

Non-Linear Fusion for Self-Paced Multi-View Clustering

Zongmo Huang

zongmohuang@gmail.com

University of Electronic Science and Technology of China
Chengdu, China

Xiaorong Pu

puxiaor@uestc.edu.cn

University of Electronic Science and Technology of China
Chengdu, China

Yazhou Ren*

yazhou.ren@uestc.edu.cn

University of Electronic Science and Technology of China
Chengdu, China

Lifang He

lih319@lehigh.edu

Lehigh University
Bethlehem, PA, USA

ABSTRACT

With the advance of the multi-media and multi-modal data, multi-view clustering (MVC) has drawn increasing attentions recently. In this field, one of the most crucial challenges is that the characteristics and qualities of different views usually vary extensively. Therefore, it is essential for MVC methods to find an effective approach that handles the diversity of multiple views appropriately. To this end, a series of MVC methods focusing on how to integrate the loss from each view have been proposed in the past few years. Among these methods, the mainstream idea is assigning weights to each view and then combining them linearly. In this paper, inspired by the effectiveness of non-linear combination in instance learning and the auto-weighted approaches, we propose Non-Linear Fusion for Self-Paced Multi-View Clustering (NSMVC), which is totally different from the conventional linear-weighting algorithms. In NSMVC, we directly assign different exponents to different views according to their qualities. By this way, the negative impact from the corrupt views can be significantly reduced. Meanwhile, to address the non-convex issue of the MVC model, we further define a novel regularizer-free modality of Self-Paced Learning (SPL), which fits the proposed non-linear model perfectly. Experimental results on various real-world data sets demonstrate the effectiveness of the proposed method.

CCS CONCEPTS

• Computing methodologies → Cluster analysis.

KEYWORDS

multi-view clustering, non-linear fusion, self-paced learning

1 INTRODUCTION

As a fundamental field of machine learning, clustering [6] has been studied extensively and a great number of classical clustering algorithms have been developed in the past few decades, such as k -means [20], density-based clustering [5], distribution-based clustering [1], subspace-based clustering [12], matrix factorization based clustering [19], hierarchical clustering [10], mean shift clustering [4], and consensus clustering [26]. Unfortunately, while in most real-world clustering tasks, an object can be usually described by multiple aspects, these conventional clustering methods only work on the single-view data. To address this issue, a series of multi-view

clustering (MVC) methods [3, 13, 14, 16, 17, 27, 28, 30, 31, 34–37] have been proposed recently and achieve much better clustering results comparing with their single-view counterparts.

For MVC analysis, finding an appropriate approach to integrate different views is the foundation of making use of the complementary information within them. To tackle this problem, most existing MVC methods [29, 33] are based on the following simple and intuitive idea: finding some measurements to weight each view and then combining them linearly, while the idea of non-linear fusion has been always neglected. On the other hand, the non-linear combining idea such as using $\ell_{2,1}$ -norm has already been applied in instance learning [3, 15, 22] and has shown better robustness comparing with the ordinary Frobenius norm. As in most distance-based machine learning models, a few outliers with large losses always dominate the objective function and result in the poor performance of these algorithms. By applying $\ell_{2,1}$ -norm, the exponent of each sample's loss is decreased to 0.5 (i.e., rooted), thus the negative impact from the corrupt samples can be effectively alleviated and the robustness of the model can be enhanced.

Inspired by $\ell_{2,1}$ -norm, there have been a few recent attempts of the implied weighting algorithms capable of mimicking its effects, with an aim of decreasing the exponent of the original loss function and thus achieving remarkable clustering results. Enlighten by the idea of parameter-free learning, a series of MVC methods based on the auto-weighted (i.e. self-weighted) idea have been developed [7, 23–25]. In these algorithms, instead of using some criterion to measure the quality of each view and then assigning weights to them, the weights of different views are directly generated from their loss values. Through this approach, the objective functions of these methods no longer have the form like the linear-weighted combination of the losses from each separate view. However, in these models, the exponents of different views' losses are still the same, which means while smaller exponents alleviate the negative impact of the corrupt views, the influence of the reliable views are also weakened. Meanwhile, in the optimizing process, these methods still need to firstly transform their models into the traditional linear weighting forms.

Different from the existing linear-weighting MVC methods, we propose non-linear fusion for self-paced multi-view clustering (NSMVC) to directly grant different exponents to different views based on their qualities. By this way, our method can alleviate the negative influence from the corrupt views in a non-linear manner, which is similar with the way of dealing with corrupt instances by $\ell_{2,1}$ -norm.

*Corresponding author.

