Adaptive Structural Co-regularization for Unsupervised Multi-view Feature Selection

Tsung-Yu Hsieh$^1$, Yiwei Sun$^1$, Suhang Wang$^2$, Vasant Honavar$^{1,2}$

$^1$Department of Computer Science and Engineering  
$^2$College of Information Sciences and Technology

The Pennsylvania State University, University Park, USA

{tuh45,yus162,swz494,vuh14}@psu.edu

Abstract—With the advent of big data, there is an urgent need for methods and tools for integrative analyses of multi-modal or multi-view data. Of particular interest are unsupervised methods for parsimonious selection of non-redundant, complementary, and information-rich features from multi-view data. We introduce Adaptive Structural Co-regularization Algorithm (ASCRA) for unsupervised multi-view feature selection. ASCRA jointly optimizes the embeddings of the different views so as to maximize their agreement with a consensus embedding which aims to simultaneously recover the latent cluster structure in the multi-view data while accounting for correlations between views. ASCRA uses the consensus embedding to guide efficient selection of features that preserve the latent cluster structure of the multi-view data. We establish ASCRA’s convergence properties and analyze its computational complexity. The results of our experiments using several real-world and synthetic data sets suggest that ASCRA outperforms or is competitive with state-of-the-art unsupervised multi-view feature selection methods.

Index Terms—Multi-view Learning, Feature Selection, Unsupervised Learning

I. INTRODUCTION

Motivation. Modern big data applications in many areas, including life sciences, health sciences, brain sciences, cognitive and behavioral sciences, environmental sciences, climate sciences, and security and surveillance, among others, call for effective methods for integrative analyses of multi-view data [1]–[4] where each view typically corresponds to a sensing modality. For example, surveillance activity may seek to identify a person using an image of the person’s face, his or her fingerprint, handwriting, and social media activity; Classification of web pages may make use of text on the page, images, and hyperlinks that link into and out of the page; An image can be described by different sets of visual feature descriptors [5]; EEG-based brain-computer interface systems can make use of features extracted from different brain regions and different frequency bands [6]; Cancer subtyping and prognosis can benefit from integrative analyses of multi-omics data, e.g., somatic mutation, copy number alteration, DNA methylation, miRNA, gene and protein expression [7]. Such applications often benefit from multi-view methods that effectively take advantage of the complementary information from multiple views [8], [9], including in particular, methods for multi-view feature selection [10], [11].

Multi-view feature selection. Effective methods for multi-view feature selection have to address several challenges including: i) the curse of dimensionality, i.e., the number of features being very large compared to the number of samples; ii) the differences in scales of measurement across different views; iii) the differences in data distribution across different views; (iv) the differences in the feature spaces associated with the different views; and (v) complex correlations among features within and across views.

Furthermore, the lack of availability of class labels for most or all of the data samples further complicates feature selection [12], [13]. In the supervised setting [11], [14], feature selection typically entails identifying a subset of features that suffice to preserve the information needed to recover the class labels. In the unsupervised setting, since no class labels are available, it is clear that such an approach is simply inapplicable. Hence, there is an urgent need for effective unsupervised methods for parsimonious selection of non-redundant, complementary, and information-rich features across the different views.

Existing approaches to unsupervised multi-view feature selection (UMVFS) fall into one of several distinct classes: (i) Multi-view concatenation based methods, e.g., [15], which first concatenate the different views of the data into a single-view before applying one of several existing unsupervised single view feature selection methods. Such methods fail to account for the differences in the feature spaces associated with the different views or the complementarity of information provided by the different views. Moreover, they exacerbate the curse of dimensionality and the computational complexity of feature selection. (ii) Multi-view integration based methods, which aim to account for interactions among the different views. Such methods, e.g., adaptive multi-view feature selection (AMFS) [16], multi-view feature selection (MVFS) [17], and adaptive unsupervised multi-view feature selection (AUMFS) [18], form a consensus representation of similarity of the multi-view data samples, typically using linear combination of similarity structures derived from each of the different views. Variants of such methods, e.g., adaptive collaborative similarity learning (ACSL) [19] and adaptive similarity and view weight (ASVW) [20], learn the consensus representation adaptively, e.g., consensus similarity matrix or consensus embedding, in order to maximally preserves the pairwise similarity of multi-view data samples (and to an extent, accommodates differences in scale, distributions, and
feature spaces of the different views). However, existing multi-view integration based methods cannot accommodate more complex non-linear interactions across views.

Fig. 1. ASCRA jointly optimizes the embeddings of different views by adaptive structural co-regularization to maximize their agreement with a consensus embedding that aims to simultaneously recover the latent cluster structure in the multi-view data while accounting for correlations between views. Sparse regression is exploited to efficiently select features across the different views that preserve the latent cluster structure.

