Multiple Self-Adaptive Correlation-Based Multiview Multilabel Learning

Changming Zhu, Yimin Yan, Duoqian Miao, Yilin Dong, and Witold Pedrycz, *Life Fellow, IEEE*

***Abstract*—In order to process multiview multilabel, multilabel, and multiview data, current learning algorithms are designed on the basis of data characteristics, correlations, etc. While these algorithms cannot express correlations among different features, instances, labels in within-view, cross-view, and consensus-view representations self-adaptively and relative accurately. To this end, this study takes the classical multiple correlations-based model as the basis and explores some laws of self-adaptive change for those correlations in multiple representations. The proposed algorithm is called multiple self-adaptive correlation-based multiview multilabel learning (MuSC-MVML). Extensive experiments on 38 datasets demonstrate the superiority of MuSC-MVML and some conclusions are addressed. 1) MuSC-MVML outperforms most compared algorithms in statistical in terms of AUC and its performance is also stable; 2) the computational cost of MuSC-MVML is moderate and on most datasets, MuSC-MVML has a relatively fast convergence; and 3) introducing some laws of self-adaptive change for those correlations can improve the ability of MuSC-MVML to process multiview multilabel datasets effectively and express correlations in multiple representations better. Furthermore, this study explains the reason that why we use alternating optimization strategy to optimize the model of MuSC-MVML and provides some suggestions that how to modify the model of MuSC-MVML to process incomplete multiview multilabel datasets with noise.**

***Index Terms*—Consensus-view, cross-view, multiview multilabel, self-adaptive, within-view.**

Received 1 February 2024; revised 26 July 2024 and 15 January 2025; accepted 22 January 2025. Date of publication 11 February 2025; date of current version 18 March 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62276164; in part by the Science and Technology Innovation Action Plan Natural Science Foundation of Shanghai under Grant 22ZR1427000; and in part by the Shanghai Oriental Talent Program-Youth Program. This article was recommended by Associate Editor Q. Shen. *(Corresponding author: Changming Zhu.)*

Changming Zhu, Yimin Yan, and Yilin Dong are with the College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China (e-mail: cmzhu@shmtu.edu.cn; 15751866070@163.com; yldong@shmtu.edu.cn).

Duoqian Miao is with the School of Electronics and Information Engineering, Tongji University, Shanghai 200070, China (e-mail: dqmiao@tongji.edu.cn).

Witold Pedrycz is with the Department of Measurement and Control Systems, Silesian University of Technology, 44-100 Gliwice, Poland, also with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2R3, Canada, and also with the Research Center of Performance and Productivity Analysis, Istinye University, Istanbul 34396, Türkiye (e-mail: wpedrycz@ualberta.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2025.3534231.

Digital Object Identifier 10.1109/TCYB.2025.3534231

I. Introduction

A. Background

MULTIVIEW multilabel, multilabel, and multiview datasets are common in real-world applications. Each instance in these datasets can be tagged by multiple class labels or/and represented by multiple sets of features. For example, in Fig. 1, we give three illustrative examples to show these datasets in some specific real-world applications. In order to tackle with these datasets, scholars have to design algorithms that conform to these data characteristics, including features, labels, and instances themselves (see Table I) [1], [2], [3], [4], [5], [6], [7], [8], [9], [10] and they have achieved effective performances and attracted scholars to improve them.

Compared with the usage of these characteristics, another strategy to solve these datasets is to use of the correlations among instances, features, and labels which are common in different representations. Indeed, information about features, labels, and instances themselves can be demonstrated in multiple forms, including within-view, cross-view, and consensus-view representations. First, within-view representation demonstrates data information expressed in a view. Second, cross-view representation is shared by two different views and its information can be treated as the source for the information of two views. Third, consensus-view representation is shared by all views and its information describes the consensus representation and source of information from all different views. On the basis of these correlations, four types of widely concerned algorithms are developed, namely algorithms focused on within-view correlations, cross-view correlations, consensus-view correlations, and self-adaptive ways [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24]. Their differences to multiple self-adaptive correlation-based multiview multilabel learning (MuSC-MVML) are given in Table II and details are reviewed in Section II. Among these algorithms, the one developed in [13] is a multilabel algorithm, the ones developed in [12], [14], [15], [17], [18], [23], [24] are multiview algorithms, and then others are multiview multilabel ones.

B. Main Problem

According to some discussions about above mentioned algorithms (see Section II and Tables I and II), there are some common shortcomings exist. 1) In order to realize the measurement of those correlations, some algorithms exploit low-rank preserving terms or self-adaptive constraints. However, they only express several or one correlation information (such as

Fig. 1. Use of multiview multilabel, multiview, and multilabel datasets in specific real-world applications. (a) Illustrative multiview multilabel scenario: two news webpages from BBC can be represented in multiview, including text and picture. For these webpages, we provide four labels which can be selected and for the top webpage, it can be tagged as "travel" and "crowd" simultaneously while for the bottom webpage, it is tagged as "sport" and "single" at the same time (the selected labels are given in red). (b) Illustrative multiview scenario: an introduction about National Museum of China can be represented in multiview, including text, picture, and video. (c) Illustrative multilabel scenario: weather forecast from Qingdao and Chongqing in five days are given and they have different labels. Among the given four selectable labels, for Qingdao, sunny and cloudy are selected while for Chongqing, cloudy, rainy, and cloudy to sunny are selected.

TABLE I
CLASSICAL MULTIVIEW ([1], [2], [3], [4]), MULTILABEL ([5], [6], [7]), AND MULTIVIEW MULTILABEL ([8], [9], [10]) ALGORITHMS AND DIFFERENCES TO OURS

Algorithm	Differences to MuSC-MVML
MVMLSS [1]	without the consideration about data correlations
SMMCL [2]	aim to optimize the propagation sequence
AMGL [3]	only use within-view instance-instance correlation
S-MVSC [4]	only use within-view and consensus-view correlations
LF-LPLC [5]	only use label-label correlations
GLOCAL [6]	
lrMMC [7]	pay more attention to consensus-view correlations
MVMLP [8]	only use within-view correlations
SSDR-MML [9]	only use within-view instance-instance correlations
LSA-MML [10]	ignore cross-view representations

TABLE II
DIFFERENCES TO MUSC-MVML FOR ALGORITHMS FOCUSED ON WITHIN-VIEW CORRELATIONS, CROSS-VIEW CORRELATIONS, CONSENSUS-VIEW CORRELATIONS, AND SELF-ADAPTIVE WAYS. "HIGH COMPUTATIONAL COST" INDICATES THE COMPUTATIONAL COST IS $O(n^3)$-LEVEL

Type	Ref.	Differences to MuSC-MVML
1	[11]	poor generalization performances
	[12]	high computational cost
	[13]	high computational cost
	[14]	only rely on instance-instance correlations
2	[15]	ignore the functions of information about labels
	[16]	high computational cost
3	[17]	unfeasible for semi-supervised classification task
	[18]	high computational cost
	[19]	ignore the cross-view correlations
	[20]	high iterations
4	[21]	highly rely on label-label correlations
	[22]	high computational cost
	[23]	cannot integrate both local and global information
	[24]	highly rely on consensus-view information

within-view feature-feature correlation) and do not study the simultaneous measurement of multiple kinds of correlation information, explore laws of self-adaptive change for multiple correlation information, and consider the self-adaptive constraints design comprehensively. Indeed, consideration of these factors will bring more concrete correlations and this is one main problem to be solved in this study. 2) Some mentioned algorithms here are only developed for multiview or multilabel data and bring high computational costs. These disadvantages also form another main problem to be solved.

C. Proposal, Contributions, and Work

To solve the above mentioned main problems, we analyze the advantages and frameworks about models in the above mentioned algorithms and refer to some ideas in these studies, especially, one in global and local multiview multilabel learning (GLMVML) [11] and the one of dual noise elimination and dynamic label correlation guided partial multilabel learning (PML-DNDC) [13].

Then, this study considers the self-adaptive measurements of the correlations in the representations about within-view, consensus-view, and cross-view. In order to express these correlations in a sound way, we explore the laws of self-adaptive change for these multiple correlation information according to the characteristics of the data further and design feasible self-adaptive constraints. The developed algorithm is called MuSC-MVML. Different from many current existing algorithms (details can be found in Section IV-D), MuSC-MVML is not just about the simple combination of current algorithms. Indeed, it has a different model and its core module is to explore the laws of self-adaptive change for these correlations.

The major contributions and novelties of this study are 1) MuSC-MVML can express correlations in multiple representations better with the exploration of self-adaptive change laws about correlations; 2) MuSC-MVML provides a better way to process multiview multilabel data, multilabel data, and multiview data simultaneously with the consideration about influence of correlations in different representations; 3) compared with the basic machines, including GLMVML and PML-DNDC, MuSC-MVML has a better performance in statistical in terms of AUC, a moderate computational cost and a relatively fast convergence.

The main work of this study includes 1) we put forward a new design concept for models of multiview multilabel learning and elaborate its theoretical fundamental, framework, optimization procedure, and computational cost; 2) we analyze influence of different correlations and laws of their self-adaptive change in the representations about within-view, consensus-view, and cross-view; and 3) we report the significant performances of MuSC-MVML compared with the baselines.

D. Framework

The study is organized as follows. Section II reviews the related works about the previous four kinds of widely concerned algorithms using correlations. Sections III and IV demonstrate the theoretical fundamental, framework, optimization procedure, and computational cost of MuSC-MVML. In Section V, we report on numerous experiments

to evaluate the proposed algorithm. In Section VI, we give some further discussions about this study. Finally, Section VII concludes this study and advises the future work.

II. Related Work

As mentioned before, there are four types of widely concerned algorithms to solve datasets with the usage of correlations. Thus, we review them and describe their advantages in detail here.

1) *Type 1: Algorithms Focused on Within-View Correlations:* Zhu et al. [11] developed GLMVML to express label-label correlations within each view and process multiview multilabel datasets well; Hajjar et al. [12] expressed correlations between labels and seek the consensus representation for cluster label matrices of different views through adding a smoothness constraint over all views; Hu et al. [13] captured correlations between labels by constructing dynamic label-level Laplacian matrix to help classifier learning. The instance-level and label-level Laplacian matrices enable the algorithm to reinforce each other, enhancing the robustness of the model. Li et al. [14] developed a method to emphasize the preservation of local residuals to enhance data representation through instance-instance correlation within views used.
2) *Type 2: Algorithms Focused on Cross-View Correlations:* Dong and Sun [15] discovered cross-view correlations by learning view-sharing and view-specific features of different views in the representation space and then they can learn the comprehensive representation of partial multiview data. By the exploration of the consistency and complementarity information across different views and with the consideration about label correlations, Zhang et al. [16] generated the common and individual representations for each instance to comprehensively characterize all of its relevant semantic labels and then to improve the performance of multilabel prediction.
3) *Type 3: Algorithms Focused on Consensus-View Correlations:* Liu et al. [17] learned a consensus-view instance-instance correlation matrix from the consensus latent data representation with the feedback information of the clustering process and the performance of clustering can be better; Zhang et al. [18] constructed a linear model to seek the correlations between the consensus latent representation for features of different views and features of each view and make the latent representation depicts data more comprehensively than each individual view with the usage of the complementarity of multiple views; Ma et al. [19] minimized the similarity between the shared and view-specific representations with consensus-view and within-view correlations considered, thereby improving diversity. Tan et al. [20] learned a consensus subspace to capture the shared instances and explore the consensus correlations among labels. They also exploit multiple individual classifiers to explore characteristics of each view. Then, one can realize an improved performance with individuality and commonality information used and enhance the robustness with respect to rare labels.
4) *Type 4: Algorithms Focused on Self-Adaptive Ways:* Liu et al. [21] constructed a crafted label correlation matrix to describe the relationships among labels self-adaptively and then utilized multiview learning and dimension reduction to exploit the high-level latent semantic label information and the latent feature information, so as to build a classifier in the low dimensional space. Wang and Xu [22] employed the KNN method to mine the correlation between the training instances with different views to generate the manifold structure self-adaptively and then exploited the relationship between labels and views, thereby reducing the impact of noisy labels with the low-rank and sparse decomposition strategy used; Liu et al. [23] expressed cross-view instance-instance correlation between two different views and the one for each view with a self-adaptive way according to the characteristics of instances so that the classification performances can be enhanced. Li et al. [24] expressed the consensus instance-instance correlation for all views with a self-adaptive way according to the characteristics of instances and labels.

