Optimized data fusion for kernel k-means clustering

Shi Yu, Léon-Charles Tranchevent, Xinhai Liu, Wolfgang Glänzel, Johan A. K. Suykens, *Senior Member, IEEE*, Bart De Moor, *Fellow, IEEE,* and Yves Moreau

Abstract—This paper presents a novel optimized kernel k-means algorithm (OKKC) to combine multiple data sources for clustering analysis. The algorithm uses an alternating minimization framework to optimize the cluster membership and kernel coefficients as a non-convex problem. In the proposed algorithm, the problem to optimize the cluster membership and the problem to optimize the kernel coefficients are all based on the same Rayleigh quotient objective, therefore the proposed algorithm converges locally. OKKC has a simpler procedure and lower complexity than other algorithms proposed in the literature. Simulated and real-life data fusion applications are experimentally studied, and the results validate that the proposed algorithm has comparable performance, moreover, it is more efficient on large scale data sets. [1]

Index Terms—Clustering, data fusion, multiple kernel learning, Fisher discriminant analysis, least squares support vector machine

1 Introduction

We present a novel optimized kernel k-means clustering (OKKC) algorithm to combine multiple data sources. The objective of k-means clustering is formulated as a Rayleigh quotient function of the between-cluster scatter and the cluster membership matrix and further combined with nonlinear dimensionality reduction in Hilbert space, where heterogeneous data sources can be easily combined as kernel matrices. The objective to optimize the kernel combination and the cluster memberships on unlabeled data is non-convex. To solve it, we apply an alternating minimization method to optimize the cluster memberships and the kernel coefficients iteratively to convergence. When the cluster membership is given, we optimize the kernel coefficients as kernel Fisher discriminants (KFD) using least squares support vector machine (LS-SVM). The objectives of KFD and k-means are combined in a unified model thus the two components optimize towards the same objective, therefore, the proposed alternating algorithm solving this objective converges locally.

Our algorithm has the same motivation as Lange and Buhmann's approach [25] to learn the optimal combination of multiple information sources as similarity matrices (kernel matrices). However, the two algorithmic approaches are different. Lange and Buhmann's algorithm uses non-negative matrix factorization to maximize posteriori estimates of data point assignments to partitions. To combine the similarity matrices, a cross-entropy objective is minimized to seek a good factorization and the weights assigned on similarity matrices are optimized. Our proposed algorithm is related to the Nonlinear Adaptive Metric Learning (NAML) algorithm proposed for clustering [8]. Although NAML is also based on multiple kernel extension of k-means clustering, the mathematical objective and the solution are different from OKKC. In NAML, the metric of k-means is constructed based on the Mahalanobis distance. NAML optimizes the objective iteratively at three levels: the cluster assignments, the kernel coefficients and the projection in the Representer Theorem. The k-means objective in our approach is constructed in Euclidean space and the algorithm optimizes the cluster assignments and kernel coefficients in a bi-level procedure. Moreover, we formulate the least squares dual problem of kernel coefficient learning as semi-infinite programming (SIP) [19], which is much more efficient and scalable than the quadratic constraint quadratic programming (QCQP) [5] formulation adopted in NAML. The cluster assignments of data points are relaxed as numerical values and optimized as the eigenspectrum of the combined kernel matrix. To avoid the over-sparseness in combining data sources resulted from L_1 regularization, we optimize the coefficients by regularizing different norms in multiple kernel combination.

The proposed method extends the idea of Multiple Kernel Learning to unsupervised problem. Relevant

- *S. Yu is with Institute of Genomics and Systems Biology, University of Chicago, Chicago, IL 60637, US. E-mail: shee.yu@gmail.com*
- *L.C. Tranchevent, X. Liu, J.A.K. Suykens, B.D. Moor, and Y. Moreau are with Department of Electrical Engineering, ESAT-SCD, and IBBT-K.U.Leuven Future Health Department, Katholieke Universiteit Leuven, Leuven, B3001, Belgium.*
- *X. Liu is also with Department of Information Science and Engineering & ERCMAMT, Wuhan University of Science and Technology, Wuhan, China.*
- *W. Glänzel is with Department of Managerial Economics, Strategy and Innovation, Centre for R & D Monitoring (ECOOM), Katholieke Universiteit Leuven, Leuven, B3000, Belgium.*

1. The Matlab implementation of OKKC algorithm is downloadable on http://homes.esat.kuleuven.be/~sistawww/bioi/syu/okkc.html

works about clustering with multiple data sources are proposed in literature, *e.g.,* Strehl and Gosh 's work about cluster ensembles [40], Zhou and Burges formulate a multi-view spectral clustering model as mixture of Markov chains [50], Tang *et al.* propose a method clustering multiple graphs using *linked matrix factorization* [41], and Chaudhuri explore clusters in the correlated projections of multiple data sources use Canonical Correlation Aanalysis [7]. However, these approaches are fundamentally different from ours because their mixture coefficients of data sources are either selected empirically or optimized implicitly.

The paper is organized as follows. Section 2 introduces the objective of k-means clustering. Section 3 formulates the problem and introduces the algorithm to solve the objective. The description of experimental data and analysis of results are presented in Section 4. Conclusion and future work are mentioned in Section 5.

2 Objective of k-means clustering

In k-means clustering, a number of k prototypes are used to characterize the data and the partitions $\{\mathcal{C}_j\}_{j=1,...,k}$ are determined by minimizing the distortion as

$$\min \sum_{j=1}^{k} \sum_{\vec{x}_i \in \mathcal{C}_j} ||\vec{x}_i - \vec{\mu}_j||^2, \tag{1}$$

where $\vec{x}_i$ is the i-th data sample, $\vec{\mu}_j$ is the prototype (mean) of the j-th partition $\mathcal{C}_j$, k is the number of partitions (usually predefined). It is known that (1) is equivalent to the trace maximization of the between-cluster scatter S_b [42][22]

$$\max_{a_{ij}} \text{trace } S_b, \tag{2}$$

where a_{ij} is the hard cluster assignment $a_{ij} \in \{0,1\}, \sum_{j=1}^{k} a_{ij} = 1$ and

$$S_b = \sum_{j=1}^{k} n_j (\vec{\mu}_j - \vec{\mu}_0)(\vec{\mu}_j - \vec{\mu}_0)^T, \tag{3}$$

where $\vec{\mu}_0$ is the global mean, $n_j = \sum_{i=1}^{N} a_{ij}$ is the number of samples in $\mathcal{C}_j$. Without loss of generality, we assume that the data $X \in \mathbb{R}^{M \times N}$ has been centered such that the global mean is $\vec{\mu}_0 = 0$. To express $\vec{\mu}_j$ in terms of X, we denote a discrete cluster membership matrix $A \in \mathbb{R}^{N \times K}$ as

$$A_{ij} = \begin{cases} \frac{1}{\sqrt{n_j}} & \text{if } \vec{x}_i \in C_j \\ 0 & \text{if } \vec{x}_i \notin C_j, \end{cases} \tag{4}$$

then $A^T A = I_k$ and the objective of k-means in (2) can be equivalently written as [49]

$$\begin{aligned} &\max_{A} \ \text{trace}\big(A^T X^T X A\big), \\ &\text{s.t.} \ \ A^T A = I_k, \quad A_{ij} \in \{0, \frac{1}{\sqrt{n_j}}\}. \end{aligned} \tag{5}$$

The discrete constraint in (5) makes the problem NP-hard to solve [16]. In literature, various methods have been proposed to the problem, such as the iterative descent method [18], the expectation-maximization method [4], the spectral relaxation method [49], the probabilistic latent variable models [34] and many others. In particular, the spectral relaxation method relaxes the discrete cluster memberships of A to numerical values, denoted as $\tilde{A}$, thus (5) is transformed to [49]

$$\begin{aligned} &\max_{\tilde{A}} \ \text{trace}\big(\tilde{A}^T X^T X \tilde{A}\big), \\ &\text{s.t.} \ \ \tilde{A}^T \tilde{A} = I_k, \quad \tilde{A}_{ij} \in \mathbb{R}. \end{aligned} \tag{6}$$

If $\tilde{A}$ is single column (binary cluster membership in A), (9) is exactly a Rayleigh quotient and the optimal $\tilde{A}^*$ is given by the eigenvector $\vec{u}_{max}$ in the largest eigenvalue pair $\{\lambda_{max}, \vec{u}_{max}\}$ of $X^T X$. If $\tilde{A}$ is a matrix (multi-cluster memberships in A), according to the Ky Fan [12] (more formal mathematical proofs available in [3], [38]), let the eigenvalues of $X^T X$ be ordered as $\lambda_{max} = \lambda_1 \geq , ..., \geq \lambda_N = \lambda_{min}$ and the corresponding eigenvectors as $\vec{u}_1, ..., \vec{u}_N$, then the optimal $\tilde{A}^*$ is given by $U_k V$, where $U_k = [\vec{u}_1, ..., \vec{u}_k]$, and V is an arbitrary $k \times k$ orthogonal matrix, and $\max \text{trace}\big(U^T X^T X U\big) = \ \lambda_1 + .. + \lambda_k$. Thus, for a given cluster number k, the k-means can be solved as an eigenvalue problem and the discrete cluster memberships of the original A can be recovered using the iterative descend k-means method on $\tilde{A}^*$ or QR decomposition [49].