Key Contributions. We introduce the Adaptive Structural Co-Regularization Algorithm (ASCR A) for unsupervised multi-view feature selection. ASCRA jointly optimizes the embeddings of all of the views so as to maximize their agreement with a consensus embedding that simultaneously recovers the latent cluster structure in the multi-view data and efficiently selects features across the different views that preserve the latent cluster structure. The embeddings are learned using structural co-regularization to maximize their agreement with the consensus embedding while enforcing additional constraints on the consensus embedding to recover the latent cluster structure. The embeddings are learned using structural co-regularization to maximize their agreement with the consensus embedding while enforcing additional constraints on the consensus embedding to recover the latent cluster structure. The coefficients of the projection are used to automatically select the features that maximally preserve multi-view cluster structure. We establish the convergence properties of ASCRA and analyze its computational complexity. We compare ASCRA with other UMVFS methods on synthetic data wherein the data distributions differ across the views; and on several real-world data sets. The results show that ASCRA is competitive with or outperforms other UMVFS methods.

Organization. The rest of the paper is organized as follows: Section II, reviews existing methods for UMVFS and their limitations. Section III and IV present the proposed method ASCRA and show how the resulting optimization problem can be solved. Section V, establishes the convergence of ASCRA and analyzes its computational complexity. Section VI describes the setup and results of empirical evaluation. Section VII concludes with a summary of the paper.

II. RELATED WORK

We briefly summarize related work on unsupervised single- and multi-view feature selection and contrast it with ASCRA.
III. Multi-view Feature Selection via Adaptive Structural Co-regularization

In this section, we introduce our approach to the unsupervised multi-view feature selection problem.

Problem Definition Assume that we are given a data set consisting of \( n \) multi-view samples, where each sample is composed of \( V \) views, with view \( i \) containing \( d_i \) features. We denote such a data set by \( X = \{ X_1, \ldots, X_V \} \) where \( X_i \in \mathbb{R}^{n \times d_i} \). The goal is to select a subset of \( d \) features from the available set of \( \sum_{i=1}^{V} d_i \) features across the \( V \) views.

Overview of ASCRA An overview of ASCRA is depicted in Fig. 1. The high level idea of ASCRA is to learn a cluster structure of the multi-view data in a latent space, and use cluster membership as surrogate labels (that play the same role as the class labels in supervised feature selection) to guide feature selection. ASCRA jointly optimizes the embeddings of all of the views as well as a consensus embedding using adaptive structural co-regularization to simultaneously recover the latent cluster structure in the multi-view data while accounting for correlations between different views. In addition, ASCRA incorporates a cluster membership constraint that allows the consensus embedding to indicate cluster membership. It learns a sparse regression model that projects multi-view data to the latent space induced by the consensus embedding and the regression coefficients are used to select the features. ASCRA unifies the learning of view-specific and consensus embeddings with sparse regression to obtain an effective unsupervised multi-view feature selection algorithm.

A. Learning Consensus Embedding

To effectively address the challenges of unsupervised multi-view learning, ASCRA attempts to jointly learn view-specific embeddings and a consensus embedding which can act as a surrogate for class label to guide feature selection.

Let \( Y_1 \cdots Y_v \) denote the \( V \) view-specific embeddings. Assume a consensus embedding \( Y_c \in \mathbb{R}^{n \times s} \), where \( s \) is the dimension of the embedding. Thus, \( Y_c[i] \), the \( i \)-th row of \( Y_c \), represents the \( i \)-th data sample in the consensus embedding space. Because we seek a consensus embedding that reflects the latent clusters in the multi-view data, we constrain the consensus embedding such that for any given data sample, for each \( i \), only one of the elements of \( Y_c[i] \) is 1 and all others are 0. Hence \( Y_c \) is a matrix where each row can be interpreted as an indicator vector denoting cluster membership of the corresponding multi-view data sample. Specifically, \( Y_c \in \text{Ind} \), where \( \text{Ind} = \{ Y \in \{0,1\}^{n \times s} | Y1 = 1 \} \).

We adopt a centroid-based co-regularization scheme to jointly optimize the view-specific embeddings and the consensus embedding, yielding the objective function:

\[
\arg\min_{Y_1, \ldots, Y_V, Y_c} \sum_{i=1}^{V} \text{tr}(Y_i^T L Y_i) + D(Y_i, Y_c) \ .
\]

(1)

Here, \( L \) is the graph Laplacian matrix [26] of the undirected weighted graph of pairwise similarities between data samples considering only the \( i \)-th view, and \( D \) is a function that quantifies the disagreement between two embeddings.

