MULTIPLE KERNEL K-MEANS CLUSTERING WITH SIMULTANEOUS SPECTRAL ROTATION

Jitao Lu[1,2] *Yihang Lu*[1,2] *Rong Wang*[2*] *Feiping Nie*[1,2] *Xuelong Li*[2]

[1] School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, P. R. China.
[2] School of Artificial Intelligence, Optics and Electronics (iOPEN), and the Key Laboratory of Intelligent Interaction and Applications (Ministry of Industry and Information Technology), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China.

ABSTRACT

Multiple kernel k-means clustering (MKKM) and its variants have been thoroughly studied over the past decades. However, most existing models utilize a spectrum-based two-step approach to solve the clustering objective, which may deviate from the final cluster labels and lead to suboptimal performance. To address this issue, we elaborate a novel MKKM-SR framework that simultaneously optimizes the discrete and continuous cluster labels by incorporating spectral rotation into MKKM. In addition, the proposed model can be easily integrated with other MKKM models to boost their performance. What's more, an efficient alternative algorithm is proposed to solve the joint optimization problem. Extensive experiments on real-world datasets demonstrate the superiorities of the proposed framework.

***Index Terms*—** kernel method, kernel k-means, multiple kernel clustering

1. INTRODUCTION

As an important extension to k-means clustering, kernel k-means clustering (KKM) applies the kernel trick [1] to handle non-linear separable data [2,3], but its performance is heavily influenced by the chosen kernel function [4]. To deal with intractable kernel selection, multiple kernel k-means (MKKM) clustering [5] and its variants were developed to learn from set of pre-specified base kernels. Formally, given a set of pre-specified base kernels $\{\mathbf{K}_p \in \mathbb{R}^{n\times n}\}_{p=1}^{v}$ in which the $\langle i,j\rangle$-th element of a kernel can be interpreted as the inner product of the corresponding data samples in a Reproducing Kernel Hilbert Space [6,7], the consensus kernel is expressed as $\mathbf{K}_{\boldsymbol{\gamma}} = \sum_{p=1}^{v} \gamma_p^2 \mathbf{K}_p$ in the classic MKKM literature, where $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_v]^T$ are the associated weights of each kernel subject to $\boldsymbol{\gamma}^T\mathbf{1} = 1, \gamma_p \geqslant 0, \forall p$. The objective of MKKM is formulated as

$$\begin{aligned} &\min_{\mathbf{Y},\boldsymbol{\gamma}} \operatorname{Tr}(\mathbf{K}_{\boldsymbol{\gamma}}(\mathbf{I}_n - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T)), \\ &\text{s.t. } \mathbf{Y} \in \text{Ind}, \boldsymbol{\gamma}^T\mathbf{1} = 1, \gamma_p \geqslant 0, \forall p, \end{aligned} \tag{1}$$

where $\mathbf{Y} \in \mathbb{B}^{n\times c}$ is zero-one valued and referred to as the **discrete cluster indicator matrix**: if the i-th sample belongs to the j-th cluster, we have $y_{ij} = 1$, otherwise $y_{ij} = 0$. $\mathbf{I}_n$ is a $n \times n$ identity matrix. $\operatorname{Tr}(\cdot)$ is the trace of a square matrix. Problem (1) can be solved by alternatively updating $\boldsymbol{\gamma}$ and $\mathbf{Y}$. The subproblem of $\boldsymbol{\gamma}$ is a standard quadratic programming (QP) problem with linear constraints and can be solved by commercial QP solvers. The $\mathbf{Y}$ subproblem, however, is hard to solve due to the discrete constraint of $\mathbf{Y}$. A two-stage approach is widely adopted [5, 8–10], in which the discrete constraint of $\mathbf{Y}$ is relaxed to allow $\mathbf{F} = \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}$ to take arbitrary real values. The problem is transformed into a trace maximization problem

$$\max_{\mathbf{F}^T\mathbf{F}=\mathbf{I}_c} \operatorname{Tr}(\mathbf{F}^T\mathbf{K}_{\boldsymbol{\gamma}}\mathbf{F}), \tag{2}$$

where the optimal $\mathbf{F}$ is formed by the eigenvectors corresponding to the largest c eigenvalues of $\mathbf{K}_{\boldsymbol{\gamma}}$. k-means clustering is applied to obtain the final cluster labels. $\mathbf{F}$ is referred to as **continuous cluster indicator matrix** in this literature.

