Integrating heterogeneous data sets for clustering

Shi Yu, Leo C. Tranchevent, Xinhai Liu, Bart De Moor, *Fellow, IEEE,* and Yves Moreau

Abstract

This paper presents a novel framework and several applications for clustering with multiple data sources (data fusion). A series of clustering algorithms are investigated in a unified framework and they are categorized as two different approaches. The first approach is ensemble clustering that finds the optimal data partition by aggregating partitions from multiple sources via consensus functions. The second approach is kernel fusion, where data sources are transformed into kernel matrices and the combination is achieved by (average) linear combination of kernels. The combined kernel matrix is then regarded as a virtual single data set so that the integration issue is transparent to the clustering procedure. Furthermore, a more flexible strategy is to weigh different data sources according to their relevance with respect to the clustering problem, so we also propose an adaptive kernel k-means algorithm (AKKC) by coupling the clustering procedure with optimization problem. The effectiveness of the proposed computational framework is demonstrated on a synthetic problem and several actual applications.

Index Terms

Clustering, data integration, clustering ensemble, kernel fusion, kernel optimization, kernel k-means, text mining, signal processing, information retrieval, disease gene identification

I. INTRODUCTION

Clustering is an important problem with many applications, a number of different algorithms and methods have emerged over the years. The goal of clustering is to group samples into homogeneous groups, where the homogeneity is usually measured by distances or proximities among samples. Recently, many applications face the requirement of clustering with multiple data sources. This is because information contained in a single data source is limited by its specific observation, therefore, combining multiple observations might facilitate the comprehensive understanding of the problem. For instance, in order to investigate memory persistence (long term or short term memory) of bacteria, a bacterium is observed at different experimental conditions and developmental times [38]. Then these experimental observations obtained at different times are combined together and categorized by clustering algorithms. In scientometrics, a strategy has been proposed to combine text mining data and bibliometrics data (hybrid clustering) to explore the structure mapping of journal sets [14], [15]. In bioinformatics, high-throughput techniques produce numerous genomic data. The challenge to endow clustering algorithms with the ability to retrieve correlated or complementary information about the underlying functional

partitions of genes and proteins has attracted much interests [43], [19], [3], [27]. Unfortunately, though the machine learning community has already focused on data fusion for classification [20] and novelty detection [7], the extension to unsupervised learning such as clustering, is still an unresolved and ongoing problem.

In this paper, we present a unified computational framework to obtain partitions from heterogeneous sources. When clustering by data fusion, determining about the *relevance* or *usefulness* of the source is a vital issue to statistically guarantee the lower bound of the performance. In other words, if the clustering algorithm is able to detect the most *relevant* or *useful* data source, we can then expect that the fusion approach works at least as well as the best individual source. To achieve the above objective, two different strategies are applied in the framework. The first approach is clustering ensemble. The second approach is kernel fusion. The effectiveness of this framework is evaluated on several experiments and applications.

The organization of this paper is as follows. Section 2 introduces the clustering ensemble approach and representative algorithms. Section 3 presents the kernel fusion approach. In particular, we propose several novel kernel fusion algorithms for clustering. In Section 4, we introduce the evaluation methods used in this paper. Section 5 concludes all the algorithms and evaluation methods in a unified computational framework for data fusion. The description of applications of analysis of results are presented in Section 6. The final discussion and conclusion is made in Section 7.

II. Ensemble clustering for data fusion

Clustering algorithms for data fusion can be roughly divided into two main categories. The first approach is ensemble clustering, also known as clustering aggregation or consensus clustering, which combines multiple partitions into a consolidate partition by consensus function. Different ensemble algorithms mainly vary on the choice of consensus functions. Ensemble clustering is originally applied on a single source where various partitions are generated by different representations, dimensionality reduction techniques, and clustering algorithms. The strategy of the ensemble clustering can be extended to data fusion, where the main difference is that partitions are varied by data sources. An underlying assumption is: if the information contained in multiple sources is highly correlated, their partitions should also contain a "common agreement" thus a consolidated partition can also be obtained. Below we will outline some well know

clustering ensemble algorithms.

HGPA, CSPA, MCLA Strehl and Ghosh [32] formulate the optimal consensus as the partition that shares the most information with the partitions to combine, as measured by the Average Normalized Mutual Information. They use three heuristic consensus algorithms based on graph partitioning, called Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper Graph Partitioning Algorithm (HGPA) and Meta Clustering Algorithm (MCLA) to obtain the combined partition.

QMI Topchy [37] formulates the combination of partitions as a categorical clustering problem. In their approach, a category utility function [18] is adopted to evaluate the quality of a "median partition" as a summary of the ensemble. Topchy proves that maximizing this category utility function implies the same clustering ensemble criterion as maximizing the generalized mutual information based on quadratic entropy. Furthermore, the maximization of the category utility function is equivalent to the square error based clustering criterion when the number of clusters is fixed. The final consensus partition is obtained by applying the k-Means algorithm on the feature space transformed by the category utility function.

EACAL Fred and Jain [9] introduce the concept of evidence accumulation clustering (EAC) that maps the individual data partitions as a clustering ensemble by constructing a co-association matrix. The entries of the co-association matrix is interpreted as votes on the pairwise co-occurrences of objects, which is computed as the number of occurrences each a pair of objects appear in the same cluster of an individual partition. Then the final consensus partition is obtained by applying single-linkage (SL) and average-linkage (AL) methods on the co-association matrix. According to their experiments, average linkage performs better than single linkage so in this paper we apply EAC-AL as the representative algorithm for comparison.

AdacVote Ayad and Kamel [4] propose a "cumulative vote weighting method" to compute an empirical probability distribution summarizing the ensemble. The goal of this ensemble is to minimize the average squared distance between the mapped partitions and the combined partition. The cumulative voting method seeks an adaptive reference partition and incrementally updates it by averaging other partitions to relax the dependence of the combined partition on the selected reference. In the adaptive cumulative voting (ACV) algorithm they proposed, the partitions are combined in decreasing order of their entropies.

III. CLUSTERING OF HETEROGENEOUS DATA SOURCES BY KERNEL FUSION

An alternative approach of clustering by data fusion is achieved by fusing similarity matrices [21]. If the similarity matrices are positive semi-definite, the data integration problem can be formulated as a kernel fusion problem. The main difference of the kernel fusion approach is that the integration is carried in kernel space before a clustering algorithm is applied (early integration), while ensemble clustering aggregates partitions after clustering (late integration). Kernel method elegantly resolves the heterogeneities of data sources by representing them as same-sized kernel matrices. Moreover, if we assume that the importance of each data source is equivalent, we can combine the kernels in an average manner, thus the issue of data integration is transparent to the clustering algorithm. The combined average kernel is a new data source, thus the partition can be obtained by standard clustering algorithms in kernel space. Furthermore, we can also apply a more machine-intelligent approach by coupling the optimization problem of kernel learning with the objective function of pattern analysis. In that case, the weights assigned on each data source can be adjusted adaptively during the clustering procedure [5]. In this paper, we propose a novel adaptive kernel k-means clustering (AKKC) algorithm to obtain partitions and optimal weights simultaneously.

A. Adaptive Kernel K-means Clustering (AKKC)

1) Objective function for clustering: The standard k-means hard clustering algorithm adopt squared Euclidean distance to measure the dissimilarity between vectors x_i and cluster representatives θ_j. Assuming the membership coefficient u_{ij}, which indicates whether the i-th sample belongs to the j-th cluster, is either 1 or 0. Then the cost function of k-means clustering becomes

$$J(\theta, U) = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij} \|x_i - \theta_j\|^2 \tag{1}$$

However, for clustering based on data integration, the *Mahalanobis* distance is preferred because it is invariant to nonsingular linear transformation [39]. It scales the distance between two objects by the inverse covariance matrix,

$$d_M(x_i, x_j) = \left[(x_i - x_j)^T C^{-1} (x_i - x_j)\right]^{\frac{1}{2}}, \tag{2}$$

where C is the covariance matrix defined as follows,

$$C = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T, \tag{3}$$

and $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the mean of $\{x_i\}_{i=1}^{n}$.

To avoid the singularity of the covariance matrix, a regularized covariance matrix is often used as

$$C = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T + \lambda I, \tag{4}$$

where I is the identity matrix and $\lambda > 0$ is the regularization parameter.

Using the distance measure defined above, k-means like clustering can be regarded as partitioning the data $\{x_i\}_{i=1}^{n}$ into k disjoint clusters, $\{l_1, l_2, \ldots, l_k\}$, which minimize the *Within Clusters Sum of Square Error* (WSSE),

$$\text{WSSE}(\{l_j\}_{j=1}^{k}) = \sum_{j=1}^{k}\sum_{x_i \in l_j} d_M(x_i, \mu_j)^2, \tag{5}$$

where $d_M(\cdot,\cdot)$ is the *Mahalanobis* distance defined in (1) and μ_j is the mean of the j-th cluster l_j. For a given data set, the summation of all pairwise distance is a constant value hence the minimization of WSSE is equal to the maximization of *Between Clusters Sum of Square Error*(BSSE) defined as follows,

$$\text{BSSE}(\{l_j\}_{j=1}^{k}) = \sum_{j=1}^{k}|l_j| d_M(\mu_j, \hat{\mu})^2, \tag{6}$$

where $|l_j|$ is the cardinality of samples in cluster j, μ_j is the mean of the j-th cluster l_j, and $\hat{\mu}$ is the global mean of $\{x_i\}_{i=1}^{n}$.

