**Fusion Multiple Kernel K-means**

**Yi Zhang**1 **, Xinwang Liu**\*1**, Jiyuan Liu**1 **, Sisi Dai**1 **, Changwang Zhang**2 **, Kai Xu**1 **, En Zhu**1

1 School of Computer, National University of Defense Technology, Changsha, China, 410073

2 CCF Theoretical Computer Science Technical Committee, Shenzhen, China, 518064 {zhangy, xinwangliu}@nudt.edu.cn

**Abstract**

Multiple kernel clustering aims to seek an appropriate com- bination of base kernels to mine inherent non-linear informa- tion for optimal clustering. Late fusion algorithms generate base partitions independently and integrate them in the fol- lowing clustering procedure, improving the overall efﬁciency. However, the separate base partition generation leads to inad- equate negotiation with the clustering procedure and a great loss of beneﬁcial information in corresponding kernel ma- trices, which negatively affects the clustering performance. To address this issue, we propose a novel algorithm, termed as *Fusion Multiple Kernel* k*-means* (FMKKM), which uni- ﬁes base partition learning and late fusion clustering into one single objective function, and adopts early fusion tech- nique to capture more sufﬁcient information in kernel matri- ces. Speciﬁcally, the early fusion helps base partitions keep more beneﬁcial kernel details, and the base partitions learn- ing further guides the generation of consensus partition in the late fusion stage, while the late fusion provides positive feedback on two former procedures. The close collaboration of three procedures results in a promising performance im- provement. Subsequently, an alternate optimization method with promising convergence is developed to solve theresul- tant optimization problem. Comprehensive experimental re- sults demonstrate that our proposed algorithm achieves state- of-the-art performance on multiple public datasets, validating its effectiveness. The code of this work is publicly available at <https://github.com/ethan-yizhang/Fusion-Multiple-Kernel>- K-means.

**Introduction**

Multiple kernel clustering (MKC) aims to extract comple- mentary information from multiple pre-speciﬁed kernels and then categorize the data with close patterns or structures into the same cluster (Zhao, Kwok, and Zhang 2009; Kloft,

Rckert, and Bartlett 2010; Kloft et al. 2011; Yu et al. 2011;

Huang, Chuang, and Chen 2011; Gnen and Alpayd1n 2011; Zhou et al. 2015; Han et al. 2016; Wang et al. 2017b; Zhou et al. 2021a; Liu et al. 2021a). Due to the ability to min- ing inherent non-linear information, MKC has been inten- sively researched and commonly applied to various applica- tions (Liu et al. 2016; Li et al. 2017; Liu et al. 2017; Bhadra,

\* Corresponding Author

Copyright © 2022, Association for the Advancement of Artiﬁcial Intelligence (www.aaai.org). All rights reserved.

Kaski, and Rousu 2017; Liu et al. 2019; Zhou et al. 2019, 2021b).

For example, Patel and Zhou et al. take the advantage of MKC and extend it with the properties of subspace (Patel and Vidal 2014; Zhou et al. 2019). Ren et al. improve the clustering performance of MKC by preserving the global and local graph structures (Ren and Sun 2020). Liu et al. assumes that an optimal kernel is in the neighborhood of the mixed kernel to reinforce the presentation of the op- timal kernel (Liu et al. 2017). Also, multiple kernel k- means (MKKM) is a popular method, which has been inten- sively studied and widely applied. It simultaneously learns the clustering partition matrix and the base kernel coefﬁ- cient in a single objective function (Huang, Chuang, and Chen 2011). Many variants of MKKM has been proposed to improve the clustering performance (Gnen and Margolin 2014; Du et al. 2015; Liu et al. 2016; Wang et al. 2017a; Bang, Yu, and Wu 2018; Liu et al. 2019; Yao et al. 2020; Liu,Zhu, and Liu 2020). For example, the research (Gnen and Margolin 2014) improves the representation ability of MKKM through generating locally adaptive mixed kernels so as to mine the sample features in depth. By considering the relationship among kernels, Liu et al. propose MKKM with matrix-induced regularization to reduce the redundancy while enhancing the diversity of selected kernels (Liu et al. 2016).

Late fusion algorithm, as a typical MKC approach, sub- stantially reduces the computational cost while achieves en- couraging clustering performance by maximally aligning weighted base partitions with the consensus one (Wang et al. 2019; Liu et al. 2018, 2020; Zhang et al. 2020; Kang et al. 2020; Liu et al. 2021b). For an instance, Liu et al. develop a late fusion method to handle incomplete data by jointly optimizing the imputation and clustering tasks (Liu et al. 2020). Liu et al. propose to directly learn the clustering la- bels and avoid the two-stage learning procedure, further im- proving the computational efﬁciency and clustering perfor- mance (Liu et al. 2021b) .

Although clustering algorithm with late fusion manner shows the above advantages, we observe that base parti- tion generation by eigenvalue decomposition is regarded as a pre-process, which may cause a signiﬁcant loss of the cru- cial information in corresponding kernel matrices. Further- more, since it is performed separately from late fusion clus-

tering, the two independent procedures cannot guide each other and lack necessary negotiation. Thus, the initial qual- ity of base partitions becomes the bottleneck, limiting the clustering performance.

To address the above issues, this paper proposes a novel multiple kernel clustering algorithm, termed as *Fusion Mul- tiple Kernel* k*-means* (FMKKM). It uniﬁes the base parti- tion learning and late fusion clustering into a single opti- mization objective function and incorporates the early fu- sion technique to capture more sufﬁcient information for the downstream clustering. Speciﬁcally, these three procedures, the base partitions learning, the early fusion, and the late fusion, are uniﬁed into one optimization objective. This en- ables them to negotiate with each other so as to achieve opti- mal clustering performance. Afterwards a six-step alternat- ing optimization method with guaranteed convergence is de- veloped to effectively solve the resultant optimization prob- lem.

The main contributions of this paper are summarized as follows,

• This paper, for the ﬁrst time, integrates the base par- tition learning and late fusion clustering into a uniﬁed optimization objective. In this way, they can positively guide each other and avoid the bottleneck caused by not ”ideal” initial partitions, leading to better clustering per- formance.

• This paper, for the ﬁrst time, proposes a novel overall process fusion manner, which uniﬁes the early fusion and the late fusion. This allows the algorithm to bring their respective advantages and strengths into play, thus fully excavating the beneﬁcial information in kernel matrices.

• We introduce a curvilinear search algorithm and develop it to solve the difﬁcult optimization problem with orthog- onality constraint. Then we carefully design a six-step al- ternate optimization method and discuss its convergence, computational complexity, and extension.

• Comprehensive experiments are conducted on multiple benchmark datasets to evaluate the effectiveness of our proposed algorithm. As demonstrated, FMKKM dramat- ically outperforms state-of-the-art competitors, validat- ingits effectiveness.

**Related Work**

In this section, we provide a brief review of MKKM and LFMVC, and then introduce the motivation of our work.

**Multiple Kernel K-means**

Given **X** ∈ Rn×d, k-means clustering aims to group **X** into k clusters, where n is the number of samples and d denotes feature dimensions. Let **Z** ∈ {0, 1}n×k be a clustering as- signment, where Ziq = 1 if sample **x**i belongs to the q-th cluster. Its objective can be presented as

 Σ Σ=1 Ziq Ⅱ**x**i — **c**q Ⅱ2 s.t. **Z1** = **1**. (1)

,i

The samples can be mapped into a reproducing kernel Hilbert space (RKHS) (Scholkopf and Smola 2001) by ker- nel tricks and then taken as the input of k-means to deal

with non-linear features. Note that, the kernel matrices can

be ·constructed as Ki,j = φφj with a mapping function

ϕ( ). Based on this, the objective can be rewritten as

m**H**in Tr (**K** (**I** — **HH**T )) s.t. **H**T**H** = **I**, (2)

where **H** ∈ Rn×k denotes clustering partition matrix.

