Robust Contrastive Multi-view Kernel Clustering

Peng Su[1,2] , **Yixi Liu**[1,2] , **Shujian Li**[1,2] , **Shudong Huang**[1,2*] , **Jiancheng Lv**[1,2]

[1]College of Computer Science, Sichuan University, Chengdu 610065, China
[2]Engineering Research Center of Machine Learning and Industry Intelligence, Ministry of Education, Chengdu 610065, China
{supeng, liuyixi}@stu.scu.edu.cn, shujianli96@163.com, {huangsd, lvjiancheng}@scu.edu.cn

Abstract

Multi-view kernel clustering (MKC) aims to fully reveal the consistency and complementarity of multiple views in a potential Hilbert space, thereby enhancing clustering performance. The clustering results of most MKC methods are highly sensitive to the quality of the constructed kernels, as traditional methods independently compute kernel matrices for each view without fully considering complementary information across views. In previous contrastive multi-view kernel learning, the goal was to bring cross-view instances of the same sample closer during the kernel construction process while pushing apart instances across samples to achieve a comprehensive integration of cross-view information. However, its inherent drawback is the potential inappropriate amplification of distances between different instances of the same clusters (i.e., false negative pairs) during the training process, leading to a reduction in inter-class discriminability. To address this challenge, we propose a Robust Contrastive multi-view kernel Learning approach (R-CMK) against false negative pairs. It partitions negative pairs into different intervals based on distance or similarity, and for false negative pairs, reverses their optimization gradient. This effectively avoids further amplification of distances for false negative pairs while simultaneously pushing true negative pairs farther apart. We conducted comprehensive experiments on various MKC methods to validate the effectiveness of the proposed method. The code is available at https://github.com/Duolaimi/rcmk_main.

1 Introduction

In data mining and machine learning, clustering has been a major problem for decades, one to which kernel technique has recently been prevalent in numerous real-world applications [Tang *et al.*, 2023; Huang *et al.*, 2019]. Kernel clustering utilizes nonlinear function to implicitly map the points to a higher-dimensional feature space and partition them into clusters with similar objects. Prevailing kernel clustering algorithms have been applied with significant success, including kernel fuzzy c-means [Gong *et al.*, 2012], kernel k-means [Dhillon *et al.*, 2004] and kernel spectral clustering [Cristianini *et al.*, 2001]. In [Xiao *et al.*, 2015], the kernelized version of LRR (RKLRR) is given to handle the corrupted data, which can explore nonlinear subspaces. [Xie *et al.*, 2017] uses smooth learning and proximal gradient-based optimization to kernelize the block diagonal representation.

The key point of above-mentioned methods is single kernel clustering (SKC). Since the kernel methods heavy reliance on the advance selection and a single pre-defined kernel, the clustering performance is unsatisfactory [Kang *et al.*, 2018]. More lethal can be their incapability of handling expansive multi-view data. Multi-view kernel clustering (MKC) is designed to deal with these problems by optimally combining a group of parameterized mapping functions to obtain excellent clustering. To distinguish the importance of inner clusters, a cluster-weighted kernel k-means algorithm (CWK^{2}M) is designed to assign specific weight for each inner cluster and then optimizes weighted sum-of-squared process by kernel k-means strategy [Liu *et al.*, 2020a]. Considering that distilling the information from kernels would detracts from retaining effective information, a hierarchical approach is provided to preserve advantageous details maximumly with a cyclic process including generating a sequence of intermediary matrices and distilling the partition information from kernel matrices [Liu *et al.*, 2021]. In the work of [Ren *et al.*, 2020], MKGC method directly learns a consensus similarity matrix and preserves the local manifold structure of the samples in kernel space in order to tackle problems of computational cost and clustering performance. [Liu *et al.*, 2020b] makes use of adaptive local kernels to effectively measure the local density around each sample and possess the capability to learn the representation of the optimal kernel. Although the above MKC methods have demonstrated excellent performance in various applications, they cannot guarantee the quality of core construction, as they construct core matrices independently for each view and neglect the correlations among multiple views and fail to overcome the scaling issues caused by inconsistent feature dimensions across multiple views.

Contrastive learning aims to extract information from multiple perspectives by emphasizing high similarity among samples sharing similar features within the dataset. The key

[*] Corresponding author

idea is to generate pair construction, where samples augmented from the identical instance are categorized as positive and the others are defined as negative [He *et al.*, 2020; Chen *et al.*, 2020]. Inspired by contrastive learning, the Contrastive Multi-view Kernel (CMK) is proposed to control kernel quality by measuring inner information from each view [Liu *et al.*, 2023].

However, employing such a pair construction instances may erroneously classify certain within-category instances as negatives, leading to the generation of false negative pairs (FNPs) [Yang *et al.*, 2023]. To address this problem, we proposed a novel multi-view learning method named Robust Contrastive Multi-view Kernel Learning (R-CMK) to tackle the gap. The proposed method can be seamlessly integrated into established multi-view kernel clustering frameworks, enabling the acquisition of multiple clusterings with robust kernels or features. In detail, the contributions are as follows:

- We propose a novel robust contrastive multi-view learning method named R-CMK. R-CMK incorporates a noise-robust contrastive loss, which helps alleviate or eliminate the impact of false-negative pairs (FNPs) introduced during the pair construction process.
- The proposed method can be applied to existing multi-view kernel learning approaches as a plug-and-play component, effectively enhancing the quality of the kernels.
- Experiments on several datasets with six recent multiple kernel clustering methods show that R-CMK can effectively improve the quality of the clusterings.

2 Related Work

In this section, we provide a concise overview of the most pertinent literature, encompassing both multiple kernel k-means (MKKM) and contrastive multi-view kernel learning. We use the term "sample" to refer to a data point that encompasses all views, while "instance" represents a sample under a specific view.