The BSSE can be expressed in a compact matrix form as

$$\text{BSSE}(\{l_j\}_{j=1}^{k}) = \text{trace}(L^T X^T C^{-1} X L), \tag{7}$$

where X is the data matrix, L is the *weighted cluster indicator matrix* $L = [l_1, l_2, \ldots, l_k]$ defined as

$$L = F(F^T F)^{-\frac{1}{2}}, \tag{8}$$

where F is the $n \times k$ cluster indicator matrix defined as follows,

$$F = f_{i,j}{}_{n\times k}, \text{where} f_{i,j} = \begin{cases} 1 & \text{if } x_i \in l_j \\ 0 & \text{if } x_i \notin l_j \end{cases}. \tag{9}$$

2) Kernel extension: The objective function defined in (7) can be extended into kernel space by a mapping implicitly specified by a symmetric kernel function Ω, which computes the inner product of the pairwise data in kernel space, that is

$$\Omega(x_i, x_j) = (\phi(x_i), \phi(x_j)), \tag{10}$$

where x_i, x_j are data points in the original space, $\phi(\cdot)$ is the kernel mapping. Let $\phi_\Omega(X)$ denote the data matrix in kernel space defined by kernel mapping Ω, therefore in kernel space the objective function for clustering can be formulated as the following trace maximization problem:

$$\max_{\Omega, L} \text{trace}\big(L^T \phi_\Omega(X)^T C_\Omega^{-1} \phi_\Omega(X) L\big). \tag{11}$$

We assume that the data in feature space has been centered, so the regularized covariance matrix has the form as,

$$C_{\hat{\Omega}} = \phi_{\hat{\Omega}}(x)\phi_{\hat{\Omega}}(x)^T + \lambda I = \hat{\Omega} + \lambda I, \tag{12}$$

where $\hat{\Omega}$ is the centered kernel matrix on X. For simplicity, from now on we denote Ω as the centered kernel matrix on X. The clustering problem can also be addressed in the framework of kernel fusion. Given a set of p centered kernel matrices, the optimal kernel matrix Ω^* that optimizes the objective function

$$F_4 = \max_{\Omega, L} \text{trace}\left(L^T \phi_\Omega(x)^T \left(\phi_\Omega(x)\phi_\Omega(x)^T + \lambda I\right)^{-1} \phi_\Omega(x) L\right), \tag{13}$$

is defined as the convex linear combination of p centered kernel matrices

$$\Omega = \left\{ \sum_{i=1}^{p} \mu_i \Omega_i \middle| \sum_{i=1}^{p} \mu_i r_i = 1, \mu_i > 0, r_i = \text{trace}(\Omega_i) \right\}.$$

3) Algorithm and proofs: Since solving L^* and Ω^* simultaneously is difficult, the solution of problem (13) is found iteratively in an alternating minimization framework [6]. The input parameters for the algorithms are the set of kernels $\Omega_1, ..., \Omega_p$, the predefined number of clusters k and the regularization parameter of covariance matrix λ. Within the alternative minimization framework, the solution is obtained by fixing L and solving Ω, then fixing Ω and solving L, iteratively until convergence.

Algorithm III.1: ADAPTIVEKMEANS$(\Omega_1, \Omega_2, ..., \Omega_p, K, \lambda)$

$L_{(0)} \leftarrow$ KERNEL K-MEANS CLUSTERING(Ω_s), $s \in 1, ..., p$

comment: obtain an initial partition

while <!convergence >

do $\begin{cases} step1 : \Omega \leftarrow \text{ADAPTIVE WEIGHING}(L_{(i)}, \Omega_1, \Omega_2, ..., \Omega_p, \lambda) \\ step2 : L_{(i+1)} \leftarrow \text{KERNEL K-MEANS CLUSTERING}(\Omega, K) \end{cases}$

comment: the convergence is determined by comparing the change of trace value in (13)

return (L_{i+1})

Step 1: Adaptive weighing of kernels for given data partition. To find the optimal kernel matrix Ω with the given L, we first consider a binary clustering problem with l, the objective function in (13) is then written as,

$$F_1 = \max_{\Omega} \text{trace}\left(l^T \phi(x)^T \left(\phi(x)\phi(x)^T + \lambda I\right)^{-1} \phi(x) l\right). \tag{14}$$

which can be proved as an equivalent problem as a regularized regression (ridge regression) problem,

$$F_3(w, \phi_\Omega) = \|\phi_\Omega(x) w - l\|^2 + \lambda \|w\|^2. \tag{15}$$

The proof of the equivalence between (14) and (15) is given in [17] [5]. We also present the brief proof in Appendix I. Based on this equivalence, we obtain the theorem as follows.

Theorem 1: Given a set of p centered kernel matrices $\Omega_1, \ldots, \Omega_p$, the optimal kernel matrix Ω is the convex linear combination of p matrices, that optimizes the equivalent objective function,

$$F_3(w, \phi_\Omega) = \|\phi_\Omega(x) w - l\|^2 + \lambda \|w\|^2. \tag{16}$$

is the solution of the convex quadratically constrained quadratic programming problem,

$$\begin{aligned} \max_{\beta, t} \quad & -\tfrac{1}{4}\beta^T \beta + \beta^T l - \tfrac{1}{4\lambda} t \\ \text{s.t.} \quad & t \geq \tfrac{1}{r_i} \beta^T \Omega_i \beta, \qquad i = 1, \ldots, n, \\ & \beta_j \geq 0, \qquad j = 1, \ldots, k, \end{aligned} \tag{17}$$

where

$$\Omega = \left\{ \sum_{i=1}^{n} \mu_i \Omega_i \middle| \sum_{i=1}^{n} \mu_i r_i = 1, \mu_i > 0, r_i = \text{trace}(\Omega_i) \right\}.$$

Proof: Rewrite (16) as a primal problem in terms of w,

$$\begin{aligned} \min_{w,\eta} \quad & F_3(w, \phi_{\mathbf{\Omega}}) = \|\eta\|^2 + \lambda\|w\|^2, \\ \text{s.t.} \quad & \eta = \phi_{\mathbf{\Omega}}(x)w - l. \end{aligned} \tag{18}$$

The Lagrangian form of (18) is

$$\mathcal{L}(\eta, w; \beta) = \|\eta\|^2 + \lambda\|w\|^2 - \beta^T \left(\phi_{\mathbf{\Omega}}(x)w - l - \eta\right), \tag{19}$$

where β is the vector of Lagrangian dual variables. The conditions for optimality follows

$$\begin{cases} \eta = -\frac{\beta}{2} \\ w = \frac{\phi_{\mathbf{\Omega}}(x)\beta}{2\lambda} \end{cases} \tag{20}$$

Therefore (19) can be rewritten as

$$\mathcal{L}(\eta, w; \beta) = -\frac{1}{4}\beta^T\beta - \frac{1}{4\lambda}\beta^T\Omega\beta + \beta^T l. \tag{21}$$

Noticing that

$$h(\beta) = \max_{\beta \geq 0} \mathcal{L}(\eta, w; \beta) = \begin{cases} \|\eta\|^2 + \lambda\|w\|^2 & \text{if } \phi_{\mathbf{\Omega}}(x)w - l \leq \eta \\ +\infty & \text{otherwise.} \end{cases} \tag{22}$$

is a "barrier" of the feasible set of the primal problem, we can use $h(\beta)$ as an objective function and rewrite (22) as

$$\min_{\eta,w} h(\beta) = \min_{\eta,w} \max_{\beta \geq 0} \mathcal{L}(\eta, w; \beta). \tag{23}$$

Since the strong duality holds, we can exchange the "max" and "min" operators in (22) without altering the value of the optimal result (Slater's condition):

$$p^* = \min_{\eta,w} \max_{\beta \geq 0} \mathcal{L}(\eta, w; \beta) = \max_{\beta \geq 0} \min_{\eta,w} \mathcal{L}(\eta, w; \beta) = d^*, \tag{24}$$

Thus we obtain the Lagrangian dual function $g(\beta)$:

$$\begin{aligned} g(\beta) &= \min_{\eta,w} \mathcal{L}(\eta, w; \beta) \\ &= \min_{\eta,w} \Big(-\frac{1}{4}\beta^T\beta - \frac{1}{4\lambda}\beta^T\Omega\beta + \beta^T l \Big) \end{aligned} \tag{25}$$

The optimal solution Ω of (25) is

$$\begin{aligned} d^* &= \min_{\mu>0,\mu_i\sum\mu_i r_i=1} \max_{\beta\geq 0} \Big\{ -\frac{1}{4}\beta^T\beta - \frac{1}{4\lambda}\beta^T\sum_{i=1}^{n}\mu_i\Omega_i\beta + \beta^T l \Big\} \\ &= \max_{\beta\geq 0} \min_{\mu>0,\sum\mu_i r_i=1} \Big\{ -\frac{1}{4}\beta^T\beta - \frac{1}{4\lambda}\beta^T\sum_{i=1}^{n}\mu_i\Omega_i\beta + \beta^T l \Big\} \\ &= \max_{\beta\geq 0} \Big\{ -\frac{1}{4}\beta^T\beta - \max_{\mu>0,\sum\mu_i r_i=1} \Big(\frac{1}{4\lambda}\beta^T\sum_{i=1}^{n}\mu_i\Omega_i\beta\Big) + \beta^T l \Big\} \\ &= \max_{\beta\geq 0} \Big\{ -\frac{1}{4}\beta^T\beta - \frac{1}{4\lambda}\max_i \Big(\frac{1}{r_i}\beta^T\Omega_i\beta\Big) + \beta^T l \Big\}. \end{aligned} \tag{26}$$

Finally, (26) can be reformulated as

$$\begin{aligned} \max_{\beta\geq 0,t} \quad & -\tfrac{1}{4}\beta^T\beta - \tfrac{1}{4\lambda}t + \beta^T l \\ \text{s.t.} \quad & t \geq \tfrac{1}{r_i}\beta^T\Omega_i\beta, \qquad i = 1,\ldots,n \\ & \beta_j \geq 0, \qquad j = 1,\ldots,k. \end{aligned} \tag{27}$$

where β are the dual variables of the primal problem in (16). Regarding (27) as a primal problem, the optimal dual variables (denoted as α_i) corresponding to each upperbound constraints in (27) are equal to $\mu_i r_i$, hence the optimal weights of μ_i on p kernels are $\frac{\alpha_i}{r_i}$, so that the optimal Ω is also obtained. This completes the proof. ■

Theorem 1 solves the optimal kernel learning for l, which contains partitions of binary clusters. The proof of a similar problem is first proposed on binary classification by Lanckriet *et al.* [20], then extended to multiple class discriminant problems by Ye *et al.* [40]. It is obvious to see, for the fixed l, learning the optimal kernel for clustering is similar to the binary classification problem. The main difference is in classification we need to solve the separating hyperplane, while in clustering we only need to learn the optimal kernel with the fixed l. Extending the binary partition l to multiple-class partitions L, we have the following theorem.

Theorem 2: Let L be the weighted cluster indicator matrix of multiple clusters defined as (8), $\phi(x)$ be the data matrix in kernel space, Ω be the kernel Gram matrix defined as $\Omega_{i,j} = \phi(x_i)\phi(x_j)^T$, $\lambda > 0$ be the regularization parameter on covariance matrix, given a set of n centered kernel matrices Ω_1, ..., Ω_n, the optimal kernel matrix Ω as the convex linear combination of n matrices which optimize (13)

$$F_4 = \max_{\Omega} \operatorname{trace}\Big(L^T\phi(x)^T\left(\phi(x)\phi(x)^T + \lambda I\right)^{-1}\phi(x)L\Big), \tag{28}$$

can be found by solving the following convex QCQP problem:

$$\begin{aligned} \max_{\beta_j, t} \quad & -\sum_{j=1}^{k} \tfrac{1}{4}\beta_j^T \beta_j - \tfrac{1}{4\lambda} t + \sum_{j=1}^{k} \beta_j^T l_j \\ \text{s.t.} \quad & t \geq \tfrac{1}{r_i} \sum_{j=1}^{k} \beta_j^T \Omega_i \beta_j, \quad i = 1, \ldots, n, \\ & \beta_j \geq 0, \quad j = 1, \ldots, k. \end{aligned} \tag{29}$$

Proof: According to Theorem 1, it is easy to obtain the solution of objective function F_4 as equal to the solution of the following function:

$$\min_{w_j, K} F_5(w_j, K) = \sum_{j=1}^{k} \Big(\|\phi(x) w_j - l_j\|^2 + \lambda \|w_j\|^2 \Big), \tag{30}$$

where $\{l_j | j = 1, \ldots, k\}$ are the k column vectors in L. Reusing the result of Theorem 2, it completes the proof. ■

By now we have solved the problem of learning the optimal Ω with given L. In the next step, we apply kernel k-means clustering to obtain partitions L from Ω.

