A Semisupervised Classification Approach for Multidomain Networks With Domain Selection

Chuan Chen, Jingxue Xin, Yong Wang, Luonan Chen, and Michael K. Ng

***Abstract*— Multidomain network classification has attracted significant attention in data integration and machine learning, which can enhance network classification or prediction performance by integrating information from different sources. Despite the previous success, existing multidomain network learning methods usually assume that different views are available for the same set of instances, and thus, they seek a consistent classification result for all domains. However, in many real-world problems, each domain has its specific instance set, and one instance in one domain may correspond to multiple instances in another domain. Moreover, due to the rapid growth of data sources, different domains may not be relevant to each other, which asks for selecting domains relevant to the target/focused domain. A key challenge under this setting is how to achieve accurate prediction by integrating different data representations without losing data information. In this paper, we propose a semisupervised classification approach for a multidomain network based on label propagation, i.e., multidomain classification with domain selection (MCS), which can deal with the cross-domain information and different instance sets in domains. In particular, with sparse weight properties, the proposed MCS can automatically identify those domains relevant to our target domain by assigning them higher weights than the other irrelevant domains. This not only significantly improves a classification accuracy but also helps to obtain optimal network partition for the target domain. From the theoretical viewpoint, we equivalently decompose MCS into two simpler subproblems with analytical solutions, which can be efficiently solved by their computational procedures. Extensive experimental results on both synthetic and real-world data sets empirically demonstrate the advantages of the proposed approach in terms of both prediction performance and domain selection ability.**

***Index Terms*— Domain selection, multidomain classification, network integration, semisupervised learning, sparsity.**

Manuscript received August 2, 2016; revised April 21, 2017 and November 16, 2017; accepted May 9, 2018. Date of publication June 14, 2018; date of current version December 19, 2018. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0505500 and Grant 2016YFB1000101, in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDB13040700, and in part by the National Natural Science Foundation of China under Grant 91529303, Grant 31771476, Grant 81471047, Grant 91730301, Grant 61671444, and Grant 61621003. *(Corresponding authors: Luonan Chen; Michael K. Ng.)*

C. Chen is with the National Engineering Research Center of Digital Life, School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China, and also with the Department of Mathematics, Hong Kong Baptist University, Hong Kong (e-mail: chenchuan@mail.sysu.edu.cn).

J. Xin is with the Key Laboratory of Systems Biology, CAS Center for Excellence in Molecular Cell Science, Institute of Biochemistry and Cell Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai 200031, China, also with the CAS Center for Excellence in Animal Evolution and Genetics, Chinese Academy of Sciences, Kunming 650223, China, also with the Key Laboratory of Management, Decision and Information Systems, Center for Excellence in Mathematical Sciences, National Center for Mathematics and Interdisciplinary Sciences, Academy of Mathematics and System Science, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Center for Excellence in Animal Evolution and Genetics, Chinese Academy of Sciences, Kunming 650223, China (e-mail: jxxin@amss.ac.cn).

Y. Wang is with the Key Laboratory of Management, Decision and Information Systems, Center for Excellence in Mathematical Sciences, National Center for Mathematics and Interdisciplinary Sciences, Key Laboratory of Management, Decision and Information Systems, Center for Excellence in Mathematical Sciences, National Center for Mathematics and Interdisciplinary Sciences, Academy of Mathematics and System Science, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, and also with the Center for Excellence in Animal Evolution and Genetics, Chinese Academy of Sciences, Kunming 650223, China (e-mail: ywang@amss.ac.cn).

L. Chen is with the Key Laboratory of Systems Biology, CAS Center for Excellence in Molecular Cell Science, Institute of Biochemistry and Cell Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai 200031, China, and also with the CAS Center for Excellence in Animal Evolution and Genetics, Chinese Academy of Sciences, Kunming 650223, China (e-mail: lnchen@sibs.ac.cn).

M. K. Ng is with the Department of Mathematics, Hong Kong Baptist University, Hong Kong (e-mail: mng@math.hkbu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2837166

I. Introduction

NETWORK-STRUCTURED data are usually represented as an undirected graph, where each node represents an instance and each edge represents a relationship between two instances. For example, in protein interaction networks, proteins are represented as nodes, and relationships among proteins, such as physical interactions and expression similarities, are represented as edges. A semisupervised learning problem on a network is to assign the instance label based on the information of the labeled instances and unlabeled instances (nodes) on a network. Graph-based semisupervised learning methods [5], [6], [16], [17], [28], [30] have been widely used in many practical applications because of their flexibility, easiness of implementation, and excellent efficiency in terms of both computational storage and cost.

In many applications, graph data may be collected from heterogeneous domains (data sources or layers) [15]. For example, social reaction networks between any two persons could be constructed on different platforms, such as Facebook, Google plus, and Instagram. In the biological analysis, The Cancer Genome Atlas (TCGA) contains bioinformatics, genome, transcriptome, and epigenome information for many cancer diseases. It is clear that we can integrate data from heterogeneous domains to identify groups of objects in a reliable manner, for example, a group of patients from a specific set of diseases or a group of users from a specific set of social platforms. The motivation is that different domains contain partial information and the integration of domains

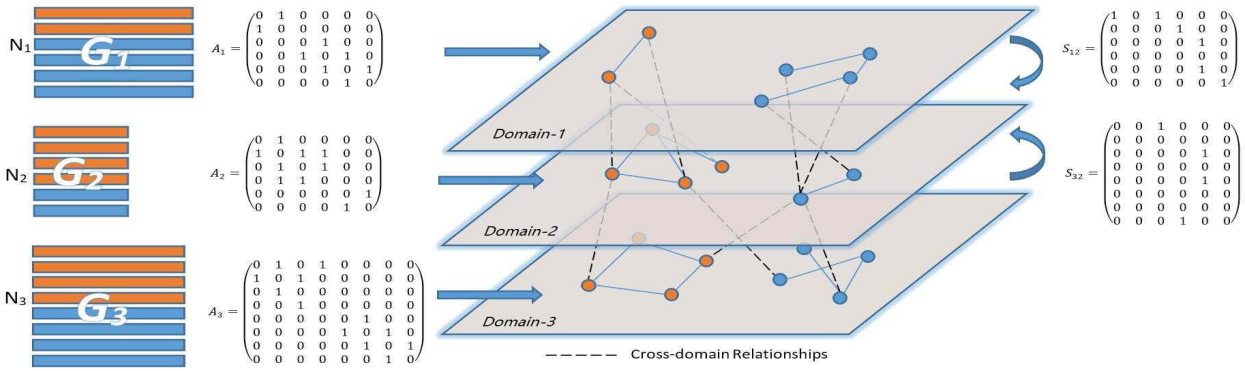

Fig. 1. Multiple domains: different domains contain different sets of instances, while an instance in a domain can be linked with several instances in the other domain through the cross-domain relationships.

enables us to combine different pieces of information together so that the reliability of prediction can be enhanced [35]. For example, the combination of mRNA expression, DNA methylation, and miRNA expression data substantially outperforms single type of data analysis. It has been shown that the accurate identification of cancer subtypes can be achieved [35] by integrating the multiple data sources. In [32], it has been demonstrated that the integration of diverse sources of relevant information from genomic and molecular to cellular and tissue contexts can significantly improve the accuracy of protein function prediction. In [34], a method for integrating multiple graphs for the video annotation problem has been proposed and studied for multimedia content analysis. The key assumption of the above-mentioned applications is that the same set of instances appears in each graph or domain. Each instance just has its different representations or different views in different graphs. In these applications, the objective is to find the most consensus group structure across different domains [22]. On the other hand, we should mention that in some works such as [37] and [38], the multigraph classification may refer to the classification of data with graph-structured features. Due to the difference, we will not discuss them in this paper.

In general, different domains can contain different sets of instances. In addition, an instance in a domain can be linked with several instances in the other domain, as shown in Fig. 1. For example, in the TCGA database, there are several sources or different types of data measured from one sample or object: DNA sequence, miRNA sequence, protein expression, mRNA sequence, DNA methylation, copy number, and so on. A traditional method is to preprocess all data sets by choosing a common set of samples or instances for data analysis. Clearly, some useful information may be removed in this preprocessing stage. The other issue for multidomain learning is that there may be strong noises or irrelevant data in the domains. It would be very important and useful for enhancing learning performance by removing noisy and irrelevant data when we integrate multiple domains together [1], [20].

In this paper, we propose a novel semisupervised learning method to tackle the above-mentioned issues, i.e., multidomain classification with domain selection (MCS). In other words, we develop a model to manage multiple domains, where different domains can have different sets of instances, and one instance in a domain can be linked with several instances in a different domain. In particular, with respect to the target/focused domain, MCS can select a suitable weight for each domain so that noisy and irrelevant data can be removed in the learning process. Specifically, the proposed method involves the optimization of two sets of variables: instance label vector and domain weights with the objective function consisting of the four terms: the label partition of instances based on the Laplacian matrix, the transfer of information from different domains based on domain weights, the given label information, and the regularization of domain weights. Then, MCS is theoretically decomposed into two subproblems, i.e., one domain-weighting subproblem for domain weights and one instance-prediction subproblem for instance label vector, which can be analytically solved, respectively. Thus, the instance label vector can be obtained by solving a linear subproblem, and the domain weights can be determined by aggregating the contribution of each domain in the instance label estimation subproblem, which can be efficiently solved iteratively. It is interesting to note that some domain weights are even set to be zero according to their degrees of contribution in the transfer of domain knowledge.

The remainder of this paper is structured as follows. In Section II, we provide a brief review of the related works on single-domain and multidomain learning. In Section III, we present the proposed model and algorithm for MCS. In Section IV, experimental results for synthetic and real-world data sets are given to demonstrate the superior performance of the proposed method to the other conventional methods. Finally, some concluding remarks are given in Section V.

II. Related Work for Single-/Multiple-Domain Learning Methods

A. Single Domain

In this section, we briefly review some semisupervised approaches for single-domain learning. Basu *et al.* [2] proposed a hidden Markov random fields-based model for semi-

supervised clustering, which incorporates supervision in the form of pairwise constraints. In [3], a variation of the standard K-means clustering algorithm was also used pairwise constraints for constraining the clustering and learning distance metrics. This approach can be regarded as a combination of constraint-based and metric-based methods. Recently, Yi *et al.* casted the dynamic semisupervised clustering process into a search problem over a feasible convex hull generated by multiple ensemble partitions. Such an approach can update the clustering results given newly received pairwise constraints.

In general, from an input data set with N instances $\{x_1, \ldots, x_l, x_{l+1}, \ldots, x_N\}$, we construct an undirected, connected, and weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V}$ and $\mathcal{E}$ represent the sets of vertices and edges, respectively, and $\mathbf{A} \in \mathbb{R}^{N\times N}$ denotes the weighted adjacency matrix. The first l instances $\{x_1, \ldots, x_l\}$ are labeled by $\{y_1, \ldots, y_l\}$, where $y_i \in \{1, -1\}$. Our goal is to predict the labels of the remaining unlabeled instances $\{y_{l+1}, \ldots, y_N\}$. The single-domain semisupervised learning [39] estimates the label of each instance based on the smoothness or consistency assumption of labels on the graph, i.e., the label of each node tends to be the same as that of its neighbors in the graph. The algorithm generates an N-dimensional real-valued vector $\mathbf{f} = (f_1, \ldots, f_l, f_{l+1}, \ldots, f_N)$, which are the label predictions. It is assumed that f_i should be close to the given label y_i in the labeled nodes, while f_i should also be close to f_j when there is an edge linking x_i and x_j or $\mathbf{A}(i, j)$ is large. The corresponding optimization problem is to minimize the following regularized quadratic function:

$$\min_{\{f_i\}_{i=1}^N} \sum_{i,j=1}^{N} \mathbf{A}(i, j)(f_i - f_j)^2 + \lambda \sum_{i=1}^{N} (f_i - y_i)^2 \tag{1}$$

where λ is a positive number and $y_{l+1} = \ldots = y_N = 0$ for the unlabeled nodes. The first term refers to the loss function penalty for the smoothness of $\{f_i\}_{i=1}^N$ and the second term refers to a penalty for the inconsistency with given labels. In a matrix form, (1) can be rewritten as follows:

$$\min_{\mathbf{f}} \mathbf{f}^T \mathbf{L}\mathbf{f} + \lambda ||\mathbf{f} - \mathbf{y}||_2^2 \tag{2}$$

where $\mathbf{L}$ is the graph Laplacian matrix [21] defined by $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Here, $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}(i, i) = \sum_{j=1}^{N} \mathbf{A}(i, j)$. In a multiclass problem, we replace $\mathbf{f}$ by $\mathbf{F} \in \mathbb{R}^{N\times K}$ if there are K possible classes, and minimize the following objective function:

$$\min_{\mathbf{F}} \operatorname{tr}(\mathbf{F}^T \mathbf{L}\mathbf{F}) + \lambda ||\mathbf{F} - \mathbf{Y}||_2^2 \tag{3}$$

where $\operatorname{tr}(\cdot)$ represents the trace of a matrix and $\mathbf{Y} \in \mathbb{R}^{N\times K}$ is defined by

$$\mathbf{Y}(i, k) = \begin{cases} 1 & \text{instance } i \text{ belongs to the } k \text{ class} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

In practice, $\arg\max_j \mathbf{F}(i, j)$ can be used as the label assignment for instance i. The closed form or analytical solution for single-domain semisupervised learning (3) or (2) is given by

$$\mathbf{F} = \lambda(\lambda\mathbf{I} + \mathbf{L})^{-1}\mathbf{Y} \tag{5}$$

where $\mathbf{I}$ is the identity matrix.