Due to the fact that the choice of kernel matrix severely inﬁuences the performance of kernel k-means, we usually assume the optimal kernel **K** can be written as a combina- tion of base kernel matrices. The objective can be extended as follows,

,in Tr(**K** (**I** — **HH**T )) s.t. **H**T**H** = **I**, **?** ∈ ∇ 1 , (3)

where ∇ 1 = {**?** ∈ Rm | Σ √p = 1, √p ≥ 0, ∀p},

**K** = Σ √**K**p and m represents the number of data

views. In literature (Huang, Chuang, and Chen 2011),**?** and **H** can be jointly solved by an alternate optimization. Then a common k-means clustering algorithm is applied to the obtained partition matrix **H** for the ﬁnal cluster assignments.

**Late Fusion Multi-view Clustering**

Late fusion algorithm substantially reduces the computa- tional cost and achieves encouraging clustering performance by maximally aligning weighted base partitions with the consensus partition (Wang et al. 2019). Given n samplesink clusters with m kernels, its objective function can be math- ematically presented as follows,

max**H**\* , **W** , **β** Tr(**H**\*T Σ βp**H**p**W**p ) + λTr(**H**\*T )

s.t. **H**\*T**H**\* = **I**k , **WW**p = **I**k , ∀p, **β** ∈ ∇2 ,

(4)

where ∇2 = {**β** ∈ Rm | Σ β = 1, βp ≥ 0, ∀p}, **H**p

denotes the p-th pre-calculated base partitions from multiple kernels, and **W**p is the p-th linear transformation matrix.

The latter Tr(**H**\*T ) is a regularization term on the con-

sensus partition to prevent **H**\* from being too far away from

prior average partition , and λ is a trade-off parameter.

According to the literature (Wang et al. 2019), a three-step alternate optimization is developed to solve Eq. (4) and the computational complexity of LFMVC is O(nk2 + mk3) per iteration. After obtaining the consensus partition matrix **H**, a a common k-means clustering algorithm is applied to obtain the ﬁnal cluster assignments.

In existing late fusion clustering algorithms (Wang et al. 2019; Liu et al. 2020, 2021b), base partitions used for clus- tering are pre-calculated and remain unchanged. They are independent of the ﬁnal learning of consensus partition and calculated separately without negotiation. As a result, unsat- isfying base partitions would directly lead to poor cluster- ing results. In other words, the pre-calculated base partitions limit the performance of the whole clustering algorithm and become a performance bottleneck. In the following part of the paper, we develop FMKKM to deal with this problem.

**Fusion Multiple Kernel K-means**

In this section, we ﬁrstly give out the objective formulation of our proposed FMKKM and then develop a six-step alter- nate optimization algorithm. Next, the convergence, compu- tational complexity, and extension of our proposed FMKKM are discussed.

**Proposed Formulation**

Eq.(4), a widely used formula for reducing the computa- tional complexity of MKC, is to align the weighted base par- titions with the consensus partition instead of fusing the base kernel matrices. It performs fusion in the partition layer, so it is called late fusion multi-view clustering, which has shown its feasibility and efﬁciency in many ﬁelds. Despite achiev- ing encouraging clustering performance, we observe that it has two disadvantages: 1) The base partition generation is a pre-process, while late fusion clustering is separately per- formed and lacks negotiation, resulting in that the cluster- ing performance is directly determined by the initial quality of base partitions. 2) Base partitions can be regarded as the low-dimensional data features extracted by eigenvalue de- composition of the corresponding kernel matrices, thus this single feature extraction process will cause the omission of some crucial information in kernels.

To overcome these shortcomings, we propose a novel fu- sion multiple kernel k-means algorithm, which uniﬁes the base partitions learning and late fusion clustering into a single optimization objective function and incorporates the early fusion of kernel matrices to capture more sufﬁcient and beneﬁcial information. Then we derive the objective formu- lation of our proposed FMKKM as follows.

min**H**\* , {**H**p } , {**W**p } , **α** , **β** , Tr (**Kα** (**I** - **H**\* **H**\*T ))

+ λ1 Σ βTr (**K**p (**I** - **H**p**H**))- λ2 Tr (**H**\*T**B** )

s.t. **H**\*T**H**\* = **I**, **HH**p = **I**, **WW**p = **I**, ∀p,

**α** , **β** ∈ ∇ 1, **?** ∈ ∇2 ,

(5)

where **B** = Σ √p**H**p**W**p , **H**\* ∈ Rn×k and **H**p ∈ Rn×k denote the consensus partition and the p-th base par- tition, **W**p ∈ Rk×k is the p-th transformation matrix, and n,m,k represent the numbers of samples, kernels, and clus- ters respectively. As seen from Eq.(5), our proposed algo- rithm integrates the early fusion and the late fusion, forming a new paradigm for further clustering. In this way, their re- spective advantages and strengths can be brought into play. Moreover, the base partitions are learnable rather than ﬁxed in LFMVC. As a result, the fusion of the partition layer and the base partition learning can negotiate with each other to obtain the optimal clustering performance.

**Alternate Optimization**

There are six variables in Eq.(5) to optimize, therefore, jointly optimizing them is difﬁcult. In order to solve the optimization, we design a six-step alternate algorithm with proved convergence. In each step, one variable is optimized while the others are ﬁxed.

**Optimization H**\* Fixing {**H**p  and **?**, the optimization in Eq. (5) w.r.t **H**\* is transformed

to

min**H**\* - Tr(**H**\*T**Kα H**\* + λ2 **H**\*T**B** ) s.t. **H**\*T**H**\* = **I**,

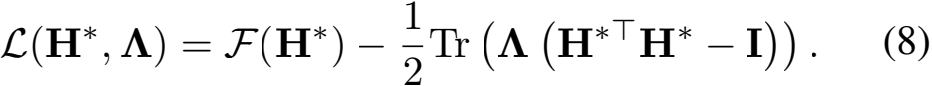
(6)

where **B** = Σ √p**H**p**W**p. Due to the orthogonality

constraint of **H**\* , we develop a curvilinear search algo- rithm to optimize **H**\* according to the literature (Wen and Yin 2013). We ﬁrst deﬁne F(**H**\* ) := -Tr(**H**\*T**Kα H**\* + λ2 **H**\*T**B** ) and rewrite the optimization in Eq.(6) as

min**H**\* F(**H**\* ) s.t. **H**\*T**H**\* = **I**. (7)

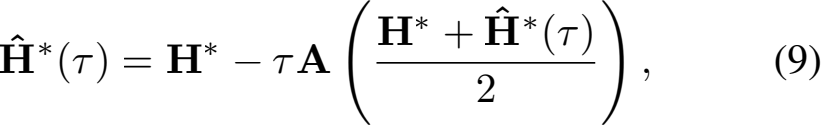
Since **H**\*T**H**\* is a symmetric matrix, the Lagrangian multiplier **Λ** corresponding to **H**\*T**H**\* = **I** is also sym- metric. The Lagrangian function of Eq.(7) is



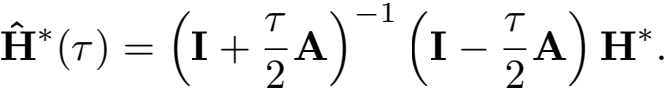
Suppose that **H**\* is a local minimizer of Eq.(7),then **H**\* satisﬁes D**H**\* L(**H**\* , **Λ**) = L(**H**\* , **Λ**) = **G** - **H**\* **G**T**H**\* and **H**\*T**H**\* = **I** with the associated Lagrangian multiplier **Λ** = **G**T**H**\* , where **G** is the derivative ofF(**H**\* ) w.r.t **H**\* .