Step 2: kernel k-means clustering. The kernel k-means clustering [11] maximizes the trace of between group distance scatter matrix in feature space as

$$\mathrm{Tr}(S_B^{\Phi}) = \mathrm{trace}\big(L^T \phi_{\Omega}(X)^T \phi_{\Omega}(X) L\big). \tag{31}$$

Since for a given centered kernel matrix, the covariance matrix only depends on λ, which is given in (12), the maximization of (40) is equivalent to

$$\mathrm{Tr}(S_B^{\Phi}) = \mathrm{trace}\big(L^T \phi_{\Omega}(X)^T C_{\Omega}^{-1} \phi_{\Omega}(X) L\big). \tag{32}$$

Thus Steps 1 and 2 of the AKKC algorithm optimize toward the same objective function. The overall algorithm converges locally.

4) Computational complexity and regularization: The complexity of AKKC is determined by the QCQP problem, which has the complexity $O(pk^3n^3)$ [40]. Because of the local optimal restriction, the performance is strongly dependent on the starting point. Practically, in heterogeneous data fusion, we try multiple starting points for each individual kernel to obtain the initial partition $L(0)$. The final result is obtained by selecting the maximum value of the objective function from multiple starting points. The overall complexity of the total algorithm is then $O(p^2k^3n^3)$. The regularization parameter λ of the covariance matrix is selected empirically, in all our following discussions, we set λ to 0.01 as a value that we observe to be practical. This

value may be problem dependent, and how to optimally set this value will be considered in future work.

When clustering with heterogenous data fusion, it is sometimes necessary to regularize the weights of data sources to avoid overfitting (penalize solutions with maximal (unit) weight on a single data set). One way to achieve this goal is to add a entropy-based regularization term [10] in the objective function. Another way is to set the minimal boundary of weights of data sources (so that each source must have some minimal contribution to the final integrated kernel) [7]. We use the minimal boundary method by setting the minimal values of weights to $\frac{0.25}{p}$, $\frac{0.5}{p}$, and $\frac{1}{p}$ respectively and compare their performance. Notice that when the boundary is set as $\frac{1}{p}$, it is equal to combining the average kernels. However, in our experiments the effect of regularization is not significant so in the following discussion all results are obtained without regularization on data sources.

B. Average combination of kernels

Instead of the complicated approach of tuning the weights in kernel fusion, one can also combine the kernels in an average manner as

$$\Omega = \sum_{i=1}^{p} \frac{1}{p} \Omega_i.$$

Regarding Ω as a new individual data that equally combines information of multiple data sources, one could apply standard clustering algorithms on this new combined data in kernel space. In this paper, we apply six standard clustering algorithms on the averagely combined kernel. Since these methods have been well studied in literature, we omit the details of their formulations here.

1) Kernel k-means: The kernel k-means algorithm [11] applied to the average kernel can be regarded as a simplified version of the AKKC algorithm using only Step 2 to obtain the partition.

2) Hierarchical Clustering: To apply hierarchical clustering methods in feature space, we first transform the kernel matrix into a distance matrix by calculating the distances between feature vectors [30],

$$\| \phi(x) - \phi(z) \|^2 = \langle \phi(x), \phi(x) \rangle - 2\langle \phi(x), \phi(z) \rangle + \langle \phi(z), \phi(z) \rangle. \tag{33}$$

Then we apply standard linkage clustering methods (single linkage, average linkage, complete linkage, and ward linkage) on the transformed distance matrix and obtain the partitions by

hierarchical clustering. In particular, the Ward linkage clustering algorithm based on average combination of kernels can be regarded as a special case of the WLCDM method proposed by Janssens *et al.* [14]. In WLCDM, the weights assigned on data sources are determined empirically, while in our paper, the weights are set as equal.

3) Spectral clustering: The spectral clustering algorithm we apply in this paper is proposed by Jordan and Weiss [16]. In our experiment, the Laplacian is constructed on the combined average kernel matrix.

IV. Clustering Evaluations for Data Fusion

A. Internal validation

1) Mean Silhouette Value (MSV): The classical silhouette value evaluates the "goodness" of partitions on the basis of two inputs: the clustering labels and the original data set. The silhouette value of a clustered object measures its distances with objects within the cluster versus the objects outside of the cluster [29]. The Silhouette value is defined as follows:

$$S(i) = \frac{\min(B(i, C_j) - W(i))}{\max[\min(B(i, C_j)), W(i)]}, \tag{34}$$

where $W(i)$ is the average distance from object i to all other objects within its cluster, and $B(i, C_j)$ is the average distance from object i to all objects in another cluster C_j . It follows from the formula that -1 $\leq S(i) \leq +1$. If the value is close to 1, it means that sample is "well clustered" and it was assigned to a very appropriate cluster. If the value is about zero, it means that sample could be assign to another closest cluster as well, and the sample lies equally far away from both clusters. If the value is close to -1, it means that sample is "misclassified" and is merely somewhere in between the clusters. The overall average silhouette width (MSV) for the entire data set is an intrinsic measurement about the overall quality of a clustering solution and it can vary with the number of clusters, which is an important clue to find the optimal cluster number.

2) Kernel Mean Silhouette Value (KMSV): For the problem of heterogeneous data fusion, the values of most internal validation indices are data dependent. The values are affected by the data structure, the dimensionality, and the scale of the data set. Apparently, the silhouette value in linear space cannot measure the "goodness" of partitions obtained in nonlinear space. Also, for some specific data structure, (e.g., the gene sequence data), calculating internal indices

is not easy. To tackle these problems, we propose a method, called the kernel silhouette value (KSV) [28], to calculate silhouette values directly from kernels. The KSV is a generalized formulation of SV, in particular, when the kernel is generated by a linear kernel function, the KSV is equivalent to the SV based on the squared Euclidean distance. The main advantages of KSV come from three aspects. First, the KSV can evaluate nonlinear partitions. Second, the KSV only requires the kernel matrix, hence it is easier to evaluate for some special data structures. Third, the calculation of KSV is more efficient than SV for high-dimensional data since all the pairwise distances can be directly retrieved from the kernels. The formulation of KSV is

$$KS(i) = \frac{\min(\varphi_B(i, C_j) - \varphi_W(i))}{\max[\min(\varphi_B(i, C_j)), \varphi_W(i)]}, \tag{35}$$

where $\varphi_W(i)$ is the average kernel distance from object i to all other objects within its cluster, and $\varphi_B(i, C_j)$ is the average kernel distance from object i to all objects in another cluster C_j. The kernel distance is calculated by the formulation presented in (33).

3) Modularity (Mod): For some sparse adjacency data, such as graph or network, the quality of a clustering can also be evaluated by calculating the modularity [25]. Up to a multiplicative constant, modularity measures the number of intra-cluster edges minus the expected number in an equivalent network with the same clusters but with edges given at random. Intuitively, in a good clustering there are more edges within (and fewer citations between) clusters than could be expected from random edge. The modularity Q_q of a clustering into q partitions is defined as

$$Q_q = \sum_{s=1}^{q} \left(e_{ss} - a_s^2 \right), \tag{36}$$

where $a_s = \sum_{r=1}^{q} e_{rs}$. Here, e_{rs} is the fraction of edges between nodes in partition r and s. Obviously, the larger the distortion between internal edges e_{ss} and expected internal edges a_s^2, the more modular the graph or network is (higher Q). By design, $Q < 1$.

B. External validation

A major problem in heterogenous data sets clustering is that internal indices might scale and evolute diversely by data sources. Moreover, when benchmarking internal indices with a varied number of clusters, they might show different trends, resulting in inconsistent suggestions about the optimal number of clusters. By contrast, external validation compares the clustering partitions obtained by the algorithm with a reference partition (usually assumed as ground-truth labels) so

it is independent of the data heterogeneity. For a data fusion problem, if the reference partition is given, performing model prediction and comparison based on external validation is easier because it gives a unique score. In this paper, we apply two external validations to evaluate the clustering performance.

1) Normalized Mutual Information (NMI): Mutual information is a symmetric measure to quantify the statistical information shared between two distributions. Let $\{c_i\}_{i=1}^n$ and $\{l_i\}_{i=1}^n$ be the set of indicators and the ground truth labels, respectively. The normalized mutual information is defined as:

$$NMI = \frac{2 \times H(\{c_i\}, \{l_i\})}{H(\{c_i\})H(\{l_i\})}, \tag{37}$$

where $H(\{c_i\}, \{l_i\})$ is the mutual information between $\{c_i\}_{i=1}^n$ and $\{l_i\}_{i=1}^n$, $H(\{c_i\})$ and $H(\{l_i\})$ are the entropy of indicators and labels. For a balanced clustering problem, if the indicators and the labels are independent, the mutual information approaches 0.

2) Rand Index (RI): For the samples $\{x_i\}_{i=1}^n$, let the vectors $\{c_i\}_{i=1}^n$ (denoted as $\mathcal{C}$) and $\{l_i\}_{i=1}^n$ (denoted as $\mathcal{P}$) be the corresponding cluster indicators and ground truth labels, respectively. Consider a pair of vectors (c_i, l_i). We refer to it as (1) a the number of pairs if both vectors belong to the same cluster in $\mathcal{C}$ and to the same group in $\mathcal{P}$, (2) b the number of pairs if both vectors belong to different clusters in $\mathcal{C}$ and to the different groups in $\mathcal{P}$. (3) c the number of pairs if the vectors belong to the same cluster in $\mathcal{C}$ and to different groups in $\mathcal{P}$, and (4) d if the vectors belong to different clusters in $\mathcal{C}$ and to the same group in $\mathcal{P}$. Rand Index is defined as [36]:

$$RI = \frac{a+b}{a+b+c+d}. \tag{38}$$

V. A UNIFIED FRAMEWORK OF CLUSTERING BY DATA FUSION

Based on the discussion of previous sections, we establish a unified framework for clustering by data fusion. Various algorithms are categorized as two main approaches, clustering ensemble and kernel fusion, and their conceptual structures are shown in Figure 1. The 13 algorithms and 5 evaluation methods that we implement are briefly defined in Table 1. The software, which contains HGPA, CSPA, and MCLA algorithms, is available for download from the author's website. We implement QMI, EAC-AL, AdacVote, and all kernel fusion algorithms in MATLAB. In ensemble clustering, the individual partition can also be obtained by various clustering

