus clustering partition $\boldsymbol{F}^*$.

while not converge **do**
  calculate $\mathbf{H}^{(t)}$ via a kernel k-means with $\boldsymbol{K}_{\gamma^{(t)}}$.
  Calculate the reduced gradient $[\triangledown\mathcal{G}(\gamma)]_p$ via Eq.(13).
  Calculate the descent direction $\boldsymbol{V}$ in Eq.(14).
  Update weight coefficients $\gamma \leftarrow \gamma + \alpha\boldsymbol{V}$ with the step size α.
  if $\max|\gamma - \gamma_{old}| \leq 10^{-4}$ **then**
    Converge.
  end if
end while

Algorithm 2 DLEFT-MKC

Input: Base partition matrices $\{\boldsymbol{F}_p\}_{p=1}^m$, the number of clusters k, trade-off parameters λ and ρ.
Output: Consensus clustering partition $\boldsymbol{F}^*$.

Initialize $\hat{\mathbf{F}} = \Phi(\boldsymbol{F}_1, \ldots, \boldsymbol{F}_m), \hat{\boldsymbol{F}}_p = \boldsymbol{F}_p, \boldsymbol{T}_p = \mathbf{I}, \gamma_p = \frac{1}{m}, \forall p, \mathbf{A} = \hat{\mathbf{F}}, \mathbf{Y} = \mathbf{0}, \mu = 0.1, \tau = 2$.
Calculate $\boldsymbol{F}^* = \arg\max_{\boldsymbol{F}^{*\top}\boldsymbol{F}^*=\boldsymbol{I}} \operatorname{Tr}\left(\boldsymbol{F}^*\left(\sum_{p=1}^m \gamma_p^2 \hat{\boldsymbol{F}}_p \boldsymbol{T}_p\right)\right)$.
while not converge **do**
  Update reconstructed partitions $\{\hat{\boldsymbol{F}}_p\}_{p=1}^m$ via Eq.(10).
  Update weight coefficients γ and consensus parition $\boldsymbol{F}^*$ by solving Algorithm 1.
  Update permutation matrices $\{\boldsymbol{T}_p\}_{p=1}^m$ by solving Eq.(16).
  Update auxiliary tensor $\mathbf{A}$ by solving Eq.(15).
  Update the Lagrange multiplier $\mathbf{Y}$ and the penalty factor μ via Eq.(17).
end while