B. Multiple Domains

Existing multidomain learning methods assume that the information collected in different domains is for the same set of instances, thus called multiview learning [10]. Blum and Mitchell [4] proposed to use the idea of cotraining for multiview learning. The algorithm is to maximize the mutual agreement on two distinct views of unlabeled data. Muslea *et al.* [24] further combined active learning in the cotraining process and proposed robust semisupervised learning algorithms. Kumar and Daume [18] applied the idea of cotraining to multiview spectral clustering problems. Christoudias *et al.* [9] presented a multiview learning approach where a conditional entropy criterion is used to detect view disagreement. Samples with view disagreement are filtered before standard multiview learning methods are applied.

Another interesting approach is to combine different kernels corresponding to different views, thus improving the learning performance. Tsuda *et al.* [33] proposed a semisupervised framework to protein classification problem. By adapting weighted combination of kernels, this approach discards noisy or irrelevant data in the learning process. Cai *et al.* [8] seeked an optimal linear combination of different networks which approximates a ground-truth community structure for multiplex networks. Zhu and Li [40] also took a similar approach to combine different networks that share a similar structure. Wang *et al.* [34] proposed a multigraph-based semisupervised learning method to a video annotation problem. The method assigns larger weights to relevant graphs and smaller weights to irrelevant graphs. Karasuyama and Mamitsuka [19] proposed a sparse multiple graph integration (SMGI) method to control the sparsity of the weights for the combination of multiple graphs. These two methods have shown an improvement in a classification performance. Wang *et al.* [36] imposed a low-rank constraint to each view with a mutual structural consensus constraint within the multiview spectral clustering framework. On the other hand, subspace learning is employed to obtain a common latent subspace shared by multiple views. Diethe *et al.* [13] generalized Fisher's discriminant analysis to explore the latent subspace spanned by multiview data. Chen *et al.* [11] developed a statistical framework that learns a predictive subspace shared by multiple views.

All the above-mentioned methods assume that different domains contain the same set of instances. In addition, one instance in a domain is not considered to be linked with other instances in a different domain, i.e., no cross linking of instances in different domains. Recently, more and more attention is paid to solve this issue. Following an unsupervised transfer learning approach [27], Cheng *et al.* [12] studied the problem of unsupervised multiple heterogeneous graphs learning where an instance can be linked to other instances in several domains. According to the clustering consensus in multiple graphs, coregularized graph clustering methods are designed to enhance a graph clustering performance. With a similar strategy, Ni *et al.* [26] proposed a framework that clusters multiple domain-specific networks sharing underlying clustering structures based on the domain similarity. They

model domain similarity as a main network with each node being a domain-specific network and formulate a two-phase regularized optimization problem. Other approaches took more advantage of the cross-domain relation. Sun *et al.* [29] studied the problem of classification in heterogeneous information networks. A probabilistic approach was proposed to learn the importance of a different metapath (cross-domain relation) as well as output the classification results that are consistent with the user guidance. Liu *et al.* [23] incorporated prior knowledge on cross-domain relation networks into multinetwork clustering by leveraging the duality between single network clustering and inferring cross-network cluster alignment.

III. Proposed Method: Multidomain Classification With Domain Selection

We develop a new semisupervised learning method for multiple domains, called MCS, which can overcome the problems in the existing multidomain learning methods, i.e., we can further consider the following situations: 1) different domains can contain different sets of instances; 2) an instance in a domain can be linked with several instances in the other domain; and 3) in particular, with respect to the focused/target domain, MCS is automatically able to distinguish its irrelevant and relevant domains for accurate integration of multiple data sources.

Similarly, defined as (1)–(3), suppose there are M domains represented as a graph $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E}_m, \mathbf{A}_m)$ with $m = 1, \ldots, M$. Here, $\mathcal{V}_m$ contains a set of instances $\{x_{m_1}, x_{m_2}, \ldots, x_{m_{N_m}}\}$ and $\mathcal{E}_m$ refers to the links among the instances. $\mathcal{G}_m$ can be governed by the affinity or association matrix $\mathbf{A}_m \in \mathbb{R}^{N_m \times N_m}$ between instances. In addition, we denote cross-domain relations between the mth domain and the m'th domain ($m \neq m'$) by an $N_m \times N_{m'}$ matrix: $\mathbf{S}_{m,m'}$. The (m_i, m'_j)th entry or element of $\mathbf{S}_{m,m'}$ refers to the connection between the m_ith instance in the mth domain and the m'_ith instance in the m'th domain. Here, we assume that this connection is also reciprocal, i.e., $\mathbf{S}^T_{m,m'} = \mathbf{S}_{m',m}$ and T denotes the transpose of a matrix. The learning problem is that given the mth domain and its labeled instances (i.e., the mth domain is the focused/target domain), we would like to determine the other unlabeled instances of the mth domain by using the domain information $\{\mathbf{A}_{m'}\}_{m'=1}^{M}$ and cross-domain relations $\{\mathbf{S}_{m,m'}\}_{m'=1,m'\neq m}^{M}$. Here, we consider the mth domain to be our target domain, and certainly, we can also choose other domain as the target domain depending on the problem.

To incorporate the cross-domain relations, we introduce a loss function to regularize the cross-domain structure. This loss function is designed to penalize classification assignment inconsistency with the given cross-domain relationships. For an instance label vector $\mathbf{f}$ in the mth domain, $\mathbf{S}_{m',m}\mathbf{f}$ represents the label estimation of the corresponding instances in the m'th domain. $y = (y_1, \ldots, y_{N_m})$, which y_i is set to be zero for any unlabeled instance-i in the mth domain. When the two instances in the m'th domain are similar, their transferred label information in $\mathbf{S}_{m',m}\mathbf{f}$ should be close, i.e., $\sum_{i,j} \mathbf{A}_{m'}(i,j)([\mathbf{S}_{m',m}\mathbf{f}]_i - [\mathbf{S}_{m',m}\mathbf{f}]_j)^2$ should be small or $\mathbf{f}^T\mathbf{L}_{m',m}\mathbf{f}$ should be small. Here,

$$\mathbf{L}_{m',m} = \frac{\mathbf{D}_{\mathbf{S}_{m,m'}\mathbf{A}_{m'}\mathbf{S}_{m',m}} - \mathbf{S}_{m,m'}\mathbf{A}_{m'}\mathbf{S}_{m',m}}{||\mathbf{D}_{\mathbf{S}_{m,m'}\mathbf{A}_{m'}\mathbf{S}_{m',m}} - \mathbf{S}_{m,m'}\mathbf{A}_{m'}\mathbf{S}_{m',m}||_F}$$

where $\mathbf{D}_{\mathbf{S}_{m,m'}\mathbf{A}_{m'}\mathbf{S}_{m',m}}$ is the diagonal matrix with diagonal entries being the row sum of $\mathbf{S}_{m,m'}\mathbf{A}_{m'}\mathbf{S}_{m',m}$ and $||*||_F$ is the Frobenious norm of a matrix. In the following discussion, we consider such a scaled Laplacian matrix.

Similar to (2), now we can define the objective function of the proposed model with the mth domain as the focused domain

$$\min_{\mathbf{f},\mathbf{w}} \mathbf{f}^T\mathbf{L}_m\mathbf{f} + \sum_{m'=1,m'\neq m}^{M} w_{m'}\mathbf{f}^T\mathbf{L}_{m',m}\mathbf{f} + \lambda\|\mathbf{f}-\mathbf{y}\|_2^2 + \gamma\,||\mathbf{w}||_2^2 \tag{6}$$

where $\mathbf{f}$ is an N_m-by-1 vector and $\mathbf{w} = [w_1, w_2, \ldots, w_M]$ (without w_m in the vector) is an $(M-1)$-by-1 vector with

$$\sum_{m'=1,m'\neq m}^{M} w_{m'} = 1, \quad w_{m'} \geq 0. \tag{7}$$

Also, λ and γ are two positive numbers to balance the terms $\|\mathbf{f}-\mathbf{y}\|_2^2$ and $||\mathbf{w}||_2^2$, respectively, in the objective function. The first term of the objective function refers to instance label estimation vector partition in the mth domain, which is the focused domain here. The weight $\mathbf{w}$ in the second term is employed to check how the other domain information contributes to the label estimation in the mth domain. The third term of (6) refers to the given labeled and the unlabeled information in the mth domain. The fourth term of (6) is a regularization quantity to control the domain selection in $\mathbf{w}$. When a domain is useful for the domain classification in (6), $w_{m'}$ is large. Otherwise (e.g., when a domain contains much noisy or irrelevant information), the regularization term forces $w_{m'}$ to be small (see Theorem 1). Clearly, by optimizing (6) constrained to (7), MCS has the advantages to consider domain selection and cross-domain information among instances over the existing methods due to the introduction of the domain weight vector $\mathbf{w}$ and cross-domain relations $\mathbf{S}_{m,m'}$. In addition, the first and second terms are also the smoothness or consistency requirement of instances, which implies that the label of each node tends to be the same as that of its neighbors in the all consistent domains.

Here, we should mention that $\mathbf{f}$ can be replaced by $\mathbf{F}$ in (3) to deal with multiclass case and similar results can be achieved in the following discussion.

A. Algorithm for MCS

Next, we show that MCS can be equivalently decomposed into two subproblems, i.e., one domain-weighting subproblem for domain weights and one instance-prediction subproblem for instance label vector, which can be analytically solved, respectively. Thus, to solve the optimization problem MCS in (6) and (7), we efficiently minimize the objective function with respect to $\mathbf{f}$ and $\mathbf{w}$ by alternatively solving the two subproblems.

Similar to the derivation of (5), by fixing $\mathbf{w}$, we can obtain the analytical solution of (6) for $\mathbf{f}$ as follows:

$$\mathbf{f} = \lambda \left(\lambda \mathbf{I} + \mathbf{L}_m + \sum_{m'=1, m' \neq m}^{M} w_{m'} \mathbf{L}_{m',m} \right)^{-1} \mathbf{y}. \tag{8}$$

We note that Laplacian matrices $\mathbf{L}_m$ and $\mathbf{L}_{m',m}$ are positive semidefinite, and $\lambda \mathbf{I} + \mathbf{L}_m + \sum_{m'=1, m' \neq m}^{M} w_{m'} \mathbf{L}_{m',m}$ is positive definite. The linear system solver in (8) can be performed by using LU factorization. Clearly, (8) is the solution of the traditional multidomain classification or semisupervised learning for multiple domains (without the consideration of the domain selection, i.e., with $\mathbf{w}$ fixed).

On the other hand, by fixing $\mathbf{f}$, (6) and (7) for solving $\mathbf{w}$ equivalently reduce to the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{w}} \mathbf{v}^T \mathbf{w} + \gamma \, ||\mathbf{w}||_2^2 \\ &\text{s.t.} \quad \mathbf{w}^T \mathbf{1} = 1, \mathbf{w} \geq 0 \end{aligned} \tag{9}$$

where $\mathbf{v} = [v_1, \ldots, v_M]$ (without v_m in the vector) is an $(M-1)$-by-1 vector where its entry $v_{m'}$ is given by $\mathbf{f}^T \mathbf{L}_{m',m} \mathbf{f}$ and $\mathbf{1}$ is an association vector with all entry being 1. Without loss of generality, we assume that the entries in $\mathbf{v}$ are sorted in the increasing order, i.e., $v_1 \leq v_2 \leq \ldots \leq v_M$.