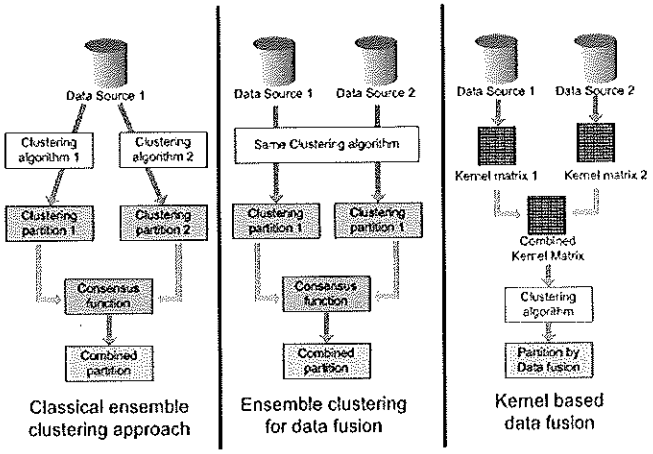


Fig. 1. **A conceptual overview of ensemble clustering and kernel fusion methods for data fusion**

TABLE I

ALGORITHMS AND EVALUATIONS IN COMPUTATIONAL FRAMEWORK

Abbreviations	Description
CSPA	Cluster-based similarity partitioning algorithm
HGPA	Hypergraph partitioning algorithm
MCLA	Meta clustering algorithm
QMI	Quadratic mutual information algorithm
EACAL	Evidence accumulation clustering by average linkage
AdacVote	Adaptive cumulative voting algorithm
AKKC	Adaptive kernel K-means clustering algorithm
KM	Kernel K-means clustering on averagely combined kernels
SL	Single linkage clustering on averagely combined kernels
AL	Average linkage clustering on averagely combined kernels
CL	Complete linkage clustering on averagely combined kernels
WL	Ward linkage clustering on averagely combined kernels
SC	Spectral clustering on averagely combined kernels
MSV	Mean Silhouette values calculated in linear space
KMSV	Mean Silhouette values calculated in kernel space
MOD	Modularity value for graph or network data
RI	Rand index validation with reference partitions
NMI	Normalized mutual information validation with reference partitions

algorithms. However, to keep the discussion concise, all the individual data partitions, as well as the partitions in the ensemble clustering algorithms, are obtained by k-means clustering in original space or kernel k-means clustering in kernel space.

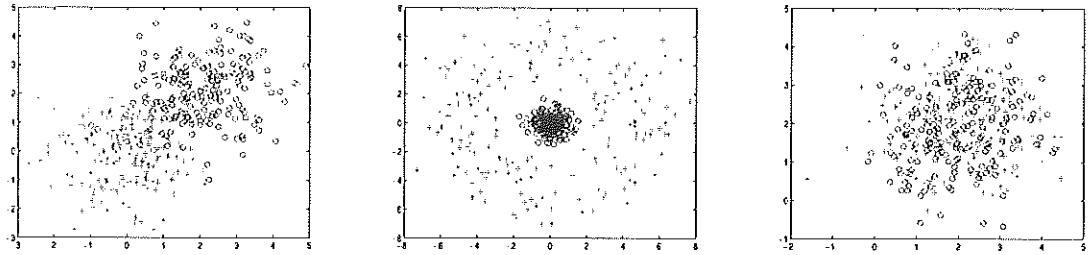


Fig. 2. Synthetic data sets. Each data source contains 400 samples equally divided into 2 classes.

VI. Experimental Results

A. Synthetic data sets

The first experiment is carried out on 3 synthetic data sources with intuitive patterns. We assume that 400 samples (equally divided into 2 labels as blue and red points in Figure 2) are represented in 3 different data sources. In the first source (denoted as normal data), the 2 clusters form two normal distributions with some degree of overlapping. In the second source (denoted as ring data), the 2 clusters show ring distributions. In the third one (denoted as random data), the labels are randomly assigned, which represents irrelevant data. Apparently, normal data contains the most informative partition in linear space while ring data contains obvious nonlinear partitions. In linear space, we apply k-means algorithm (set $k = 2$) on the 3 data sources. To obtain the nonlinear clustering, we first map the data sources into kernel space by two Radial Basis Functions (RBF) kernels with different width. The width of the RBF function is selected empirically as twice the average distance of a data point to its nearest neighbor in the whole data source (denoted as RBF$\times$1). For the other RBF kernel, twice this width is taken [7] (denoted as RBF$\times$2). Then we apply kernel k-means algorithm to obtain the partitions. The clustering performances of individual data sources obtained in linear and nonlinear space are shown in Figure 3 (brown bars).

We fuse the data sources for clustering, first in linear space then in nonlinear space. In linear space fusion, we combine the 3 data sources in their original dimensions. Since there is no natural dependency among samples in the above 3 synthetic data sources, we assume that all samples within the same label are equivalent to each other. Thus, in each run of clustering, we randomly permute the samples within each label and repeat the clustering 50 times. The

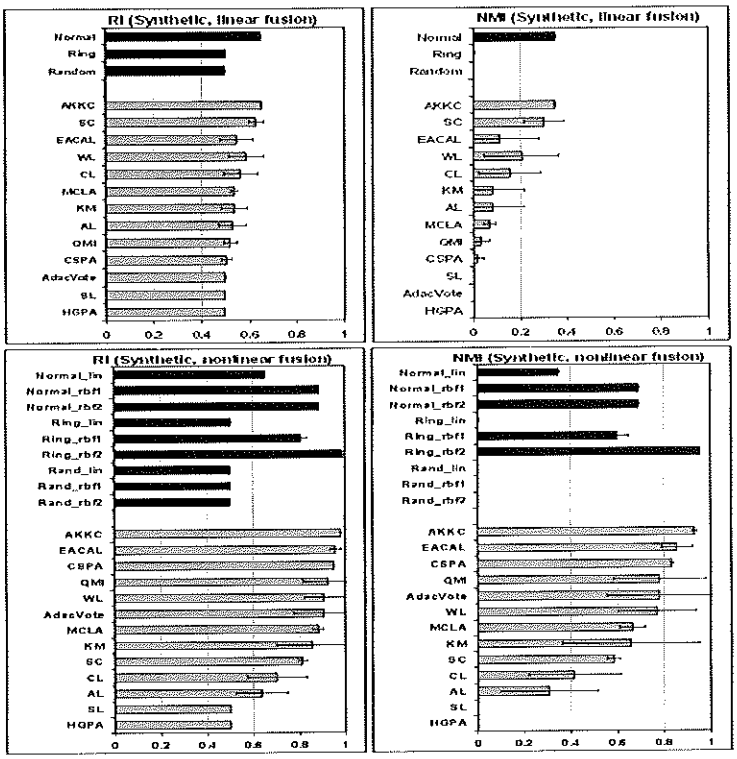


Fig. 3. **Performance of clustering on synthetic data fusion.** The upper two figures show RI and NMI of data fusion in linear space. The bottom two figures show results in nonlinear space. The performance of individual data source are shown as brown bars, the data fusion performances are shown in blue bars.

clustering performance is evaluated by comparing the program partitions with the labels of the samples. The performances obtained by all algorithms in linear space are shown in the left figure of Figure 3 (blue bars). AKKC obtains the best performance, which is almost the same as the best performance obtained on the individual data sets. This is because the optimization procedure assigns dominant weights to the most relevant data source (normalized weights: normal data 0.8495, ring data 0.0089, rand data 0.1416). Other algorithms do not perform as well, which is probably because in linear space, 2 out of 3 sources (ring data and rand data) are quite uninformative w.r.t. the clustering problem. Since other algorithms cannot determine the most relevant source and bias automatically towards it, the overall performance is dominated by the irrelevant sources.

In the next step, we combine data sources in nonlinear space. For clustering ensemble algorithms, 3 partitions are obtained on each data source (1 from original dimension and 2 from

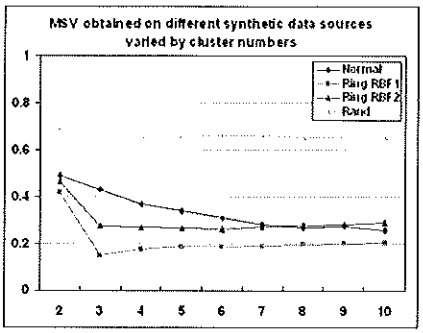


Fig. 4. **The MSV and KMSV obtained on synthetic data sources are affected by the heterogeneity of data.** The MSV of normal and rand data are calculated in the original dimensions of data sources. The KMSV are calculated on two RBF kernel matrices of ring data source.

RBF kernels) and then 9 partitions in total are combined to obtain the consensus partition. For kernel fusion methods, 9 kernels (1 linear and 2 RBF) are combined as an integrated kernel for clustering. The performance obtained by individual data (brown bars) and all fusion algorithms in nonlinear space are shown in Figure 3 (blue bars). When clustering with a single data source, nonlinear mapping significantly improves the performance on normal data and ring data. The best performance is obtained by applying the RBF×2 kernel to the ring data source. In data fusion, AKKC again performs best by data fusion and the weights assigned on sources are strongly biased towards the best RBF kernel (normal linear: 0.0001, normal rbf1: 0.0137, normal rbf2: 0.0166, ring linear: 0, ring rbf1: 0.4109, ring rbf2: 0.5497, rand linear: 0.0014, rand rbf1: 0, rand rbf2:0). Ensemble clustering algorithms perform much better in nonlinear space. This is because in nonlinear space, 4 out of 9 partitions are quite informative (normal RBF×1, normal RBF×2, ring RBF×1, ring RBF×2) so they provides a sufficient "common agreement" for clustering ensemble algorithms to obtain informative consolidate partition.

The evaluation of clustering by data fusion based on external validations easily assess the quality of partitions by unique ground-truth labels. However, the internal validation indices behave differently across data sources. To show this, in Figure 4 we present the MSV and KMSV obtained on different data sources with various numbers of clusters. It is hard to compare the quality of clustering models across data sources by internal validation. In this paper, we only compare internal indices within the same data source. When comparing internal indices on a

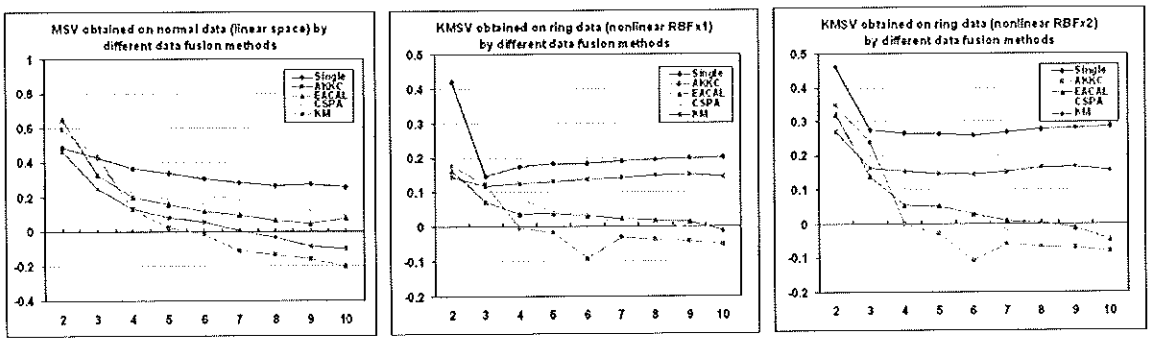


Fig. 5. **The MSV and KMSV curves obtained on synthetic data sources with various numbers of clusters.** For normal data, we present MSV in linear space. For ring data, we present KMSV values calculated on two RBF kernels

variable number of clusters for each data source, it is still straightforward to draw the conclusions about the optimal cluster number. The MSV or KMSV obtained on single source (normal and ring data) are maximized when at 2 clusters. We also calculate the indices obtained by different data fusion clustering algorithms, and the results of normal data (linear space) and ring data (2 nonlinear space) are shown in Figure 5.