Since the embeddings from different views may differ in orientation, we propose to regularize the structural differences between the embeddings using a variant of co-regularization [24]. Specifically, we define:

\[
D(Y_i, Y_c) = \frac{K_{Y_i} - K_{Y_c}}{\|K_{Y_i}\|_F \|K_{Y_c}\|_F}^2 ,
\]

(2)

where \( K_{Y_i} \) is the kernel matrix constructed from data samples considering only the \( i \)-th view. Normalizing the kernel matrices by their norms ensures that each of the view-specific embedding and the consensus embedding are mapped to a common scale for comparison. In this paper, we use a linear kernel for constructing \( K \), i.e., \( K_{Y_i} = Y_i Y_i^T \), and thus, we have:

\[
D(Y_i, Y_c) = \frac{Y_i Y_i^T - Y_c Y_c^T}{\|Y_i Y_i^T\|_F \|Y_c Y_c^T\|_F}^2 \ = 2 - \frac{2}{\sqrt{sc}} \text{tr}(Y_i Y_i^T Y_c Y_c^T)
\]

(3)

where we denote \( s = \|Y_i Y_i^T\|_F^2 \) and \( c = \|Y_c Y_c^T\|_F^2 \). Note that the constraint \( Y_i Y_i^T = 1 \) implies that \( \|Y_i Y_i^T\|_F^2 = s \) is exactly equal to \( s \), the dimension of the embedding.

As shown by Nie et al. [27], the simple averaging across all views of the disagreement between embeddings of the different views and consensus embedding could be problematic when not all views are equally reliable. Hence, following [27], we incorporate an adaptive weighting of the views, yielding the following objective function:

\[
\arg\min_{Y_1, \ldots, Y_V, Y_c} \sum_{i=1}^{V} \left[ \text{tr}(Y_i^T L_i Y_i) + \frac{2}{p_i} \left( 1 - \frac{\text{tr}(Y_i Y_i^T Y_c Y_c^T)}{\sqrt{sc}} \right) \right]
\]

subject to \( Y_i Y_i^T = 1 \), \( Y_c \in \text{Ind} \), \( p_i \geq 0 \), \( \forall i \) and \( \sum_{i=1}^{V} p_i = 1 \).

B. Multi-view Feature Selection

The crucial idea behind our approach is that the consensus embedding constructed using the procedure summarized above recovers the latent cluster structure, or the surrogate cluster labels, of the multi-view data. Given a clustering of the multi-view data samples, we want to select an optimal subset of features to preserve the latent cluster structure recovered from the original data. To achieve this, we integrate the learning of the consensus embedding described above with feature selection. Specifically, we seek to learn projection matrices \( W_i, i = 1, \ldots, V \) such that the projected data matrices, \( X_i' = X_i W_i \), approximate the matrix \( Y_i \). We impose \( \ell_2,1 \) norm penalty on the projection matrices \( W_i \) to enforce row sparsity. Now we are ready to specify the optimization problem to be
solved by ASCRA, our proposed algorithm for unsupervised multi-view feature selection:

\[
\begin{align*}
\arg\min_{Y_{t}, Y_{s}, W_{t}} \sum_{i=1}^{V} \left( tr(Y_{t}^{T} L Y_{t}) + \frac{2}{p_{i}} \left( 1 - \frac{tr(Y_{i}^{T} Y_{t} Y_{t}^{T})}{\sqrt{SC}} \right) \right) + \alpha \left( \|X_{\omega} W_{t} - Y_{s}\|_{F}^{2} + \beta \|W_{i}\|_{2,1} \right) \\
\text{s.t. } Y_{t}^{T} Y_{t} = I, Y_{s} \in \text{Ind}, p_{i} \geq 0, \sum_{i=1}^{V} p_{i} = 1,
\end{align*}
\]

Upon convergence of the algorithm, the \(l_2\) norm of the rows of the projection matrices can be used to rank the features by their relative importance.

IV. OPTIMIZATION METHOD

The optimization problem to be solved for ASCRA, as specified by Eq.(5), is not convex with respect to all of the variables. Hence, we adopt an alternating iterative optimization strategy to solve the problem. That is, we fix some of the parameters, and optimize the rest, iterating until convergence.

1) Optimizing \(Y_{s}\): To update \(Y_{s}\), we fix the other variables except \(Y_{s}\). Then Eq.(5) reduces to

\[
\begin{align*}
\arg\min_{Y_{s} \in \text{Ind}} \sum_{i=1}^{V} \left[ \frac{2}{p_{i}} \left( 1 - \frac{tr(Y_{i}^{T} Y_{t} Y_{t}^{T})}{\sqrt{SC}} \right) \right] + \alpha \|X_{\omega} W_{t} - Y_{s}\|_{F}^{2} \\
\text{s.t. } Y_{t}^{T} Y_{t} = I, Y_{s} \in \text{Ind}, p_{i} \geq 0, \sum_{i=1}^{V} p_{i} = 1.
\end{align*}
\]