Theorem 1: The optimal solution of the problem in (9) is analytically given by

$$w_{m'} = \begin{cases} \dfrac{\theta - v_{m'}}{2\gamma} & m' \leq P \\ 0 & m' > P \end{cases} \tag{10}$$

where

$$\theta = \frac{2\gamma + \sum_{i=1}^{P} v_i}{\min\{P, M-1\}} \tag{11}$$

and

$$P = \arg\max_{m'} (\theta - v_{m'} > 0). \tag{12}$$

Proof: Equation (9) is a quadratic optimization problem whose Lagrangian function is given by

$$\mathfrak{L}(\mathbf{w}, \boldsymbol{\beta}, \theta) = \mathbf{v}^T \mathbf{w} + \gamma \, \mathbf{w}^T \mathbf{w} - \boldsymbol{\beta}^T \mathbf{w} - \theta(\mathbf{w}^T \mathbf{1} - 1)$$

where $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_M] \geq 0$ (without β_m) and θ are the Lagrangian multipliers. The optimal solution $\mathbf{w}^*$ satisfies the Karush–Kuhn–Tucker condition [7]

$$\partial_{\mathbf{w}} \mathfrak{L}(\mathbf{w}^*, \boldsymbol{\beta}, \theta) = \mathbf{v} + 2\gamma \, \mathbf{w}^* - \boldsymbol{\beta} - \theta \mathbf{1} = 0 \tag{13}$$

$$\mathbf{w}^* \geq 0, \mathbf{w}^{*T} \mathbf{1} - 1 = 0 \tag{14}$$

$$\boldsymbol{\beta} \geq 0 \tag{15}$$

$$\forall \; 1 \leq m' \leq M \text{ except } m, \; w_{m'}^* \beta_{m'} = 0. \tag{16}$$

From (13), $\mathbf{w}_{m'}$ can be computed as

$$w_{m'} = \frac{\beta_{m'} + \theta - v_{m'}}{2\gamma}. \tag{17}$$

There are three cases for consideration.

1) When $\theta - v_{m'} > 0$, since $\beta_{m'} \geq 0$, we get $w_{m'} > 0$. From condition (16), we see $w_{m'}^* \beta_{m'} = 0$, which indicates $\beta_{m'} = 0$ and, therefore, $w_{m'} = (\theta - v_{m'}/2\gamma)$.
2) When $\theta - v_{m'} = 0$, since $w_{m'}^* \beta_{m'} = 0$ and $w_{m'} = (\beta_{m'}/2\gamma)$, we have $\beta_{m'} = 0$ and $w_{m'} = 0$.
3) When $\theta - v_{m'} < 0$, since $w_{m'} \geq 0$, then we have $\beta_{m'} > 0$. From $w_{m'}^* \beta_{m'} = 0$, we have $w_{m'} = 0$.

Therefore, since $\{v_{m'}\}_{m'=1, m' \neq m}^{M}$ is in the increasing order, using the positive integer $P = \arg\max_{m'}(\theta - v_{m'} > 0)$, the optimality conditions are summarized as follows:

$$w_{m'} = \begin{cases} \dfrac{\theta - v_{m'}}{2\gamma} & m' \leq P \\ 0 & m' > P. \end{cases} \tag{18}$$

By using $\mathbf{w}^T \mathbf{1} = 0$, θ can be computed by

$$\theta = \frac{2\gamma + \sum_{i=1}^{P} v_i}{\min\{P, M-1\}}. \tag{19}$$

The result follows. ■

We can see from (9) that $\mathbf{w}$ tends to have only one nonzero entry with small γ, while all entries in $\mathbf{w}$ tend to be the same with large γ. In between two extreme cases, we obtain sparse solutions in which only some entries of $\mathbf{w}$ are nonzero.

Based on the two analytically solutions (8) for $\mathbf{f}$ and (10) for $\mathbf{w}$, the proposed computational procedure for MCS is summarized in Algorithm 1.

Algorithm 1 Algorithm for MCS

Input: M graphs with $\{\mathbf{A}_{m'}\}_{m'=1}^{M}$, the focused/target domain m and cross-domain relations $\{\mathbf{S}_{m',m}\}_{m'=1, m' \neq m}^{M}$, label vector $\mathbf{y}$, number of class K, and input parameters λ, γ.

Output: label estimation vector $\mathbf{f}$ and domain selection weights $\mathbf{w}$.

1. Initialize $\mathbf{w}$;
2. Optimize $\mathbf{f}$ according to (8);
3. Optimize $\mathbf{w}$ according to (10);
4. Repeat Lines 2 and 3 until convergence

For Line 2, since the graph Laplacian matrix is usually sparse, the computational complexity of a linear system given by (8) can be nearly $O(E)$ [31], where E is the number of nonzero entries in $\mathbf{L}_m + \sum_{m'=1, m' \neq m}^{M} w_{m'} \mathbf{L}_{m',m}$. For Line 3, each iteration of the loop takes $O(M)$ computations to get the maximal number $P = \arg\max_{m'}(\theta - v_{m'} > 0)$, and thus, the computational complexity of the entire process is $O(M^2)$. Therefore, the overall computational complexity of our algorithm is nearly $O(\text{Iter}(E + M^2))$ with Iter being the number of iterations. In practice, since the graph Laplacian matrix is sparse and M and Iter are usually small values ($M \leq 30$), it is expected that our algorithm can be computed effectively and efficiently.

B. Domain Selection Properties

According to Theorem 1, a sparse solution can be obtained for $\mathbf{w}$, since $w_{m'}$ refers to the weight of the relevancy of the m'th domain to the mth domain in the learning process. By using the results in Theorem 1, we know that the proposed model can have a property that can perform domain selection.

On the other hand, by the calculation of $\mathbf{w}$ by (18), $w_i - w_j = (v_i - v_j)/2\gamma$ indicating that w_i and w_j are close once v_i and v_j are close. Moreover, $(v_i - v_j)/2\gamma$ can be further written as

$$\frac{(v_i - v_j)}{2\gamma} = \frac{\mathbf{f}^T\mathbf{L}_{i,m}\mathbf{f} - \mathbf{f}^T\mathbf{L}_{j,m}\mathbf{f}}{2\gamma} = \frac{\langle \mathbf{L}_{i,m} - \mathbf{L}_{j,m}, \mathbf{f}^T\mathbf{f}\rangle_F}{2\gamma}.$$

Thus, when $\mathbf{L}_{i,m}$ and $\mathbf{L}_{j,m}$ are similar to each other in terms of $\langle \mathbf{L}_{i,m} - \mathbf{L}_{j,m}, \mathbf{f}^T\mathbf{f}\rangle_F \approx 0$, then $w_i - w_j \approx 0$. Furthermore, for the difference between w_i and w_j, we can derive the following inequality:

$$\begin{aligned}(w_i - w_j)^2 &= \left(\frac{\langle \mathbf{L}_{i,m} - \mathbf{L}_{j,m}, \mathbf{f}^T\mathbf{f}\rangle_F}{2\gamma}\right)^2 \\ &\le \frac{1}{4\gamma^2}||\mathbf{L}_{i,m} - \mathbf{L}_{j,m}||_F^2||\mathbf{f}^T\mathbf{f}||_F^2 \\ &= \frac{1}{2\gamma^2}(1 - \langle \mathbf{L}_{i,m}, \mathbf{L}_{j,m}\rangle)||\mathbf{f}||_2^4.\end{aligned}$$

For $||\mathbf{f}||_2$, we have

$$||\mathbf{f}||_2 \le \lambda|| \left(\lambda\mathbf{I} + \mathbf{L}_m + \sum_{m'=1, m'\neq m}^{M} w_{m'}\mathbf{L}_{m',m}\right)^{-1} ||_2||y||_2.$$

Since $\lambda\mathbf{I} + \mathbf{L}_m + \sum_{m'=1,m'\neq m}^{M} w_{m'}\mathbf{L}_{m',m}$ is positive definite and $||\mathbf{L}||_2^2$ is equal to the maximum eigenvalue of $\mathbf{L}$, $||(\lambda\mathbf{I} + \mathbf{L}_m + \sum_{m'=1,m'\neq m}^{M} w_{m'}\mathbf{L}_{m',m})^{-1}||_2^2 \le 1/\lambda$. Thus, we have $||\mathbf{f}||_2 \le \sqrt{\lambda}||y||_2$. Then, we can get

$$|w_i - w_j| \le \frac{\lambda||y||_2^2}{\sqrt{2}\gamma}\sqrt{1 - \langle \mathbf{L}_{i,m}, \mathbf{L}_{j,m}\rangle_F}. \tag{20}$$

This indicates that two weights w_i and w_j for which graph Laplacian matrices are highly correlated in terms of $\langle \mathbf{L}_{i,m}, \mathbf{L}_{j,m}\rangle_F$ have similar values. We can also see that $|w_i - w_j|$ will tend to 0 as γ increases by the inequality (20). These results show that our approach provides grouping information for a number of different domains.

IV. Experimental Results

In this section, we mainly compare our algorithm with six methods mentioned in Section II, which all have the similar problem setting as that of our approach. The description of each method will be discussed in the following with further performance discussion as follows.

1) *Single-Domain Semisupervised Classification (SSC) [39]:* As an illustration to examine whether cross-domain relationship can help to enhance the accuracy of classification result, SSC given by (3) will be compared as a baseline single-domain semisupervised learning method.
2) *Optimal Multigraph (OMG) Semisupervised Learning [34]:* This method aims to minimize the following formulation:

$$\begin{aligned}\min_{\mathbf{f},\alpha} \ & \sum_{m=1}^{M} \alpha_m^r\left(\mathbf{f}^T\mathbf{L}_m\mathbf{f} + \lambda||\mathbf{f} - \mathbf{y}||_2^2\right) \\ \text{s.t.} \ & \alpha\mathbf{1} = 1\end{aligned} \tag{21}$$

where $r \ge 1$. The optimization is done by updating $\mathbf{f}$ and α alteratively. Here, r is chosen manually. When r tends to be infinity, $\alpha = \mathbf{1}/\mathbf{M}$ will be an optimal value. On the other hand, $\alpha = \mathbf{e}_{k_{min}}$ if we decrease r to 1. However, the weights cannot be exactly 0 except for the case $r = 1$. This means that OMG always gives nonzero weights to even noisy graphs, which will cause performance deterioration. In practice, α_m is also very sensitive to the setting of r, which causes the performance highly unstable and easily affected by noise. The computational time complexity is $O(\text{Iter} \times E)$ with E being the number of nonzero entries in $\sum_m^M \alpha_m^r\mathbf{L}_m$.
3) *SMGI [19]:* The formulation is given by the following optimization problem:

$$\begin{aligned}\min_{\mathbf{f},\mu} \ & \sum_{m=1}^{M} \mu_m\mathbf{f}^T\mathbf{L}_m\mathbf{f} + \lambda_1||\mathbf{f} - \mathbf{y}||_2^2 + \lambda_2||\mu||_2^2 \\ \text{s.t.} \ & \mu\mathbf{1} = 1, \mu \ge 0\end{aligned} \tag{22}$$

where λ_1 and λ_2 are positive constants and $\mu = \{\mu_1, \ldots, \mu_M\}$. The first term of the objective function penalizes the smoothness of the score $\mathbf{f}$ on all M graphs (represent different views on the same set of instances). The optimization is also done by updating $\mathbf{f}$ and μ alternatively. The time complexity is $O(\text{Iter}(E + M^2))$ with E being the number of nonzero entries in $\sum_m^M \mu_m\mathbf{L}_m$. Both OMG and SMGI have sparse weighting coefficients, which automatically select important graphs and eliminates irrelevant graphs. However, instead of general multidomain learning, both methods are designed for only multiview learning, which aim to find a global classification results for all view of data. Therefore, "focused/target domain" is not necessarily involved in these methods and the classification will tend to smooth the similar graphs occupied in number instead of graphs we are interested in. However, we still put them into comparison as multiview baseline methods to illustrate the important role of domain selection in improving classification accuracy.
4) *Coregularized Multidomain Graph Clustering (CGC) With Focused Domain [12]:* The CGC method was originally designed to handle unsurprised multidomain learning problem with sparse weighting properties. It focuses on only one interested domain each time to improve the clustering accuracy. Therefore, it is worthy to make the comparison with our method. The formulation is defined as follows:

$$\begin{aligned}\min_{\mathbf{H}_m,\mu} \ & ||\mathbf{A}_m - \mathbf{H}_m\mathbf{H}_m^T||_F^2 + \lambda_1||\mathbf{H}_m - \mathbf{y}||_2^2 + \lambda_2||\mu||_2^2 \\ & + \sum_{m'=1,m'\neq m}^{M} \mu_{m'}||\mathbf{S}_{m,m'}\mathbf{H}_{m'}(\mathbf{S}_{m,m'}\mathbf{H}_{m'})^T - \mathbf{H}_m\mathbf{H}_m^T||_F^2 \\ \text{s.t.} \ & \mathbf{H}_m \ge 0, \mu \ge 0, \mu\mathbf{1} = 1\end{aligned} \tag{23}$$

where m is the focused domain. For $m' = 1 \ldots M$, $\mathbf{H}_{m'}$ is calculate from single-domain clustering by nonnegative matrix factorization (NMF) approach, while the

whole optimization is also reached by the NMF methods. The time complexity of CGC is $O(\text{Iter}(N_{\max}^3 + N_{\max}^2 K))$, where $N_{\max} = \max_m\{N_m\}$.