B. Digit Recognition Data

In the second experiment, we adopt two UCI digit recognition data sources for clustering analysis. The first source is pen-based digit recognition data created by trainers input on a tablet with 500 by 500 pixels resolution. The second source is optical recognition data generated by the NIST OCR system [26]. In the literature, both data sources have been used separately as benchmark data for classification and clustering analysis. In this paper, we combine them for clustering analysis for the first time. The assumption here is similar to the synthetic data experiment: we assume that all samples within the same digit class are equivalent. For each clustering run, we randomly select 1500 samples (150 for each digit class) from each source and combine them for clustering and repeated 50 times. Since the application data sources presented in this paper mainly contain linear patterns: from now on we will focus on the data fusion approach in linear space. Clustering by data fusion in nonlinear space is certainly a very interesting topic and it will be one of the main directions in our future work. To evaluate the performance, we use the labels of 10 digit classes from the original UCI repository as reference partitions.

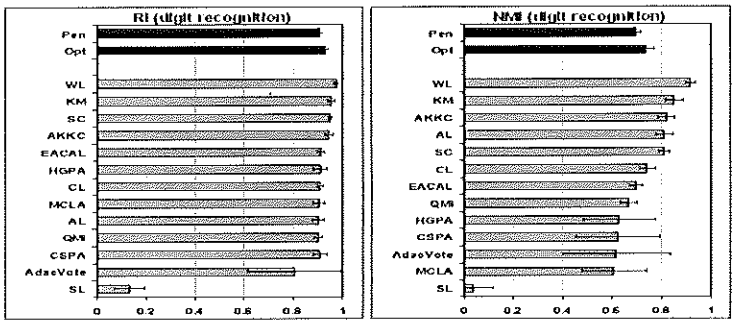


Fig. 6. **Performance of clustering on digit recognition data fusion**

The two UCI data sources are first clustered separately and their individual performances are shown in Figure 6 as brown bars. Optical data is more informative because its RI and NMI score are significantly higher than those of the pen-based data. When combining these two sources for clustering, 4 algorithms obtain significant improvement (WL, KM, SC, and AKKC) compared to individual data. To show how kernel fusion improve the clustering performance, we present a figure showing the plot of similarity matrices of optical data, pen data, and the combined matrix in Figure 7. It clearly shows that the optical data has better quality for clustering than pen data. When merging the similarity matrices together, the within-class similarities are enhanced. Especially, in optical data, most of the error come from digit 9 and digit 0 because they seem to be quite similar in the optical similarity matrix. However, in the pen data the partition between digit 9 and digit 0 is much clearer. After merging their matrices together, the distinction of digit 9 and digit 0 is much more obvious than it is in single data source.

The evolution of MSV with various numbers of clusters on pen and optical data sources is shown in Figure 8. Since optical data has better quality for clustering problem, its MSV curve correctly predicts the optimal cluster number (10 clusters). On the contrary, pen data is less informative than optical data and its MSV curve is not very reliable. We also show the MSV curve of 4 best fusion methods in the figure. On optical data, the 4 best algorithms show consistent trend over various numbers of clusters. However, on pen data, their trends are much different. In this experiment, the kernel fusion approach seems to work better than the ensemble clustering methods. The reason is similar to the previous experiment: when combining only 2 heterogeneous data sources, one is informative and another one is uninformative, the number

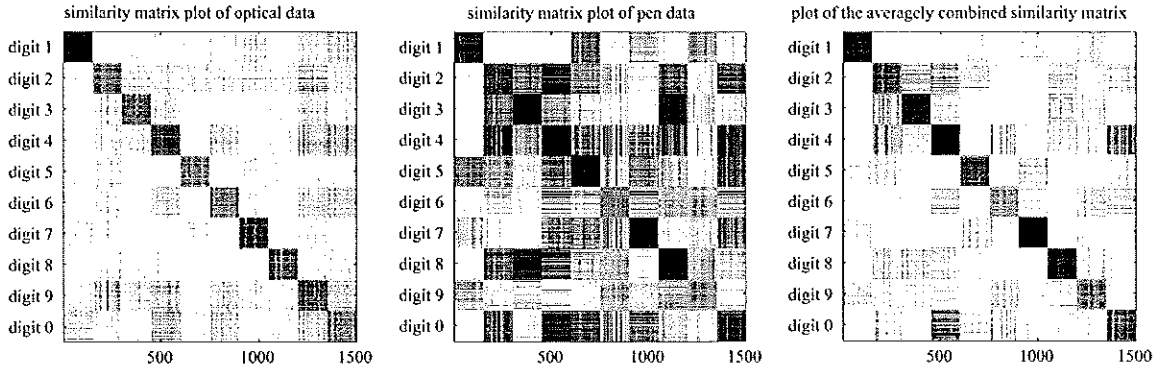


Fig. 7. **Plots of similarity matrices generated by 2 digit recognition data sources.** The data is grouped from digit 1 to 0 so the distinctions among each clusters are more obvious. The dots in the figure are generated by setting a threshold 0.8 on similarity values.

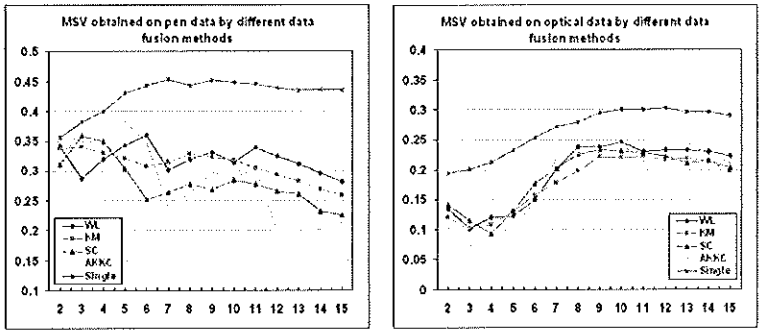


Fig. 8. **Evolution of MSV indices on digit recognition data sources with various numbers of clusters.**

of partitions to combine is insufficient for clustering ensemble methods to retrieve consolidate partition.

C. Journal sets clustering

The third experiment arises from a real application in scientometrics analysis [23]. We adopt a data set obtained from Web of Science (WoS) database by Thomson Scientific which contains articles, letters, notes and reviews from year 2002 till 2006. In the preprocessing steps, the ambiguities of journal names spelling, change of journal names, spelling of author names, bibliographic data and citations are resolved. We only keep the journals with more than 50 papers and more than 30 references or citations, resulting in 8,305 types of journals. We select

1,869 types of journals, which contain 7 field categories according to the labels provided by Essential Science Indicators (ESI). To obtain the structure mapping of these journals into 7 journal sets, we create two heterogeneous data sources and combine them in clustering analysis.

The first source is text mining data. The titles, abstracts, and keywords of these 1,869 journals are indexed by a text mining program based on the Jakarta Lucene API without controlled vocabulary. The index result contains 9,473,061 terms and we cut the Zipf curve of terms at the head and the tail to remove the rare terms, stop words, and common words. After Zipf cut, 669,860 meaningful terms are used to represent the journal in the vector space model (text data) and the weights of terms are calculated by TF-IDF weighting scheme. The second source is citation data. For these 1,869 journals, we only consider citations between papers from 2002 till 2006 and aggregate all paper-level citations into journal-by-journal citations. The direction of citations is ignored and symmetric citation data is obtained. To evaluate the performance, we reference the seven (ESI) labels as the ground-truth labels of journal assignment.

The performances evaluated by RI and NMI are shown in Figure 9. When clustering by single data source, text mining data provides more accurate journal partitions than citation data does. When combining text data and citation data for clustering, different strategies get quite diverse results. Again, ensemble clustering algorithms do not perform well when combining one informative source with another uninformative data source. Their RI and NMI scores are a compromise between the individual performance of text data and citation data. On the contrary, kernel fusion methods show satisfying results, where performances of several data fusion algorithms are almost equal to the best individual data. The mean values of weights assigned to each data source (learned by AKKC from 50 random repetitions) are 0.6139 for the text data and 0.3861 for the citation data. The AKKC weight is also strongly biased toward the good data source. Other average fusion methods (SC, KM, and WL) also perform well, probably because the kernel created on text data is dense while the kernel constructed on the citation data is sparse. When these two kernels are averagely combined, the effect of the sparse kernel matrix (citation data that has low performance) is dominated by the dense kernel matrix (text data that has good performance).

Since the dimensionality of text mining data is immense (669,860), the standard Matlab implementation is slow to calculate MSV in that space. In this case, KMSV is more efficient for computing, so we use KMSV to validate clusterings of text mining data source. As for citation

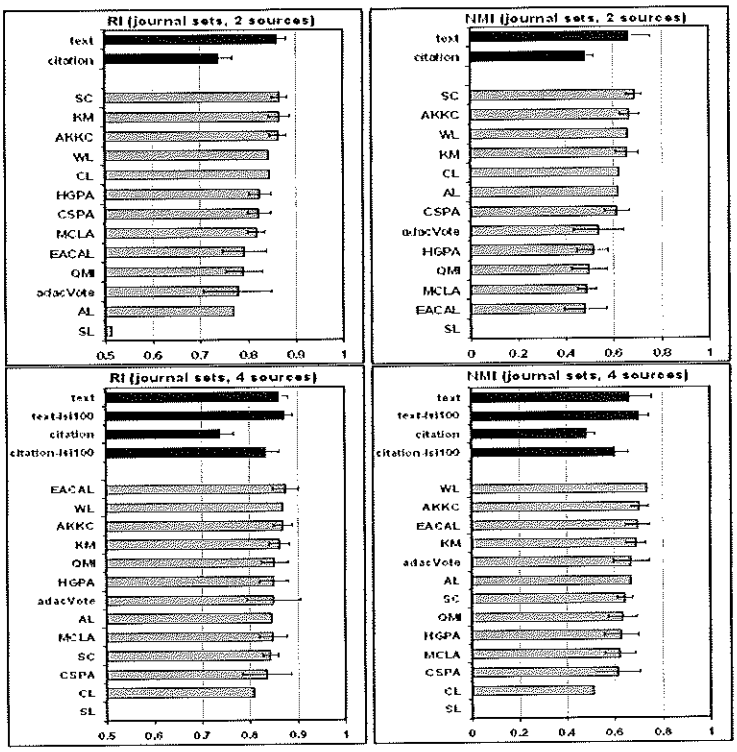


Fig. 9. **Performance of clustering obtained on journal set data fusion.** The upper two figures are fusion of 2 sources. The bottom two figures are fusion of 4 sources.

data it is a sparse adjacency matrix, thus the quality of clustering is evaluated by modularity. The curves of KMSV on text mining data and modularity on citation data obtained at various numbers of clusters are shown in Figure 10. In this experiment, comparing with single data source, data fusion approaches provide a more accurate estimation of the underlying partition of the journal data set.