TABLE III
Definitions of a Multiview Multilabel Dataset

Notation	Dimensionality
X_j, Y_j, $\widetilde{X_j}$, $\widetilde{Y_j}$	$n \times d_j$, $n \times c_j$, $n \times d_j$, $n \times c_j$
V_j, S_j, U_j, W_j	$d_j \times d_j$, $n \times n$, $c_j \times c_j$, $d_j \times c_j$
X_c, Y_c, $\widetilde{X_c}$, $\widetilde{Y_c}$	$n \times l$, $n \times k$, $n \times l$, $n \times k$
V_c, S_c, U_c, W_c	$l \times l$, $n \times n$, $k \times k$, $l \times k$
X_{ij}, Y_{ij}, $\widetilde{X_{ij}}$, $\widetilde{Y_{ij}}$	$n \times d_{ij}$, $n \times c_{ij}$, $n \times d_{ij}$, $n \times c_{ij}$
V_{ij}, S_{ij}, U_{ij}, W_{ij}	$d_{ij} \times d_{ij}$, $n \times n$, $c_{ij} \times c_{ij}$, $d_{ij} \times c_{ij}$

III. Theoretical Fundamental

A. Classical Multiple Correlations-Based Model

By summarizing the above mentioned algorithms, it has been found that a classical multiview multilabel model to solve datasets with v views always include the feature-oriented part, the label-oriented part, and the associated part.

In order to elaborate on this classical model, we use Table III to define a multiview multilabel dataset first. Here V, S, U, and W describe the corresponding feature-feature, instance-instance, label-label, and feature-label correlation matrices, respectively. $\widetilde{\mathcal{C}}$ is the latent representation of $\mathcal{C}$. Subscript "j," "c," and "ij" indicate the information about the jth view, consensus information, and cross-view information between the ith view and the jth view. For example, X_j and Y_j represent the jth view and corresponding label matrix, while X_c and Y_c represent consensus representation for all views. The meanings of symbols in "dimensionality" are as follows. n: numbers of instances; d_j (c_j): number of features (classes) for the jth view; l (k): number of features (classes) for consensus-view representation; d_{ij} (c_{ij}): number of features (classes) for cross-view representation between the ith view and the jth view. Then, we take the jth view for example.

1) For X_j, it can be reconstructed with a feature-feature correlation matrix V_j used. Elements of V_j describe the similarities between features. Then, $||X_j - \widetilde{X}_j V_j||_F^2$ can be treated as the feature-oriented term where $||\star||_F$ represents the Frobenius norm. 2) For Y_j, it can be reconstructed through the usage of a label-label correlation matrix U_j whose elements describe the similarities between labels. Then, $||Y_j - \widetilde{Y}_j U_j||_F^2$ can be treated as the label-oriented term. 3) For X_j and Y_j, they can be associated by a feature-label correlation matrix W_j whose elements describe the relationships between features and labels. Then, the associated term can be written as $||Y_j - X_j W_j||_F^2$. 4) The classical model for the jth view can be composed by the previous three terms and a common objective model is given in (1), where ηs are the corresponding hyperparameters. Here, we notice that during the realization of a classical multiview multilabel algorithm, the feasible correlation matrices are trainable and independent from the data. Namely, we always initialize their values and optimize them in an iterative way. The terms given above are used for constructing the model only. The correlation matrices cannot be calculated from data directly

$$\min \eta_1 \left|\left|X_j - \widetilde{X}_j V_j\right|\right|_F^2 + \eta_2 \left|\left|Y_j - \widetilde{Y}_j U_j\right|\right|_F^2 + \eta_3 \left|\left|Y_j - X_j W_j\right|\right|_F^2. \tag{1}$$

Since within-view, cross-view, and consensus-view are three common representations to express a multiview multilabel dataset, the previous classical model can be migrated to corresponding cross-view form (see (2)) and corresponding consensus-view form (see (3))

$$\min \eta_4 \left|\left|X_{ij} - \widetilde{X_{ij}} V_{ij}\right|\right|_F^2 + \eta_5 \left|\left|Y_{ij} - \widetilde{Y}_{ij} U_{ij}\right|\right|_F^2 + \eta_6 \left|\left|Y_{ij} - X_{ij} W_{ij}\right|\right|_F^2 \tag{2}$$

$$\min \eta_7 \left|\left|X_c - \widetilde{X}_c V_c\right|\right|_F^2 + \eta_8 \left|\left|Y_c - \widetilde{Y}_c U_c\right|\right|_F^2 + \eta_9 ||Y_c - X_c W_c||_F^2. \tag{3}$$

B. Related Self-Adaptive Terms and Laws

We summarize and analyze some existing work, including [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24]. It is found that these correlations can be described in self-adaptive ways and their changes follow some laws. For example, 1) if two instances are strongly correlated, their corresponding features and the predictive labels might be more similar; 2) if two labels are strongly correlated, their corresponding outputs might be more similar; and 3) if two instances are strongly correlated, their corresponding cross-view and consensus-view representations are more similar.

In order to realize the law 1), we utilize two regularizer terms and take the jth view as example. Namely, for X_j, its predictive label matrix is denoted as $Y_j = X_j W_j$, then x_j^a is the ath instance of X_j and y_j^a is the ath row of Y_j and it is the corresponding predictive label of x_j^a. If instance x_j^a and instance x_j^b are strongly correlated, the similarity between x_j^a and x_j^b or the one between y_j^a and y_j^b will be large. Then, we define two regularizer terms as

$$\sum_{a,b}^{n} \frac{1}{2} S_j^{ab} \left|\left|x_j^a - x_j^b\right|\right|_2^2 = tr\left(X_j^T L_{S_j} X_j\right) \tag{4}$$

$$\sum_{a,b}^{n} \frac{1}{2} S_j^{ab} \left|\left|y_j^a - y_j^b\right|\right|_2^2 = tr\left(Y_j^T L_{S_j} Y_j\right) = tr\left(W_j^T X_j^T L_{S_j} X_j W_j\right) \tag{5}$$

where S_j^{ab} describes the instance-instance correlation between instance x_j^a and instance x_j^b and L_{S_j} is the Laplacian matrix for S_j.

In order to realize the law (ii), we also utilize a regularizer term and still take the jth view as example. Since Y_j can be derived from $X_j W_j$ and its pth (or qth) column y_{jp} (or y_{jq}) describes the output of the pth (or qth) label. Thus, referring to (5), the regularizer term can be written as

$$\sum_{p,q}^{c_j} \frac{1}{2} U_j^{pq} ||y_{jp} - y_{jq}||_2^2 = tr\left(Y_j L_{U_j} Y_j^T\right) = tr\left(X_j W_j L_{U_j} W_j^T X_j^T\right) \tag{6}$$

where U_j^{pq} describes the label-label correlation between the pth and qth labels. Similar with L_{S_j}, L_{U_j} is the Laplacian matrix for U_j.

Then, referring to Section III-A, these regularizer terms can also be migrated to cross-view form (see (7)–(9)) and consensus-view form (see (10)–(12)), where $\Rightarrow$ describes the migration operation and $L_{S_{ij}}$, $L_{U_{ij}}$, L_{S_c}, and L_{U_c} are the Laplacian matrices for S_{ij}, U_{ij}, S_c, and U_c, respectively

$$tr\left(X_j^T L_{S_j} X_j\right) \Rightarrow tr\left(X_{ij}^T L_{S_{ij}} X_{ij}\right) \tag{7}$$

$$tr\left(W_j^T X_j^T L_{S_j} X_j W_j\right) \Rightarrow tr\left(W_{ij}^T X_{ij}^T L_{S_{ij}} X_{ij} W_{ij}\right) \tag{8}$$

$$tr\left(X_j W_j L_{U_j} W_j^T X_j^T\right) \Rightarrow tr\left(X_{ij} W_{ij} L_{U_{ij}} W_{ij}^T X_{ij}^T\right) \tag{9}$$

$$tr\left(X_j^T L_{S_j} X_j\right) \Rightarrow tr\left(X_c^T L_{S_c} X_c\right) \tag{10}$$

$$tr\left(W_j^T X_j^T L_{S_j} X_j W_j\right) \Rightarrow tr\left(W_c^T X_c^T L_{S_c} X_c W_c\right) \tag{11}$$

$$tr\left(X_j W_j L_{U_j} W_j^T X_j^T\right) \Rightarrow tr\left(X_c W_c L_{U_c} W_c^T X_c^T\right). \tag{12}$$

Moreover, in order to realize the law (iii), we suppose the similarity information (namely, correlation) between instances will be kept in different representations and as we know that if the correlation between the ath instance and the bth instance is larger, their corresponding cross-view and consensus-view representations are more similar. So we utilize the following regularizer terms to realize this law where x_c^a, x_c^b, y_c^a, y_c^b, x_{ij}^a, x_{ij}^b, y_{ij}^a, and y_{ij}^b are the information for the two instances in cross-view and consensus-view representations. Indeed, with the realization of this law, the relationship between within-view representation and the other two representations is built

$$\sum_{a,b}^{n} \frac{1}{2} S_j^{ab} \left|\left|x_c^a - x_c^b\right|\right|_2^2 = tr\left(X_c^T L_{S_j} X_c\right) \tag{13}$$

$$\sum_{a,b}^{n} \frac{1}{2} S_j^{ab} \left|\left|y_c^a - y_c^b\right|\right|_2^2 = tr\left(Y_c^T L_{S_j} Y_c\right) \tag{14}$$

$$\sum_{a,b}^{n}\frac{1}{2}S_j^{ab}\left|\left|x_{ij}^a-x_{ij}^b\right|\right|_2^2=tr\left(X_{ij}^TL_{S_j}X_{ij}\right) \tag{15}$$

$$\sum_{a,b}^{n}\frac{1}{2}S_j^{ab}\left|\left|y_{ij}^a-y_{ij}^b\right|\right|_2^2=tr\left(Y_{ij}^TL_{S_j}Y_{ij}\right). \tag{16}$$

Followed by above laws, the correlations can be expressed self-adaptively.

C. Strategies to Avoid the Over-Fitting Problems

As is known to all that handling multiple correlations and parameters raises concerns about potential over-fitting problems. Indeed, over-fitting problems are always caused by too large feature space and rank space, namely feature redundancy and rank redundancy. Thus, to this end, we refer to the studies in [25], [26] and find that matrix max-norm can make the metric matrices (in our study, are correlation matrices) be low rank and detecting the redundant columns for the feature matrices can make these feature spaces be sparse.

Inspired by the ideas in [25], [26], we first to adopt the matrix max-norm to enforce the low rank property on correlation matrices and let them be the constraints as below

$$\text{s.t.}\ \ ||\theta_j||_{\max},||\theta_c||_{\max},||\theta_{ij}||_{\max}\le\lambda^2 \tag{17}$$

where $\theta_j\triangleq\{V_j,S_j,U_j,W_j\}$, $\theta_c\triangleq\{V_c,S_c,U_c,W_c\}$, $\theta_{ij}\triangleq\{V_{ij},S_{ij},U_{ij},W_{ij}\}$, and $\lambda\ge 0$ is a tuning parameter. Then, since these correlation matrices are related with the features and labels, thus (17) can be replaced by the formulated as

$$\text{s.t.}\begin{cases}\left|\left|X_j\right|\right|_{2,\infty},||X_c||_{2,\infty},\left|\left|X_{ij}\right|\right|_{2,\infty}\le\lambda\\ \left|\left|Y_j\right|\right|_{2,\infty},||Y_c||_{2,\infty},\left|\left|Y_{ij}\right|\right|_{2,\infty}\le\lambda\end{cases} \tag{18}$$

where $||\star||_{2,\infty}$ is the matrix $\ell_{2,\infty}$ norm.

