Multiple Kernel k-means with Incomplete Kernels

Xinwang Liu, Miaomiao Li
School of Computer
National University
of Defense Technology
Changsha, China, 410073

Lei Wang
School of Computer Science
and Software Engineering
University of Wollongong
NSW, Australia, 2522

Yong Dou, Jianping Yin, En Zhu
School of Computer
National University
of Defense Technology
Changsha, China, 410073

Abstract

Multiple kernel clustering (MKC) algorithms optimally combine a group of pre-specified base kernels to improve clustering performance. However, existing MKC algorithms cannot efficiently address the situation where some rows and columns of base kernels are absent. This paper proposes a simple while effective algorithm to address this issue. Different from existing approaches where incomplete kernels are firstly imputed and a standard MKC algorithm is applied to the imputed kernels, our algorithm integrates imputation and clustering into a unified learning procedure. Specifically, we perform multiple kernel clustering directly with the presence of incomplete kernels, which are treated as auxiliary variables to be jointly optimized. Our algorithm does not require that there be at least one complete base kernel over all the samples. Also, it adaptively imputes incomplete kernels and combines them to best serve clustering. A three-step iterative algorithm with proved convergence is designed to solve the resultant optimization problem. Extensive experiments are conducted on four benchmark data sets to compare the proposed algorithm with existing imputation-based methods. Our algorithm consistently achieves superior performance and the improvement becomes more significant with increasing missing ratio, verifying the effectiveness and advantages of the proposed joint imputation and clustering.

Introduction

The recent years have seen many effort devoted to designing effective and efficient multiple kernel clustering (MKC) algorithms (Zhao, Kwok, and Zhang 2009; Yu et al. 2012; Gönen and Margolin 2014; Du et al. 2015; Liu et al. 2016; Li et al. 2016; Cao et al. 2015a; Zhang et al. 2015; Cao et al. 2015b; Zhang et al. 2016). They aim to optimally combine a group of pre-specified base kernels to perform data clustering. For example, the work in (Zhao, Kwok, and Zhang 2009) proposes to find the maximum margin hyperplane, the best cluster labeling, and the optimal kernel simultaneously. A novel optimized kernel k-means algorithm is presented in (Yu et al. 2012) to combine multiple data sources for clustering analysis. In (Gönen and Margolin 2014), the kernel combination weights are allowed to adaptively change to capture the characteristics of individual samples. Replacing the squared error in k-means with an $\ell_{2,1}$-norm based one, the work in (Du et al. 2015) develops a robust multiple kernel k-means (MKKM) algorithm that simultaneously finds the best clustering labels and the optimal combination of kernels. Observing that existing MKKM algorithms do not sufficiently consider the correlation among base kernels, the work in (Liu et al. 2016) designs a matrix-induced regularization to reduce the redundancy and enhance the diversity of the selected kernels. These algorithms have been applied to various applications and demonstrated attractive clustering performance (Yu et al. 2012; Gönen and Margolin 2014).

One underlying assumption commonly adopted by the above-mentioned MKC algorithms is that all of the base kernels are complete, i.e., none of the rows or columns of any base kernel shall be absent. In some practical applications such as Alzheimer's disease prediction (Xiang et al. 2013) and cardiac disease discrimination (Kumar et al. 2013), however, it is not uncommon to see that some views of a sample are missing, and this causes the corresponding rows and columns of related base kernels unfilled. The presence of incomplete base kernels makes it difficult to utilize the information of all views for clustering. A straightforward remedy may firstly impute incomplete kernels with a filling algorithm and then perform a standard MKC algorithm with the imputed kernels. Some widely used filling algorithms include zero-filling, mean value filling, k-nearest-neighbor filling and expectation-maximization (EM) filling (Ghahramani and Jordan 1993). Recently, more advanced imputation algorithms have been developed (Trivedi et al. 2010; Xu, Tao, and Xu 2015; Bhadra, Kaski, and Rousu 2016; Shao, He, and Yu 2015; Liu et al. 2014; 2015). The work in (Trivedi et al. 2010) constructs a full kernel matrix for an incomplete view with the help of the other complete view (or equally, base kernel). By exploiting the connections of multiple views, the work in (Xu, Tao, and Xu 2015) proposes an algorithm to accomplish multi-view learning with incomplete views, where different views are assumed to be generated from a shared subspace. In (Shao, He, and Yu 2015), a multi-incomplete-view clustering algorithm is proposed. It learns latent feature matrices for all the views and generates a consensus matrix so that the difference between each view and the consensus is minimized. In addition, by modelling both within-view and between-view relationships among kernel values, an approach is proposed in (Bhadra, Kaski, and

Rousu 2016) to predict missing rows and columns of a base kernel. Though demonstrating promising clustering performance in various applications, the above "two-stage" algorithms share a drawback that they disconnect the processes of imputation and clustering, and this prevents the two learning processes from negotiating with each other to achieve the optimal clustering. *Can we design a clustering-oriented imputation algorithm to enhance a kernel for clustering?*

To address this issue, we propose an absent multiple kernel k-means algorithm that integrates imputation and clustering into a single optimization procedure. In our algorithm, the clustering result at the last iteration guides the imputation of absent kernel elements, and the latter is in turn used to conduct the subsequent clustering. These two procedures are alternately performed until convergence. By this way, the imputation and clustering processes are seamlessly connected, with the aim to achieve better clustering performance. The optimization objective of the proposed absent multiple kernel clustering algorithm is carefully designed and an efficient algorithm with proved convergence is developed to solve the resultant optimization problem. Extensive experimental study is carried out on four multiple kernel learning (MKL) benchmark data sets to evaluate the clustering performance of the proposed algorithm. As indicated, our algorithm significantly outperforms existing two-stage imputation methods, and the improvement is particularly significant at high missing ratios, which is desirable. It is expected that the simplicity and effectiveness of this clustering algorithm will make it a good option to be considered for practical applications where incomplete views or kernels are encountered.