We note that

\[
\|X_{\omega} W_{t} - Y_{s}\|_{F}^{2} = \|X_{\omega} W_{t}\|_{F}^{2} + \|Y_{s}\|_{F}^{2} - 2\|X_{\omega} W_{t} Y_{s}\|_{F}^{2} + \|W_{t}\|_{F}^{2}
\]

wherein the first two terms on the right hand side are independent of \(Y_{s}\). Thus, solving the optimization problem specified by Eq.(6) is equivalent to maximizing

\[
\begin{align*}
f(Y_{s}) &= \sum_{i=1}^{V} \left[ \frac{2}{p_{i} \sqrt{SC}} \left( \frac{tr(Y_{i}^{T} Y_{t} Y_{t}^{T})}{p_{i}} - \frac{tr(Y_{i}^{T} Y_{s} Y_{s}^{T})}{c} \right) Y_{s} + \alpha X_{\omega} W_{t} \right].
\end{align*}
\]

The optimization problem in Eq.(8) is closely related to mixed-integer quadratic programming (MIQP) problem [28] which can be solved using locally linear approximation [29]. The gradient of \(f\) in Eq.(8) with respect to \(Y_{s}\) is given by

\[
\nabla f = \sum_{i=1}^{V} \left[ \frac{2}{p_{i} \sqrt{SC}} \left( \frac{tr(Y_{i}^{T} Y_{t} Y_{t}^{T})}{p_{i}} - \frac{tr(Y_{i}^{T} Y_{s} Y_{s}^{T})}{c} \right) Y_{s} + \alpha X_{\omega} W_{t} \right].
\]

Let \(\nabla f(Y_{s})\) denote the gradient evaluated at the \(t\)-th iteration. Then, a locally linear approximation of the quadratic function in Eq.(9), denoted as \(\hat{f}(Y_{s})\), is given by

\[
\begin{align*}
\hat{f}(Y_{s}) &= f(Y_{s}) + \nabla f(Y_{s})^{T}(Y_{s} - Y_{s}) \\
&= tr\left( \nabla f(Y_{s}) Y_{s} \right) + f(Y_{s}) - tr\left( \nabla f(Y_{s}) Y_{s} \right).
\end{align*}
\]

Note that in the right hand side of the second line of Eq.(10), all terms except the first term are independent of \(Y_{s}\). Thus, solving the optimization problem given by Eq.(8) reduces to:

\[
Y_{s} = \arg\max_{Y_{s} \in \text{Ind}} tr\left( Y_{s}^{T} \nabla f(Y_{s}) \right),
\]

and the closed-form solution of which is given by

\[
\forall i = 1, \cdots, n, \quad (Y_{s})_{ij} = \begin{cases} 1 & \text{if } j = \arg\max_{k} \left( \nabla f(Y_{s}) \right)_{ik} \quad (12) \\ 0 & \text{otherwise}. \end{cases}
\]

Following [29], we devise an iterative scheme to update \(Y_{s}\) which is summarized in Algorithm 1.

The physical intuition behind this optimization strategy can be understood as follows. First note that \(Y_{s}^{T} Y_{t}\) is the similarity matrix constructed from the view-specific embedding \(Y_{t}\) using a linear kernel. The columns of the product of the matrices \(Y_{s} Y_{t}^{T}\) and \(Y_{t}^{T} Y_{s}\) summarize the cluster membership of the data instances given the current cluster assignment. The clusters are updated based on cluster membership statistics across all of the views so as to assign each data sample to the “nearest” cluster defined by the consensus embedding.

2) Optimizing \(Y_{t}\): When we fix all variables except the view-specific embeddings \(Y_{t}, i = 1, \cdots, n\) in Eq.(5), we note that the \(Y_{s}\)s become independent of each other, allowing us to update each view independently of the others. Hence, optimizing the \(i\)th view reduces to solving the problem

\[
\arg\min_{Y_{t}} \text{tr}\left( Y_{t}^{T} \left( L_{i} - \frac{2 Y_{t}^{T} Y_{t}^{T}}{p_{i} \sqrt{SC}} \right) Y_{t} \right),
\]

which has a closed-form solution given by the eigen-decomposition on the matrix \(L_{i} - \frac{2 Y_{t}^{T} Y_{t}^{T}}{p_{i} \sqrt{SC}}\) [30]. The columns of \(Y_{t}\) are comprised of the eigenvectors corresponding to the \(s\) smallest eigenvalues. The matrix \(Y_{t} Y_{t}^{T}\) is the similarity matrix in the embedding space under the linear kernel. Note that the graph Laplacian is obtained by \(L_{i} = D_{i} - S_{i}\), where \(S_{i}\) is the original similarity matrix of view \(i\) and \(D_{i}\) is the corresponding degree matrix. Therefore, subtracting the view-specific graph Laplacian \(L_{i}\) by \(Y_{t} Y_{t}^{T}\) is equivalent to modifying the view-specific graph Laplacian with information supplied by the other views (through the consensus embedding) about the structure of the data. This update rule allows the consensus embedding to shape the view-specific embeddings.