Second, to pursue sparse features, a natural idea is to use the $\ell_{2,1}$ norm on X_j, X_c, and X_{ij} to enforce group sparsity, that is, we expect many columns of X_j, X_c, and X_{ij} to be zeros. If a whole column of X_j, X_c, and X_{ij} is zero, then the corresponding features are detected as irrelevant. The corresponding model is given below where $\kappa_j,\alpha_c,\gamma_{ij}\ge 0$ are tuning parameters

$$\kappa_j\left|\left|X_j\right|\right|_{2,1}+\alpha_c||X_c||_{2,1}+\gamma_{ij}\left|\left|X_{ij}\right|\right|_{2,1}. \tag{19}$$

With these above strategies, the over-fitting problems can be avoided efficiently.

IV. Methodology

A. Framework of MuSC-MVML

Based on comments in Section III, we develop MuSC-MVML and its framework is given as below where $\theta=\{\theta_j,\theta_c,\theta_{ij}\}$ and $I^{V_j}\in\mathbb{R}^{d_j\times 1}$, $I^{U_j}\in\mathbb{R}^{c_j\times 1}$, $I^{S_j}\in\mathbb{R}^{n\times 1}$, $I^{V_c}\in\mathbb{R}^{l\times 1}$, $I^{U_c}\in\mathbb{R}^{k\times 1}$, $I^{S_c}\in\mathbb{R}^{n\times 1}$, $I^{V_{ij}}\in\mathbb{R}^{d_{ij}\times 1}$, $I^{U_{ij}}\in\mathbb{R}^{c_{ij}\times 1}$, and $I^{S_{ij}}\in\mathbb{R}^{n\times 1}$ are nine identity matrices. Moreover, we let the correlation between the information and itself be 0

$$\min_{\theta}\ \mathcal{L}=f_j+f_{c-1}+f_{ij-1}+f_{c-2}+f_{ij-2} \tag{20}$$

$$\text{s.t.}\begin{cases}0\le\theta_j,\theta_c,\theta_{ij}\le 1\\ V_jI^{V_j}=I^{V_j},U_jI^{U_j}=I^{U_j},S_jI^{S_j}=I^{S_j}\\ V_cI^{V_c}=I^{V_c},U_cI^{U_c}=I^{U_c},S_cI^{S_c}=I^{S_c}\\ V_{ij}I^{V_{ij}}=I^{V_{ij}},U_{ij}I^{U_{ij}}=I^{U_{ij}},S_{ij}I^{S_{ij}}=I^{S_{ij}}\\ W_jI^{U_j}=I^{V_j},W_j^TI^{V_j}=I^{U_j},W_cI^{U_c}=I^{V_c}\\ W_c^TI^{V_c}=I^{U_c},W_{ij}I^{U_{ij}}=I^{V_{ij}},W_{ij}^TI^{V_{ij}}=I^{U_{ij}}\\ V_j^{aa}=S_j^{aa}=\cdots=W_{ij}^{aa}=0\\ \left|\left|X_j\right|\right|_{2,\infty},||X_c||_{2,\infty},\left|\left|X_{ij}\right|\right|_{2,\infty}\le\lambda\\ \left|\left|Y_j\right|\right|_{2,\infty},||Y_c||_{2,\infty},\left|\left|Y_{ij}\right|\right|_{2,\infty}\le\lambda\end{cases}$$

where $f_j=\sum_{j=1}^{v}\Phi(j)$, $f_{ij-1}=\sum_{i=1,i\ne j}^{v}\sum_{j=1}^{v}\Omega(j)$, $f_{c-2}=\sum_{j=1}^{v}\Psi(j)$, $f_{ij-2}=\sum_{i=1,i\ne j}^{v}\sum_{j=1}^{v}\Theta(j)$, and

$$\begin{aligned}\Phi(j)=&\kappa_1\left|\left|X_j-\widetilde{X}_jV_j\right|\right|_F^2+\kappa_2\left|\left|Y_j-\widetilde{Y}_jU_j\right|\right|_F^2\\&+\kappa_3\left|\left|Y_j-X_jW_j\right|\right|_F^2+\kappa_4tr\left(X_j^TL_{S_j}X_j\right)+\kappa_j\left|\left|X_j\right|\right|_{2,1}\\&+\kappa_5tr\left(W_j^TX_j^TL_{S_j}X_jW_j\right)+\kappa_6tr\left(X_jW_jL_{U_j}W_j^TX_j^T\right)\end{aligned} \tag{21}$$

$$\begin{aligned}f_{c-1}=&\alpha_1\left|\left|X_c-\widetilde{X}_cV_c\right|\right|_F^2+\alpha_2\left|\left|Y_c-\widetilde{Y}_cU_c\right|\right|_F^2\\&+\alpha_3||Y_c-X_cW_c||_F^2+\alpha_4tr\left(X_c^TL_{S_c}X_c\right)+\alpha_c||X_c||_{2,1}\\&+\alpha_5tr\left(W_c^TX_c^TL_{S_c}X_cW_c\right)+\alpha_6tr\left(X_cW_cL_{U_c}W_c^TX_c^T\right)\end{aligned} \tag{22}$$

$$\begin{aligned}\Omega(j)=&\gamma_1\left|\left|X_{ij}-\widetilde{X_{ij}}V_{ij}\right|\right|_F^2+\gamma_2\left|\left|Y_{ij}-\widetilde{Y}_{ij}U_{ij}\right|\right|_F^2\\&+\gamma_3\left|\left|Y_{ij}-X_{ij}W_{ij}\right|\right|_F^2+\gamma_4tr\left(X_{ij}^TL_{S_{ij}}X_{ij}\right)\\&+\gamma_{ij}\left|\left|X_{ij}\right|\right|_{2,1}+\gamma_5tr\left(W_{ij}^TX_{ij}^TL_{S_{ij}}X_{ij}W_{ij}\right)\\&+\gamma_6tr\left(X_{ij}W_{ij}L_{U_{ij}}W_{ij}^TX_{ij}^T\right)\end{aligned} \tag{23}$$

$$\Psi(j)=\beta_1tr\left(X_c^TL_{S_j}X_c\right)+\beta_2tr\left(Y_c^TL_{S_j}Y_c\right) \tag{24}$$

$$\Theta(j)=\delta_1tr\left(X_{ij}^TL_{S_j}X_{ij}\right)+\delta_2tr\left(Y_{ij}^TL_{S_j}Y_{ij}\right). \tag{25}$$

Here, the κs, αs, γs, βs, and δs are tuning parameters.

B. Optimization

In order to optimize (20), we adopt a three-step updating strategy and in each iteration, we update the correlations, features of data, and labels of data sequentially.

First, we adopt alternating optimization strategy to update each correlation with the gradient descent way. In simple speaking, in the t-th iteration, in order to update a correlation $\mathcal{C}$, we fix others except $\mathcal{C}$ and then compute $\nabla_{\mathcal{C}}$ which is the total derivative for $\mathcal{C}$. According to (20), the $\nabla_{\mathcal{C}}$ include 11 forms (see Table IV), where $[\mathcal{E}]_{\mathcal{C}}^{\mathcal{A}}$ has the same dimensionality of $\mathcal{C}$ and its pth row and qth column element is $||\mathcal{A}_{p,:}^T-\mathcal{A}_{q,:}^T||^2$, where $\mathcal{A}_{p,:}$ and $\mathcal{A}_{q,:}$ stand for the pth and qth rows of $\mathcal{A}$, respectively. Moreover, diag$([1/||\mathcal{C}^d||])$ is a diagonal matrix and the dth element on the diagonal in this matrix is depended on $\mathcal{C}^d$ which is the dth row of $\mathcal{C}$. Then, we update correlation $\mathcal{C}$ with $\mathcal{C}(t+1)\leftarrow\mathcal{C}(t)-\nabla_{\mathcal{C}}$. Concrete formulations about these $\nabla_{\mathcal{C}}$s are given below.

1) Updating V_j: To update V_j, we fix others and optimization problem (20) can be reduced to

$$\min_{V_j}\kappa_1\left|\left|X_j-\widetilde{X}_jV_j\right|\right|_F^2. \tag{26}$$

TABLE IV
FORMS OF DERIVATIVE W.R.T. $\mathcal{C}$ AND $\mathcal{A}$, $\mathcal{B}$, AND $\mathcal{D}$ DESCRIBE DIFFERENT TERMS

Form	Computational results
$\frac{\partial\|\|\mathcal{C}-\mathcal{A}\mathcal{B}\|\|_F^2}{\partial\mathcal{C}}$	$2(\mathcal{C}-\mathcal{A}\mathcal{B})$
$\frac{\partial\|\|\mathcal{A}-\mathcal{B}\mathcal{C}\|\|_F^2}{\partial\mathcal{C}}$	$2(\mathcal{B}^T\mathcal{B}\mathcal{C}-\mathcal{B}^T\mathcal{A})$
$\frac{\partial\|\|\mathcal{A}-\mathcal{B}\mathcal{C}\mathcal{D}\|\|_F^2}{\partial\mathcal{C}}$	$2(\mathcal{B}^T\mathcal{B}\mathcal{C}\mathcal{D}\mathcal{D}^T-\mathcal{B}^T\mathcal{A}\mathcal{D}^T)$
$\frac{\partial\|\|\mathcal{A}-\mathcal{B}\mathcal{C}\mathcal{C}^T\mathcal{D}\|\|_F^2}{\partial\mathcal{C}}$	$-2(\mathcal{D}\mathcal{A}^T\mathcal{B}\mathcal{C}+\mathcal{B}^T\mathcal{A}\mathcal{D}^T\mathcal{C})+4\mathcal{B}^T\mathcal{B}\mathcal{C}\mathcal{C}^T\mathcal{D}\mathcal{D}^T\mathcal{C}$
$\frac{\partial\|\|\mathcal{A}\mathcal{C}-\mathcal{B}\mathcal{C}\|\|_F^2}{\partial\mathcal{C}}$	$2(\mathcal{A}-\mathcal{B})^T(\mathcal{A}-\mathcal{B})\mathcal{C}$
$\frac{\partial tr(\mathcal{A}L_{\mathcal{C}}\mathcal{A}^T)}{\partial\mathcal{C}}$	$\frac{1}{2}[\mathcal{E}]_{\mathcal{C}}^{\mathcal{A}}$
$\frac{\partial tr(\mathcal{C}\mathcal{A}\mathcal{C}^T)}{\partial\mathcal{C}}$	$\mathcal{C}\mathcal{A}^T+\mathcal{C}\mathcal{A}$
$\frac{\partial tr(\mathcal{C}^T\mathcal{A}\mathcal{C})}{\partial\mathcal{C}}$	$\mathcal{A}\mathcal{C}+\mathcal{A}^T\mathcal{C}$
$\frac{\partial tr(\mathcal{A}\mathcal{C}^T\mathcal{B}\mathcal{C}\mathcal{A}^T)}{\partial\mathcal{C}}$	$\mathcal{B}^T\mathcal{C}\mathcal{A}^T\mathcal{A}+\mathcal{B}\mathcal{C}\mathcal{A}^T\mathcal{A}$
$\frac{\partial tr(\mathcal{A}\mathcal{C}\mathcal{B}\mathcal{C}^T\mathcal{A}^T)}{\partial\mathcal{C}}$	$\mathcal{A}^T\mathcal{A}\mathcal{C}\mathcal{B}^T+\mathcal{A}^T\mathcal{A}\mathcal{C}\mathcal{B}$
$\frac{\partial\|\|\mathcal{C}\|\|_{2,1}}{\partial\mathcal{C}}$	$diag(\frac{1}{\|\|\mathcal{C}^d\|\|})\mathcal{C}$

Then, the gradient of (26) with respect to V_j is

$$\nabla_{V_j}=2\kappa_1\left(\widetilde{X}_j^T\widetilde{X}_jV_j-\widetilde{X}_j^TX_j\right). \tag{27}$$

2) Updating S_j: To update S_j, we fix others and optimization problem (20) can be reduced to

$$\begin{aligned}\min_{S_j}\ &\kappa_4tr\left(X_j^TL_{S_j}X_j\right)+\kappa_5tr\left(W_j^TX_j^TL_{S_j}X_jW_j\right)\\&+\beta_1tr\left(X_c^TL_{S_j}X_c\right)+\beta_2tr\left(Y_c^TL_{S_j}Y_c\right)\\&+\sum_{i=1,i\neq j}^{v}\left[\delta_1tr\left(X_{ij}^TL_{S_j}X_{ij}\right)+\delta_2tr\left(Y_{ij}^TL_{S_j}Y_{ij}\right)\right].\end{aligned} \tag{28}$$

