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,szw494,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.

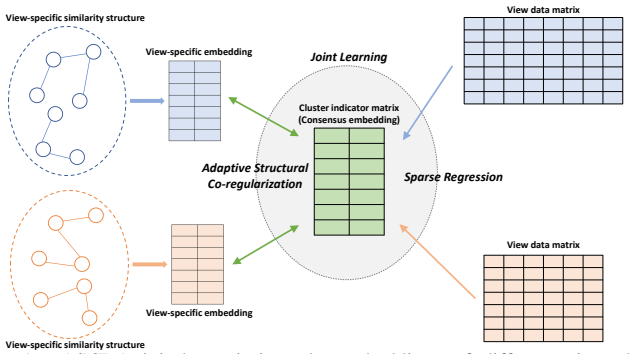


Fig. 1. ASCRA jointly optimizes the embeddings of different views by adaptive structural co-regularization to maximize their agreement with a *consensus embedding* that aims to simultaneously recover the latent cluster structure in the multi-view data while accounting for correlations between views. 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 (ASCRA) 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 in the multi-view data. An $l_{2,1}$-norm penalized sparse regression model is employed to project the multi-view data to the latent space induced by the consensus embedding. 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.

A. Multi-View concatenation based methods

The simplest approach to unsupervised multi-view feature selection involves concatenating the multiple views into a single view and then applying any of the single view unsupervised feature selection methods on the resulting single view data. For example, [15] introduced a general framework to unify a broad class of unsupervised (and supervised) feature selection based on ideas from spectral graph theory. Given a similarity measure for constructing matrix of pairwise similarities between data samples, the framework can be used to select features based on their ability to preserve similarity between the data samples in a lower-dimensional space induced by the Laplacian Eigenmap [21]–[23].

While it is straightforward to reduce the problem of multi-view feature selection to its single-view counterpart by simply concatenating the different views, such an approach fails to account for the differences in the feature spaces or the complementary information provided by the different views. Moreover this exacerbates the curse of dimensionality and the computational complexity of feature selection.

B. Multi-view integration based methods

Multi-view integration based feature selection methods 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 the multi-view data samples typically by a linear combination of similarity structures in spectral space [24], [25], e.g., graph Laplacian matrices, derived from each of the different views. An $l_{2,1}$-norm penalized robust sparse regression is used to map the data into clusters and $l_{2,1}$-norm induced row sparsity is then used to select relevant features that maximally preserve the cluster structure. Other multi-view integration based feature selection methods, e.g., adaptive collaborative similarity learning (ACSL) [19], unlike AUMFS which linearly combines the similarity structures from the different views, adaptively learns a consensus similarity graph. ACSL learns a sparse regression model that projects data from the different views to the consensus embedding, which is derived from the consensus similarity graph, and leverages the sparse model to perform feature selection. The Adaptive Similarity and View Weight (ASVW) method [20] learns the consensus similarity matrix adaptively from the multi-view data and adopts local preserving projection [22] with structure sparsity constraint to select important features. However, none of the preceding multi-view integration based feature selection methods can accommodate more complex, possibly non-linear, interactions across views. In addition, these methods do not constrain the embedding to yield cluster membership of multi-view data samples in the embedding space which can serve as surrogate labels for guiding the selection of features.

As we shall see below, ASCRA aims to overcome the limitations of the state-of-the-art methods for UMVFS.

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 $Y_* \in \mathbb{R}^{n \times s}$, where s is the dimension of the embedding. Thus, $Y_*[i]$, the ith row of Y_*, 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 $Y_*[i]$ is 1 and all others are 0. Hence Y_* is a matrix where each row can be interpreted as an indicator vector denoting cluster membership of the corresponding multi-view data sample. Specifically, $Y_* \in Ind$, where $Ind = \left\{ Y \in \{0,1\}^{n \times s} | Y\mathbf{1} = \mathbf{1} \right\}$.

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

$$\underset{Y_i^T Y_i = I, Y_* \in Ind}{\arg\min} \left[\sum_{i=1}^{V} tr(Y_i^T L_i Y_i) + D(Y_i, Y_*) \right]. \tag{1}$$

Here, L_i 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:

$$D(Y_i, Y_*) \triangleq \left\| \frac{K_{Y_i}}{\|K_{Y_i}\|_F} - \frac{K_{Y_*}}{\|K_{Y_*}\|_F} \right\|_F^2, \tag{2}$$

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:

$$\begin{aligned} D(Y_i, Y_*) &= \left\| \frac{Y_i Y_i^T}{\|Y_i Y_i^T\|_F} - \frac{Y_* Y_*^T}{\|Y_* Y_*^T\|_F} \right\|_F^2 \\ &= 2 - \frac{2}{\sqrt{sc}} tr\left(Y_i Y_i^T Y_* Y_*^T\right) \end{aligned} \tag{3}$$

where we denote $s = \|Y_i Y_i^T\|_F^2$ and $c = \|Y_* Y_*^T\|_F^2$. Note that the constraint $Y_i^T Y_i = I$ implies that $\|Y_i Y_i^T\|_F^2$ 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:

$$\underset{Y_*, Y_i, p_i}{\arg\min} \sum_{i=1}^{V} \left[tr(Y_i^T L_i Y_i) + \frac{2}{p_i} \left(1 - \frac{tr\left(Y_i Y_i^T Y_* Y_*^T\right)}{\sqrt{sc}} \right) \right] \tag{4}$$

subject to $Y_i^T Y_i = I$, $Y_* \in Ind$, $p_i \geq 0, \forall i$ 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_*. We impose $l_{2,1}$ 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:

$$\begin{aligned}\arg\min_{Y_*,Y_i,p_i,W_i}\sum_{i=1}^{V}\Bigg[&tr(Y_i^T L_i Y_i)+\frac{2}{p_i}\left(1-\frac{tr\left(Y_i Y_i^T Y_* Y_*^T\right)}{\sqrt{sc}}\right)\\&+\alpha\left(\|X_i W_i - Y_*\|_F^2+\beta\|W_i\|_{2,1}\right)\Bigg] \qquad (5)\\ s.t.\ \ &Y_i^T Y_i = I, Y_* \in Ind, p_i \geq 0, \sum_{i=1}^{V} p_i = 1.\end{aligned}$$

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_*: To update Y_*, we fix the other variables except Y_*. Then Eq.(5) reduces to

$$\arg\min_{Y_*\in Ind}\sum_{i=1}^{V}\left[\frac{2}{p_i}\left(1-\frac{tr\left(Y_i Y_i^T Y_* Y_*^T\right)}{\sqrt{sc}}\right)+\alpha\|X_i W_i - Y_*\|_F^2\right] \qquad (6)$$

We note that

$$\|X_i W_i - Y_*\|_F^2 = \|X_i W_i\|_F^2 + n - 2tr(Y_*^T X_i W_i) \qquad (7)$$

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

$$f(Y_*) = \sum_{i=1}^{V}\left[\frac{tr\left(Y_i Y_i^T Y_* Y_*^T\right)}{p_i\sqrt{sc}}+\alpha tr\left(Y_*^T X_i W_i\right)\right]. \qquad (8)$$

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_* is given by

$$\nabla f = \sum_{i=1}^{V}\left[\frac{2}{p_i\sqrt{sc}}\left(Y_i Y_i^T - \frac{tr\left(Y_i Y_i^T Y_* Y_*^T\right)}{c}Y_* Y_*^T\right)Y_* + \alpha X_i W_i\right]. \qquad (9)$$

Let $\nabla f(Y_*^{(t)})$ denote the gradient evaluated at the t-th iteration. Then, a locally linear approximation of the quadratic function in Eq.(8), denoted as $\hat{f}(Y_*^{(t)})$, is given by

$$\begin{aligned}\hat{f}(Y_*^{(t)}) &= f(Y_*^{(t)}) + tr\left(\nabla f(Y_*^{(t)})^T (Y_* - Y_*^{(t)})\right)\\ &= tr\left(\nabla f(Y_*^{(t)})^T Y_*\right) + f(Y_*^{(t)}) - tr\left(\nabla f(Y_*^{(t)})^T Y_*^{(t)}\right)\end{aligned} \qquad (10)$$

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

$$Y_* = \arg\max_{Y_*\in Ind} tr\left(Y_*^T \nabla f(Y_*^{(t)})\right), \qquad (11)$$

Algorithm 1 Update Y_*

Input: consensus embedding $Y_*^{(0)}$
Output: optimized consensus embedding Y_*

```
set t = 0.
repeat
    compute Y_*^{(t+1)} ∈ Ind, using Eqs.(9-12) , s.t Y_*^{(t+1)} = arg max_{i=1,...,t} f̂(Y_*^{(i)})
until Y_*^{(t+1)} = Y_*^{(i)}, ∃i ≤ t
return Y_*^{(t+1)}
```

and the closed-form solution of which is given by

$$\forall i = 1,\cdots,n,\quad (Y_*)_{i,j} = \begin{cases} 1 & \text{if } j = \arg\max_k \left(\nabla f(Y_*^{(t)})\right)_{i,k} \\ 0 & \text{otherwise.}\end{cases} \qquad (12)$$

Following [29], we devise an iterative scheme to update Y_* which is summarized in Algorithm 1.

The physical intuition behind this optimization strategy can be understood as follows. First note that $Y_l Y_l^T$ is the similarity matrix constructed from the view-specific embedding Y_l using a linear kernel. The columns of the product of the matrices $Y_l Y_l^T$ and $Y_*^{(t)}$ 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_i: When we fix all variables except the view-specific embeddings $Y_i, i = 1,\cdots,n$ in Eq.(5), we note that the Y_is become independent of each other, allowing us to update each view independently of the others. Hence, optimizing the ith view reduces to solving the problem

$$\arg\min_{Y_i^T Y_i = I} tr\left(Y_i^T\left[L_i - \frac{2Y_* Y_*^T}{p_i\sqrt{sc}}\right]Y_i\right), \qquad (13)$$

which has a closed-form solution given by the eigen-decomposition on the matrix $L_i - \frac{2}{p_i\sqrt{sc}}\cdot Y_* Y_*^T$ [30]. The columns of Y_i are comprised of the eigenvectors corresponding to the s smallest eigenvalues. The matrix $Y_* Y_*^T$ 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 $Y_* Y_*^T$ 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_i: When we fix all variables except the projection matrices $W_i, i = 1,\cdots,n$ in Eq.(5) we note that W_is are independent of each other, allowing us to independently optimize the projection matrix of each view. Hence, optimizing the projection matrix for the ith view reduces to solving the optimization problem given by