a) Brief related works: Several MKKM variants have been developed over the past decade. Huang *et al*. proposed multiple kernel fuzzy c-means (MKFC) to apply the multiple kernel settings to fuzzy clustering [11]. Gönen *et al*. proposed localized multiple kernel k-means (LMKKM) to fuse kernels with sample-specific weights [8]. Du *et al*. used $\ell_{2,1}$-norm to measure the distances between data points and cluster centers and proposed a robust multiple kernel k-means clustering (RMKKM) algorithm to improve the robustness with respect to noises and outliers [12]. Liu *et al*. argued that existing MKKM models don't significantly consider the correlation among kernels and designed a matrix-induced regularization to enhance kernel fusion [9]. Unlike previous works that

*Corresponding author. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1403501, in part by the Natural Science Basic Research Program of Shaanxi (Program No. 2021JM-071), in part by the National Natural Science Foundation of China under Grant 62176212, Grant 61936014 and Grant 61772427, and in part by the Fundamental Research Funds for the Central Universities under Grant G2019KY0501.

require the optimal kernel to be a weighted sum of base kernels, Liu *et al*. proposed to enhance the representability of the optimal kernel by allowing it to reside in the neighborhood of the later one [10]. Recently, Yao *et al*. proposed a strategy to select a diverse subset from the pre-specified base kernels to enhance the diversity, which interprets each kernel as encoded by others and make the selected subset minimizing the total encoding cost [13].

b) Our contributions: We observe that previous MKKM works focused on integrating well-crafted regularizers with the original MKKM, but rarely considered to improve its optimization. As mentioned above, state-of-the-art MKKM methods and their variants all adopt the two-stage spectral relaxation approach to solve the clustering objective. However, the solution obtained from two-stage methods may deviate far from the true discrete solution, leading to suboptimal or bad performance. We elaborate a novel MKKM-SR framework to address the issue. The proposed framework simultaneously considers the discrete and continuous cluster indicator matrices by incorporating spectral rotation [14] into MKKM to penalize the deviation. In addition, unlike previous works that rely on commercial QP solvers to solve the coefficients of each kernel, we propose to directly obtain them with a closed-form solution, which is more efficient and numerically stable.

2. PROPOSED METHOD

2.1. Problem Formulation

Among classic MKKM methods and its variants, the continuous cluster indicator matrix $\mathbf{F}$ is obtained by solving the clustering objective, but it is unable to consider the cluster memberships of samples. On the other hand, the discrete cluster indicator matrix $\mathbf{Y}$ is capable of considering cluster memberships, but it's obtained by applying discretization procedures like k-means to the continuous one, which may deviate from the clustering objective. In order to fill the gap between $\mathbf{Y}$ and $\mathbf{F}$ caused by two-stage solution, we propose to combine them together to get their own advantages. To be specific, we bridge them by injecting the spectral rotation objective into the model. Spectral rotation pushes the continuous cluster indicator matrix $\mathbf{F}$ to approximate the discrete cluster indicator matrix $\mathbf{Y}$. As a result, the continuous one not only satisfies the clustering objective, but also perfectly negotiates with the discrete one. Noteworthily, it's been proved that the zero-one valued discrete cluster indicator matrix used in classic spectral rotation is difficult to approximate the continuous one [15, 16], because the former is orthogonal but the latter is orthonormal, *i.e*., the lengths of their rows are different. Thus, we propose to approximate the *scaled* discrete indicator matrix $\mathbf{Y}$ introduced in improve spectral rotation (ISR) [15]. A trade-off hyperparameter λ is added to balance the approximation and the clustering objective.