Except for the challenge of effective view integration, most conventional MVC methods also face the non-convex problem and thus usually stuck into suboptimal local solutions. To alleviate the non-convex issue, we further introduce the self-paced learning (SPL) mechanism [18] to our model. Imitating the learning process of human-beings, self-paced learning firstly trains the MVC model with the samples that have smaller losses, and as the iteration forwards, the samples with higher losses will gradually take part in the training process. In this way, the noisy samples and outliers will not join the training in the early iteration times, thus the robustness of MVC model can be significantly enhanced. Meanwhile, to fit the non-linear MVC model, a novel regularizer-free self-paced learning modality is also developed in this work. Through this SPL modality, the objective function of each view is completely constituted with the clustering losses of the selected instances and thus can avoid the influence from the value of the additional SPL regularizer.

Overall, we propose NSMVC to promote the clustering performance of MVC method in both view level and instance level. In the view level, a novel learning paradigm based on non-linear fusion is developed to plenary exploit the complementary information in different views. Moreover, to address the non-convex issue, we design a regularizer-free self-paced learning scenario to progressively train the MVC model from simplicity to complexity in the instance level. Through this approach, the robustness of the MVC model can be significantly enhanced.

The main contributions of the paper are summarized as follows:

- To the best of our knowledge, this is the first attempt to develop a view-level non-linear fusion method in the multi-view clustering task.
- A novel regularizer-free self-paced learning paradigm is designed to fit the non-linear model as well as alleviate the non-convex issue of conventional multi-view clustering method.
- An effective optimizing approach to solve the proposed NSMVC model is derived, and experimental results on multiple real-world data sets demonstrate the effectiveness of the proposed method.

The rest of this paper is organized as follows. We give a brief review of the literature of multi-view clustering and self-paced learning in Section 2. The details of the proposed DSMVC as well as its convergence and computational complexity analyses are presented in Section 3. The experimental results and conclusion are respectively described in Sections 4 and 5.

2 RELATED WORK

2.1 Multi-View Clustering

While in most real-world clustering tasks, objects can be described from multiple respects, the conventional clustering methods only can deal with single-view data. To make full use of the complementary information from different views and obtain better clustering result, a great number of multi-view clustering methods have been proposed in the past decade. In co-training approach for multi-view spectral clustering (co-train) [16] and co-regularized multi-view spectral clustering (co-reg) [17], Kumar et al. firstly put forward the fundamental assumption of MVC algorithms that among different views, the assignment of samples should be consistent. Due to the diversity of the inherent characteristics of different views,

it is important for MVC methods to find an appropriate integrating approach for each separate view. To this end, a great number of relative MVC methods have been proposed in past few years. Based on the kernel learning, Tzortzis and Likas [28] proposed multi-view kernel k-means clustering (MVKKM), in which each view is assigned with a weight according to its quality. Xu et al [35] proposed the weighted multi-view clustering with feature selection (WMCFS), which weights different views based on their clustering performance and utilizes feature selection to promote the efficiency and effectiveness of the MVC model. In multi-view clustering with multi-view capped-norm k-means (CAMVC) [8], Huang et al. assign weights for different views and implement the capped-norm loss in the objective function to achieve more stable clustering results with different initializations.

However, for most conventional linear-weighting methods, as they need some measurements to determine the weights of different views, it is inevitable to introduce additional hyper-parameters in their models. According to the parameter-free principle, a series of auto-weighted MVC models have been proposed recently and have shown superior performance comparing with the conventional linear ones.

To decrease the number of parameters in MVC model, Nie et al. [21, 23] proposed an auto-weighted approach to assign weights to different views according to their losses. Huang et al. [7] further applied this idea into the deep matrix decomposition based MVC model and achieve remarkable results on various real-world data sets. In [24], Ren et al. not only use the auto-weighted approach to address the view quality issue, but also apply $\ell_{2,1}$ -norm to tackle the noisy issue. Since the weight of each view is generated by their loss, these models are actually represented in the non-linear forms and the negative impact from less reliable views is significantly alleviated.

Different from the existing methods that use the auto-weighting strategy for non-linear multi-view clustering, we directly assign different exponents to each view according to their qualities.

2.2 Self-Paced Learning

For most machine learning tasks, the non-convex issue is one of the most crucial factors that makes the models stuck into suboptimal solutions easily. To tackle this issue, taking advantage of self-paced learning (SPL) [18] and curriculum learning [2] can be an ideal solution.

Imitating the mechanism of human-learning, SPL firstly trains the model with easy samples and gradually select more complex samples in the following training steps. In [11], Jiang et al. theoretically proved that applying SPL is beneficial in alleviating non-convex issue. The general form of self-paced learning can be written as follows:

$$\min_{v_i, w} \sum_{i=1}^n v_i f(x_i, y_i, w) - \lambda \sum_{i=1}^n v_i \quad (1)$$

$$s.t. \quad v_i \in \{0, 1\}.$$

From Eq. (1), only the samples with the loss smaller than λ will take part in the training process. As the iteration times increases, SPL gradually increases the value of λ to let more samples join

the training, thus the model will be trained from simplicity to complexity.