$$\arg\min_{W_i}\|X_i W_i - Y_*\|_F^2 + \beta\|W_i\|_{2,1}. \qquad (14)$$

To solve Eq.(14), the solution is found to be

$$W_i = \left(X_i^T X_i + \beta U_i\right)^{-1} X_i^T Y_* \tag{15}$$

where $U_i \in \mathbb{R}^{d_i \times d_i}$ is a diagonal matrix whose elements are $\frac{1}{2\|(W_i)_{j,:}\|_F}, j = 1,\ldots,d_i$, and $(W_i)_{j,:}$ is the jth 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 $\phi_i^2 = 2 - \frac{2}{\sqrt{sc}} \cdot tr\left(Y_i Y_i^T Y_* Y_*^T\right)$. 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

$$\underset{p_i \geq 0, \sum_{i=1}^V p_i = 1}{\arg\min} \quad \phi_i^2 / p_i, \tag{16}$$

which is solved by [27]:

$$p_i = \phi_i / \sum_{j=1}^{V} \phi_j. \tag{17}$$

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 Line3, 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.

Algorithm 2 ASCRA

Input: Multi-view data X, dimension of embedding s, number of features to select d, tuning parameters α, β

Output: Selected feature subset, consensus embedding Y_*, projection matrix W_i

1: initialize $p_i = 1/V$, W_i random matrix, and $U_i = I$, $i = 1,\ldots,V$.
2: compute view-specific similarity graph S_i, graph Laplacians $L_i = D_i - S_i$, and view-specific embedding $Y_i = \arg\min tr(Y_i^T L_i Y_i), i = 1,\ldots,V$.
3: compute Y_* using Eq.(12) with input matrix = $\sum_{i=1}^V Y_i / p_i$.
4: **repeat**
5: Update Y_* with Algorithm 1.
6: Update $Y_i, i = 1,\ldots,V$ by Eq.(13).
7: Update $W_i, i = 1,\ldots,V$ by Eq.(15).
8: Update $p_i, i = 1,\ldots,V$ by Eq.(17).
9: **until** convergence
10: Calculate $\|(W_i)_{j,:}\|_2, i = 1,\ldots,V, j = 1,\ldots,d_i$, rank them in descending order, and select the top d features.

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 $Y_* \in \mathbb{R}^{n \times s}$ and $Q \in \mathbb{R}^{n \times s}$,

$$tr(Y_*^T Q) = \sum_{i=1}^{s} (Y_*^T Q)_{i,i} = \sum_{i=1}^{n} (Q)_{i,j_i}, \text{where } (Y_*)_{i,j_i} = 1. \tag{18}$$

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

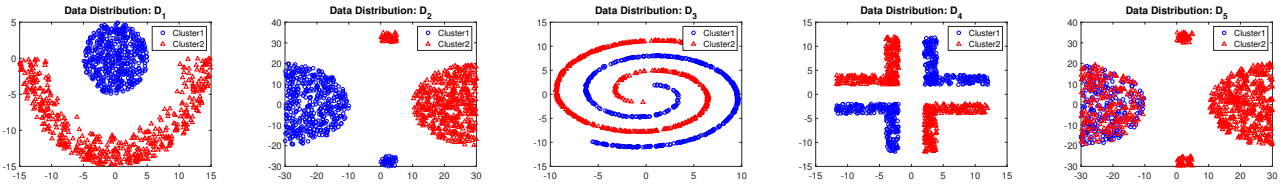
$$tr(Y_*^T Q) \geq tr(Y^T Q), \forall Y \in Ind. \tag{19}$$

Secondly, since $Y_*^{(t+1)}$ is the maximizer of $max_{i=1,\ldots,t} \hat{f}(Y_*^{(i)})$, we have: if $\hat{f}(Y_*^{(t+1)}) > \hat{f}(Y_*^{(t)})$, then the algorithm proceeds to the next iteration; and if $\hat{f}(Y_*^{(t+1)}) \leq \hat{f}(Y_*^{(t)})$, then the criterion is violated. Hence, it must be the case that $\exists j < t$ such that $Y_*^{(t+1)} = \arg\max \hat{f}(Y_*^{(j)})$. Consequently, $Y_*^{(t+1)} = Y_*^{(j+1)}$, 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_*, and each of the Y_i, W_i, p_i. To update Y_*, it executes an iterative procedure that repeats Eq.(9-12). The complexity of updating Y_* 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 ϕ_i^2. In summary, ASCRA's computational complexity is $O(n^2 s + \max_i d_i^3)$ for each update iteration.

On the other hand, ASCRA needs to store the values of consensus embedding, view-specific Laplacian matrices, 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 $O(\sum_{i=1}^V d_i s)$, and



(a) Crescent (D_1) (b) Outlier (D_2) (c) Two spirals (D_3) (d) Corners (D_4) (e) Outlier + noise (D_5)

Fig. 2. Sample distributions used in simulation study. All the samples are in $\mathbb{R}^2$.

adaptive view weights consume $O(V)$. In summary, ASCRA requires $O(V \cdot n^2 + \sum_{i=1}^{V} d_i s)$ 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 2×3 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.