To cluster data in nonlinear space, the objective in (6) can be generalized using the feature map $\phi(\cdot) : \mathbb{R} \to \mathcal{F}$ on X, then the centered data in Hilbert space $\mathcal{F}$ is denoted as X^Φ, given by

$$X^\Phi = [\phi(\vec{x}_1) - \vec{\mu}_0^\Phi, \phi(\vec{x}_2) - \vec{\mu}_0^\Phi, ..., \phi(\vec{x}_N) - \vec{\mu}_0^\Phi], \tag{7}$$

where $\phi(\vec{x}_i)$ is the feature map applied on the column vector of the i-th data point in $\mathcal{F}$, μ_0^Φ is the global mean in $\mathcal{F}$. The inner product $X^T X$ corresponds to $X^{\Phi T} X^\Phi$ in Hilbert space and can be combined using the kernel trick $\kappa(\vec{x}_u, \vec{x}_v) = \phi(\vec{x}_u)^T \phi(\vec{x}_v)$, where $\kappa(\cdot, \cdot)$ is a Mercer kernel. We denote G as the centered kernel matrix as $G = PKP$, where P is the centering matrix $P = I_N - (1/N)\vec{1}_N \vec{1}_N^T$, I_N is the $N \times N$ identity matrix, $\vec{1}_N$ is a column vector of N ones. Note that the trace of between-cluster scatter $\text{trace}(S_b^\Phi)$ takes the form of a series of dot products in the centered Hilbert space. Rewriting the dot products into Mercer kernel, we have [35]

$$\text{trace}\big(S_b^\Phi\big) = \text{trace}\big(A^T G A\big). \tag{8}$$

To incorporate multiple data sources (kernels), we assume that $X_1, ..., X_p$ are p different representations of the same N objects. We extend the clustering problem from single data set to multiple data sets by combining multiple centered kernel matrices $G_r, \ (r = 1, ..., p)$ in a parametric linear additive manner as

$$\mathbf{\Omega} = \left\{ \sum_{r=1}^{p} \theta_r G_r \ \middle| \ \forall \theta_r \geq 0, \ \sum_{r=1}^{p} \theta_r^\delta = 1 \right\}, \tag{9}$$

where θ_r are coefficients of the kernel matrices, δ is a parameter determining the norm of constraint posed on coefficients (*e.g.*, see relevant L_2, L_p-norm MKL work [23], [48]), G_r are normalized kernel matrices [33] centered in the Hilbert space. Kernel normalization ensures that $\phi(\vec{x}_i)^T\phi(\vec{x}_i) = 1$ thus makes the kernels comparable to each other. The k-means objective in (8) is thus extended to $\mathcal{F}$ and multiple data sets are incorporated, given by

$$\boxed{\text{Q1:}} \quad \max_{A,\vec{\theta}} \mathcal{J}_{Q1} = \text{trace}\big(A^T\mathbf{\Omega}A\big), \tag{10}$$
$$\text{s.t.} \;\; A^TA = I_k, \;\; A_{ij} \in \{0, \frac{1}{\sqrt{n_j}}\},$$
$$\mathbf{\Omega} = \sum_{r=1}^{p} \theta_r G_r,$$
$$\theta_r \geq 0, \;\; r = 1, ..., p$$
$$\sum_{r=1}^{p} \theta_r^{\delta} = 1.$$

3 Bi-level optimization of k-means on multiple kernels

The objective in (10) is difficult to be optimized analytically because the data is unlabeled, moreover, the discrete cluster memberships make the problem NP hard. Our strategy is to optimize the two parameters iteratively (the same spirit as the EM algorithm optimizing the latent variables iteratively). Notice that A represents the cluster membership and $\vec{\theta}$ determines the coefficients of data sources, we could maximize $\mathcal{J}_{Q1}$ with respect to A, keeping $\vec{\theta}$ fixed (as a single data set clustering problem). In the second phase we maximize $\mathcal{J}_{Q1}$ with respect to $\vec{\theta}$, keeping A fixed (as a supervised learning MKL problem on labeled data). Care must be exercised when $\delta = 1$ because the optimization may pick a single scatter who has the largest trace thus it may result in a trivial solution clustering a single data source, known as the sparse solution. In data integration, the sparseness is useful to distinguish relevant sources from a large number of irrelevant data sources. However, in some applications, there are usually a small number of sources and most of these data sources are carefully selected and preprocessed. They thus often are directly relevant to the problem. In these cases, a sparse solution may be too selective to thoroughly combine the complementary information in the data sources. While the performance on benchmark data may be good, the selected sources may not be as strong on truly novel problems in unsupervised learning where the quality of the information is much lower. We may thus expect the performance of such solutions to degrade significantly on actual real-world applications. A traditional solution to avoid sparseness in integration is posing additional regularization, *e.g.*, an entropy term, in the object function. However, in that case one needs to estimate an additional coefficient posed on the regularization term. In our approach, we resolve this issue by setting the δ parameter to positive numbers other than 1 which yields non-sparse solution in kernel combination. Next, we will show that when the memberships are given, the problem in Q1 can be transformed as kernel Fisher discriminant (KFD) in $\mathcal{F}$.

3.1 Optimizing the kernel coefficients as simplified KFD

Given a single data set and labels of two classes, to find the linear discriminant in $\mathcal{F}$ we need to maximize

$$\max_{\vec{w}} \frac{\vec{w}^T S_b^{\Phi} \vec{w}}{\vec{w}^T (S_w^{\Phi} + \rho I)\vec{w}} \;\; , \tag{11}$$

where $\vec{w}$ is the non-linear projection in $\mathcal{F}$, S_b^{Φ} and S_w^{Φ} are respectively the between-class and the within-class scatters in $\mathcal{F}$, ρ is the regularization term to ensure the positive definiteness of the denominator. For k multiple classes, denote $W = [\vec{w}_1, ..., \vec{w}_k]$ as the matrix where each column corresponds to the discriminative direction of 1vsA classes. Based on Representer Theorem [36], the projection is in the span of the images of data points in $\mathcal{F}$ thus $\vec{w} = \sum_i^N q_i \phi(\vec{x}_i)$. Following the derivations of Mika *et al.* [31], we replace $\vec{w}$ with $\vec{q}$, transform the dot products by the kernel function and rewrite (11) in its dual form:

$$\max_{\vec{q}} \frac{\vec{q}^T \Gamma_B \vec{q}}{\vec{q}^T (\Gamma_W + \rho I)\vec{q}} \;, \tag{12}$$

where $\Gamma_B = GAA^TG$ as the matrix representation of between-class scatter in Hilbert space, $\Gamma_W = GG - GAA^TG$ is the within-class scatter [6], [33]. Analogously, we could extend the one-dimensional optimal projection to a space spanned by $Q = [\vec{q}_1, ..., \vec{q}_k]$ and formulate the multi-class objective as

$$\max_{Q} \text{trace} \; \big[Q^T(\Gamma_W + \rho I)Q\big]^{-1}\big(Q^T\Gamma_B Q\big). \tag{13}$$

Various solutions are available to solve (13) and yield different KFD variants. In our approach, we adopt a simple criterion assuming that the projection of within-cluster scatter is a constant value [18], [20]. In other words, if the within-class scatter is isotropic, the norm vectors of discriminant projections are merely the eigenvectors of the between-class scatter [14]. Thus we only need to optimize Q over Γ_B. If we let $Q \in \mathbb{R}^{N\times k}$ be any matrix with full column rank, then, essentially, there is no upperbound and maximization is also meaningless. Therefore, we restrict the solution to the case when Q has orthonormal columns [20]. Then, there exists $\hat{Q} \in \mathbb{R}^{N\times(N-k)}$ such that $\mathcal{Q} = \big(Q, \hat{Q}\big)$ is an orthogonal matrix. Furthermore, because Γ_B is positive semi-definite, we have

$$\begin{aligned} \text{trace}\big(Q^T\Gamma_B Q\big) &\leq \text{trace}\big(Q^T\Gamma_B Q\big) + \text{trace}\big(\hat{Q}^T\Gamma_B\hat{Q}\big) \\ &= \text{trace}\big(\mathcal{Q}^T\Gamma_B\mathcal{Q}\big) \\ &= \text{trace}\big(\Gamma_B\big). \end{aligned} \tag{14}$$

Notice that the right side term in (14) is exactly the objective of clustering, and the left side term is its lower

bound as a simplified KFD objective. Therefore, instead of maximizing $\text{trace}(\Gamma_B)$ composed by multiple kernels which may stuck in a trivial solution, we maximize its lower bound via KFD. According to the proof of Rayleigh quotient, the bound is tight if we take the leading k eigenvectors of Γ_B as Q.