Given a set of pre-specified base kernels $\{\mathbf{K}_p\}_{p=1}^v$, we propose to fuse them by $\mathbf{K}_{\boldsymbol{\alpha}} = \sum_{p=1}^v \frac{1}{\alpha_p}\mathbf{K}_p$, where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_v]^T$ are the learned coefficients of each kernel. As we'll demonstrate in Section 2.2, this fusion schema effectively avoids the reliance on commercial QP solvers. Our proposed MKKM-SR framework is then formulated as

$$\begin{aligned}&\min_{\mathbf{F},\mathbf{R},\mathbf{Y},\boldsymbol{\alpha}} \operatorname{Tr}(\mathbf{K}_{\boldsymbol{\alpha}}(\mathbf{I}_n - \mathbf{F}\mathbf{F}^T)) + \lambda\|\mathbf{F}\mathbf{R} - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\|_F^2,\\ &\text{s.t. } \mathbf{F}^T\mathbf{F} = \mathbf{I}, \mathbf{R}^T\mathbf{R} = \mathbf{I}, \mathbf{Y} \in \text{Ind}, \boldsymbol{\alpha}^T\mathbf{1} = 1, \alpha_p > 0, \forall p,\end{aligned} \tag{3}$$

where $\mathbf{R}$ is a rotation matrix to adjust the continuous cluster indicator matrix $\mathbf{F}$ to approximate the scaled discrete cluster indicator matrix $\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}$. It's clear that both $\mathbf{F}$ and $\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}$ are orthonormal, resulted in more reasonable approximation than the classic spectral rotation.

2.2. Optimization

Problem (3) seems difficult to solve directly. In this section, we propose an alternative optimization approach to solve it.

Step 1: Update $\mathbf{F}$ with $\mathbf{R}, \mathbf{Y}$ and $\boldsymbol{\alpha}$ fixed. Problem (3) can be rewritten as

$$\max_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} \operatorname{Tr}(\mathbf{F}^T\mathbf{K}_{\boldsymbol{\alpha}}\mathbf{F}) + 2\lambda \operatorname{Tr}(\mathbf{F}^T\mathbf{B}), \tag{4}$$

where $\mathbf{B} \triangleq \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{\frac{1}{2}}\mathbf{R}^T$. The Lagrangian function of problem (4) can be written as

$$\mathcal{L}(\mathbf{F}, \boldsymbol{\Gamma}) = \operatorname{Tr}(\mathbf{F}^T\mathbf{K}_{\boldsymbol{\alpha}}\mathbf{F}) + 2\lambda \operatorname{Tr}(\mathbf{F}^T\mathbf{B}) - \operatorname{Tr}(\boldsymbol{\Gamma}(\mathbf{F}^T\mathbf{F} - \mathbf{I}_c)), \tag{5}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{c\times c}$ denotes the matrix composed of Lagrangian multipliers. Therefore, the optimal $\mathbf{F}$ should satisfy the KKT condition:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{F}} = 2\mathbf{K}_{\boldsymbol{\alpha}}\mathbf{F} + 2\lambda\mathbf{B} - 2\mathbf{F}\boldsymbol{\Gamma} = 0, \tag{6}$$

which can be solved by generalized power iteration [17].

Step 2: Update $\mathbf{R}$ with $\mathbf{F}, \mathbf{Y}$ and $\boldsymbol{\alpha}$ fixed. Problem (3) can be rewritten as

$$\max_{\mathbf{R}^T\mathbf{R}=\mathbf{I}} \operatorname{Tr}(\mathbf{R}^T\mathbf{N}), \tag{7}$$

where $\mathbf{N} \triangleq \mathbf{F}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} \in \mathbb{R}^{c\times c}$. Let $\mathbf{N} = \mathbf{U}_\mathbf{N}\boldsymbol{\Sigma}_\mathbf{N}\mathbf{V}_\mathbf{N}^T$ be the singular value decomposition (SVD) of $\mathbf{N}$, we have

$$\operatorname{Tr}(\mathbf{R}^T\mathbf{N}) = \operatorname{Tr}(\boldsymbol{\Sigma}_\mathbf{N}\mathbf{V}_\mathbf{N}^T\mathbf{R}^T\mathbf{U}_\mathbf{N}) = \operatorname{Tr}(\boldsymbol{\Sigma}_\mathbf{N}\boldsymbol{\Phi}) = \sum_{i=1}^c \sigma_{ii}\phi_{ii}, \tag{8}$$

where $\boldsymbol{\Phi} \triangleq \mathbf{V}_\mathbf{N}^T\mathbf{R}^T\mathbf{U}_\mathbf{N}$, σ_{ii} and ϕ_{ii} are the $\langle i,i\rangle$-th elements of matrices $\boldsymbol{\Sigma}_\mathbf{N}$ and $\boldsymbol{\Phi}$, respectively.