To improve the performance of ensemble clustering algorithms, we involve 2 new data sources and mix them with the previously used 2 data sources. The new data is obtained by applying latent semantic indexing (LSI) on text and citation data. We trace the eigenvalues during dimensionality reduction and find in the 100-dimensional space spanned by principal components, about 90% of the information contained in text and citation data is preserved. So, we reduce text and citation data to the 100-dimensional space and construct 2 new data sets (text-LSI100 and citation-LSI100). All 4 data sources are then fused for clustering. Though in literature, the empirical optimal number of partitions for a clustering ensemble is suggested around 20, however, in order

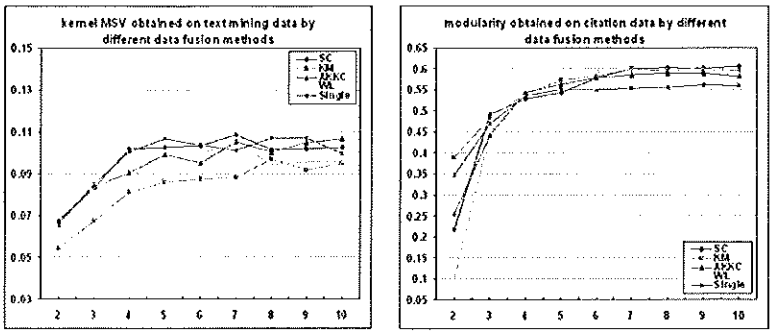


Fig. 10. **KMSV and MOD curve of journal clustering data sources on various numbers of clusters.** Data fusion clustering indicates the right optimal cluster number. In KMSV curve, SC, WL, and AKKC reach the maximum value at 7 clusters. The modularity curves obtained by KM, SC, and AKKC indicate that 7 clusters are optimal. However, the curves produced by single data clustering indicate the wrong optimal cluster number. The KMSV curve for text data alone reach the maximum at 5 clusters and 8 clusters, while the elbow of modularity curve for citation data alone is also at 5 clusters.

to keep the problem concise and clear, in this paper we use 4 data sources to address the problem.

By dimensionality reduction, the performance of citation data is significantly improved (RI scores increase from 0.7383 to 0.8332, NMI scores increase from 0.4844 to 0.6013). There is also considerable improvement on text data but it is not significant. The important discovery is, when combining these 4 sources, the performances obtained by the clustering ensemble are significantly improved. For example, EAC-AL obtains RI score of 0.8746 and NMI score of 0.6969, which is much better than the results with 2 sources (RI 0.7923, NMI 0.4811). This is mainly because among the 4 combined sources, 3 of them have good quality (text, text-LSI100, and citation-LSI100), so that clustering ensemble algorithms are able to find agreement among individual partitions and obtain stable consensus partitions. On the contrary, kernel fusion methods are affected much by the 2 new data sets, and their results are almost the same as with previous experiments.

The effectiveness of this application is that we can create a consolidated journal set network. Without the data fusion approach, the partitions obtained on text data and citation data may be different so it would be difficult to combine them in a unified structure. While in our approach both sources are used to obtain a consolidate partition with citation linkages for journal sets. We can also visualize the journal set network as shown in Figure 10. To interpret the journal partitions, we manually check the categories of journals contained in each cluster and map each

TABLE II

PARTITIONS OF JOURNALS OBTAINED BY AKKC CLUSTERING ALGORITHM AND THE POSSIBLE MAPPING TO ESI LABELS

Partitions	Example journals	Mapped ESI label
1.	J. Pharma. Therapy, E. J. Pharma., I. J. Pharma., B. J. Pharma., Pharma. Research, M. Pharma.	*Pharmacology Toxicology*
2.	J. Person and Social Psych., Person and Social Psych. Bulletin, Psych. Sci., Psych. B. R., Memory and Recognition, Psych. Bulletin	*Psychology*
3.	Hydrobiologia, J. Fish Biology, Theriogenology, M. Biology, Limnology and Oceanography, J.E.M. Biology and Ecology	*Plant and Animal Science*
4.	PNAS, Cell, Nature, Science, Neuron, Molecular and cellular biology	*Multidisciplinary*
5.	Plant Physiology, Plant Cell, Plant Journal, J. E. Botany, Plant M. B., Planta	*Molecular Biology and Genetics*
6.	Physical R. L., Physical R. B., Physical R. D., A. Physics L. B., Physical R. E. Physical R. A.	*Physics*
7.	A. J. Psychia., J. Clinical Psychia., Bio. Psychia., A. G. Psychia, B. J. Psychia., Bipolr disorders	*Neuroscience*

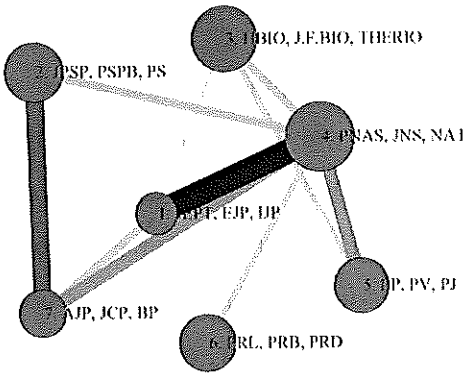


Fig. 11. **Visualization of journal sets structural mapping.** The vertexes are journal sets obtained by clustering. The edges are number of citations among journal sets. Big vertex represents for large number of journals. Thick edge represents for large number of citations. The clustering result in this figure is obtained by AKKC. The network is visualized by Pajek.

cluster (obtained by algorithm) to the ESI labels. The mapping between journal partitions with example journals and ESI labels are shown in Table 1. According to the journal set network and ESI label mapping, it is obvious to see the degree of interactions among journal sets. For example, many citations exist between *Pharmacology Toxicology* and *Multidisciplinary*, also there are strong citations between *Psychology* and *Neuroscience*. By contrast, for some less relevant domains such as *Molecular Biology and Genetics* and *Physics*, also *Psychology* and *Plant and Animal Science*, the citations between them are rare.

D. Disease gene clustering by genomic data fusion

The fourth experiment comes from bioinformatics research. We investigate a real biological problem of clustering human disease-causing genes. The ground truth labels of genes come from domain knowledge which are obtained from the Online Mendelian Inheritance in Man (OMIM) database. These genes have also been used as a benchmark data previously in gene prioritization systems. It consists of 537 human genes (620 occurrences) categorized in 29 diseases. We assume that in an optimal space, genes relevant to the same diseases are tightly related, so that they will form, to a certain extent, natural clusters by diseases. Most of these diseases are complex, resulting from the interplay of both environmental and numerous distinct genetic factors. It is thus often difficult to identify disease clusters with a single data set.

At the first step, we apply text mining techniques to retrieve information about these genes from MEDLINE biomedical literature repository. In our early study, we found that for the same set of documents, the choice of text mining configuration is a significant factor determining the quality of textual gene profiles in biological validations. In this paper, we also vary the text mining model by changing the domain vocabularies and weighting schemes and obtain 15 different models. We adopt 5 different ontologies, which describe different aspects of information contained in domain knowledge, to construct domain vocabularies. Based on these domain vocabularies, we index 288,177 MEDLINE titles and abstracts, and merge the document-by-term profiles into gene-by-term profiles with reference of gene-to-abstract of EntrezGene GeneRIF. The term score of textual profile is determined by the occurrence of the term in the MEDLINE corpus, calculated in 3 different schemes. We try IDF, TF-IDF, and binary scheme to calculate the value of each term. In this way a gene is expressed in a high-dimensional space spanned by the terms, denoted as textual gene profile. The details about the construction of textual gene profiles are available in [12], [42].

To derive a benchmark to investigate the clustering performance, we enumerate all the pairwise combinations of the 29 diseases (resulting in 406 binary clustering tasks). We realize that for a complicate situation, such as human disease genes, overfitting on specific pairwise diseases clustering might cause large generalization error on other pairs. So, we do not consider the good or bad performance of any single clustering task of pairwise disease clustering. Instead, all 406 tasks are repeated 50 times (resulting in 20,300 runs of clustering per kernel). In each repetition,

the average performance of all 406 tasks is used as the performance indicator of this run. Then the mean value of 50 repetitions is reported as final result. Since the identification of overlapping clusters is not the main goal of our experiment, the overlapping genes are removed for each clustering task. Overlapping clusters is obviously an important feature of gene clustering, so we will address this problem in future work.

We first investigate the performance obtained by the single text mining model. Similar to the case in gene prioritization, the clustering performance of textual profiles is also strongly affected by the choice of domain vocabularies and term weighting schemes. As shown in Figure 12, IDF based profiles generally perform better than TF-IDF and binary profiles. For IDF profiles, eVOC, LDDB, and OMIM vocabularies perform better than GO and MeSH. In practical research, it is a non-trivial issue to select the best textual model for clustering because we cannot split the data into training and test data and use training data for model selection. In our experiment, we simply combine all these 15 textual profiles for data fusion based clustering. After data fusion, the performance is significantly improved. The RI and NMI score obtained by WL, EACAL, and AKKC are much better than the best individual textual profile (LDDB-IDF). Since each textual profile is a specific view about how the genes are expressed in literature, the merit is that combining multiple views for clustering can indeed improve the biological relevance of partitions. We notice that an alternative choice of combining different text mining views is to merge the domain vocabularies. To compare its performance, we merge the 5 domain vocabularies as a big vocabulary and re-index the MEDLINE corpus to obtain the textual profiles (with 23,574 terms), denoted as allvoc-IDF, allvoc-TFIDF, and allvoc-binary. We also try to index the MEDLINE corpus without using any vocabulary and obtain huge textual profiles containing 259,815 terms, denoted as novoc-IDF, novoc-TFIDF, and novoc-binary. As shown in Figure 12, allvoc and novoc profiles all perform worse than the best individual profile, which means the semantic content of the domain vocabulary is context dependent and only enriched in its original ontology. When discarding the ontological context and merging the vocabularies together, we actually lose this semantic effect. Moreover, the "merging after splitting" approach effectively combines semantic effects of different ontologies and improves the biological relevance of the model.