5) *Multinetwork Clustering via Cross-Domain Cluster Alignment (MCA) [23]:* Based on the duality between single network clustering and inferring cross-network cluster alignment, MCA incorporated prior knowledge on cross-domain relationships $\mathbf{S}_{m',m}$ into multinetwork clustering. The optimization problem is as follows:

$$\begin{aligned}\min_{\mathbf{H}_m\geq 0,\mathbf{W}_{m',m}\geq 0} &\sum_{m'\neq m} ||\mathbf{S}_{m',m} - \mathbf{H}_{m'}\mathbf{W}_{m',m}\mathbf{H}_m||_F^2 \\ &+ \sum_m \mu_m \text{Tr}(\mathbf{H}_m^T \mathbf{L}_m \mathbf{H}_m) \\ &+ \sum_{m'\neq m} \lambda_{m'm}||\mathbf{W}_{m',m}||_1 \\ \text{s.t.}\quad &\forall m, \mathbf{H}_m^T\mathbf{H}_m = \mathbf{I} \end{aligned} \tag{24}$$

where $\mathbf{W}_{m',m}$ is used to represent the cross-domain cluster alignment matrix between domain m' and domain m. The time complexity for MCA is $O(\text{Iter}(N_{\max}^2 K + MN_{\max}K^2))$. Although this method is not originally designed for semisupervised learning, it is worthy to make the comparison as a multidomain baseline.

6) Meta-Path Selection With User-Guided Network Clustering (PSC) [29]: PSC is proposed a probabilistic approach to learn the weight for metapath consistent with the user guidance, and generates clusters under the learned weights of metapaths, which optimizes the following function iteratively:

$$\begin{aligned}\max_{\mu,\theta_{ik},\beta_{kj,m'}} \sum_i \Bigg(\sum_{m'} \Bigg(&\sum_j \mu_{m'}\mathbf{S}_{m',m}(i,j)\log \sum_k \theta_{ik}\beta_{kj,m'} \\ &+ \log\Gamma(\mu_{m'}n_{im'} + N_{m'}) \\ &- \sum_j \log\Gamma(\mu_{m'}\mathbf{S}_{m',m}(i,j)+1)\Bigg) \\ &+ \sum_k \mathbf{1}_{\{i\in\mathcal{L}_k\}}\lambda\log\theta_{ik}\Bigg) \end{aligned} \tag{25}$$

where θ_{ik} and $\beta_{kj,m'}$ refer to the possibility of instance x_{m_i} and $x_{m'_j}$ belonging to the kth cluster, and $\mathcal{L}_k$ denotes the labeled samples' set for cluster k and $n_{im'} = \sum_j \mathbf{S}_{m',m}(i,j)$. The alternative optimization of PSC contains an inner loop of an EM-algorithm and another inner loop of a gradient descent algorithm. The total time complexity is $O(\text{Iter}(\text{Iter}_1(K\sum_{m'}|E_{m'}|) + \text{Iter}_2\sum_{m'}|E_{m'}|))$, where Iter, Iter_1 and Iter_2, is the number of iterations for outer loop, EM inner loop, and gradient descent inner loop, and $|E_{m'}|$ is the number of nonzero entries in $\mathbf{S}_{m',m}$.

In the following experiments, the parameters of each algorithm are tuned for optimal performance of all methods, by either using a fivefold cross validation or following the parameter strategy in the original papers.

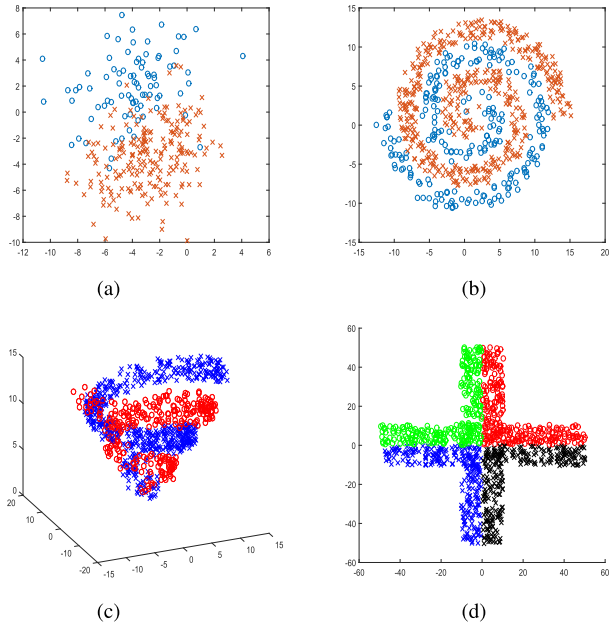

Fig. 2. Four synthetic data sets. (a) Data set A: $N = 300$. (b) Data set B: $N = 500$. (c) Data set C: $N = 900$. (d) Data set D: $N = 1100$. Data sets A, B, and C are for two-class cases where Data sets B and C have 1-D and 2-D manifold structures embedded in the 3-D space, respectively. Data set D is for a four-class case with 1-D manifold.

A. Evaluation Metrics

In the following sections, we consider two metrics to evaluate the classification result. The first one is Rand index (RI) to measure the similarity between the classification results with the ground truth, which is defined as follows:

$$\text{RI} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

where "TP" refers to true positive case that assigns vertices of the same group into the same cluster, "TN" refers to true negative case that assigns vertices of different clusters into different clusters, "FP" refers to false positive case that assigns vertices of different clusters into the same cluster, and "FN" refers to false negative case that assigns vertices of the same group into different clusters.

The second one is Accuracy (Acc), which is defined as the degree of right classification of a model over all examples

$$\text{Acc} = \frac{\sum_i^N \chi(f(i) = y(i))}{N}$$

where $\chi(\cdot)$ is a function having the value 1 when the predicted class label $f(i)$ equals to the true class label $y(i)$ and, otherwise, having the value 0.

B. Examples of Multiview Learning

1) Synthetic Graph Generation: In this section, we test our algorithm with the four existing methods listed earlier on four different data sets varying from different sizes and structures, shown in Fig. 2, including Data sets B and C having 1-D or 2-D manifold structure embedded in the 3-D space and Data set D for a multiclass case.

TABLE I
BASIC NETWORK STATISTICS FOR FOUR SYNTHETIC DATA SETS (EACH DATA SET CONTAINS 30 NETWORKS)

	Dataset A	Dataset B	Dataset C	Dataset D
AWD	[0.2718, 0.4009]	[0.1356, 0.2103]	[0.0921, 0.1152]	[0.7775, 0.1126]
MWD	[15.62, 19.43]	[11.33, 15.54]	[12.32, 17.58]	[12.25, 18.36]
ACC	[0.3484, 0.7913]	[0.2292, 0.7698]	[0.2238, 0.7380]	[0.3215, 0.7307]

We generated 10 relevant graphs in which only a small part of nodes have edges to their neighbors using the k nearest neighbor graph (k-NN graph) as follows:

$$\mathbf{A}_{i,j} = \begin{cases} \exp\left(-\dfrac{d(x_i, x_j)}{2\sigma^2}\right), & i \in \mathfrak{N}_j \text{ or } j \in \mathfrak{N}_i \\ 0 & \text{otherwise.} \end{cases}$$

where $\mathfrak{N}_j$ indicates the index set of x_i values' k nearest neighbor. $d(x_i, x_j)$ indicates a distance metric function and σ is a positive constant parameter.

First, we randomly selected 10 center nodes from the entire data set, and picked up k nearest neighbor data points for each of those center nodes as small subsets of data points. After that, k-NN graphs are created using each small subset of data points. Here, k is randomly chosen from $\{20, 21, ..., 30\}$. In this case, one single graph only provides partial information of graph structures, and these 10 different graphs should be integrated well to estimate the original graph and achieve a good performance. Here, σ is set as the mean of all data point pair distances.

For those irrelevant graphs, we also created two types of data to evaluate graph selection ability of each method. The first type is by reordering the index order of $\{x_i\}_{i=1}^n$. The second type is constructed by randomly generating the element of $\mathbf{A}_{i,j}$ from uniform distribution in $[0, 1]$. For each type, graphs are generated 10 times. Thus, we have total $M = 30$ graphs, including 10 relevant graphs and 20 irrelevant graphs. Without loss of generality, let us set the first 10 graphs to be relevant and the rest 20 graphs to be irrelevant. In this case, one single graph cannot provide complete information of graph structures, and integrating multiple graphs properly is the key to achieve a good performance. Here, all the networks are over the same set of instances, and thus, $\mathbf{S}_{m,m'} = \mathbf{I}$ for all m and m' values.

Basic network statistics, including average weighted degree, maximal weighted degree, and average clustering coefficient, for four synthetic data sets are listed in Table I, where clustering coefficient [25] is the measure of the extent to which one vertex's neighbors are also neighbors of each other. Due to space limitation, we only provide the network statistic range of each data set. Here, the ratio of the number of labeled data points l to the number of entire data points n is set at four different levels, i.e., 10%, 15%, 20%, and 25%.

2) Numerical Comparison: As we can see in Table II, all the six multiview/domain-based methods outperform the single-domain methods SSC, indicating that the cross-domain relationships can indeed play a role in enhancing the accuracy of the clustering result. Among all the methods, MCS can give higher RI/Acc value for almost all the four data sets under different label rates compared with the other baselines (except in a few cases with low label rate, PSC reaches higher value).

According to the previous discussion, the obtained optimal weight for each domain highly depends on the initial weights, which may result in a clustering error, while the involvement of "focused domain" in MCS will reduce the possibility of wrongly assigning weights to irrelevant domains. Here, we should notice that, although "focused domain" is introduced in CGC, the performance of CGC is not satisfactory. One possible reason is that the computed solutions of CGC are sensitive to initial guess of the input parameters due to the computational procedure of NMF in each iteration. In addition, the global optimal solution of CGC is not guaranteed because of its nonconvex optimization procedure. This instability is also the main reason why CGC performs even worse than OMG and SMGI under high label rate. As an illustration of stability, standard derivation (std) for 100 trials is also shown in Table II. For all the methods, std decreases as the label rate increases. The std for MCS is much lower than that of the other methods in nearly all cases. In contrast, the std for CGC is much higher than that of the other methods as discussed earlier, which illustrates the instability of CGC. We also notice that the performance of MCA is poorer than that of the other multiview/domain baselines in most cases. We remark that MCA is originally designed as an unsupervised multidomain method without domain selection in which many irrelevant domains are used and, therefore, its performance can be very bad. In this section, each value is computed on average in 100 trials.

Note that the first 10 domains are relevant to the target domain, while the rest 20 domains are irrelevant. We calculated the average optimal weight of each domain for the "domain-selection" methods (OMG, SMGI, CGC, PSC, and MCS) with different label rates in 100 trials based on Data set A. We emphasize that only MCS correctly assigns positive weight (an average of 0.1111) to each relevant domain and 0 to irrelevant ones in each trial of all experiments. For other method, such as CGC, positive weights (the average value is 0.0786) are given to each relevant domain, while weights (an average of 0.0146) are given to each irrelevant one, resulting in the poor classification performance in Table II. The instability of CGC also results in wrongly assigning weights to each domain even for the data with a high label rate. In our experiments, we should mention that although OMG and SMGI do not handle "focused domain," optimal weight can be more correctly assigned to relevant graphs as the label rate increases, indicating that the label information can help to distinguish relevant or irrelevant domains.

C. Examples of Multidomain Learning

1) Synthetic Graphs' Generation: In this section, we performed the evaluation using the similar setting shown in Fig. 2, where each of the 30 graphs corresponds to data sets with similar structures but different sizes. Taking Data set A as an example, the data sets of different sizes are constructed, as shown in Fig. 3.