The model in (14) is also known as the *Kernel Orthogonal Centroid* [20] and is applied to dimension reduction based clustering in kernel space [32]. Similar strategy is also used in probabilistic clustering modeling to estimate the latent variables in an orthogonal space of dimensionality reduction [34]. Other assumptions different from (14) are also proposed, for example, assuming that the projections of total scatter Γ_T are orthogonal to each other $Q^T\Gamma_T Q = I$ which is related to *Uncorrelated linear discriminant analysis* [26], [27]; optimizing the between-class scatter and within-class scatter simultaneously, which yields a standard KFD criterion and a general Rayleigh quotient. All these alternative KFD criteria and constraints could be easily extended to multiple data sources using the similar model proposed in this paper. The reason of preferring (14) in our approach is that we have a simple model. Combining (10) and (14) together, the complete objective of the proposed algorithm in Hilbert space is

$$\boxed{\text{Q2:}} \quad \max_{A,\vec{\theta}} \mathcal{J}_{Q2} = \text{trace}\big(\mathcal{Q}^T\mathbf{\Omega}AA^T\mathbf{\Omega}\mathcal{Q}\big), \tag{15}$$

$$\begin{aligned} \text{s.t.} \quad & A^TA = I_k, \quad A_{ij} \in \{0, \frac{1}{\sqrt{n_j}}\} \\ & \mathcal{Q}^T\mathcal{Q} = I_N, \quad \mathcal{Q} \in \mathbb{R}^{N\times N} \\ & \mathbf{\Omega} = \sum_{r=1}^{p} \theta_r G_r, \\ & \theta_r \geq 0, \quad r = 1, ..., p \\ & \sum_{r=1}^{p} \theta_r^\delta = 1. \end{aligned}$$

Notice that $\mathcal{Q}$ is real orthogonal matrix so it is also unitary, thus $\mathcal{Q}^T\mathcal{Q} = \mathcal{Q}\mathcal{Q}^T = I_N$. Then, $\mathcal{Q}$ actually has no effect in our objective because (we drop the constraints for simplicity)

$$\begin{aligned} \max_A \mathcal{J}_{Q2} &= \text{trace}\big(\mathcal{Q}^T\mathbf{\Omega}AA^T\mathbf{\Omega}\mathcal{Q}\big) \\ &= \text{trace}\big(A^T\mathbf{\Omega}\mathcal{Q}\mathcal{Q}^T\mathbf{\Omega}A\big) \\ &= \text{trace}\big(\Gamma_B\big). \end{aligned} \tag{16}$$

As seen, when assuming projections are orthogonal, the proposed clustering method does not really consider either the lower dimensional projection Q obtained in KFD or $\mathcal{Q}$, in contrast, it only updates $\vec{\theta}$ as the new combination of multiple kernels for the next clustering iteration. The reason of keeping $\mathcal{Q}$ is merely to emphasize the objective function as a bi-level Rayleigh quotient: the inner Rayleigh quotient yields cluster assignments and the outer quotient yields mixture coefficients. With respect to the first Rayleigh quotient, A is not unitary because it is discrete and AA^T is a block diagonal matrix. Though spectral relaxation, $\tilde{A}$ in (6) becomes unitary therefore firstly we solve $\tilde{A}$ by taking the dominant eigenvectors of $\mathbf{\Omega\Omega}$. Next, we obtain the discrete cluster assignments A via QR decomposition or k-means on $\tilde{A}$ [49].

Notice that if we do not assume that $\mathcal{Q}$ is unitary, the objective in (15) is still solvable where the only difference is the clustering step involves the update of $\mathcal{Q}$. Moreover, since the projection matrix contains dual variables, if the KFD step involving multiple kernels is properly modeled as a convex problem and solved as a dual, one can obtain $\mathcal{Q}$ and $\vec{\theta}$ directly thus the overall algorithm still has a bi-level structure.

Concerning the second Rayleigh quotient, Γ_B is fixed when A is given, the goal is to maximize the trace of $\mathcal{Q}^T\Gamma_B\mathcal{Q}$. As mentioned before, we optimize its tight lowerbound via KFD. It is known there is a close connection between Fisher Discriminant Analysis and the least squares problem [14]. Moreover, KFD is related to the least squares formulation of SVM [31], known as least squares SVM (LS-SVM) proposed by Suykens *et al.* [39]. Notice that LS-SVM also solves a simplified KFD problem by taking the squared error in the SVM cost function which corresponds to minimizing solely the within-class scatter [39]. To optimize the fusion of multiple kernels, we model LS-SVM as multiple kernel learning. The orthogonal constraints on Q corresponds to constraints in LS-SVM forcing the orthogonality of dual variables in multi-class classification. Notice that with the orthogonal constraint, the problem is closely related to the *high-order orthogonal iteration* in tensor methods [10], which recently has also been applied to combine multiple matrices for clustering.

3.2 The role of cluster assignment

It is worth clarifying the transformations of cluster assignment in the proposed algorithm. In problem Q2, we first maximize $\mathcal{J}_{Q2}$ using the fixed $\vec{\theta}$ to obtain $\tilde{A}$. From $\tilde{A}$ we obtain the discrete weighted cluster indicator matrix A, which is regarded as the one-vs-others (1vsA) coding of the cluster assignments because each column of A actually distinguishes one cluster from the other clusters. When A is given, the between-cluster scatter Γ_B is fixed, thus the problem of optimizing the coefficients of multiple kernel matrices is equivalent to optimizing a KFD [31] problem using multiple kernel matrices. To transform A to class labels as the input of KFD, we define F, given by

$$F_{ij} = \begin{cases} +1 & \text{if } A_{ij} > 0, \quad i = 1, ..., N, \ j = 1, ..., k \\ -1 & \text{if } A_{ij} = 0, \quad i = 1, ..., N, \ j = 1, ..., k \end{cases}, \tag{17}$$

as an affinity matrix using $\{+1, -1\}$ to discriminate the cluster assignments. In the second iteration step, to maximize $\mathcal{J}_{Q2}$ with respect to $\vec{\theta}$, we formulate it as the optimization of LS-SVM on multiple kernel matrices using the affinity matrix F as input.

3.3 Solving the simplified KFD as LS-SVM using multiple kernels

In LS-SVM, the cost function of the classification error is defined as a least squares term [39] and the inequalities in the constraint are replaced by equalities, given by

$$\begin{aligned} \min_{\vec{w},b,\vec{e}} \quad & \frac{1}{2}\vec{w}^T\vec{w} + \frac{1}{2}\lambda\vec{e}^T\vec{e} \\ \text{s.t.} \quad & y_i[\vec{w}^T\phi(\vec{x}_i) + b] = 1 - e_i, \quad i = 1, ..., N, \end{aligned} \tag{18}$$

where $\vec{w}$ is the norm vector of separating hyper-plane, $\vec{x}_i$ are data samples, $\phi(\cdot)$ is the feature map, y_i are the cluster assignments represented in the affinity function F, $\lambda > 0$ is a positive regularization parameter, $\vec{e}$ are the least squares error terms. The squared error in the cost function of LS-SVM corresponds to minimizing the within-class scatter for class label +1 and -1. Taking the conditions for optimality from the Lagrangian, eliminating $\vec{w}, \vec{e}$, defining $\vec{y} = [y_1, ..., y_N]^T$ and $Y = diag(y_1, ..., y_N)$, one obtains the following linear system [39]:

$$\left[\begin{array}{c|c} 0 & \vec{y}^T \\ \hline \vec{y} & YKY + I/\lambda \end{array}\right] \left[\begin{array}{c} b \\ \hline \vec{\alpha} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \vec{1} \end{array}\right], \tag{19}$$

where $\vec{\alpha}$ are unconstrained dual variables, K is the kernel matrix obtained by kernel trick as $\kappa(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i)^T\phi(\vec{x}_j)$. Without loss of generality, we denote $\vec{\beta} = Y\vec{\alpha}$ such that (19) becomes

$$\left[\begin{array}{c|c} 0 & \vec{1}^T \\ \hline \vec{1} & K + Y^{-2}/\lambda \end{array}\right] \left[\begin{array}{c} b \\ \hline \vec{\beta} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline Y^{-1}\vec{1} \end{array}\right]. \tag{20}$$