2.1 Multiple Kernel K-Means

MKKM is a typical multi-view learning algorithm. In the context of multi-view learning, each view corresponds to a kernel. MKKM posits that the optimal kernel matrix is a linear combination of individual single-view kernels, shown as

$$\mathbf{K}_\theta = \sum_{p=1}^{V} \theta_p^q \mathbf{K}_p, \tag{1}$$

where $\{\mathbf{K}_p\}_{p=1}^V$ represents pre-computed kernel matrices for individual views, considered as a set of base matrices. θ_p denotes the weight of the p-th base matrix, and q is a smoothing factor used to control the smoothness of the weight coefficients, typically set to 2 to avoid trivial solutions. MKKM addresses the simultaneous learning of the weight coefficients $\boldsymbol{\theta}$ for each base matrix and the clustering partition matrix $\mathbf{H}$ for the samples by solving the following optimization problem

$$\min_{\boldsymbol{\theta}\in\Theta} \min_{\mathbf{H}\in\Phi} Tr\left(\mathbf{K}_{\boldsymbol{\theta}}\left(\mathbf{I}-\mathbf{H}\mathbf{H}^\top\right)\right), \tag{2}$$

where $\Theta = \{\boldsymbol{\theta} \in \mathbb{R}^V \,|\, \sum_{p=1}^V \theta_p = 1, \theta_p \geq 0, \forall p\}$ and $\Phi = \{\mathbf{H} \in \mathbb{R}^{n\times k} | \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k\}$.

MKKM and its variants usually alternate between optimizing $\boldsymbol{\theta}$ and $\mathbf{H}$ to solve Eq. (2). In the initial step, $\mathbf{H}$ is optimized with $\boldsymbol{\theta}$ held fixed. For a given set of weights $\boldsymbol{\theta}$, optimizing Eq. (2) with respect to $\mathbf{H}$ is equivalent to solving the following problem

$$\max_{\mathbf{H}\in\Phi} Tr\left(\mathbf{H}^\top \mathbf{K}_{\boldsymbol{\theta}} \mathbf{H}\right). \tag{3}$$

Eq. (3) characterizes the classical objective of kernel k-means clustering. Its optimal solution involves the eigenvectors corresponding to top k largest eigenvalues of $\mathbf{K}_{\boldsymbol{\theta}}$, and this can be computed using available algorithmic packages. In the second step, $\boldsymbol{\theta}$ is optimized with $\mathbf{H}$ held fixed. For a specific $\mathbf{H}$, the optimization in Eq. (2) with respect to $\boldsymbol{\theta}$ reduces to the following form

$$\min_{\boldsymbol{\theta}\in\Theta} \sum_{p=1}^{V} = \theta_p^2 Tr\left(\mathbf{K}_p\left(\mathbf{I}-\mathbf{H}\mathbf{H}^\top\right)\right), \tag{4}$$

which possesses a theoretically closed-form solution.

2.2 Contrastive Multi-view Kernel Learning

Contrastive Multi-view Kernel (CMK) [Liu *et al.*, 2023] learning introduces a novel kernel generation paradigm based on contrastive learning. It aims to learn a set of projection coefficients for diverse views, followed by utilizing a kernel function to map the projected features into the corresponding Hilbert space. In this space, the objective is to maximize the similarity between positive sample pairs and minimize the similarity between negative sample pairs. Positive pairs are constructed by instances across views of the same sample, while negative sample pairs are formed by instances across different samples.

Specifically, given $\{x_i\}_{i=1}^N \subseteq \mathcal{X}$ as a set of N samples in a single view, and $\left\{\{x_i^p\}_{i=1}^N\right\}_{p=1}^V \subseteq \{\mathcal{X}^p\}_{p=1}^V$ as V views' data where $x_i^p \in \mathbb{R}^{d_p}$. CMK first projects the multi-view data into a unified latent space $\mathcal{X}_h \subseteq \mathbb{R}^{d_l}$ to eliminate the dimension difference via

$$h_i^p = f_{\mathbf{W}^p}\left(x_i^p\right) = x_i^p \mathbf{W}^p, \tag{5}$$

in which $\mathbf{W}^p \in \mathbb{R}^{d_p \times p}$. Then CMK normalizes the data representations $\mathbf{h}_i^p$ through dividing by the L_2-norm

$$\mathbf{z}_i^p = f_N\left(\mathbf{h}_i^p\right) = \mathbf{h}_i^p / L_2(\mathbf{h}_i^p). \tag{6}$$

Selecting a known kernel mapping, such as Gaussian mapping, CMK projects the representations $\{\mathbf{z}_i^p\}_{i,p=1}^{N,V}$ into the corresponding Hilbert space $\mathcal{H}$, yielding the following kernel function.

$$\begin{aligned} k_x^p\left(\mathbf{x}_i^p, \mathbf{x}_j^{p'}\right) &= k_z\left(\mathbf{z}_i^p, \mathbf{z}_j^{p'}\right) \\ &= k_z\left(f_N\left(f_{\mathbf{W}^p}\left(x_i^p\right)\right), f_N\left(f_{\mathbf{W}^p}\left(x_j^p\right)\right)\right). \end{aligned} \tag{7}$$

Unlike conventional contrastive learning, CMK treats multi-view data as a natural augmentation of samples. It utilizes

Eq. (7) to compute the similarity of sample pairs, where the loss for the i-th instance on the p-th view is denoted as

$$\ell_{i,p} = \frac{1}{V-1} \sum_{p'=1, p' \neq p}^{V} -\log \frac{\exp(k_z(\mathbf{z}_i^p, \mathbf{z}_i^{p'}))}{\sum_{j,p'' \in \mathcal{A}_{i,p}} \exp(k_z(\mathbf{z}_i^p, \mathbf{z}_j^{p''}))}, \tag{8}$$

where

$$\mathcal{A}_{i,p} = \{1, 2, \cdots, N\} \times \{1, 2, \cdots, V\} \setminus \{(i,p)\}. \tag{9}$$

By optimizing the overall loss function that includes both positive and negative pairs, CMK can effectively learn cross-view consistency, leading to improved performance in downstream clustering algorithms.