Due to the effectiveness of SPL, in these years, a series of MVC methods that utilizes SPL to promote the clustering performance have been proposed. Xu et al. [32] were the first to use SPL in the multi-view clustering and reveal the applicability of SPL in solving MVC problem. In [9], Huang et al. extended the idea of self-paced learning to feature learning and proposed a novel MVC method which alternatively performs sample learning and feature selection in a self-paced manner.

In this study, self-paced learning is not merely applied to tackle the non-convex issue, but also plays an important role in guiding the non-linear learning process.

3 NON-LINEAR FUSION FOR SELF-PACED MULTI-VIEW CLUSTERING

3.1 Problem Definition

Assuming that we are given a dataset with n instances in m views $\{X^v\}_{v=1}^m$, where $X^v = \{x_1^v, x_2^v, \dots, x_n^v\} \in \mathbb{R}^{d^v \times n}$, d^v is the dimension of the feature vector in the v -th view. Our target is to partition n instances into k clusters by making use of the complementary information from multiple views. Specifically, in this work, we aim to obtain better clustering results by utilizing the virtues of the non-linear fusion and self-paced learning.

3.2 Proposed Model

Inspired by the auto-weighted algorithms and non-linear learning's effectiveness in instance learning, in this paper, we develop a novel MVC model approach which is totally different from the conventional linear-combination form. The model of our method can be written as:

$$\sum_{v=1}^m \phi(v) \eta(v) \quad (2)$$

s.t. $\phi(v) \geq 0, \quad \eta(v) \in (0, 1]$,

where $\phi(v)$ and $\eta(v)$ respectively represent the loss of the v^{th} view and its exponent.

In Eq. (2), the corrupt views are expected to be assigned with the smaller $\eta(v)$ values. By this way, such views will be less influential during the optimizing process. Considering the extreme circumstance, when the value of $\eta(v)$ approaches 0, the contribution of the v^{th} view towards Eq. (2) will be close to the constant 1, thus the v^{th} view has no impact on the final clustering result. By contrast, by granting more reliable views with higher $\eta(v)$ values, the MVC model will be more sensitive to the variations of the corresponded $\phi(v)$ values and these views will play more important roles during the training process. Therefore, our proposed non-linear model in Eq. (2) could significantly alleviate the negative influence from the corrupt views while maintain the availability of more reliable views.

To further address non-convex issue as well as fit the non-linear model shown in Eq. (2), we design a novel regularizer-free SPL

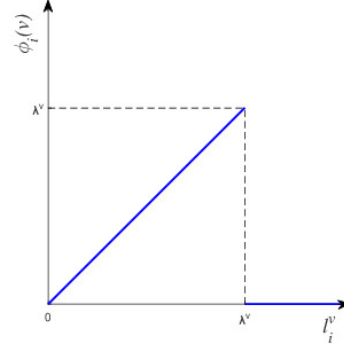


Figure 1: Illustration of the relation between $\phi_i(v)$ and l_i^v .

paradigm. Concretely, we define $\phi(v)$ as:

$$\phi(v) = \sum_{i=1}^n \phi_i(v) = \sum_{i=1}^n \lceil \max(1 - l_i^v / \lambda^v, 0) \rceil \times l_i^v. \quad (3)$$

Here, $\phi_i(v) \geq 0$ denotes the i^{th} sample's contribution to $\phi(v)$. $\lambda^v \geq 0$ represents the control parameter of the self-paced learning in the v^{th} view and $\lceil \cdot \rceil$ means the rounding up operation, while $l_i^v \geq 0$ and n denote the loss of the i^{th} instance in the v^{th} view and the total number of samples respectively. The relationship of $\phi_i(v)$ and l_i^v can be described by Figure. 1. Specifically, the value of l_i^v is computed by:

$$l_i^v = \|x_i^v - C^v b_i\|_2^2. \quad (4)$$

$C^v = \{c_1^v, c_2^v, \dots, c_k^v\} \in \mathbb{R}^{d^v \times k}$ represents the centers of clusters in the v^{th} view, and k denotes the predefined number of clusters. $B = \{b_1, b_2, \dots, b_n\} \in \mathbb{R}^{k \times n}$ reflects the clustering assignment of each data point and is shared by all the views. Concretely, if the j^{th} sample is assigned to the i^{th} cluster then $b_{ij} = 1$, otherwise $b_{ij} = 0$. Therefore, the value of l_i^v physically means the squared Euclidean distance between the instance and the center of its belonging cluster.