To incorporate multiple kernels for multiple classes, we follow the approaches of Lanckriet *et al.* [24] and Ye *et al.* [45] formulating the LS-SVM MKL as a QCQP problem. From now on, we restrict the discussion to binary class for simplicity because in QCQP modeling the extension from binary class to multiple classes is straightforward. Notice that the δ parameter regularizes the norm of coefficients in $\vec{\theta}$ to avoid sparse solution of data fusion. According to [48], the δ parameter in the primal problem corresponds to υ in the dual problem under the constraints $\frac{1}{\delta} + \frac{1}{\upsilon} = 1$. Since $\delta \geq 1$ thus υ can be ∞ or any values from 1 to 2. The complete QCQP formulation for LS-SVM MKL is given by (see [48] for complete proof)

$$\begin{aligned} \min_{\vec{\beta},t} \quad & \frac{1}{2}t + \frac{1}{2\lambda}\vec{\beta}^T\vec{\beta} - \vec{\beta}^T Y^{-1}\vec{1} \\ \text{s.t.} \quad & \sum_{i=1}^{N}\beta_i = 0, \\ & t \geq ||\vec{g}||_\upsilon, \ \upsilon = \infty \ \text{ or } \ \upsilon \in [1, 2] \\ & \vec{g} = [\vec{\beta}^T K_1\vec{\beta}, ..., \vec{\beta}^T K_p\vec{\beta}]^T. \end{aligned} \tag{21}$$

In particular, it is worth noticing that a discriminant analysis model on multiple kernels is proposed in [46]. In their work, their model is exactly derived on the basis of KFD, and the solution is given by a QCQP (equation (34) in [46]), which is exactly equivalent to (21). Therefore, the equivalence between KFD and LS-SVM has been mathematically proven.

Notice that in (21) when $\upsilon = \infty$, $\delta = 1$ thus the primal problem is regularized by the L_1-norm, which is more likely to yield sparse solution of data fusion (a single data source takes dominant weights). Setting υ between 1 and 2 can avoid the sparse solution and may perform better on specific problems. In clustering, the kernels are preprocessed using the *kernel centering* [33] and centered for all samples thus K_r is equal to G_r. The kernel coefficients θ_r correspond to the dual variables bounded by the L_υ-norm constraint in (21). The column vector of F, denoted as F_j, $j = 1, ..., k$ correspond to the k number of $Y_1, ..., Y_k$ in (20), where $Y_j = diag(F_j)$, $j = 1, ..., k$. The bias term b can be solved independently using the optimal $\vec{\beta}^*$ and the optimal $\vec{\theta}^*$, thus can be dropped out from (21). To solve (21), we decompose it as iterations of the master problem as optimizing the kernel coefficients and a slave problem as a single kernel SVM learning [37], known as SIP formulation of SVMs. Therefore, for the LS-SVM MKL problem presented in (24), in SIP formulation it corresponds to iterations of an unconstrained QP problem, which can be solved as a linear system, and a coefficient optimization problem, which is also a small linear system if $\delta = 1$ or a small relaxed convex problem if $\delta > 1$.

In supervised learning, the regularization term λ of LS-SVM is often optimized on the validation data. To tackle the problem, we transform the effect of regularization as an identity kernel matrix in $\frac{1}{2}\vec{\beta}^T\left(\sum_{r=1}^{p}\theta_r G_r + \theta_{p+1}I\right)\vec{\beta}$, where $\theta_{p+1} = 1/\lambda$. Then the problem of combining p kernels with the regularization parameter is equivalent to combining $p+1$ kernels without regularization parameter where the last kernel is an identity matrix with the optimal coefficient corresponding to $1/\lambda$. This method has been mentioned by Lanckriet *et al.* [24] to tackle the estimation of the regularization parameter in the soft margin SVM. It has also been used by Ye *et al.* [46] to jointly estimate the optimal kernel for discriminant analysis. Concluding the previous discussion, the SIP formulation of the LS-SVM MKL is given by (notice that now $\vec{\theta}$ is regularized by δ as a primal problem)

$$\begin{aligned} \max_{\vec{\theta},u} \quad & u \\ \text{s.t.} \quad & \theta_r \geq 0, \ r = 1, ..., p+1 \\ & \sum_{r=1}^{p+1}\theta_r^\delta \leq 1, \quad \delta \geq 1 \\ & \sum_{r=1}^{p+1}\theta_r f_r(\vec{\beta}) \geq u, \quad \forall\vec{\beta} \\ & f_r(\vec{\beta}) = \sum_{q=1}^{k}\left(\frac{1}{2}\vec{\beta}_q^T G_r\vec{\beta}_q - \vec{\beta}_q^T Y_q^{-1}\vec{1}\right), \quad r = 1, ..., p+1. \end{aligned} \tag{22}$$

The pseudocode to solve the LS-SVM MKL in (22) is presented in Algorithm 3.1. $G_1, ..., G_p$ are centered

kernel matrices of multiple sources, an identity matrix is set as G_{p+1} to estimate the regularization parameter, $Y_1, ..., Y_k$ are the $N \times N$ diagonal matrices constructed from F. The ε is a fixed constant as the stopping rule of SIP iterations and is set empirically as 0.0001 in our implementation. Normally the SIP takes about ten iterations to converge. In Algorithm 3.1, Step 1 optimizes $\vec{\theta}$ as a linear programming and Step 3 is simply a linear problem as

$$\left[\begin{array}{c|c} 0 & \vec{1}^T \\ \hline \vec{1} & \Omega^{(\tau)} \end{array}\right] \left[\begin{array}{c} b^{(\tau)} \\ \hline \vec{\beta}^{(\tau)} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline Y^{-1}\vec{1} \end{array}\right], \qquad (23)$$

where $\Omega^{(\tau)} = \sum_{r=1}^{p+1} \theta_j^{(\tau)} G_r$.

Algorithm 3.1: SIP-LS-SVM-MKL($G_1, ..., G_p, F$)

Obtain the initial guess $\vec{\beta}^{(0)} = [\vec{\beta}_1^{(0)}, ..., \vec{\beta}_k^{(0)}]$
$\tau = 0$
while $(\Delta u > \varepsilon)$
do
- $step1$: Fix $\vec{\beta}$, solve $\vec{\theta}^{(\tau)}$then obtain $u^{(\tau)}$
- $step2$: Compute the kernel combination $\mathbf{\Omega}^{(\tau)}$
- $step3$: Solve the single LS-SVM for the optimal $\vec{\beta}^{(\tau)}$
- $step4$: Compute $f_1(\vec{\beta}^{(\tau)}), ..., f_{p+1}(\vec{\beta}^{(\tau)})$
- $step5$: $\Delta u = |1 - \frac{\sum_{j=1}^{p+1} \theta_j^{(\tau)} f_j(\vec{\beta}^{(\tau)})}{u^{(\tau)}}|$
- $step6$: $\tau := \tau + 1$

comment: τ is the indicator of the current loop
return $(\vec{\theta}^{(\tau)}, \vec{\beta}^{(\tau)})$

3.4 Optimized data fusion for kernel k-means Clustering (OKKC)

Now we have clarified the two algorithmic components to optimize the objective Q2 as defined in (15). The main characteristic is that the cluster assignments and the coefficients of kernels are optimized iteratively and adaptively until convergence. The coefficients assigned to multiple kernel matrices leverage the effect of different kernels in data integration to optimize the objective of clustering. The δ (υ) parameter further regularizes the sparsity of coefficients assigned to multiple kernels. Comparing to the average combination of kernel matrices, the optimized combination approach is more robust to noisy and irrelevant data sources. We name the proposed algorithm optimized kernel k-means clustering (OKKC) and its pseudocode is presented in Algorithm 3.2.