Content

Comprehensive Performance

Algorithms	Liver	BBCSport	ProteinFold	Willow	Plant	PsortNeg	Scene15	CCV	Flower102	Reuters
Avg-KKM	54.2±0.0	63.2±1.4	29.0±1.5	22.2±0.3	61.3±0.9	41.0±1.4	43.2±1.8	19.6±0.6	27.1±0.8	45.5±1.5
SB-KKM	57.9±0.1	71.4±0.1	33.8±1.3	26.8±0.3	51.2±1.1	55.3±0.0	39.3±0.2	20.1±0.2	33.0±1.0	47.2±0.0
MKKM	55.0±0.3	63.0±1.5	27.0±1.1	22.0±0.2	56.1±0.6	51.9±0.3	41.2±0.1	18.0±0.5	22.4±0.5	45.4±1.5
LMKKM	53.7±1.1	63.9±1.4	22.4±0.7	22.6±0.2	-	-	40.9±0.1	18.6±0.1	-	-
ONKC	52.9±1.9	63.4±1.4	36.3±1.5	22.6±0.4	41.4±0.2	40.2±0.6	39.9±1.4	22.4±0.3	39.5±0.7	41.8±1.2
MKKM-MR	51.3±0.0	63.2±1.5	34.7±1.8	22.9±0.4	50.3±0.8	39.7±0.5	38.4±1.1	21.2±0.9	40.2±0.9	46.2±1.4
LKAM	60.0±0.0	73.9±0.5	37.7±1.2	27.1±0.1	47.6±0.0	40.5±0.4	41.4±0.5	20.4±0.3	41.4±0.8	45.5±0.0
LFMVC	54.5±0.0	76.4±2.9	33.0±1.4	26.4±0.5	59.5±0.6	45.5±0.3	45.8±1.0	25.1±0.5	38.4±1.2	45.7±1.6
NKSS	55.9±0.0	64.1±1.2	36.4±0.7	25.5±0.6	39.2±0.1	48.2±1.0	40.4±0.3	20.0±0.2	41.7±0.8	37.7±1.4
SPMKC	54.5±0.0	51.3±1.9	17.8±0.5	26.3±0.2	51.4±0.1	25.0±0.6	38.0±0.1	16.2±0.2	25.6±0.4	26.8±0.0
HMKC	55.4±0.0	91.1±3.7	35.3±1.5	32.7±0.5	64.2±0.1	49.1±0.0	50.5±0.1	32.8±0.5	47.7±1.3	46.8±0.3
SMKKM	53.9±0.0	64.2±1.6	34.7±1.9	22.4±0.4	49.5±0.5	41.5±0.0	43.6±1.0	22.2±0.7	42.5±0.8	45.5±0.7
OPLFMVC	54.6±0.1	89.2±3.2	31.1±2.6	27.3±1.0	47.3±3.1	46.1±2.3	43.9±1.8	23.7±0.9	30.4±1.0	43.9±1.0
LSMKKM	58.3±0.0	73.4±1.0	36.3±1.5	24.8±0.2	57.1±0.8	45.7±0.1	44.5±1.6	21.5±0.9	43.8±1.0	47.1±1.0
AIMC	52.8±0.0	70.4±0.0	33.6±0.0	25.5±0.0	47.9±0.0	45.4±0.0	44.5±0.0	24.5±0.0	41.0±0.0	43.2±0.0
OMSC	53.0±0.0	89.0±0.0	31.8±0.0	28.1±0.0	56.5±0.0	39.5±0.0	41.7±0.0	25.1±0.0	38.9±0.0	42.4±0.0
HFLSMKKM	57.4±0.0	51.6±1.3	33.8±1.1	24.2±0.5	43.6±0.1	31.3±0.6	41.7±0.4	18.5±0.3	35.8±0.8	37.5±0.8
GMC	51.0±0.2	88.2±0.0	29.3±0.0	21.2±0.5	39.4±0.0	25.2±0.0	26.9±0.6	16.8±0.4	34.1±0.0	-
LTBPL	58.3±0.0	96.5±0.0	32.1±1.1	28.8±0.0	48.2±0.0	29.1±0.0	40.1±0.7	-	-	-
UGLTL	53.6±0.0	99.1±0.2	51.1±1.7	37.1±2.0	68.6±1.2	92.2±0.0	94.4±5.1	43.7±1.3	65.8±2.3	-
WTNNM	53.3±0.0	95.2±0.0	43.2±1.7	32.0±0.2	68.0±0.1	64.8±0.0	76.1±1.2	47.7±0.0	61.7±0.9	-
KCGT	54.8±0.2	74.4±1.2	33.4±1.2	26.1±0.4	52.4±0.6	44.9±0.4	45.5±0.9	23.9±0.5	39.5±0.8	43.0±0.8
DLEFT-MKC	**86.4±0.0**	**99.2±0.1**	**66.5±2.9**	**84.9±0.4**	**94.1±0.1**	**96.0±0.0**	**96.2±0.1**	**81.5±2.7**	**79.9±2.2**	**97.0±4.0**

Table 1: Empirical comparison of the proposed DLEFT-MKC with dozens of recent MKC algorithms on ten benchmark datasets in terms of ACC.

Convergence and Evolution

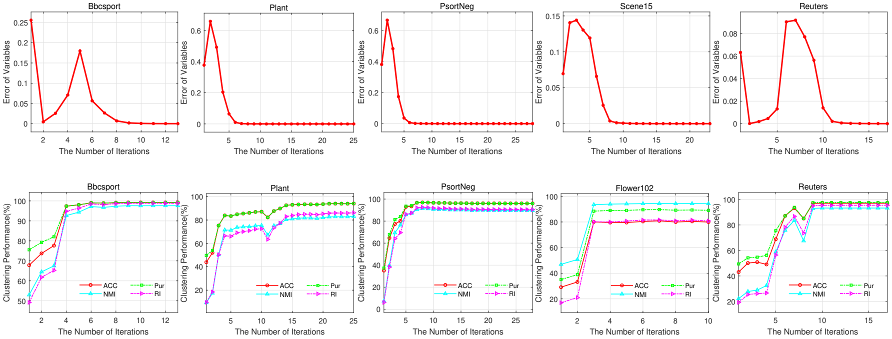

Figure 2: The evolution of error values and clustering performance during the clustering learning process of our proposed DLEFT-MKC across iterations.