Note that $\boldsymbol{\Phi}\boldsymbol{\Phi}^T = \mathbf{V}_\mathbf{N}^T\mathbf{R}^T\mathbf{U}_\mathbf{N}\mathbf{U}_\mathbf{N}^T\mathbf{R}\mathbf{V}_\mathbf{N} = \mathbf{I}_c$, which means $\sum_{j=1}^c \phi_{ij}^2 = 1, i \in \{1, \ldots, c\}$, so $-1 \leqslant \phi_{ii} \leqslant 1$. Meanwhile, $\sigma_{ii} \geqslant 0, \forall i$ since σ_{ii} are the singular values of $\mathbf{N}$. Therefore, $\operatorname{Tr}(\mathbf{R}^T\mathbf{N}) \leqslant \sum_{i=1}^c \sigma_{ii}$ and the equality holds when $\phi_{ii} = 1$, *i.e*., $\boldsymbol{\Phi} \triangleq \mathbf{V}_\mathbf{N}^T\mathbf{R}^T\mathbf{U}_\mathbf{N} = \mathbf{I}_c$. Thus, we obtain the optimal solution to problem (7) as $\mathbf{R}^* = \mathbf{U}_\mathbf{N}\mathbf{V}_\mathbf{N}^T$.

Step 3: Update $\mathbf{Y}$ with $\mathbf{F}, \mathbf{R}$ and α fixed. Problem (3) can be reduced to

$$\max_{\mathbf{Y}\in\text{Ind}} \text{Tr}(\mathbf{U}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}), \tag{9}$$

where $\mathbf{U} \triangleq \mathbf{FR}$. Next, we propose an efficient iterative algorithm based on coordinate descent to solve $\mathbf{Y}$ *row by row*, during which all other rows are fixed except the i-th row being updated to minimize problem (9).

Denote $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_c]$ and $\mathbf{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_c]$ where $\boldsymbol{y}_j$ and $\boldsymbol{u}_j$ are the j-th columns of $\mathbf{Y}$ and $\mathbf{U}$, respectively. Then, problem Eq. (9) can be rewritten as

$$\max_{\mathbf{Y}\in\text{Ind}} g(\mathbf{Y}) = \sum_{j=1}^{c} \frac{\boldsymbol{u}_j^T\boldsymbol{y}_j}{\sqrt{\boldsymbol{y}_j^T\boldsymbol{y}_j}}. \tag{10}$$

There are c candidate solutions when updating a row of $\mathbf{Y}$, each with varying position of element 1 in that row, denoting them as $\{\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(c)}\}$. Additionally, $\mathbf{Y}^{(0)}$ refers to a matrix with all elements of the active row being zeros. Obviously, it is a good idea to turn Eq. (10) into two subtractions, which still maintain the optimal solution it would have. So the problem can be equivalently transformed subtracting a constant $g(\mathbf{Y}^{(0)})$ as following.

$$\max_{l\in\{1,\ldots,c\}} \mathcal{L}(\mathbf{Y}^{(l)}) = \frac{\boldsymbol{u}_l^T\boldsymbol{y}_l^{(l)}}{\sqrt{\boldsymbol{y}_l^{(l)T}\boldsymbol{y}_l^{(l)}}} - \frac{\boldsymbol{u}_l^T\boldsymbol{y}_l^{(0)}}{\sqrt{\boldsymbol{y}_l^{(0)T}\boldsymbol{y}_l^{(0)}}}. \tag{11}$$

Without loss of generality, we assume the m-th candidate $\mathbf{Y}^{(m)}$ equals to the latest iteration $\mathbf{Y}$, the i-th row of whom is being updated. For different l, we observe that Eq. (11) can be efficiently calculated as the following two cases.