With Figure. 1, Eq. (3) can be written in another form:

$$\phi(v) = \|(X^v - C^v B) \text{diag}(\mathbf{w}^v)\|_F^2, \quad (5)$$

where $\mathbf{w}^v = [w_1^v, w_2^v, \dots, w_n^v]$, $w_i^v \in \{0, 1\}$, and $w_i^v = 1$ only when $l_i^v \leq \lambda^v$.

From Eq. (5), we can find out that Eq. (3) actually has the same selection result as the following formula in the conventional self-paced learning manner [18] like Eq. (1):

$$\min_{\mathbf{w}^v} \sum_{i=1}^n w_i^v l_i^v - w_i^v \lambda^v \quad (6)$$

s.t. $w_i^v \in \{0, 1\}$.

When all the $\eta(v)$ values are equal to 1, solving Eq. (6) is equivalent to minimizing Eq. (5). However, with the existence of an additional regularizer, the value of Eq. (6) is absolutely less than or equal to 0. Due to this characteristic, such conventional SPL paradigm is not suitable for the non-linear model in Eq. (2) that constrains $\eta(v) \in (0, 1]$. For instance, if $\eta(v)$ is 0.5, then the non-linear model works only when $\phi(v) \geq 0$, which is impracticable to the SPL paradigm in Eq. (6). Instead, in Eq. (3), the novel SPL paradigm

abandons the regularizer term. By this way, the objective function is totally constituted with the clustering loss of the selected samples and thus ensures the non-negativity of $\phi(v)$.

During the iterative optimizing process, our method gradually increases the value of λ^v until all the samples have joined the training, thus the MVC model will be trained from simplicity to complexity.

Another point that deserves our attention is that since λ^v regulates the participation of samples in the v^{th} view, itself can be an evidence of the quality of the corresponded view. Generally speaking, a more reliable view has a smaller $\lambda(v)$ and vice versa. With the constraint that $\eta(v) \in (0, 1]$, the value of $\eta(v)$ is:

$$\eta(v) = \frac{\min_u \lambda^u}{\lambda^v}. \quad (7)$$

Integrating Eq. (2) to (7), the objective of NSMVC becomes:

$$\min_{C^v, B, w^v} \sum_{v=1}^m \|(X^v - C^v B) \text{diag}(w^v)\|_F^{2 \min \lambda^u / \lambda^v}. \quad (8)$$

3.3 Optimization

To start the optimizing process, our method firstly initializes the cluster centers C^v and assignment matrix B randomly. After that, the objective function Eq. (8) is optimized by iteratively updating each variable while others are fixed.

3.3.1 Step1: Fix C^v , B , update λ^v , w^v , and $\eta(v)$.

To take the advantage of self-paced learning and enhance the robustness of the MVC model, our method should firstly determine the participation of samples in the beginning of each iteration. Assuming that the whole SPL process needs T iterations, the updating rule of λ^v in the t^{th} iteration is:

$$\lambda^v = \min(l_i^v) + (\alpha + (t - 1) \times \beta) \times (\max(l_i^v) - \min(l_i^v)) \quad (9)$$

In Eq. (9), the value of β is computed by:

$$\beta = \frac{1 - \alpha}{T - 1} \quad (10)$$

s.t. $\alpha \in [0, 1]$.

By this way, we could control the starting point of the self-paced learning process as well as finish it in the promised iteration times T . For instance, when we set the start point α as 0.5 and T as 6, from Eq. (9), the value of β will become 0.1. Then, in the first iteration, only the samples with losses smaller than the mean value of the $\max(l_i^v)$ and the $\min(l_i^v)$ will be selected. In the final (6^{th}) iteration, the value of λ^v will be just large enough to let all the samples join the training.

After deciding the value of λ^v for v^{th} view, the values in w^v can be obtained naturally. When all λ^v are acquired, the value of each $\eta(v)$ will be calculated by Eq. (7).

3.3.2 Step2: Fix λ^v , $\eta(v)$, and w^v , update C^v and B alternately.

(a) Fix B , update C^v :

When $\eta(v)$, w^v , and B are fixed, optimizing Eq. (8) is equivalent to solving the following problem for each view:

$$\min_{C^v} \sum_{v=1}^m \|(X^v - C^v B) \text{diag}(w^v)\|_F^{2 \eta(v)}. \quad (11)$$

Algorithm 1 The NSMVC Algorithm.

Input: Data set X^v , $v = 1, 2, \dots, m$; Cluster number k ; SPL start point α and iteration times T .