Related Work

Kernel k-means clustering (KKM)

Let $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ be a collection of n samples, and $\phi(\cdot) : \mathbf{x} \in \mathcal{X} \mapsto \mathcal{H}$ be a feature mapping that maps $\mathbf{x}$ onto a reproducing kernel Hilbert space $\mathcal{H}$. The objective of kernel k-means clustering is to minimize the sum-of-squares loss over the cluster assignment matrix $\mathbf{Z} \in \{0,1\}^{n\times k}$, which can be formulated as the following optimization problem,

$$\begin{aligned} \min\nolimits_{\mathbf{Z}\in\{0,1\}^{n\times k}} \quad & \sum\nolimits_{i=1,c=1}^{n,k} Z_{ic}\|\phi(\mathbf{x}_i) - \boldsymbol{\mu}_c\|_2^2 \\ s.t. \quad & \sum\nolimits_{c=1}^{k} Z_{ic} = 1, \end{aligned} \tag{1}$$

where $n_c = \sum_{i=1}^n Z_{ic}$ and $\boldsymbol{\mu}_c = \frac{1}{n_c}\sum_{i=1}^n Z_{ic}\phi(\mathbf{x}_i)$ are the size and centroid of the c-th cluster.

The optimization problem in Eq.(1) can be rewritten as the following matrix-vector form,

$$\min\nolimits_{\mathbf{Z}\in\{0,1\}^{n\times k}} \quad \mathrm{Tr}(\mathbf{K}) - \mathrm{Tr}(\mathbf{L}^{\frac{1}{2}}\mathbf{Z}^\top\mathbf{K}\mathbf{Z}\mathbf{L}^{\frac{1}{2}}) \;\; s.t. \;\; \mathbf{Z}\mathbf{1}_k = \mathbf{1}_n, \tag{2}$$

where $\mathbf{K}$ is a kernel matrix with $K_{ij} = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)$, $\mathbf{L} = \mathrm{diag}([n_1^{-1}, n_2^{-1}, \cdots, n_k^{-1}])$ and $\mathbf{1}_\ell \in \mathbb{R}^\ell$ is a column vector with all elements being 1.

The variable $\mathbf{Z}$ in Eq.(2) is discrete, and this makes the optimization problem difficult to solve. A common approach is to relax $\mathbf{Z}$ to take real values. Specifically, by defining $\mathbf{H} = \mathbf{Z}\mathbf{L}^{\frac{1}{2}}$ and letting $\mathbf{H}$ take real values, a relaxed version of the above problem can be obtained as

$$\min\nolimits_{\mathbf{H}} \; \mathrm{Tr}\left(\mathbf{K}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\right) \;\; s.t. \;\; \mathbf{H} \in \mathbb{R}^{n\times k}, \; \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \tag{3}$$

where $\mathbf{I}_k$ is an identity matrix with size $k \times k$. The optimal $\mathbf{H}$ for Eq.(3) can be obtained by taking the k eigenvectors having the larger eigenvalues of $\mathbf{K}$ (Jegelka et al. 2009).

Multiple kernel k-means clustering (MKKM)

In a multiple kernel setting, each sample has multiple feature representations defined by a group of feature mappings $\{\phi_p(\cdot)\}_{p=1}^m$. Specifically, each sample is represented as $\phi_{\boldsymbol{\beta}}(\mathbf{x}) = [\beta_1\phi_1(\mathbf{x})^\top, \cdots, \beta_m\phi_m(\mathbf{x})^\top]^\top$, where $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_m]^\top$ consists of the coefficients of the m base kernels. These coefficients will be optimized during learning. Based on the definition of $\phi_{\boldsymbol{\beta}}(\mathbf{x})$, a kernel function can be expressed as

$$\kappa_{\boldsymbol{\beta}}(\mathbf{x}_i, \mathbf{x}_j) = \phi_{\boldsymbol{\beta}}(\mathbf{x}_i)^\top\phi_{\boldsymbol{\beta}}(\mathbf{x}_j) = \sum\nolimits_{p=1}^m \beta_p^2\kappa_p(\mathbf{x}_i, \mathbf{x}_j). \tag{4}$$

By replacing the kernel matrix $\mathbf{K}$ in Eq.(3) with $\mathbf{K}_{\boldsymbol{\beta}}$ computed via Eq.(4), the objective of MKKM can be written as

$$\begin{aligned} &\min\nolimits_{\mathbf{H},\boldsymbol{\beta}} \; \mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\right) \\ &\qquad s.t. \;\; \mathbf{H} \in \mathbb{R}^{n\times k}, \; \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \; \boldsymbol{\beta}^\top\mathbf{1}_m = 1, \; \beta_p \geq 0, \; \forall p. \end{aligned} \tag{5}$$

This problem can be solved by alternately updating $\mathbf{H}$ and $\boldsymbol{\beta}$: i) **Optimizing $\mathbf{H}$ given $\boldsymbol{\beta}$**. With the kernel coefficients $\boldsymbol{\beta}$ fixed, $\mathbf{H}$ can be obtained by solving a kernel k-means clustering optimization problem shown in Eq.(3); ii) **Optimizing $\boldsymbol{\beta}$ given $\mathbf{H}$**. With $\mathbf{H}$ fixed, $\boldsymbol{\beta}$ can be optimized via solving the following quadratic programming with linear constraints,

$$\min\nolimits_{\boldsymbol{\beta}} \; \sum\nolimits_{p=1}^m \beta_p^2\mathrm{Tr}\left(\mathbf{K}_p(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\right) \;\; s.t. \;\; \boldsymbol{\beta}^\top\mathbf{1}_m = 1, \; \beta_p \geq 0, \; \forall p. \tag{6}$$

As noted in (Yu et al. 2012; Gönen and Margolin 2014), using a convex combination of kernels $\sum_{p=1}^m \beta_p\mathbf{K}_p$ to replace $\mathbf{K}_{\boldsymbol{\beta}}$ in Eq.(5) is not a viable option, because this could make only one single kernel be activated and all the others assigned with zero weights. Other recent work using ℓ_2-norm combination can be found in (Kloft et al. 2011; 2009; Cortes, Mohri, and Rostamizadeh 2009; Liu et al. 2013).