TABLE II

CLUSTERING PERFORMANCES (RI/ACCURACY) AND STANDARD DERIVATION (STD) FOR DIFFERENT BASELINES. EACH NUMBER IS CALCULATED IN AVERAGE BY TESTING 100 TRIALS ACCORDING TO EACH EXPERIMENTAL SETTING

Datasets	Label Rate	Metrics	SSC	OMG	SMGI	CGC	MCA	PSC	**MCS**
Dataset A	10%	$\mathrm{RI}_{\mathrm{std}}$	0.6315	$0.7015_{0.0871}$	$0.7183_{0.0651}$	$0.7471_{0.1324}$	$0.7173_{0.0321}$	$\mathbf{0.8321}_{0.0973}$	$0.8271_{0.0546}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.4087	$0.5890_{0.2013}$	$0.6256_{0.1300}$	$0.6793_{0.2260}$	$0.6235_{0.0060}$	$\mathbf{0.8045}_{0.1311}$	$0.7978_{0.0739}$
	15%	$\mathrm{RI}_{\mathrm{std}}$	0.6431	$0.7332_{0.0632}$	$0.7321_{0.0691}$	$0.7594_{0.1106}$	$0.7173_{0.0321}$	$0.8519_{0.0811}$	$\mathbf{0.8662}_{0.0518}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.4651	$0.6546_{0.1169}$	$0.6525_{0.1288}$	$0.6997_{0.1776}$	$0.6235_{0.0660}$	$0.8310_{0.1077}$	$\mathbf{0.8498}_{0.0681}$
	20%	$\mathrm{RI}_{\mathrm{std}}$	0.7015	$0.7641_{0.0516}$	$0.8141_{0.0431}$	$0.7612_{0.1034}$	$0.7173_{0.0321}$	$0.8910_{0.0667}$	$\mathbf{0.9230}_{0.0369}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.5890	$0.7072_{0.0812}$	$0.7800_{0.0592}$	$0.7026_{0.1647}$	$0.6235_{0.0660}$	$0.8819_{0.0852}$	$\mathbf{0.9212}_{0.0432}$
	25%	$\mathrm{RI}_{\mathrm{std}}$	0.7312	$0.9031_{0.0451}$	$0.9102_{0.0367}$	$0.8523_{0.0642}$	$0.7173_{0.0321}$	$0.9344_{0.0522}$	$\mathbf{0.9651}_{0.0228}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.6508	$0.8972_{0.0562}$	$0.9059_{0.0449}$	$0.8315_{0.0853}$	$0.6235_{0.0660}$	$0.9342_{0.0578}$	$\mathbf{0.9646}_{0.0193}$
Dataset B	10%	$\mathrm{RI}_{\mathrm{std}}$	0.6415	$0.7015_{0.0841}$	$0.7183_{0.0610}$	$0.7471_{0.1210}$	$0.7322_{0.0120}$	$\mathbf{0.8541}_{0.0973}$	$0.8371_{0.0521}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.4490	$0.5890_{0.1947}$	$0.6256_{0.1246}$	$0.6793_{0.2066}$	$0.6527_{0.0223}$	$\mathbf{0.8339}_{0.1291}$	$0.8112_{0.0699}$
	15%	$\mathrm{RI}_{\mathrm{std}}$	0.6611	$0.7232_{0.0573}$	$0.7321_{0.0522}$	$0.7594_{0.1011}$	$0.7322_{0.0120}$	$\mathbf{0.9037}_{0.0827}$	$0.8862_{0.0451}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.4782	$0.6354_{0.1132}$	$0.6525_{0.0973}$	$0.6997_{0.1624}$	$0.6527_{0.0223}$	$\mathbf{0.8979}_{0.1030}$	$0.8758_{0.0580}$
	20%	$\mathrm{RI}_{\mathrm{std}}$	0.7215	$0.8041_{0.0488}$	$0.8109_{0.0357}$	$0.7762_{0.0862}$	$0.7322_{0.0120}$	$0.9321_{0.0576}$	$\mathbf{0.9530}_{0.0295}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.6320	$0.7662_{0.0800}$	$0.7756_{0.0721}$	$0.7258_{0.1518}$	$0.6527_{0.0223}$	$0.9316_{0.0927}$	$\mathbf{0.9536}_{0.0436}$
	25%	$\mathrm{RI}_{\mathrm{std}}$	0.7356	$0.9131_{0.0433}$	$0.9102_{0.0310}$	$0.8423_{0.0511}$	$0.7322_{0.0120}$	$0.9769_{0.0421}$	$\mathbf{0.9851}_{0.0198}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.6590	$0.9095_{0.0525}$	$0.9059_{0.0379}$	$0.8181_{0.0683}$	$0.6527_{0.0223}$	$0.9739_{0.0297}$	$\mathbf{0.9793}_{0.0117}$
Dataset C	10%	$\mathrm{RI}_{\mathrm{std}}$	0.6315	$0.7215_{0.0819}$	$0.7383_{0.0598}$	$0.7671_{0.1103}$	$0.7353_{0.0519}$	$\mathbf{0.8536}_{0.0837}$	$0.8400_{0.0501}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.4187	$0.6320_{0.1636}$	$0.6639_{0.1073}$	$0.7119_{0.1714}$	$0.6584_{0.0948}$	$\mathbf{0.8332}_{0.1111}$	$0.8151_{0.0671}$
	15%	$\mathrm{RI}_{\mathrm{std}}$	0.6511	$0.7332_{0.0538}$	$0.7621_{0.0511}$	$0.7794_{0.0978}$	$0.7353_{0.0519}$	$0.8617_{0.0681}$	$\mathbf{0.8962}_{0.0431}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.4444	$0.6546_{0.0996}$	$0.7040_{0.0811}$	$0.7306_{0.1453}$	$0.6584_{0.0948}$	$0.8439_{0.0899}$	$\mathbf{0.8885}_{0.0545}$
	20%	$\mathrm{RI}_{\mathrm{std}}$	0.7115	$0.8441_{0.0411}$	$0.8409_{0.0337}$	$0.7862_{0.0842}$	$0.7353_{0.0519}$	$0.9324_{0.0425}$	$\mathbf{0.9580}_{0.0287}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.6113	$0.8206_{0.0549}$	$0.8163_{0.0451}$	$0.7406_{0.1226}$	$0.6584_{0.0948}$	$0.9320_{0.0475}$	$\mathbf{0.9583}_{0.0264}$
	25%	$\mathrm{RI}_{\mathrm{std}}$	0.7456	$0.9331_{0.0387}$	$0.9302_{0.0309}$	$0.8723_{0.0611}$	$0.7353_{0.0519}$	$0.9475_{0.0351}$	$\mathbf{0.9878}_{0.0151}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.6767	$0.9328_{0.0431}$	$0.9295_{0.0350}$	$0.8578_{0.0799}$	$0.6584_{0.0948}$	$0.9481_{0.0356}$	$\mathbf{0.9809}_{0.0083}$
Dataset D	10%	$\mathrm{RI}_{\mathrm{std}}$	0.6415	$0.7315_{0.0763}$	$0.7283_{0.0683}$	$0.7671_{0.1073}$	$0.7883_{0.0407}$	$0.8211_{0.0655}$	$\mathbf{0.8460}_{0.0512}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.4290	$0.6514_{0.1427}$	$0.6453_{0.1304}$	$0.7119_{0.1667}$	$0.7437_{0.0589}$	$0.7896_{0.0892}$	$\mathbf{0.8231}_{0.0683}$
	15%	$\mathrm{RI}_{\mathrm{std}}$	0.6565	$0.7532_{0.0658}$	$0.7521_{0.0613}$	$0.8100_{0.0973}$	$0.7883_{0.0407}$	$0.8876_{0.0641}$	$\mathbf{0.9012}_{0.0423}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.4630	$0.6896_{0.1088}$	$0.6877_{0.1020}$	$0.7744_{0.1346}$	$0.7437_{0.0589}$	$0.8776_{0.0823}$	$\mathbf{0.8948}_{0.0530}$
	20%	$\mathrm{RI}_{\mathrm{std}}$	0.7015	$0.8641_{0.0411}$	$0.8709_{0.0431}$	$0.8362_{0.0831}$	$0.7883_{0.0407}$	$0.9183_{0.0534}$	$\mathbf{0.9590}_{0.0278}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.5890	$0.8471_{0.0541}$	$0.8560_{0.0564}$	$0.8100_{0.1116}$	$0.7437_{0.0589}$	$0.9157_{0.0636}$	$\mathbf{0.9592}_{0.0253}$
	25%	$\mathrm{RI}_{\mathrm{std}}$	0.7256	$0.9431_{0.0357}$	$0.9402_{0.0339}$	$0.8423_{0.0841}$	$0.7883_{0.0407}$	$0.9489_{0.0339}$	$\mathbf{0.9878}_{0.0161}$
		$\mathrm{Acc}_{\mathrm{std}}$	0.6401	$0.9436_{0.0374}$	$0.9405_{0.0362}$	$0.8182_{0.1124}$	$0.7437_{0.0589}$	$0.9495_{0.0340}$	$\mathbf{0.9809}_{0.0089}$

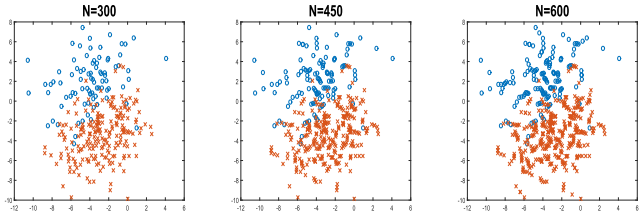

Fig. 3. Data sets with similar structures but different sizes. Left: $N = 300$. Middle: $N = 450$. Right: $N = 600$.

In this way, interrelationship $\mathbf{S}_{i,j}$ between any two graphs $\mathbf{A}_i$ and $\mathbf{A}_j$ can be constructed. For relevant graphs, we generated $\mathbf{S}_{i,j}$ by connecting corresponding class in different graphs. For irrelevant graphs, $\mathbf{S}_{i,j}$ are randomly constructed from uniform distribution in [0, 1]. We still kept the first 10 graphs as relevant graphs, shuffled the index order of $\{x_i\}_{i=1}^n$ for the next 10 graphs and constructed the last 10 graphs by randomly generating the element of $\mathbf{S}_{i,j}$ from uniform distribution in [0, 1]. Basic network statistics are the same as in Table I. The label rate is also set at four different levels: 10%, 15%, 20%, and 25%.

2) Numerical Comparison: Since multiview methods OMG and SMGI cannot be applied to the domains over different sets of instances with the cross-domain relation, only multidomain methods (MCA, PSC, and CGC) are used as the baselines for the comparison in this section. However, the single-domain method SSC can still be performed on the focused domain.

As we can see in Table III, all the four multidomain learning methods outperform the single-domain method SSC, indicating the role of cross-domain relationship in enhancing the accuracy of clustering result. We note that the performance of PSC and MCS is better than that of CGC and MCA. The accuracy of PSC and MCS is enhanced significantly from the supervision learning (as the label rate increases). We see that the performance of MCS is better than that of PSC in all cases. This is because the framework of PSC focuses on "metapath" information (cross-domain relationship $\mathbf{S}_{m',m}$) without domain information ($\mathbf{A}_{m'}$).

The optimal weights for each domain of different methods are also calculated. We emphasize that only MCS correctly assigns positive weights (the average value is 0.1111) to relevant graphs (or domains) and 0 to irrelevant ones in each trial of all experiments. For other methods, taking CGC as illustration, positive weights (the average value is 0.1052) are given to each relevant domain, while weights (the average value is 0.0146) are given to irrelevant domains, which results in the poor performance in Table III. Here, we should mention that the computational cost for MCS is much less than MCA, PSC, and CGC in all tests.