We deﬁne **A** := **GH**\*T - **H**\* **G**T and the differential operator ∇L := **AH**\* = L(**H**\* , **Λ**) = **G** - **H**\* **G**T**H**\* . Note that ∇L = 0 if and only if **A** = 0.

To preserve the orthogonality constraint, the next iteration of **H**\* can be calculated as follows.

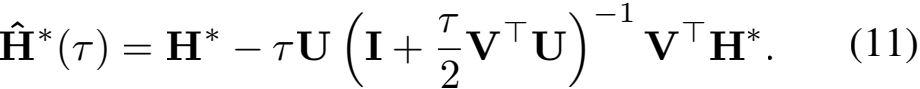


where τ is a step size. From Eq.(9), we can easily obtain

 (10)

**Theorem 1** *The* \* (τ ) *computed by Eq.(10) satisﬁes* \* (τ )T \* (τ ) = **H**\*T**H**\* = **I** *for any skew-symmetric* **A** *and* τ ∈ R*.*

Because the matrix inversion in Eq.(10) seems computa- tionally expensive, it is not a good option to directly com- pute. We suppose **U** = [**G**, **H**\*] and **V** = [**H**\* , **G**], then **A** = **GH**\*T -**H**\* **G**T can be rewritten as **A** = **UV**T . After that, we apply the SMW formula to **I** **A** = **I** **UV**T and obtain (**I** **A**) —1 = **I** - **U** (**I** **V**T**U**) —1 **V**T . Finally we have the expression of next iteration of **H**\* with cheaper cost as follows.



Note that, τ can be selected by a one-dimensional line search strategy, such as Armijo-Wolfe’s rule. Obtaining **V**T**U** = **A**T = (**GH**\*T - **H**\* **G**T )T needs 2nk2 ﬁops and the inversion takes O(k3 ) due to turning inverting Rn ×n to inverting R2k×2k. The ﬁnal computational complexity of solving **H**\* is 4nk2 + O(k3 ) per iteration. The curvilinear search algorithm procedure to solve Eq.(6) is outlined in Al- gorithm 1.

|  |
| --- |
| Algorithm 1: Solving **H** with orthogonality constraint via curvilinear search algorithm |
| 1: **Input: H**, F(**H**) and ∈ . |
| 2: **Output: H**. |
| 3: Initialize t = 0 and **G** = DF(**H**). |
| 4: **repeat** |
| 5: **U** ← [**G**, **H**] and **V** ← [**H**, **G**]. |
| 6: Select τt according to the Armijo-Wolfe’s rule. |
| 7: **H**t+1 ← **H** - τ**U** (**I** **V**T**U**)-1 **V**T**H**. |
| 8: t ← t + 1. |
| 9: **until** ||∇Lt || ≤ ∈ |

**Optimization** {**H**p  and **?**, the optimization in Eq. (5) w.r.t each **Hp** can be rewritten as,

min**H**p -Tr(λ1 β**HK**p**H**p + λ2γp**H**\*T**H**p**W**p ) s.t. **HH**p = **I**.

(12)

Like **H**\* , the optimization of **H**p can be solved by Algo- rithm 1 with computational complexity 4nk2 + O(k3 ) per iteration.

**Optimization** {**W**p  and**?**, the optimization in Eq. (5) w.r.t each **Wp** is reduced

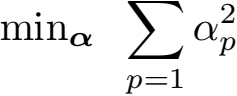
to

max**W**p Tr(**WHH**\* ) s.t. **WW**p = **I**. (13)

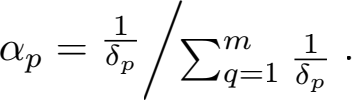
Eq. (13) can be efﬁciently solved by SVD with computa- tional complexity O(nk2 ).

**Optimization α** Fixing **H**\* , {**H**p  and **?**, the optimization in Eq. (5) is equivalent to the op- timization problem as follows

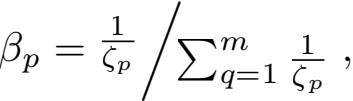
m

δp s.t. **α** ∈ ∇ 1 , (14)

where δp = Tr(**K**p (**I** - **H**\* **H**\*T )) and the computational complexity of calculating δp is O(n2 k). This could be easily solved as follows

 (15)

**Optimization β** Fixing **H**\* , {**H**p  and**?**, it could be solved just like optimizing **α** and the op- timal solution is

 (16)

where ζp = Tr(**K**p (**I** - **H**p**H**)).

**Optimization ?** Fixing **H**\* , {**H**p  and **β**, the optimization in Eq. (5) is equivalently rewritten as follows

m

max**吖** γp θp s.t. **?** ∈ ∇2 , (17)

Algorithm 2: Fusion Multiple Kernel K-means

1: **Input:** {**H**p 1 , λ2 and ∈ .

2: Initialize **α** = **1**/m, **β** = **1**/m, **?** = **1**/√m, **H**\* , {**H**p  , {**W**p  t = 0.

3: **repeat**

4: Update **H**\* with ﬁxed {**H**p  and**?** via Algorithm 1.

5: Update {**H**p  **H**\* , {**W**p  , **α** , **β** and**?** via Algorithm 1.

6: Update {**W**p  **H**\* , {**H**p  , **α** , **β** and**?**by optimizing Eq. (13).

7: Update **α** with ﬁxed **H**\* , {**H**p  and**?**by solving Eq. (15).

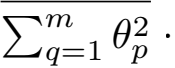
8: Update **β** with ﬁxed **H**\* , {**H**p  and**?**by solving Eq. (16).

9: Update **?** with ﬁxed **H**\* , {**H**p  and **β** by solving Eq. (18).

10: t ← t + 1.

11: **until** (obj(t-1) - obj(t))/obj(t) ≤ ∈

where θp = Tr(**H**\*T**H**p**W**p ). We can easily obtain the op- timal closed-form solution as follows

γp = θp /√ (18)

The whole algorithm optimizing Eq. (5) is outlined in Al- gorithm 2, where obj(t) denotes the objective loss value at the t-th iteration.

**Discussion and Extension**

**Convergence** Note that the objective loss value in Eq.(2) is monotonically decreased when one variable is optimized with the others ﬁxed and the objective function is lower- bounded. Therefore, the whole optimization algorithm is proved to converge to a local optimum, as validated by our experimental results in Figure (1).

**Computational Complexity** As seen from the optimiza- tion procedure in Algorithm 2, the computational complex- ityof our proposed FMKKM at each iteration is O(mn2 k + tmnk2 + tmk3 ), where n, mand k represent the numbers of samples, kernels, and clusters, respectively, and t denotes the maximum number of iterations in the optimization of **H**\* and {**H**p  .