3 The Proposed Method

In this section, we introduce the proposed method R-CMK in detail, then develop a gradient descent algorithm to optimize the objective effectively. Finally, we discuss its computational complexity.

3.1 Robust Contrastive Multi-view Kernel

CMK proposed by [Liu *et al.*, 2023] leverage complementary information from the data views to compute quality kernels based on the paradigm of contrastive learning, which uses the cross-view instances of same sample as positive pairs and the cross-sample instances in different views as negative pairs. This would inevitably introduce noisy labels, because there may exist some negative pairs belongs to same cluster. These pairs are called as false negative pairs (FNPs). To address the aformentioned issue, we correspondingly propose to integrate robust contrastive learning into the CMK framework.

Similar to CMK, we also need to choose a kernel function to project the acquired representations into an implicit Hilbert space $\mathcal{H}$. For ease of discussion, we take Gaussian kernel function as an illustration in the following

$$k_z(\mathbf{z}_i^p, \mathbf{z}_j^p) = e^{-\gamma \|\mathbf{z}_i^p - \mathbf{z}_j^p\|_2^2}. \tag{10}$$

R-CMK defines the robust contrastive loss of the i-th sample in the v-the view as (11), then compute the average loss of all instances

$$\ell_{i,p} = \frac{1}{NV} \left(\sum_{p'=1, p' \neq p}^{V} \ell^{pos} + \sum_{j=1, j \neq i}^{N} \sum_{p'=1}^{V} \ell^{neg} \right). \tag{11}$$

Denoting the distance between z_i^p and $z_j^{p'}$ as

$$d(\mathbf{z}_i^p, \mathbf{z}_i^{p'}) = e^{-k_z\left(\mathbf{z}_i^p, \mathbf{z}_i^{p'}\right)}, \tag{12}$$

then

$$\ell^{pos}\left(\mathbf{z}_i^p, \mathbf{z}_i^{p'}\right) = d^2(\mathbf{z}_i^p, \mathbf{z}_i^{p'}). \tag{13}$$

We measure the similarity or distance of sample pairs with kernel function and directly minimize the distance of positive pairs. Due to the presence of FNPs, we do not simply maximize the distance between negative sample pairs. Inspired by [Yang *et al.*, 2023], we define a noise-robust negative pairs loss as

$$\begin{aligned} &\ell^{neg}\left(z_i^p, z_j^{p'}\right) \\ &= \frac{1}{m} \max\left(\left(m - d\left(z_i^p, z_j^{p'}\right)\right) d^{\frac{1}{2}}\left(z_i^p, z_j^{p'}\right), 0\right)^2, \end{aligned} \tag{14}$$

where m is a pre-defined parameter and will be discussed later. Considering instances across all views, the overall loss takes the following form

$$\ell_c = \frac{1}{NV} \sum_{i,p=1}^{N,V} \ell_{i,p}. \tag{15}$$

By minimizing the loss function, the distance between positive pairs can be effectively reduced as well as false negative pairs. The distances between true negative sample pairs gradually widen. For negative sample pairs that fall in between, the model tends to maintain their original distances or reduce the optimization gradient.

3.2 Gradient Computing and Optimization

In this section, we employ the gradient descent algorithm to optimize the objective function, and utilize the chain rule to calculate the gradients of variables $\{\mathbf{W}^p\}_{p=1}^V$. For the positive pair loss, applying the chain rule yields

$$\frac{\partial \ell^{pos}\left(\mathbf{z}_i^p, \mathbf{z}_i^{p'}\right)}{\partial \mathbf{W}^p} = \frac{\partial \ell^{pos}}{\partial d} \cdot \frac{\partial d}{\partial \mathbf{z}_i^p} \cdot \frac{\partial \mathbf{z}_i^p}{\partial \mathbf{h}_i^p} \cdot \frac{\partial \mathbf{h}_i^p}{\partial \mathbf{W}^p}. \tag{16}$$

Eq. (16) is derived from the product of four components, where the first, second, and fourth components only involve the composite partial derivatives of basic elementary functions. We focus on deriving the third part. Denoting $z_{j'}$ and $h_{i'}$ as the j'-th and i'-th element of $\mathbf{z}_i^p$ and $\mathbf{h}_i^p$ respectively, we can derive the following expression

$$\frac{\partial \mathbf{z}_i^p}{\partial \mathbf{h}_i^p} = \left[\sum_{j'=1}^{d_l} \frac{\partial z_{j'}}{\partial h_1}, \cdots \sum_{j'=1}^{d_l} \frac{\partial z_{j'}}{\partial h_{i'}}, \cdots, \sum_{j'=1}^{d_l} \frac{\partial z_{j'}}{\partial h_{d_l}} \right], \tag{17}$$

in which

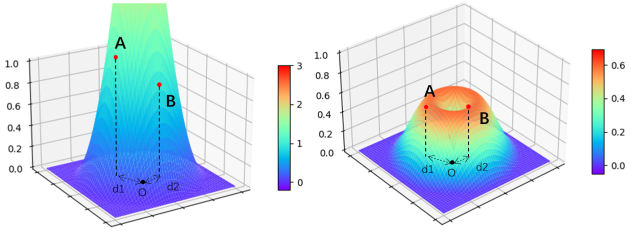
$$\frac{\partial z_{j'}}{\partial h_{i'}} = \mathbb{1}_{i'=j'} (\sum_{k=1}^{d_l} h_k^2)^{-1/2} + h_{i'} h_{j'} (\sum_{k=1}^{d_l} h_k^2)^{-3/2}. \tag{18}$$