Algorithm 3.2: OKKC($G_1, G_2, ..., G_p, k$)

comment: Obtain the $\mathbf{\Omega}^{(\mathbf{0})}$ by the initial guess of $\vec{\theta}^{(0)}$
$\tilde{A}^{(0)} \leftarrow \text{PCA}(\mathbf{\Omega}^{(\mathbf{0})}\mathbf{\Omega}^{(\mathbf{0})}, k)$
$A^{(0)} \leftarrow \text{K-MEANS}(\tilde{A})$
$\gamma = 0$
while $(\Delta A > \epsilon)$
do
- $step1$: $F^{(\gamma)} \leftarrow A^{(\gamma)}$
- $step2$: $\mathbf{\Omega}^{(\gamma+1)}$ $\leftarrow$ SIP-LS-SVM-MKL$(G_1, G_2, ..., G_p, F^{(\gamma)})$
- $step3$: $\tilde{A}^{(\gamma+1)} \leftarrow \text{PCA}(\mathbf{\Omega}^{(\gamma+\mathbf{1})}\mathbf{\Omega}^{(\gamma+\mathbf{1})}, k)$
- $step4$: $A^{(\gamma+1)} \leftarrow \text{K-MEANS}(\tilde{A}^{(\gamma+1)})$ OR $A^{(\gamma+1)} \leftarrow \text{QR}(\tilde{A}^{(\gamma+1)})$
- $step5$: $\Delta A = ||A^{(\gamma+1)} - A^{(\gamma)}||^2 / ||A^{(\gamma+1)}||^2$
- $step6$: $\gamma := \gamma + 1$

return $(A^{(\gamma)}, \theta_1^{(\gamma)}, ..., \theta_p^{(\gamma)})$

3.5 Computational Complexity

The proposed OKKC algorithm has several advantages over some similar algorithms proposed in the literature. The optimization procedure of OKKC is bi-level, which is simpler than the tri-level architecture of the NAML algorithm. The kernel coefficients in OKKC is optimized as LS-SVM MKL, which can be solved efficiently as a convex SIP problem. When $\delta = 1$, the kernel coefficients are obtained as iterations of two linear systems: a single kernel LS-SVM problem and a linear problem to optimize the kernel coefficients. The time complexity of OKKC is $O\{\gamma[N^3 + \tau(N^2 + p^3)] + lkN^2\}$, where γ is the number of OKKC iterations, $O(N^3)$ is the complexity of eigenvalue decomposition, τ is the number of SIP iterations, the complexity of LS-SVM based on conjugate gradient method is $O(N^2)$, the complexity of optimizing kernel coefficients is $O(p^3)$, l is the fixed iteration of k-means clustering, p is the number of kernels, and $O(lkN^2)$ is the complexity of k-means to finally obtain the cluster assignment. In contrast, the complexity of NAML algorithm is $O\{\gamma(N^3 + N^3 + pk^2N^2 + pk^3N^3)\}$, where the complexities of obtaining cluster assignment and projection are all $O(N^3)$, the complexity of solving QCQP based problem is $O(pk^2N^2 + pk^3N^3)$, and k is the number of clusters. Obviously, the complexity of OKKC is much smaller than NAML because of the simplified KFD criterion and the SIP formulation of learning multiple kernels.

4 Experimental Results

The proposed algorithm is evaluated on public data sets and real application data to study the empirical performance. In particular, we systematically compare

it with the NAML algorithm on clustering performance, computational efficiency and the effect of data fusion.

TABLE 1
Summary of the data sets

Data set	Dimension	Instance	Class	Function	Nr. of kernels
iris	4	150	3	RBF	10
wine	13	178	3	RBF	10
yeast	17	384	5	RBF	10
satimage	36	480	6	RBF	10
pen digit	16	800	10	RBF	10
disease	-	620	2	-	9
GO	7403	620	2	linear	
MeSH	15569	620	2	linear	
OMIM	3402	620	2	linear	
LDDB	890	620	2	linear	
eVOC	1659	620	2	linear	
KO	554	620	2	linear	
MPO	3446	620	2	linear	
Uniprot	520	620	2	linear	
journal	669860	1424	7	linear	4

4.1 Data Sets and Experimental Settings

We adopt five data sets from the UCI machine learning repository and two data sets from real-life bioinformatics and scientometrics applications. The five UCI data sets are: Iris, Wine, Yeast, Satimage and Pen digit recognition. The original Satimage and Pen digit data contain a large amount of data points, so we sample 80 data points from each class and construct the data sets. For each data set, we generate ten RBF kernel matrices using different kernel widths σ in the RBF function $\kappa(\vec{x}_i, \vec{x}_j) = \exp(-||\vec{x}_i - \vec{x}_j||^2/2\sigma^2)$. We denote the average sample covariance of data set as c, then the σ values of the RBF kernels are respectively equal to $\{\frac{1}{4}c, \frac{1}{2}c, c, ..., 7c, 8c\}$. These ten kernel matrices are combined to simulate a kernel fusion problem for clustering analysis.

We also apply the proposed algorithm on data sets of two real applications. The first data set is taken from a bioinformatics application using biomedical text mining to cluster disease relevant genes [47]. We select controlled vocabularies (CVocs) from nine bio-ontologies for text mining and store the terms as bag-of-words respectively. The nine CVocs are used to index the title and abstract of around 290,000 human gene-related publications in MEDLINE to construct the doc-by-term vectors. According to the mapping of genes and publications in Entrez GeneRIF, the doc-by-term vectors are averagely combined as gene-by-term vectors, which are denoted as the term profiles of genes and proteins. The term profiles are distinguished by the bio-ontologies where the CVocs are selected and labeled as GO, MeSH, OMIM, LDDB, eVOC, KO, MPO, SNOMED and UniProtKB. Using these term profiles, we evaluate the performance of clustering a benchmark data set consisting of 620 disease relevant genes categorized in 29 genetic diseases. The numbers of genes categorized in the diseases are very imbalanced, moreover, some genes are simultaneously related to several diseases. To obtain meaningful clusters and evaluations, we enumerate all the pairwise combinations of the 29 diseases (406 combinations). In each run, the related genes of each paired diseases combination are selected and clustered into two groups, then the performance is evaluated using the disease labels. The genes related to both diseases in the paired combination are removed before clustering (totally there are less than 5% genes being removed). Finally, the average performance of all the 406 paired combinations is used as the overall clustering performance.

The second real-life data set is taken from a scientometric application [28]. The raw experimental data contains more than six million published papers from 2002 to 2006 (i.e., articles, letters, notes, reviews, etc.) indexed in the Web of Science (WoS) data based provided by Thomson Scientific. In our preliminary study of clustering of journal sets, the titles, abstracts and keywords of the journal publications are indexed by text mining program using no controlled vocabulary. The index contains 9,473,601 terms and we cut the Zipf curve [51] of the indexed terms at the head and the tail to remove the rare terms, stopwords and common words, which are known as usually irrelevant, also noisy for the clustering purpose. After the Zipf cut, 669,860 terms are used to represent the journal publications in vector space models where the terms are attributes and the weights are calculated by four weighting schemes: TF-IDF, IDF, TF and binary. The publication-by-term vectors are then aggregated to journal-by-term vectors as the representations of journal data. From the WoS database, we refer to the Essential Science Index (ESI) labels and select 1424 journals as the experimental data in this paper. The distributions of ESI labels of these journals are balanced because we want to avoid the affect of skewed distributions in cluster evaluation. In experiment, we cluster the 1424 journals simultaneously into 7 clusters and evaluate the results with the ESI labels.

We summarize the number of samples, classes, dimensions and the number of combined kernels in Table 1. The disease and journal data sets have very high dimensionality so the kernel matrices are constructed using the linear kernel functions. An element in the matrix is then equivalent to the value of cosine similarity of two vectors. The data sets used in experiments are provided with labels, therefore the performance is evaluated as comparing the automatic partitions with the labels using Adjusted Rand Index (ARI) [21] and Normalized Mutual Information (NMI) [40].