Case 1: When $l = m$, we have $\boldsymbol{y}_l^{(l)} = \boldsymbol{y}_l = \boldsymbol{y}_l^{(0)} + \boldsymbol{\delta}$ denoting $\boldsymbol{\delta} \in \mathbb{R}^n$ as a vector with the i-th element being 1 and rest being 0s, so $\boldsymbol{u}_l^T\boldsymbol{y}_l^{(l)} = \boldsymbol{u}_l^T\boldsymbol{y}_l, \sqrt{\boldsymbol{y}_l^{(l)T}\boldsymbol{y}_l^{(l)}} = \sqrt{\boldsymbol{y}_l^T\boldsymbol{y}_l}, \boldsymbol{u}_l^T\boldsymbol{y}_l^{(0)} = \boldsymbol{u}_l^T\boldsymbol{y}_l - u_{il}, \sqrt{\boldsymbol{y}_l^{(0)T}\boldsymbol{y}_l^{(0)}} = \sqrt{\boldsymbol{y}_l^T\boldsymbol{y}_l - 1}$, where u_{il} is the i-th element of $\boldsymbol{u}_l$.

Case 2: When $l \neq m$, we have $\boldsymbol{y}_l^{(0)} = \boldsymbol{y}_l = \boldsymbol{y}_l^{(l)} - \boldsymbol{\delta}$, so $\boldsymbol{u}_l^T\boldsymbol{y}_l^{(l)} = \boldsymbol{u}_l^T\boldsymbol{y}_l + u_{il}, \sqrt{\boldsymbol{y}_l^{(l)T}\boldsymbol{y}_l^{(l)}} = \sqrt{\boldsymbol{y}_l^T\boldsymbol{y}_l + 1}, \boldsymbol{u}_l^T\boldsymbol{y}_l^{(0)} = \boldsymbol{u}_l^T\boldsymbol{y}_l, \sqrt{\boldsymbol{y}_l^{(0)T}\boldsymbol{y}_l^{(0)}} = \sqrt{\boldsymbol{y}_l^T\boldsymbol{y}_l}$.

In summary, $\mathcal{L}(\mathbf{Y}^{(l)})$ can be efficiently calculated as

$$\mathcal{L}(\mathbf{Y}^{(l)}) = \begin{cases} \frac{\boldsymbol{u}_l^T\boldsymbol{y}_l}{\sqrt{\boldsymbol{y}_l^T\boldsymbol{y}_l}} - \frac{\boldsymbol{u}_l^T\boldsymbol{y}_l - u_{ik}}{\sqrt{\boldsymbol{y}_l^T\boldsymbol{y}_l - 1}}, & \text{if } l = m, \\ \frac{\boldsymbol{u}_l^T\boldsymbol{y}_l + u_{ik}}{\sqrt{\boldsymbol{y}_l^T\boldsymbol{y}_l + 1}} - \frac{\boldsymbol{u}_l^T\boldsymbol{y}_l}{\sqrt{\boldsymbol{y}_l^T\boldsymbol{y}_l}}, & \text{if } l \neq m. \end{cases} \tag{12}$$

Then, it is explicit that the optimal solution is the one that maximizing Eq. (12).

Step 4: Update $\boldsymbol{\alpha}$ with $\mathbf{F}, \mathbf{R}$ and $\mathbf{Y}$ fixed. Problem (3) becomes

$$\min_{\boldsymbol{\alpha}^T\mathbf{1}=1, \alpha_p>0, \forall p} \sum_{p=1}^{v} \frac{1}{\alpha_p} \text{Tr}(\mathbf{K}_p(\mathbf{I}_n - \mathbf{F}\mathbf{F}^T)). \tag{13}$$