**Extension** Firstly, our proposed FMKKM adopts the sim- plest MKKM to guide the learning of **H**\* and {**H**p  . Therefore,other similarity-based methods can be used to ex- tend this work to further improve clustering performance. Secondly, our proposed FMKKM integrates two fusion stages together, forming a novel overall process fusion paradigm, thus they can give full play to theirrespective ad- vantages and make up for their shortcomings. This idea can be easily extended to various ﬁelds, not only MKC, but also other machine learning tasks. Moreover, the novel overall process fusion paradigm achieves great success on cluster- ing performance that may inspire more research on fusion algorithms.

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | #Samples | #Kernels | #Clusters |
| Texas | 187 | 2 | 5 |
| Wisconsin | 265 | 2 | 5 |
| Football | 248 | 9 | 20 |
| BBCSport | 544 | 2 | 5 |
| Willow | 911 | 3 | 7 |
| Flower17 | 1360 | 7 | 17 |
| Flower102 | 8189 | 4 | 102 |
| ALOI-100 | 10800 | 4 | 100 |
| Reuters | 18758 | 5 | 6 |

Table 1: Datasets used in our experiments.

**Experiment and Analysis**

In this section, we carry out a comprehensive experiment on multiple benchmark datasets in order to evaluate the effec- tiveness of our proposed FMKKM. The clustering perfor- mance, evolution of the objective value and the learned **H**\* , parameter sensitivity, weight coefﬁcients, and running time are carefully analyzed.

**Experiment Settings**

Multiple public datasets are adopted to evaluate the per- formance of our proposed FMKKM, including *Texas*1 , *Wisconsin*1 , *Football*2 , *BBCSport*3 , *Willow*4 , *Flower17*5 , *Flower102*5 , *ALOI-100*6 , *Reuters*7 . The detail information of datasets is summarized in Table 1. It can be observed that the numbers of samples vary from hundreds to nearly 20 thousand,the numbers of kernels and clusters also show con- siderable variation, which enables the experiment to better evaluate the performance of different clustering algorithms.

For all datasets, the true number of clusters k is prespec- iﬁed and set as the input of algorithms. We apply three widely used criteria to evaluate the clustering performance, i.e. clustering accuracy (ACC), normalized mutual informa- tion (NMI), and rand index (RI). For all algorithms, we re- peat each experiment 50 times with random initialization to reduce the randomness effect caused by k-means, and report their means and standard variations. All experiments are per- formed on a PC with Intel Core i9-10900X CPU and 64G RAM.

Along with our proposed FMKKM, we run another ten algorithms chosen from recent MKC literature for compari- son. Speciﬁcally, **Avg-KKM** (*baseline*) obtains the consen- sus kernel by uniformly combines base kernels and then per- forms kernel k-means on it. We also select ﬁve classical algorithms, including **MKKM** (Huang, Chuang, and Chen 2011), **LMKKM** (Gnen and Margolin 2014), **MKKM- MR** (Liu et al. 2016), **LKAM** (Li et al. 2016) and **ONKC** (Liu et al. 2017). Additionally, we choose four most recent methods,i.e. **LFMVC** (Wang et al. 2019), **SMKKM** (Liu, Zhu, and Liu 2020), **NKSS** (Zhou et al. 2019) and **SPMKC**

1<https://linqs-data.soe.ucsc.edu/public/lbc/> 2<http://mlg.ucd.ie/aggregation/>

3<http://mlg.ucd.ie/datasets/>

4<http://www.di.ens.fr/willow/research/stillactions/> 5<https://www.robots.ox.ac.uk/>~ vgg/data/ﬁowers/

6http://elki.dbs.iﬁ.lmu.de/wiki/DataSets/MultiView/ 7<https://kdd.ics.uci.edu/databases/reuters21578/>

(Ren and Sun 2020). Their source codes are publicly avail- able and we directly use them without revision.

**Experiment Results**

**Overall Clustering Performance** Table 2 presents the ACC, NMI and RI comparison of the above algorithms. From this table, we have the following observations:

• Late fusion MVC (Wang et al. 2019), as a recent rep- resentation, does signiﬁcantly improve most early-fusion MKC algorithm in term of clustering performance and complexity, yet its aforementioned shortcomings lead to poor performance in many situations. For example, other algorithms exceeds LFMVC by 4.9%, 2.9%, 0.8%, 3.3%, 0.2% and 0.3% on Texas, Wisconsin, Willow, Flower102, ALOI-100, Reuters datasets in term of ACC. Meanwhile, our proposed FMKKM dramatically im- proves the clustering performance and exceeds LFMVC by 12.6%, 2.7%, 7.2%, 8.7%, 2.0%, 1.0%, 4.6%, 4.0% and 0.6% on the nine datasets in term of ACC.

• Besides, the recently proposed SMKKM (Liu, Zhu, and Liu 2020) extends the widely used supervised kernel alignment criterion and achieves comparable or bet- ter clustering performance, however its improvement of performance is marginal. Meanwhile, our proposed FMKKM outperforms SMKKM signiﬁcantly. For exam- ple, FMKKM exceeds it by 20%, 12.2%, 7.1%, 21.7%, 5.9%, 3.4%, 1.6%, 7.3% and 0.8% on nine benchmark datasets in term of ACC.

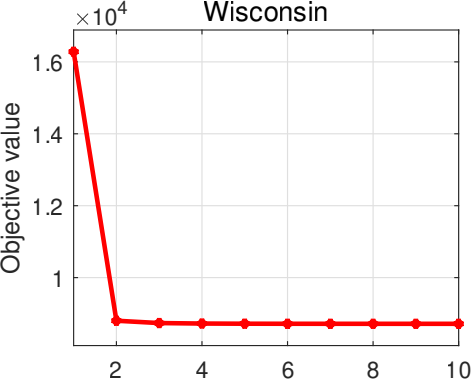
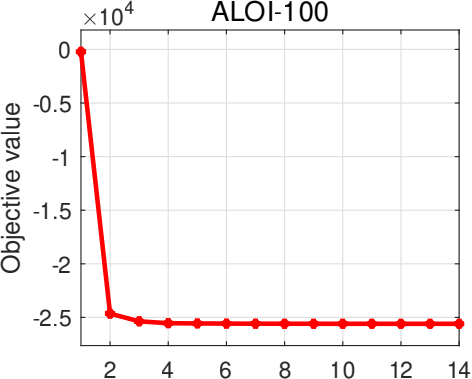
• In recent works, NKSS (Zhou et al. 2019) adopts the neighbour kernel and subspace technique, yet SPMKC (Ren and Sun 2020) attempts preserving the global and local graph structures. Although they have achieved encouraging clustering performance, our pro- posed FMKKM outperforms these recent work. Speciﬁ- cally, FMKKM exceeds NKSS and SPMKC by 18.2%, 31.8%, 5.4%, 21.2%, 2.8%, 19.4%, 1.3%, 7.0% and 19.6%, 25.7%, 19.7%, 34%, 1.9%, 32.9%, 17.4%, 20.2%, 19.5% in term of ACC on all datasets, respec- tively.

In summary, FMKKM shows superior clustering perfor- mance on all datasets compared with other algorithms, val- idating the effectiveness of the proposed overall process fu- sion manner. We expect that its novel paradigm and supe- rior performance will attract intensive research and common application in community. In addition, we point out that ’- ’ in Table 2 indicates that the results are unavailable due to out-of-memory error, non convergence or too long execution time, which is caused by cubic computational and memory complexity.