Then, the gradient of (28) with respect to S_j is

$$\begin{aligned}\nabla_{S_j}=\frac{1}{2}\Bigg(&\kappa_4[\mathcal{E}]_{S_j}^{X_j^T}+\kappa_5[\mathcal{E}]_{S_j}^{W_j^TX_j^T}+\beta_1[\mathcal{E}]_{S_j}^{X_c^T}\\&+\beta_2[\mathcal{E}]_{S_j}^{Y_c^T}+\sum_{i=1,i\neq j}^{v}[\delta_1[\mathcal{E}]_{S_j}^{X_{ij}^T}+\delta_2[\mathcal{E}]_{S_j}^{Y_{ij}^T}]\Bigg).\end{aligned} \tag{29}$$

3) Updating U_j: To update U_j, we fix others and optimization problem (20) can be reduced to

$$\min_{U_j}\kappa_2\left|\left|Y_j-\widetilde{Y}_jU_j\right|\right|_F^2+\kappa_6tr\left(X_jW_jL_{U_j}W_j^TX_j^T\right). \tag{30}$$

Then, the gradient of (30) with respect to U_j is

$$\nabla_{U_j}=2\kappa_2\left(\widetilde{Y}_j^T\widetilde{Y}_jU_j-\widetilde{Y}_j^TY_j\right)+\tfrac{1}{2}\kappa_6[\mathcal{E}]_{U_j}^{X_jW_j}. \tag{31}$$

4) Updating W_j: To update W_j, we fix others and optimization problem (20) can be reduced to

$$\begin{aligned}\min_{W_j}\kappa_3\left|\left|Y_j-X_jW_j\right|\right|_F^2&+\kappa_5tr\left(W_j^TX_j^TL_{S_j}X_jW_j\right)\\&+\kappa_6tr\left(X_jW_jL_{U_j}W_j^TX_j^T\right).\end{aligned} \tag{32}$$

Then, the gradient of (32) with respect to W_j is

$$\begin{aligned}\nabla_{W_j}=&2\kappa_3\left(X_j^TX_jW_j-X_j^TY_j\right)+\kappa_5\left(X_j^TL_{S_j}X_jW_j+X_j^TL_{S_j}^TX_jW_j\right)\\&+\kappa_6\left(X_j^TX_jW_jL_{U_j}^T+X_j^TX_jW_jL_{U_j}\right).\end{aligned} \tag{33}$$

5) Updating V_c: To update V_c, we fix others and optimization problem (20) can be reduced to

$$\min_{V_c}\alpha_1\left|\left|X_c-\widetilde{X}_cV_c\right|\right|_F^2. \tag{34}$$

Then, the gradient of (34) with respect to V_c is

$$\nabla_{V_c}=2\alpha_1\left(\widetilde{X}_c^T\widetilde{X}_cV_c-\widetilde{X}_c^TX_c\right). \tag{35}$$

6) Updating S_c: To update S_c, we fix others and optimization problem (20) can be reduced to

$$\min_{S_c}\alpha_4tr\left(X_c^TL_{S_c}X_c\right)+\alpha_5tr\left(W_c^TX_c^TL_{S_c}X_cW_c\right). \tag{36}$$

Then, the gradient of (36) with respect to S_c is

$$\nabla_{S_c}=\tfrac{1}{2}\left(\alpha_4[\mathcal{E}]_{S_c}^{X_c^T}+\alpha_5[\mathcal{E}]_{S_c}^{W_c^TX_c^T}\right). \tag{37}$$

7) Updating U_c: To update U_c, we fix others and optimization problem (20) can be reduced to

$$\min_{U_c}\alpha_2\left|\left|Y_c-\widetilde{Y}_cU_c\right|\right|_F^2+\alpha_6tr\left(X_cW_cL_{U_c}W_c^TX_c^T\right). \tag{38}$$

Then, the gradient of (38) with respect to U_c is

$$\nabla_{U_c}=2\alpha_2\left(\widetilde{Y}_c^T\widetilde{Y}_cU_c-\widetilde{Y}_c^TY_c\right)+\tfrac{1}{2}\alpha_6[\mathcal{E}]_{U_c}^{X_cW_c}. \tag{39}$$

8) Updating W_c: To update W_c, we fix others and optimization problem (20) can be reduced to

$$\begin{aligned}\min_{W_c}\alpha_3||Y_c-X_cW_c||_F^2&+\alpha_5tr\left(W_c^TX_c^TL_{S_c}X_cW_c\right)\\&+\alpha_6tr\left(X_cW_cL_{U_c}W_c^TX_c^T\right).\end{aligned} \tag{40}$$

Then, the gradient of (40) with respect to W_c is

$$\begin{aligned}\nabla_{W_c}=&2\alpha_3\left(X_c^TX_cW_c-X_c^TY_c\right)+\alpha_5\left(X_c^TL_{S_c}X_cW_c+X_c^TL_{S_c}^TX_cW_c\right)\\&+\alpha_6\left(X_c^TX_cW_cL_{U_c}^T+X_c^TX_cW_cL_{U_c}\right).\end{aligned} \tag{41}$$

9) Updating V_{ij}: To update V_{ij}, we fix others and optimization problem (20) can be reduced to

$$\min_{V_{ij}}\gamma_1\left|\left|X_{ij}-\widetilde{X}_{ij}V_{ij}\right|\right|_F^2. \tag{42}$$

Then, the gradient of (42) with respect to V_{ij} is

$$\nabla_{V_{ij}}=2\gamma_1\left(\widetilde{X}_{ij}^T\widetilde{X}_{ij}V_{ij}-\widetilde{X}_{ij}^TX_{ij}\right). \tag{43}$$

10) Updating S_{ij}: To update S_{ij}, we fix others and optimization problem (20) can be reduced to

$$\min_{S_{ij}}\gamma_4tr\left(X_{ij}^TL_{S_{ij}}X_{ij}\right)+\gamma_5tr\left(W_{ij}^TX_{ij}^TL_{S_{ij}}X_{ij}W_{ij}\right). \tag{44}$$

Then, the gradient of the (44) with respect to S_{ij} is

$$\nabla_{S_{ij}}=\tfrac{1}{2}\left(\gamma_4[\mathcal{E}]_{S_{ij}}^{X_{ij}^T}+\gamma_5[\mathcal{E}]_{S_{ij}}^{W_{ij}^TX_{ij}^T}\right). \tag{45}$$

11) Updating U_{ij}: To update U_{ij}, we fix others and optimization problem (20) can be reduced to

$$\min_{U_{ij}}\gamma_2\left|\left|Y_{ij}-\widetilde{Y}_{ij}U_{ij}\right|\right|_F^2+\gamma_6tr\left(X_{ij}W_{ij}L_{U_{ij}}W_{ij}^TX_{ij}^T\right). \tag{46}$$

Then, the gradient of (46) with respect to U_{ij} is

$$\nabla_{U_{ij}}=2\gamma_2\left(\widetilde{Y}_{ij}^T\widetilde{Y}_{ij}U_{ij}-\widetilde{Y}_{ij}^TY_{ij}\right)+\tfrac{1}{2}\gamma_6[\mathcal{E}]_{U_{ij}}^{X_{ij}W_{ij}}. \tag{47}$$

12) Updating W_{ij}: To update W_{ij}, we fix others and optimization problem (20) can be reduced to

$$\min_{W_{ij}} \gamma_5 tr\Big(W_{ij}^T X_{ij}^T L_{S_{ij}} X_{ij} W_{ij}\Big) + \gamma_6 tr\Big(X_{ij} W_{ij} L_{U_{ij}} W_{ij}^T X_{ij}^T\Big) + \gamma_3 \big|\big|Y_{ij} - X_{ij} W_{ij}\big|\big|_F^2. \quad (48)$$

Then, the gradient of (48) with respect to W_{ij} is

$$\begin{aligned}\nabla_{W_{ij}} = {} & 2\gamma_3\Big(X_{ij}^T X_{ij} W_{ij} - X_{ij}^T Y_{ij}\Big) \\ & + \gamma_5\Big(X_{ij}^T L_{S_{ij}} X_{ij} W_{ij} + X_{ij}^T L_{S_{ij}}^T X_{ij} W_{ij}\Big) \\ & + \gamma_6\Big(X_{ij}^T X_{ij} W_{ij} L_{U_{ij}}^T + X_{ij}^T X_{ij} W_{ij} L_{U_{ij}}\Big). \end{aligned} \quad (49)$$

Second, on the base of the updated correlations, we update the features of the data. In simple speaking, for X_j, X_c, and X_{ij}, according to Table IV, the gradient of (20) with respect to them are given as below, respectively, and in these formulations, the correlations have been updated by the above (26)–(49), while $d \in [1, n]$

$$\begin{aligned}\nabla_{X_j} = {} & 2\kappa_1\big(X_j - \widetilde{X}_j V_j\big) + \kappa_j diag\left(\frac{1}{\big|\big|X_j^d\big|\big|}\right) X_j \\ & + 2\kappa_3\Big(X_j W_j W_j^T - Y_j W_j^T\Big) + \kappa_4\Big(L_{S_j} X_j + L_{S_j}^T X_j\Big) \\ & + \kappa_5\Big(L_{S_j}^T X_j W_j W_j^T + L_{S_j} X_j W_j W_j^T\Big) \\ & + \kappa_6\Big(X_j W_j L_{U_j}^T W_j^T + X_j W_j L_{U_j} W_j^T\Big) \end{aligned} \quad (50)$$

$$\begin{aligned}\nabla_{X_c} = {} & 2\alpha_1\big(X_c - \widetilde{X}_c V_c\big) + \alpha_j diag\left(\frac{1}{\big|\big|X_c^d\big|\big|}\right) X_c \\ & + 2\alpha_3\big(X_c W_c W_c^T - Y_c W_c^T\big) + \alpha_4\big(L_{S_c} X_c + L_{S_c}^T X_c\big) \\ & + \alpha_5\big(L_{S_c}^T X_c W_c W_c^T + L_{S_c} X_c W_c W_c^T\big) \\ & + \alpha_6\big(X_c W_c L_{U_c}^T W_c^T + X_c W_c L_{U_c} W_c^T\big) \\ & + \sum_{j=1}^{v}\Big(\beta_1\big(L_{S_j} X_c + L_{S_j}^T X_c\big)\Big) \end{aligned} \quad (51)$$

$$\begin{aligned}\nabla_{X_{ij}} = {} & 2\gamma_1\big(X_{ij} - \widetilde{X_{ij}} V_{ij}\big) + \gamma_j diag\left(\frac{1}{\big|\big|X_{ij}^d\big|\big|}\right) X_{ij} \\ & + 2\gamma_3\Big(X_{ij} W_{ij} W_{ij}^T - Y_{ij} W_{ij}^T\Big) + \gamma_4\Big(L_{S_{ij}} X_{ij} + L_{S_{ij}}^T X_{ij}\Big) \\ & + \gamma_5\Big(L_{S_{ij}}^T X_{ij} W_{ij} W_{ij}^T + L_{S_{ij}} X_{ij} W_{ij} W_{ij}^T\Big) \\ & + \gamma_6\Big(X_{ij} W_{ij} L_{U_{ij}}^T W_{ij}^T + X_{ij} W_{ij} L_{U_{ij}} W_{ij}^T\Big) \\ & + \delta_1\Big(L_{S_j} X_{ij} + L_{S_j}^T X_{ij}\Big). \end{aligned} \quad (52)$$

Then, on the base of the above three formulations, we update the features by $X_j(t+1) \leftarrow X_j(t) - \nabla_{X_j}$, $X_c(t+1) \leftarrow X_c(t) - \nabla_{X_c}$, and $X_{ij}(t+1) \leftarrow X_{ij}(t) - \nabla_{X_{ij}}$.