The Proposed Algorithm

Formulation

Let $\mathbf{s}_p$ $(1 \leq p \leq m)$ denote the sample indices for which the p-th view is present and $\mathbf{K}_p^{(cc)}$ be used to denote the kernel sub-matrix computed with these samples. Note that this setting is consistent with the literature, and it is even more general since it does not require that there be at least one complete view across all the samples, as assumed in (Trivedi et al. 2010).

The absence of rows and columns from base kernels makes clustering challenging. Existing two-stage approaches first impute these base kernels and then apply a conventional clustering algorithm with them. We have the following two arguments. Firstly, although such imputation is sound from the perspective of "general-purpose", it may not be an optimal option when it has been known that the imputed kernels are used for clustering. This is because for most, if not all, practical tasks a belief holds that these pre-selected base kernels or views (when in their complete form) shall, more or less, be able to serve clustering. However, such a belief was not exploited by these two-stage approaches as prior knowledge to guide the imputation process. Secondly, from the perspective that the ultimate goal is to appropriately cluster data, we shall try to directly pursue the clustering result, by treating the absent kernel elements as auxiliary unknowns during this course. In other words, imputed kernels could be merely viewed as the by-products of clustering.

These two arguments motivate us to seek a more natural and reasonable manner to deal with the absence in multiple kernel clustering. That is to perform imputation and clustering in a joint way: 1) impute the absent kernels under the guidance of clustering; and 2) update the clustering with the imputed kernels. By this way, *the above two learning processes can be seamlessly coupled and they are allowed to negotiate with each other to achieve better clustering*. In specific, we propose the multiple kernel k-means algorithm with incomplete kernels as follows,

$$\begin{aligned}\min_{\mathbf{H},\,\boldsymbol{\beta},\,\{\mathbf{K}_p\}_{p=1}^m}\ & \mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\beta}}(\mathbf{I}_n-\mathbf{H}\mathbf{H}^\top)\right)\\ s.t.\ & \mathbf{H}\in\mathbb{R}^{n\times k},\ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k,\ \boldsymbol{\beta}^\top\mathbf{1}_m=1,\ \beta_p\geq 0,\\ & \mathbf{K}_p(\mathbf{s}_p,\mathbf{s}_p)=\mathbf{K}_p^{(cc)},\ \mathbf{K}_p\succeq 0,\ \forall p,\end{aligned} \tag{7}$$

The only difference between the objective function in Eq.(7) and that of traditional MKKM in Eq.(5) lies at the incorporation of optimizing $\{\mathbf{K}_p\}_{p=1}^m$. Note that the constraint $\mathbf{K}_p(\mathbf{s}_p,\mathbf{s}_p)=\mathbf{K}_p^{(cc)}$ is imposed to ensure that $\mathbf{K}_p$ maintains the known entries during the course. Though the model in Eq.(7) is simple, it admits the following advantages: 1) Our objective function is more direct and well targets the ultimate goal, i.e., clustering, by integrating kernel completion and clustering into one unified learning framework, where the kernel imputation is treated as a by-product; 2) Our algorithm works in a MKL scenario (Rakotomamonjy et al. 2008), which is able to naturally deal with a large number of base kernels and adaptively combine them for clustering; 3) Our algorithm does not require any base kernel to be completely observed, which is however necessary for some of the existing imputation algorithms such as (Trivedi et al. 2010). Besides, our algorithm is parameter-free once the number of clusters to form is specified.

Alternate optimization

Although Eq.(7) is not difficult to understand, the positive semi-definite (PSD) constraints on $\{\mathbf{K}_p\}_{p=1}^m$ make it difficult to optimize. In the following, we design an efficient algorithm to solve it. In specific, we design a three-step algorithm to solve this problem in an alternate manner:

i) **Optimizing $\mathbf{H}$ with fixed $\boldsymbol{\beta}$ and $\{\mathbf{K}_p\}_{p=1}^m$**. Given $\boldsymbol{\beta}$ and $\{\mathbf{K}_p\}_{p=1}^m$, the optimization in Eq.(7) for $\mathbf{H}$ reduces to a standard kernel k-means problem, which can be efficiently solved as Eq.(3);

Algorithm 1 Proposed Multiple Kernel k-means with Incomplete Kernels

```
Input: $\{\mathbf{K}_p^{(cc)}\}_{p=1}^m$, $\{\mathbf{s}_p\}_{p=1}^m$ and $\epsilon_0$.
Output: $\mathbf{H}$, $\boldsymbol{\beta}$ and $\{\mathbf{K}_p\}_{p=1}^m$.
Initialize $\boldsymbol{\beta}^{(0)}=\mathbf{1}_m/m$, $\{\mathbf{K}_p^{(0)}\}_{p=1}^m$ and $t=1$.
repeat
    $\mathbf{K}_{\boldsymbol{\beta}}^{(t)}=\sum_{p=1}^m\left(\beta_p^{(t-1)}\right)^2\mathbf{K}_p^{(t-1)}$.
    Update $\mathbf{H}^{(t)}$ by solving Eq.(3) with $\mathbf{K}_{\boldsymbol{\beta}}^{(t)}$.
    Update $\{\mathbf{K}_p^{(t)}\}_{p=1}^m$ with $\mathbf{H}^{(t)}$ by Eq.(12).
    Update $\boldsymbol{\beta}^{(t)}$ by solving Eq.(6) with $\mathbf{H}^{(t)}$ and $\{\mathbf{K}_p^{(t)}\}_{p=1}^m$.
    $t=t+1$.
until $\left(\mathrm{obj}^{(t-1)}-\mathrm{obj}^{(t)}\right)/\mathrm{obj}^{(t)}\leq\epsilon_0$
```