TABLE I
Synthetic Data Composition

Dataset	D_1	D_2	D_3	D_4	D_5
Set 1 (2 views)	✓	✓			
Set 2 (2 views)	✓				✓
Set 3 (3 views)	✓	✓	✓		
Set 4 (4 views)	✓	✓	✓	✓	

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.

TABLE II

CLUSTERING RESULTS (PURITY±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).

Dataset	# Features	Laplacian	SPEC	AUMFS	ACSL	ASVW	ASCRA
Multi-omics	20	0.7071 ± 0	0.7071 ± 0	0.7071 ± 0.0000	0.7071 ± 0	0.7071 ± 0	0.7071 ± 0
	40	0.7071 ± 0	0.7071 ± 0	0.7072 ± 0.0004	0.7071 ± 0	0.7071 ± 0	0.7071 ± 0
	60	0.7071 ± 0	0.7071 ± 0	0.7072 ± 0.0004	0.7071 ± 0	0.7071 ± 0	0.7071 ± 0
	80	0.7071 ± 0	0.7071 ± 0	0.7071 ± 0.0000	0.7071 ± 0	0.7071 ± 0	0.7071 ± 0
	100	0.7071 ± 0	0.7071 ± 0	0.7072 ± 0.0004	0.7071 ± 0	0.7071 ± 0	0.7071 ± 0
Network1	20	0.4549 ± 0.0027	0.4538 ± 0.0035	0.4483 ± 0.0036	0.4457 ± 0.0030	**0.4692** ± 0.0064	**0.4726** ± 0.0033
	40	0.4601 ± 0.0062	0.4618 ± 0.0075	0.4550 ± 0.0059	0.4614 ± 0.0032	**0.4953** ± 0.0083	0.4807 ± 0.0048
	60	0.4662 ± 0.0076	0.4686 ± 0.0100	0.4641 ± 0.0071	0.4755 ± 0.0076	**0.4965** ± 0.0127	0.4862 ± 0.0076
	80	0.4701 ± 0.0083	0.4738 ± 0.0087	0.4727 ± 0.0076	0.4692 ± 0.0061	**0.4917** ± 0.0106	**0.4916** ± 0.0089
	100	0.4759 ± 00063	0.4747 ± 0.0081	0.4767 ± 0.0072	0.4761 ± 0.0082	**0.4938** ± 0.0118	**0.4931** ± 0.0094
Network2	20	0.3146 ± 0.0097	0.3162 ± 0.0060	0.3141 ± 0.0103	**0.3258** ± 0.0123	0.3156 ± 0.0072	**0.3297** ± 0.0103
	40	0.3185 ± 0.0081	0.3186 ± 0.0094	0.3376 ± 0.0121	0.3471 ± 0.0144	0.3203 ± 0.0107	**0.3639** ± 0.0134
	60	0.3122 ± 0.0059	0.3148 ± 0.0104	0.3477 ± 0.0131	0.3558 ± 0.0155	0.3366 ± 0.0140	**0.3789** ± 0.0165
	80	0.3157 ± 0.0097	0.3148 ± 0.0099	0.3467 ± 0.0155	0.3603 ± 0.0160	0.3525 ± 0.0087	**0.3809** ± 0.0181
	100	0.3188 ± 0.0102	0.3187 ± 0.0119	0.3499 ± 0.0144	**0.3801** ± 0.0206	0.3541 ± 0.0170	**0.3904** ± 0.0233
Handwritten digits	20	0.5984 ± 0.0267	0.6016 ± 0.0184	0.6705 ± 0.0336	0.7519 ± 0.0424	**0.7603** ± 0.0399	**0.7904** ± 0.0384
	40	0.6934 ± 0.0387	0.6967 ± 0.0404	0.6778 ± 0.0390	0.7807 ± 0.0578	0.7758 ± 0.0641	**0.8445** ± 0.0592
	60	0.6829 ± 0.0423	0.6814 ± 0.0512	0.6715 ± 0.0372	0.7861 ± 0.0579	0.7867 ± 0.0508	**0.8598** ± 0.0717
	80	0.6771 ± 0.0405	0.6970 ± 0.0417	0.7220 ± 0.0382	0.7891 ± 0.0706	**0.8242** ± 0.0701	**0.8485** ± 0.0478
	100	0.6937 ± 0.0479	0.7021 ± 0.0387	0.7281 ± 0.0360	0.7965 ± 0.0603	0.7924 ± 0.0634	**0.8644** ± 0.0592
MSRC-v1	20	0.5705 ± 0.0656	0.5467 ± 0.0604	**0.6186** ± 0.0642	0.6062 ± 0.0521	0.5374 ± 0.0351	**0.6507** ± 0.0414
	40	**0.6436** ± 0.0538	0.5833 ± 0.0418	0.6145 ± 0.0598	**0.6479** ± 0.0613	**0.6705** ± 0.0566	**0.6941** ± 0.0651
	60	**0.6476** ± 0.0598	0.6057 ± 0.0462	0.5791 ± 0.0470	**0.6807** ± 0.0473	**0.6450** ± 0.0412	**0.6798** ± 0.0526
	80	**0.6788** ± 0.0599	0.6348 ± 0.0495	0.6214 ± 0.0728	0.6331 ± 0.0708	0.6681 ± 0.0390	**0.7167** ± 0.0552
	100	**0.7133** ± 0.0620	0.6288 ± 0.0568	0.5862 ± 0.0381	**0.7043** ± 0.0833	0.6479 ± 0.0588	**0.7495** ± 0.0648

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. $Y_* \in Ind$. 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 AUMFS requires that the entries of the embedding must be non-negative and the ASCRA imposes a stronger constraint that the embedding indicates cluster membership.