**Convergence and Evolution** As discussed above, our proposed FMKKM is theoretically guaranteed to converge into a local optimum. To show this point in practice, we plot the objective value curves of FMKKM w.r.t. the number of iterations on Wisconsin and ALOI-100 datasets, as shown in Figure 1. It can be seen that the objective value mono- tonically decreases and the algorithm quickly converges.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algrithmn | Texas | Football | Wisconsin | BBCSport | Willow | Flower17 | Flower102 | ALOI-100 | Reuters |
| ACC | | | | | | | | | |
| Avg-KKM | 46.7 ± 1.2 | 72.4 ± 2.4 | 52.9 ± 0.4 | 63.5 ± 1.5 | 22.2 ± 0.3 | 51.2 ± 2.2 | 26.6 ± 0.6 | 64.8 ± 1.3 | 45.5 ± 1.5 |
| MKKM | 58.9 ± 1.4 | 74.3 ± 1.8 | 54.1 ± 2.8 | 63.8 ± 1.6 | 22.2 ± 0.4 | 44.7 ± 1.4 | 22.5 ± 0.5 | 6.4 ± 0.1 | 45.4 ± 1.5 |
| LMKKM | 51.0 ± 1.7 | 52.0 ± 2.2 | 46.0 ± 0.7 | 64.0 ± 1.3 | 22.5 ± 0.2 | 37.7 ± 2.1 | - | - | - |
| ONKC | 54.9 ± 0.5 | 77.7 ± 3.3 | 56.5 ± 0.3 | 63.6 ± 1.4 | 22.5 ± 0.5 | 54.1 ± 1.6 | 39.5 ± 1.0 | 68.0 ± 1.2 | 41.8 ± 1.2 |
| MKKM-MR | 54.4 ± 0.4 | 77.3 ± 2.2 | 55.8 ± 0.6 | 63.6 ± 1.4 | 22.7 ± 0.4 | 58.3 ± 1.3 | 40.2 ± 0.9 | 68.3 ± 1.0 | 46.2 ± 1.4 |
| LKAM | 51.8 ± 0.2 | 76.8 ± 2.3 | 56.7 ± 0.2 | 73.7 ± 0.5 | 27.0 ± 0.2 | 49.8 ± 1.2 | 41.2 ± 0.9 | 64.2 ± 0.7 | 45.5 ± 0.0 |
| LFMVC | 54.0 ± 0.9 | 78.4 ± 2.6 | 53.6 ± 0.5 | 76.6 ± 2.8 | 26.2 ± 0.5 | 61.3 ± 1.0 | 38.4 ± 1.1 | 68.1 ± 1.0 | 45.7 ± 1.6 |
| RMKKM | 46.4 ± 1.0 | 72.4 ± 1.9 | 53.7 ± 0.6 | 63.6 ± 1.5 | 22.2 ± 0.4 | 52.9 ± 2.1 | 30.6 ± 1.0 | - | 45.5 ± 1.5 |
| SMKKM | 46.6 ± 1.2 | 68.9 ± 2.6 | 53.7 ± 0.6 | 63.6 ± 1.4 | 22.3 ± 0.6 | 58.9 ± 1.0 | 41.4 ± 1.2 | 64.8 ± 1.3 | 45.5 ± 0.7 |
| NKSS | 48.4 ± 0.7 | 49.3 ± 2.0 | 55.4 ± 0.1 | 64.1 ± 1.2 | 25.4 ± 0.4 | 42.9 ± 1.0 | 41.7 ± 0.8 | 65.1 ± 1.2 | - |
| SPMKC | 47.0 ± 0.6 | 55.4 ± 2.5 | 41.1 ± 1.0 | 51.3 ± 1.9 | 26.3 ± 0.2 | 29.4 ± 0.9 | 25.6 ± 0.4 | 51.9 ± 1.5 | 26.8 ± 0.0 |
| FMKKM | **66.6** ± **1.0** | **81.1** ± **2.7** | **60.8** ± **1.1** | **85.3** ± **0.3** | **28.2** ± **0.4** | **62.3** ± **1.2** | **43.0** ± **1.4** | **72.1** ± **1.3** | **46.3** ± **2.4** |
| NMI | | | | | | | | | |
| Avg-KKM | 30.2 ± 0.6 | 77.9 ± 1.4 | 34.2 ± 0.8 | 43.7 ± 1.2 | 5.7 ± 0.2 | 49.7 ± 1.6 | 45.9 ± 0.4 | 77.6 ± 0.6 | 27.4 ± 0.4 |
| MKKM | 11.6 ± 0.6 | 79.1 ± 0.8 | 28.2 ± 2.3 | 44.0 ± 1.3 | 5.7 ± 0.1 | 44.5 ± 1.2 | 42.7 ± 0.2 | 22.3 ± 0.2 | 27.3 ± 0.4 |
| LMKKM | 13.5 ± 1.4 | 59.6 ± 1.5 | 15.1 ± 1.5 | 44.0 ± 0.9 | 5.7 ± 0.2 | 38.9 ± 1.5 | - | - | - |
| ONKC | 31.6 ± 1.0 | 80.4 ± 1.8 | 31.9 ± 0.2 | 43.9 ± 0.7 | 6.0 ± 0.4 | 52.7 ± 0.7 | 56.1 ± 0.4 | 79.7 ± 0.5 | 22.3 ± 0.4 |
| MKKM-MR | 31.4 ± 0.5 | 79.9 ± 1.5 | 32.7 ± 0.3 | 43.7 ± 1.1 | 6.2 ± 0.3 | 56.6 ± 0.7 | 56.7 ± 0.4 | 80.7 ± 0.4 | 25.3 ± 0.7 |
| LKAM | 29.2 ± 0.5 | 79.6 ± 1.3 | 36.2 ± 0.1 | 65.3 ± 1.1 | **8.3** ± **0.3** | 49.6 ± 0.5 | 56.8 ± 0.4 | 77.8 ± 0.3 | **29.9** ± **0.0** |
| LFMVC | 28.4 ± 0.8 | 83.0 ± 2.3 | 32.4 ± 0.6 | 59.1 ± 2.8 | 7.7 ± 0.5 | 59.1 ± 0.5 | 54.9 ± 0.5 | 79.5 ± 0.4 | 27.4 ± 0.4 |
| RMKKM | 30.4 ± 0.3 | 77.7 ± 1.0 | 31.1 ± 0.6 | 43.8 ± 1.2 | 5.7 ± 0.2 | 51.9 ± 0.9 | 48.8 ± 0.5 | - | 27.4 ± 0.4 |
| SMKKM | 27.4 ± 1.4 | 75.8 ± 1.6 | 31.2 ± 0.6 | 44.0 ± 0.9 | 5.8 ± 0.4 | 57.2 ± 0.7 | 58.2 ± 0.5 | 77.6 ± 0.6 | 27.7 ± 0.2 |
| NKSS | 19.7 ± 0.5 | 57.7 ± 1.1 | 32.8 ± 0.1 | 51.1 ± 0.4 | 5.8 ± 0.3 | 46.0 ± 0.5 | 58.6 ± 0.2 | 78.4 ± 0.5 | - |
| SPMKC | 10.2 ± 1.0 | 57.8 ± 1.1 | 4.0 ± 0.8 | 29.9 ± 3.1 | 7.1 ± 0.1 | 27.5 ± 0.4 | 42.3 ± 0.2 | 69.4 ± 1.0 | 0.6 ± 0.0 |
| FMKKM | **32.9** ± **2.1** | **85.0** ± **1.8** | **39.5** ± **0.5** | **69.2** ± **0.5** | 8.2 ± 0.2 | **59.7** ± **0.9** | **58.3** ± **0.5** | **82.2** ± **0.5** | 27.6 ± 0.4 |
| RI | | | | | | | | | |
| Avg-KKM | 20.4 ± 0.6 | 59.8 ± 2.6 | 28.1 ± 0.7 | 39.6 ± 2.0 | 3.1 ± 0.1 | 32.4 ± 1.8 | 15.1 ± 0.5 | 50.9 ± 1.5 | 21.8 ± 1.4 |
| MKKM | 14.1 ± 0.6 | 61.0 ± 1.3 | 24.8 ± 2.2 | 40.0 ± 2.1 | 3.2 ± 0.1 | 27.6 ± 1.3 | 12.0 ± 0.4 | 1.9 ± 0.1 | 21.8 ± 1.4 |
| LMKKM | 11.8 ± 1.0 | 31.9 ± 2.6 | 7.1 ± 0.5 | 40.4 ± 1.4 | 3.2 ± 0.2 | 20.7 ± 1.5 | - | - | - |
| ONKC | 24.0 ± 0.9 | 65.1 ± 3.3 | 26.9 ± 0.4 | 39.9 ± 1.4 | 3.2 ± 0.2 | 35.4 ± 0.9 | 25.0 ± 0.6 | 55.2 ± 1.3 | 20.3 ± 0.3 |
| MKKM-MR | 23.5 ± 0.7 | 64.0 ± 2.3 | 26.6 ± 0.4 | 39.7 ± 1.9 | 3.4 ± 0.2 | 40.2 ± 0.9 | 25.6 ± 0.7 | 56.0 ± 1.5 | 23.1 ± 0.6 |
| LKAM | 21.6 ± 0.2 | 63.6 ± 2.0 | 31.8 ± 0.3 | 62.2 ± 1.2 | 4.6 ± 0.2 | 31.2 ± 0.9 | 27.2 ± 0.7 | 52.8 ± 0.7 | **24.1** ± **0.0** |
| LFMVC | 22.1 ± 1.0 | 66.8 ± 3.7 | 27.9 ± 0.8 | 57.5 ± 3.8 | 4.5 ± 0.3 | 44.3 ± 0.7 | 25.4 ± 1.1 | 53.8 ± 1.0 | 22.1 ± 1.6 |
| RMKKM | 20.7 ± 0.5 | 59.4 ± 1.9 | 23.6 ± 0.7 | 39.8 ± 2.0 | 3.1 ± 0.1 | 34.8 ± 1.3 | 18.1 ± 0.8 | - | 21.8 ± 1.4 |
| SMKKM | 18.1 ± 1.4 | 56.1 ± 2.9 | 23.6 ± 0.7 | 40.1 ± 1.6 | 3.2 ± 0.2 | 40.9 ± 1.1 | 27.5 ± 0.9 | 51.0 ± 1.5 | 22.1 ± 0.8 |
| NKSS | 17.0 ± 0.6 | 30.2 ± 1.5 | 29.3 ± 0.1 | 44.2 ± 0.6 | 3.4 ± 0.3 | 24.1 ± 0.8 | 27.5 ± 0.5 | 54.3 ± 1.3 | - |
| SPMKC | 7.8 ± 2.0 | 26.7 ± 1.7 | -0.8 ± 0.6 | 21.8 ± 3.5 | 4.6 ± 0.1 | 12.4 ± 0.4 | 14.5 ± 0.4 | 32.2 ± 2.4 | 0.1 ± 0.0 |
| FMKKM | **35.6** ± **2.3** | **69.8** ± **3.3** | **36.9** ± **1.3** | **68.9** ± **0.5** | **5.3** ± **0.2** | **45.4** ± **1.2** | **29.6** ± **1.0** | **56.3** ± **1.6** | 22.5 ± 1.8 |