ii) **Optimizing $\{\mathbf{K}_p\}_{p=1}^m$ with fixed $\boldsymbol{\beta}$ and $\mathbf{H}$**. Given $\boldsymbol{\beta}$ and $\mathbf{H}$, the optimization in Eq.(7) with respect to $\{\mathbf{K}_p\}_{p=1}^m$ is equivalent to the following optimization problem,

$$\begin{aligned}\min_{\{\mathbf{K}_p\}_{p=1}^m}\ & \sum\nolimits_{p=1}^m\beta_p^2\mathrm{Tr}\left(\mathbf{K}_p(\mathbf{I}_n-\mathbf{H}\mathbf{H}^\top)\right)\\ & s.t.\ \mathbf{K}_p(\mathbf{s}_p,\mathbf{s}_p)=\mathbf{K}_p^{(cc)},\ \mathbf{K}_p\succeq 0,\ \forall p.\end{aligned} \tag{8}$$

Directly solving the optimization problem in Eq.(8) appears to be computationally intractable because it involves multiple kernel matrices. Looking into this optimization problem, we can find that the constraints are separately defined on each $\mathbf{K}_p$ and that the objective function is a sum over each $\mathbf{K}_p$. Therefore, we can equivalently rewrite the problem in Eq.(8) as m independent sub-problems, as stated in Eq.(9),

$$\min_{\mathbf{K}_p}\ \mathrm{Tr}\left(\mathbf{K}_p\mathbf{U}\right)\quad s.t.\ \mathbf{K}_p(\mathbf{s}_p,\mathbf{s}_p)=\mathbf{K}_p^{(cc)},\ \mathbf{K}_p\succeq 0, \tag{9}$$

where $\mathbf{U}=\mathbf{I}_n-\mathbf{H}\mathbf{H}^\top$ and $p=1,\cdots,m$.

Considering that $\mathbf{K}_p$ is PSD, we can decompose $\mathbf{K}_p$ as $\mathbf{A}_p\mathbf{A}_p^\top$. Inspired by the work in (Trivedi et al. 2010), we write $\mathbf{A}_p=[\mathbf{A}_p^{(c)};\mathbf{A}_p^{(m)}]$ with $\mathbf{A}_p^{(c)}\mathbf{A}_p^{(c)\top}=\mathbf{K}_p^{(cc)}$. In this way, the optimization problem in Eq.(9) can be rewritten as

$$\min_{\mathbf{A}_p^{(m)}}\ \mathrm{Tr}\left(\left[\mathbf{A}_p^{(c)};\mathbf{A}_p^{(m)}\right]^\top\begin{bmatrix}\mathbf{U}^{(cc)} & \mathbf{U}^{(cm)}\\ \mathbf{U}^{(cm)\top} & \mathbf{U}^{(mm)}\end{bmatrix}\left[\mathbf{A}_p^{(c)};\mathbf{A}_p^{(m)}\right]\right), \tag{10}$$

where the matrix $\mathbf{U}$ is expressed in a blocked form as $\begin{bmatrix}\mathbf{U}^{(cc)} & \mathbf{U}^{(cm)}\\ \mathbf{U}^{(cm)\top} & \mathbf{U}^{(mm)}\end{bmatrix}$.

By taking the derivative of Eq.(10) with respect to $\mathbf{A}_p^{(m)}$ and letting it vanish, we can obtain an analytical solution to the optimal $\mathbf{A}_p^{(m)}$ as

$$\mathbf{A}_p^{(m)}=\left(\mathbf{U}^{(mm)}\right)^{-1}\mathbf{U}^{(cm)\top}\mathbf{A}_p^{(c)}. \tag{11}$$

Correspondingly, we have a closed-form expression for the optimal $\mathbf{K}_p$ in Eq.(9):

$$\begin{bmatrix}\mathbf{K}_p^{(cc)} & -\mathbf{K}_p^{(cc)}\mathbf{U}^{(cm)}(\mathbf{U}^{(mm)})^{-1}\\ -(\mathbf{U}^{(mm)})^{-1}\mathbf{U}^{(cm)\top}\mathbf{K}_p^{(cc)} & (\mathbf{U}^{(mm)})^{-1}\mathbf{U}^{(cm)\top}\mathbf{K}_p^{(cc)}\mathbf{U}^{(cm)}(\mathbf{U}^{(mm)})^{-1}\end{bmatrix}. \tag{12}$$

iii) **Optimizing $\boldsymbol{\beta}$ with fixed $\mathbf{H}$ and $\{\mathbf{K}_p\}_{p=1}^m$.** Given $\mathbf{H}$ and $\{\mathbf{K}_p\}_{p=1}^m$, the optimization in Eq.(7) for $\boldsymbol{\beta}$ is a quadratic programming with linear constraints, which can be efficiently solved as in Eq.(6).

In sum, our algorithm for solving Eq.(7) is outlined in Algorithm 1, where the absent elements of $\{\mathbf{K}_p^{(0)}\}_{p=1}^m$ are initially imputed with zeros and $\mathrm{obj}^{(t)}$ denotes the objective value at the t-th iteration. It is worth pointing out that the objective of Algorithm 1 is guaranteed to be monotonically decreased when optimizing one variable with others fixed at each iteration. At the same time, the objective is lower-bounded by zero. As a result, our algorithm is guaranteed to converge. Also, as shown in the experimental study, it usually converges in less than 30 iterations. As MKKM, our algorithm solves an eigen-decomposition and a QP problem per iteration, which brings no much extra computation since imputation is done analytically in Eq.(12).