Let $h_p \triangleq \text{Tr}(\mathbf{K}_p(\mathbf{I}_n - \mathbf{F}\mathbf{F}^T))$, the Lagrangian of problem (13) can be expressed as

$$\mathcal{L}(\boldsymbol{\alpha}, \beta) = \sum_{p=1}^{v} \frac{1}{\alpha_p} h_p + (\sum_{p=1}^{v} \alpha_p - 1)\beta, \tag{14}$$

where β is the Lagrangian multiplier. According to the KKT condition, we obtain the optimal $\boldsymbol{\alpha}$ as

$$\alpha_p = \frac{\sqrt{h_p}}{\sum_{q=1}^{v}\sqrt{h_q}}, \forall p = \{1, 2, \ldots, v\}. \tag{15}$$

Complexity Analysis. Denote the number of iterations of $\mathbf{F}$ step, $\mathbf{Y}$ step and whole algorithm as t_1, t_2 and t, the time complexity of the proposed MKKM-SR framework is $\mathcal{O}((nc^2t_1 + (nc^2 + c^3) + (n^2c)t_2 + n^2v)t)$. Comparing with previous works utilizing eigenvalue decomposition of $\mathcal{O}(n^3)$ complexity, our method is more efficient.

3. EXPERIMENTS

3.1. Evaluation Protocol

Five benchmark datasets are employed to evaluate the performance of the proposed model, they are all used in previous literatures [5, 12, 18, 19] and publicly available.

We compare the proposed MKKM-SR model with state-of-the-art multiple kernel k-means clustering models, including `MKKM` [5], `LMKKM` [8], `RMKKM` [12], `MKKM-MR` [9], `ONKC` [10], `MKKM-RK` [13] and a single kernel model `KKM-avg` applying KKM to averaged kernels. They are implemented in MATLAB and downloaded from the authors' pages.

Three widely adopted metrics are employed to evaluate the clustering performance, including clustering accuracy (ACC), normalized mutual information (NMI) and adjusted Rand index (ARI). For all comparison models with hyperparameter(s), we follow the guidance of their authors to search for the values and report the *best* results.

3.2. Result Analyses and Discussions

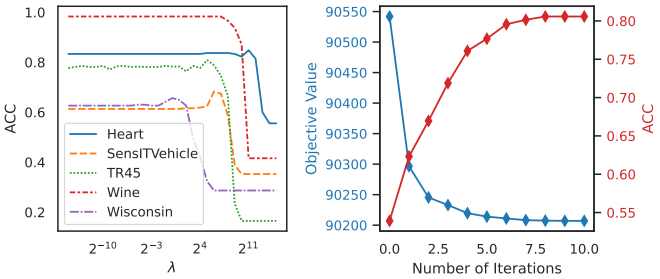
1) Clustering Performance: The experimental results on the aforementioned benchmark datasets with respect to ACC, NMI and ARI are reported in Table 1, from which we derive the following analyses: ***a)*** MKKM-SR significantly outperforms MKKM. On `Heart` dataset, it achieved $+\mathbf{0.34}$ gain of ACC, which is a huge improvement for clustering tasks. The verifies the effectiveness of our simultaneous optimization framework. ***b)*** Despite its success, MKKM-SR is extremely simple and does not rely on well-crafted regularizers, while MKKM-WR and ONKC achieved good performance via extra matrix-induced regularization and optimal neighborhood kernel learning schema. Nevertheless, the two are comparable in general. ***c)*** MKKM-SR outperforms other models while remaining comparable with them. It's promising

Table 1: Clustering performance with respect to ACC, NMI and ARI on 5 datasets. Best results are in bold.

