Multi-View Fusion with Extreme Learning Machine for Clustering

YONGSHAN ZHANG, China University of Geosciences, China
JIA WU, Macquarie University, Australia
CHUAN ZHOU, Chinese Academy of Sciences, China
ZHIHUA CAI, China University of Geosciences, China
JIAN YANG, Macquarie University, Australia
PHILIP S. YU, University of Illinois at Chicago, Chicago, IL

Unlabeled, multi-view data presents a considerable challenge in many real-world data analysis tasks. These data are worth exploring because they often contain complementary information that improves the quality of the analysis results. Clustering with multi-view data is a particularly challenging problem as revealing the complex data structures between many feature spaces demands discriminative features that are specific to the task and, when too few of these features are present, performance suffers. Extreme learning machines (ELMs) are an emerging form of learning model that have shown an outstanding representation ability and superior performance in a range of different learning tasks. Motivated by the promise of this advancement, we have developed a novel multi-view fusion clustering framework based on an ELM, called MVEC. MVEC learns the embeddings from each view of the data via the ELM network, then constructs a single unified embedding according to the correlations and dependencies between each embedding and automatically weighting the contribution of each. This process exposes the underlying clustering structures embedded within multi-view data with a high degree of accuracy. A simple yet efficient solution is also provided to solve the optimization problem within MVEC. Experiments and comparisons on eight different benchmarks from different domains confirm MVEC's clustering accuracy.

CCS Concepts: • **Computing methodologies** → *Unsupervised learning*;

Additional Key Words and Phrases: Multi-view clustering, multi-view embedding, extreme learning machine, unsupervised learning

This work is supported in part by the National Natural Science Foundation of China (Grant Nos. 61773355 and 61872360), the MQNS under Grant No. 9201701203, the MQEPS under Grant No. 9201701455, the MQRSG under Grant No. 95109718, the 2018 Collaborative Research Project between Macquarie University and Data61, the Youth Innovation Promotion Association CAS (No. 2017210), the US NSF under grants III-1526499, III-1763325, III-1909323, SaTC-1930941, and CNS-1626432 and the National Scholarship for Building High Level Universities, China Scholarship Council (No. 201706410005).

Authors' addresses: Y. Zhang and Z. Cai, China University of Geosciences, School of Computer Science, 388 Lumo Rd, Wuhan, Hubei, 430074, China; emails: yszhang.cug@gmail.com, zhcai@cug.edu.cn; J. Wu and J. Yang, Macquarie University, Department of Computing, Faculty of Science and Engineering, Balaclava Rd, Sydney, NSW, 2109, Australia; emails: {jia.wu, jian.yang}@mq.edu.au; C. Zhou, Chinese Academy of Sciences, Institute of Information Engineering, 89 Minzhuang Rd., Beijing, 100093, China; email: zhouchuan@iie.ac.cn; P. S. Yu, University of Illinois at Chicago, Department of Computer Science, 851 Morgan St., Chicago, IL, 60607; email: psyu@uic.edu.

2157-6904/2019/10-ART53 $15.00
https://doi.org/10.1145/3340268

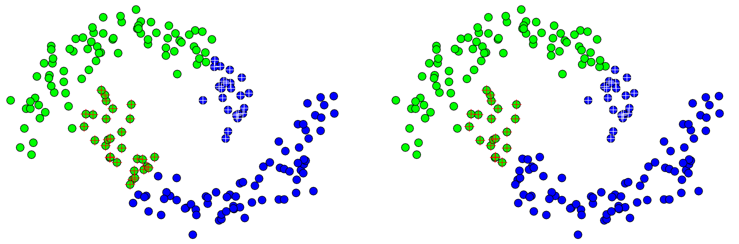
ACM Reference format:
Yongshan Zhang, Jia Wu, Chuan Zhou, Zhihua Cai, Jian Yang, and Philip S. Yu. 2019. Multi-View Fusion with Extreme Learning Machine for Clustering. *ACM Trans. Intell. Syst. Technol.* 10, 5, Article 53 (October 2019), 23 pages.
https://doi.org/10.1145/3340268

1 INTRODUCTION

In many real-world data analysis tasks, the data to be analyzed often comes from heterogeneous sources and, therefore, represents multiple views of the same or similar information. For example, news articles covering the same story might be obtained from the BBC, Reuters, and The Guardian [8]. Vehicle signal data might be derived from an amalgamation of different acoustic and seismic sensors [3]. Image data can be described in different ways, e.g., as a wavelet texture, an edge direction histogram, or a color moment [52], and so on. These multi-view datasets often provide complementary information to each other, and leveraging the correlations and interactions between these views is usually beneficial to learning performance. Given the prevalence of unlabeled multi-view data, multi-view clustering is becoming more popular as a way to integrate all views in an unsupervised manner [27, 38, 48].

Multi-view clustering is a special learning paradigm, where similar objects are clustered into groups and the remaining dissimilar objects are clustered into another group by leveraging the information hidden within the data [1, 34]. There are numerous multi-view clustering methods, most of which are simply extensions of classical single-view clustering methods [44, 45]. For instance, nonnegative matrix factorization [21] reveals the underlying clustering structures embedded in multi-view data by reaching a common consensus between the views. Co-regularized multi-view spectral clustering [19] centers on developing a clustering hypothesis across all views and then assimilating the clusters from each view that are consistent with the initial premise. Auto-weighted multi-graph learning [26] is a parameter-free clustering method for multi-view data, which, as the name suggests, automatically learns the contribution weightings of multiple similarity graphs. Despite solid demonstrated performance with particular multi-view problems, the current multi-view clustering methods have been designed to work with original multi-view feature spaces and, therefore, do not fully consider the discriminative capacity of the data. Performance can be unstable and susceptible to noise. Given these observations, we find it fundamental and vital to learn a unified representation that integrates all the different views within multi-view data. Such a representation would have the power to reveal better discriminative features and, in turn, further improve clustering performance.

Generally, nonlinear feature mapping is an effective technique for learning discriminative representations from original data. As an emerging type of learning model, extreme learning machines (ELMs) have shown great potential for converting original data into a new feature space. For this reason, the topic has attracted great attention in recent years [36, 46]. An ELM can be used to train a "generalized" single-hidden layer feed-forward neural network (SLFN) and, for a number of applications, different variants of an ELM have provided an effective solution for learning representative features. For example, Kasun et al. [17] designed an ELM variant that imposes an orthogonality constraint on the connected network parameters and can then extract representative features. The approach shows promising performance in hand-written digit recognition. The augmented ELM method developed by Uzair and Mian [39] uses a global ELM-based autoencoder model to learn class-specific ELM-based autoencoder models so as to reveal extra discriminative representations for domain adaption applications. ELM-based clustering methods [11] cluster the feature spaces learned by the ELM rather than the features in the original data. The superiority of this ELM feature mapping method has been empirically verified in experiments. Each of these



(a) Clustering result for original data (b) Clustering result for learned features

Fig. 1. The discriminative capacity of the original Toy data (a) and the learned features with the Toy dataset (b). All solid circles represent different labels. The labels with crosses indicate an incorrect prediction. Clustering accuracy for the original data was 0.740, while the clustering accuracy for the nonlinear features learned by an ELM was 0.805.

studies demonstrates that designing an effective learning method based on the more discriminative representations learned by an ELM is a desirable undertaking. Figure 1 illustrates an example of the difference in discriminative capacity between the original data and the nonlinear features learned by an ELM on the Toy dataset. The clustering accuracy for the nonlinear features is higher than that for the original data.

Based on the above encouraging results, we propose a novel multi-view fusion clustering framework based on an ELM, called MVEC. MVEC offers an unsupervised learning technique for determining suitable network parameters as the initial inputs to an ELM network. Individual embeddings of each view in a multi-view dataset are nonlinearly learned by the ELM. These embeddings are more discriminative than original data. One commonly shared and unified embedding is then built from an automatic and self-adaptive weighting process, which reflects the contribution of each learned embedding given the correlations and dependencies between them. The unified embedding reveals the consistent clustering structures embedded within multi-view data with greater accuracy and, as such, provides an interpretable and effective solution to multi-view clustering problems. Experiments and comparisons on a diverse selection of real-world multi-view clustering tasks confirm MVEC's effectiveness and efficiency.

The advantages of MVEC and its main contributions are summarized below.

- MVEC is a novel and generalized multi-view clustering model that learns a unified and representative embedding from a given set of multi-view data via an ELM;
- MVEC unearths the consistent structures within multi-view data in an unsupervised manner;
- The contribution of each individual embedding to the unified embedding is automatically weighted through a self-adaptive process;
- The algorithm is optimized with a novel, effective, and efficient iterative solution; and
- Experiments with eight real-world problems verify that MVEC improves clustering accuracy at efficient speeds. In addition, we have provided analyses of other pertinent factors, including convergence, parameter sensitivity, and running time.

The rest of this article is structured as follows: Section 2 briefly surveys related work on multi-view clustering and ELMs. Section 3 introduces important notations and the problem description. Section 4 sets out the proposed MVEC method in detail along with the novel optimization algorithm. The analyses of convergence and time complexity follow in Section 5. The experiments and comparisons are provided in Section 6. Finally, we conclude the article in Section 7.

2 RELATED WORK

2.1 Multi-View Clustering

As more and more information becomes available from different sources, datasets are increasingly comprised of multiple views (i.e., multiple modalities) [33, 35, 40]. Acquiring valid label information is expensive in some domains, such as disease diagnosis [10] or multilingual analysis [18]. Combined, these two factors mean there is an abundance of unlabeled multi-view data and, consequently, the popularity of multi-view clustering is on the rise [6, 20, 53]. Multi-view clustering is a special unsupervised learning paradigm that clusters similar features from different views into the same group without the need for labels, while dissimilar objects are clustered into a separate group and set aside. By exploring the interactions and correlations between the different views, multi-view clustering approaches use the data's underlying structures to improve clustering performance.

To date, numerous multi-view clustering algorithms have been proposed to solve a range of real-world application problems, and many have been effective for the tasks they were designed to accomplish. Overall, these methods can be broadly categorized into two main approaches: graph-based clustering and subspace-based clustering [24].

Graph-based clustering approaches construct an affinity graph of each view, which mainly rely on the quality of the obtained real-world datasets. Among this type of approach, linked matrix factorization [38] decomposes affinity graph matrices of multiple views into various characteristic matrices and a shared matrix, then clusters the data according to the results. The multi-modal spectral clustering method [4] explores a common graph Laplacian matrix by integrating features from heterogeneous views to boost multi-view clustering. The self-weighted multi-view clustering method [27] explores a Laplacian rank-constrained graph as a common graph of all the views. It also includes an auto-weighting mechanism. Although graph-based clustering approaches gain the state-of-the-art performance, there still exist at least two limits, i.e., unreliable similarity matrix and improper neighbor assignment. Actually, for one thing, such methods directly conduct the followed procedure based on the constructed similarity matrix from original data but they rarely modify it. For another, those methods may suffer from inferior performance due to noises and outlying entries are contained in real-world datasets.