We further combine the previously discussed 15 textual profiles with gene experimental profiles obtained from 11 other biological genomic sources. The type of data source, the number of features, the kernel functions, and references of the data sets are presented in Table 3. The

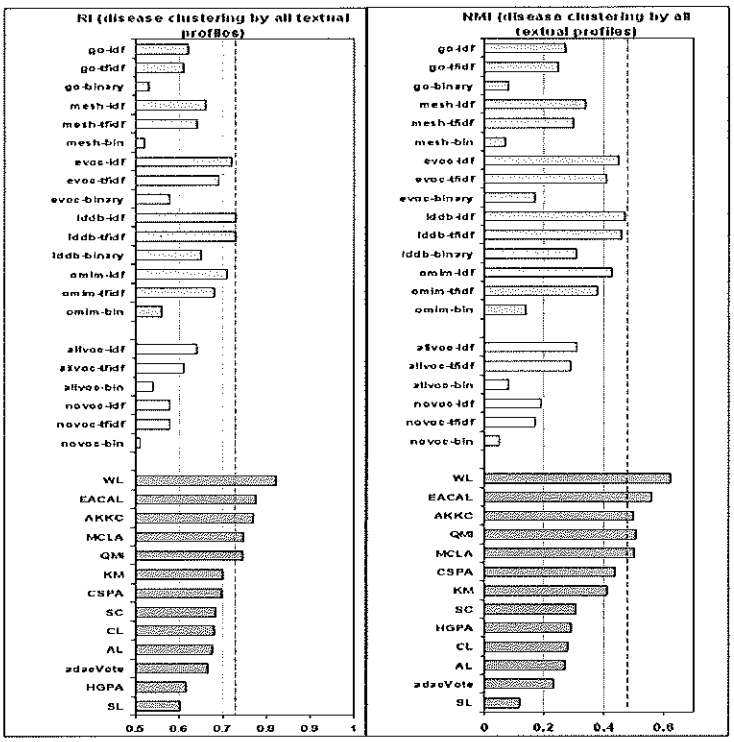


Fig. 12. **Performance of disease relevant gene clustering obtained on textual gene profiles and their integration.** The orange bars are performance obtained on individual textual profiles. The white bars are the non-voc and all-voc performances. The blue bars are data fusion results by combining 15 textual profiles for clustering. The red dotted line is the performance of best individual textual profile.

TABLE III

LIST OF DATA SOURCES AND THEIR CORRESPONDING KERNELS USED FOR CLUSTERING.

Data source	No. of features	Kernel matrices	Availability and Reference
Microarray 1	158	linear	[33]
Microarray 2	158	linear	[35]
EST	167	linear	[8]
Gene Ontology	8643	linear	[13]
KEGG	314	linear	[18]
Interpro	4708	linear	[24]
Motif	674	linear	[2]
Sequence (Blast)	-	blast score	[41]
Sequence (String)	-	2,3,4mer String kernel	[22]
Text GO	10249	linear (idf, tf-idf and binary)	[12], [42]
Text MeSH	17201	linear (idf, tf-idf and binary)	[12], [42]
Text eVOC	1496	linear (idf, tf-idf and binary)	[12], [42]
Text LDDB	933	linear (idf, tf-idf and binary)	[12], [42]
Text OMIM	3462	linear (idf, tf-idf and binary)	[12], [42]

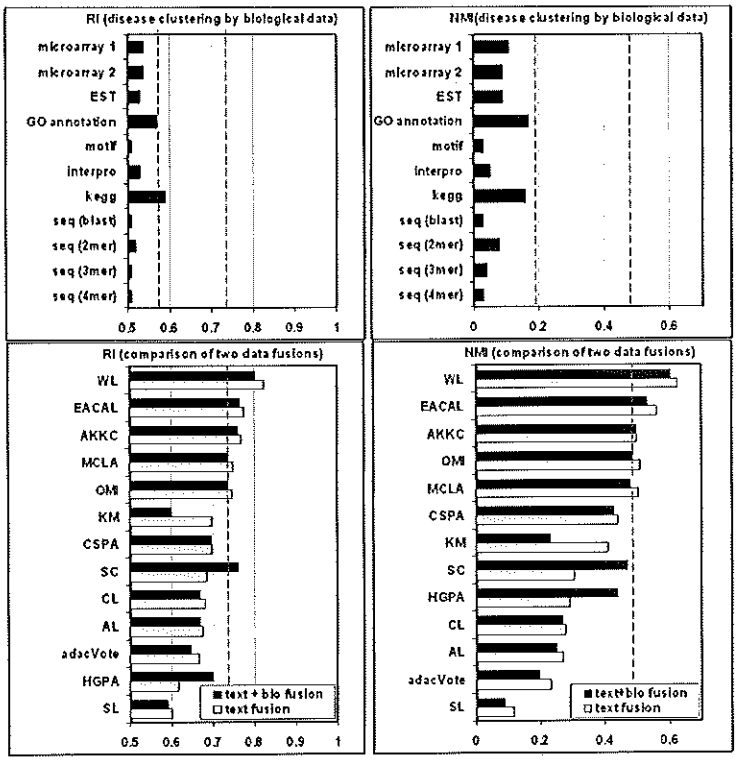


Fig. 13. **Performance of disease relevant gene clustering obtained by combing textual data with biological data.** The upper two figures are RI and NMI scores of individual biological data source. The bottom two figures are the comparison of fusing textual profiles with fusing textual and biological profiles. The blue and red dotted lines are respectively the worst and best performance on individual textual profile.

identification of complex diseases via clustering is a difficult problem, it is often hard to discover disease relevant partitions from a single experimental data. Moreover, since textual profiles only contains reported knowledge, while novel discovery may only exist in biological experimental data, the integration of text-mining data with biological experimental data is a promising approach to balance the reliability and novelty of biological knowledge discoveries. Unfortunately, as shown in Figure 13, the performances on the 11 selected biological data are not comparable with the textual performance. Hence, in this problem, combining textual profiles with biological data is actually fusing informative sources with noninformative sources, which is a typical scenario in heterogeneous data fusion. Our motivation, combining textual data with biological data, is thus to investigate whether the data fusion model can bias towards the most informative source; and furthermore perform no worse than the best individual source. We

compare the performance of two data fusion approaches. The first approach is the fusion of 15 textual profiles discussed earlier. The second approach is the fusion of 15 textual plus 11 biological profiles. As shown in Figure 13, the performances of WL, EACAL, and AKKC is robust with respect to the uninformative sources, and the performance of the two approaches are almost the same. Especially, WL performs well on the disease gene clustering problem and its fusion performance improvement is very significant. By contrast, the KM algorithm decreases significantly when biological data is involved. In particular, AKKC and KM are the same type of algorithms based on kernel k-means clustering, where the main difference is AKKC adaptively weighs the data sources while KM averagely combines all sources. So in the case of combining informative source with uninformative source, the adaptive algorithm can bias automatically towards the "good" source.

In this experiment we present an approach of combining different text mining models and biological data sources, to identify disease relevant partition of genes. Though in our problem biological data performs worse than textual data, however, this is fully dependent on the specific data and problem. If the clustering analysis is carried out for other problems where biological data is more informative, the situations and results might be totally different.

E. Statistical significance of performance

To guarantee the statistical significance of the discussed results, we apply t-test to calculate the p value of clustering performance obtained by randomization. Since the topic of this paper is to compare results of data fusion with results on a single data source, in Table 4 we compare data fusion performance with the best individual data source of all previously discussed experiments. All statistics reported in the table are obtained from 50 random repetitions.

VII. Discussion and Conclusion

In this paper, we investigated the problem of clustering by data fusion, which is a new problem with the potential of many interesting and useful applications. Our main contributions can be summarized as follows.

As one of the main results, we proposed a computational framework to combine heterogeneous data sources for clustering. This framework consists of two main approaches, ensemble clustering and kernel fusion. In the ensemble clustering approach we discussed some important existing

TABLE IV

STATISTICAL ANALYSIS OF PERFORMANCES

	RI			NMI		
	mean	std.	p value	mean	std.	p value
1.a Synthetic data sources fusion in linear space						
Best single data	0.6506	0.0032	-	0.3505	0.0038	-
AKKC	0.6523	0.0030	0.0097	0.3525	0.0061	0.0489
SC	0.6275	0.0337	5.10×10^{-6}	0.2927	0.0932	2.93×10^{-5}
WL	0.5875	0.0737	2.56×10^{-8}	0.2068	0.1592	5.84×10^{-9}
1.b Synthetic data sources fusion in nonlinear space						
Best single data	0.9900	0	-	0.9596	0	-
AKKC	0.9807	0.0021	4.98×10^{-52}	0.9309	0.0063	4.27×10^{-54}
CSPA	0.9528	0.0022	4.46×10^{-19}	0.8351	0.0062	1.83×10^{-115}
EACAL	0.9355	0.0361	4.16×10^{-18}	0.8027	0.0891	6.69×10^{-22}
2 Digit recognition data fusion						
Best single data	0.9278	0.0123	-	0.7312	0.0317	-
WL	0.9783	0	1.17×10^{-18}	0.9248	0	9.17×10^{-27}
KM	0.9520	0.0146	2.24×10^{-14}	0.8434	0.0353	1.86×10^{-30}
SC	0.9507	0.0087	3.62×10^{-18}	0.8002	0.0272	3.14×10^{-20}
AKKC	0.9473	0.0143	7.51×10^{-11}	0.8206	0.0333	1.32×10^{-24}
3.a Journal sets data fusion with 2 sources						
Best single data	0.8618	0.0197	-	0.6627	0.0920	-
SC	0.8704	0.0173	0.1529	0.6873	0.0320	0.0691
KM	0.8659	0.0226	0.5454	0.6580	0.0470	0.7558
AKKC	0.8635	0.0178	0.7796	0.6683	0.0406	0.6997
WL	0.8441	0	2.60×10^{-4}	0.6623	0	0.9695
3.b Journal sets data fusion with 4 sources						
Best single data	0.8730	0.0169	-	0.7062	0.0361	-
EACAL	0.8764	0.0192	0.5576	0.7001	0.0468	0.6453
KM	0.8698	0	0.3980	0.7364	0	6.25×10^{-4}
AKKC	0.8698	0.0207	0.7313	0.7044	0.0340	0.9828
KM	0.8632	0.0216	0.1199	0.6877	0.0413	0.1395
4.a Disease gene clustering by fusing 15 textual sources						
Best single data	0.7308	0.0047	-	0.4706	0.0077	-
WL	0.8229	0	7.97×10^{-47}	0.6253	0	6.45×10^{-46}
EACAL	0.7783	0.0035	3.39×10^{-31}	0.5655	0.0065	1.69×10^{-33}
AKKC	0.7673	0.0046	3.32×10^{-25}	0.5431	0.0085	4.34×10^{-27}
MCLA	0.7483	0.0043	7.50×10^{-15}	0.5037	0.0078	4.60×10^{-16}
QMI	0.7477	0.0043	1.89×10^{-14}	0.5139	0.0084	2.46×10^{-19}
4.b Disease gene clustering by fusing 15 textual profiles and 11 biological sources						
Best single data	0.7308	0.0047	-	0.4706	0.0077	-
WL	0.8040	0	7.40×10^{-42}	0.6061	0	9.92×10^{-44}
EACAL	0.7632	0	1.24×10^{-28}	0.5320	0	7.30×10^{-31}
AKKC	0.7464	0.0056	1.09×10^{-11}	0.5019	0.0096	8.90×10^{-14}
MCLA	0.7380	0.0009	5.06×10^{-8}	0.4813	0.0016	3.99×10^{-7}
QMI	0.7365	0.0035	9.31×10^{-5}	0.4890	0.0060	3.18×10^{-10}

algorithms and extended them to data fusion problems. In the kernel fusion approach, we created several new clustering algorithms in kernel space. For k-means clustering, we proposed an adaptive clustering algorithm to weigh different data sources by learning the optimal kernel. For hierarchical clustering, we extended the classical linkage methods into kernel distance space and implemented four algorithms based on average combination of kernels. Of course, there is a trade-off in using weighted approach with respect to the cost of computation complexity. Most of the algorithms in the framework are easy to implement, furthermore, the framework is also extendable to create novel and complex algorithms.