For the negative pair loss, its partial derivative form is similar to Eq. (16), shown as

$$\begin{aligned} \frac{\partial \ell^{neg}\left(\mathbf{z}_i^p, \mathbf{z}_j^{p'}\right)}{\partial \mathbf{W}^p} &= \frac{\partial \ell^{neg}}{\partial d} \cdot \frac{\partial d}{\partial \mathbf{z}_i^p} \cdot \frac{\partial \mathbf{z}_i^p}{\partial \mathbf{h}_i^p} \cdot \frac{\partial \mathbf{h}_i^p}{\partial \mathbf{W}^p} \\ &+ \mathbb{1}_{p'=p} \left(\frac{\partial \ell^{neg}}{\partial d} \cdot \frac{\partial d}{\partial \mathbf{z}_i^{p'}} \cdot \frac{\partial \mathbf{z}_i^{p'}}{\partial \mathbf{h}_i^{p'}} \cdot \frac{\partial \mathbf{h}_i^{p'}}{\partial \mathbf{W}^p} \right). \end{aligned} \tag{19}$$

The gradient of ℓ^{neg} *w.r.t* d is nonmonotonic. When $d > m$, Eq. (14) produces no gradient. When $d \leq m$, the gradient of ℓ^{neg} *w.r.t* d is

$$\begin{aligned} \frac{\partial \ell^{neg}}{\partial d} &= \frac{\left(\partial \frac{1}{m} d^3 - 2d^2 + md\right)}{\partial d} \\ &= \frac{3}{m}\left(d - \frac{m}{3}\right)(d - m). \end{aligned} \tag{20}$$



(a) (b)

Figure 1: The loss surface of the vanilla term and the robust term

As shown in Figure 1, the loss surface defined by Eq. (14) can be divided into two regions: the hole area $(0 < d < m/3)$ and the deceleration area $(m/3 < d < m)$. In the hole area, the gradient direction of the robust term is opposite to that of the vanilla term, whereas in the deceleration area, the optimization gradient is slowed down. Based on equations Eq. (11) and Eq. (15), we can compute the gradient of ℓ_c with respect to $\mathbf{W}^p$, shown as

$$\frac{\partial \ell_c}{\partial \mathbf{W}^p} = \frac{1}{N^2V^2} \sum_{i,p=1}^{N,V} \left(\sum \frac{\partial \ell^{pos}}{\partial \mathbf{W}^p} + \sum \frac{\partial \ell^{neg}}{\partial \mathbf{W}^p} \right). \tag{21}$$

Denoting the learning rate as α, we proceed to update the weights using the following formula

$$\mathbf{W}^p = \mathbf{W}^p - \alpha \frac{\partial \ell_c}{\partial \mathbf{W}^p}. \tag{22}$$

Iterative optimization continues until the loss function converges or reaches the specified maximum number of iterations.

3.3 Complexity

In this subsection, we analyze the computational complexity of the proposed method. In one iteration of the algorithm, two main operations are performed. Firstly, each view's features are multiplied by their respective projection matrix and L2-normalized to standardize the feature scale. Secondly, the gradient of the loss function is calculated, and the projection coefficient matrix is updated according to Eq. (22). For simplicity, let's assume that the original feature dimensions for each view are $\overline{d_p}$. In practice, this can be substituted with the average dimension across views.

In the first step, the mathematical operation for feature projection involves matrix multiplication between the feature matrix and the projection matrix. The time complexity of this process is $O(NV\overline{d_p}d_l)$. Normalization involves dividing the projected features by their corresponding L2 norms, with a time complexity of $O(NVd_l)$. Therefore, the overall time complexity of the first step is $O(NV\overline{d_p}d_l + NVd_l)$ or $O(NV\overline{d_p}d_l)$.

In the second step, there are a total of N^2V^2 positive and negative sample pairs. Each sample pair contributes to a loss, and whether it's based on Eq. (16) or Eq. (19), each loss only affects the gradient of the projection coefficients for the corresponding view. The time complexity of gradient calculation is $O(\overline{d_p}d_l)$, disregarding constant factors. Hence, the overall time complexity of the second step is $O(N^2V^2\overline{d_p}d_l)$.

The proposed method has an overall time complexity of $O(N^2V^2\overline{d_p}d_l)$ and a space complexity of $O(N^2V^2)$, which could result in significant memory consumption, making it less suitable for large-scale data. Therefore, running in batches is an inevitable choice. Specifically, for a random batch of multi-view data, the corresponding loss can be computed as

$$\ell_{i,p} = \frac{1}{N_bV} \left(\sum_{p'=1,p'\neq p}^{V} \ell^{pos} + \sum_{j=1,j\neq i}^{N_b} \sum_{p'=1}^{V} \ell^{neg} \right). \tag{23}$$

At this point, the space complexity for each batch operation is $O(N_b^2V^2)$, significantly reducing memory requirements. Additionally, gradient-based optimization algorithms such as Adam [Kingma and Ba, 2017] can be employed during training, and GPU acceleration can be utilized to shorten the training time.

4 Experiments

In this section, we conduct extensive experiments to analyze the quality of kernels established by traditional method, CMK and R-CMK. Then we apply these kernels to downstream tasks and analyze the clustering performance under different kernel-based methods.