Subspace-based clustering approaches learn subspace representations from the original multi-view data, which are conspicuous for efficiency and excellent clustering performance. The approach presented in [7] learns the subspace representations while maintaining the consistencies between different views. The latent multi-view approach [48] generates a shared latent representation from all feature views to explore the data's underlying structures for clustering. The multi-view low-rank sparse approach [2] constructs a unified affinity matrix from multiple views to explore a shared subspace representation. Different from graph-based approaches, subspace-based approaches are based on the assumption that the views are generated from a single latent source, and the variation within the views is independent with such latent source. Those methods are often conducted to discriminate each view with the shared variable independently and then updating the parameters for the shared space.

All of these methods were developed on original multi-view feature spaces. They do not fully consider the discriminative properties of the data.

2.2 Extreme Learning Machines

ELMs are an emerging learning model for training "generalized" SLFNs with the aim of providing superior performance and fast learning speeds for complicated application problems [9, 29, 42]. As opposed to conventional neural networks, which require iterative parameter tuning, ELMs need very little parameter adjustment [5, 15]. Typically, basic ELMs are partitioned into two main steps: ELM feature mapping and ELM parameter solving. In the first step, the original data is transformed into a latent representation via nonlinear feature mapping and connected network parameters. The input network parameters are randomly generated, and the selected activation function should be infinitely differentiable. In the second step, the output network parameters are analytically solved with a generalized Moore-Penrose (MP) inverse and the minimum norm least-squares solution to a general linear system without the need for iterative learning. The detailed learning procedure for training an ELM network is shown in Algorithm 1.

```
ALGORITHM 1: Training procedure for ELM
Input:
  Training Data; Activation Function; Hidden neuron number;
Output:
  Output network parameters;
1: Randomly assign input network parameters and hidden biases;
2: Calculate the hidden layer output matrix with training data using matrix multiplication;
3: Calculate output network parameters according to Moore-Penrose generalized inverse.
```

In reality, neural networks can naturally extract features that are more discriminative than the original data. As a neural network training method, ELMs show an outstanding representation ability. For example, ELMs can learn a compressed representation derived from a low-dimensional feature space or a sparse representation from a high-dimensional feature space [43, 46]. Further, they can retain the potential information hidden in the original data while learning different representations. Due to this outstanding representation ability, ELMs have been applied to a range of clustering tasks. Huang et al. [14] were the first to propose an unsupervised ELM method to solve both embedding and clustering tasks. The method is based on Laplacian eigenmaps and spectral clustering. In [23], Liu et al. presented a novel ELM variant to preserve the manifold data structure and maximize the separability of different classes for both embedding and clustering. Liu et al. [22] developed a dual data representation based on a graph clustering method that depends on both the original data and a nonlinear feature representation obtained by an ELM. The above-mentioned methods are based on generic data. For multi-view data, in [41], Wang et al. advanced a general multi-view clustering framework based on ELM feature mapping. The framework can be generalized to other multi-view clustering approaches, and the representation ability of the ELM feature mapping can be analytically verified. For both generic data and multi-view data, those ELM-based methods show the capability of learning discriminative features for different learning tasks. Given that ELM feature mapping learns discriminative features from the original data, it has the potential to benefit a diverse range of learning tasks.

Our contribution to this growing field of discovery is a complete multi-view fusion clustering framework that incorporates an ELM and automatically weights individual embeddings to create a unified embedding across an entire set of multi-view data.

3 NOTATION AND PROBLEM DESCRIPTION

Throughout this article, boldface uppercase letters denote a matrix (e.g., $\mathbf{X}$), boldface lowercase letters denote a vector (e.g., $\mathbf{x}$) and italics denote a scalar (e.g., x). Calligraphic letters (e.g., $\mathcal{X}$) are used to denote different sets. For any matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$, $\mathbf{x}_i$ denotes the ith column vector of $\mathbf{X}$, where $\mathbf{x}^j$ means the jth row vector of $\mathbf{X}$ and x_{ji} indicates the ith row jth column element of $\mathbf{X}$. The Frobenius norm (F-norm) is denoted as $||\mathbf{X}||_F = \sqrt{\sum_{j=1}^{J} \sum_{i=1}^{I} x_{ji}^2}$. The transpose and inverse

Table 1. List of Important Notations

Notation	Description
x	Normal italic letter denotes a scalar
$\mathbf{x}$	Boldface lowercase denotes a vector
$\mathbf{X}$	Boldface uppercase letter denotes a matrice
$\mathcal{X}$	Calligraphic letter denotes a set
$\lVert \cdot \rVert_F$	Denotes Frobenius norm of matrix
N	Denotes number of instances in multi-view dataset
D_v	Denotes number of features in the vth view feature space
V	Denotes number of feature views in multi-view dataset
L	Denotes number of hidden nodes for ELM network
M	Denotes dimension of embedding (i.e., number of output nodes for ELM network)

operators of a matrix $\mathbf{X}$ are further represented as $\mathbf{X}^T$ and $\mathbf{X}^{-1}$, respectively. These notations are summarized in Table 1 along with some additional important notations.

Consider an unlabeled multi-view dataset, $\mathcal{X} = \{\mathbf{X}^{(v)}\}_{v=1}^{V}$ are with V different feature views, where $\mathbf{X}^{(v)} \in \mathbb{R}^{D_v \times N}$ is the vth feature view with N instances and D_v features, and $D = \sum_{v=1}^{V} D_v$ is the total number of features in the multi-view dataset. Our goal is to partition N instances into K clusters by exploiting the information in every different view V of the data. This is the learning task for multi-view clustering. More precisely, our aim is to accurately perform clustering from a unified embedding that captures the complementary information across all views so as to provide an effective solution for multi-view clustering problems.

4 THE PROPOSED MVEC METHOD

This section begins by presenting MVEC's formulation. The formulation of the algorithm to solve the proposed formulation follows. A conceptual view of this framework is illustrated in Figure 2.

4.1 Formulation of MVEC

4.1.1 Unsupervised Parameter Initialization. In diverse ELM learning models, network parameter vectors are usually randomly selected from an open set of arbitrary vectors, which may mean more hidden nodes are required to improve performance. In general, the network parameter vectors in an ELM should easily and accurately separate different samples into different groups. Most existing methods based on ELMs use label information to determine which vectors are the most suitable [49, 50]. For some sources of data, acquiring accurate label information means human annotation, and that is expensive. And, with an ELM, establishing effective network parameter vectors without label information is a challenging problem.

Typically, the difference vectors of between-class samples in a specific dataset should show outstanding discriminative characteristics for a range of learning tasks [32]. However, in unsupervised learning scenarios, little or no label information is known. Motivated by [51], MVEC determines the network parameters for the ELM by calculating the difference between pairwise samples. To overcome the deficiency of label information, we use a large threshold to control the difference for choosing pair-wise samples, which increases the probability of drawing two very distinctive samples. Suppose that $\mathbf{x}_{r1}$ and $\mathbf{x}_{r2}$ are randomly chosen samples from the original input data, the weight vector $\mathbf{w}$ for all input nodes to one hidden node can be calculated as follows:

$$\mathbf{w} = \mu(\mathbf{x}_{r2} - \mathbf{x}_{r1}); \tag{1}$$

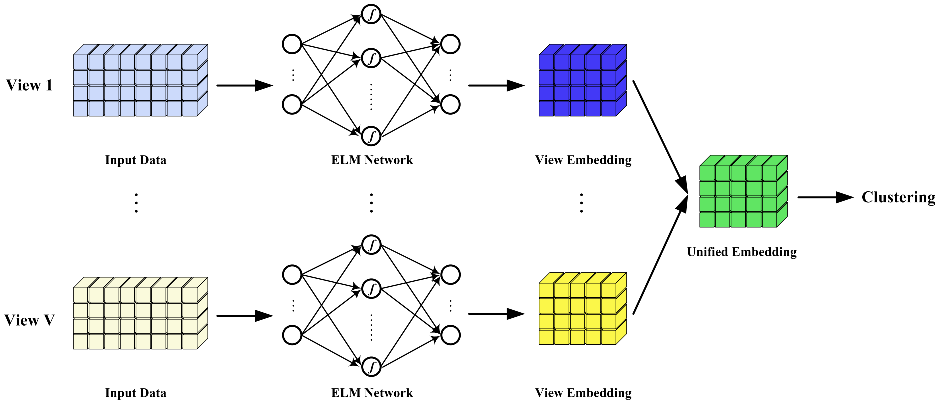


Fig. 2. A conceptual view of MVEC.

where μ is the normalized factor. During ELM feature mapping, the original data $\mathbf{x}$ is transformed into a latent representation:

$$\mu\mathbf{x}^T\mathbf{w} + a = \mu\mathbf{x}^T(\mathbf{x}_{r2} - \mathbf{x}_{r1}) + a; \tag{2}$$

where a is the bias for the hidden node in relation to the weight vector $\mathbf{w}$. Notably, samples x_{r1} and x_{r2} from different class-label groups should show a distinct feature mapping, which may provide more discriminative information when learning the network parameters. The assumption that x_{r1} and x_{r2} are mapped to -1 and 1 is represented as follows:

$$\begin{cases} \mu\mathbf{x}_{r1}^T(\mathbf{x}_{r2} - \mathbf{x}_{r1}) + a = -1 \\ \mu\mathbf{x}_{r2}^T(\mathbf{x}_{r2} - \mathbf{x}_{r1}) + a = 1. \end{cases} \tag{3}$$

By solving Equation (3), the normalization factor can be calculated as

$$\mu = \frac{2}{||\mathbf{x}_{r2} - \mathbf{x}_{r1}||_{L_2}^2}; \tag{4}$$

and the corresponding bias can be determined from

$$a = \frac{(\mathbf{x}_{r2} + \mathbf{x}_{r1})^T(\mathbf{x}_{r2} - \mathbf{x}_{r1})}{||\mathbf{x}_{r2} - \mathbf{x}_{r1}||_{L_2}^2}. \tag{5}$$

The network parameter vectors and the biases for the other hidden nodes can be determined in the same way. For simplicity, the network parameter matrix that connects the input nodes and the hidden nodes is denoted as $\mathbf{W}$, and the bias vector for the hidden nodes is denoted as $\mathbf{a}$. The above-mentioned network parameters are drawn from a constrained set of difference vectors for the pairwise samples instead of an open set of arbitrary vectors. This is the foundation for MVEC, which can provide a more appropriate solution to unsupervised parameter initialization. Appropriate network parameter initialization is vital to the subsequent learning process (i.e., view-independent embedding learning and consensus embedding learning). Compared to randomly selected network parameters, constrained network parameters provide superior input parameters and lead to a satisfactory performance, as proven in the experiments in Section 6.9. Moreover, this is an unsupervised parameter initialization method that does not require explicit label information for learning.