Cluster Partitions Analysis

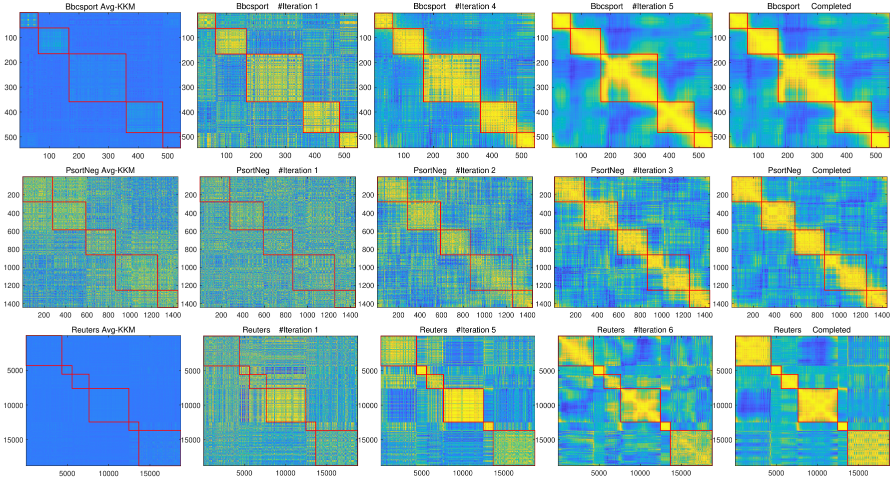

Figure 3: The leftmost figure denotes the clustering partition learned by avg-KKM. Four right figures represent the clustering partitions of DLEFT-MKC during the learning process.

Parameter Sensitivity Analysis

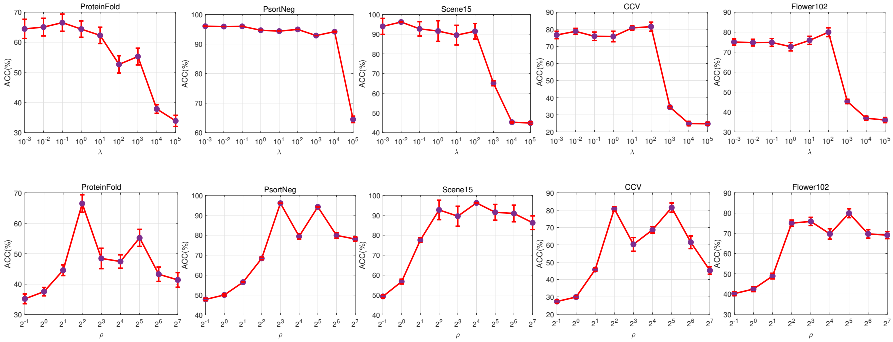

Figure 4: The effect on clustering performance with varying parameter λ (1^{st} line) and ρ (2^{nd} line) of the proposed DLEFT-MKC.

Ablation Study

Algorithms	Liver	BBCSport	ProteinFold	Willow	Plant	PsortNeg	Scene15	CCV	Flower102	Reuters
$\mathcal{L}_1$	54.1 $\pm$ 0.2	63.4 $\pm$ 1.3	30.0 $\pm$ 2.1	22.2 $\pm$ 0.3	56.0 $\pm$ 0.5	38.5 $\pm$ 0.6	43.8 $\pm$ 1.6	19.6 $\pm$ 0.6	27.2 $\pm$ 0.9	45.1 $\pm$ 0.3
$\mathcal{L}_2$	62.3 $\pm$ 0.0	84.8 $\pm$ 8.7	**66.7** $\pm$ 2.9	71.2 $\pm$ 0.5	93.7 $\pm$ 0.0	95.8 $\pm$ 0.0	93.8 $\pm$ 4.3	77.2 $\pm$ 2.5	74.6 $\pm$ 2.1	96.9 $\pm$ 0.0
$\mathcal{L}_3$	51.0 $\pm$ 0.0	43.1 $\pm$ 0.7	13.7 $\pm$ 0.7	19.5 $\pm$ 0.6	31.9 $\pm$ 1.0	44.4 $\pm$ 3.2	54.6 $\pm$ 2.7	25.5 $\pm$ 1.1	60.6 $\pm$ 1.5	51.7 $\pm$ 0.0
$\mathcal{L}_4$	82.3 $\pm$ 0.0	86.8 $\pm$ 0.3	44.6 $\pm$ 2.4	79.2 $\pm$ 2.5	72.7 $\pm$ 0.1	89.6 $\pm$ 0.0	86.3 $\pm$ 5.2	73.0 $\pm$ 1.7	80.4 $\pm$ 1.8	87.1 $\pm$ 2.2
$\mathcal{L}_5$	60.6 $\pm$ 0.0	96.6 $\pm$ 0.1	56.1 $\pm$ 2.3	76.7 $\pm$ 4.3	91.0 $\pm$ 0.0	94.6 $\pm$ 0.0	95.1 $\pm$ 2.9	72.4 $\pm$ 1.6	**81.7** $\pm$ 2.7	94.8 $\pm$ 1.7
Proposed	**86.4** $\pm$ 0.0	**99.2** $\pm$ 0.1	66.5 $\pm$ 2.9	**84.9** $\pm$ 0.4	**94.1** $\pm$ 0.1	**96.0** $\pm$ 0.0	**96.2** $\pm$ 0.1	**81.5** $\pm$ 2.7	79.9 $\pm$ 2.2	**97.0** $\pm$ 4.0