Overall, these results indicate that MCS could eliminate irrelevant graphs or domains efficiently, and MCS achieves

TABLE III

CLUSTERING PERFORMANCES (RI/ACCURACY) AND STANDARD DERIVATION (STD) FOR DIFFERENT MULTIDOMAIN LEARNING METHODS. EACH NUMBER IS CALCULATED ON AVERAGE BY TESTING 100 TRIALS ACCORDING TO EACH EXPERIMENTAL SETTING

Datasets	Label Rate	Accuracy	SSC	CGC	MCA	PSC	**MCS**
Dataset A	10%	$\mathtt{RI}_{\mathtt{std}}$	0.6215	$0.7171_{0.1302}$	$0.7639_{0.0431}$	$0.8243_{0.1031}$	$\mathbf{0.8271}_{0.0583}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.3948	$0.6231_{0.2683}$	$0.7069_{0.0679}$	$0.7940_{0.1400}$	$\mathbf{0.7978}_{0.0789}$
	15%	$\mathtt{RI}_{\mathtt{std}}$	0.6331	$0.7294_{0.1275}$	$0.7639_{0.0431}$	$0.8460_{0.0813}$	$\mathbf{0.8471}_{0.0482}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.4254	$0.6474_{0.2417}$	$0.7069_{0.0679}$	$0.8231_{0.1084}$	$\mathbf{0.8246}_{0.0642}$
	20%	$\mathtt{RI}_{\mathtt{std}}$	0.6515	$0.7412_{0.1132}$	$0.7639_{0.0431}$	$0.8739_{0.0853}$	$\mathbf{0.8872}_{0.0332}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.4458	$0.6690_{0.1997}$	$0.7069_{0.0679}$	$0.8599_{0.1114}$	$\mathbf{0.8771}_{0.0427}$
	25%	$\mathtt{RI}_{\mathtt{std}}$	0.6812	$0.7823_{0.0812}$	$0.7639_{0.0431}$	$0.9059_{0.0832}$	$\mathbf{0.9231}_{0.0214}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.5380	$0.7349_{0.1196}$	$0.7069_{0.0679}$	$0.9006_{0.1030}$	$\mathbf{0.9213}_{0.0250}$
Dataset B	10%	$\mathtt{RI}_{\mathtt{std}}$	0.6315	$0.7571_{0.1230}$	$0.8044_{0.0537}$	$0.8313_{0.1173}$	$\mathbf{0.8471}_{0.0532}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.4287	$0.6960_{0.1997}$	$0.7666_{0.0750}$	$0.8034_{0.1582}$	$\mathbf{0.8246}_{0.0709}$
	15%	$\mathtt{RI}_{\mathtt{std}}$	0.6511	$0.7894_{0.1009}$	$0.8044_{0.0537}$	$0.9057_{0.1001}$	$\mathbf{0.9062}_{0.0451}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.4315	$0.7453_{0.1456}$	$0.7666_{0.0750}$	$0.9004_{0.1240}$	$\mathbf{0.9010}_{0.0558}$
	20%	$\mathtt{RI}_{\mathtt{std}}$	0.7115	$0.8062_{0.0813}$	$0.8044_{0.0537}$	$0.9278_{0.0682}$	$\mathbf{0.9330}_{0.0287}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.6113	$0.7691_{0.1131}$	$0.7666_{0.0750}$	$0.9268_{0.0781}$	$\mathbf{0.9326}_{0.0320}$
	25%	$\mathtt{RI}_{\mathtt{std}}$	0.7456	$0.8423_{0.0789}$	$0.8044_{0.0537}$	$0.9324_{0.0641}$	$\mathbf{0.9451}_{0.0207}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.6767	$0.8182_{0.1055}$	$0.7666_{0.0750}$	$0.9320_{0.0717}$	$\mathbf{0.9457}_{0.0214}$
Dataset C	10%	$\mathtt{RI}_{\mathtt{std}}$	0.6615	$0.7571_{0.1213}$	$0.8146_{0.0389}$	$0.8119_{0.1137}$	$\mathbf{0.8200}_{0.0537}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.4795	$0.6960_{0.1969}$	$0.7807_{0.0534}$	$0.7770_{0.1568}$	$\mathbf{0.7881}_{0.0733}$
	15%	$\mathtt{RI}_{\mathtt{std}}$	0.6911	$0.7694_{0.1091}$	$0.8146_{0.0389}$	$0.8769_{0.1182}$	$\mathbf{0.8862}_{0.0476}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.5639	$0.7155_{0.1680}$	$0.7807_{0.0534}$	$0.8638_{0.1539}$	$\mathbf{0.8758}_{0.0613}$
	20%	$\mathtt{RI}_{\mathtt{std}}$	0.7415	$0.8062_{0.0991}$	$0.8146_{0.0389}$	$0.8932_{0.0876}$	$\mathbf{0.9380}_{0.0267}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.6696	$0.7691_{0.1379}$	$0.7807_{0.0534}$	$0.8847_{0.1115}$	$\mathbf{0.9381}_{0.0289}$
	25%	$\mathtt{RI}_{\mathtt{std}}$	0.7656	$0.8723_{0.0731}$	$0.8146_{0.0389}$	$0.9079_{0.0771}$	$\mathbf{0.9478}_{0.0231}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.7096	$0.8578_{0.0956}$	$0.7807_{0.0534}$	$0.9031_{0.0949}$	$\mathbf{0.9484}_{0.0233}$
Dataset D	10%	$\mathtt{RI}_{\mathtt{std}}$	0.6415	$0.7671_{0.1136}$	$0.8053_{0.0573}$	$0.8034_{0.0976}$	$\mathbf{0.8160}_{0.0602}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.4590	$0.7119_{0.1765}$	$0.7678_{0.0799}$	$0.7652_{0.1365}$	$\mathbf{0.7826}_{0.0825}$
	15%	$\mathtt{RI}_{\mathtt{std}}$	0.6565	$0.8100_{0.1052}$	$0.8053_{0.0573}$	$0.8195_{0.0971}$	$\mathbf{0.8512}_{0.0553}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.4630	$0.7744_{0.1455}$	$0.7678_{0.0799}$	$0.7874_{0.1325}$	$\mathbf{0.8300}_{0.0735}$
	20%	$\mathtt{RI}_{\mathtt{std}}$	0.7015	$0.8362_{0.0921}$	$0.8053_{0.0573}$	$0.8676_{0.0861}$	$\mathbf{0.9000}_{0.0396}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.5890	$0.8100_{0.1237}$	$0.7678_{0.0799}$	$0.8517_{0.1131}$	$\mathbf{0.8933}_{0.0497}$
	25%	$\mathtt{RI}_{\mathtt{std}}$	0.7256	$0.8423_{0.0884}$	$0.8053_{0.0573}$	$0.8831_{0.0743}$	$\mathbf{0.9092}_{0.0331}$
		$\mathtt{Acc}_{\mathtt{std}}$	0.6401	$0.8182_{0.1182}$	$0.7678_{0.0799}$	$0.8718_{0.0960}$	$\mathbf{0.9047}_{0.0406}$

the highest predictive performance among all methods under the setting of synthetic data.

D. Example of 20-Newsgroup Data

1) Data Generation: In this paper, we further evaluated the effectiveness of MCS using 20-Newsgroup data sets (document × term frequency),[1] which is a collection of approximately 20 000 newsgroup documents across 20 different topics covering computer science, talk, religion, and so on. All frequencies are weighted with the tf-idf scheme defined in [14], which is a numerical statistic intending to reflect the importance of a word to a document in a collection.

We used 12 newsgroups of three topics, including Comp, Rec, and Talk listed in Table IV. Each topic corresponds to three underlying clustering structure with four clusters (newsgroups). In this paper, we generated 10 domains for each topic, which contain 200 randomly sampled documents from the four newsgroups (50 documents from each group). As for similarity measurement, cosine similarity for pairwise documents is calculated (top 10% largest entries are kept, while the rest are set to be 0) to construct the affinity matrix $\mathbf{A}_i$ $(1 \leq i \leq 30)$. As a result, we have 30 domains (Comp Domain: $\mathcal{G}_1$–$\mathcal{G}_{10}$, Rec Domain: $\mathcal{G}_{11}$–$\mathcal{G}_{20}$, and Talk Domain: $\mathcal{G}_{21}$–$\mathcal{G}_{30}$) corresponding to three different topics.

[1] http://qwone.com/ jason/20Newsgroups/

TABLE IV

30 DOMAINS FROM 12 NEWSGROUPS IN THREE TOPICS

Comp	Rec	Talk
comp.os.ms-windows.misc	rec.autos	talk.politics.guns
comp.sys.ibm.pc.hardware	rec.motorcycles	talk.politics.mideast
comp.sys.mac.hardware	rec.sport.baseball	talk.politics.misc
comp.graphics	rec.sport.hockey	talk.religion.misc

TABLE V

BASIC NETWORK STATISTICS FOR 20-NEWSGROUP DATA SET

	Comp: Domain	Rec: Domain	Talk: Domain
AWD	[0.2877, 0.3370]	[0.2713, 0.4154]	[0.4258, 0.6259]
MWD	[3.838, 7.756]	[2.582, 6.831]	[3.975, 6.061]
ACC	[0.2934, 0.3470]	[0.3053, 0.4164]	[0.4166, 0.4810]

Basic network statistics for 20-Newsgroup data set are listed in Table V. Due to space limitation, we only provide the network statistic range for Comp Domain, Rec Domain, and Talk Domain.

The interrelation or cross-domain relation $\mathbf{S}_{i,j}$ $(1 \leq i, j \leq 30)$ between any two domains $\mathcal{G}_i$ and $\mathcal{G}_j$ is constructed as follows. For any two domains generated from the same topics, document in one domain is randomly mapped to documents with the same label (e.g., rec.autos) in another domain. For any two domains generated from different topics, the documents are randomly mapped together without considering label. The label rate is still set at four different levels 10%, 15%, 20%, and 25%.

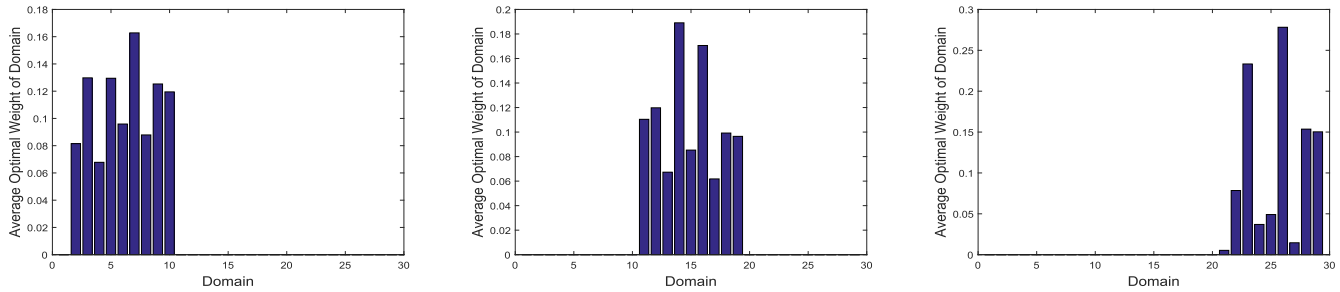

Fig. 4. Optimal weights for different domains. Each entry is mean value over 100 trials. The focused domain is from left to right: $\mathcal{G}_1$, $\mathcal{G}_{11}$, and $\mathcal{G}_{21}$.

TABLE VI

AVERAGE PERFORMANCES FOR FIVE METHODS WITH DIFFERENT FOCUSED DOMAINS AND LABEL RATES

Focused Domain	Label Rate	Metrics	SSC	CGC	MCA	PSC	**MCS**
Comp: $\mathcal{G}_1$	10%	RI	0.6318	0.6433	0. 6970	0.7842	**0.8225**
		Acc	0.4391	0.4758	0.5784	0.7377	**0.7915**
	15%	RI	0.6705	0.7014	0. 6970	0.8244	**0.8912**
		Acc	0.5075	0.5888	0.5784	0.7941	**0.8822**
	20%	RI	0.7245	0.7123	0. 6970	0.8765	**0.9205**
		Acc	0.6380	0.6130	0.5784	0.8633	**0.9183**
	25%	RI	0.7449	0.7433	0. 6970	0.9066	**0.9476**
		Acc	0.6755	0.6727	0.5784	0.9015	**0.9482**
Rec: $\mathcal{G}_{11}$	10%	RI	0.7822	0.7512	0.7714	0.8531	**0.8971**
		Acc	0.7348	0.6862	0.7185	0.8326	**0.8897**
	15%	RI	0.8344	0.7931	0.7714	0.8824	**0.9236**
		Acc	0.8076	0.7506	0.7185	0.8709	**0.9219**
	20%	RI	0.8799	0.8017	0.7714	0.9012	**0.9669**
		Acc	0.8677	0.7628	0.7185	0.8948	**0.9661**
	25%	RI	0.8997	0.8211	0.7714	0.9326	**0.9773**
		Acc	0.8929	0.7896	0.7185	0.9322	**0.9742**
Talk: $\mathcal{G}_{21}$	10%	RI	0.7492	0.7013	0.7337	0.8956	**0.9015**
		Acc	0.6829	0.5885	0.6555	0.8878	**0.8952**
	15%	RI	0.8270	0.7142	0.7337	0.9376	**0.9495**
		Acc	0.7976	0.6170	0.6555	0.9377	**0.9501**
	20%	RI	0.8468	0.7500	0.7337	0.9388	**0.9613**
		Acc	0.8242	0.6842	0.6555	0.9390	**0.9613**
	25%	RI	0.8652	0.7834	0.7337	0.9439	**0.9628**
		Acc	0.8485	0.7365	0.6555	0.9444	**0.9626**

2) Numerical Results: In this section, we also only compared our method with multidomain baselines (MCA, PSC, and CGC) together with the single-domain method SSC.