4.1 Datasets

To demonstrate the effectiveness of the proposed R-CMK generation paradigm, we conducted comprehensive experiments using several commonly-used datasets:

- **3 Sources** is a multi-view text dataset collected from three news sources, containing 169 samples distributed across 6 classes.
- **Cora** [Bisson and Grimal, 2012] is composed of 4 views, including content, inbound, outbound, and cites, of the documents, containing 2708 samples categorized into 7 labels.
- **Handwritten** [Nie *et al.*, 2017] is a digit dataset comprising a total of two thousand samples distributed across 10 classes, with each class containing 200 samples.
- **Yale** is a classical face database consisting of 165 grayscale images of 15 individuals. Each subject has 11 images collected under different facial expressions and configurations. There are three types of features representing three views.
- **BBCSport** [Greene and Cunningham, 2006] comprises 737 documents from the 2004-2005 BBC Sport website, covering five sports categories: athletics, cricket, football, rugby, and tennis.
- **Movies617** [Bisson and Grimal, 2012] includes two matrices for clustering tasks. With 617 movies across 17 genres, it combines information from 1878 keywords and 1398 actors to determine genres. Each movie involves at least 2 actors and 2 keywords, ensuring robust data for reliable genre classification.

Datasets	AVG-KKM			MKKM			MKKM-MIR			ONKC			LF-MVC			SimpleMKKM		
	Trad.	CMK	Ours	Trad.	CMK	Ours	Trad.	CMK	Ours	Trad.	CMK	Ours	Trad.	CMK	Ours	Trad.	CMK	Ours
ACC(%)																		
3Sources	53.85	**78.11**	76.92	54.44	**77.51**	**77.51**	51.48	70.41	**71.01**	53.25	71.01	**72.19**	50.89	**72.19**	71.01	51.78	69.11	**69.20**
Cora	33.90	59.56	**64.88**	38.29	59.34	**64.88**	37.16	59.27	**64.36**	38.59	59.16	**64.25**	38.22	57.75	**63.96**	38.42	59.20	**64.24**
Handwritten	75.20	91.65	**93.80**	79.60	91.25	**93.30**	76.25	91.95	**94.20**	77.85	92.00	**94.35**	77.80	91.90	**94.30**	77.15	91.95	**94.33**
Yale	61.21	55.15	**63.63**	**64.85**	58.18	**64.85**	58.79	**61.21**	59.39	58.79	**60.61**	60.00	**69.09**	58.18	57.58	**65.06**	55.58	57.39
BBCSport	56.80	83.09	**87.13**	56.80	79.41	**85.66**	59.01	82.17	**82.35**	58.82	82.17	**82.35**	58.82	82.17	**82.35**	59.13	82.02	**82.38**
Movies	23.82	26.74	**30.79**	24.96	**27.39**	26.58	24.31	**27.71**	26.74	25.61	29.71	**29.98**	25.45	27.39	**29.66**	24.03	27.46	**28.26**
100Leaves	80.50	**88.63**	78.38	52.75	77.13	**83.88**	76.69	78.75	**79.19**	82.31	**86.38**	81.06	81.00	**84.75**	83.25	80.52	**83.74**	78.15
PUR(%)																		
3Sources	64.50	**81.66**	80.47	65.09	**81.07**	**81.07**	58.58	77.51	**79.29**	62.72	78.70	**79.88**	57.99	**78.70**	**78.70**	60.00	**78.73**	**78.37**
Cora	39.84	63.07	**67.13**	43.28	62.22	**67.13**	43.02	62.67	**66.32**	45.38	62.37	**66.29**	45.35	60.78	**65.99**	45.28	62.41	**66.27**
Handwritten	77.25	91.65	**93.80**	79.60	91.25	**93.30**	76.25	91.95	**94.20**	77.85	92.00	**94.35**	77.80	91.90	**94.30**	77.15	91.95	**94.33**
Yale	61.21	55.76	**64.24**	**64.85**	58.79	65.45	59.39	**61.21**	60.00	59.39	**60.61**	**60.61**	**69.70**	58.18	58.18	**65.58**	56.09	57.97
BBCSport	65.62	83.09	**87.13**	64.89	82.17	**85.66**	71.69	82.17	**82.35**	71.88	82.17	**82.35**	71.88	82.17	**82.35**	71.77	82.02	**82.38**
Movies	27.23	29.98	**32.25**	27.07	31.12	**31.77**	28.20	**31.28**	29.98	30.96	31.44	**31.93**	30.96	30.79	**31.44**	28.22	30.66	**31.56**
100Leaves	82.75	**90.31**	81.12	57.44	80.56	**85.06**	79.69	**81.94**	81.06	84.88	**88.75**	82.44	84.31	**87.56**	84.75	83.53	**86.53**	81.05
NMI(%)																		
3Sources	44.50	**70.04**	69.31	44.91	**69.37**	**69.37**	38.37	63.00	**65.46**	43.03	63.59	**66.39**	38.29	**64.08**	63.59	38.79	60.95	**61.32**
Cora	16.51	39.40	**46.96**	18.35	39.64	**46.96**	17.34	37.89	**44.30**	19.76	37.60	**44.24**	20.06	36.63	**44.30**	19.85	37.72	**44.34**
Handwritten	70.26	82.85	**86.31**	72.30	82.11	**85.90**	67.84	83.16	**87.23**	68.73	83.30	**87.38**	69.89	83.18	**87.19**	68.70	83.15	**87.37**
Yale	63.64	59.11	**67.52**	66.94	62.43	**67.53**	61.78	61.14	**62.96**	61.78	60.24	**64.35**	**70.47**	60.07	60.66	**67.42**	58.98	60.73
BBCSport	41.44	67.22	**73.30**	41.84	60.93	**73.46**	40.00	62.28	**62.63**	41.05	62.61	**62.63**	41.05	62.14	**62.63**	40.50	62.10	**62.67**
Movies	23.98	26.33	**29.02**	26.06	26.31	**26.67**	23.22	**25.72**	25.51	**27.45**	25.60	27.19	**28.02**	25.02	27.52	25.13	25.05	**26.79**
100Leaves	91.94	**95.28**	90.78	78.25	90.28	**91.90**	89.67	**91.75**	89.67	92.62	**94.83**	90.88	92.37	**94.17**	91.68	92.13	**93.85**	90.35

Table 1. The clustering results on seven datasets (%). The best results are marked in bold.