Table 2: Ablation study of the proposed DLEFT-MKC. The best result are highlighted in bold.

Running Time Comparison

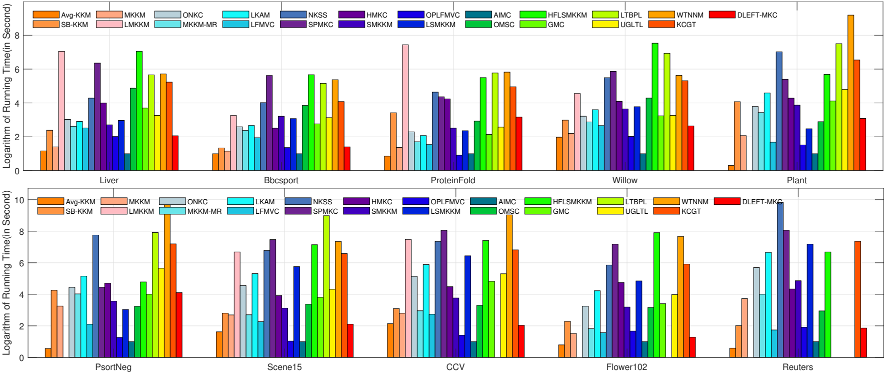

Figure 5: Time complexity comparison of all algorithms on benchmark datasets. For better clarity, we scaled the values and adopted logarithmic values in second.

Performance with noisy data

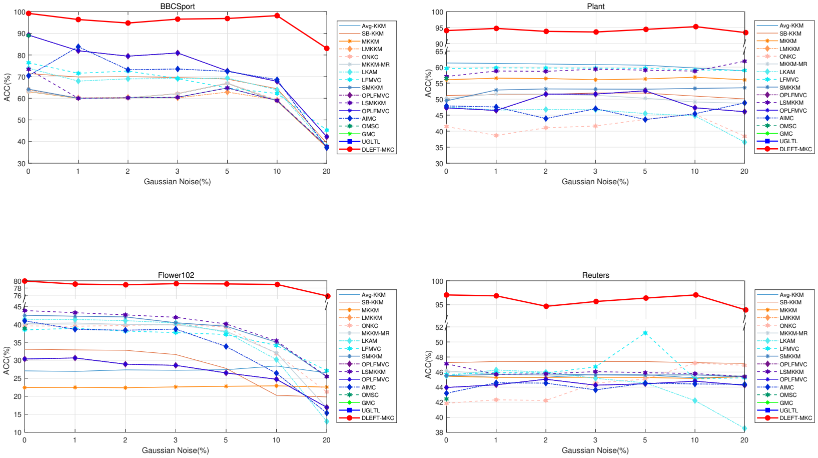

Figure 6: Comparison of comprehensive clustering performance fluctuations between DLEFT-MKC algorithm and comparative algorithm under different degrees of Gaussian noise (0, 1, 2, 3, 5, 10, 20%).

Content

Conclusion

This paper introduces a novel Multiple Kernel clustering framework known as **D**ynamic **L**at**E**-**F**usion Multiple Kernel Clustering with Robust **T**ensor Learning (DLEFT-MKC) via min-max optimization, which is simple yet effective and efficient. Specifically, For the first time, DLEFT-MKC integrates a min-max optimization paradigm into tensor-based MKC, enhancing both performance and robustness; the framework dynamically reconstructs base partitions from LFMVC, effectively overcoming their representational bottleneck. Additionally, tensor learning is employed to capture the high-order correlations and uncover latent structures across views. To solve the resultant optimization problem, we design an innovative and efficient strategy to combine the RGDM with ADMM. Experimental results demonstrate that our proposed DLEFT-MKC significantly outperforms other state-of-the-art MKC algorithms in terms of clustering performance and computation efficiency across benchmark datasets.

Thanks for Your Listening!