Experimental Result

Experimental settings

The proposed algorithm is experimentally evaluated on four widely used MKL benchmark data sets shown in Table 1. They are Oxford Flower17[1], Oxford Flower102[2], Columbia Consumer Video (CCV)[3] and Caltech102[4]. For Flower17, Flower102 and Caltech102 data sets, all kernel matrices are pre-computed and can be publicly downloaded from the above websites. For Caltech102, we use its first ten base kernels for evaluation. For CCV data set, we generate six base kernels by applying both a linear kernel and a Gaussian kernel on its SIFT, STIP and MFCC features, where the widths of the three Gaussian kernels are set as the mean of all pairwise sample distances, respectively.

Table 1: Datasets used in our experiments.

Dataset	#Samples	#Kernels	#Classes
Flower17	1360	7	17
Flower102	8189	4	102
Caltech102	3060	10	102
CCV	6773	6	20

We compare the proposed algorithm with several commonly used imputation methods, including zero filling (ZF), mean filling (MF), k-nearest-neighbor filling (KNN) and the alignment-maximization filling (AF) proposed in (Trivedi et al. 2010). The algorithms in (Xu, Tao, and Xu 2015; Shao, He, and Yu 2015; Zhao, Liu, and Fu 2016) are not incorporated into our experimental comparison since they only consider the absence of input features while not the rows/columns of base kernels. Compared with (Bhadra, Kaski, and Rousu 2016), the imputation algorithm in (Trivedi et al. 2010) is much simpler and more computationally efficient. Therefore, we choose (Trivedi et al. 2010) as a representative algorithm to demonstrate the advantages and effectiveness of joint optimization on imputation and clustering. The widely used MKKM (Gönen and Margolin 2014) is applied with these imputed base kernels. These two-stage methods are termed ZF+MKKM, MF+MKKM, KNN+MKKM and AF+MKKM in this experiment, respectively. We do not include the EM-based imputation algorithm due to its high computational cost, even for small-sized samples. The Matlab codes of kernel k-means and MKKM are publicly downloaded from `https://github.com/mehmetgonen/lmkkmeans`.

Following the literature (Cortes, Mohri, and Rostamizadeh 2012), all base kernels are centered and scaled so that we have $\kappa_p(\mathbf{x}_i, \mathbf{x}_i) = 1$ for all i and p. For all data sets, it is assumed that the true number of clusters is known and it is set as the true number of classes. To generate incomplete kernels, we create the index vectors $\{\mathbf{s}_p\}_{p=1}^m$ as follows. We first randomly select $\mathrm{round}(\varepsilon * n)$ samples, where $\mathrm{round}(\cdot)$ denotes a rounding function. For each selected sample, a random vector $\mathbf{v} = (v_1, \cdots, v_m) \in [0,1]^m$ and a scalar v_0 ($v_0 \in [0,1]$) are then generated, respectively. The p-th view will be present for this sample if $v_p \geq v_0$ is satisfied. In case none of $v_1, \cdots, v_m$ can satisfy this condition, we will generate a new $\mathbf{v}$ to ensure that at least one view is available for a sample. Note that this does not mean that we require a complete view across all the samples. After the above step, we will be able to obtain the index vector $\mathbf{s}_p$ listing the samples whose p-th view is present. The parameter ε, termed missing ratio in this experiment, controls the percentage of samples that have absent views, and it affects the performance of the algorithms in comparison. Intuitively, the larger the value of ε is, the poorer the clustering performance that an algorithm can achieve. In order to show this point in depth, we compare these algorithms with respect to ε. Specifically, ε on all the four data sets is set as $[0.1 : 0.1 : 0.9]$.

The widely used clustering accuracy (ACC), normalized mutual information (NMI) and purity are applied to evaluate the clustering performance. For all algorithms, we repeat each experiment for 50 times with random initialization to reduce the affect of randomness caused by k-means, and report the best result. Meanwhile, we randomly generate the "incomplete" patterns for 30 times in the above-mentioned way and report the statistical results. The aggregated ACC, NMI and purity are used to evaluate the goodness of the algorithms in comparison. Taking the aggregated ACC for example, it is obtained by averaging the averaged ACC achieved by an algorithm over different ε.