4.2 Results

The overall clustering results are shown in Table 2. For each data set, we present the best and the worst performance of clustering obtained on single kernel matrix. We compare three different approaches to combine multiple kernel matrices: the average combination of all kernel matrices in kernel k-means clustering, the proposed OKKC algorithm and the NAML algorithm. For OKKC, only results obtained when $\delta = 1$ is presented in Table 2 because NAML only concerns L_1-norm

TABLE 2
Overall results of clustering performance

	best individual		worst individual		average combine			OKKC				NAML			
	ARI	NMI	ARI	NMI	ARI	NMI	time(sec)	ARI	NMI	itr	time(sec)	ARI	NMI	itr	time(sec)
Iris	0.7302 (0.0690)	0.7637 (0.0606)	0.6412 (0.1007)	0.7047 (0.0543)	0.7132 (0.1031)	0.7641 (0.0414)	0.22 (0.13)	0.7516 (0.0690)	0.7637 (0.0606)	7.8 (3.7)	5.32 (2.46)	0.7464 (0.0207)	0.7709 (0.0117)	9.2 (2.5)	15.45 (6.58)
Wine	0.3489 (0.0887)	0.3567 (0.0808)	0.0387 (0.0175)	0.0522 (0.0193)	0.3188 (0.1264)	0.3343 (0.1078)	0.25 (0.03)	0.3782 (0.0547)	0.3955 (0.0527)	10 (4.0)	18.41 (11.35)	0.2861 (0.1357)	0.3053 (0.1206)	6.7 (1.4)	16.92 (3.87)
Yeast	0.4246 (0.0554)	0.5022 (0.0222)	0.0007 (0.0025)	0.0127 (0.0038)	0.4193 (0.0529)	0.4994 (0.0271)	2.47 (0.05)	0.4049 (0.0375)	0.4867 (0.0193)	7 (1.7)	81.85 (14.58)	0.4256 (0.0503)	0.4998 (0.0167)	10 (2)	158.20 (30.38)
Satimage	0.4765 (0.0515)	0.5922 (0.0383)	0.0004 (0.0024)	0.0142 (0.0033)	0.4891 (0.0476)	0.6009 (0.0278)	4.54 (0.07)	0.4996 (0.0571)	0.6004 (0.0415)	10.2 (3.6)	213.40 (98.70)	0.4911 (0.0522)	0.6027 (0.0307)	8 (0.7)	302 (55.65)
Pen digit	0.5818 (0.0381)	0.7169 (0.0174)	0.2456 (0.0274)	0.5659 (0.0257)	0.5880 (0.0531)	0.7201 (0.0295)	15.95 (0.08)	0.5904 (0.0459)	0.7461 (0.0267)	8 (4.38)	396.48 (237.51)	0.5723 (0.0492)	0.7165 (0.0295)	8 (4.2)	1360.32 (583.74)
Disease genes	0.7585 (0.0043)	0.5281 (0.0078)	0.5900 (0.0014)	0.1928 (0.0042)	0.7306 (0.0061)	0.4702 (0.0101)	931.98 (1.51)	0.7641 (0.0078)	0.5395 (0.0147)	5 (1.5)	1278.58 (120.35)	0.7310 (0.0049)	0.4715 (0.0089)	8.5 (2.6)	3268.83 (541.92)
Journal sets	0.6644 (0.0878)	0.7203 (0.0523)	0.5341 0.0580	0.6472 0.0369	0.6774 (0.0316)	0.7458 (0.0268)	63.29 (1.21)	0.6812 (0.0602)	0.7420 (0.0439)	8.2 (4.4)	1829.39 (772.52)	0.6294 (0.0535)	0.7108 (0.0355)	9.1 (6.1)	4935.23 (3619.50)

All the results are mean values of 20 random repetitions and the standard deviation (in parentheses).The tolerance value ϵ is set to 0.05. The individual kernels and average kernels are clustered using kernel k-means [17]. The OKKC is programmed using Matlab functions *eig*, *linsolve* and *linprog*. The δ is set to 1 in this table. The disease gene data is clustered by OKKC using the explicit regularization parameter λ (set to 0.0078) because the linear kernel matrices constructed from gene-by-term profiles are very sparse (a gene normally is only indexed by a small number of terms in the high dimensional vector space). In this case, the joint estimation assigns dominant coefficients on the identity matrix and decreases the clustering performance. The optimal λ value is selected among ten values uniformly distributed on the log scale from 2^{-5} to 2^{-4}. For other data sets, the λ values are estimated automatically and their values are shown as λ_{okkc} in Figure 1. The NAML is programmed as the algorithm proposed in [8] using Matlab and MOSEK [1]. We try forty-one different λ values for NAML on the log scale from 2^{-20} to 2^{20} and the highest mean values and their deviations are presented. In general, the performance of NAML is not very sensitive to the λ values. The optimal λ values for NAML are shown in Figure 1 as λ_{naml}. The computational time (no underline) is evaluated on Matlab v7.6.0 + Windows XP SP2 installed on a Laptop computer with Intel Core 2 Duo 2.26GHz CPU and 2G memory. The computational time (underlined) is evaluated on Matlab v7.9.0 installed on a dual Opteron 250 Unix system with 7Gb memory.

regularization. As shown, the performance obtained by OKKC is comparable to the results of the best individual kernel matrices. OKKC is also comparable to NAML on all the data sets, moreover, on Wine, Pen, Disease, and Journal data, OKKC performs significantly better than NAML (as shown in Table 3). The computational time used by OKKC is also smaller than NAML. Since OKKC and NAML use almost the same number of iterations to converge, the efficiency of OKKC is mainly brought by its bi-level optimization procedure and the linear system solution based on SIP formulation. In contrast, NAML optimizes three variables in a tri-level procedure and involves many inverse computation and eigenvalue decompositions on kernel matrices. Furthermore, in NAML, the kernel coefficients are optimized as a QCQP problem. When the number of data points and the number of classes are large, QCQP problem may have memory issues. In our experiment, when clustering Pen digit data and Journal data, the QCQP problem causes memory overflow on a laptop computer. Thus we have to solve them on a Unix system with larger amount of memory. On the contrary, the SIP formulation used in OKKC significantly reduces the computational burden of optimization and the clustering problem usually takes 25 to 35 minutes on the ordinary laptop.

We also compare the kernel coefficients optimized by OKKC (δ = 1) and NAML on all the data sets. As shown in Figure 1, NAML algorithm often selects a single kernel for clustering (a sparse solution for data fusion). In contrast, OKKC algorithm often combines two or three kernel matrices in clustering. When combining p kernel matrices, the regularization parameters λ estimated in OKKC are shown as the coefficients of an additional $(p+1)$-th identity matrix (the last bar in the figures, except on disease data because λ is also pre-selected), moreover, in OKKC it is easy to see that $\lambda = (\sum_{r=1}^{p} \theta_r)/\theta_{p+1}$. The λ values of NAML are selected empirically according to the clustering performance. Practically, to determine the optimal regularization parameter in clustering analysis is hard because the data is unlabeled thus the model cannot be validated. Therefore, the automatic estimation of λ in OKKC is useful and reliable in clustering.

Apart from OKKC and NAML, we also apply six other

TABLE 3
Significance test of clustering performance.

data	OKKC vs. single		OKKC vs. NAML		OKKC vs. average	
	ARI	NMI	ARI	NMI	ARI	NMI
iris	0.2213	0.8828	0.7131	0.5754	0.2282	0.9825
wine	0.2616	0.1029	**0.0085**(+)	**0.0048**(+)	0.0507	**0.0262**(+)
yeast	0.1648	**0.0325**(-)	0.1085	**0.0342**(-)	0.2913	**0.0186**(-)
satimage	0.1780	0.4845	0.6075	0.8284	0.5555	0.9635
pen	**0.0154**(+)	0.2534	**3.9e-11**(+)	**3.7e-04**(+)	0.4277	**0.0035**(+)
disease	**1.3e-05**(+)	**1.9e-05**(+)	**4.6e-11**(+)	**3.0e-13**(+)	**7.8e-11**(+)	**1.6e-12**(+)
journal	0.4963	0.2107	**0.0114**(+)	**0.0096**(+)	0.8375	0.7626

The presented numbers are p values evaluated by paired t-tests on 20 random repetitions. When the null hypothesis is rejected, "+" represents that the performance of OKKC is higher than the comparing approaches. "-" means that the performance of OKKC is lower.

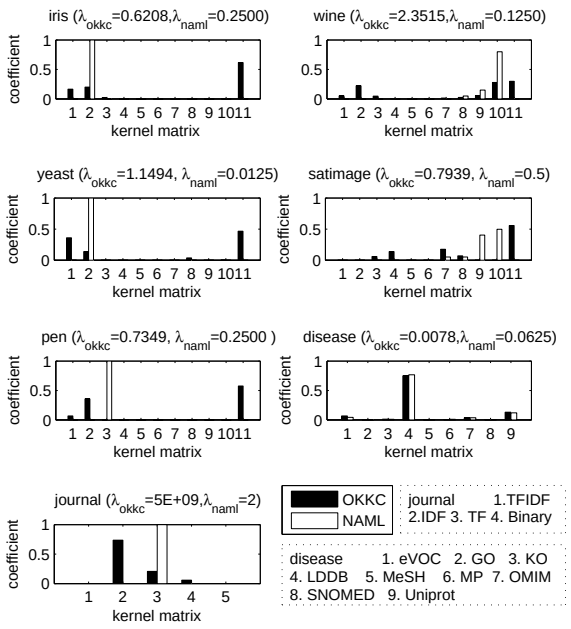


Fig. 1. Kernel coefficients learned by OKKC and NAML. Both algorithms optimize coefficients using L_1 norm regularization. For OKKC applied on iris, wine, yeast, satimage, pen and journal data, the last coefficients correspond to the inverse values of the regularization parameters.