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).

Dataset	# Features	Laplacian	SPEC	AUMFS	ACSL	ASVW	ASCRA
Multi-omics	20	0.0061 ± 0.0001	0.0007 ± 0.0003	$\mathbf{0.0091} \pm 0.0007$	0.0031 ± 0.0015	0.0043 ± 0.0004	0.0033 ± 0.0003
	40	0.0079 ± 0.0020	0.0001 ± 0.0000	$\mathbf{0.0084} \pm 0.0014$	0.0026 ± 0.0018	0.0037 ± 0.0007	0.0039 ± 0.0002
	60	0.0109 ± 0.0035	0.0001 ± 0.0000	0.0076 ± 0.0036	0.0016 ± 0.0013	$\mathbf{0.0123} \pm 0.0030$	0.0040 ± 0.0013
	80	0.0109 ± 0.0007	0.0001 ± 0.0000	0.0057 ± 0.0010	0.0039 ± 0.0038	$\mathbf{0.0098} \pm 0.0013$	0.0035 ± 0.0001
	100	0.0073 ± 0.0020	0.0000 ± 0.0000	0.0072 ± 0.0010	0.0030 ± 0.0021	$\mathbf{0.0127} \pm 0.0038$	0.0033 ± 0.0011
Network1	20	0.1335 ± 0.0039	0.1332 ± 0.0043	0.1270 ± 0.0068	0.1141 ± 0.0024	$\mathbf{0.1412} \pm 0.0049$	$\mathbf{0.1436} \pm 0.0040$
	40	0.1465 ± 0.0081	0.1468 ± 0.0099	0.1411 ± 0.0112	0.1415 ± 0.0063	$\mathbf{0.1592} \pm 0.0036$	$\mathbf{0.1603} \pm 0.0042$
	60	0.1508 ± 0.0068	0.1508 ± 0.0072	0.1463 ± 0.0108	0.1504 ± 0.0111	$\mathbf{0.1657} \pm 0.0072$	$\mathbf{0.1661} \pm 0.0037$
	80	0.1551 ± 0.0083	0.1567 ± 0.0093	0.1467 ± 0.0081	0.1499 ± 0.0068	$\mathbf{0.1712} \pm 0.0090$	$\mathbf{0.1706} \pm 0.0069$
	100	0.1583 ± 0.0067	0.1599 ± 0.0084	0.1535 ± 0.0068	0.1566 ± 0.0069	$\mathbf{0.1692} \pm 0.0066$	$\mathbf{0.1727} \pm 0.0070$
Network2	20	$\mathbf{0.1871} \pm 0.0066$	$\mathbf{0.1860} \pm 0.0052$	0.1386 ± 0.0096	0.1335 ± 0.0091	0.1810 ± 0.0084	0.1579 ± 0.0080
	40	0.1807 ± 0.0076	0.1823 ± 0.0068	0.1691 ± 0.0084	0.1552 ± 0.0121	0.1836 ± 0.0071	$\mathbf{0.2030} \pm 0.0090$
	60	0.1792 ± 0.0063	0.1787 ± 0.0055	0.1763 ± 0.0091	0.1676 ± 0.0138	0.1914 ± 0.0081	$\mathbf{0.2159} \pm 0.0118$
	80	0.1819 ± 0.0114	0.1817 ± 0.0089	0.1841 ± 0.0109	0.1789 ± 0.0134	0.2005 ± 0.0071	$\mathbf{0.2170} \pm 0.0125$
	100	0.1834 ± 0.0089	0.1818 ± 0.0090	0.1872 ± 0.0113	0.1977 ± 0.0131	0.2024 ± 0.0104	$\mathbf{0.2214} \pm 0.0172$
Handwritten digits	20	0.5451 ± 0.0150	0.5554 ± 0.0208	0.6463 ± 0.0210	0.7405 ± 0.0211	0.6879 ± 0.0274	$\mathbf{0.7733} \pm 0.0261$
	40	0.6482 ± 0.0256	0.6488 ± 0.0247	0.6716 ± 0.0244	0.7511 ± 0.0383	0.7504 ± 0.0414	$\mathbf{0.8292} \pm 0.0385$
	60	0.6330 ± 0.0299	0.6416 ± 0.0341	0.6532 ± 0.0208	0.7767 ± 0.0421	0.7807 ± 0.0337	$\mathbf{0.8416} \pm 0.0448$
	80	0.6362 ± 0.0301	0.6552 ± 0.0256	0.6808 ± 0.0229	0.7723 ± 0.0548	$\mathbf{0.8078} \pm 0.0397$	$\mathbf{0.8332} \pm 0.0313$
	100	0.6536 ± 0.0278	0.6556 ± 0.0260	0.6877 ± 0.0171	0.7775 ± 0.0438	0.7882 ± 0.0396	$\mathbf{0.8504} \pm 0.0341$
MSRC-v1	20	0.4673 ± 0.0477	0.4383 ± 0.0432	0.5016 ± 0.0597	0.5005 ± 0.0374	0.4226 ± 0.0327	$\mathbf{0.5690} \pm 0.0383$
	40	0.5383 ± 0.0415	0.4553 ± 0.0272	0.4832 ± 0.0465	0.5366 ± 0.0480	$\mathbf{0.5773} \pm 0.0389$	$\mathbf{0.5909} \pm 0.0580$
	60	0.5429 ± 0.0547	0.4635 ± 0.0352	0.4513 ± 0.0311	$\mathbf{0.5674} \pm 0.0491$	0.5384 ± 0.0285	$\mathbf{0.5871} \pm 0.0567$
	80	0.5786 ± 0.0533	0.4978 ± 0.0332	0.4915 ± 0.0619	0.5473 ± 0.0591	0.5602 ± 0.0343	$\mathbf{0.6294} \pm 0.0577$
	100	0.5979 ± 0.0499	0.4875 ± 0.0397	0.4669 ± 0.0340	0.6093 ± 0.0735	0.5462 ± 0.04871	$\mathbf{0.6733} \pm 0.0603$