4.1.2 View-Independent Embedding Learning. In general, multi-view data is often not sufficiently discriminative for the learning task at hand. Therefore, to improve the discriminative qualities of the data, it is vital to further explore other representations. As a neural network training algorithm, ELM is specifically designed for nonlinear feature mapping and can, therefore, learn new and potentially more distinct representations [47]. For the vth ($1 \leq v \leq V$) view data $\mathbf{X}^{(v)} \in \mathbb{R}^{D_v \times N}$, assume there is an ELM network with L hidden nodes. Here, the nonlinear feature mapping is represented as the following equation:

$$\mathbf{H}^{(v)} = g\left(\mathbf{X}^{(v)^T}\mathbf{W}^{(v)} + \mathbf{1}_N \mathbf{a}^{(v)^T}\right); \tag{6}$$

where $\mathbf{W}^{(v)} \in \mathbb{R}^{D_v \times L}$ is the constrained input weight matrix and $\boldsymbol{a}^{(v)} \in \mathbb{R}^L$ is the constrained hidden layer bias vector of the vth view in the ELM network. (The learning process for $\mathbf{W}^{(v)}$ and $\boldsymbol{a}^{(v)}$ can refer to Section 4.1.1.) $\mathbf{1}_N \in \mathbb{R}^N$ is the constant vector of all 1's and $g(\cdot)$ is an activation function for hidden layer. $\mathbf{H}^{(v)} \in \mathbb{R}^{N \times L}$ is the nonlinear representation of the vth view $\mathbf{X}^{(v)}$, where the jth row and ith column element can be represented by $h_i(\mathbf{x}_j^{(v)}) = g(\mathbf{x}_j^{(v)^T}\mathbf{w}_i + a_i)$.

The nonlinear representation $\mathbf{H}^{(v)}$ of the vth view is also called the hidden layer output matrix. Unlike the unknown feature mapping in other learning models, feature mapping in ELM is explicit, and it varies with different parameter settings as well as different activation functions. ELM feature mapping is a simple technique, but it can nonlinearly transform the original data into another feature space. Further, it can reduce noise while retaining informative information. Existing research also proves that learning models with ELM feature mapping can produce satisfactory performance [13, 23]

To formulate the proposed MVEC learning model, the first step is to learn view-independent embeddings for all views within the entire ELM network. This is accomplished with

$$\min_{\mathbf{B}^{(v)},\mathbf{Z}^{(v)}} \sum_{v=1}^{V} \left\|\mathbf{H}^{(v)}\mathbf{B}^{(v)} - \mathbf{Z}^{(v)}\right\|_F^2 + \lambda \sum_{v=1}^{V} \left\|\mathbf{B}^{(v)}\right\|_F^2; \tag{7}$$

where $\mathbf{H}^{(v)} \in \mathbb{R}^{N \times L}$ is the nonlinear representation obtained from the ELM feature mapping. $\mathbf{B}^{(v)} \in \mathbb{R}^{L \times M}$ is the output weight matrix of the ELM network, and $\mathbf{Z}^{(v)} \in \mathbb{R}^{N \times M}$ is the view-independent embedding learned by the whole ELM network. In Equation (7), the first term is the loss function to simulate the relation between the nonlinear representations and the view-independent representations. The second term is the regularization term to avoid overfitting. The parameter λ is used to balance the loss function and the regularization term. Instead of using original data, view-independent embedding learning can maximize the data information within each individual view. This is an appealing characteristic for diverse learning tasks. Due to the nature of neural network structures, ELMs have outstanding representation ability across a diverse range of learning applications, as proven in [16]. By using the constrained network parameters explained in the previous section, the ELM is able to learn more discriminative representations from the original multi-view data. Further, this style of learning process provides a solid foundation for exploring a commonly shared embedding across the different individual representations, which is the core idea behind MVEC.

4.1.3 Consensus Embedding Learning. With multi-view data, different views often provide information about complementarity and/or consistency in the data. Learning a common graph or subspace that captures these properties is prevalent in the literature [19, 21]. The main idea of our method is to learn a shared embedding via an ELM network structure that is able to handle a variety of different clustering tasks. Consensus embedding learning is a crucial part in MVEC, which explores the complementary and/or consistent information within multi-view data for clustering. To develop a commonly shared embedding, the learning model is formulated according to the nonlinear view embeddings $\mathbf{Z}^{(v)}$ described in Section 4.1.2. The commonly shared embedding is represented as

$$\min_{\mathbf{Z}^{(v)},\mathbf{Z}^*} \sum_{v=1}^{V} \left\|\mathbf{Z}^{(v)} - \mathbf{Z}^*\right\|_F; \tag{8}$$

where $\mathbf{Z}^* \in \mathbb{R}^{N\times M}$ is the commonly shared embedding learned across all views. This model minimizes the difference between the view-individual embeddings $\mathbf{Z}^{(v)}$ and the shared embedding $\mathbf{Z}^*$, which is helpful for obtaining a common, discriminative, and informative representation of the given multi-view data.

The formulation in Equation (8) is not convex nor smooth. Therefore, it is difficult to solve Equation (8) in an effective manner. It is feasible to rewrite Equation (8) as a re-weighted formulation, which is well-accepted in multi-view learning models [45] because a re-weighted scheme can automatically assign appropriate weights to different views to measure their contributions to the learning task. Inspired by this, we have rewritten Equation (8) as a more tractable formulation:

$$\min_{\mathbf{Z}^{(v)},\mathbf{Z}^*} \sum_{v=1}^{V} \beta_v \left\|\mathbf{Z}^{(v)} - \mathbf{Z}^*\right\|_F^2; \tag{9}$$

where β_v is denoted by the form below and is the weight for vth view and determined automatically from the current $\mathbf{Z}^{(v)}$ and $\mathbf{Z}^*$.

$$\beta_v = \left(2\left\|\mathbf{Z}^{(v)} - \mathbf{Z}^*\right\|_F\right)^{-1}. \tag{10}$$

The self-weighting strategy to measure the contribution of each individual view is recently well-accepted in multi-view learning tasks, which avoids introducing additional parameters. For clustering problem, the auto-weighted multiple graph learning method in [26] is a parameter-free method, which determines a proper weight for each graph automatically without introducing extra parameters. The self-weighted multi-view clustering method in [27] explores a Laplacian rank-constrained graph to be the centroid of the affiliation graph for each view with different confidences. For feature extraction problem, the multi-view unsupervised feature extraction method in [54] is presented to learn low-dimensional features for multi-view data with structured graphs and determine suitable weights for each view automatically without requiring an additive parameter. For feature selection problem, the multi-view unsupervised feature selection method in [12] leverages the learning mechanism to characterize the common structures with adaptive similarities and view weights for selecting informative features from multi-view data. The self-weighting strategy in multi-view learning tasks can be refer to other literatures.

4.1.4 The Objective Function.

In the previous sections, we introduced each component of the MVEC learning model, i.e., Equation (7) and Equation (9), and MVEC can be regarded as a two-step method. It is clear that these two steps performed separately cannot guarantee an optimal clustering result. We integrate Equation (7) and Equation (9) by introducing the parameter γ to balance their relationship. Thus, the optimization framework for MVEC is as follows:

$$\min_{\mathbf{B}^{(v)},\mathbf{Z}^{(v)},\mathbf{Z}^*} \sum_{v=1}^{V} \left\|\mathbf{H}^{(v)}\mathbf{B}^{(v)} - \mathbf{Z}^{(v)}\right\|_F^2 + \gamma \sum_{v=1}^{V} \beta_v \left\|\mathbf{Z}^{(v)} - \mathbf{Z}^*\right\|_F^2 + \lambda \sum_{v=1}^{V} \left\|\mathbf{B}^{(v)}\right\|_F^2, \tag{11}$$

where the first and third terms are used to learn the representative embeddings from each view through the ELM network, and the second term is used to learn the commonly-shared embedding from the individual embeddings. This joint optimization formulation automatically calculates the weights for each individual view, i.e., the measure of each view's importance to the learning task.

Consequently, any clustering method (e.g., K-means, fuzzy C-Means, spectral clustering, etc.) could easily be incorporated into MVEC. In this paper, we have used K-means to conduct the clustering because this method is able to reveal the underlying data structures in multi-view data.

4.2 Optimization for MVEC

This section demonstrates an alternating algorithm to the traditional approach for solving Equation (11). For ease of representation, Equation (11) is denoted as

$$\mathcal{F}\left(\mathbf{B}^{(v)}, \mathbf{Z}^{(v)}, \mathbf{Z}^*\right) = \sum_{v=1}^{V} \left\|\mathbf{H}^{(v)}\mathbf{B}^{(v)} - \mathbf{Z}^{(v)}\right\|_F^2 + \gamma \sum_{v=1}^{V} \beta_v \left\|\mathbf{Z}^{(v)} - \mathbf{Z}^*\right\|_F^2 + \lambda \sum_{v=1}^{V} \left\|\mathbf{B}^{(v)}\right\|_F^2. \tag{12}$$

It is clear that the above optimization formulation is convex if we update one variable while fixing the other two variables. This procedure is repeated until convergence. The pseudocode for the MVEC algorithm is presented above as Algorithm 2.

(1) Updating $\mathbf{B}^{(v)}$ *with a Fixed* $\mathbf{Z}^{(v)}$ *and* $\mathbf{Z}^*$. To calculate the derivative of the objective function w.r.t $\mathbf{B}^{(v)}$, we fix $\mathbf{Z}^{(v)}$ and $\mathbf{Z}^*$ and remove irrelevant items. The optimization problem for updating $\mathbf{B}^{(v)}$ becomes:

$$\mathcal{F}\left(\mathbf{B}^{(v)}\right) = \sum_{v=1}^{V} \left\|\mathbf{H}^{(v)}\mathbf{B}^{(v)} - \mathbf{Z}^{(v)}\right\|_F^2 + \lambda \sum_{v=1}^{V} \left\|\mathbf{B}^{(v)}\right\|_F^2. \tag{13}$$

Taking the derivative of Equation (13) w.r.t $\mathbf{B}^{(v)}$ and setting the derivative to zero, we have

$$\frac{\partial \mathcal{F}\left(\mathbf{B}^{(v)}\right)}{\partial \mathbf{B}^{(v)}} = 2\mathbf{H}^{(v)^{\mathrm{T}}}\left(\mathbf{H}^{(v)}\mathbf{B}^{(v)} - \mathbf{Z}^{(v)}\right) + 2\lambda\mathbf{B}^{(v)} = 0; \tag{14}$$

ALGORITHM 2: MVEC

Input: Multi-view data $\mathcal{X} = \{\mathbf{X}^{(v)}\}_{v=1}^{V}$, where $\mathbf{X}^{(v)} \in \mathbb{R}^{D_v \times N}$, Parameters γ and λ, number of clusters C, number of hidden nodes L, dimension of embedding M;