TABLE 4
Comparison of clustering algorithms on real-application data sets.

Data Set	Algorithm	ARI	NMI
disease data	OKKC1	0.7641 ± 0.0078	0.5395 ± 0.0147
	OKKC2	0.7027 ± 0.0036	0.4385 ± 0.0142
	NAML	0.7310 ± 0.0049	0.4715 ± 0.0089
	CSPA	0.7011 ± 0.0065	0.4479 ± 0.0097
	HGPA	0.6245 ± 0.0035	0.3015 ± 0.0071
	MCLA	0.7596 ± 0.0021	0.5268 ± 0.0087
	QMI	0.7458 ± 0.0039	0.5084 ± 0.0063
	EACAL	0.7741 ± 0.0041	0.5542 ± 0.0068
	AdacVote	0.7300 ± 0.0045	0.4093 ± 0.0100
journal data	OKKC1	0.6812 ± 0.0602	0.7420 ± 0.0439
	OKKC2	0.6968 ± 0.0953	0.7509 ± 0.0531
	NAML	0.6294 ± 0.0535	0.7108 ± 0.0355
	CSPA	0.6523 ± 0.0475	0.7038 ± 0.0283
	HGPA	0.6668 ± 0.0621	0.7098 ± 0.0334
	MCLA	0.6507 ± 0.0639	0.7007 ± 0.0343
	QMI	0.6363 ± 0.0683	0.7058 ± 0.0481
	EACAL	0.6670 ± 0.0586	0.7231 ± 0.0328
	AdacVote	0.6617 ± 0.0542	0.7183 ± 0.0340

The experimental settings are the same as mentioned in Table II.

clustering algorithms in two real-applications and the results are shown in Table IV. OKKC1 is the proposed model using $\delta = 1$, OKKC2 sets $\delta = 2$. GSPA, HGPA and MCLA are clustering ensemble methods proposed in [40], QMI is proposed by [43], EACAL is proposed by [15], and AdacVote is proposed by [2]. Among all the algorithms compared, only OKKC and NAML optimize mixture coefficients of data sources explicitly. We also notice that EACAL seems performing quite well on disease data but not successful on journal data. OKKC is comparable to the best candidates in comparison, which indicates that the optimized data fusion indeed improves the performance. On journal data, two δ values yield comparable performance whereas on disease data the performance of OKKC2 degrades significantly, which is probably because some CVocs are irrelevant to the disease identification task thus the non-sparse integration involving all the CVocs is less favorable than the sparse integration.

When using spectral relaxation, the optimal cluster number of k-means can be estimated by checking the plot of eigenvalues [44]. We can also use the same technique to find the optimal cluster number of data fusion using OKKC. To demonstrate this, we cluster all the data sets using different k values and plot the eigenvalues in Figure 2. As shown, the obtained eigenvalues with various k are slightly different with each other because when k is different, the optimized kernel coefficients are also different. However, we also find that even the kernel fusion results are different, the plots of eigenvalues obtained from the combined kernel matrix are quite similar to each other. In practical explorative analysis, one may be able to determine the optimal and consistent cluster number using OKKC with various k values. The results show that OKKC can also be applied to find the clusters using the eigenvalues.

5 Conclusion and Future Work

The paper presented OKKC, a data fusion algorithm for kernel k-means clustering, where the coefficients of kernel matrices in the combination are optimized automatically. The proposed algorithm extends the classical k-means clustering algorithm in Hilbert space, where multiple heterogeneous data sats are represented as kernel matrices and combined for data fusion. The objective of OKKC is formulated as a Rayleigh quotient function of two variables, the cluster assignment A and the kernel coefficients $\vec{\theta}$, which are optimized iteratively towards the same objective. The proposed algorithm is shown to converge locally and implemented as an integration of kernel k-means clustering and LS-SVM multiple kernel learning.

The experimental results on UCI data sets and real application data sets validated the proposed method. The proposed OKKC algorithm obtained comparable result with the best individual kernel matrix and the NAML algorithm. Moreover, in several data sets it performs significantly better. Because of its simple optimization procedure and low computational complexity, the computational time of OKKC is always smaller than the NAML. The proposed algorithm also scales up well on large data sets thus it is more easy to run on ordinary machines.

The bi-level optimization procedure proposed algorithm can be easily extended to incorporate different criteria in clustering and KFD. It is also possible to deal with overlapping cluster membership, known as "soft

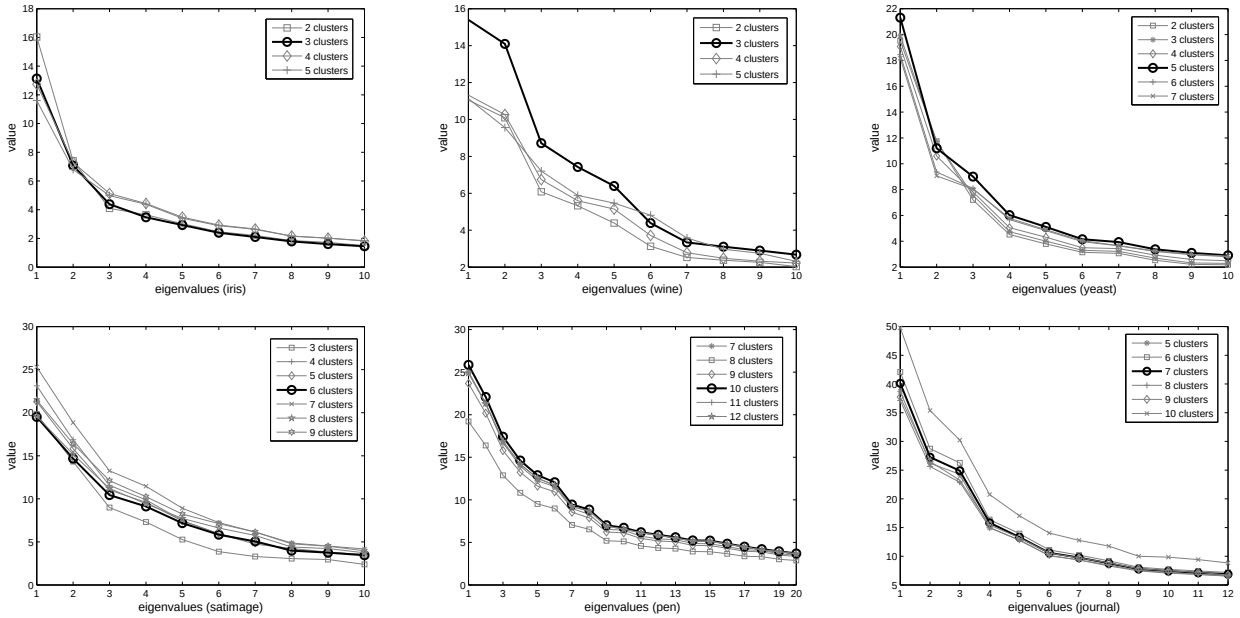


Fig. 2. Eigenvalues of optimally combined kernels of data sets obtained by OKKC. The δ parameter is set to 1. For each data set we try four to six k values including the one suggested by the reference labels, which is shown as a bold dark line, other values are shown as grey lines. The eigenvalues in disease gene clustering are not shown because there are 406 different clustering tasks.

clustering". In many application such as bioinformatics, a gene or protein may be simultaneously related to several biomedical concepts so it is necessary to have a "soft clustering" algorithm to combine multiple data sources. Notice that the spectral relaxation of k-means has similar objective function as spectral clustering using *normalized Laplacian matrix* [11], [44]. Thus, the proposed method can also be used to clustering multiple graphs [41], [50] in an optimized way.