TABLE IV

CLUSTERING PERFORMANCE ON SYNTHETIC DATA

Data set	Metric	AUMFS	ACSL	ASVW	ASCRA
Set 1	NMI	**0.9595**	**0.9595**	0.4320	**0.9595**
	Accuracy	**0.9950**	**0.9950**	0.8113	**0.9950**
Set 2	NMI	0.3557	0.0013	0.3229	**0.3569**
	Accuracy	0.7902	0.7910	0.7611	**0.7912**
Set 3	NMI	**0.9595**	0.5430	0.0368	**0.9595**
	Accuracy	**0.9950**	0.8723	0.6122	**0.9950**
Set 4	NMI	**0.9595**	0.3737	0.5498	**0.9595**
	Accuracy	**0.9950**	0.7731	0.8753	**0.9950**

TABLE V

CLUSTERING PERFORMANCE OF CONSENSUS EMBEDDING ON SYNTHETIC DATA

Data set	Metric	AUMFS	ACSL	ASCRA
Set 1	NMI	**0.9812**	0.0013	0.9595
	Accuracy	**0.9980**	0.5210	0.9950
Set 2	NMI	**0.3692**	0.0000	0.3436
	Accuracy	**0.8010**	0.5060	0.7800
Set 3	NMI	0.9467	0.0020	**0.9530**
	Accuracy	0.9930	0.5260	**0.9940**
Set 4	NMI	0.9406	0.0023	**0.9467**
	Accuracy	0.9920	0.5280	**0.9930**

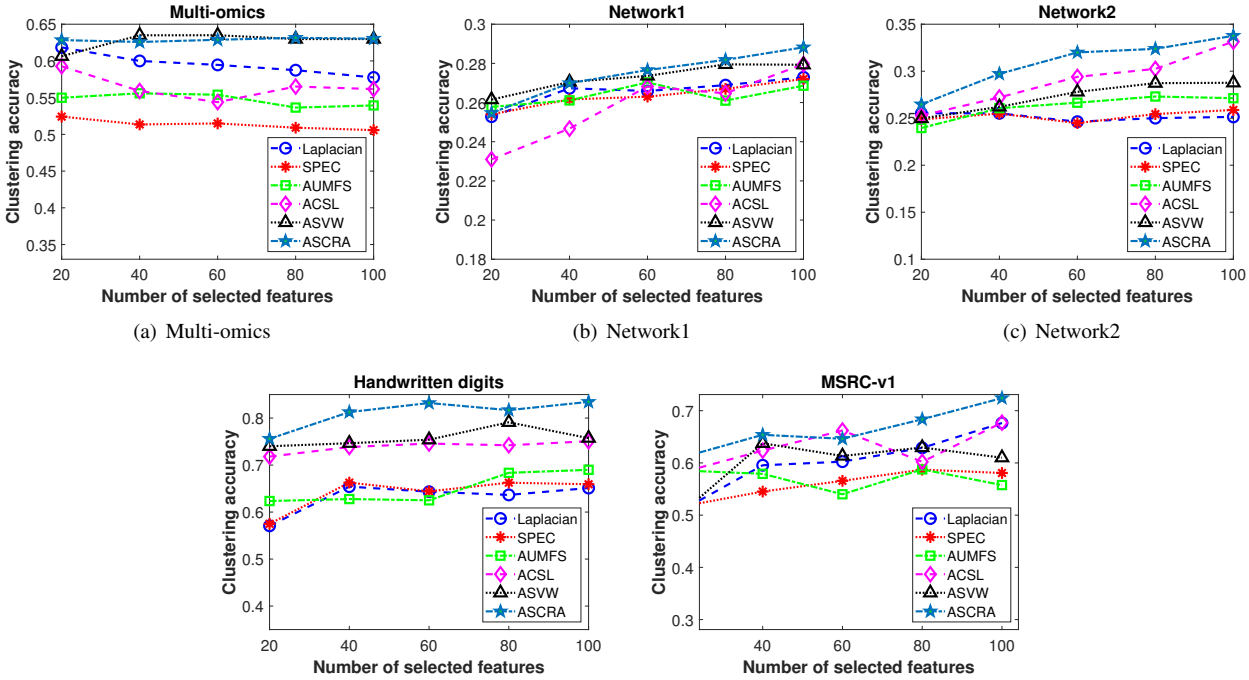
Parameter Sensitivity Analysis. To examine how the choice of the hyper-parameters, α and β, affect the performance of ASCRA, we conducted a grid search over the space spanned by $\alpha \in \{10^p : p = -4, -3, \ldots, 1\}$ and $\beta \in \{10^p : p = -3, -2, \ldots, 2\}$. For each parameter combination, 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 α and β.