3) Optimizing \(W_{t}\): When we fix all variables except the projection matrices \(W_{t}, i = 1, \cdots, n\) in Eq.(5) we note that \(W_{t}\)s are independent of each other, allowing us to independently optimize the projection matrix of each view. Hence, optimizing the projection matrix for the \(i\)th view reduces to solving the optimization problem given by

\[
\arg\min_{W_{t}} \|X_{\omega} W_{t} - Y_{t}\|_{F}^{2} + \beta \|W_{t}\|_{2,1}.
\]
To solve Eq.(14), the solution is found to be

$$ W_i = (X_i^T X_i + \beta U_i)^{-1} X_i^T Y_i $$

(15)

where $U_i \in \mathbb{R}^{d_i \times d_i}$ is a diagonal matrix whose elements are $\frac{1}{\sum_j |W_{ij}|}, j = 1, \ldots, d_i$, and $(W_{ij})$ is the $j$th row vector of $W_i$. Note that $U_i$ is dependent of $W_i$, and therefore we iteratively solve $W_i$ and update $U_i$ until convergence. We leave the exploration of other methods [31]–[33] to this optimization problem as a possible future direction.

4) Optimize $p_i$: To avoid notational clutter, let $\phi_i^2 = 2 - \sqrt{\lambda_i}$, and $tr(Y_i^T W_i Y_i)$. By fixing all variables except $p_i$, $i = 1, \ldots, n$ in the optimization problem specified by Eq.(5), optimizing $p_i$ reduces to solving

$$ \arg \min_{p_i \geq 0, \sum_i p_i = 1} \phi_i^2 / p_i, $$

(16)

which is solved by [27]:

$$ p_i = \phi_i / \sum_{j=1}^V \phi_j, $$

(17)

A. ASCRA: Putting it all together

Putting everything together, we obtain the ASCRA algorithm which is summarized in Algorithm 2. Given multi-view data matrices, the algorithm first computes view-specific similarity matrices and computes the corresponding view-specific embeddings in Line 2. In Line 3, the view-specific embeddings are aggregated to obtain the initial state of consensus embedding. From Line 4 to Line 9, the algorithm alternates between updating consensus embedding, view-specific embeddings, projection matrices, and adaptive view weights while fixing others until convergence. Upon termination, ASCRA evaluates the magnitude of the rows of the projection matrices and ranks them in descending order in Line 10. Relevant features are selected according to the order.

V. THEORETICAL ANALYSES

In this section, we prove the convergence of ASCRA and provide the time complexity of ASCRA.

A. Convergence of ASCRA

To establish the convergence of Algorithm 2, we need to show that each of the update rule in Algorithm 2 ensures the value of the objective function does not increase with $t$ (iteration). The proofs for the update rules in lines 6-8 of Algorithm 2 can be found in [19], [27], [34]. Hence, we limit our focus to the update rule in line 5. We first show that the update rule in Eq.(12) gives the optimal solution to Eq.(11). Note that given $Y_i \in \mathbb{R}^{n \times s}$ and $Q \in \mathbb{R}^{n \times s}$.

$$ tr(Y_i^T Q) = \sum_{i=1}^s (Y_i^T Q)_{i,i} = \sum_{i=1}^a (Q)_{i,j}, \text{where } (Y_i)_{i,j} = 1. $$

(18)

If we pick $j_1 = \arg \max_k (Q)_{i,k}, \forall i = 1, \ldots, n$, we have $(Q)_{i,j_1} \geq (Q)_{i,k},$ and $\sum_{i=1}^a (Q)_{i,j_1} \geq \sum_{i=1}^a (Q)_{i,k}, \forall i = 1, \ldots, n$ and $k = 1, \ldots, s$. Hence, if we assign $(Y_i)_{i,j} = 1, j = \arg \max_k (Q)_{i,k}, \forall i = 1, \ldots, n$, we have

$$ tr(Y_i^T Q) \geq tr(Y_i^T Q), \forall Y_i \in \mathbb{Ind}. $$

(19)

Secondly, since $Y_{n+1}$ is the maximizer of $\max_{j=1,\ldots,d} f(Y_{s}^{(t)})$, we have: if $f(Y_{s}^{(t+1)}) > f(Y_{s}^{(t)}), $ then the algorithm proceeds to the next iteration; and if $f(Y_{s}^{(t+1)}) \leq f(Y_{s}^{(t)}), $ then the criterion is violated. Hence, it must be the case that $\exists j < t$ such that $Y_{s}^{(t+1)} = \arg \max f(Y_{s}^{(j)})$. Consequently, $Y_{s}^{(t+1)} = Y_{s}^{(j+1)}$, which terminates the algorithm. It follows that Algorithm 1 is non-decreasing. Hence, together with the non-decreasing properties of other update steps, the objective function of ASCRA has to be non-decreasing over successive iterations.