Experimental results

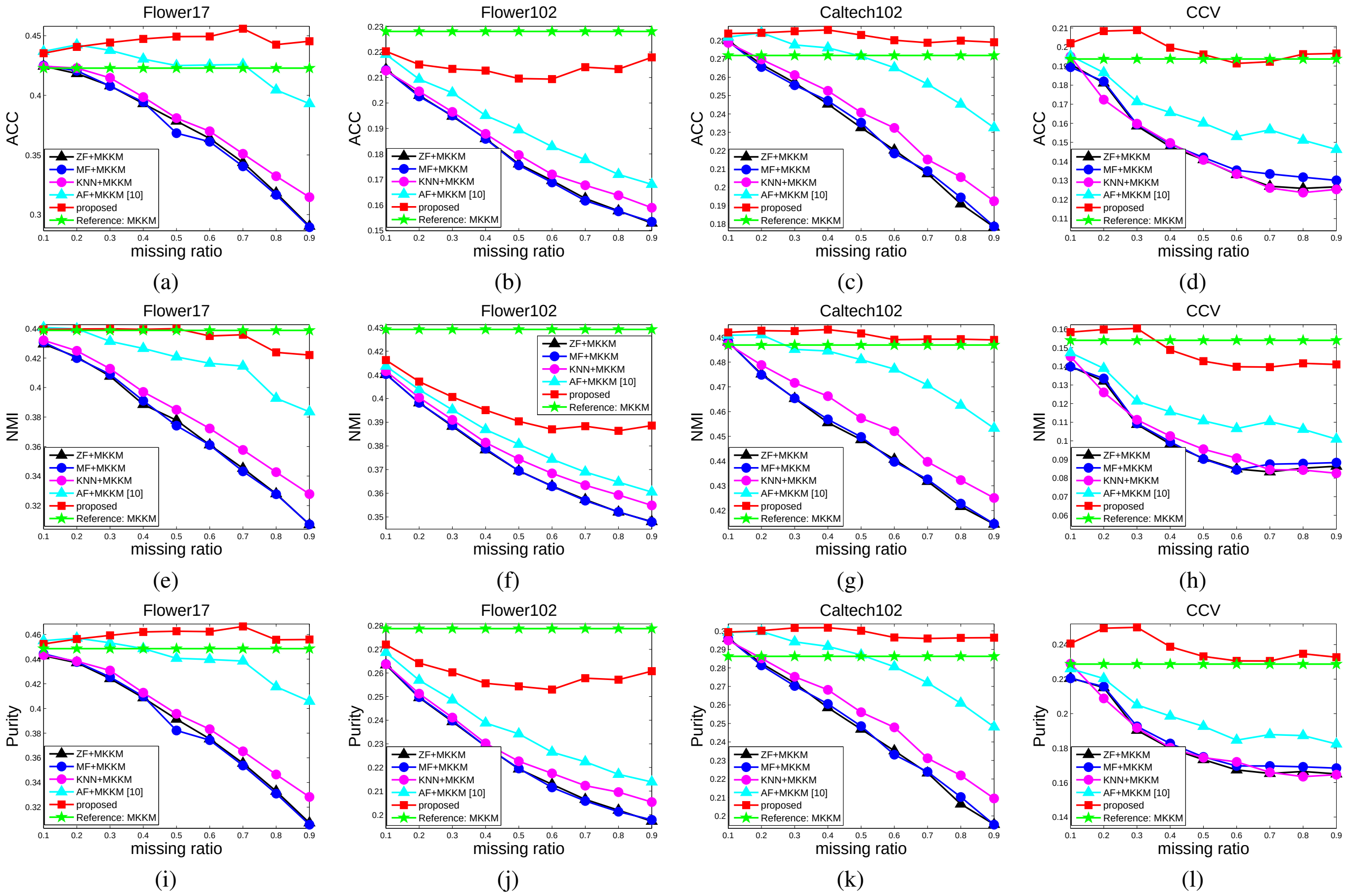
Figure 1 presents the ACC, NMI and purity comparison of the above algorithms with different missing ratios on the four data sets. To help understand the performance achieved by our algorithm, we also provide MKKM as a reference. Note that there is not any absence in the base kernels of MKKM. As observed: 1) The proposed algorithm (in red) consistently demonstrates the overall best performance among the MKKM methods with absent kernels in all the sub-figures; 2) The improvement of our algorithm is more significant with the increase of missing ratio. For example, it improves the second best algorithm (AF+MKKM) by nearly five percentage points on Flower102 in terms of clus-

[1] http://www.robots.ox.ac.uk/~vgg/data/flowers/17/

[2] http://www.robots.ox.ac.uk/~vgg/data/flowers/102/

[3] http://www.ee.columbia.edu/ln/dvmm/CCV/

[4] http://files.is.tue.mpg.de/pgehler/projects/iccv09/



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 1: Clustering accuracy, NMI and purity comparison with the variation of missing ratios on four data sets. **Note that MKKM (in green) is provided as a reference. There is not any absence in its base kernels.**

Table 2: Aggregated ACC, NMI and purity comparison (mean±std) of different clustering algorithms on four data sets.

Datasets	ZF+MKKM	MF+MKKM	KNNF+MKKM	AF+MKKM (Trivedi et al. 2010)	Proposed
			ACC		
Flower17	37.09 ± 0.42	36.93 ± 0.48	37.88 ± 0.62	42.46 ± 0.59	**44.56 ± 0.61**
Flower102	17.95 ± 0.15	17.92 ± 0.16	18.26 ± 0.14	19.09 ± 0.17	**21.40 ± 0.18**
Caltech102	23.10 ± 0.26	23.15 ± 0.24	23.87 ± 0.26	26.56 ± 0.22	**28.22 ± 0.27**
CCV	14.80 ± 0.16	15.03 ± 0.16	14.73 ± 0.19	16.51 ± 0.25	**19.91 ± 0.32**
			NMI		
Flower17	37.40 ± 0.35	37.38 ± 0.40	38.36 ± 0.46	41.85 ± 0.42	**43.50 ± 0.42**
Flower102	37.39 ± 0.08	37.39 ± 0.08	37.83 ± 0.09	38.32 ± 0.11	**39.55 ± 0.10**
Caltech102	44.90 ± 0.15	44.94 ± 0.14	45.67 ± 0.18	47.74 ± 0.14	**49.10 ± 0.18**
CCV	10.11 ± 0.13	10.23 ± 0.13	10.25 ± 0.16	11.76 ± 0.19	**14.80 ± 0.20**
			Purity		
Flower17	38.61 ± 0.40	38.49 ± 0.48	39.38 ± 0.56	43.96 ± 0.54	**45.92 ± 0.53**
Flower102	22.44 ± 0.12	22.43 ± 0.11	22.82 ± 0.14	23.63 ± 0.15	**25.95 ± 0.14**
Caltech102	24.62 ± 0.25	24.66 ± 0.26	25.44 ± 0.27	28.15 ± 0.22	**29.87 ± 0.25**
CCV	18.26 ± 0.15	18.48 ± 0.16	18.33 ± 0.20	19.83 ± 0.26	**23.79 ± 0.28**

tering accuracy when the missing ratio is 0.9 (see Figure 1(c)); 3) The variation of our algorithm with respect to the missing ratio is relatively smaller when compared with other algorithms, demonstrating its stability in the case of intensive absence; and 4) The performance of our algorithm is the closest one to or even better than the performance of MKKM (in green) in multiple cases.