Table 2: Aggregated ACC, NMI and RI comparison (mean±std) of different clustering algorithms on all benchmark datasets. Best results are marked in bold.

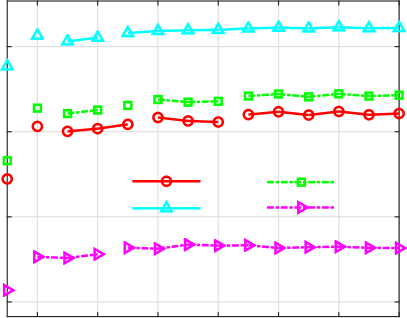


Wisconsin

ALOI-100

Performance (%)

Performance (%)



ACC

NMI

Pur RI

70

60

50

40

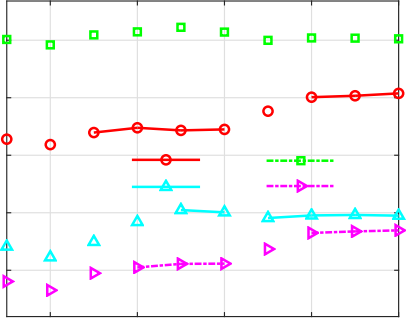
30

80

70

60

50



ACC

NMI

Pur RI

2 4 6 8 10 The Number of Iterations

2 4 6 8 10 12 14 The Number of Iterations

The Number of Iterations The Number of Iterations

Figure 1: The objective values of FMKKM varies with iterations (left) and the evolution of the learned consensus partition matrix **H**\* (right).

In fact, the convergence is achieved in less than 15 itera- tions on most cases. In addition, to show the evolution of the learned consensus partition ofFMKKM, we take **H**\* at each iteration to calculate clustering performance, and plot

them in Figure 1. As observed, the clustering performance of FMKKM usually increases and then remains stable with tiny ﬁuctuation in most cases, which sufﬁciently demonstrates the effectiveness of our algorithm. These results consider-

9114

9

8

7

6

5

4

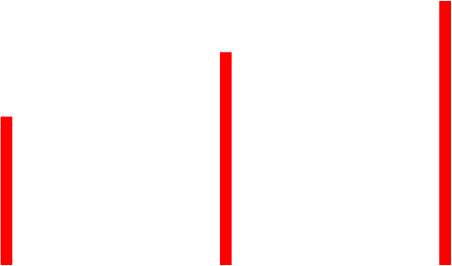
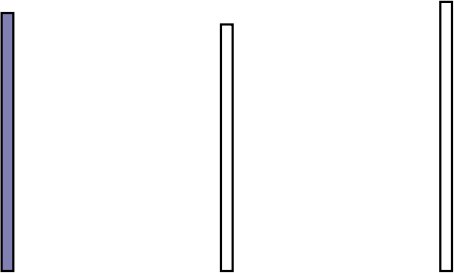
3