Model	Heart			SensITVehicle			TR45			Wine			Wisconsin		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
KKM-avg	0.8222	0.3240	0.4131	0.5307	0.1036	0.1033	0.7261	0.6803	0.6169	0.9719	0.8804	0.9134	0.5283	0.3238	0.2671
LMKKM [8]	0.7926	0.2777	0.3400	0.4240	0.0674	0.0605	0.7261	0.6803	0.6169	0.9663	0.8635	0.8962	0.4566	0.1629	0.0730
RMKKM [12]	0.7630	0.2049	0.2738	0.4127	0.0549	0.0487	0.5623	0.5223	0.4544	0.9719	0.8897	0.9149	0.4151	0.0403	0.0334
MKKM-MR [9]	0.8333	0.3522	0.4424	0.6560	0.2113	0.2400	0.7522	0.7253	0.6689	0.9775	0.9091	0.9295	0.5925	0.3989	0.3416
ONKC [10]	0.8370	0.3622	0.4524	0.5433	0.1143	0.1149	0.7870	0.7086	0.6877	0.9775	0.9065	0.9309	0.5962	0.3360	0.3073
MKKM-RK [13]	0.7630	0.2049	0.2738	0.5373	0.1083	0.1084	0.7203	0.6689	0.6065	0.9663	0.8748	0.8992	0.5849	0.3687	0.3254
MKKM [5]	0.5037	0.0000	-0.0036	0.5567	0.1388	0.1584	0.7478	0.7181	0.6648	0.9719	0.8829	0.9122	0.5623	0.2532	0.2028
MKKM-SR	**0.8481**	**0.3884**	**0.4829**	**0.6827**	**0.2456**	**0.2827**	**0.8087**	**0.7399**	**0.7044**	**0.9831**	**0.9261**	**0.9471**	**0.6566**	**0.4019**	**0.3934**

Table 2: Averaged CPU time in seconds over 10 independent runs. Fastest results are in bold.

Dataset	Heart	SensIT Vehicle	TR45	Wine	Wisc-onsin
MKKM [5]	0.1495	0.8092	0.6679	0.1094	0.1700
LMKKM [8]	27.1959	51.3104	86.1698	7.2845	0.9162
RMKKM [12]	2.3987	156.5064	34.0909	3.5840	5.6207
MKKM-MR [9]	0.0897	0.6395	0.8951	0.0551	0.0388
ONKC [10]	0.1714	2.5156	1.4399	0.1477	0.1462
MKKM-RK [13]	2.2513	3.6767	2.4169	1.5845	3.7196
MKKM-SR	**0.0530**	**0.5178**	**0.1826**	**0.0119**	**0.0348**



(a) ACC with varying λ on benchmark datasets

(b) How objective value and ACC evolves on TR45 dataset

Fig. 1: Parameter sensitivity and convergence curves

to boost their performance by incorporating with our framework, and we'll leave this for future work.

2) Time Cost: We further compare the time cost to demonstrate the efficiency of our method. All comparison models are launched for 10 individual runs on an Arch Linux PC with Intel i7-7700 @ 4.2GHz CPU and MATLAB R2020b environment. The averaged time cost and standard deviation are reported in Table 2. We observe that our model runs faster than all comparison models. This is because the time complexities of previous works are cubic to n, while that of our MKKM-SR is just quadratic to n.

3) Parameter Sensitivity: Our framework has one hyper-parameter λ to balance the clustering objective and the discrepancy between $\mathbf{F}$ and $\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}$. From the perspective of optimization, λ is only involved when updating $\mathbf{F}$. When λ is too small, it won't encourage $\mathbf{F}$ to bridge the clustering objective and $\mathbf{Y}$, and our method degenerates to conventional two-step approach. When λ is too large, spectral rotation would dominate the model and the clustering objective is discarded, so the model would finally output a trivial solution.

To verify our theoretical analyses, we plot how λ affects the clustering performance in Figure 1(a). It's clear that our model obtains stable clustering performance when λ is not too large, and the performance drastically decreases when λ becomes very large, this verifies our analyses above. Notice that the performance peaks is reached when the two parts are well balanced, the potential of simultaneous optimization is maximized in this case.

4) Convergence: Figure 1(b) illustrates the objective value and corresponding ACC of our method on TR45 dataset at each iteration. The objective monotonically decreases and converges within 10 iterations, which again verifies the efficiency of our optimization algorithm. Meanwhile, the ACC increases after each iteration, confirming the effectiveness of our method.