We attribute the superiority of our algorithm to its joint optimization on imputation and clustering. On one hand, the imputation is guided by the clustering results, which makes the imputation more directly targeted at the ultimate goal. On the other hand, this meaningful imputation is beneficial

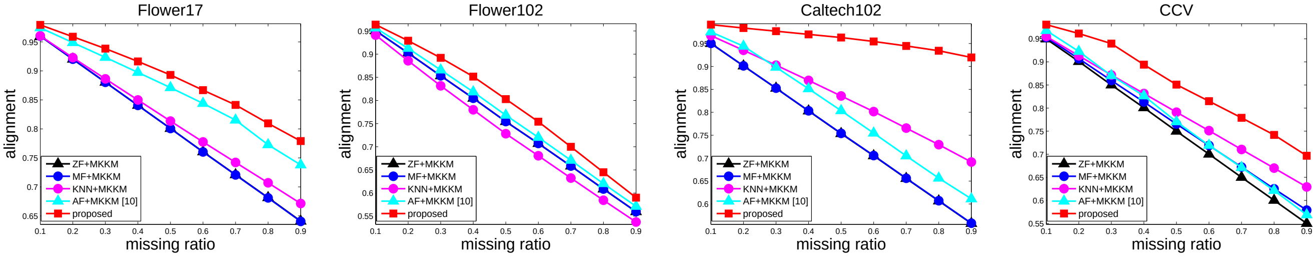


Figure 2: Kernel alignment between the original kernels and the imputed kernels by different algorithms under different missing ratios.

Table 3: Aggregated alignment between the original kernels and the imputed kernels (mean±std) on four data sets.

Datasets	ZF+MKKM	MF+MKKM	KNNF+MKKM	AF+MKKM (Trivedi et al. 2010)	Proposed
Flower17	80.07 ± 0.08	80.05 ± 0.08	81.45 ± 0.06	86.50 ± 0.08	**88.70 ± 0.12**
Flower102	75.55 ± 0.05	75.55 ± 0.05	73.35 ± 0.04	76.71 ± 0.05	**79.20 ± 0.06**
Caltech102	74.40 ± 0.05	74.43 ± 0.05	83.32 ± 0.05	80.00 ± 0.05	**95.99 ± 0.03**
CCV	75.03 ± 0.07	76.60 ± 0.07	79.15 ± 0.06	77.10 ± 0.07	**85.11 ± 0.24**

to refine the clustering results. These two learning processes negotiate with each other, leading to improved clustering performance. In contrast, ZF+MKKM, MF+MKKM, KNN+MKKM and AF+MKKM algorithms do not fully take advantage of the connection between the imputation and clustering procedures. This could produce imputation that does not well serve the subsequent clustering as originally expected, affecting the clustering performance. The aggregated ACC, NMI and purity, and the standard deviation are reported in Table 2, where the one with the highest performance is shown in bold. Again, we observe that the proposed algorithm significantly outperforms ZF+MKKM, MF+MKKM, KNN+MKKM and AF+MKKM algorithms, which is consistent with our observations in Figure 1.

Besides comparing the above-mentioned algorithms in terms of clustering performance, we would like to gain more insight on how close the imputed base kernels (as a by-product of our algorithm) are to the ground-truth, i.e., the original, complete base kernels. To do this, we calculate the alignment between the ground-truth kernels and the imputed ones. The kernel alignment, a widely used criterion to measure the similarity of two kernel matrices, is used to serve this purpose (Cortes, Mohri, and Rostamizadeh 2012). We compare the alignment resulted from our algorithm with those from existing imputation algorithms. The results under various missing ratios are shown in Figure 2. As observed, the kernels imputed by our algorithm align with the ground-truth kernels much better than those obtained by the existing imputation algorithms. In particular, our algorithm wins the second best one (KNN+MKKM) by more than 22 percentage points on Caltech102 when the missing ratio is 0.9. The aggregated alignment and the standard deviation are reported in Table 3. We once again observe the significant superiority of our algorithm to the compared ones. These results indicate that our algorithm can not only achieve better clustering performance, but is also able to produce better imputation result by exploiting the prior knowledge of "serve clustering".

From the above experiments, we conclude that the proposed algorithm: 1) effectively addresses the issue of row/columns absence in multiple kernel clustering; 2) consistently achieves performance superior to the comparable ones, especially in the presence of intensive absence; and 3) can better recover the incomplete base kernels by taking into account the goal of clustering. In short, our algorithm well utilizes the connection between imputation and clustering procedures, bringing forth significant improvements on clustering performance. In addition, our algorithm is theoretically guaranteed to converge to a local minimum according to (Bezdek and Hathaway 2003). In the above experiments, we observe that the objective value of our algorithm does monotonically decrease at each iteration and that it usually converges in less than 30 iterations. Two examples of the evolution of the objective value on Flower17 and Flower102 are demonstrated in Figure 3.

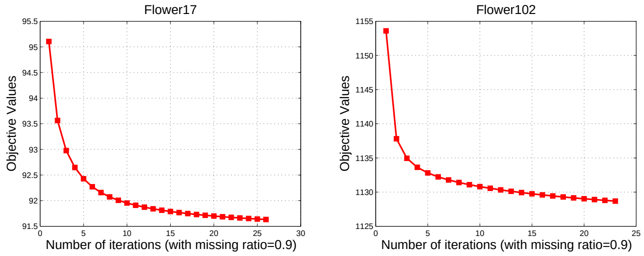


Figure 3: Evolution of the objective value in our algorithm.

Conclusion

While MKC algorithms have recently demonstrated promising performance in various applications, they are not able to effectively handle the scenario where base kernels are incomplete. This paper proposes to jointly optimize the kernel imputation and clustering to address this issue. It makes these two learning procedures seamlessly integrated to achieve better clustering. The proposed algorithm effectively solves the resultant optimization problem, and it demonstrates well improved clustering performance via extensive experiments on benchmark data sets, especially when the missing ratio is high. In the future, we plan to further improve the clustering performance by considering the correlations of different base kernels (Bhadra, Kaski, and Rousu 2016).