TABLE VI

THE ACTUAL RUNNING TIME, IN SECONDS, PER UPDATE ITERATION OF EACH MULTI-VIEW FEATURE SELECTION METHOD.

	AUMFS	ACSL	ASVW	ASCRA
Multi-omics	429.3	518.3	84.8	48.6
Network1	11.00	1.01	1.14	1.60
Network2	2.12	0.47	0.35	0.74
Handwritten digits	29.19	2.88	2.60	3.14
MSRC-v1	0.79	0.18	0.62	0.08

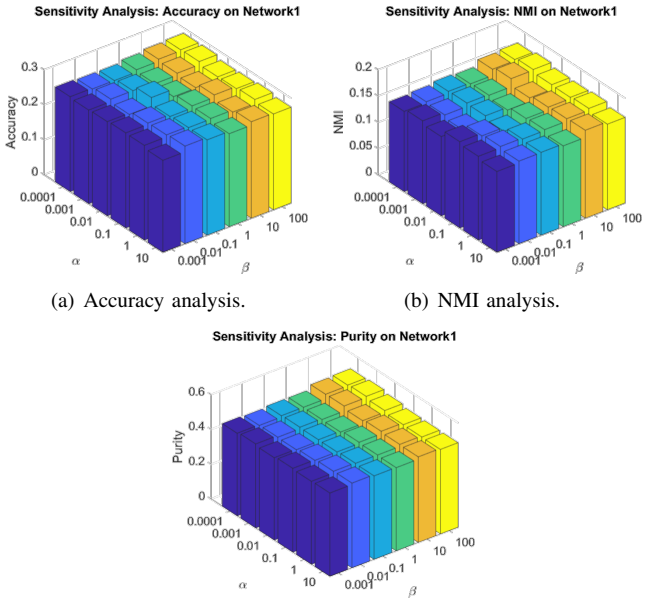
Empirical Computational Complexity. Recall that the complexity per iteration of the proposed algorithm is $O(n^2 s + \max_i d_i^3)$. Recall that the worst-case runtime complexities of the methods compared in our experiments, the complexities are, AUMFS: $O(n^3 + (\sum_i d_i)^3)$, ACSL: $O(n^2 s + (\sum_i d_i)^3)$, ASVW: $O(\sum_i d_i \times \max\{d_i^2, n \times k \times s\})$, 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 ACSL 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 ACSL and ASVW, as expected from a larger constant factor associated with the n^2 term.



(a) Multi-omics (b) Network1 (c) Network2

(d) Handwritten digits (e) MSRC-v1

Fig. 3. Clustering accuracy of K-Means applied to real-world data encoded using only the features selected by different feature selection methods for different choices of the number of features to be selected.



(a) Accuracy analysis. (b) NMI analysis.

(c) Purity analysis.

Fig. 4. Parameter sensitivity analysis performed on Network1 dataset.

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

Honavar). The content is solely the responsibility of the authors and does not necessarily represent the official views of the sponsors.

REFERENCES

[1] H. Wang, F. Nie, and H. Huang, "Multi-view clustering and feature learning via structured sparsity," in *International conference on machine learning*, pp. 352–360, 2013.

[2] Z. Ding, M. Shao, and Y. Fu, "Robust multi-view representation: A unified perspective from multi-view learning to domain adaption.," in *IJCAI*, pp. 5434–5440, 2018.

[3] S. Bickel and T. Scheffer, "Multi-view clustering.," in *ICDM*, vol. 4, pp. 19–26, 2004.

[4] S. Sun, "A survey of multi-view machine learning," *Neural computing and applications*, vol. 23, no. 7-8, pp. 2031–2038, 2013.

[5] Y. Zhou, Y. Sun, and V. Honavar, "Improving image captioning by leveraging knowledge graphs," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 283–293, IEEE, 2019.

[6] S.-L. Wu, Y.-T. Liu, T.-Y. Hsieh, Y.-Y. Lin, C.-Y. Chen, C.-H. Chuang, and C.-T. Lin, "Fuzzy integral with particle swarm optimization for a motor-imagery-based brain–computer interface," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 1, pp. 21–28, 2016.

[7] Y. El-Manzalawy, T. Hsieh, M. Shivakumar, D. Kim, and V. Honavar, "Min-redundancy and max-relevance multi-view feature selection for predicting ovarian cancer survival using multi-omics data 06 biological sciences 0604 genetics," *BMC Medical Genomics*, vol. 11, 9 2018.