Output: Consensus embedding matrix $\mathbf{Z}^*$;

```
for $v = 1$ to $V$ do
    Initialize $\mathbf{W}^{(v)} \in \mathbb{R}^{D_v \times L}$ and $\mathbf{a}^{(v)} \in \mathbb{R}^{L}$ by Equation (1) and Equation (5);
    Calculate $\mathbf{H}^{(v)} = g(\mathbf{X}^{(v)^T}\mathbf{W}^{(v)} + \mathbf{1}_N\mathbf{a}^{(v)^T})$;
    Initialize $\mathbf{B}^{(v)} \in \mathbb{R}^{L \times M}$, and $\mathbf{Z}^{(v)} \in \mathbb{R}^{N \times M}$ with randomization;
end for
Initialize $\mathbf{Z}^* \in \mathbb{R}^{N \times M}$ with randomization;
repeat
    for $v = 1$ to $V$ do
        $\mathbf{B}^{(v)} = (\mathbf{H}^{(v)^{\mathrm{T}}}\mathbf{H}^{(v)} + \lambda\mathbf{I})^{-1}\mathbf{H}^{(v)^{\mathrm{T}}}\mathbf{Z}^{(v)}$;
        $\mathbf{Z}^{(v)} = \frac{\mathbf{H}^{(v)}\mathbf{B}^{(v)} + \gamma\beta_v\mathbf{Z}^*}{1+\gamma\beta_v}$;
    end for
    $\mathbf{Z}^* = \frac{\sum_{v=1}^{V}\beta_v\mathbf{Z}^{(v)}}{\sum_{v=1}^{V}\beta_v}$;
    for $v = 1$ to $V$ do
        Update the parameter $\beta_v$ by Equation (10);
    end for
until Convergence
Apply $K$-means to conduct clustering on the consensus embedding matrix $\mathbf{Z}^*$;
```

This derives a closed-form solution to update $\mathbf{B}^{(v)}$,

$$\mathbf{B}^{(v)} = \left(\mathbf{H}^{(v)^{\mathrm{T}}}\mathbf{H}^{(v)} + \lambda\mathbf{I}\right)^{-1}\mathbf{H}^{(v)^{\mathrm{T}}}\mathbf{Z}^{(v)}. \tag{15}$$

(2) Updating $\mathbf{Z}^{(v)}$ with a fixed $\mathbf{B}^{(v)}$ and $\mathbf{Z}^*$. To calculate the derivative of the objective function w.r.t $\mathbf{Z}^{(v)}$, we fix $\mathbf{B}^{(v)}$ and $\mathbf{Z}^*$ and remove irrelevant items. The optimization problem for updating $\mathbf{Z}^{(v)}$ becomes:

$$\mathcal{F}\left(\mathbf{Z}^{(v)}\right) = \sum_{v=1}^{V}\left\|\mathbf{H}^{(v)}\mathbf{B}^{(v)} - \mathbf{Z}^{(v)}\right\|_F^2 + \gamma\sum_{v=1}^{V}\beta_v\left\|\mathbf{Z}^{(v)} - \mathbf{Z}^*\right\|_F^2. \tag{16}$$

Taking the derivative of Equation (16) w.r.t $\mathbf{Z}^{(v)}$ and setting the derivative to zero, we have

$$\frac{\partial\mathcal{F}\left(\mathbf{Z}^{(v)}\right)}{\partial\mathbf{Z}^{(v)}} = 2\left(\mathbf{Z}^{(v)} - \mathbf{H}^{(v)}\mathbf{B}^{(v)}\right) + 2\gamma\beta_v\left(\mathbf{Z}^{(v)} - \mathbf{Z}^*\right) = 0. \tag{17}$$

This derives a closed-form solution to update $\mathbf{Z}^{(v)}$,

$$\mathbf{Z}^{(v)} = \frac{\mathbf{H}^{(v)}\mathbf{B}^{(v)} + \gamma\beta_v\mathbf{Z}^*}{1 + \gamma\beta_v}. \tag{18}$$

(3) Updating $\mathbf{Z}^*$ with a fixed $\mathbf{B}^{(v)}$ and $\mathbf{Z}^{(v)}$. To calculate the derivative of the objective function w.r.t $\mathbf{Z}^*$, we fix $\mathbf{B}^{(v)}$ and $\mathbf{Z}^{(v)}$ and remove irrelevant items. The optimization problem for updating W becomes:

$$\mathcal{F}\left(\mathbf{Z}^*\right) = \gamma\sum_{v=1}^{V}\beta_v\left\|\mathbf{Z}^{(v)} - \mathbf{Z}^*\right\|_F^2. \tag{19}$$

Taking the derivative of Equation (19) w.r.t $\mathbf{Z}^*$ and setting the derivative to zero, we have

$$\frac{\partial\mathcal{F}\left(\mathbf{Z}^*\right)}{\partial\mathbf{Z}^*} = 2\sum_{v=1}^{V}\beta_v\left(\mathbf{Z}^{(v)} - \mathbf{Z}^*\right) = 0; \tag{20}$$

This derives a closed-form solution to update $\mathbf{Z}^*$,

$$\mathbf{Z}^* = \frac{\sum_{v=1}^{V}\beta_v\mathbf{Z}^{(v)}}{\sum_{v=1}^{V}\beta_v}. \tag{21}$$

In both Equation (18) and Equation (21), the parameter β_v can be calculated using Equation (10). Overall, updating one variable by fixing the other two produces a solution in closed form. In Algorithm 2, the objective function value of Equation (12) monotonically decreases when using the alternating update rules until the algorithm converges. Successful convergence is theoretically proven in Section 5.1 and experimentally verified in Section 6.6.

5 ANALYSES FOR MVEC

5.1 Convergence Analysis

This section includes an analysis of Algorithm 2 in Section 5.1, followed by a time complexity analysis in Section 5.2.

Theorem 1. *Updating $\mathbf{B}^{(v)}$ using Equation (15) will monotonically decreases the objective function in Equation (12).*

Proof. Equation (15) is the solution to the following problem,

$$\min_{\mathbf{B}^{(v)}}\left\|\mathbf{H}^{(v)}\mathbf{B}^{(v)} - \mathbf{Z}^{(v)}\right\|_F^2 + \lambda\left\|\mathbf{B}^{(v)}\right\|_F^2. \tag{22}$$

It is obvious that the above function describes a convex problem, where the the optimal solution for $\mathbf{B}^{(v)}$ can be obtained by calculating the derivative of the function w.r.t $\mathbf{B}^{(v)}$. Therefore, we can prove the following inequation holds in the tth step,

$$\begin{aligned}&\left\|\mathbf{H}^{(v)}\mathbf{B}^{(v)^{t+1}} - \mathbf{Z}^{(v)^{t}}\right\|_F^2 + \lambda\left\|\mathbf{B}^{(v)^{t+1}}\right\|_F^2 \\ &\le \left\|\mathbf{H}^{(v)}\mathbf{B}^{(v)^{t}} - \mathbf{Z}^{(v)^{t}}\right\|_F^2 + \lambda\left\|\mathbf{B}^{(v)^{t}}\right\|_F^2.\end{aligned} \tag{23}$$

Combining Equation (12) and Equation (23), we have

$$\mathcal{F}\left(\mathbf{B}^{(v)^{t+1}}, \mathbf{Z}^{(v)^{t}}, \mathbf{Z}^{*t}\right) \le \mathcal{F}\left(\mathbf{B}^{(v)^{t}}, \mathbf{Z}^{(v)^{t}}, \mathbf{Z}^{*t}\right). \tag{24}$$

Thus, $\mathcal{F}(\mathbf{B}^{(v)}, \mathbf{Z}^{(v)}, \mathbf{Z}^{*})$ monotonically decreases using the updating rule in Equation (15) and Theorem 1 is proved. □

Theorem 2. *Updating $\mathbf{Z}^{(v)}$ using Equation (18) will monotonically decreases the objective function in Equation (12).*

Proof. Equation (18) is the solution of the following problem:

$$\min_{\mathbf{Z}^{(v)}} \left\|\mathbf{H}^{(v)}\mathbf{B}^{(v)} - \mathbf{Z}^{(v)}\right\|_F^2 + \lambda\beta_v\left\|\mathbf{Z}^{(v)} - \mathbf{Z}^{*}\right\|_F^2. \tag{25}$$

Denote $\mathcal{G}(\mathbf{B}^{(v)}, \mathbf{Z}^{(v)}, \mathbf{Z}^{*})$ as

$$\mathcal{G}\left(\mathbf{B}^{(v)}, \mathbf{Z}^{(v)}, \mathbf{Z}^{*}\right) = \frac{1}{\lambda}\left\|\mathbf{H}^{(v)}\mathbf{B}^{(v)} - \mathbf{Z}^{(v)}\right\|_F^2. \tag{26}$$

Then, Equation (33) can be rewritten as:

$$\min_{\mathbf{Z}^{(v)}} \beta_v\left\|\mathbf{Z}^{(v)} - \mathbf{Z}^{*}\right\|_F^2 + \mathcal{G}\left(\mathbf{B}^{(v)}, \mathbf{Z}^{(v)}, \mathbf{Z}^{*}\right). \tag{27}$$

Note that $\beta_v = (2\|\mathbf{Z}^{(v)} - \mathbf{Z}^{*}\|_F)^{-1}$, and we can derive

$$\begin{aligned}&\frac{\left\|\mathbf{Z}^{(v)^{t+1}} - \mathbf{Z}^{*t}\right\|_F^2}{2\left\|\mathbf{Z}^{(v)^{t+1}} - \mathbf{Z}^{*t}\right\|_F} + \mathcal{G}\left(\mathbf{B}^{(v)^{t+1}}, \mathbf{Z}^{(v)^{t+1}}, \mathbf{Z}^{*t}\right) \\ &\le \frac{\left\|\mathbf{Z}^{(v)^{t}} - \mathbf{Z}^{*t}\right\|_F^2}{2\left\|\mathbf{Z}^{(v)^{t}} - \mathbf{Z}^{*t}\right\|_F} + \mathcal{G}\left(\mathbf{B}^{(v)^{t+1}}, \mathbf{Z}^{(v)^{t}}, \mathbf{Z}^{*t}\right).\end{aligned} \tag{28}$$

According to a lemma in [25], for any non-zero matrix $\mathbf{P}$ and $\mathbf{Q}$, the following inequality holds:

$$\|\mathbf{P}\|_F - \frac{\|\mathbf{P}\|_F^2}{2\|\mathbf{Q}\|_F} \le \|\mathbf{Q}\|_F - \frac{\|\mathbf{Q}\|_F^2}{2\|\mathbf{Q}\|_F}. \tag{29}$$