B. Computational Complexity of ASCRA

At each update iteration, the ASCRA updates $Y_i$, and each of the $Y_i, W_i, p_i$. To update $Y_i$, it executes an iterative procedure that repeats Eq.(9-12). The complexity of updating $Y_i$ is dominated by the complexity of multiplication of the matrices involved and can be bounded by $O(k \cdot n^2 s)$, where $k$ is the number of iterations need for Algorithm 1 to converge ($k$ is less than 5 in all our numerical experiments). Solving for $Y_i$ requires matrix multiplication, which is of complexity $O(n^2 s)$, and an eigen decomposition to find top $s$ eigenvectors, which is also of complexity $O(n^2 s)$. To update $W_i$, it requires inverting a $d_i \times d_i$ matrix, which has complexity $O(d_i^3)$. Lastly, it takes $O(n^2 s)$ to update $p_i$, where the complexity comes from evaluating $\phi_i^2$. In summary, ASCRA’s computational complexity is $O(n^2 s + \text{max}_{d_i} d_i^3)$ for each update iteration.

On the other hand, ASCRA needs to store the values of consensus embedding, view-specific Laplacians, projection matrices (as well as the diagonal matrices $U_i$), and adaptive view weights during iterative update. The consensus embedding requires $O(ns)$ space, view specific embeddings take $O(V \cdot n^2)$, projection matrices yield $O(\sum_{i=1}^a d_i s)$, and...
adaptive view weights consume $O(V)$. In summary, ASCRA requires $O(V \cdot n^2 + \sum_{i=1}^{V} d_i s_i)$ space complexity.

VI. EXPERIMENTAL EVALUATION

In this section, we conduct experiments to evaluate the effectiveness of the proposed framework ASCRA.

A. Data Sets

The experiments are conducted on five real-world datasets and four simulated datasets. The purpose of using simulated data is to examine the conditions under which ASCRA outperforms other multi-view methods.

Real-World Data. The real-world data used here include:

- **Multi-omics** [35]: This is an ovarian cancer multi-omics data set which consists of 3 views, 19963 dimensional methylation feature, 216 dimensional protein feature, and 17673 dimensional gene expression feature, respectively. The data samples are classified into 2 categories, namely long term and short term survival.

- **Networks** [36]: These data sets consist of network node representations learned from a multi-view network. The original network data of Network1 comes from the publicly available Flickr data set and has 10 classes. We use 3 views among the total of 5 views. The original network data of Network2 comes from the Last.fm data set and has 11 classes. We use 5 of them in our experiments. The node representation of each view, which is 128 dimensional vector, is extracted independently using the Node2Vec [37] algorithm. We find the common set of nodes that are present in each view, which results in 1775 samples in Network1 and 496 samples in Network2.

- **Handwritten digits** [38]: This data set consists of 2000 samples of handwritten digits of 0 to 9. Every sample has 6 views, 76 dimensional Fourier coefficients of the character shapes, 216 dimensional profile correlations, 64 dimensional Karhunen-love coefficients, 240 dimensional pixel averages in $2 \times 3$ windows, 47 dimensional Zernike moment and 6 dimensional morphological features.

- **MSRC-v1** [39]: This data set consists of 210 images from 7 classes, and every image is described by 5 sets of features, 24 dimensional color moment, 576 dimensional HOG feature [40], 512 dimensional GIST feature [41], 256 dimensional LBP feature [42], and 254 dimensional CENT feature [43].

**Synthetic Data.** In order to fully understand ASCRA, we generated 4 synthetic multi-view data sets where the data distribution varies significantly across views, similar to [44], as shown in Figure 2. The compositions of the 4 synthetic multi-view data sets are summarized in Table I and their characteristics are described as follows: Sets 1, 3, and 4 exhibit different sample distributions in each view with clear cluster structures whereas Set 2 contains a significant amount of noise in one view.

B. Experimental Setup

To evaluate the effectiveness of these unsupervised feature selection algorithms, we first use each algorithm to select a subset of features. We then perform clustering on the selected features to evaluate the quality of the selected features. We run experiments with the number of features to be selected set to 20, 40, 60, 80, and 100. The selected subset of features is clustered using K-means algorithm with the number of clusters equal to the known number of classes represented in the data. For synthetic data, we only select two features for clustering.

** Compared Methods.** We compare ASCRA with several representative and state-of-the-art methods, which include 2 single-view methods, Laplacian score (Laplacian) [45] and spectral feature selection (SPEC) [15]; and 3 multi-view methods, AUMFS [18], ACSL [19], and ASVW [20].