ACKNOWLEDGMENT

The work was supported by (i) Research Council KUL: ProMeta, GOA Ambiorics, GOA MaNet, CoEEF/05/006, PFV/10/016 SymBioSys, START 1, Optimization in Engineering(OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants; (ii) FWO: G.0302.07(SVM/Kernel), G.0318.05 (subfunctionalization), G.0553.06 (VitamineD), research communities (ICCoS, ANMMM, MLDM); G.0733.09 (3UTR), G.082409 (EGFR); (iii) IWT: PhD Grants, Eureka-Flite+, Silicos; SBO-BioFrame, SBO-MoKa, SBO LeCoPro, SBO Climaqs, SBO POM, TBM-IOTA3, O&O-Dsquare; (iv) IBBT; (v) Belgian Federal Science Policy Office: IUAP P6/25 (BioMaGNet, Bioinformatics and Modeling: from Genomes to Networks, 2007C2011), IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011); (vi) FOD: Cancer plans; (vii) Flemish Government: Center for R & D Monitoring (ECOOM); (viii) EU-RTD: ERNSI: European Research Network on System Identification; FP7-HEALTH CHeartED; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940); (ix) National Natural Science Foundation of China (Grant No. 61105058).

REFERENCES

[1] E. D. Andersen, and K. D. Andersen, "The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm", *High Perf. Optimization*, pp. 197–232, 2000.
[2] H. G. Ayad, and M. S. Kamel, "Cumulative Voting Consensus Method for Partitions with a Variable Number of Clusters", *IEEE Trans. PAMI*, vol.30(1), pp,160-173, 2008.
[3] R. Bhatia, *Matrix Analysis*, Springer-Verlag, New York, 1997.
[4] C. M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.
[5] S. Boyd, and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
[6] G. Baudat, and F. Anouar, "Generalized Discriminant Analysis Using a Kernel Approach", *Nerual Computation*, vol. 12(10), pp. 2385-2404, 2000.
[7] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via Canonical Correlation Analysis", *in Proceedings of 26th ICML*, 2009.
[8] J. Chen, Z. Zhao, J. Ye, and H. Liu, "Nonlinear adaptive distance metric learning for clustering", *Proc. of ACM SIGKDD 07*, 2007.
[9] I. Csiszar and G. Tusnady, "Information geometry and alternating minimization procedures", *Statistics and Decisions*, Supplbementary Issue 1, pp. 205-237, 1984.
[10] L. De Lathauwer, B. D. Moor, and and J. Vandewalle, "On the best rank-1 and rank-(r_1, r_2,..., r_n) approximation of higher-order tensors", *SIAM J. Matrix Anal. Appl.*, vol. 21(4), pp.1324-1342, 2000.
[11] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means, Spectral Clustering, and Normalized Cuts", *in Proceedings of ACM KDD 04*, pp. 551-556, 2004.
[12] C. Ding, and X. He, "K-means Clustering via Principal Component Analysis", *in Proc. of ICML 2004*, pp. 225-232, 2004.
[13] C. Ding, and X. He, "Linearized cluster assignment via spectral ordering", *Proc. of ICML 2004*, 2004.
[14] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification (2nd Edition)*, John Wiley & Sons Inc., 2001.

[15] A. L. N. Fred, A. K. Jain, "Combining Multiple Clusterings Using Evidence Accumulation", *IEEE Trans. PAMI*, vol.27(6), pp.835-850, 2005.
[16] M. R. Garey, and D.S. Johnson, *Computers and Intractability: A Guide to NP-Completeness*, W. H. Freeman, New York, 1979.
[17] M. Girolami, "Mercer Kernel-Based Clustering in Feature Space", *IEEE Trans. Neural Networks*, vol. 13(3), pp. 780-784, 2002.
[18] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd Edition)*, Springer, 2009.
[19] R. Hettich, and K. O. Kortanek, "Semi-infinite programming: theory, methods, and applications", *SIAM Review*, vol. 35(3), pp.380-429, 1993.
[20] P. Howload, and H. Park, "Generalizing discriminant analysis using the generalized singular value decomposition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.26(8), pp.995-1006, 2004.
[21] L. Hubert, and P. Arabie, "Comparing partitions", *Journal of Classification*, vol. 2(1), pp.193-218, 1985.
[22] A. K. Jain, and R. C. Dubes, *Algorithms for clustering data*, Prentice Hall, New Jersey, 1988.
[23] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskow, K. R. Mueller, and A. Zien, "Efficient and Accurate L_p-norm MKL", *in Advances in Neural Information Processing Systems 21*, pp. 997-1005, 2009.
[24] G. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui, and M.I. Jordan, "Learning the kernel Matrix with Semidefinite Programming", *Journal of Machine Learning Research*, vol. 5, pp. 27-72, 2004.
[25] T. Lange, and J.M. Buhmann, "Fusion of Similarity Data in Clustering", *Proc. of NIPS 2005*, 2005.
[26] Y. Liang, C. Li, W. Gong, and Y. Pan, "Uncorrelated linear discriminant analysis based on weighted pairwise Fisher criterion", *Pattern Recognition*, vol. 40, pp. 3606-3615, 2007.
[27] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition", *IEEE Trans. on Neural Networks*, vol. 20(1), pp. 103-123, 2009.
[28] X. Liu, S. Yu, Y. Moreau, B. De Moor, W. Glänzel, F. Janssens, "Hybrid Clustering of Text Mining and Bibliometrics Applied to Journal Sets", *Proc. of the SIAM Data Mining Conference 09*, 2009.
[29] J. Ma, J.L. Sancho-Gómez, and S.C. Ahalt, "Nonlinear Multiclass Discriminant Analysis", *IEEE Signal Processing Letters*, vol. 10(7), pp. 196-199, 2003.
[30] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University, 2003.
[31] S. Mika, G. Rätsch, J. Weston, and B. Schölkopf, "Fisher discriminant analysis with kernels", *IEEE Neural Networks for Signal Processing IX*, pages 41-48, 1999.
[32] C. H. Park, and H. Park, "Efficient nonlinear dimension reduction for clustered data using kernel functions", *Proceeding of the 3rd IEEE International Conference on Data Mining*, pp. 243-250, 2003.
[33] J. Shawe-Taylor and N. Cristianin, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.
[34] G. Sanguinetti, "Dimensionality reduction of clustered data sets", *IEEE TPAMI*, vol.30(3), pp. 535-540, 2008.
[35] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem", *Neural Computation*, vol.10, pp.1299-1319, 1998.
[36] B. Schölkopf, R. Herbrich, and A.J. Smola, "A Generalized Representer Theorem", *Proc. of the 14th COLT and 5th ECCLT*, pp. 416-426, 2001.
[37] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large Scale Multiple Kernel Learning", *Journal of Machine Learning Research*, vol. 7, pp. 1531-1565, 2006.
[38] G.W. Stewart, and J.G. Sun, *Matrix perturbation theory*, Academic Press, Boston, 1999.
[39] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific Publishing Co. Pte. Ltd., Singapore, 2002.
[40] A. Strehl, and J. Ghosh, "Clustering Ensembles: a knowledge reuse framework for combining multiple partitions", *Journal of Machine Learning Research*, vol. 3, pp.583-617, 2002.
[41] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with Multiple Graphs",
[42] S. Theodoridis, and K. Koutroumbas, *Pattern Recognition (2nd Edition)*, Elsevier Science, USA.
[43] A. Topchy, A. K. Jain, and W. Punch, "Clustering Ensembles: Models of Consensus and Weak Partitions", *IEEE Trans. PAMI*, vol.27, pp.1866-1881, 2005.
[44] U. von Luxburg, "A tutorial on spectral clustering", *Statistics and Computing*, vol. 17(4), pp. 395-416, 2007.
[45] J. Ye, Z. Zhao, and M. Wu, "Discriminative K-Means for Clustering", *Proc. of NIPS 2007*, 2007.
[46] J.P. Ye, S.W. Ji, and J.H. Chen, Multi-class Discriminant Kernel Learning via Convex Programming, *Journal of Machine Learning Research*, vol. 9, pp. 719-758, 2008.
[47] S. Yu, L.-C. Tranchevent, B. De Moor, and Y. Moreau, "Gene prioritization and clustering by multi-view text mining", *BMC Bioinformatics*, vol. 11(28), 2010.
[48] S. Yu, T. Falck, A. Daemen, L. C. Tranchevent, J. Suykens, B. De Moor, and Y. Moreau, "L_2-norm multiple kernel learning and its application to biomedical data fusion", *BMC Bioinformatics*, vol. 11:309, 2010.
[49] H. Zha, C. Ding, M. Gu, X. He, and H. Simon, "Spectral Relaxation for K-means. Clustering", *in Proceedings of Advances in Nerual Information Processing*, vol. 14, pp. 1057-1064, 2001.
[50] D. Zhou, and C. J. C. Burges, "Spectral Clustering and Transductive Learning with Mulitple Views", *in Proceedings of 24th ICML*, 2007.
[51] G.K. Zipf, *Human behaviour and the principle of least effort. An introduction to human ecology*, Addison-Wesley, 1949.