Output: The final cluster center matrix C^v , assignment matrix B , $v = 1, 2, \dots, m$.

```

1: Initialize  $C^v$  and  $B$  randomly.
2: repeat
3:   for each view  $v$  do
4:     According to Eq. (9) update  $\lambda^v$  to let more samples join the training.

5:   for each sample  $i$  do
6:     Update  $w_i^v = 1$  if  $l_i^v \leq \lambda^v$ , otherwise  $w_i^v = 0$ .
7:   end for
8: end for
9: Update  $\eta(v)$  for each view according to Eq. (7).
10: repeat
11:   for each view  $v$  do
12:     Fix  $\eta(v)$ ,  $w^v$  and  $B$ , update  $C^v$  according to Eq. (15).
13:   end for
14:   Fix  $\eta(v)$ ,  $C^v$  and  $w^v$ , update  $B$  according to Eqs. (16) and (17).
15: until convergence or exceed the maximal number of iterations
16: until all data points are selected
17: return  $C^v$  and  $B$ ,  $v = 1, 2, \dots, m$ .
```

From Eq. (11), the optimal C^v of each view can be separately obtained by finding the solution of:

$$\min_{C^v} \|(X^v - C^v B) \text{diag}(w^v)\|_F^2. \quad (12)$$

Then, Eq. (12) can be transformed into the form of matrix's trace:

$$\min_{C^v} \text{Tr}((X^v - C^v B) \text{diag}^2(w^v) (X^v - C^v B)^T). \quad (13)$$

Regarding Eq. (13) as a function $J(C^v)$ and its gradient is:

$$\frac{\partial J}{\partial C^v} = 2X^v \text{diag}^2(w^v) B^T - 2C^v B \text{diag}^2(w^v) B^T. \quad (14)$$

Setting this gradient to $\mathbf{0}$, then the updating rule of C^v is:

$$C^v = X^v \text{diag}^2(w^v) B^T (B \text{diag}^2(w^v) B^T)^{-1}. \quad (15)$$

(b) Fix C^v , update B :

As for assignment matrix B , due to the non-linearity of our model, it cannot be solved by the conventional route that divides the loss function in the instance level and finds the optimal b_i for each sample separately. To address this issue, we design a simple and effective solving approach, which guarantees to decrease the value of objective function with the time complexity $O(n)$.

Concretely, we sequentially update b_i for each sample one by one. When updating b_i , we define $\theta_i(v)$ for the i^{th} sample in the v^{th} view as:

$$\theta_i(v) = \phi(v) - l_i^v. \quad (16)$$

Substituting Eq. (16) in Eq. (11), b_i can be obtained by solving:

$$\arg \min_{b_i} \sum_{v=1}^m (\theta_i(v) + \|(x_i^v - C^v b_i) w_i^v\|_2^2)^{\eta(v)}. \quad (17)$$

Since the possible alternatives' number of b_i is the predefined cluster number k , Eq. (17) can be easily addressed by exhaustive search. When the optimal b_i is obtained according to Eq. (17), the $\phi(v)$ value in Eq. (16) for each view can be updated and will be used to

find the optimal $b_{(i+1)}$ for the $(i+1)^{th}$ sample. As we usually have the fact that $k \ll n$, this step only needs $O(n)$ operations.

In *Step 2*, C^v and B are alternatively updated until convergence or exceed the maximal iteration times.

The above three steps correspond to an entire iteration of the optimization. Our method keeps the optimization process running until all the samples have taken part in the training process. When the whole training process is finish, the final cluster center matrices C^1, C^2, \dots, C^m and assignment matrix B reflect the clustering result.

The entire procedure of NSMVC is summarized in Algorithm 1.

3.4 Convergence Analysis

As the NSMVC finishes in a fixed number of iterations and *Step 1* only plays an instance selection role for the next step, thus we only need to focus on the convergence trend in the alternatively updating process of *Step 2*. In *Part (a)* of *Step 2*, C^v is updated by finding the optimal solution of Eq. (11), so that the value of Eq. (8) is guaranteed to decrease. In *Part (b)* of *Step 2*, as we find the optimal b_i for each instance sequentially, the value of Eq. (8) decreases monotonously. With the non-negativity of $\phi(v)$ and the monotone bounded convergence theorem, NSMVC is guaranteed to converge to a local minimum.

3.5 Computational Complexity Analysis

Let D and P denote the maximal feature dimensionality and the maximal number of iterations for alternatively updating C^v and B . For *Step 1*, as the λ^v of each view is generated from the minimal and maximal clustering loss of all the samples, obtaining λ^v needs $O(nD)$ operations. Thus, updating all the λ^v ($v = 1, 2, \dots, m$) needs $O(nmD)$ operations. After that, the values of w^v and $\eta(v)$ can be naturally computed with $O(n)$ and m operations respectively. Therefore, the time complexity of *Step 1* is $O(nD)$. For *Step 2*, since $diag(w^v)$ is diagonal, the time complexity of updating C^v by Eq. (15) is $O(nDk)$. Therefore, updating all the C^v ($v = 1, 2, \dots, m$) needs $O(nmDk)$ operations. Then, as discussed in the *Part (b)* of 3.3.1, updating B needs $O(n)$ operations, thus the total time complexity of *Step 2* is $O(PnmDk)$. Since the entire self-paced learning process needs T iteration times, the overall computational complexity of NSMVC is $O(PTnmDk)$. With the fact that $T \ll n$ usually holds and applying k -means individually on m views needs $O(PmnDk)$ operations. In summary, the proposed NSMVC shares similar computational complexity with the traditional k -means method, which is linear to the data size n .