- Laplacian [45]: It computes Laplacian score of each feature, which reflects the locality preserving power of a feature. Features are then ranked by Laplacian scores.
- SPEC [15]: It adopts spectral graph theory for selecting features.
- AUMFS [18]: AUMFS linearly combines view-specific Laplacian matrices and employs sparse regression for feature selection.
- ACSL [19]: It constructs collaborative similarity matrix and employs sparse regression for feature selection.
- ASVW [20]: It constructs consensus similarity matrix and employs sparse locality preserving projection.

Whenever an algorithm requires a similarity graph between data samples, we use the same set of view-specific similarity graphs. For algorithms that involve defining embeddings, we set the dimension of embeddings to the number of clusters.
For parameter tuning, we either follow the recommendations provided by the authors of each method, if available, or empirically tune them using grid search and report the best results. For experiment with synthetic data, we only compare the performance of multi-view methods.

**Evaluation Metrics.** We assess the performance of the feature selection methods in terms of their clustering performance by applying K-means algorithm on the selected features, and performance is evaluated by the normalized mutual information (NMI), clustering accuracy (ACC), and purity. To obtain robust estimates of performance, each experiment is repeated 20 times and the mean results and standard deviation are reported. The results are further examined by statistical test with confidence threshold set to 0.01.

**C. Experimental Results**

**Performance on Real-World Data.** The clustering performance of the different methods on real-world data, as measured by purity, Normalized Mutual Information (NMI), and accuracy, are summarized in Table II, Table III and Figure 3 respectively. The results suggest that ASCRA performs significantly better than other methods on Multi-omics data in terms of accuracy, except that ASVW significantly outperforms ASCRA when 60 features are selected. On Network1, ASCRA is competitive with (with no significant difference) ASVW and outperforms others in both NMI and accuracy. In terms of purity, ASVW outperforms ASCRA when 40 or 60 features are selected for clustering. On Network2, ASCRA outperforms all methods in all metrics. The only exception is that Laplacian and SPEC significantly outperforms ASCRA in terms of NMI when 20 features are selected. On handwritten digits, ASCRA has the best clustering performance in all metrics. On MSRC-v1, ASCRA achieves better or comparable performance than all other methods in terms of NMI and accuracy, and it is constantly among the best performers in terms of purity.

**Performance on Synthetic Data.** The results of our experiments with synthetic data are reported in Table IV. We find that ASCRA outperforms other methods especially when the data samples are tightly clustered (as in Sets 1, 3, and 4). We attribute ASCRA’s superior performance to its ability to handle differences in data distributions across the views. We observe that ASCRA selects almost the same subset of features from all simulated data sets used in our experiments, yielding nearly identical clustering performance on all data sets. It is suggesting that its performance is relatively unaffected by differences in data distributions across the views and the present of noise in some of the views.

We further analyze the effect of the cluster indicator constraint on learning the consensus embedding, i.e. $Y_i \in \text{Ind}$. In the experiment with synthetic data, we extract the consensus embedding from methods including AUMFS, ACSL, and ASCRA. ASVW is excluded since it only constructs consensus similarity matrix but not consensus embedding. We apply K-means algorithm on the consensus embeddings for clustering and evaluate the performance. Note that since the consensus embedding generated by ASCRA is already a cluster indicator, we directly evaluate cluster performance on the consensus embedding without applying any clustering algorithm. The results are reported in Table V. We can observe that the embeddings of AUMFS and ASCRA yield better clustering performance than ACSL. The superior performance of AUMFS and ASCRA can be attributed to the constraint that they impose on the consensus embedding. The AUFS requires that the entries of the embedding must be non-negative and the ASCRA imposes a stronger constraint that the embedding indicates cluster membership.
Parameter Sensitivity Analysis. To examine how the choice of the hyper-parameters, α and β, affect the performance of ASCRA, we conducted a grid search over the space spanned by $\alpha \in \{10^p : p = -4, -3, \ldots, 1\}$ and $\beta \in \{10^p : p = -3, -2, \ldots, 2\}$. For each parameter combination, we selected 100 features, employed K-means for clustering, and evaluated the clustering performance. Due to space constraints, we show the results of this experiment on only one of the real-world data sets, i.e., Network 1 data, in Figure 4. The results of our parameter sensitivity analysis (on Network dataset as well as other datasets for which results are omitted) show that the performance of the algorithm is relatively stable over a broad range of choices of the hyper-parameters $\alpha$ and $\beta$.

Empirical Computational Complexity. Recall that the complexity per iteration of the proposed algorithm is $O(n^2 s + \max\{d_i^2\})$. Recall that the worst-case runtime complexities of the methods compared in our experiments, the complexities are, AUMFS: $O(n^3 + (\sum d_i)^3)$, ACSL: $O(n^2 s + (\sum d_i)^3)$, ASVW: $O(\sum d_i \times \max\{d_i^2, n \times k \times s\})$, where $k$ is a user-specified parameter. The measured runtime per update iteration of each multi-view feature selection method is summarized in Table VI. To ensure fair comparison, all of the algorithms were executed on an identical hardware configuration and operating system. We observe that AUMFS requires longest runtime across all datasets. ASCRA takes much shorter time than ACGL and ASVW, when the dimensionality of the data is much larger than the number of samples, as in the case of Multi-omics and MSRC-v1 data. On the other hand, when the number of samples is much larger than the dimensionality of the data, the ASCRA requires slightly more time than ACGL and ASVW, as expected from a larger constant factor associated with the $n^2$ term.