We demonstrate the effectiveness of the computational framework on three real applications. These applications arise from different research domains: multi-model signal processing, information retrieval, and bioinformatics. In these applications, we implemented 13 different data fusion algorithms and cross-compared their clustering performance with clustering results obtained on individual data sources. The main finding is that by data fusion, without the non-trivial validation of performance on each data source, we gain significant enhancement on overall performance or equivalent performance as the best individual data source. Kernel fusion method seems to work well on heterogeneous data fusion. Especially, Ward linkage and adaptive clustering (AKKC) always perform well in all experiments. By contrast, for the ensemble clustering approach sometimes it might be hard to obtain consolidated partitions with insufficient individual partitions, we have also shown that if we first generate multiple partitions on each data source then fuse sufficient partitions, the performance of the clustering ensemble can also be significantly improved.

Clustering by data fusion is a new problem and there are many further issues needed to be addressed in further studies, for example, the integration of data sources in nonlinear space, identification of optimal cluster number among different data sources, etc. Some other ongoing issues are also important for large-scale problems (with immense number of data samples and data sources) [34], for example, how to reduce the computation complexity of integration algorithms, how to find an efficient heuristic to combine data sources, etc. Furthermore, the problem of combining data fusion among data sources with feature selection techniques on individual data source is also an interesting issue to be discussed in future work.

REFERENCES

[1] S. Aerts, D. Lambrechts, S. Maity, P. Van Loo, B. Coessens, F. De Smet, L.C. Tranchevent, B. De Moor, P. Marynen, B. Hassan, P. Carmeliet, and Y. Moreau, "Gene prioritization through genomic data fusion", Nature Biotechnology, vol. 24, no.5, pp. 537-544, 2006.

[2] S. Aerts, P. Van Loo, G. Thijs, H. Mayer, R. de Martin, Y. Moreau, and B. De Moor, "Toucan 2: the all-inclusive open source workbench for regulatory sequence analysis", Nucleic Acids Research, vol. 33, pp. W393-396, 2005.

[3] S. Asur, D. Ucar, and S. Parthasarathy, "An ensemble framework for clustering proteinprotein interaction networks", Bioinformatics, vol. 23, no.13, pp. i29-i40, 2007.

[4] H.G. Ayad and M.S. Kamel, "Cumulative Voting Consensus Method for Partitions with a Variable Number of Clusters", IEEE Trans. PAMI, vol. 30, issue 1, pp. 160-173, 2008.

[5] J. Chen, Z. Zhao, J.P. Ye, and H. Liu, "Nonlinear Adaptive Distance Metric Learning for Clustering", ACM KDD07, 2007.

[6] I. Csiszar and G. Tusnady, "Information geometry and alternating minimization procedures", Statistics and Decisions, Supplementary Issue 1, pp. 205-237, 1984.

[7] T. De Bie, L.C. Tranchevent, L.M. Van Oeffelen, and Y. Moreau, "Kernel based data fusion for gene prioritization", Bioinformatics, vol 23, no. 13, i125-i132, 2007.

[8] P. Flicek, B.L. Aken, K. Beal, B. Ballester, M. Caccamo, Y. Chen, L. Clarke, G. Coates, F. Cunningham, and T. Cutts, "Ensemble 2008", Nucleic Acids Research, vol. 36, pp. D707-714, 2008.

[9] A.L.N. Fred and A.K. Jain, "Combining Multiple Clusterings Using Evidence Accumulation", IEEE Trans. PAMI, vol.27, issue 6, 835-850, 2005.

[10] J. Friedman J and J. Meulman, "Clustering objects on subsets of attributes", Technical report, Stanford University, 2002.

[11] M. Girolami, "Mercer Kernel-Based Clustering in Feature Space", IEEE Tran. Neural Networks, vol. 13, no. 3, 2002.

[12] P. Glenisson, B. Coessens, S. Van Vooren, J. Mathys, Y. Moreau and B. De Moor, "TXTGate: profiling gene groups with text-based information", Genome Biology, vol. 5, no. 43, 2004.

[13] The Gene Ontology Consortium, "Gene Onotology: tool for the unification of biology", Nature Genetics, vol. 25, pp. 25-29, 2000.

[14] F. Janssens, W. Glänzel, and B. De Moor, "Dynamic hybrid clustering of bioinformatics by incorporating text mining and citation analysis", Proc. 13th ACM SIGKDD, 2007.

[15] F. Janssens, W. Glänzel, and B. De Moor, "A hybrid mapping of information science", Scientometrics, vol. 75, no. 3, pp. 607-631, 2008.

[16] M. Ng.A. Jordan and Y. Weiss, "On spectral clustering: analysis and an algorithm", Proc. of Adavnaces in Nerual Information Processing Systems, vol. 14, 2002.

[17] S. Mika, "Kernel Fisher Discriminants", Ph.D dissertation, Technology University of Berlin, 2002.

[18] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, and T. Tokimatsu, "KEGG for linking genomes to life and the environment", Nucleic Acids Research, vol. 36, pp. D480-484, 2008.

[19] R. Kustra and A. Zagdanski, "Data-Fusion in Clustering Microarray Data: Balancing Discovery and Interpretability", IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 99, no.1, 2007, doi:10.1109/TCBB.2007.70267.

[20] G. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui, and M.I. Jordan, "Learning the kernel Matrix with Semidefinite Programming", JMLR, vol. 5, pp.27-72, 2004.

[21] T. Lange and J.M. Buhmann, "Fusion of Similarity Data in Clustering", NIPS 2005.

[22] C. Leslie and R. Kuang, "Fast String Kernels using Inexact Matching for Protein Sequences", Journal of Machine Learning Research, vol. 5, pp. 1435-1455, 2004.

[23] X.H. Liu, S. Yu, Y. Moreau, B. De Moor, W. Glänzel, and F. Janssens, "Hybrid Clustering of Text Mining and Bibliometrics Applied to Journal Sets", in Proceeding of SIAM Data Mining 09, Sparks, USA, 2009.

[24] N.J. Mulder, R. Apweiler, T.K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bork, V. Buillard, L. Cerutti, and R. Copley, "New developments in the InterPro database", Nucleic Acids Research, vol 35, pp. D224-228, 2007.

[25] M.E.J. Newman, "Modularity and community structure in networks", PNAS US, vol. 103, no. 23, 2006.

[26] National Institute of Standards and Technology, "NIST form-based handprint recognition system (release 2.2)", `http://www.itl.nist.gov/iaui/894.03/databases/defs/nist_ocr.html`.

[27] K. Pelckmans, S. Van Vooren, B. Coessens, J.A.K. Suykens, and B. De Moor, "Mutual Spectral Clustering: Microarray Experiments Versus Text Corpus", PMSB 2006.

[28] N.L.M.M. Pochet, F. Ojeda, F. De Smet, T. De Bie, J.A.K. Suykens, and B. De Moor, "Kernel Clustering for Knowledge Discovery in Clinical Microarray Data Analysis", Kernel Methods in Bioengineering, Signal and Image Processing (C,C-Valls, J.L.R-Alvarez and M.M-Ramon, editors), IDEA group publishing, 2007.

[29] P. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis", Journal of Computational and Applied Mathematics, vol. 20, no. 1, pp.5365, 1987.

[30] J. Shawe-Taylor and N. Cristianini, "Kernel methods for Pattern Analysis", Cambrdige University Press, 2004.

[31] B. Schölkopf, "The Kernel Trick for Distances", in Proceeding of NIPS, pp. 301-307, 2000.

[32] A. Strehl and J. Ghosh, "Cluster Ensembles: A Knowledge Reuse Framework for Combining Multiple Partitions", Journal of Machine Learning Research, vol. 3, pp. 583-617, 2002.

[33] C.G. Son, S. Bilke, S. Davis, B.T. Greer, J.S. Wei, C.C. Whiteford, Q.R. Chen, N. Cenacchi, and J. Khan, "Database of mRNA gene expression profiles of multiple human organs", Geneome Research, vol. 15, pp. 443-450.

[34] 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.

[35] A.I. Su, M.P. Cooke, K.A. Ching, Y. Hakak, J.R. Walker, T. Wiltshire, A.P. Orth, R.G. Vega, L.M. Sapinoso, and A. Moqrich, "Large-scale analysis of the human and mouse transcriptomes", Proc. Nalt. Acad. Sci. USA, vol. 99, pp. 4465-4470, 2002.

[36] S. Theodoridis and K. Koutroumbas, "Pattern Recognition (2nd Edition)", Elsevier Science, 2003, USA.

[37] 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.

[38] D.M. Wolf, L.F. Bodin, I. Bischofs, G. Price, J. Keasling, and A.P. Arkin, "Memory in Microbes: Quantifying History-Dependent Behavior in a Bacterium", PLOS one, vol. 3, issue 2, e1700, 2008.

[39] R. Xu and D. Wunsch II, "Survey of Clustering Algorithms", IEEE Trans. on Neural Networks, vol. 16, no. 3, 2005.

[40] 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.

[41] J. Ye, S. McGinnis, and T.L. Madden, "BLAST: improvements for better sequence anaysis", Nucleic Acids Research, vol. 34, pp. W6-9, 2006.

[42] S. Yu, S. Van Vooren, L.C. Tranchevent, B. De Moor, and Y. Moreau, "Comparison of vocabularies, representations and ranking algorithms for gene prioritization by text mining", Bioinformatics, vol. 24, no. 16, pp. i119-i125, 2008.

[43] Z. Yu, H-S. Wong, and H. Wang, "Graph-based consensus clustering for class discovery from gene expression data", Bioinformatics, vol.23, no.21, pp. 2888-2896, 2007, doi:10.1093/bioinformatics/btm463.