- **100 Leaves** dataset comprises 100 types of plant leaves, totaling 1,600 data samples. Each sample is characterized by fine-scale edges, shape descriptors, and texture histograms.

4.2 Multiple Kernel Algorithms

In this paper, we apply kernels constructed by traditional methods, CMK, and R-CMK, respectively, to different multiple kernel learning approaches. We compare the clustering performance of nine algorithms, including as follows.

- **AVG-KKM** assigns equal weights to the kernel matrices of all views and combines them to obtain a unified kernel. The kernel k-means clustering algorithm (KKM) is then applied on this unified kernel.
- **MKKM** [Huang *et al.*, 2012] is the most classic multi-view kernel learning algorithm. It employs an iterative optimization method to update clustering partitions and kernel weights until convergence. We replicate this algorithm in our experiments and compared it with other more advanced algorithms.
- **MKKM-MIR** [Liu *et al.*, 2016] reduces redundancy among base kernels and enhances diversity by incorporating an effective matrix-induced regularization.
- **ONKC** [Liu *et al.*, 2017] aims to find an optimal kernel within the neighborhood of the linearly combined kernel, thereby improving the representability of the optimal kernel and reinforcing the interaction between kernel learning and clustering.
- **LF-MVC** [Wang *et al.*, 2019] LF-MVC first performs partition based on the kernels of individual views, and then integrates them to obtain the optimal partition.
- **SimpleMKKM** [Liu, 2023] introduces a novel clustering loss by minimizing alignment on kernel weights and maximizing it on kernel partitions. The authors reformulate it as a minimization problem and propose a reduced gradient descent algorithm for optimization.

In addition to AVG-KKM and MKKM, for other methods, we utilize publicly available code released by the authors and employe the settings recommended in their papers or code.

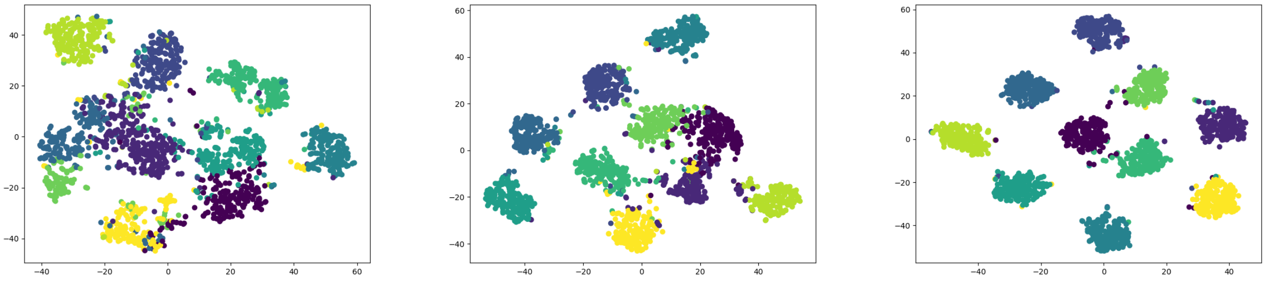
4.3 Training Procedure

R-CMK aims to learn V groups of projection coefficients, mapping views with different feature dimensions into a common latent space for unified measurement of similarity between instances from different views. The model training process consists of three steps, including data preprocessing, contrastive loss optimization, and downstream clustering tasks.

1) Data preprocessing: Due to potential variations in scale among different view features, we first normalize each view separately, scaling them to the range $(0, 1)$. This helps prevent abrupt changes or unnatural fluctuations in the loss function values during the optimization process.

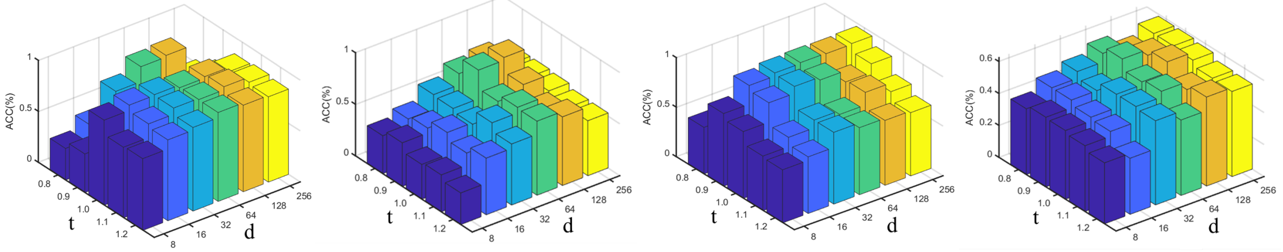
2) Contrastive loss optimization: As described in Section 3.1, we project the data, normalize it, and then calculate the robust loss. We utilize the Adam algorithm to optimize the objective function.

3) Downstream clustering: The optimal projection coefficient matrix obtained by contrastive learning is used to project and normalize the original data, and the resulting representation is employed to compute the kernel matrix. In downstream tasks, any kernel-based multi-view learning method can be utilized.



(a) Traditional (b) CMK (c) R-CMK

Figure 2: The t-SNE visualization of the clustering results on Handwritten dataset.



(a) 100Leaves (b) BBCSport (c) Handwritten (d) Yale

Figure 3: The effect of parameters on four datasets.

4.4 Experiment Results

In the experiment, we employ a Gaussian kernel function. Table 1 elaborate on the ACC, purity and NMI across all methods. The best results are marked in bold. From this table, we can draw the following conclusions.