[8] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *arXiv preprint arXiv:1304.5634*, 2013.

[9] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Information Fusion*, vol. 38, pp. 43–54, 2017.

[10] R. Zhang, F. Nie, X. Li, and X. Wei, "Feature selection with multi-view data: A survey," *Information Fusion*, vol. 50, pp. 158 – 167, 2019.

[11] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 94, 2018.

[12] S. Alelyani, J. Tang, and H. Liu, "Feature selection for clustering: A review," in *Data Clustering*, pp. 29–60, Chapman and Hall/CRC, 2018.

[13] M. Qian and C. Zhai, "Robust unsupervised feature selection," in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[14] S. Wang, J. Tang, and H. Liu, "Feature selection," *Encyclopedia of Machine Learning and Data Mining*, pp. 1–9, 2016.

[15] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proceedings of the 24th international conference on Machine learning*, pp. 1151–1157, ACM, 2007.

[16] Z. Wang, Y. Feng, T. Qi, X. Yang, and J. J. Zhang, "Adaptive multi-view feature selection for human motion retrieval," *Signal Processing*, vol. 120, pp. 691–701, 2016.

[17] J. Tang, X. Hu, H. Gao, and H. Liu, "Unsupervised feature selection for multi-view data in social media," in *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 270–278, SIAM, 2013.

[18] Y. Feng, J. Xiao, Y. Zhuang, and X. Liu, "Adaptive unsupervised multi-view feature selection for visual concept recognition," in *Asian conference on computer vision*, pp. 343–357, Springer, 2012.

[19] X. Dong, L. Zhu, X. Song, J. Li, and Z. Cheng, "Adaptive collaborative similarity learning for unsupervised multi-view feature selection," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 2064–2070, AAAI Press, 2018.

[20] C. Hou, F. Nie, H. Tao, and D. Yi, "Multi-view unsupervised feature selection with adaptive similarity and view weight," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 9, pp. 1998–2011, 2017.

[21] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in neural information processing systems*, pp. 585–591, 2002.

[22] X. He and P. Niyogi, "Locality preserving projections," in *Advances in neural information processing systems*, pp. 153–160, 2004.

[23] Y. Sun, N. Bui, T.-Y. Hsieh, and V. Honavar, "Multi-view network embedding via graph factorization clustering and co-regularized multi-view agreement," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 1006–1013, IEEE, 2018.

[24] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Advances in neural information processing systems*, pp. 1413–1421, 2011.

[25] K. He, Y. Sun, D. Bindel, J. Hopcroft, and Y. Li, "Detecting overlapping communities from local spectral subspaces," in *2015 IEEE International Conference on Data Mining*, pp. 769–774, IEEE, 2015.

[26] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[27] F. Nie, L. Tian, and X. Li, "Multiview clustering via adaptively weighted procrustes," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2022–2030, ACM, 2018.

[28] R. Lazimy, "Mixed-integer quadratic programming," *Mathematical Programming*, vol. 22, no. 1, pp. 332–349, 1982.

[29] C. Buchheim and L. Trieu, "Quadratic outer approximation for convex integer programming with box constraints," in *International Symposium on Experimental Algorithms*, pp. 224–235, Springer, 2013.

[30] S. Wang, J. Tang, and H. Liu, "Embedded unsupervised feature selection," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[31] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l 2, 1-norm minimization," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pp. 339–348, AUAI Press, 2009.

[32] X. Lian, M. Wang, and J. Liu, "Finite-sum composition optimization via variance reduced gradient descent," in *Artificial Intelligence and Statistics*, pp. 1159–1167, 2017.

[33] T.-Y. Hsieh, E.-M. Yasser, Y. Sun, and V. Honavar, "Compositional stochastic average gradient for machine learning and related applications," in *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 740–752, Springer, 2018.

[34] C. Hou, F. Nie, D. Yi, and Y. Wu, "Feature selection via joint embedding learning and sparse regression," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, p. 1324, 2011.

[35] M. Goldman, B. Craft, T. Swatloski, M. Cline, O. Morozova, M. Diekhans, D. Haussler, and J. Zhu, "The ucsc cancer genomics browser: update 2015," *Nucleic acids research*, vol. 43, no. D1, pp. D812–D817, 2014.

[36] Y. Sun, S. Wang, T.-Y. Hsieh, X. Tang, and V. Honavar, "Megan: A generative adversarial network for multi-view network embedding," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, 2019.

[37] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, ACM, 2016.

[38] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

[39] J. Winn and N. Jojic, "Locus: Learning object classes with unsupervised segmentation," in *null*, pp. 756–763, IEEE, 2005.

[40] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.

[41] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[42] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 971–987, 2002.

[43] J. Wu and J. M. Rehg, "Where am i: Place instance and category recognition using spatial pact," in *2008 Ieee Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[44] C.-T. Lin, T.-Y. Hsieh, Y.-T. Liu, Y.-Y. Lin, C.-N. Fang, Y.-K. Wang, G. Yen, N. R. Pal, and C.-H. Chuang, "Minority oversampling in kernel adaptive subspaces for class imbalanced datasets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 5, pp. 950–962, 2017.

[45] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in neural information processing systems*, pp. 507–514, 2006.