### Table III: Clustering results (NMI±std) of different feature selection algorithms on real-world data. Boldface figure indicates best performing method, or methods (when there is no significant difference among the best performing methods).

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Features</th>
<th>Laplacian</th>
<th>SPEC</th>
<th>AUMFS</th>
<th>ACGL</th>
<th>ASVW</th>
<th>ASCRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-omics</td>
<td>40</td>
<td>0.9950</td>
<td>0.0031</td>
<td>0.0007</td>
<td>0.0007</td>
<td>0.0007</td>
<td>0.0031</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>0.9950</td>
<td>0.0031</td>
<td>0.0007</td>
<td>0.0007</td>
<td>0.0007</td>
<td>0.0031</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>0.9950</td>
<td>0.0031</td>
<td>0.0007</td>
<td>0.0007</td>
<td>0.0007</td>
<td>0.0031</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.9950</td>
<td>0.0031</td>
<td>0.0007</td>
<td>0.0007</td>
<td>0.0007</td>
<td>0.0031</td>
</tr>
</tbody>
</table>

### Table IV: Clustering performance on synthetic data.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Metric</th>
<th>AUMFS</th>
<th>ACGL</th>
<th>ASVW</th>
<th>ASCRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>NMI</td>
<td>0.9950</td>
<td>0.9950</td>
<td>0.9950</td>
<td>0.9950</td>
</tr>
<tr>
<td></td>
<td>Accuracy</td>
<td>0.9995</td>
<td>0.9950</td>
<td>0.9995</td>
<td>0.9995</td>
</tr>
</tbody>
</table>

### Table V: Clustering performance of consensus embedding on synthetic data.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Metric</th>
<th>AUMFS</th>
<th>ACGL</th>
<th>ASCRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>NMI</td>
<td>0.9470</td>
<td>0.9530</td>
<td>0.9470</td>
</tr>
<tr>
<td></td>
<td>Accuracy</td>
<td>0.9970</td>
<td>0.9970</td>
<td>0.9970</td>
</tr>
</tbody>
</table>

### Table VI: The actual running time, in seconds, per update iteration of each multi-view feature selection method.

<table>
<thead>
<tr>
<th>Method</th>
<th>AUMFS</th>
<th>ACGL</th>
<th>ASVW</th>
<th>ASCRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-omics</td>
<td>425.9</td>
<td>518.3</td>
<td>84.8</td>
<td>48.6</td>
</tr>
<tr>
<td>Network1</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
</tr>
<tr>
<td>Network2</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
</tr>
<tr>
<td>Handwritten</td>
<td>50.0</td>
<td>50.0</td>
<td>50.0</td>
<td>50.0</td>
</tr>
<tr>
<td>digits</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
</tr>
<tr>
<td>MSRC-v1</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
</tr>
</tbody>
</table>
VII. CONCLUSION

Modern big data applications across a variety of areas call for integrative analyses of multi-view data. Existing approaches to unsupervised multi-view feature selection fail to account for the differences in the feature spaces associated with the different views or the complementarity of correlations among the different views. The novel adaptive structural regularization based approach to unsupervised multi-view feature selection introduced in this paper overcomes some of the key limitations of the existing methods. It jointly optimizes the embeddings of the different views to produce a consensus embedding that recovers the latent cluster structure in the multi-view data, and then proceeds to select a subset of features that maximally preserves the cluster structure. We designed a suitable objective function when optimized instantiates the proposed approach to unsupervised multi-view feature selection. We provided a computationally efficient alternating iterative optimization based solution of the resulting optimization problem, yielding ASCRA. We established the convergence of ASCRA and analyzed its computational complexity. We reported results of our experiments with several real-world as well as simulated data sets that clearly demonstrate that ASCRA outperforms or is competitive with the state-of-the-art unsupervised multi-view feature selection methods. Our experiments also show the performance of ASCRA is robust across a broad range of its hyper-parameter settings.

Acknowledgements

This work was funded in part by grants from the NIH NCATS through the grant UL1 TR002014 and by the NSF through the grants 1518732, 1640834, and 1636795, the Penn State Center for Big Data Analytics and Discovery Informatics, the Edward Frymoyer Endowed Professorship in Information Sciences and Technology at Pennsylvania State University and the Sudha Murty Distinguished Visiting Chair in Neurocomputing and Data Science funded by the Pratiksha Trust at the Indian Institute of Science (both held by Vasant
REFERENCES