Third, with the updated correlations and features, we can update the labels of data. Namely, $Y_j(t+1) = X_j(t+1)W_j(t+1)$, $Y_{ij}(t+1) = X_{ij}(t+1)W_{ij}(t+1)$, and $Y_c(t+1) = X_c(t+1)W_c(t+1)$.

According to the above three-step updating strategy, the optimization procedure will be terminate until the changes about the normalized value of $\mathcal{L}$ is lesser than some threshold values. Once the optimization procedure is terminated, we can get the optimal correlations and corresponding feature and label spaces. Moreover, the feature space and rank space can be small so that we can avoid the over-fitting problems. For convenience, we use Fig. 2 to demonstrate the framework of MuSC-MVML in simple.

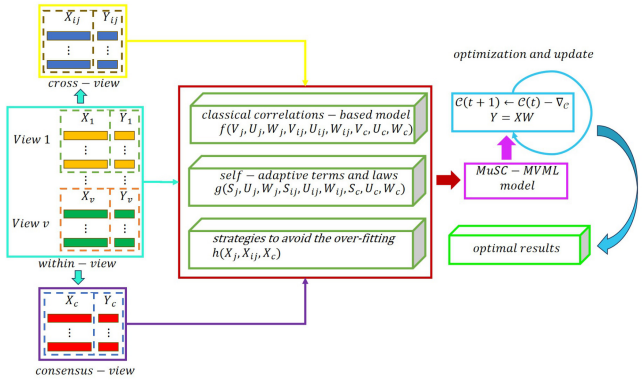

Fig. 2. Illustration of the framework of MuSC-MVML.

TABLE V
COMPUTATIONAL COST OF $\nabla_{\mathcal{C}}$

$\nabla_{\mathcal{C}}$	Computational cost
∇_{V_j}	$O(d_j^2(2n+d_j))$
∇_{S_j}	$O(6n^2)$
∇_{U_j}	$O(c_j^2(2n+c_j+1))$
∇_{W_j}	$O(4d_j^2 n + 3d_j n^2 + d_j c_j(5d_j+2c_j))$
∇_{X_j}	$O(nd_j^2 + 5n^2 d_j + n^2 c_j + 2nc_j^2 + 10nd_j c_j)$
∇_{V_c}	$O(l^2(n+l))$
∇_{S_c}	$O(2n^2)$
∇_{U_c}	$O(k^2(2n+k+1))$
∇_{W_c}	$O(4l^2 n + 3ln^2 + lk(5l+2k))$
∇_{X_c}	$O(nl^2 + n^2 k + 2nk^2 + 10nlk + n^2 l(5+2v))$
$\nabla_{V_{ij}}$	$O(d_{ij}^2(2n+d_{ij}))$
$\nabla_{S_{ij}}$	$O(2n^2)$
$\nabla_{U_{ij}}$	$O(c_{ij}^2(2n+c_{ij}+1))$
$\nabla_{W_{ij}}$	$O(4d_{ij}^2 n + 3d_{ij} n^2 + d_{ij} c_{ij}(5d_{ij}+2c_{ij}))$
$\nabla_{X_{ij}}$	$O(nd_{ij}^2 + 7n^2 d_{ij} + n^2 c_{ij} + 2nc_{ij}^2 + 10nd_{ij} c_{ij})$

C. Computational Cost

According to Sections IV-A and IV-B, the computations of $\nabla_{\mathcal{C}}$ determine the computational cost of MuSC-MVML. Then, we analyze the corresponding computational costs of them in Table V. According to Table V, we can get the total computation cost of MuSC-MVML as below

$$\begin{aligned} O(\mathcal{L}) = {} & \sum_{j=1}^{v}\big[O(\nabla_{V_j}) + O(\nabla_{S_j}) + O(\nabla_{U_j}) \\ & + O(\nabla_{W_j}) + O(\nabla_{X_j})\big] + O(\nabla_{V_c}) + O(\nabla_{S_c}) + O(\nabla_{U_c}) \\ & + O(\nabla_{W_c}) + O(\nabla_{X_c}) + \sum_{j=1}^{v}\sum_{i=1, i\neq j}^{v}\big[O(\nabla_{V_{ij}}) \\ & + O(\nabla_{S_{ij}}) + O(\nabla_{U_{ij}}) + O(\nabla_{W_{ij}}) + O(\nabla_{X_{ij}})\big]. \end{aligned} \quad (53)$$

Since in generally, $n > d_j, c_j, l, k, d_{ij}, c_{ij}$, thus $O(\mathcal{L}) \leq v[2O(C^{(1)}n) + O(C^{(2)}n^2) + 2O(C^{(3)}n + C^{(4)}n^2)] + 2O(C^{(5)}n) + O(C^{(6)}n^2) + 2O(C^{(7)}n + C^{(8)}n^2) + v(v-1)[2O(C^{(9)}n) +$

$O(C^{(10)}n^2) + 2O(C^{(11)}n + C^{(12)}n^2)] \leq (2v + 2 + 2v^2 - 2v)[O(C^{(13)}n) + O(C^{(14)}n + C^{(15)}n^2)] + (v + 1 + v^2 - v)O(C^{(16)}n^2) = (v^2+1)(2O(C^{(13)}n)+2O(C^{(14)}n+C^{(15)}n^2)+O(C^{(16)}n^2))$, where $C^{(\star)}$ represents a constant.

According to the above computation, it is found the maximum computational cost for MuSC-MVML is $(v^2 + 1)(2O(Sn)+2O(Rn+Qn^2)+O(Tn^2))$, where S, R, Q, and T are four constants and this causes the total computational cost of MuSC-MVML is closely related to the number of views and instances and compared with the existing algorithms [12], [13], [16], [18], [22] whose computational costs are $O(n^3)$-levels, the computational cost of MuSC-MVML is much smaller.

D. Difference Between Ours and the Existing Algorithms

In previous contents, including Tables I and II, we have mentioned many multiview multilabel, multilabel, multiview algorithms, and reviewed four kinds of widely concerned algorithms using correlations. Thus, in this section, we state the difference between MuSC-MVML and these algorithms clearly. Indeed, as is a multiview multilabel learning algorithm with self-adaptive correlations used, compared with these mentioned algorithms, our MuSC-MVML have significant differences as below.

First, MuSC-MVML has a different model. As we know, the model of MuSC-MVML introduces some $\ell_{2,\infty}$ norm and $\ell_{2,1}$ norm terms and it considers the sound self-adaptive measurements of multiple correlations according to some laws which is also the core module of MuSC-MVML. Indeed, some compared algorithms take different ways, including creating new instances to handle with the data and some compared algorithms ignore those laws to some extents. All mentioned algorithms above here cannot measure multiple within-view, cross-view, and consensus-view correlations in a self-adaptive way simultaneously neither.

Second, MuSC-MVML has an ability to process multiview multilabel data, multilabel data, and multiview data simultaneously. But some mentioned existing algorithms do not possess this ability due to they only focus on the processing of multiview data or multilabel data while some other algorithms cannot effectively utilize this ability due to they ignore the influence of correlations in different representations.

Third, although some classical algorithms adopt self-adaptive correlations [21], [22], [23], [24], there still exist some significant differences between ours and these algorithms. Indeed, for our developed MuSC-MVML, we measure the correlations self-adaptively with three laws (see Section III-B) considered. While for those classical algorithms, they will not consider too much. For example, for the models in [21], [22], [23], [24], they only consider the law that if two instances are strongly correlated, their corresponding features or/and the predictive labels might be more similar and then they design the self-adaptive terms to measure correlations.

Moreover, in our experiments, we select parts of above mentioned algorithms here and use statistical analysis (see Section V-C) to demonstrate the significant differences between MuSC-MVML and them further.

TABLE VI
DETAILED INFORMATION OF USED DATASETS

Order	Data	Instance-label-view	Scenario
1	Mfeat	2000-10-6	Handwritten digit
2	Reuters	111740-6-5	News article
3	Corel	1000-10-4	Image
4	VOC	9963-20-2	Image
5	MIR	23691-38-2	Retrieval evaluation
6	3Source	169-3-3	News article
7	MSRC-v1	210-7-3	Image
8	Cal101-20	2386-20-6	Image
9	Arts	5000-26-/	Web page
10	Business	5000-30-/	Web page
11	Computers	5000-33-/	Web page
12	Education	5000-33-/	Web page
13	Entertainment	5000-21-/	Web page
14	Health	5000-32-/	Web page
15	Recreation	5000-22-/	Web page
16	Reference	5000-33-/	Web page
17	Science	5000-40-/	Web page
18	Social	5000-39-/	Web page
19	Society	5000-27-/	Web page
20	Enron	1702-53-/	Email Corpus
21	Corel5K	5000-374-/	Image
22	Image	2000-5-/	Image
23	Medical	978-45-/	Clinical
24	Language Log	1459-75-/	Language log
25	RCV1V2(subset1)	6000-101-/	Article
26	RCV1V2(subset2)	6000-101-/	Article
27	Bibtex	7395-159-/	Bibtex entries
28	Delicious	16105-983-/	Web page
29	Eur-Lex(Sm)	19348-201-/	Law
30	Bookmark	87856-208-/	Bookmark entries
31	Nuswide	269468-81-/	Image
32	TMC2007-500	28596-22-/	Aviation safety
33	Stackex-Chemistry	6961-175-/	Stack exchange
34	Stackex-Chess	1675-227-/	Stack exchange
35	Stackex-Cooking	10491-400-/	Stack exchange
36	Stackex-Cs	9270-274-/	Stack exchange
37	Stackex-Philosophy	3971-233-/	Stack exchange
38	NUS-WIDE	810-81-6	Image

V. EXPERIMENTS

A. Experimental Setup

1) *Data Setting:* Table VI shows the datasets we employed in the experiments and the application scenarios of them are also given. The first eight datasets are multiview and the final one is multiview multilabel. For others, they are multilabel. All datasets are available in the following repositories or from following organizations. Namely, UCI,[1] Mulan,[2] University of Oxford,[3] LIACS Medialab at Leiden University,[4] University College Dublin,[5] Jose M. Moyano (jmoyano@uco.es),[6] Microsoft,[7] and Caltech Library.[8] For each dataset, we randomly sample 70% data for training and use the remaining 30% data for testing (unlabeled data). Then, for the training set, we carry out the ten-fold cross-validation and based on the partitions of training set, we adjust the parameter values of an algorithm and get the optimal parameter values.[9]

[1] http://archive.ics.uci.edu/ml/datasets/
[2] http://mulan.sourceforge.net/datasets-mlc.html
[3] http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html
[4] https://press.liacs.nl/mirflickr/#sec_download
[5] http://mlg.ucd.ie/datasets/3sources.html
[6] http://www.uco.es/kdis/mllresources/
[7] http://research.microsoft.com/en-us/projects/ObjectClassRecognition/
[8] https://data.caltech.edu/records/mzrjq-6wc02
[9] Best parameter values correspond to the best AUC on training set and all compared algorithms share same partitions and a same way to select optimal parameter values.