4 EXPERIMENTAL RESULTS

4.1 Experimental Setup

Data sets: Handwritten Numerals¹ data set is chosen from UCI machine learning repository, which contains 2000 points in 10 classes corresponded to numerals (0-9). Each instance is described by the following six views: 216 profile correlations, 76 Fourier coefficients of the character shapes, 64 Karhunen-Love coefficients, 6 morphological features, 240 pixel averages in 2×3 windows, and 47 Zernike moments.

MSRCv1² is an image data set that constituted with 210 images over seven classes including tree, building, airplane, cow, face, car, and bicycle, each providing 30 images. For each image, it is described from 5 aspects: 24 Color Moments, 576 HOG features, 512 features, 256 LBP features, and 254 Centrist features.

The rest four data sets originate from the universities in Cornell, Texas, Washington and Wisconsin³. For each data set, there are two views, i.e., the content view and the citation view. According to the ground truth, these samples can be grouped into five classes: student, project, course, staff, and faculty.

The detailed characteristics of data sets is shown in Table 1. The image samples of Handwritten Numerals and MSRCv1 data sets are presented in Figure 2.

Comparing Methods: To demonstrate the effectiveness of the proposed NSMVC model, we compare it with eight existing state-of-the-art multi-view clustering methods:

- Co-train: A co-training based approach for multi-view spectral clustering [16].
- Co-reg: A centroid based co-regularized multi-view spectral clustering method [17].
- MVKMM: The multi-view kernel k -means clustering method proposed by [28].
- AMGL: An auto-weighted multiple graph learning method for multi-view clustering [23].
- CAMVC: The robust capped-norm multi-view clustering method proposed by [8].
- MSPL: A multi-view self-paced learning method for multi-view clustering [32].
- SAMVC: A self-paced and auto-weighted multi-view clustering method [24].
- DMVC: An auto-weighted multi-view clustering method based on deep matrix decomposition [7].

In the above algorithms, MVKMM and CAMVC adopt the conventional linear combination strategy, AMGL, SAMVC, and DMVC apply the auto-weighting method for each view, and MSPL and SAMVC incorporate the self-paced learning strategy into multi-view clustering regimes to improve clustering results.

To make a comprehensive comparison, we also employ k -means clustering on each single view (e.g., KM(1) means applying KM on the first view). Moreover, we concatenate the features of each view, and use k -means clustering on the joint view representation of the data, denoted as KM(All).

Implementation and Evaluation Metrics: For the KM method, we use the `kmeans` function in Matlab to form the clusters. For the MSPL, we follow the original paper to reproduce it. For the rest, we directly use the source codes from the authors and follow the suggesting parameter settings by corresponding publications. For all the methods, the number of clusters is always set to the number of ground truth classes. For the proposed method, the hyperparameters α and T are selected in the ranges of $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ and $\{3, 4, 5, 6, 7, 8\}$, respectively.

In order to measure the quality of clustering results, we adopt three widely used evaluation metrics: accuracy (ACC), Purity, and normalized mutual information (NMI). The higher value of each

¹<https://archive.ics.uci.edu/ml/datasets.php>

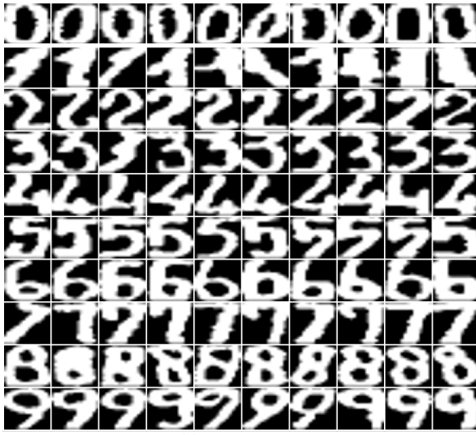
²<https://www.microsoft.com/en-us/research/project/image-understanding>

³<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/>

Table 1: Summary of the data sets used in the experiments.

View	Handwritten Numerals	MSRCv1	Cornell	Texas	Washington	Wisconsin
1	Profile correlations (216)	Color Moments (24)	Citation (195)	Citation (187)	Citation (230)	Citation (265)
2	Fourier coefficients (76)	HOG (576)	Content (1703)	Content (1398)	Content (2000)	Content (1703)
3	Karhunen coefficients (64)	GIST (512)	-	-	-	-
4	Morphological (6)	LBP (256)	-	-	-	-
5	Pixel averages (240)	Centrist (254)	-	-	-	-
6	Zernike moments (47)	-	-	-	-	-
# Samples	2000	210	195	187	230	265
# Classes	10	7	5	5	5	5

* Numbers in parentheses are the number of features in each view.