Table VI shows the average accuracy over 100 trials for each case. As shown in Table VI, MCS achieves a better performance compared with other methods when varying both focused domain and label rate. MCS is more efficient and can benefit significantly from the supervised information as the label rate increases. In Fig. 4, the optimal weight for each domain is also reported. It can be seen that MCS is able to utilize the relevant domains while filtering the irrelevant domains to dramatically improve the accuracy, which validates the effectiveness of our method.

E. Example of Real-World Patient Data

1) Data Generation: In this section, we studied a real-world application: cancer subtype classification problem. We compared two lung cancer subtypes, i.e., lung adenocarcinoma (LUAD) and lung squamous cell carcinoma (LUSC), among multiple data types, including gene expression, microRNA expression, and DNA methylation data from TCGA. We downloaded level-3 data for each data type and platform. The detailed information of samples is described in Table VII. As normal samples of LUAD and LUSC are both from the same tissue lung, we combined them together as normal samples and then filtered out those features with missing value along all samples in the same platform. In this way, we formed five different domains representing different platforms over different sets of samples, as shown in Table VIII. Except $\mathcal{G}_2$ which only contains two subtypes of cancer, all the other domains contain both normal and two subtypes.

As for similarity measurement for pairwise samples, we utilized the tool called similarity network fusion from [35] where Gaussian kernel is used to measure the similarity between each pair of samples $\mathbf{A}_i$ $(1 \leq i \leq 5)$ which is better than Pearson correlation coefficient, as shown in Fig. 5. Basic network statistics for five domains in Patient data set are listed in Table IX.

TABLE VII

TWO LUNG CANCER SUBTYPES, LUAD AND LUSC, AMONG MULTIPLE DATA TYPES, INCLUDING GENE EXPRESSION, MICRORNA EXPRESSION, AND DNA METHYLATION DATA FROM TCGA

Cancer Type	Data Type	Platform	State	Sample Size
LUSC	RNASeq	UNC__IlluminaHiSeq_RNASeq	Normal	37
			Tumor	125
	miRNA	BCGSC__IlluminaGA_miRNASeq	Normal	0
			Tumor	63
		BCGSC__IlluminaHiSeq_miRNASeq	Normal	46
			Tumor	458
	Methylation	JHU_USC__HumanMethylation27	Normal	24
			Tumor	126
		JHU_USC__HumanMethylation450	Normal	32
			Tumor	475
LUAD	RNASeq	UNC__IlluminaHiSeq_RNASeq	Normal	17
			Tumor	224
	miRNA	BCGSC__IlluminaGA_miRNASeq	Normal	0
			Tumor	136
		CGSC__IlluminaHiSeq_miRNASeq	Normal	45
			Tumor	342
	Methylation	JHU_USC__HumanMethylation27	Normal	27
			Tumor	134
		HU_USC__HumanMethylation450	Normal	42
			Tumor	370

TABLE VIII

FIVE DIFFERENT DOMAINS REPRESENTING FIVE DIFFERENT PLATFORMS

Domain	Platform	Sample Type	Size
$\mathcal{G}_1$	UNC_IlluminaHiSeq_RNASeq	Normal/LUAD/LUSC	403
$\mathcal{G}_2$	BCGSC_IlluminaGA_miRNASeq	LUAD/LUSC	199
$\mathcal{G}_3$	BCGSC__IlluminaHiSeq_miRNASeq	Normal/LUAD/LUSC	891
$\mathcal{G}_4$	JHU_USC__HumanMethylation27	Normal/LUAD/LUSC	311
$\mathcal{G}_5$	JHU_USC__HumanMethylation450	Normal/LUAD/LUSC	919

TABLE IX

BASIC NETWORK STATISTICS FOR FIVE DOMAINS IN PATIENT DATA SET

	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_3$	$\mathcal{G}_4$	$\mathcal{G}_5$
AWD	0.0311	0.0409	0.0692	0.0130	0.0013
MWD	0.8238	0.3504	1.6313	0.1493	0.0127
ACC	1.7086	1.3253	1.5803	1.2064	1.3244

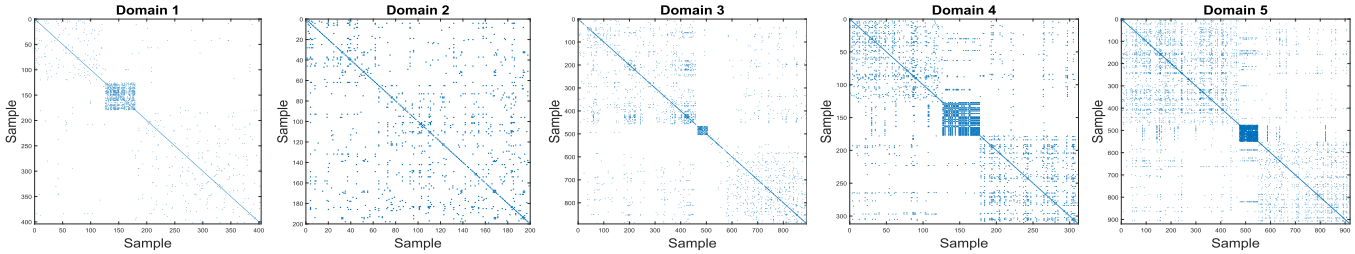

Fig. 5. Similarity matrices for different domains using Gaussian kernel. From left to right: $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{G}_3$, $\mathcal{G}_4$, and $\mathcal{G}_5$.

TABLE X
AVERAGE PERFORMANCES FOR FIVE METHODS WITH DIFFERENT FOCUSED DOMAINS AND LABEL RATES

Focused Domain	Label Rate	Metrics	SSC	CGC	MCA	PSC	**MCS**
$\mathcal{G}_1$	10%	RI	0.6311	0.6521	0.7139	0.6829	**0.7298**
		Acc	0.3671	0.4479	0.6164	0.5426	**0.6482**
	15%	RI	0.6369	0.6421	0.7139	0.7014	**0.7694**
		Acc	0.3909	0.4113	0.6164	0.5888	**0.7155**
	20%	RI	0.6512	0.6721	0.7139	0.7288	**0.7872**
		Acc	0.4447	0.5122	0.6164	0.6463	**0.7421**
	25%	RI	0.6802	0.7115	0.7139	0.7437	**0.7959**
		Acc	0.5352	0.6113	0.6164	0.6734	**0.7546**
$\mathcal{G}_2$	10%	RI	0.5861	0.6023	0.6943	0.6712	**0.6957**
		Acc	0.3540	0.4087	0.5718	0.5095	**0.5753**
	15%	RI	0.6120	0.6312	0.6943	0.6784	**0.7086**
		Acc	0.3493	0.4275	0.5718	0.5303	**0.6050**
	20%	RI	0.6218	0.6431	0.6943	0.7042	**0.7356**
		Acc	0.3662	0.4351	0.5718	0.5952	**0.6590**
	25%	RI	0.6427	0.6631	0.6943	0.7321	**0.7431**
		Acc	0.4136	0.4846	0.5718	0.6525	**0.6724**
$\mathcal{G}_3$	10%	RI	0.5805	0.6134	0.6621	0.6611	**0.6625**
		Acc	0.3281	0.3362	0.4814	0.4782	**0.4827**
	15%	RI	0.5980	0.6312	0.6621	0.6653	**0.6790**
		Acc	0.3348	0.3675	0.4814	0.4916	**0.5319**
	20%	RI	0.6290	0.6432	0.6621	0.6782	**0.7041**
		Acc	0.3581	0.4155	0.4814	0.5297	**0.5950**
	25%	RI	0.6562	0.7018	0.6621	0.7011	**0.7299**
		Acc	0.4620	0.5897	0.4814	0.5881	**0.6484**
$\mathcal{G}_4$	10%	RI	0.6999	0.7327	0.8047	0.7952	**0.8142**
		Acc	0.5853	0.6536	0.7670	0.7536	**0.7802**
	15%	RI	0.7520	0.7542	0.8047	0.8096	**0.8348**
		Acc	0.6876	0.6912	0.7670	0.7738	**0.8081**
	20%	RI	0.7851	0.8012	0.8047	0.8357	**0.8766**
		Acc	0.7390	0.7621	0.7670	0.8094	**0.8634**
	25%	RI	0.8065	0.8123	0.8047	0.8471	**0.8825**
		Acc	0.7695	0.7776	0.7670	0.8246	**0.8710**
$\mathcal{G}_5$	10%	RI	0.6650	0.6821	0.7035	0.6973	**0.7325**
		Acc	0.4906	0.5404	0.5936	0.5791	**0.6533**
	15%	RI	0.6777	0.7012	0.7035	0.7098	**0.7589**
		Acc	0.5283	0.5883	0.5936	0.6076	**0.6989**
	20%	RI	0.6902	0.7086	0.7035	0.7253	**0.7675**
		Acc	0.5616	0.6050	0.5936	0.6395	**0.7125**
	25%	RI	0.7037	0.7103	0.7035	0.7467	**0.7956**
		Acc	0.5940	0.6087	0.5936	0.6786	**0.7542**

In this way, interrelationship or cross-domain relation $\mathbf{S}_{i,j}$ between any two graphs $\mathbf{A}_i$ and $\mathbf{A}_j$ is constructed by linking the same sample in different domains.

2) Numerical Results: In the experiments, we compare the proposed method MCS with the multidomain baselines (MCA, PSC, and CGC) and SSC. The focused domain is chosen from $\mathcal{G}_1$ to $\mathcal{G}_5$, and the label rate in this section is still set as 10%, 15%, 20%, and 25%. Table X shows the average RI for all five methods, taken over 100 trials.

Here, we could see that MCS has a significantly higher RI/Acc rate than those of SSC, CGC, MCA, and PSC, indicating that MCS has nice property of graph selection and benefits from cross-domain information in the real-world application.

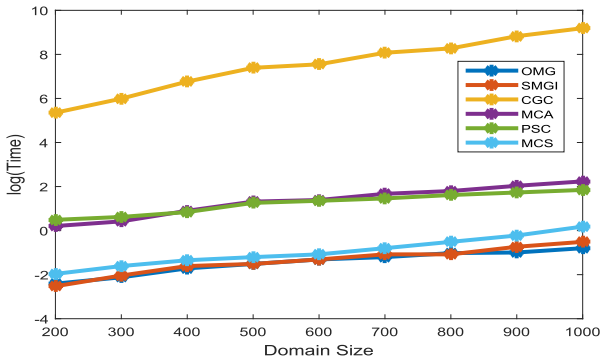

Fig. 6. Computational time comparison among OMG, SMGI, CGC, MCA, PSC, and MCS based on synthetic data sets.

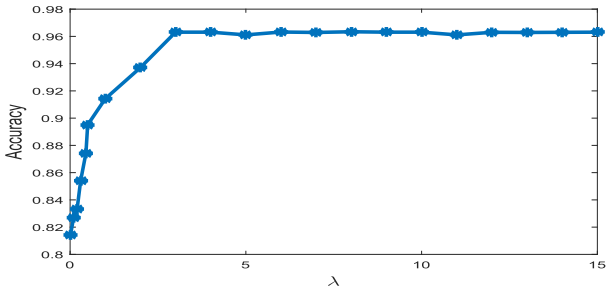

Fig. 7. Sensitivity of λ based on Data set A with label rate $= 25\%$. Each entry is mean value over 100 trials in terms of RI.

F. Scalability and Stability Analysis

In this section, we first analyze the scalability for each method by comparing the computational cost with data size increased. Then, we will further analyze the stability of the proposed MCS method by studying the parameters' sensitivity of λ and γ, which also helps us to understand the impact of supervision and domain selection, respectively.

1) Scalability Analysis: Although the computational complexity is given in the first part of this section. We further compared the computation cost for each method as the size of domain increases in Fig. 6 based on synthetic Data set A. According to Fig. 6, MCS, OMG, and SMGI reach the lowest computational cost, followed by PSC and MCA, while the computational cost of CGC is the highest among all the baselines. This is also the case when applied to other data sets in Sections IV-A–IV-D.