TABLE VII
RANGE OF PARAMETERS FOR OTHER COMPARED ALGORITHMS AND THE FEASIBLE SETTINGS IN OUR EXPERIMENTS (IN BOLD), WHERE l IS THE NUMBER OF LABELS

Algorithm	Parameter symbol and meaning	Range and feasible setting
MVMLSS [1]	iterations k	**30**
SMMCL [2]	tradeoff parameter α	**1**
	tradeoff parameter β	**0.5**
AMGL [3]	no parameter to handle	/
S-MVSC [4]	regularization parameter λ	**0.02**, 0.06, ..., 0.18, 0.22
MLPL [17]	nearest neighbors c	5, ..., **20**, **25**, ..., 50
	penalty factor λ	**1**
LMSC [18]	hyperparameter λ	**0.01**, 0.1, 1, 10, 100
	weight α	0.1, 0.2, ..., **0.5**, ..., 1
	network parameter γ	**0.001**
	dimensionality k	10, 20, ..., **100**
LF-LPLC [5]	ratio of clusters r	**0.1**
	nearest neighbors k	**10**
GLOCAL [6]	tradeoff parameter λ	**1**
	regularization parameter λ_2	$2^{-5}, 2^{-4}, \mathbf{2^{-3}}, ..., 2^{0}$
	regularization parameters λ_3, λ_4	$0, 10^{-6}, ..., \mathbf{10^{-3}}, ..., 10^{0}$
	cluster number g	$2^{0}, 2^{1}, ..., \mathbf{2^{4}}, ..., 2^{7}$
	dimensionality k	3, 5, 10, **15**, 20, 25, 30
lrMMC [7]	tradeoff parameter μ	$\mathbf{10^{-12}}$
	tradeoff parameter λ	$10^{-4}, ..., \mathbf{10^{-1}}, ..., 10^{2}, 10^{3}$
PML-DNDC [13]	balancing parameters λ_1, λ_4	$10^{-6}, \mathbf{10^{-5}}, ..., 10^{-1}$
	scalar weight λ_2	$\mathbf{10^{-4}}, ..., 10^{0}, 10^{1}$
	scalar weight λ_3	$10^{-6}, \mathbf{10^{-5}}, ..., 10^{-1}$
MVMLP [8]	weight α_{ij}	0.1, 0.2, ..., **0.5**, ..., 1
	weight β_{ij}	**0.1**, ..., 0.9, 1
	tradeoff parameter b	**1**
	tradeoff parameter μ_i	**0.01**
GLMVML [11]	parameter to k-means K	$(1, 2, ..., \mathbf{5}, ..., 10) \times l$
	regularization parameters $\lambda_0 \sim \lambda_3$	$2^{-5}, ..., 2^{-1}, \mathbf{2^{0}}$
	regularization parameters $\lambda_4 \sim \lambda_7$	$10^{-6}, ..., \mathbf{10^{-3}}, ..., 10^{0}$
	complementary parameters λ_8, λ_9	$\mathbf{10^{-2}}, 10^{-1}, ..., 10^{1}, 10^{2}$
ICM2L [20]	tradeoff parameter α	0.1, 0.2, ..., **0.6**, ..., 1
	tradeoff parameter β	0.1, 0.3, ..., **0.7**, ..., 1.9
ELSMML [21]	regularization coefficient β_1	$10^{-1}, \mathbf{10^{0}}, ..., 10^{2}$
	regularization coefficient β_2	$10^{-3}, 10^{-2}, \mathbf{10^{-1}}, ..., 10^{2}$
	regularization coefficient β_3	$10^{-3}, \mathbf{10^{-2}}, ..., 10^{1}$
TFMDD [22]	balancing parameter λ	**1**, 10
	balancing parameter γ	$\mathbf{10^{-5}}, 10^{-4}, 10^{-3}$
	balancing parameter β	$10^{3}, 10^{4}, \mathbf{10^{5}}$
	nearest neighbors k	1, 2, ..., **6**, ..., 10
	disambiguation threshold t_d	**0.1**, 0.2, ..., 0.8, 0.9

Then, we use the optimal parameter values to form a model for the algorithm and validate its performances with the test set. After repeating the random sampling and experiments for five times independently, the average performances and corresponding standard deviations can be gotten. Moreover, since the five sets about optimal parameter values cannot be averaged, thus we will demonstrate their feasible settings in Section V-D.

2) *Baseline Algorithms:* We select 15 algorithms from Tables I and II for comparisons. Namely, multiview algorithms MVMLSS [1], SMMCL [2], AMGL [3], S-MVSC [4], MLPL [17], and LMSC [18]; multilabel ones LF-LPLC [5], GLOCAL [6], lrMMC [7], and PML-DNDC [13]; and multiview multilabel ones MVMLP [8], GLMVML [11], ICM2L [20], ELSMML [21], and TFMDD [22].
3) *Parameter Setting:* For compared algorithms, optimal parameters can be selected from corresponding ranges which are given in the original studies and Table VII. For MuSC-MVML, tuning parameter λ is set as 1 which can be referred to [25] and for others, optimal κ_1–κ_3, α_1–α_3, and γ_1–γ_3 can be selected from the set $\{0.1, 0.2, \ldots, 0.8, 0.9\}$, optimal κ_4–κ_6, α_4–α_6, γ_4–γ_6, $\beta_1, \beta_2, \delta_1$, and δ_2 can be selected from the set $\{10^{-6}, 10^{-5}, \ldots, 10^{-1}, 10^{0}\}$, optimal parameters in κ_j, α_c, and γ_{ij} can be selected from the set $\{0.1, 0.3, \ldots, 1.7, 1.9\}$, and *elements in a correlation matrix are initialized in equipartition*. Moreover, threshold value for optimization procedure, i.e., $\mathcal{L}$ is set as 0.01.
4) *Evaluation:* We adopt three metrics for performance comparisons. They are AUC, training time (in seconds), and convergence. Then, we further compare the different influence of parameters and show the statistical analysis, ablation study results, and the ability to express correlations in multiple representations of MuSC-MVML.
5) Experimental environment: Computational environment is a node of compute cluster with 32 CPUs (Intel Core i7-6950X), operation system is RedHat Linux Enterprise 8.0, and the coding environment is MATLAB 2020a.

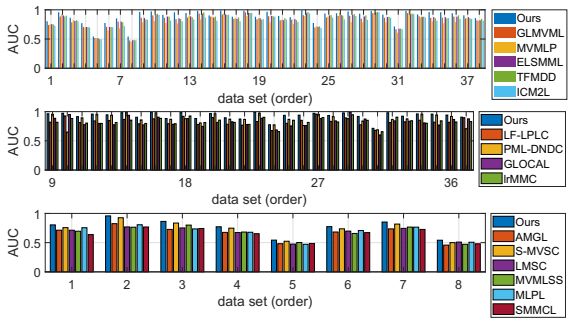

Fig. 3. AUC comparisons for the used datasets. Top: multiview multilabel case, middle: multilabel case, and bottom: multiview case.

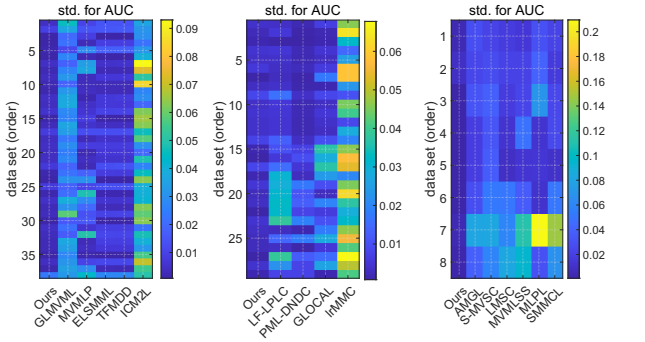

Fig. 4. Standard deviation (std) comparisons about AUC. Left: multiview multilabel case, middle: multilabel case, and right: multiview case.

B. Results on All Datasets With the Used Metrics

Figs. 3 and 4 demonstrate the AUC and corresponding standard deviations of all algorithms on all datasets (orders in the "dataset (order)" axis of these subfigures correspond to the ones in Table VI). Then, Fig. 5 demonstrates the training time comparison results and for each dataset, we let the training time of MuSC-MVML be 1 and the results of other algorithms are scaled. On the basis of the reported results given in these three figures, Fig. 6 demonstrates a summarization that in terms of AUC and training time, compared with other algorithms, how much performance improvements can MuSC-MVML bring. In this figure, each value represents an incremental ratio. Take value 0.0993 in the row "AUC" and column "LMSC" as an example, this value describes that MuSC-MVML brings a 9.93% incremental ratio on AUC compared with LMSC. Moreover, Fig. 7 demonstrates the convergence of MuSC-MVML on all used datasets. In this

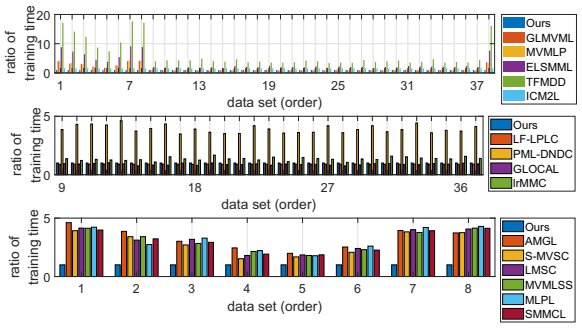

Fig. 5. Training time comparisons. Top: multiview multilabel case, middle: multilabel case, and bottom: multiview case.

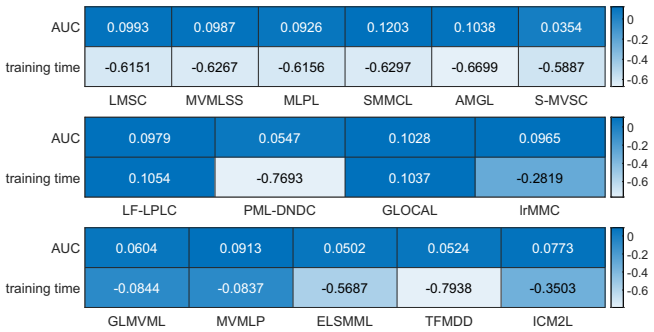

Fig. 6. Difference between MuSC-MVML and compared algorithms. Each value represents an incremental ratio.

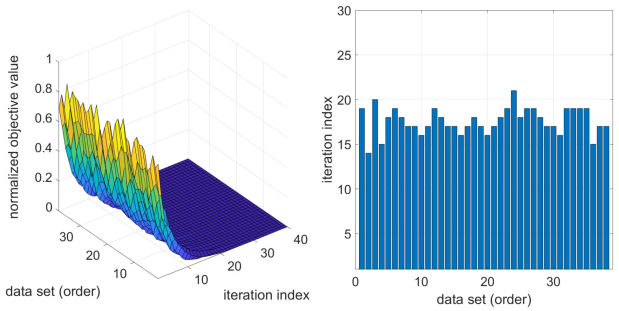

Fig. 7. Convergence of MuSC-MVML on the used datasets and the objective value has been normalized.

figure, iteration index stands for the index when the changes of normalized objective value is smaller than 0.01.

According to these figures, some conclusions can be drawn. First, in terms of AUC, the performances of MuSC-MVML are more stable and it outperforms other algorithms in average. Second, although MuSC-MVML considers the self-adaptive measurement of multiple correlations and its model becomes more complicated, its training time still be feasible and not be increased too much, especial for multiview datasets and multiview multilabel dataset. Finally, MuSC-MVML tends to converge within 20 iterations in our experiments for most of the used datasets.

C. Statistical Analysis

Friedman–Nemenyi statistical test [27][10] is used to check if the differences between MuSC-MVML and other compared

[10]We adopt AUC for the elaboration.

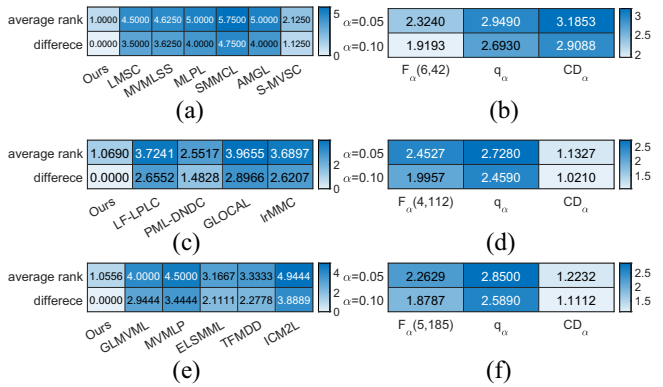

Fig. 8. Statistical analysis for MuSC-MVML in terms of AUC. (a) and (b) multiview case. (c) and (d) multilabel case. (e) and (f) multiview multilabel case.

algorithms are significant or not. On the basis of AUC results on all datasets, Fig. 8 demonstrates the average ranks of all used algorithms, rank differences between MuSC-MVML and others, and corresponding statistical values.