Acknowledgements

This work was supported by the National Natural Science Foundation of China (project No. U1435219, 61403405, 61672528 and 61232016). The authors would like to thank Dr. Luping Zhou for discussions on the application of our algorithm to the Alzheimers disease prediction.

References

Bezdek, J. C., and Hathaway, R. J. 2003. Convergence of alternating optimization. *Neural, Parallel Sci. Comput.* 11(4):351–368.

Bhadra, S.; Kaski, S.; and Rousu, J. 2016. Multi-view kernel completion. In *arXiv:1602.02518*.

Cao, X.; Zhang, C.; Fu, H.; Liu, S.; and Zhang, H. 2015a. Diversity-induced multi-view subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 586–594.

Cao, X.; Zhang, C.; Zhou, C.; Fu, H.; and Foroosh, H. 2015b. Constrained multi-view video face clustering. *IEEE Trans. Image Processing* 24(11):4381–4393.

Cortes, C.; Mohri, M.; and Rostamizadeh, A. 2009. L2 regularization for learning kernels. In *UAI*, 109–116.

Cortes, C.; Mohri, M.; and Rostamizadeh, A. 2012. Algorithms for learning kernels based on centered alignment. *JMLR* 13:795–828.

Du, L.; Zhou, P.; Shi, L.; Wang, H.; Fan, M.; Wang, W.; and Shen, Y.-D. 2015. Robust multiple kernel k-means clustering using ℓ_{21}-norm. In *IJCAI*, 3476–3482.

Ghahramani, Z., and Jordan, M. I. 1993. Supervised learning from incomplete data via an EM approach. In *NIPS*, 120–127.

Gönen, M., and Margolin, A. A. 2014. Localized data fusion for kernel k-means clustering with application to cancer biology. In *NIPS*, 1305–1313.

Jegelka, S.; Gretton, A.; Schölkopf, B.; Sriperumbudur, B. K.; and von Luxburg, U. 2009. Generalized clustering via kernel embeddings. In *KI 2009: Advances in Artificial Intelligence, 32nd Annual German Conference on AI*, 144–152.

Kloft, M.; Brefeld, U.; Sonnenburg, S.; Laskov, P.; Müller, K.; and Zien, A. 2009. Efficient and accurate lp-norm multiple kernel learning. In *NIPS*, 997–1005.

Kloft, M.; Brefeld, U.; Sonnenburg, S.; and Zien, A. 2011. l_p-norm multiple kernel learning. *JMLR* 12:953–997.

Kumar, R.; Chen, T.; Hardt, M.; Beymer, D.; Brannon, K.; and Syeda-Mahmood, T. F. 2013. Multiple kernel completion and its application to cardiac disease discrimination. In *ISBI*, 764–767.

Li, M.; Liu, X.; Wang, L.; Dou, Y.; Yin, J.; and Zhu, E. 2016. Multiple kernel clustering with local kernel alignment maximization. In *IJCAI*, 1704–1710.

Liu, X.; Wang, L.; Yin, J.; Zhu, E.; and Zhang, J. 2013. An efficient approach to integrating radius information into multiple kernel learning. *IEEE Trans. Cybernetics* 43(2):557–569.

Liu, X.; Wang, L.; Zhang, J.; and Yin, J. 2014. Sample-adaptive multiple kernel learning. In *AAAI*, 1975–1981.

Liu, X.; Wang, L.; Yin, J.; Dou, Y.; and Zhang, J. 2015. Absent multiple kernel learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, 2807–2813.

Liu, X.; Dou, Y.; Yin, J.; Wang, L.; and Zhu, E. 2016. Multiple kernel k-means clustering with matrix-induced regularization. In *AAAI*, 1888–1894.

Rakotomamonjy, A.; Bach, F. R.; Canu, S.; and Grandvalet, Y. 2008. Simplemkl. *JMLR* 9:2491–2521.

Shao, W.; He, L.; and Yu, P. S. 2015. Multiple incomplete views clustering via weighted nonnegative matrix factorization with $\ell_{2,1}$ regularization. In *ECML PKDD*, 318–334.

Trivedi, A.; Rai, P.; Daumé III, H.; and DuVall, S. L. 2010. Multiview clustering with incomplete views. In *NIPS 2010: Machine Learning for Social Computing Workshop, Whistler, Canada*.

Xiang, S.; Yuan, L.; Fan, W.; Wang, Y.; Thompson, P. M.; and Ye, J. 2013. Multi-source learning with block-wise missing data for alzheimer's disease prediction. In *ACM SIGKDD*, 185–193.

Xu, C.; Tao, D.; and Xu, C. 2015. Multi-view learning with incomplete views. *IEEE Trans. Image Processing* 24(12):5812–5825.

Yu, S.; Tranchevent, L.-C.; Liu, X.; Glänzel, W.; Suykens, J. A. K.; Moor, B. D.; and Moreau, Y. 2012. Optimized data fusion for kernel k-means clustering. *IEEE TPAMI* 34(5):1031–1039.

Zhang, C.; Fu, H.; Liu, S.; Liu, G.; and Cao, X. 2015. Low-rank tensor constrained multiview subspace clustering. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 1582–1590.

Zhang, C.; Fu, H.; Hu, Q.; Zhu, P.; and Cao, X. 2016. Flexible multi-view dimensionality co-reduction. *IEEE Trans. Image Processing*.

Zhao, B.; Kwok, J. T.; and Zhang, C. 2009. Multiple kernel clustering. In *SDM*, 638–649.

Zhao, H.; Liu, H.; and Fu, Y. 2016. Incomplete multimodal visual data grouping. In *IJCAI*, 2392–2398.