Therefore, we can derive

$$\begin{aligned}&\left\|\mathbf{Z}^{(v)^{t+1}} - \mathbf{Z}^{*t}\right\|_F - \frac{\left\|\mathbf{Z}^{(v)^{t+1}} - \mathbf{Z}^{*t}\right\|_F^2}{2\left\|\mathbf{Z}^{(v)^{t}} - \mathbf{Z}^{*t}\right\|_F} \\ &\le \left\|\mathbf{Z}^{(v)^{t}} - \mathbf{Z}^{*t}\right\|_F - \frac{\left\|\mathbf{Z}^{(v)^{t}} - \mathbf{Z}^{*t}\right\|_F^2}{2\left\|\mathbf{Z}^{(v)^{t}} - \mathbf{Z}^{*t}\right\|_F}.\end{aligned} \tag{30}$$

Summing Equation (28) and Equation (30) on both sides, we have

$$\begin{aligned}&\left\|\mathbf{Z}^{(v)^{t+1}} - \mathbf{Z}^{*t}\right\|_F + \mathcal{G}\left(\mathbf{B}^{(v)^{t+1}}, \mathbf{Z}^{(v)^{t+1}}, \mathbf{Z}^{*t}\right)\\ &\quad\leq \left\|\mathbf{Z}^{(v)^{t}} - \mathbf{Z}^{*t}\right\|_F + \mathcal{G}\left(\mathbf{B}^{(v)^{t+1}}, \mathbf{Z}^{(v)^{t}}, \mathbf{Z}^{*t}\right).\end{aligned} \tag{31}$$

The inequality illustrates that the objective function of Equation (12) will monotonically decrease in each iteration. Combining Equation (31) and Equation (12), we have

$$\mathcal{F}\left(\mathbf{B}^{(v)^{t+1}}, \mathbf{Z}^{(v)^{t+1}}, \mathbf{Z}^{*t}\right) \leq \mathcal{F}\left(\mathbf{B}^{(v)^{t+1}}, \mathbf{Z}^{(v)^{t}}, \mathbf{Z}^{*t}\right). \tag{32}$$

Thus, $\mathcal{F}(\mathbf{B}^{(v)}, \mathbf{Z}^{(v)}, \mathbf{Z}^{*})$ monotonically decreases using the updating rule in Equation (18) and Theorem 2 is proved. A similar proof can be also found in [54]. □

Theorem 3. *Updating $\mathbf{Z}^{*}$ using Equation (21) will monotonically decreases the objective function in Equation (12).*

Proof. Equation (21) is the solution to the following problem,

$$\min_{\mathbf{Z}^{*}} \sum_{v=1}^{V} \beta_v \left\|\mathbf{Z}^{(v)} - \mathbf{Z}^{*}\right\|_F^2. \tag{33}$$

Equation (33) is partitioned into V subproblems. Following a similar procedure to that in Theorem 2, we can prove each subproblem is a convex problem w.r.t $\mathbf{Z}^{*}$, which means the objective function monotonically decreases when using the updating rule in Equation (21). Therefore, we can prove the following inequation holds in the tth step:

$$\sum_{v=1}^{V} \beta_v \left\|\mathbf{Z}^{(v)^{t+1}} - \mathbf{Z}^{*t+1}\right\|_F^2 \leq \sum_{v=1}^{V} \beta_v \left\|\mathbf{Z}^{(v)^{t+1}} - \mathbf{Z}^{*t}\right\|_F^2, \tag{34}$$

which is same as:

$$\sum_{v=1}^{V} \left\|\mathbf{Z}^{(v)^{t+1}} - \mathbf{Z}^{*t+1}\right\|_F \leq \sum_{v=1}^{V} \left\|\mathbf{Z}^{(v)^{t+1}} - \mathbf{Z}^{*t}\right\|_F. \tag{35}$$

Combining Equation (12) and Equation (35), we have

$$\mathcal{F}\left(\mathbf{B}^{(v)^{t+1}}, \mathbf{Z}^{(v)^{t+1}}, \mathbf{Z}^{*t+1}\right) \leq \mathcal{F}\left(\mathbf{B}^{(v)^{t+1}}, \mathbf{Z}^{(v)^{t+1}}, \mathbf{Z}^{*t}\right). \tag{36}$$

Thus, $\mathcal{F}(\mathbf{B}^{(v)}, \mathbf{Z}^{(v)}, \mathbf{Z}^{*})$ monotonically decreases using the updating rule in Equation (21) and Theorem 3 is proved. □

Combining Equation (24), Equation (32), and Equation (36), we can get

$$\begin{aligned}&\mathcal{F}\left(\mathbf{B}^{(v)^{t+1}}, \mathbf{Z}^{(v)^{t+1}}, \mathbf{Z}^{*t+1}\right) \leq \mathcal{F}\left(\mathbf{B}^{(v)^{t+1}}, \mathbf{Z}^{(v)^{t+1}}, \mathbf{Z}^{*t}\right)\\ &\leq \mathcal{F}\left(\mathbf{B}^{(v)^{t+1}}, \mathbf{Z}^{(v)^{t}}, \mathbf{Z}^{*t}\right) \leq \mathcal{F}\left(\mathbf{B}^{(v)^{t}}, \mathbf{Z}^{(v)^{t}}, \mathbf{Z}^{*t}\right).\end{aligned} \tag{37}$$

Therefore, the alternating update rules in Algorithm 2 have been proven to monotonically decrease the objective function of Equation (12).

Table 2. Statistics for the Eight Multi-View Datasets Used in the Experiments

Feature Type	BBCSport	Blog	FOX	CNN	WebKB	3Sources	Yale	Digits
1	TextA(3183)	Text(5390)	Text(2711)	Text(3695)	Content(3000)	BBC(3560)	Intensity(4096)	FAC(216)
2	TextB(3203)	Tag(2003)	Image(996)	Image(996)	Link(1084)	Reuters(3068)	LBP(3304)	FOU(76)
3	-	-	-	-	-	Guardian(3631)	Gabor(6750)	KAR(64)
4	-	-	-	-	-	-	-	PIX(240)
Instances	544	1000	1523	2107	1051	169	165	2000
Clusters	5	5	4	7	2	6	15	10

5.2 Time Complexity Analysis

In our case, $M \leq L$ and $M \leq N$. The calculations for $\mathbf{H}^{(v)}$, $\mathbf{B}^{(v)}$, $\mathbf{Z}^{(v)}$ and $\mathbf{Z}^*$ are the contributors to MVEC's time complexity. It takes $O(N{\cdot}L{\cdot}D_v)$ to calculate $\mathbf{H}^{(v)}$. The time complexity for calculating $\mathbf{B}^{(v)}$ is $O(L^3 + N{\cdot}L^2 + N{\cdot}L{\cdot}M)$, and the cost for calculating $\mathbf{Z}^{(v)}$ is $O(N{\cdot}L{\cdot}M)$. Calculating $\mathbf{Z}^*$ is computationally trivial. Assume that T is the number of iterations, then the overall cost for the proposed MVEC can simply be denoted as $O(T(L^3 + N{\cdot}L^2))$. In general, parameter L does not need to be very large, so the proposed MVEC method can be applied to diverse application problems with a desirable efficiency.

6 EXPERIMENTS

The experiments we conducted to validate the effectiveness and efficiency of MVEC involved multi-view clustering tasks with a range of real-world datasets. All experiments were performed on a PC with an Intel(R) Core(TM) 2.70GHZ CPU and 8GB of RAM using Matlab R2016a.

6.1 Dataset

We selected eight publicly available real-world multi-view datasets, each drawn from different domains and/or data types including text, numerals, and images. Using a diverse selection of datasets provides a comprehensive evaluation of the algorithm's performance. Statistics for each of the eight datasets are summarized in Table 2, and a brief description follows.

BBCSport. A dataset that includes 544 sports news articles with five different categories. This is a synthetic dataset with two textual views.

Blog. A dataset crawled from Blogcatalog. It includes 1,000 instances with five different categories. The two views are text in posts and the tags related to the posts.

FOX. A dataset comprising 1,523 news articles with four distinct categories obtained from FOX and two views – one image, one text.

CNN. A news dataset collected from CNN that contains 2,107 news articles in seven clusters. Each article is described with text and images.

WebKB. 1,051 web pages from four different universities, with each represented as a content view and a link view.

3Sources. A dataset of 169 shared stories across six diverse topics. All stories were reported by three news agencies, (BBC, Reuters, and The Guardian), which represent the three views.

Yale. A dataset of 165 images of human faces that have each been described by 15 different people. Intensity, local binary patterns (LBP), and Gabor feature views have been extracted from each image.

Digits. A dataset that includes ten different categories of 2,000 handwritten digits from "0" to "9". Four of its feature views were used in our experiments: FAC, FOU, KAR, and PIX.

6.2 Baseline Methods

To assess MVEC's effectiveness, we compared MVEC with both ELM-based and non-ELM-based clustering methods. With the single view methods, multiple view features were concatenated into a unified feature space. Each of the baseline methods is briefly described below.

ELM-Based Clustering Methods. **(a) ELMCKM** is a single-view clustering method based on Fisher's linear discriminant analysis (LDA) and solved with a K-means kernel [13]; **(b) US-ELM** is a single-view clustering method that extends ELM to unsupervised scenarios with manifold regularization [14]; **(c) ELMJEC** is a single-view clustering method based on a discriminative embedded clustering method that simultaneously learns the embeddings and clustering tasks [23]; **(d) CoregSC-ELM** extends the co-regularized multi-view spectral clustering method to ELM feature representation rather than using the original data [41]; **(e) MMSC-ELM** is an extension to the multi-modal spectral clustering method, again, using ELM feature representations instead of the original data [41]; **(f) MRSC-ELM** extends the robust multi-view spectral clustering method with ELM feature representation instead of using the original data [41].

Non-ELM-Based Clustering Methods. **(a) SEC** is a single view spectral clustering method that imposes a linearity regularization on the objective function [28]; **(b) AMGL** is a single view spectral clustering method that imposes a linearity regularization on the objective function [26]; **(c) MultiNMF** is a multi-view clustering method that searches for a compatible clustering solution via joint nonnegative matrix factorization [21]; **(d) RMKMC** is a multi-view K-means clustering method that learns a shared cluster indicator to solve different problems [3]; **(e) CoregSC** is a centroid based multi-view clustering method by learning a consensus clustering representation across views [19]; **(f) MMSC** is a multi-modal spectral clustering method to explore a common graph Laplacian matrix from multiple views [4].

6.3 Evaluation Metrics

We evaluated each of the different algorithms according to their clustering accuracy (ACC) and normalized mutual information (NMI). These metrics are well-accepted criteria for assessing performance in unsupervised learning tasks [31, 37]. The larger the ACC and NMI, the better the clustering performance. The formulations for each metric follow.

Clustering Accuracy (ACC). Given a data point $\mathbf{x}_i$, let p_i be the clustering label and q_i be the true class label:

$$ACC = \frac{1}{N}\sum_{i=1}^{N}\delta(q_i, map(p_i)); \tag{38}$$

where $map(\cdot)$ is a permutation function that maps the label of each cluster to a ground truth label. The best permutation mapping can be found by the Kuhn-Munkres algorithm [30]. $\delta(q,p)$ is an indicator function, where the value is one if $q = p$ and zero otherwise.