2

1

0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Avg-KKM | | | | LFMVC | | | | | |  | | | | | |  | | | | | |  | | | | | |  | | | | |  | | | | | | | | | | | |  | | |
|  | | MKKM LMKKM | | | | RMKKM SMKKM | | | | | |  | | | | | |  | | | | | |  | | | | | |  | | | | |  | | | | | | | |  | | | |  | |  |
|  | | ONKC  MKKM-MR | | | | NKSS  SPMKC | | | | | |  | | | | | |  | | | | | |  | | | | | |  | | | | |  | | | | | | | |  | | | |  | |  |
|  | | LKAM L\_\_\_ | | | | FMKKM | | | | | |  | | | | | |  | | | | | |  | | | | | |  | | | | |  | |  | | | | | |  | | |  |  | |  |
|  | |  | | |  |  | | | | | |  | |  | | | |  | |  | | | |  | | | | | |  | |  | | |  | |  | | | | | |  | | | |  | |  |
|  | |  | | |  |  | |  |  | | |  | |  | | | |  | |  | | | |  | |  | | | |  | |  | | |  | |  | | |  | | |  | |  | |  |  |  |
|  | |  | | |  |  |  |  |  | | |  | |  | | | |  | |  | | | |  | |  |  | | |  | |  | |  |  | |  | | |  | | |  | |  | |  |  |  |
|  |  |
|  | |  | |  |  |  |  |  |  | |  |  |  |  |  | | |  |  |  |  | | |  |  |  |  | |  |  | |  | |  |  | |  |  |  |  | |  |  | |  | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Texas Football Wisconsin BBCSport Willow Flower17 Flower102 ALOI-100 Reuters

Logar ithm of Running Time(in second)

, , , , , I

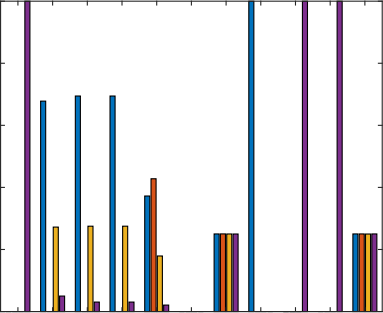
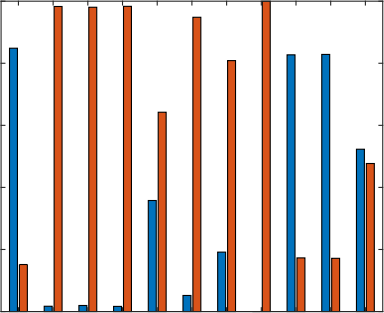
Figure 2: Running time comparison of different algorithms on nine benchmark datasets. Node that, in order to observe more clearly, we scale the values and let execution time of Avg-KKM be reference.

ably show the effectiveness and necessity of the learning procedure.

**Weight Coefﬁcients Analysis** We further study the weight coefﬁcients learned by compared algorithms on all datasets. The results are plotted in Figure 3. As seen, the ker- nel weights learned by many algorithms such as ONKC, MKKM-MiR, LKAM and especially NKSS are highly sparse on Wisconsin and ALOI-100 datasets. This sparsity would make algorithm focus on a certain preferred kernel matrix and may cause insufﬁcient fusion, leading to poor performance. However, the ﬁnal kernel weights √ learned by our proposed FMKKM are non-sparse on all datasets, which promotes the fusion procedure. Meanwhile, α and β, the co- efﬁcient for serving the clustering, pay more attention to a certain kernel to complementally supplement the deﬁcien- ciesof the learning of √ .

Wisconsin

ALOI-100

1

1

0.8

Kernel Weights

Kernel Weights

0.8

0.6

0.6

0.4

0.4

0.2

0.2

0

MKKM

ONKC

MKKM-MR

LKAM

LFMVC

RMKKM

SMKKM

NKSS

MKKM

ONKC

MKKM-MR

LKAM

LFMVC

RMKKM

SMKKM

NKSS

0



FMKKM-

FMKKM-

FMKKM-

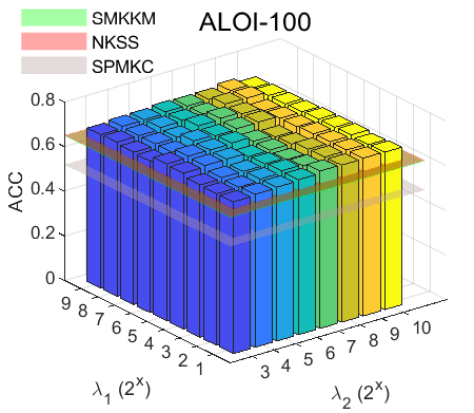
FMKKM-

FMKKM-

FMKKM-

Figure 3: The kernel weights learned by various compared algorithms.

**Parameter Sensitivity Analysis** As can be seen in Eq. (5), FMKKM introduces the regularization parameter λ1 and λ2 to tradeoff different fusion stages. We conduct experiments to study the sensitivity and effect of these parameters on the clustering performance on all datasets. Figure 4 presents the ACC of FMKKM on Wisconsin and ALOI-100 datasets by varying λ 1 in 2[1:9] and λ2 in 2[3:10] , respectively. In addi- tion, the best results of recently proposed SMKKM, NKSS and SPMKC are also provided as baselines for reference. From the observation, FMKKM achieves stable clustering performance across a wide range of λ 1 and λ2 and there is an internal connection between λ1 and λ2 .



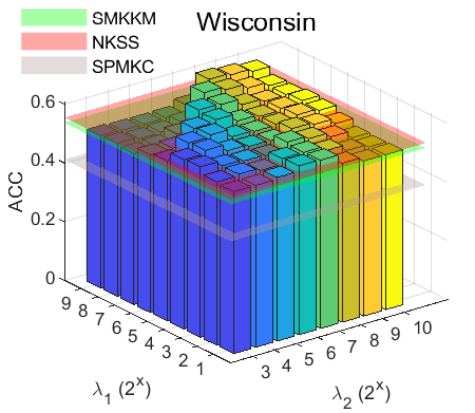


Figure 4: The sensitivity of FMKKM with the variation of λ 1 and λ2 in term of ACC.

**Running Time Comparison** Finally, we report the execu- tion time of all compared algorithms on all datasets, as plot- ted in Figure 2. As observed, our proposed FMKKM does not greatly increase the computational cost despite signiﬁ- cant improvement of the clustering performance. Moreover, as discussed above, since the computational complexity is O(mn2 k + tmnk2 + tmk3 ) at each iteration, our proposed FMKKM may have advantages on dealing with large-scale tasks rather than those algorithms with high computational complexity of O(n3 ).

**Conclusion**

In this paper, we propose a novel FMKKM algorithm, which simultaneously carries out the early fusion of base ker- nels and the late fusion of base partitions, and integrate the base partition learning into the clustering procedure. In this way, FMKKM enhances the mutual negotiation and posi- tive guidance among the base partitions learning, the clus- tering optimization of the early fusion stage and late fu- sion stage. In order to solve the resultant optimization prob- lem, we carefully develop a six-step alternate algorithm with guaranteed convergence. Comprehensive experimental re- sults show the leading performance and the effectiveness of our proposed FMKKM.

**Acknowledgements**

This work was supported by the National Key R&D Program of China (project no. 2020AAA0107100) and the National Natural Science Foundation of China (project no. 61922088, 61906020, 61872371 and 62006237).

**References**

Bang, S.; Yu, Y.; and Wu, W. 2018. Robust Multiple Kernel k-means Clustering using Min-Max Optimization. arXiv:1803.02458.

Bhadra, S.; Kaski, S.; and Rousu,J. 2017. Multi-view kernel completion. *Machine Learning*, 106(5): 713一739.

Du, L.; Zhou, P.; Shi, L.; Wang, H.; Fan, M.; Wang, W.; and Shen, Y.-D. 2015. Robust multiple kernel k-means using l21-norm. In *Twenty-fourth international joint conference on artiﬁcial intelligence*.

Gnen, M.; and Alpayd1n, E. 2011. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12: 2211一2268.