According to this figure and refer to [27], we first carry out Friedman test. 1) For multiview case, we adopt eight datasets and seven algorithms (i.e., $N = 8$ and $k = 7$) for experiments and we get Friedman statistic as follows. $\chi_F^2 = [12 \times N/k(k+1)][1.0000^2 + 4.5000^2 + 4.6250^2 + 5.0000^2 + 5.7500^2 + 5.0000^2 + 2.1250^2 - (k(k+1)^2/4)] = 31.2321$, $F_F = [(N-1)\chi_F^2/N(k-1) - \chi_F^2] = 13.0383$, $F_{0.05}(k-1,(k-1)(N-1)) = F_{0.05}(6,42) = 2.3240$, and $F_{0.10}(k-1,(k-1)(N-1)) = F_{0.10}(6,42) = 1.9193$. Since $F_F > F_{0.05}(6,42)$ and $F_F > F_{0.10}(6,42)$, so we reject the null-hypothesis and draw a conclusion that the differences between all compared algorithms on multiple datasets are significant. 2) For multilabel case and multiview multilabel case, we draw a same conclusion.

Then, we carry out Nemenyi test for pairwise comparisons. 1) For multiview case, since $N = 8$ and $k = 7$, thus critical value at $q_{0.05}$ is 2.9490 and corresponding critical difference (CD) is $CD_{0.05} = q_{0.05}\sqrt{(k \cdot (k+1)/6 \cdot N)} = 3.1853$ while the one at $q_{0.10}$ is 2.6930 and corresponding CD is $CD_{0.10} = q_{0.10}\sqrt{(k \cdot (k+1)/6 \cdot N)} = 2.9088$. Since under the case of $CD_{0.05}$ and $CD_{0.10}$, rank differences between MuSC-MVML and S-MVSC are smaller than $CD_{0.05}$ and $CD_{0.10}$, so we say on these cases, the performance of MuSC-MVML is not significant better than the one of S-MVSC. Then, for other cases, since the corresponding rank differences are larger than $CD_{0.05}$ and $CD_{0.10}$, so MuSC-MVML outperforms these algorithms significantly under those cases. 2) For multilabel and multiview multilabel cases, since rank differences between MuSC-MVML and others are larger than corresponding $CD_{0.05}$ and $CD_{0.10}$, thus MuSC-MVML outperforms these algorithms significantly. Especially, for ELSMML and TFMDD which are two algorithms with self-adaptive correlations adopted, their performances are significant worse than our MuSC-MVML and this also validates that self-adaptive approach in our proposed algorithm is significantly differ from and enhances some existing algorithms.

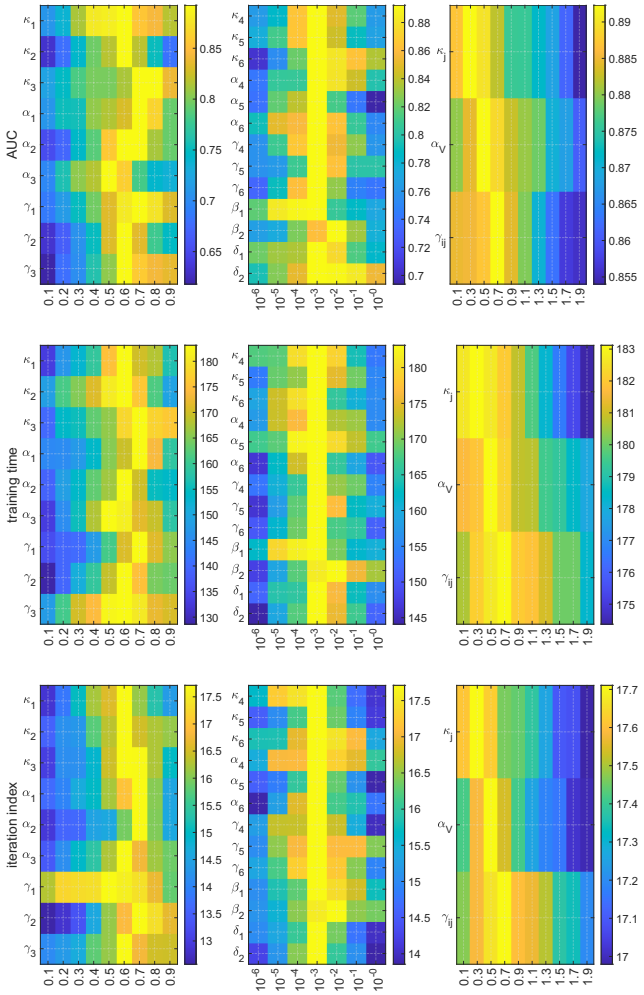

Fig. 9. Average influence of parameter values for MuSC-MVML.

In general, our MuSC-MVML performs best as demonstrated by statistical tests, especially for multilabel case and multiview multilabel case.

D. Influence of Parameters

For each algorithm, different parameter values lead to diverse average AUC, training time, and convergence. After carry out the experiments, the feasible settings for parameters about compared algorithms can be found in Table VII. Then, in terms of MuSC-MVML, Fig. 9 shows the average performances vary with the parameters on all used datasets and from this figure, it is found that if parameters to classical multiple correlations-based model parts are set as 0.6 or 0.7 and parameters to related self-adaptive terms are set as 10^{-3}, we can produce better AUC while such a setting leads a higher training time and iteration indexes. Moreover, other parameters have little effects on the performance. Thus, to the study, after optimal parameters selected, we set the feasible settings as below. We set κ_1–κ_3, α_1–α_3, and γ_1–γ_3 be 0.6, κ_j, α_c, and γ_{ij} be 0.9, and others be 10^{-3}.

E. Ablation Study

Framework of MuSC-MVML (i.e., (20)) consists of associated, feature-oriented, self-adaptive, label-oriented,

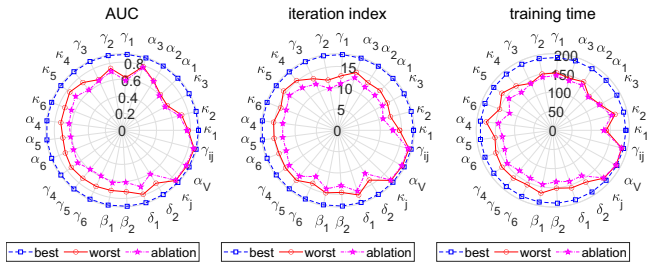

Fig. 10. Ablation study for MuSC-MVML where only one term removed is considered here.

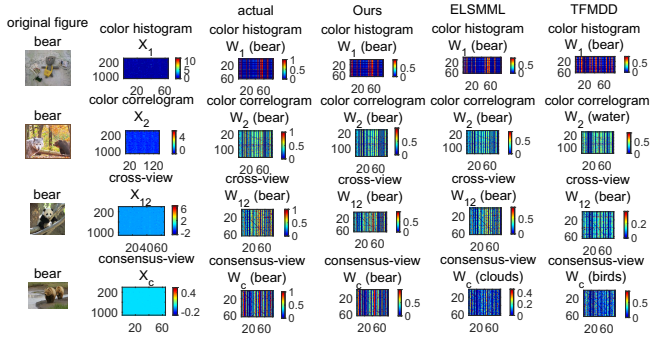

Fig. 11. Visualization for the example about NUS-WIDE.

regularization terms, etc. Each term can be represented by consensus-view, within-view, and cross-view representations. These terms correspond to multiple parameters and different terms have different influence on the performances of MuSC-MVML. So, in order to validate the influence of these terms, we carry out ablation study further. Namely, we set a parameter be 0 which equals to removing the corresponding term and see the average vary of performances about AUC, training time, and convergence on all used datasets. The results are given in Fig. 10, and in this figure, curve "best"/"worst" stands for the best/worst performances when we adjust the parameter values while curve "ablation" stands for the performances when we set the parameter values be 0, namely removing the corresponding terms. According to this figure, it can be seen that removing self-adaptive terms brings a greater reduction to the performances and this indicates that considering the laws of self-adaptive change for multiple correlation information expressed in different representations can improve the ability of algorithms to process multiview multilabel datasets effectively.

F. Visualization Experiment on Correlation Expression

In order to validate the ability of MuSC-MVML to express the correlations, we adopt class "bear" in NUS-WIDE for demonstration (see Fig. 11). For this class, we select four original pictures in random for display. Then, as we know, each instance in NUS-WIDE is related with 6 views. These six views are color histogram $(64 - D)$, color correlogram $(144 - D)$, edge direction histogram $(73 - D)$, wavelet texture $(128 - D)$, block-wise color moments extracted over 5×5 fixed grid partitions $(225 - D)$, and bag of words based on SIFT descriptions $(500 - D)$. Then, for the adopted class, we

select the view color histogram and the view color correlogram for representation and give their corresponding information which are denoted as X_1, X_2, X_{12}, and X_c. Then, take feature-label correlation matrices W_1, W_2, W_{12}, and W_c for example, we demonstrate the actual results and the corresponding optimal results optimized by ours and compared two multiview multilabel algorithms ELSMML and TFMDD since they both adopt self-adaptive correlations. For each optimal feature-label correlation matrix, we can use it to update the corresponding label matrix and the predicted label is given in the parentheses next to the correlation matrix. According to Fig. 11, we can see that MuSC-MVML can express the correlations well and predict the labels more accurate while for other compared algorithms, the optimized correlations have a great differences to the actual ones and the predicted results are terrible.

VI. Further Discussion

A. Why We Use Alternating Optimization Approach

In this study, we adopt three-step updating strategy for the solution of our model and in this strategy, the alternating optimization approach is the main segment and the core idea of alternating optimization approach is the alternating direction method of multipliers (ADMMs) which is originally proposed in the mid-1970s [28]. As we know, ADMM is intended to blend the decomposability of dual ascent with the superior convergence properties of the method of multipliers and it is well suited to distributed convex optimization, and in particular to large-scale problems arising in statistics, machine learning, and related areas. Thus, on the base of ADMM, alternating optimization approach demonstrates its superiority to solve practical problems [29]. Indeed, alternating optimization approach has been widely used in diverse practical scenarios. For example, Wang et al. [30] made full use of the alternating optimization algorithm with proved convergence to efficiently optimize some models; Li et al. [31] adopted the alternating optimization approach to solve a notconvex problem and the model can be applied to handwritten datasets recognition problem and even the incomplete clustering task. Chen et al. [32] also used this approach to train a discriminative sparse representation learning model. This model can explore complementary and consistent information by integrating the sparse regularization item and a consensus regularization item, respectively, and then it can be feasible in diverse practical scenarios, including document translation, news article, etc.

Since many scholars have validated the effectiveness of alternating optimization approach and the model solved by this approach still can be applied into diverse practical scenarios, thus, in this study, we also use alternating optimization approach. Moreover, since the datasets used in our experiments are also from different practical scenarios (see Table VI), thus this can also validate that our MuSC-MVML can be effectively implemented in diverse practical scenarios.

B. How to Process Incomplete Datasets With Noise

With the arrival of big data era, traditional sampling equipments have no ability to capture all data information and some data maybe exist noisy or missing values. This leads to an incomplete and noisy problem which also affect the performances of MuSC-MVML. In terms of this problem, there are also some valuable studies are developed. For example, on the base of the study about [33], Wen et al. [34] further integrated view-specific deep feature extraction network, weighted representation fusion module, classification module, and view-specific deep decoder network simultaneously to process data with incomplete labels and missing views. Their model can effectively reduce the negative influence caused by incomplete labels and views and sufficiently explore the available data and label information to obtain the most discriminative feature extractor and classifier; Liu et al. [35] developed a transformer-based incomplete multiview multilabel learning framework which including two transformer-style based modules for cross-view features aggregation and multilabel classification, respectively. Their study can be adaptable to arbitrary multiview and multilabel data; Xu et al. [36] developed a structured low-rank matrix recovery method to effectively remove view discrepancy and improve discriminancy through the recovery of the structured low-rank matrix. Moreover, their proposed method can handle any zero-mode noise variable that contains a wide range of noise.