(a) Handwritten



(b) MSRCv1

Figure 2: An example of two image data sets. Each row represents ten image samples from a single class.

Table 2: Results on Handwritten Numerals.

Methods	ACC(%)	Purity(%)	NMI(%)
KM(1)	57.71±4.98	63.71±3.69	60.27±2.41
KM(2)	62.99±6.72	65.38±5.03	64.38±2.76
KM(3)	70.53±7.27	73.38±5.84	70.93±3.59
KM(4)	38.09±1.55	43.80±0.97	47.76±0.24
KM(5)	70.05±6.89	72.56±6.36	70.38±3.99
KM(6)	52.10±2.93	55.95±2.33	50.01±1.82
KM(All)	50.72±4.17	56.01±2.44	57.37±1.64
Co-train	73.28±5.87	74.92±3.89	71.04±2.15
Co-reg	78.09±6.89	80.63±5.36	75.50±2.91
MVKKM	62.18±3.34	65.56±2.40	65.80±1.19
AMGL	81.22±6.53	84.24±4.99	86.89±2.66
CAMVC	74.98±7.96	78.84±6.90	78.07±4.25
MSPL	80.26±3.93	83.60±3.22	82.80±2.25
SAMVC	75.37±12.71	79.74±11.91	82.62±12.49
DMVC	79.91±8.56	83.77±6.72	85.18±4.01
NSMVC	88.52±6.40	90.53±4.63	89.10±2.25

Table 3: Results on MSRCv1.

Methods	ACC(%)	Purity(%)	NMI(%)
KM(1)	35.76±2.38	37.88±2.45	24.25±2.50
KM(2)	62.69±6.60	64.60±5.59	54.16±4.45
KM(3)	62.00±5.52	64.90±4.17	57.03±3.74
KM(4)	47.29±1.55	49.55±0.97	41.38±0.24
KM(5)	54.64±6.79	55.55±5.25	45.18±3.19
KM(All)	46.29±3.10	46.29±3.03	42.07±2.18
Co-train	66.55±5.77	69.33±4.46	58.18±3.46
Co-reg	41.52±4.31	44.21±3.89	35.36±3.61
MVKKM	70.19±3.73	70.95±3.31	61.61±3.07
AMGL	69.74±7.04	71.81±5.01	68.35±3.24
CAMVC	67.88±5.18	71.14±3.49	62.86±2.73
MSPL	50.19±6.01	52.29±4.94	43.30±2.66
SAMVC	65.31±8.82	68.19±8.01	61.57±6.28
DMVC	62.57±10.49	63.67±10.48	58.75±8.67
NSMVC	74.65±5.62	77.14±4.44	66.65±5.00

metric indicates the better performance. Each experiment was repeated for 30 times, and the mean and standard deviation of each metric in each data set were reported.

4.2 Clustering Results

Tables 2-7 show the clustering performance of all the comparison methods on each data set in terms of ACC, Purity, and NMI. In each

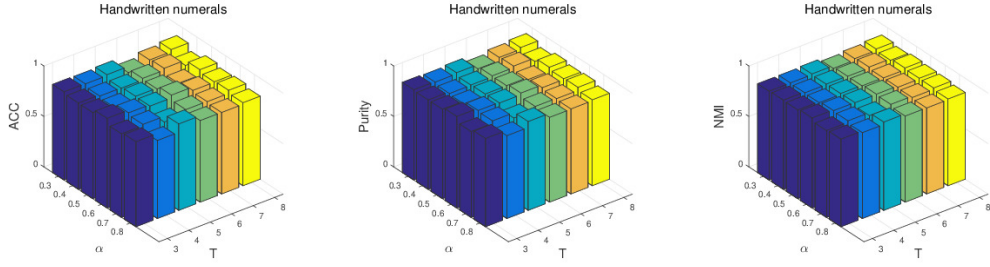


Figure 3: Clustering performance w.r.t. different parameter settings on Handwritten Numerals.

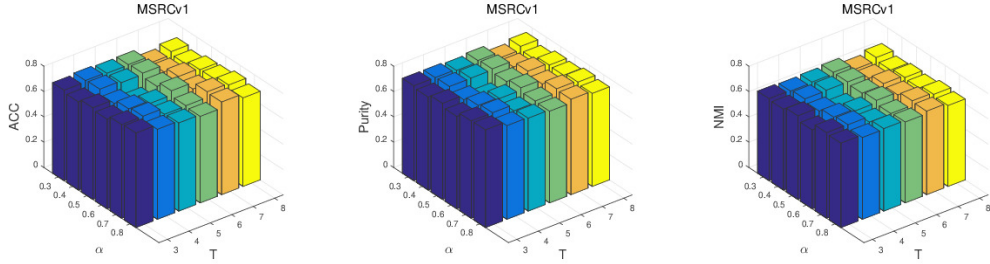


Figure 4: Clustering performance w.r.t. different parameter settings on on MSRCv1.