Gnen, M.; and Margolin, A. A. 2014. Localized Data Fu- sion for Kernel k-Means Clustering with Application to Can- cer Biology. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Pro- cessing Systems 2014, December 8-13 2014, Montreal, Que- bec, Canada*, 1305一1313.

Han, Y.; Yang, K.; Yang, Y.; and Ma, Y. 2016. Localized multiple kernel learning with dynamical clustering and ma- trix regularization. *IEEE transactions on neural networks and learning systems*, 29(2): 486一499.

Huang, H.-C.; Chuang, Y.-Y.; and Chen, C.-S. 2011. Mul- tiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 20(1): 120一134.

Kang, Z.; Zhao, X.; Peng, C.; Zhu, H.; Zhou, J. T.; Peng, X.; Chen, W.; and Xu, Z. 2020. Partition level multiview subspace clustering. *Neural Networks*, 122: 279一288.

Kloft, M.; Brefeld, U.; Sonnenburg, S.; and Zien, A. 2011. Lp-norm multiple kernel learning. *The Journal of Machine Learning Research*, 12: 953一997.

Kloft, M.; Rckert, U.; and Bartlett, P. L. 2010. A unifying

view of multiple kernel learning. In *Joint European Con- ference on Machine Learning and Knowledge Discovery in Databases*, 66一81. Springer.

Li,M.; Liu,X.; Wang,L.; Dou,Y.; Yin,J.; and Zhu,E. 2016. Multiple Kernel Clustering with Local Kernel Alignment Maximization. In *Proceedings of the Twenty-Fifth Inter- national Joint Conference on Artiﬁcial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, 1704一1710.

Li, T.; Dou, Y.; Liu, X.; Zhao, Y.; and Lv, Q. 2017. Multiple kernel clustering with corrupted kernels. *Neurocomputing*, 267: 447一454.

Liu, J.; Liu, X.; Yang, Y.; Wang, S.; and Zhou, S. 2021a. Hierarchical multiple kernel clustering. In *Pro- ceedings of the Thirty-ﬁfth AAAI Conference on Artiﬁcial Intelligence,(AAAI-21), Virtually*.

Liu, X.; Dou, Y.; Yin, J.; Wang, L.; and Zhu, E. 2016. Mul- tiple kernel k-means clustering with matrix-induced regular- ization. In *Proceedings of the AAAI Conference on Artiﬁcial Intelligence*, volume 30.

Liu, X.; Li, M.; Tang, C.; Xia, J.; Xiong, J.; Liu, L.; Kloft, M.; and Zhu, E. 2020. Efﬁcient and effective regularized in- complete multi-view clustering. *IEEE transactions on pat- tern analysis and machine intelligence*.

Liu,X.; Liu,L.; Liao,Q.; Wang, S.; Zhang,Y.; Tu, W.; Tang, C.; Liu, J.; and Zhu, E. 2021b. One Pass Late Fusion Multi- view Clustering. In *International Conference on Machine Learning*, 6850一6859. PMLR.

Liu, X.; Zhou, S.; Wang, Y.; Li, M.; Dou, Y.; Zhu, E.; and Yin, J. 2017. Optimal neighborhood kernel clustering with multiple kernels. In *Proceedings of the AAAI Conference on Artiﬁcial Intelligence*, volume 31.

Liu, X.; Zhu, E.; and Liu, J. 2020. SimpleMKKM:

Simple Multiple Kernel K-means. *arXiv preprint*

*arXiv:2005.04975*.

Liu, X.; Zhu, X.; Li, M.; Wang, L.; Tang, C.; Yin, J.; Shen, D.; Wang, H.; and Gao, W. 2018. Late fusion incomplete multi-view clustering. *IEEE transactions on pattern analy- sis and machine intelligence*, 41(10): 2410一2423.

Liu, X.; Zhu, X.; Li, M.; Wang, L.; Zhu, E.; Liu, T.; Kloft, M.; Shen, D.; Yin, J.; and Gao, W. 2019. Multiple kernel k- means with incomplete kernels. *IEEE transactions on pat- tern analysis and machine intelligence*, 42(5): 1191一1204.

Patel, V. M.; and Vidal, R. 2014. Kernel sparse subspace clustering. In *2014 ieee international conference on image processing (icip)*, 2849一2853. IEEE.

Ren, Z.; and Sun, Q. 2020. Simultaneous global and lo- cal graph structure preserving for multiple kernel clustering. *IEEE transactions on neural networks and learning systems*, 32(5): 1839一1851.

Scholkopf, B.; and Smola, A. J. 2001. *Learning with ker- nels: support vector machines, regularization, optimization, and beyond*. MIT press.

Wang, S.; Liu, X.; Zhu, E.; Tang, C.; Liu, J.; Hu, J.; Xia, J.; and Yin, J. 2019. Multi-view Clustering via Late Fusion Alignment Maximization. In *IJCAI*, 3778一3784.

Wang, Y.; Liu, X.; Dou, Y.; and Li, R. 2017a. Approximate Large-scale Multiple Kernel k-means Using Deep Neural Network. In *IJCAI*, 3006一3012.

Wang, Y.; Liu, X.; Dou, Y.; and Li, R. 2017b. Multiple ker- nel clustering framework with improved kernels. *Discover*, 1(2): 3一4.

Wen, Z.; and Yin, W. 2013. A feasible method for optimiza- tion with orthogonality constraints. *Mathematical Program- ming*, 142(1): 397一434.

Yao, Y.; Li,Y.; Jiang,B.; and Chen,H. 2020. Multiple kernel k-means clustering by selecting representative kernels. *IEEE Transactions on Neural Networks and Learning Systems*.

Yu, S.; Tranchevent,L.; Liu,X.; Glanzel,W.; Suykens, J. A.; De Moor, B.; and Moreau, Y. 2011. Optimized data fusion for kernel k-means clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5): 1031一1039.

Zhang, P.; Liu, X.; Xiong, J.; Zhou, S.; Zhao, W.; Zhu, E.; and Cai,Z. 2020. Consensus One-step Multi-view Subspace Clustering. *IEEE Transactions on Knowledge and Data En- gineering*.

Zhao, B.; Kwok, J. T.; and Zhang, C. 2009. Multiple Kernel Clustering. In *SDM*, 638一649.

Zhou, P.; Du, L.; Shi, L.; Wang, H.; and Shen, Y.-D. 2015. Recovery of corrupted multiple kernels for clustering. In

*Twenty-Fourth International Joint Conference on Artiﬁcial Intelligence*.

Zhou, S.; Liu, X.; Li, M.; Zhu, E.; Liu, L.; Zhang, C.; and Yin, J. 2019. Multiple kernel clustering with neighbor- kernel subspace segmentation. *IEEE transactions on neural networks and learning systems*, 31(4): 1351一1362.

Zhou, S.; Ou, Q.; Liu, X.; Wang, S.; Liu, L.; Wang, S.; Zhu, E.; Yin, J.; and Xu, X. 2021a. Multiple Kernel Clustering With Compressed Subspace Alignment. *IEEE Transactions on Neural Networks and Learning Systems*, 1一12.

Zhou, S.; Ou, Q.; Liu, X.; Wang, S.; Liu, L.; Wang, S.; Zhu, E.; Yin, J.; and Xu, X. 2021b. Multiple Kernel Clustering With Compressed Subspace Alignment. *IEEE Transactions on Neural Networks and Learning Systems*.