These above mentioned studies inspire us to modify the model of MuSC-MVML and solve an incomplete and noisy problem simultaneously. Namely, we can add an error term E to model noise [36] and E is related with the feature matrix X by the following (54). Here, P is a common mapping function shared by all views to project X onto a low-dimension subspace, where the discrepancy among views can be reduced. Z is a low-rank representation of PX with respect to an complete dictionary A and ϵ is a tradeoff parameter, $||Z||_\star$ is the nuclear norm about Z. $||E||_\tau$ is a regularization term determined by the noise type, such as $||E||_1$ for random noise, $||E||_F$ for Gaussian noise, and $||E||_{2,1}$ for outliers. Indeed, since E includes the information about noise and A includes the complete information about the features, thus this formulation can be treated as a solution to tackle the incomplete problem with noise

$$\begin{aligned} \min_{Z,E,P} \quad & ||Z||_\star + \epsilon||E||_\tau \\ \text{s.t.} \quad & PX = AZ + E. \end{aligned} \tag{54}$$

Specifically to MuSC-MVML, we take within-view representations and random noise as the example and for each X_j, we define a corresponding A_j as the complete dictionary. Then, we let E_j be the error term to include the noisy information, Z_j be the low-rank representation of PX_j and the following formulation can be added into the model of MuSC-MVML and the new model is named as IMuSC-MVML

$$\begin{aligned} \min_{Z_j,E_j,P} \quad & \left|\left|Z_j\right|\right|_\star + \epsilon\left|\left|E_j\right|\right|_1 \\ \text{s.t.} \quad & PX_j = A_jZ_j + E_j. \end{aligned} \tag{55}$$

The optimization of IMuSC-MVML can also be referred to Section IV-B and in order to validate its performances, we randomly remove some information and add some noise about datasets Mfeat, Arts, and NUS-WIDE and then use IMuSC-MVML and MuSC-MVML for comparison (see Table VIII).

TABLE VIII
COMPARISON ABOUT IMUSC-MVML AND MUSC-MVML ON THREE CLASSICAL DATASETS. THE RESULTS ARE GIVEN IN AVERAGE

IMuSC-MVML	AUC	std. for AUC	training time	iteration
Mfeat	0.7931	0.0172	6.42	20
Arts	0.9532	0.0121	61.32	17
NUS-WIDE	0.8361	0.0092	11.03	19
MuSC-MVML	AUC	std. for AUC	training time	iteration
Mfeat	0.7611	0.0245	5.93	18
Arts	0.8855	0.0140	54.87	16
NUS-WIDE	0.7573	0.0312	9.41	17

According to the results, we find that IMuSC-MVML achieves a better average AUC than MuSC-MVML while the training time and iterations add a little much due to it should update more terms. Thus, this indicates that we can modify the model of MuSC-MVML to process an incomplete and noisy problem. Moreover, much training time and more iterations of IMuSC-MVML inspire us to research further in the future.

VII. CONCLUSION AND FUTURE WORK

There are many existing learning algorithms are developed to process multiview multilabel, multilabel, and multiview data. Some of them are developed on the basis of data characteristics and some are designed on the basis of data correlations. But those algorithms always exist a main drawback that they cannot express correlations in multiple representations self-adaptively and relative accurately. This drawback makes these algorithms hard to reflect the influence of correlations in different representations to the performance accurately and causes that these algorithms have not an good ability to process different kinds of data simultaneously well.

In order to overcome this drawback, on the base of classical multiple correlations-based model, this study explores some laws of self-adaptive change for correlations among different instances, features, labels in consensus-view, within-view, and cross-view representations and develops a MuSC-MVML. Experiments completed on several benchmark datasets demonstrate the superiority of MuSC-MVML and some conclusions are addressed. 1) MuSC-MVML outperforms some classical compared multiview, multilabel, and multiview multilabel algorithms in statistical in terms of AUC and its performance is also stable; 2) although some laws of self-adaptive change for correlations expressed in different representations are introduced into the model and this makes the model be more complexity, the computational cost of MuSC-MVML is still moderate and will not be increased too much. Moreover, on most datasets, MuSC-MVML has a relatively fast convergence; 3) introducing these laws can improve the ability of algorithms to process multiview multilabel datasets effectively and our MuSC-MVML can express correlations in multiple representations and reflect influence of these correlations on the performance much better.

Furthermore, we also explained that why we use alternating optimization strategy to optimize the model of MuSC-MVML and provide some suggestions that how to modify the model of MuSC-MVML so as to process the incomplete multiview multilabel datasets with noise. While, how to reduce the training time and iterations still is an open problem and inspire us to research more feasible algorithms to process complicated multiview multilabel datasets and solve this open problem in the future.

REFERENCES

[1] S. Sun and Q. Zhang, "Multiple-view multiple-learner semi-supervised learning," *Neural Process. Lett.*, vol. 34, pp. 229–240, Aug. 2011.

[2] C. Gong, "Exploring commonality and individuality for multimodal curriculum learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1926–1933.

[3] F. Nie, J. Li, and X. Li, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1881–1887.

[4] Z. Hu, F. Nie, W. Chang, S. Hao, R. Wang, and X. Li, "Multiview spectral clustering via sparse graph learning," *Neurocomputing*, vol. 384, pp. 1–10, Apr. 2020.

[5] W. Weng, Y. Lin, S. Wu, Y. Li, and Y. Kang, "Multilabel learning based on label-specific features and local pairwise label correlation," *Neurocomputing*, vol. 273, pp. 385–394, Jan. 2018.

[6] Y. Zhu, J. T. Kwok, and Z.-H. Zhou, "Multilabel learning with global and local label correlation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1081–1094, Jun. 2018.

[7] M. Liu, Y. Luo, D. Tao, C. Xu, and Y. Wen, "Low-rank multiview learning in matrix completion for multilabel image classification," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2778–2784.

[8] Z. He, C. Chen, J. Bu, P. Li, and D. Cai, "Multiview based multilabel propagation for image annotation," *Neurocomputing*, vol. 168, pp. 853–860, Nov. 2015.

[9] B. Qian, X. Wang, J. Ye, and I. Davidson, "A reconstruction error based framework for multilabel and multiview learning," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 594–607, Mar. 2015.

[10] C. Zhang, Z. Yu, Q. Hu, P. Zhu, X. Liu, and X. Wang, "Latent semantic aware multiview multilabel classification," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4414–4421.

[11] C. Zhu, D. Miao, Z. Wang, R. Zhou, L. Wei, and X. Zhang, "Global and local multiview multilabel learning," *Neurocomputing*, vol. 371, pp. 67–77, Jan. 2020.

[12] S. E. Hajjar, F. Dornaika, and F. Abdallah, "One-step multiview spectral clustering with cluster label correlation graph," *Inf. Sci.*, vol. 592, pp. 97–111, May 2022.

[13] Y. Hu, X. Fang, P. Kang, Y. Chen, Y. Fang, and S. Xie, "Dual noise elimination and dynamic label correlation guided partial multilabel learning," *IEEE Trans. Multimedia*, vol. 26, pp. 5641–5656, 2024.

[14] J. Li, P. Kang, W. Sun, and Z. Jiang, "Local residual preserving non-negative matrix factorization for multiview clustering," *Neurocomputing*, vol. 600, Oct. 2024, Art. no. 128054, doi: 10.1016/j.neucom.2024.128054.

[15] W. Dong and S. Sun, "Partial multiview representation learning with cross-view generation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 17239–17253, Dec. 2024, doi: 10.1109/TNNLS.2023.3300977.

[16] Q. Zhong, G. Lyu, and Z. Yang, "Align while fusion: A generalized nonaligned multiview multilabel classification method," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 24, 2024, doi: 10.1109/TNNLS.2024.3387577.

[17] B.-Y. Liu, L. Huang, C.-D. Wang, J.-H. Lai, and P. S. Yu, "Multiview clustering via proximity learning in latent representation space," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 2, pp. 973–986, Feb. 2023.

[18] C. Zhang et al., "Generalized latent multiview subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Jan. 2020.

[19] J. Ma, W. Kou, M. Lin, C. C. M. Cho, and B. Chiu, "Multiclass and multilabel classifications by consensus and complementarity-based multiview latent space projection," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 3, pp. 1705–1718, Mar. 2024.

[20] Q. Tan, G. Yu, J. Wang, C. Domeniconi, and X. Zhang, "Individuality- and commonality-based multiview multilabel learning," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1716–1727, Mar. 2021.

[21] B. Liu et al., "Multiview multilabel learning with high-order label correlation," *Inf. Sci.*, vol. 624, pp. 165–184, May 2023.

[22] Z. Wang and Y. Xu, "A two-stage multiview partial multilabel learning for enhanced disambiguation," *Knowl. Based Syst.*, vol. 293, Jun. 2024, Art. no. 111680.

[23] B.-Y. Liu, L. Huang, C.-D. Wang, S. Fan, and P. S. Yu, "Adaptively weighted multiview proximity learning for clustering," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1571–1585, Mar. 2021.
[24] Z. Li, C. Tang, X. Liu, X. Zheng, W. Zhang, and E. Zhu, "Consensus graph learning for multiview clustering," *IEEE Trans. Multimedia*, vol. 24, pp. 2461–2472, May 2021.
[25] Y. Cong, J. Liu, B. Fan, P. Zeng, H. Yu, and J. Luo, "Online similarity learning for big data with overfitting," *IEEE Trans. Big Data*, vol. 4, no. 1, pp. 78–89, Mar. 2018.
[26] S. Sun, Z. Dong, and J. Zhao, "Conditional random fields for multiview sequential data modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1242–1253, Mar. 2022.
[27] J. Demsar, "Statistical comparisons of classifiers over multiple datasets," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 1–30, 2006.
[28] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximations," *Comput. Math. Appl.*, vol. 2, no. 1, pp. 17–40, 1976.
[29] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends® Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
[30] S. Wang et al., "Multiview clustering via late fusion alignment maximization," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3778–3784.
[31] M. Li, S. Wang, X. Liu, and S. Liu, "Parameter-free and scalable incomplete multiview clustering with prototype graph," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 300–310, Jan. 2024.
[32] J. Chen, S. Yang, X. Peng, D. Peng, and Z. Wang, "Augmented sparse representation for incomplete multiview clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 3, pp. 4058–4071, Mar. 2024.
[33] J. Wen et al., "Adaptive graph completion based incomplete multiview clustering," *IEEE Trans. Multimedia*, vol. 23, pp. 2493–2504, Aug. 2020.
[34] J. Wen et al., "Deep double incomplete multiview multilabel learning with incomplete labels and missing views," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 11396–11408, Aug. 2024, doi: 10.1109/TNNLS.2023.3260349.
[35] C. Liu, J. Wen, X. Luo, and Y. Xu, "Incomplete multiview multilabel learning via label-guided masked view-and category-aware transformers," in *Proc. 37th AAAI Conf. Artif. Intell.*, 2023, pp. 8816–8824.
[36] J. Xu et al., "Modal-regression-based structured low-rank matrix recovery for multiview learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1204–1216, Mar. 2021.

Changming Zhu received the Ph.D. degree from the School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China, in 2015.

He is currently a professor with the College of Information Engineering, Shanghai Maritime University, Shanghai. His research interesting include image process and multiview learning.

Yimin Yan received the B.S. degree from the School of Artificial Intelligence and Information Technology, Nanjing University of Chinese Medicine, Nanjing, China, in 2020. She is currently pursuing the postgraduate degree with the College of Information Engineering, Shanghai Maritime University, Shanghai, China.

Her research interesting include pattern recognition and multiview learning.

Duoqian Miao received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1997.

He is currently a Professor with the School of Electronics and Information Engineering, Tongji University, Shanghai, China. His research interests include soft computing, rough sets, and machine learning.

Yilin Dong received the Ph.D. degree from the School of Automation, Southeast University, Nanjing, China, in 2020. He is currently with the College of Information Engineering, Shanghai Maritime University, Shanghai, China.

His research interests include multiview learning and pattern recognition.

Witold Pedrycz (Life Fellow, IEEE) received the M.Sc., D.Sc., and Ph.D. degrees from the Silesian University of Technology, Gliwice, Poland, in 1978, 1980, and 1984, respectively.

He is a Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also with Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland. He has published numerous papers in the above areas. He is extensively involved in editorial activities. His main research interests include computational intelligence, fuzzy modeling and granular computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and software engineering.

Prof. Pedrycz is an Editor-in-Chief of *Information Sciences* and associate editor of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS. He served as an Associate Editor for IEEE TRANSACTIONS ON FUZZY SYSTEMS. He is a member of a number of editorial boards of other international journals.