Normalized Mutual Information (NMI): Let C^t be the ground truth clusters and C^e be the clusters produced by the algorithm. The mutual information is denoted as

$$MI(C^t, C^e) = \sum_{c_i^t \in C^t, c_j^e \in C^e} p\left(c_i^t, c_j^e\right) log \frac{p\left(c_i^t, c_j^e\right)}{p\left(c_i^t\right)p\left(c_j^e\right)}; \tag{39}$$

where $p(c_i^t)$ and $p(c_j^e)$ are the probabilities that a data point is arbitrarily selected from the cluster c_i^t and c_j^e, respectively. $p(c_i^t, c_j^e)$ denotes the joint probability that a data point belongs to both clusters

Table 3. Clustering Result Comparison of the ELM-Based Clustering Methods

Baseline Method	BBCSport	Blog	FOX	CNN	WebKB	3Sources	Yale	Digits
	ACC±std							
ELMCKM	0.390±0.062	0.361±0.052	0.460±0.020	0.240±0.016	0.909±0.071	0.314±0.038	0.432±0.041	0.531±0.051
US-ELM	0.549±0.045	0.493±0.047	0.474±0.040	0.260±0.006	0.882±0.045	0.472±0.033	0.521±0.034	0.576±0.052
ELMJEC	0.345±0.004	0.330±0.022	0.460±0.027	0.231±0.016	0.784±0.039	0.317±0.022	0.573±0.019	0.807±0.074
CoregSC-ELM	0.552±0.035	0.502±0.013	0.606±0.001	0.289±0.013	0.907±0.008	0.500±0.042	0.574±0.062	0.779±0.063
MMSC-ELM	0.482±0.040	0.315±0.024	0.574±0.014	0.255±0.015	0.810±0.002	0.442±0.023	0.584±0.038	0.706±0.069
MRSC-ELM	0.706±0.044	0.585±0.007	0.490±0.011	0.306±0.017	0.898±0.005	0.543±0.051	0.566±0.038	0.635±0.038
MVEC	**0.829±0.071**	**0.637±0.074**	**0.785±0.088**	**0.545±0.035**	**0.963±0.048**	**0.556±0.059**	**0.595±0.063**	**0.820±0.059**
	NMI±std							
ELMCKM	0.128±0.042	0.091±0.038	0.111±0.014	0.042±0.005	0.498±0.061	0.124±0.042	0.496±0.034	0.502±0.047
US-ELM	0.285±0.040	0.211±0.011	0.119±0.014	0.050±0.010	0.429±0.037	0.370±0.020	0.577±0.026	0.544±0.013
ELMJEC	0.045±0.062	0.054±0.017	0.123±0.030	0.035±0.008	0.085±0.029	0.092±0.031	0.610±0.014	0.766±0.037
CoregSC-ELM	0.267±0.013	0.259±0.027	0.264±0.001	0.100±0.008	0.503±0.002	0.488±0.035	0.619±0.028	0.745±0.024
MMSC-ELM	0.261±0.059	0.233±0.029	0.264±0.003	0.084±0.006	0.107±0.001	0.430±0.009	0.656±0.012	0.799±0.024
MRSC-ELM	0.431±0.030	0.320±0.029	0.162±0.002	0.119±0.011	0.509±0.003	0.470±0.036	0.599±0.037	0.596±0.012
MVEC	**0.728±0.061**	**0.483±0.025**	**0.644±0.043**	**0.376±0.019**	**0.726±0.001**	**0.551±0.038**	**0.660±0.030**	**0.802±0.027**

The best results are highlighted in bold.

c_i^t and c_j^e. The normalized mutual information used in our experiments is defined as follows:

$$NMI(C^t, C^e) = \frac{MI(C^t, C^e)}{\sqrt{H(C^t)H(C^e)}}; \tag{40}$$

where $H(C^t)$ and $H(C^e)$ are the entropies of C^t and C^e.

6.4 Experimental Setting

We employed a sigmoid function as the activation function for the hidden layer of the MVEC framework, which is a common choice in the literature [5, 49]. The number of hidden nodes was simply set to 200 for all benchmark datasets in the experiments. The dimensionality of the embedding was selected from $\{5, 50\}$ in units of 5. Additionally, we set the two parameters γ and λ using a grid-search strategy from $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. Note that these parameters need to be tuned for each dataset, so we used optimal values for each dataset to conduct the experiments. The results reported in the following sections were obtained by repeating a K-means algorithm 30 times.

6.5 Performance Comparison

We begin our discussion on performance with the comparisons between MVEC and the ELM-based clustering methods. Comparisons to the non-ELM-based methods are discussed in Section 6.5.2.

6.5.1 Comparison with ELM-Based Methods. Table 3 shows the clustering results for the ELM-based baselines. The best results appear in bold, which means the method in that row outperformed the other methods. From Table 3, we can see that MVEC produced promising results compared to the other ELM-based clustering methods in terms of both ACC and NMI. A summary of our observations follows.

- In terms of ACC, MVEC significantly outperformed the other ELM-based clustering methods on all multi-view datasets. The enhancement to ACC MVEC provided is obvious. For instance, on the CNN dataset, MVEC showed a 23% improvement to the ACC of the other methods.

Table 4. Clustering Result Comparison of the Non-ELM-based Clustering Methods

Baseline Method	BBCSport	Blog	FOX	CNN	WebKB	3Sources	Yale	Digits
	ACC±std							
SEC	0.726±0.028	0.586±0.036	0.478±0.001	0.267±0.010	0.942±0.006	0.439±0.032	0.537±0.049	0.704±0.054
AMGL	0.356±0.002	0.259±0.011	0.443±0.005	0.232±0.006	0.781±0.014	0.376±0.023	0.502±0.056	0.659±0.087
MultiNMF	0.473±0.027	0.635±0.046	0.767±0.027	0.442±0.033	0.793±0.035	0.536±0.044	0.541±0.052	0.671±0.073
RMKMC	0.458±0.006	0.376±0.004	0.420±0.007	0.211±0.003	0.952±0.002	0.521±0.006	0.479±0.007	0.601±0.002
CoregSC	0.598±0.079	0.601±0.021	0.748±0.001	0.363±0.021	0.954±0.001	0.549±0.040	0.575±0.051	0.788±0.072
MMSC	0.559±0.004	0.350±0.001	0.631±0.003	0.249±0.001	0.809±0.002	0.438±0.001	0.588±0.002	**0.836±0.001**
MVEC	**0.829±0.071**	**0.637±0.074**	**0.785±0.088**	**0.545±0.035**	**0.963±0.048**	**0.556±0.059**	**0.595±0.063**	0.820±0.059
	NMI±std							
SEC	0.608±0.016	0.418±0.019	0.155±0.001	0.056±0.001	0.646±0.003	0.362±0.041	0.601±0.029	0.756±0.022
AMGL	0.039±0.006	0.035±0.010	0.063±0.059	0.030±0.004	0.003±0.014	0.176±0.021	0.593±0.031	0.792±0.044
MultiNMF	0.342±0.029	0.468±0.040	0.615±0.012	0.359±0.018	0.382±0.029	0.492±0.051	0.602±0.023	0.665±0.026
RMKMC	0.353±0.001	0.338±0.001	0.114±0.004	0.024±0.005	0.701±0.002	0.336±0.003	0.550±0.002	0.635±0.001
CoregSC	0.427±0.029	0.466±0.033	0.498±0.001	0.182±0.017	0.711±0.003	0.528±0.027	0.632±0.041	0.773±0.033
MMSC	0.364±0.004	0.339±0.005	0.284±0.002	0.085±0.007	0.119±0.004	0.379±0.008	0.646±0.003	**0.823±0.005**
MVEC	**0.728±0.061**	**0.483±0.025**	**0.644±0.043**	**0.376±0.019**	**0.726±0.001**	**0.551±0.038**	**0.660±0.030**	0.802±0.027

The best results are highlighted in bold.

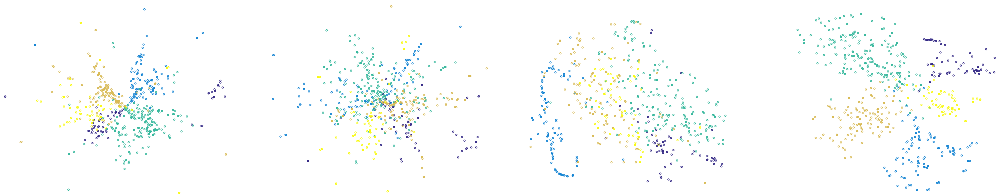
- In terms of NMI, MVEC greatly outperformed the other ELM-based clustering methods on all multi-view datasets, most notably the BBCSport datasets with an NMI improvement of more than 19%.
- ELMCKM's performance was poor. ELMJEC was also inferior to MVEC in joint embedding and clustering, which is mainly attributed to its neglect of the complementary information within multi-view data.
- The other ELM-based baselines simply extend existing multi-view clustering methods to a random feature mapping space and, therefore, these methods also demonstrated poor clustering performance.

The superiority of MVEC compared to these ELM-based clustering methods is clear from these results. We find that exploring and leveraging the complementary information within multi-view data is quite beneficial and can boost clustering performance with multi-view clustering tasks. Further, learning a shared embedding based on the original multi-view data via an ELM network also contributes to better performance.

6.5.2 Comparison with Non-ELM-Based Methods. Table 4 reports the comparison results for MVEC and the non-ELM-based clustering methods, with the best results in bold. As shown, MVEC produced outstanding ACC and NMI results compared to the other non-ELM-based clustering methods. Overall, the highlights are summarized as follows:

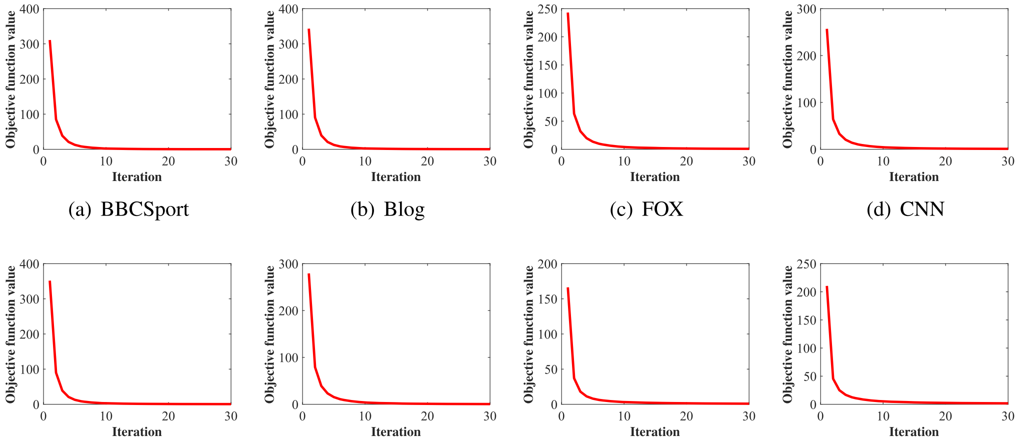
- Compared to AMGL, MVEC performed significantly better because MVEC learns a common embedding that considers the consistencies in the information when clustering.
- MVEC performed better than MultiNMF because MVEC reduces noise and retains helpful information when it learns the common embedding, which further improves the clustering results.
- Compared to CoregSC and MMSC, MVEC's higher performance on most of the datasets is mainly due to learning the common embedding via an ELM network structure.