1) The kernels constructed based on CMK and R-CMK for multi-view kernel learning methods have shown significant performance improvement across multiple datasets. For instance, in the case of the MKKM method on the Cora dataset, compared to the kernel constructed by traditional methods, the accuracy was enhanced by 54.98% and 68.35% for CMK and R-CMK, respectively. Kernels constructed based on R-CMK, when compared to the CMK method, demonstrated even higher performance improvements across various datasets.

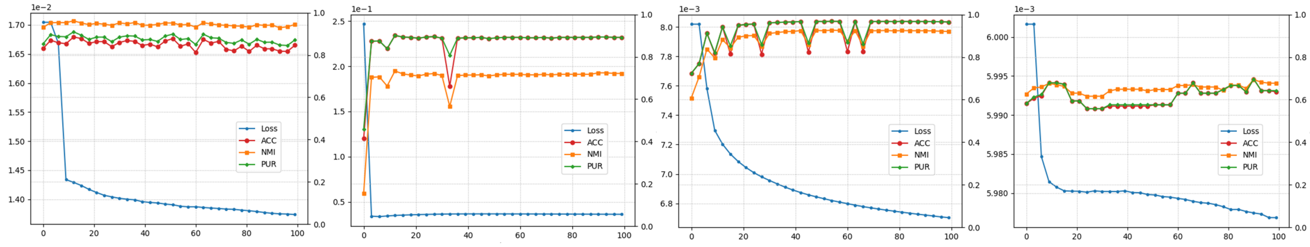
2) When the quality of the constructed kernel is high, many multi-view kernel learning methods can achieve good performance, even the simplest one like AVG-KKM. For instance, on the Handwritten dataset, the kernels constructed using CMK and R-CMK reached accuracies of 91.65% and 93.80%, respectively. These performances are comparable to or even surpass some more complex methods. We employ t-SNE technique to visualize the clustering results obtained using the AVG-KKM method on the Handwritten dataset, as depicted in Figure 2.

3) In some configurations, the performance of CMK and R-CMK experience a decline, which could be attributed to multiple factors. One possibility is that the experimented parameters may not have been optimal. Additionally, the clustering results might be influenced by the inherent characteristics of the multi-view kernel learning algorithm. For example, on the Yale dataset, the highest accuracies are achieved by AVG-KKM and MKKM at 63.63% and 64.85%, respectively, using the kernel constructed by R-CMK. MKKM-MIR and ONKC with the kernel constructed by CMK achieve their best accuracies of 61.21% and 60.61%, respectively. LF-MVC and SimpleMKKM demonstrated their highest accuracies of 69.09% and 65.06%, with kernels constructed through traditional methods.

In addition to accuracy, R-CMK shows similar conclusions across various methods in other clustering metrics, such as purity and NMI. Overall, R-CMK proves to be stable and effective across different evaluation criteria.

4.5 Parameter Analysis

In this section, we investigate the influence of different parameter setting to clustering performance. During the training of the model, m is a crucial parameter that essentially reflects the partitioning of negative pairs. Due to variations in model configurations, we did not adopt the approach in [Yang *et al.*, 2023], which sets m as the sum of the average distance between positive pairs and the average distance between negative pairs. Instead, we use this value as a baseline, multiplying it by a tunable parameter in $(0.5, 2]$ for flexible



(a) 100Leaves

(b) BBCSport

(c) Handwritten

(d) Yale

Figure 4: Loss value (left axis) and clustering performance (right axis) of each epoch on four datasets.

adjustment:

$$m = \frac{1}{2}\left(\frac{1}{Np}\sum d\left(\mathbf{z}_i^p, \mathbf{z}_i^{p'}\right) + \frac{1}{Nn}\sum_{i\neq j} d\left(\mathbf{z}_i^p, \mathbf{z}_j^{p'}\right)\right) * t \tag{24}$$

Another essential parameter is the dimension d_l of the projected features. As outlined in Section 3.1, we choose a projection dimension d_l and initialize the relevant projection matrices for different views. If d_l is too small, it might lead to the loss of information in the projected features; conversely, if d_l is too large, it could escalate the complexity of optimization and storage requirements. Typically, we begin with $d_l = 16$ and progressively double the value until reaching the anticipated performance.

To obtain the optimal parameter settings, we conduct a grid search on four datasets including 100Leaves, BBCSport, Handwritten and Yale. We explore the parameter t in the range of [0.8, 0.9, 1.0, 1.1, 1.2] and d_l in the range of [8, 16, 32, 64, 128, 256]. For the sake of presentation, we do not exhaustively list all available parameters. Many datasets exhibit optimal or satisfactory performance within this parameter range. Taking the aforementioned four datasets as examples, the clustering results are illustrated in Figure 3. It can be observed that the dimensionality d_l of the projection feature is a more critical parameter. An intuitive understanding is that if d_l is too small, the projection feature will lose too much information from the original view. If d_l is too large, it will lead to redundancy in the projection coefficient matrix, making it challenging to optimize. Hence, both scenarios can result in a decline in clustering performance.

4.6 Convergence Analysis

We analyze the convergence of the loss function and clustering performance based on the iteration count of the proposed method. MKKM is utilized as the downstream clustering algorithm, and we run MKKM in each iteration, examining the clustering results on above four datasets. The maximum number of epoch is set to 100. It is advisable to initially run the program under the CMK setting to establish some discriminative power between negative pairs and positive pairs before transitioning to the R-CMK setting. This prevents the robust loss from hindering the model's ability to fit true negative pairs. In this paper, we adopt a strategy where, within a sliding window (default size is 5, meaning the program runs under the CMK setting at least 5 times), the loss is switched to the robust loss when the average distance of negative pairs exceeds the average distance of positive pairs. The loss function curve and performance curves are illustrated in Figure 4. It can be observed that the loss values for different datasets converge within 100 epochs. Although the accuracy curves fluctuate to some extent, the overall trend still shows an increase as the loss values decrease, eventually reaching a stable point or fluctuating around that value.