4. CONCLUSION

In this paper, we proposed a novel multiple kernel k-means clustering framework, namely MKKM-SR. It's capable of optimizing the discrete and continuous cluster labels simultaneously, where improved spectral rotation is employed to incorporate them into a single model. In comparison with previous works, our method effectively avoids severe information loss arisen from the two-stage optimization strategy. We intend to leave out well-crafted regularizers for ablation study, and extensive experiments demonstrated its superb performance over conventional MKKM, revealing the advantage of simultaneous optimization. Our framework is extremely flexible and can be well-integrated with other MKKM models to further boost their performance.

5. REFERENCES

[1] Mark Girolami, "Mercer kernel-based clustering in feature space," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 780–784, 2002.

[2] Shusen Wang, Alex Gittens, and Michael W Mahoney, "Scalable kernel k-means clustering with nyström approximation: relative-error bounds," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 431–479, 2019.

[3] Giacomo Meanti, Luigi Carratino, Lorenzo Rosasco, and Alessandro Rudi, "Kernel methods through the roof: Handling billions of points efficiently," in *NeurIPS*, 2020.

[4] John Shawe-Taylor and Nello Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

[5] Shi Yu, Leon Tranchevent, Xinhai Liu, Wolfgang Glanzel, Johan AK Suykens, Bart De Moor, and Yves Moreau, "Optimized data fusion for kernel k-means clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1031–1039, 2011.

[6] Laurent Schwartz, "Sous-espaces hilbertiens d'espaces vectoriels topologiques et noyaux associés (noyaux reproduisants)," *Journal d'analyse mathématique*, vol. 13, no. 1, pp. 115–256, 1964.

[7] Bernhard Schölkopf and Alexander Johannes Smola, *Learning with Kernels: support vector machines, regularization, optimization, and beyond*, Adaptive computation and machine learning series. MIT Press, 2002.

[8] Mehmet Gönen and Adam A. Margolin, "Localized data fusion for kernel k-means clustering with application to cancer biology," in *NIPS*, 2014, pp. 1305–1313.

[9] Xinwang Liu, Yong Dou, Jianping Yin, Lei Wang, and En Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *AAAI*, 2016, pp. 1888–1894.

[10] Xinwang Liu, Sihang Zhou, Yueqing Wang, Miaomiao Li, Yong Dou, En Zhu, and Jianping Yin, "Optimal neighborhood kernel clustering with multiple kernels," in *AAAI*, 2017, pp. 2266–2272.

[11] Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen, "Multiple kernel fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 1, pp. 120–134, 2011.

[12] Liang Du, Peng Zhou, Lei Shi, Hanmo Wang, Mingyu Fan, Wenjian Wang, and Yi-Dong Shen, "Robust multiple kernel k-means using $\ell_{2,1}$-norm," in *IJCAI*, 2015, pp. 3476–3482.

[13] Yaqiang Yao, Yang Li, Bingbing Jiang, and Huanhuan Chen, "Multiple kernel k-means clustering by selecting representative kernels," *IEEE Trans. Neural Netw. Learn. Syst.*, 2020.

[14] Jin Huang, Feiping Nie, and Heng Huang, "Spectral rotation versus k-means in spectral clustering," in *AAAI*, 2013.

[15] Xiaojun Chen, Feiping Nie, Joshua Zhexue Huang, and Min Yang, "Scalable normalized cut with improved spectral rotation.," in *IJCAI*, 2017, pp. 1518–1524.

[16] Yanwei Pang, Jin Xie, Feiping Nie, and Xuelong Li, "Spectral clustering by joint spectral embedding and spectral rotation," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 247–258, 2018.

[17] Feiping Nie, Rui Zhang, and Xuelong Li, "A generalized power iteration method for solving quadratic problem on the stiefel manifold," *Science China Information Sciences*, vol. 60, no. 11, pp. 112101, 2017.

[18] Jiyuan Liu, Xinwang Liu, Siwei Wang, Sihang Zhou, and Yuexiang Yang, "Hierarchical multiple kernel clustering," in *AAAI*, 2021, pp. 8671–8679.

[19] Xinwang Liu, Xinzhong Zhu, Miaomiao Li, Lei Wang, En Zhu, Tongliang Liu, Marius Kloft, Dinggang Shen, Jianping Yin, and Wen Gao, "Multiple kernel k-means with incomplete kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1191–1204, 2019.