Thus, the superiority of MVEC over non-ELM-based clustering methods on all the datasets is also verified through these comprehensive experiments, with one exception–the Digits dataset. Here, MMSC performed slightly better than MVEC because the characteristic indicators in the



(a) Original Data (b) USELM (c) CoregSC (d) MVEC

Fig. 3. Visualization of the original BBCSport data compared to the BBCSport data embedding learned by US-ELM, CoregSC, and MVEC.



(a) BBCSport (b) Blog (c) FOX (d) CNN

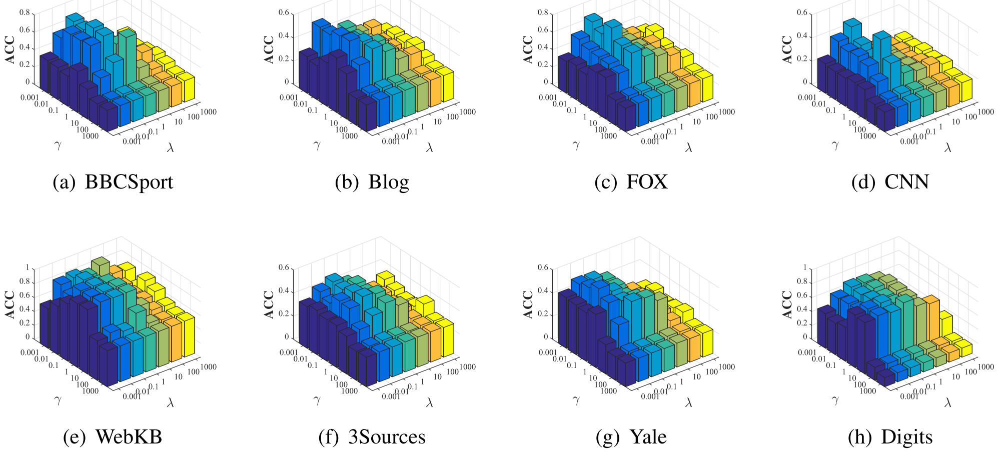
(e) WebKB (f) 3Sources (g) Yale (h) Digits

Fig. 4. Convergence learning curve of MVEC on all multi-view datasets.

original data are explicit, and MMSC directly benefits from this. Considering such data is not generally available in the real world, MMSC is difficult to generalize. MVEC achieved solid results over all the datasets, which serves as an indicator of its superior capacity for generalization. We conclude that nonlinear representation learning via an ELM network structure can be of benefit to a diverse range of multi-view clustering tasks. The illustrated embedding results from the BBCSport dataset shown in Figure 3, support this conclusion on an intuitive level, showing that MVEC is able to unearth more discriminative data structures than the other baseline methods.

6.6 Convergence Study

To solve Equation (12), we devised an alternating optimization algorithm to monotonically decrease the objective function value. Figure 4 shows the learning curves for all datasets from our experiments to test convergence. The default value for parameters γ and λ was 0.1. As the figure shows, the objective function values rapidly decreased and converged within 10 iterations on all datasets, and the iterative optimization algorithm quickly converged on a solution to Equation (12), which supports the algorithm's effectiveness and efficiency. Moreover, this convergence guarantee demonstrates that the results from MVEC will reach a sufficient level to at least approximate the optimal solution.



(a) BBCSport (b) Blog (c) FOX (d) CNN

(e) WebKB (f) 3Sources (g) Yale (h) Digits

Fig. 5. Variations in terms of MVEC's clustering accuracy w.r.t different values of the parameters γ and λ on all multi-view datasets.

6.7 Parameter Sensitivity Study

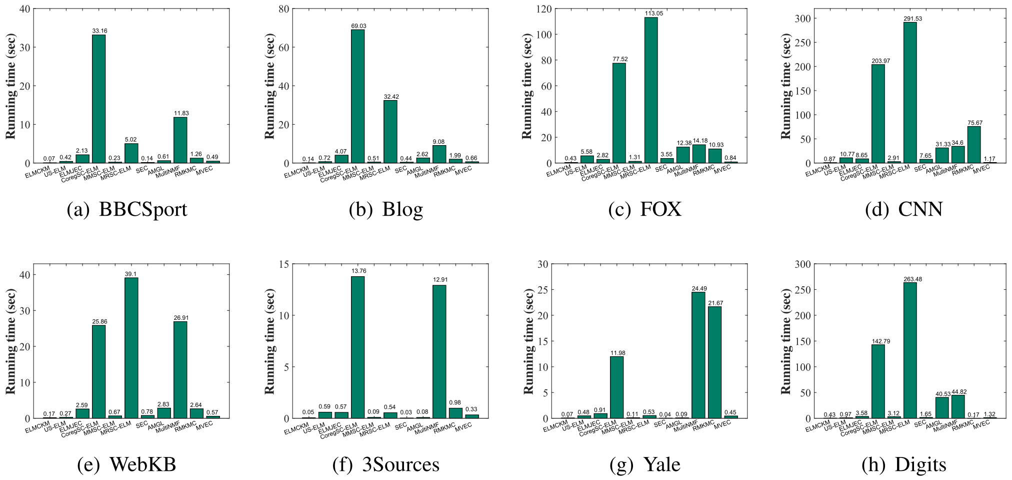
In MVEC, choosing suitable values for the parameters γ and λ is key to ensuring a good result. In our parameter sensitivity study of γ and λ, we used a grid-search strategy and varied the parameters γ and λ within $\{0.001, 0.01, 0.1, 1, 10, 100, 1,000\}$. The results in terms of clustering accuracy for all datasets are shown in Figure 5. From these figures, we observe that performance varied according to the value of both parameters, but MVEC's best performance on all datasets sat at the top-left corner. For example, MVEC's best performance on the BBCSport dataset occurred when we fixed $\lambda = 0.01$ or $\lambda = 0.1$ with a small γ. On the FOX dataset, $\lambda = 0.1$ returned the best performance. And, with the Digits dataset, setting $\gamma = 10$ produced very superior results. Other similar findings can be found in Figure 5. Therefore, it is imperative to determine which values for parameters γ and λ are most suitable for the given multi-view clustering task.

6.8 Running Time Study

The results of our running time study on all datasets are provided in Figure 6. For each compared method, the CPU running time varied according to the particular characteristics of the dataset. ELMCKM, MMSC-ELM, and MVEC returned superior performance in terms of CPU running time, while CoregSC-ELM, MRSC-ELM, and MultiNMF performed poorly. The running time for both US-ELM and RMKMC was inferior on the FOX and CNN datasets. But ELMJEC performed well on the 3Sources and Yale datasets. Overall, MVEC proved to be a computationally efficient method that shows good clustering performance with a fast learning speed.

6.9 Case Study on Network Parameter Initialization

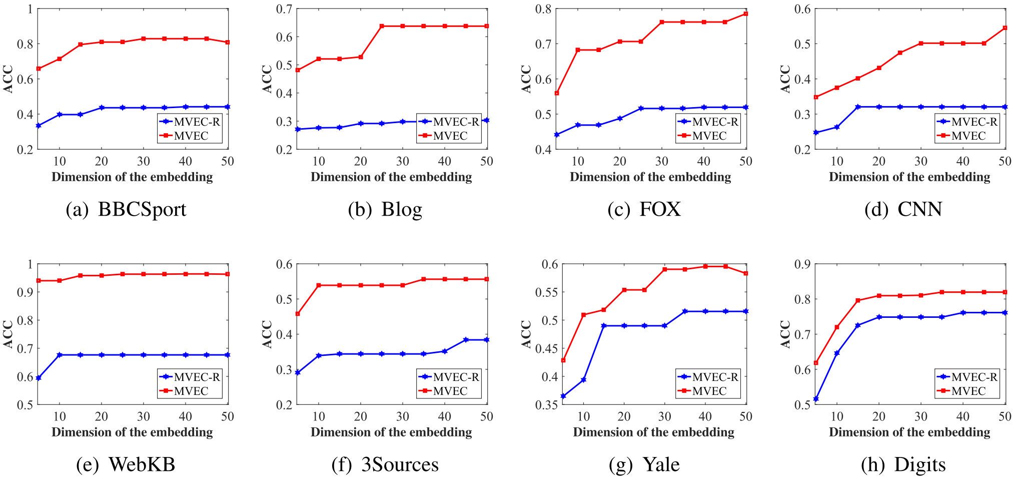
To assess the importance of our unsupervised parameter initialization technique, we compared two versions of MVEC, one with and one without unsupervised parameter initialization, denoted as MVEC-R. In MVEC-R, the initial input weights were selected randomly. Both methods were tested with learned embeddings of varying dimensions on all datasets. The results appear in Figure 7, where we observe that MVEC significantly outperformed MVEC-R. This demonstrates that our technique for parameter initialization takes full advantage of the underlying information in the original data, which tremendously benefits the model's clustering performance.



(a) BBCSport (b) Blog (c) FOX (d) CNN

(e) WebKB (f) 3Sources (g) Yale (h) Digits

Fig. 6. Performance comparison in terms of CPU running time (measured by second) on all multi-view datasets.



(a) BBCSport (b) Blog (c) FOX (d) CNN

(e) WebKB (f) 3Sources (g) Yale (h) Digits

Fig. 7. Case study on network parameter initialization for all multi-view datasets. MVEC and MVEC-R denote the proposed method with and without the proposed unsupervised parameter learning technique, respectively.

7 CONCLUSION

In this article, we proposed a novel multi-view fusion clustering framework, called MVEC, which benefits from the representation learning ability of an ELM. MVEC is designed to perform multi-view clustering tasks by constructing a unified embedding from the individual embeddings using an ELM network structure. We first learn individual embeddings from each view, and minimize the difference between view-independent embeddings and the commonly shared embedding to explore the commonly shared embedding. An effective and efficient alternating solution is introduced to solve the formulation of MVEC. The ability of the ELM to leverage the correlations and dependencies within multi-view data significantly improves the discriminatory power of the features, and comparative experiments on eight multi-view datasets demonstrate that MVEC improves clustering accuracy with fast running times compared to a range of baselines.

REFERENCES

[1] Steffen Bickel and Tobias Scheffer. 2004. Multi-view clustering. In *Proceedings of the 2004 International Conference on Data Mining*, Vol. 4. 19–26.

[2] Maria Brbić and Ivica Kopriva. 2018. Multi-view low-rank sparse subspace clustering. *Pattern Recognition* 73 (2018), 247–258.