5 Conclusion

In this paper, we propose a robust multi-view kernel construction method against false negative pairs. This method identifies potential false negative pairs by assessing the similarity between negative pairs. During the optimization process based on gradient descent, it reverses the optimization gradient specifically for false negative pairs, effectively narrowing the distance between them. Comprehensive experimental results demonstrate that our proposed method significantly enhances the quality of constructed kernels, thereby notably improving the performance of multi-view kernel clustering. We also analyze the impact of different hyperparameters and convergence speed of the optimization process. The results indicate that it converges quickly but is somewhat sensitive to the dimensionality d_l of projection features. Therefore, appropriate experimentation and selection are necessary. In the future, we plan to delve deeper into how to more reasonably construct negative pairs in the context of multi-view learning. Our aim is to identify or avoid the existence of false negative pairs and effectively reduce the distance between samples within clusters.

Acknowledgments

This work was partially supported by the National Science Foundation of China under Grant 62106164 and 62376175, the 111 Project under Grant B21044, and the Sichuan Science and Technology Program under Grants 2021ZDZX0011.

References

[Bisson and Grimal, 2012] Gilles Bisson and Clément Grimal. Co-clustering of multi-view datasets: A parallelizable approach. In *2012 IEEE 12th International Conference on Data Mining*, pages 828–833, 2012.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, volume 119, 2020.

[Cristianini *et al.*, 2001] Nello Cristianini, John Shawe-Taylor, and Jaz Kandola. Spectral kernel methods for clustering. *Advances in Neural Information Processing Systems*, 14, 2001.

[Dhillon *et al.*, 2004] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 551–556, 2004.

[Gong *et al.*, 2012] Maoguo Gong, Yan Liang, Jiao Shi, Wenping Ma, and Jingjing Ma. Fuzzy c-means clustering with local information and kernel metric for image segmentation. *IEEE Transactions on Image Processing*, 22(2):573–584, 2012.

[Greene and Cunningham, 2006] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pages 377–384, 2006.

[He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020.

[Huang *et al.*, 2012] Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen. Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 20(1):120–134, 2012.

[Huang *et al.*, 2019] Shudong Huang, Zhao Kang, Ivor W Tsang, and Zenglin Xu. Auto-weighted multi-view clustering via kernelized graph learning. *Pattern Recognition*, 88:174–184, 2019.

[Kang *et al.*, 2018] Zhao Kang, Xiao Lu, Jinfeng Yi, and Zenglin Xu. Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2312–2318, 2018.

[Kingma and Ba, 2017] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[Liu *et al.*, 2016] Xinwang Liu, Yong Dou, Jianping Yin, Lei Wang, and En Zhu. Multiple kernel k-means clustering with matrix-induced regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, pages 1888–1894, 2016.

[Liu *et al.*, 2017] Xinwang Liu, Sihang Zhou, Yueqing Wang, Miaomiao Li, Yong Dou, En Zhu, and Jianping Yin. Optimal neighborhood kernel clustering with multiple kernels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI Conference on Artificial Intelligence, pages 2266–2272, 2017.

[Liu *et al.*, 2020a] Jing Liu, Fuyuan Cao, Xiao-Zhi Gao, Liqin Yu, and Jiye Liang. A cluster-weighted kernel k-means method for multi-view clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4860–4867, 2020.

[Liu *et al.*, 2020b] Jiyuan Liu, Xinwang Liu, Jian Xiong, Qing Liao, Sihang Zhou, Siwei Wang, and Yuexiang Yang. Optimal neighborhood multiple kernel clustering with adaptive local kernels. *IEEE Transactions on Knowledge and Data Engineering*, 34(6):2872–2885, 2020.

[Liu *et al.*, 2021] Jiyuan Liu, Xinwang Liu, Siwei Wang, Sihang Zhou, and Yuexiang Yang. Hierarchical multiple kernel clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8671–8679, 2021.

[Liu *et al.*, 2023] Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Qing Liao, and Yuanqing Xia. Contrastive multi-view kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9552–9566, 2023.

[Liu, 2023] Xinwang Liu. Simplemkkm: Simple multiple kernel k-means. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5174–5186, 2023.

[Nie *et al.*, 2017] Feiping Nie, Guohao Cai, and Xuelong Li. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, page 2408–2414, 2017.

[Ren *et al.*, 2020] Zhenwen Ren, Simon X Yang, Quansen Sun, and Tao Wang. Consensus affinity graph learning for multiple kernel clustering. *IEEE Transactions on Cybernetics*, 51(6):3273–3284, 2020.

[Tang *et al.*, 2023] Yiming Tang, Zhifu Pan, Xianghui Hu, Witold Pedrycz, and Renhao Chen. Knowledge-induced multiple kernel fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[Wang *et al.*, 2019] Siwei Wang, Xinwang Liu, En Zhu, Chang Tang, Jiyuan Liu, Jingtao Hu, Jingyuan Xia, and Jianping Yin. Multi-view clustering via late fusion alignment maximization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3778–3784, 2019.

[Xiao *et al.*, 2015] Shijie Xiao, Mingkui Tan, Dong Xu, and Zhao Yang Dong. Robust kernel low-rank representation. *IEEE Transactions on Neural Networks and Learning Systems*, 27(11):2268–2281, 2015.

[Xie *et al.*, 2017] Xingyu Xie, Xianglin Guo, Guangcan Liu, and Jun Wang. Implicit block diagonal low-rank representation. *IEEE Transactions on Image Processing*, 27(1):477–489, 2017.

[Yang *et al.*, 2023] Mouxing Yang, Yunfan Li, Peng Hu, Jinfeng Bai, Jiancheng Lv, and Xi Peng. Robust multi-view clustering with incomplete information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1055–1069, 2023.