2) Stability Analysis: We first study the performance of MCS with a different setting of λ. According to Fig. 7, the performance of MCS gets better with the increase of λ and tends to be stable after some value (in this case, $\lambda = 3$). Thus, it is easy for user to choose a suitable λ in their implements. We should mention that the role of semisupervision is not

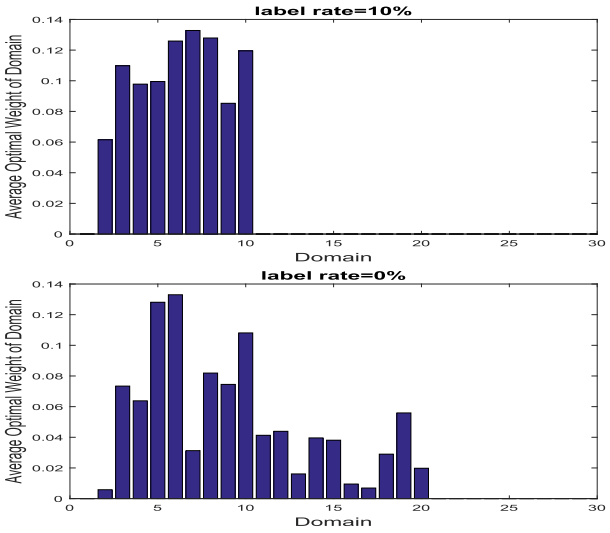

Fig. 8. Weights for different domains. Top: label Rate = 10%. Bottom: label Rate = 0%. Each entry is mean value over 100 trials. The focused domain is the first domain.

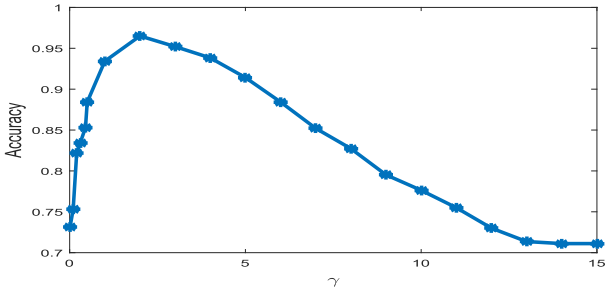

Fig. 9. Sensitivity of γ based on Data set A with label rate = 25%. Each entry is mean value over 100 trials in terms of RI.

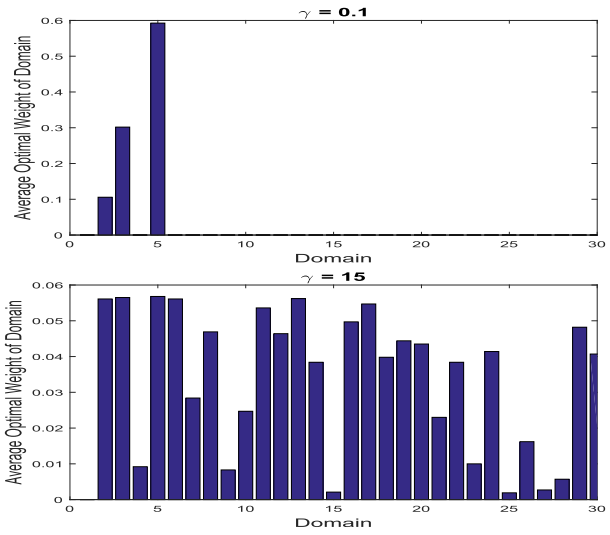

Fig. 10. Weights for different domains. Top: $\gamma = 0.1$. Bottom: $\gamma = 15$. Each entry is mean value over 100 trials.

only in guidance of clusters but also guidance in the domain selection. Here, we show a case when $\lambda = 0$ (label rate = 0%) in Fig. 8. Remark that the first 10 domains are relevant domains, while the rest are irrelevant to the target domain. In this sense, the lack of label information will result in the error weighting, as shown in Fig. 8. This also explains the low accuracy performance of MCS when λ is small.

We further study the influence with a different setting of γ. As discussed in Section III-A, $\mathbf{w}$ have sparse positive entries with small γ, while all entries in $\mathbf{w}$ will tend to the same value with large γ. In between two extremes, we obtain sparse solutions.

In Fig. 9, we evaluate the performance of MCS with γ within the range of $[0, 15]$. It is easy to see that the optimal γ locates within $[1, 3]$, which reflects that the setting of γ should not be either too small or too large. When γ is too small, weights will be assigned to only a few domains, such that no enough relevant domains will be integrated. However, when γ is too large, weights will be averagely assigned to all domains, such that irrelevant domains will be involved. Here, we also show two extreme cases with $\gamma = 0.1$ and 15 for illustration (see Fig. 10).

V. Conclusion

Integrating multiple data sources is an important problem in data mining research. Robust and flexible approaches that can incorporate multiple sources to enhance domain classification performance are highly desirable. In this paper, we proposed a new approach MCS for integrating multiple domains under the semisupervised spectral clustering framework. MCS can consider the cross-domain information and individual instance sets of multiple domains, comparing with the traditional methods. In particular, with the appealing properties of the sparsity of domain weights, by which irrelevant domains can be easily eliminated, MCS method is able to obtain optimal graph partition performance for the focused domain. From the computational viewpoint, we equivalently decompose MCS into two simpler subproblems with two analytical solutions, which can be efficiently solved iteratively. We also present the efficient optimization algorithm in MCS, which allows the clear interpretation of our formulation. Experimental results have demonstrated that MCS has a superior domains' selection ability as well as the highest prediction performance among the state-of-the-art methods for integrating multiple domains.

References

[1] A. Argyriou, M. Herbster, and M. Pontil, "Combining graph laplacians for semi-supervised learning," in *Advances in Neural Information Processing Systems*, vol. 18. Cambridge, MA, USA: MIT Press, 2006, pp. 67–74.

[2] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semi-supervised clustering," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 59–68.

[3] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 11.

[4] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, 1998, pp. 92–100.

[5] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 19–26.

[6] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Jan. 2006.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[8] D. Cai, Z. Shao, X. He, X. Yan, and J. Han, "Community mining from multi-relational networks," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*. Berlin, Germany: Springer, 2005, pp. 445–452.

[9] C. Christoudias, R. Urtasun, and T. Darrell. (2012). "Multi-view learning in the presence of view disagreement." [Online]. Availble: https://arxiv.org/abs/1206.3242
[10] X. Chang, D. Tao, and C. Xu. (2013). "A survey on multi-view learning." [Online]. Available: https://arxiv.org/abs/1304.5634
[11] N. Chen, J. Zhu, and E. P. Xing, "Predictive subspace learning for multi-view data: A large margin approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 361–369.
[12] W. Cheng, Z. Guo, X. Zhang, and W. Wang,"CGC: A flexible and robust approach to integrating co-regularized multi-domain graph for clustering," *Trans. Knowl. Discovery Data*, vol. 10, no. 4, 2015, Art. no. 46.
[13] T. Diethe, D. R. Hardoon, and J. Shawe-Taylor, "Multiview Fisher discriminant analysis," in *Proc. NIPS Workshop Learning Multiple Sources*, Whistler, BC, Canada, 2008.
[14] M. Ebbesson and C. Issal, "Document clustering," M.S. thesis, Dept. Comput. Sci. Eng., Chalmers Univ. Technol., Göteborg, Sweden, 2010.
[15] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han, "Graph-based consensus maximization among multiple supervised and unsupervised models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 585–593.
[16] M. Herbster, M. Pontil, and L. Wainer, "Online learning over graphs," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 305–312.
[17] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 290–297.
[18] A. Kumar and H. Daume, III, "A co-training approach for multi-view spectral clustering," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 393–400.
[19] M. Karasuyama and H. Mamitsuka, "Multiple graph label propagation by sparse integration," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 1999–2012, Dec. 2013.
[20] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, Dec. 2004.
[21] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[22] B. Long, P. S. Yu, and Z. M. Zhang, "A general model for multiple view unsupervised learning," in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 822–833.
[23] R. Liu, W. Cheng, H. Tong, W. Wang, and X. Zhang, "Robust multi-network clustering via joint cross-domain cluster alignment," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2015, pp. 291–300.
[24] I. Muslea, S. Minton, and C. A. Knoblock, "Active learning with multiple views," *J. Artif. Intell. Res.*, vol. 27, no. 1, pp. 203–233, 2006.
[25] M. E. J. Newman, "Random graphs with clustering," *Phys. Rev. Lett.*, vol. 103, p. 058701, Jul. 2009.
[26] J. Ni, H. Tong, W. Fan, and X. Zhang, "Flexible and robust multi-network clustering," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 835–844.
[27] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[28] M. Szummer and T. Jaakkola, "Partially labeled classification with Markov random walks," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2002, pp. 945–952.
[29] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu, "PathSelClus: Integrating meta-path selection with user-guided Object clustering in heterogeneous information networks," *ACM Trans. Knowl. Discovery Data*, vol. 7, no. 3, 2013, Art. no. 11.
[30] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: From transductive to semi-supervised learning," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 824–831.
[31] D. A. Spielman and S.-H. Teng, "Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems," in *Proc. 36th Annu. ACM Symp. Theory Comput.*, 2004, pp. 81–90.
[32] H. Shin, A. M. Lisewski, and O. Lichtarge, "Graph sharpening plus graph integration: A synergy that improves protein functional classification," *Bioinformatics*, vol. 23, no. 23, pp. 3217–3224, 2007.
[33] K. Tsuda, H. J. Shin, and B. Schölkopf, "Fast protein classification with multiple networks," *Bioinformatics*, vol. 21, pp. ii59–ii65, Jan. 2005.
[34] M. Wang, X.-S. Hua, R. Hong, J. Tang, G.-J. Qi, and Y. Song, "Unified video annotation via multigraph learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 733–746, May 2009.
[35] B. Wang *et al.*, "Similarity network fusion for aggregating data types on a genomic scale," *Nature Methods*, vol. 11, pp. 333–337, Jan. 2014.
[36] Y. Wang, W. Zhang, L. Wu, X. Lin, M. Fang, and S. Pan. (2016). "Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering." [Online]. Available: https://arxiv.org/abs/1608.05560
[37] J. Wu, S. Pan, X. Zhu, and Z. Cai, "Boosting for multi-graph classification," *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 416–429, Mar. 2015.
[38] J. Wu, X. Zhu, C. Zhang, and P. S. Yu, "Bag constrained structure pattern mining for multi-graph classification," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 10, pp. 2382–2396, Oct. 2014.
[39] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, vol. 16, no. 16, pp. 321–328.
[40] G. Zhu and K. Li, "A unified model for community detection of multiplex networks," in *Proc. Int. Conf. Web Inf. Syst. Eng.* Thessaloniki, Greece: Springer, 2014.
[41] J. Yi, L. Zhang, T. Yang, W. Liu, and J. Wang, "An efficient semi-supervised clustering algorithm with sequential constraints," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining.*, 2015, pp. 1405–1414.

Chuan Chen received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2012, and the Ph.D. degree from Hong Kong Baptist University, Hong Kong, in 2016.

He is currently an Associate Research Fellow with the School of Data and Computer Science, Sun Yat-sen University. His current research interests include machine learning, numerical linear algebra, and numerical optimization.

Jingxue Xin received the bachelor's degree in mathematics from Wuhan University, Wuhan, China, in 2013. She is currently pursuing the Ph.D. degree with the Institute of Applied Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China.

Her current research interests include operations' research and computational systems' biology.

Yong Wang received the bachelor's degree in mathematics and physics from Inner Mongolia University, Hohhot, China, in 1999, the master's degree in operations research and control theory from the Dalian University of Technology, Dalian, China, in 2002, and the Ph.D. degree in operations research and control theory from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, in 2005.

He is currently a Professor with the Institute of Applied Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences. His current research interests include mathematical modeling and algorithm analysis in bioinformatics.

Luonan Chen received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 1984, and the M.S. and Ph.D. degrees from Tohoku University, Sendai, Japan, in 1988 and 1991, respectively.

Since 1997, he has been an Associate Professor with Osaka Sangyo University, Osaka, Japan, where he became a Full Professor. Since 2010, he has been a Professor and the Executive Director with the Key Laboratory of Systems Biology, Institute of Biochemistry and Cell Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Beijing, China. In recent years, he published over 300 journal papers and two monographs in the area of systems' biology. His current research interests include systems' biology, computational biology, and nonlinear dynamics.

Michael K. Ng received the B.Sc. and M.Phil. degrees from The University of Hong Kong, Hong Kong, in 1990 and 1992, respectively, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 1995.

From 1995 to 1997, he was a Research Fellow with the Computer Sciences Laboratory, Australian National University, Canberra, ACT 2601, Australia. From 1997 to 2005, he was an Assistant/Associate Professor with The University of Hong Kong. He is currently a Chair Professor with the Department of Mathematics, Hong Kong Baptist University, Hong Kong. His current research interests include bioinformatics, image processing, scientific computing, and data mining.

Dr. Ng serves on the editorial boards of international journals. http://www.math.hkbu.edu.hk/∼mng.