[3] Xiao Cai, Feiping Nie, and Heng Huang. 2013. Multi-view K-means clustering on big data. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. 2598–2604.

[4] Xiao Cai, Feiping Nie, Heng Huang, and Farhad Kamangar. 2011. Heterogeneous image feature integration via multimodal spectral clustering. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. 1977–1984.

[5] Jiuwen Cao, Zhiping Lin, Guang-Bin Huang, and Nan Liu. 2012. Voting based extreme learning machine. *Information Sciences* 185, 1 (2012), 66–77.

[6] Kamalika Chaudhuri, Sham M. Kakade, Karen Livescu, and Karthik Sridharan. 2009. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th International Conference on Machine Learning*. 129–136.

[7] Hongchang Gao, Feiping Nie, Xuelong Li, and Heng Huang. 2015. Multi-view subspace clustering. In *Proceedings of the 2015 IEEE International Conference on Computer Vision*. 4238–4246.

[8] Derek Greene. 2009. A matrix factorization approach for integrating multiple data views. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*. 423–438.

[9] Saima Hassan, Mojtaba Ahmadieh Khanesar, Jafreezal Jaafar, and Abbas Khosravi. 2017. Comparative analysis of three approaches of antecedent part generation for an IT2 TSK FLS. *Applied Soft Computing* 51 (2017), 130–144.

[10] Lifang He, Chun-Ta Lu, Hao Ding, Shen Wang, Linlin Shen, Philip S. Yu, and Ann B. Ragin. 2017. Multi-way multilevel kernel modeling for neuroimaging classification. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. 356–364.

[11] Qing He, Xin Jin, Changying Du, Fuzhen Zhuang, and Zhongzhi Shi. 2014. Clustering in extreme learning machine feature space. *Neurocomputing* 128 (2014), 88–95.

[12] Chenping Hou, Feiping Nie, Tao Hong, and Dongyun Yi. 2017. Multi-view unsupervised feature selection with adaptive similarity and view weight. *IEEE Transactions on Knowledge and Data Engineering* 29, 9 (2017), 1998–2011.

[13] Gao Huang, Tianchi Liu, Yan Yang, Zhiping Lin, Shiji Song, and Cheng Wu. 2015. Discriminative clustering via extreme learning machine. *Neural Networks* 70 (2015), 1–8.

[14] Gao Huang, Shiji Song, Jatinder ND Gupta, and Cheng Wu. 2014. Semi-supervised and unsupervised extreme learning machines. *IEEE Transactions on Cybernetics* 44, 12 (2014), 2405–2417.

[15] G.-B. Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. 2012. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics* 42, 2 (2012), 513–529.

[16] Liyanaarachchi Lekamalage Chamara Kasun, Yan Yang, Guang-Bin Huang, and Zhengyou Zhang. 2016. Dimension reduction with extreme learning machine. *IEEE Transactions on Image Processing* 25, 8 (2016), 3906–3918.

[17] Liyanaarachchi Lekamalage Chamara Kasun, Hongming Zhou, Guang-Bin Huang, and Chi Man Vong. 2013. Representational learning with ELMs for big data. *IEEE Intelligent Systems* 28, 6 (2013), 31–34.

[18] Young-Min Kim, Massih-Reza Amini, Cyril Goutte, and Patrick Gallinari. 2010. Multi-view clustering of multilingual documents. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 821–822.

[19] Abhishek Kumar, Piyush Rai, and Hal Daume. 2011. Co-regularized multi-view spectral clustering. In *Advances in Neural Information Processing Systems*. 1413–1421.

[20] Shao-Yuan Li, Yuan Jiang, and Zhi-Hua Zhou. 2010. Partial multi-view clustering. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. 1968–1974.

[21] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. 2013. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. 252–260.

[22] Tianchi Liu, Chamara Kasun Liyanaarachchi Lekamalage, Guang-Bin Huang, and Zhiping Lin. 2018. An adaptive graph learning method based on dual data representations for clustering. *Pattern Recognition* 77 (2018), 126–139.

[23] Tianchi Liu, Chamara Kasun Liyanaarachchi Lekamalage, Guang-Bin Huang, and Zhiping Lin. 2018. Extreme learning machine for joint embedding and clustering. *Neurocomputing* 277 (2018), 78–88.

[24] Feiping Nie, Guohao Cai, Jing Li, and Xuelong Li. 2018. Auto-weighted multi-view learning for image clustering and semi-supervised classification. *IEEE Transactions on Image Processing* 27, 3 (2018), 1501–1511.

[25] Feiping Nie, Heng Huang, Xiao Cai, and Chris H. Ding. 2010. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *Advances in Neural Information Processing Systems*. 1813–1821.

[26] Feiping Nie, Jing Li, and Xuelong Li. 2016. Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. 1881–1887.

[27] Feiping Nie, Jing Li, and Xuelong Li. 2017. Self-weighted multiview clustering with multiple graphs. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2564–2570.

[28] Feiping Nie, Zinan Zeng, Ivor W. Tsang, Dong Xu, and Changshui Zhang. 2011. Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering. *IEEE Transactions on Neural Networks* 22, 11 (2011), 1796–1808.

[29] Huimin Pei, Kuaini Wang, Qiang Lin, and Ping Zhong. 2018. Robust semi-supervised extreme learning machine. *Knowledge-Based Systems* 159 (2018), 203–220.

[30] Michael D. Plummer and László Lovász. 1986. *Matching Theory*. Vol. 29. North-Holland Publishing Company, Amsterdam.

[31] Lei Shi, Liang Du, and Yi-Dong Shen. 2014. Robust spectral learning for unsupervised feature selection. In *Proceedings of the 2014 IEEE International Conference on Data Mining*. 977–982.

[32] Yu Su, Shiguang Shan, Xilin Chen, and Wen Gao. 2011. Classifiability-based discriminatory projection pursuit. *IEEE Transactions on Neural Networks* 22, 12 (2011), 2050–2061.

[33] Chang Tang, Jiajia Chen, Xinwang Liu, Miaomiao Li, Pichao Wang, Minhui Wang, and Peng Lu. 2018. Consensus learning guided multi-view unsupervised feature selection. *Knowledge-Based Systems* 160 (2018), 49–60.

[34] Chang Tang, Xinzhong Zhu, Xinwang Liu, Miaomiao Li, Pichao Wang, Changqing Zhang, and Lizhe Wang. 2019. Learning a joint affinity graph for multiview subspace clustering. *IEEE Transactions on Multimedia* 21, 7 (2019), 1724–1736. DOI: https://doi.org/10.1109/TMM.2018.2889560

[35] Chang Tang, Xinzhong Zhu, Xinwang Liu, and Lizhe Wang. 2019. Cross-view local structure preserved diversity and consensus learning for multi-view unsupervised feature selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 595–604.

[36] Jiexiong Tang, Chenwei Deng, and Guang-Bin Huang. 2016. Extreme learning machine for multilayer perceptron. *IEEE Transactions on Neural Networks and Learning Systems* 27, 4 (2016), 809–821.

[37] Jiliang Tang, Xia Hu, Huiji Gao, and Huan Liu. 2013. Unsupervised feature selection for multi-view data in social media. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. 270–278.

[38] Wei Tang, Zhengdong Lu, and Inderjit S. Dhillon. 2009. Clustering with multiple graphs. In *Proceedings of the 9th IEEE International Conference on Data Mining*. 1016–1021.

[39] Muhammad Uzair and Ajmal Mian. 2017. Blind domain adaptation with augmented extreme learning machine features. *IEEE Transactions on Cybernetics* 47, 3 (2017), 651–660.

[40] Hao Wang, Yan Yang, Bing Liu, and Hamido Fujita. 2019. A study of graph-based system for multi-view clustering. *Knowledge-Based Systems* 163 (2019), 1009–1019.

[41] Qiang Wang, Yong Dou, Xinwang Liu, Qi Lv, and Shijie Li. 2016. Multi-view clustering with extreme learning machine. *Neurocomputing* 214 (2016), 483–494.

[42] Zhelong Wang, Donghui Wu, Raffaele Gravina, Giancarlo Fortino, Yongmei Jiang, and Kai Tang. 2017. Kernel fusion based extreme learning machine for cross-location activity recognition. *Information Fusion* 37 (2017), 1–9.

[43] Chi Man Wong, Chi Man Vong, Pak Kin Wong, and Jiuwen Cao. 2018. Kernel-based multilayer extreme learning machines for representation learning. *IEEE Transactions on Neural Networks and Learning Systems* 29, 3 (2018), 757–762.

[44] Jinglin Xu, Junwei Han, and Feiping Nie. 2016. Discriminatively embedded k-means for multi-view clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5356–5364.

[45] Jinglin Xu, Junwei Han, Feiping Nie, and Xuelong Li. 2017. Re-weighted discriminatively embedded K-means for multi-view clustering. *IEEE Transactions on Image Processing* 26, 6 (2017), 3016–3027.

[46] Yimin Yang and Q. M. Jonathan Wu. 2016. Multilayer extreme learning machine with subnetwork nodes for representation learning. *IEEE Transactions on Cybernetics* 46, 11 (2016), 2570–2583.

[47] Yimin Yang, Q. M. Jonathan Wu, and Yaonan Wang. [n.d.]. Autoencoder with invertible functions for dimension reduction and image reconstruction. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48, 7 ([n.d.]), 1065–1079.

[48] Changqing Zhang, Qinghua Hu, Huazhu Fu, Pengfei Zhu, and Xiaochun Cao. 2017. Latent multi-view subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 30. 4279–4287.

[49] Yongshan Zhang, Jia Wu, Zhihua Cai, Peng Zhang, and Ling Chen. 2016. Memetic extreme learning machine. *Pattern Recognition* 58 (2016), 135–148.

[50] Yongshan Zhang, Jia Wu, Chuan Zhou, and Zhihua Cai. 2017. Instance cloned extreme learning machine. *Pattern Recognition* 68 (2017), 52–65.

[51] Wentao Zhu, Jun Miao, and Laiyun Qing. 2014. Constrained extreme learning machine: A novel highly discriminative random feedforward neural network. In *Proceedings of the 2014 International Joint Conference on Neural Networks*. 800–807.

[52] Xiaofeng Zhu, Xuelong Li, and Shichao Zhang. 2016. Block-row sparse multiview multilabel learning for image classification. *IEEE Transactions on Cybernetics* 46, 2 (2016), 450–461.

[53] Fuzhen Zhuang, George Karypis, Xia Ning, Qing He, and Zhongzhi Shi. 2012. Multi-view learning via probabilistic latent semantic analysis. *Information Sciences* 199 (2012), 20–30.

[54] Wenzhang Zhuge, Feiping Nie, Chenping Hou, and Dongyun Yi. 2017. Unsupervised single and multiple views feature extraction with structured graph. *IEEE Transactions on Knowledge and Data Engineering* 29, 10 (2017), 2347–2359.

Received December 2018; revised March 2019; accepted June 2019