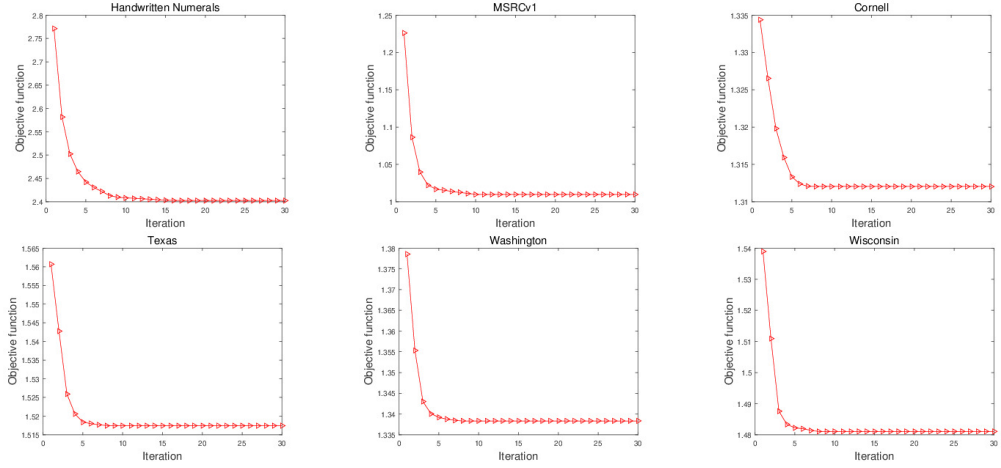


Figure 5: Convergence curves of NSMVC on all data sets.

column, the best results and the comparable results under the t -test with 5% significance level are highlighted in boldface. From these results, we have the following observations. First, the proposed NSMVC method almost always outperform the baseline methods on all data sets. This is mainly because our method adaptively assigns different exponents to different views according to their qualities. Therefore, the proposed NSMVC reduces the negative impact of the corrupt views as well as maintains the influences of the more reliable views. Moreover, it can be found that NSMVC always perform better than the conventional auto-weighted MVC methods like AMGL, DMVC, and SAMVC in which the exponents are consistent among all the views, which empirically confirm the

effectiveness of the novel non-linear fusion technique. Further, by taking advantage of the self-paced learning to alleviate the non-convex issue, the proposed NSMVC also performs smaller standard deviations comparing with other methods, which indicates the better robustness of our method.

4.3 Convergence Study

This section analyses the convergence of our method. Figure 5 shows the convergence curves on six data sets in the first SPL process. In these curves, the abscissa means the iteration number while the ordinate denotes the objective value of Eq. (8). From this figure, we can see that the proposed optimization algorithm

Table 4: Results on Cornell.

Methods	ACC(%)	Purity(%)	NMI(%)
KM(1)	42.70±2.14	44.96±1.00	8.60±3.36
KM(2)	45.56±5.87	48.56±3.58	12.34±5.42
KM(All)	47.47±6.42	49.69±5.01	13.54±6.96
Co-train	40.62±1.27	46.41±0.88	14.48±1.40
Co-reg	42.39±1.09	44.10±0.36	5.65±2.45
MVKKM	41.64±3.72	44.72±1.03	7.27±1.83
AMGL	42.68±0.40	43.81±0.26	3.74±0.37
CAMVC	44.10±2.74	49.16±2.12	9.81±4.95
MSPL	44.09±3.18	46.19±2.65	8.78±4.62
SAMVC	43.43±0.82	44.69±0.50	6.82±3.11
DMVC	44.30±2.74	46.72±2.82	12.42±3.99
NSMVC	50.62±6.45	60.21±2.76	26.56±3.94

Table 5: Results on Texas.

Methods	ACC(%)	Purity(%)	NMI(%)
KM(1)	55.51±1.63	57.18±1.25	7.90±4.50
KM(2)	55.56±6.07	60.04±4.61	16.18±10.90
KM(All)	56.63±5.84	60.41±5.04	14.63±10.38
Co-train	48.13±2.75	58.00±0.89	14.22±1.78
Co-reg	53.37±2.67	56.04±0.22	4.57±1.89
MVKKM	52.84±5.63	56.84±0.89	7.70±3.71
AMGL	56.13±0.43	56.84±0.25	5.43±0.51
CAMVC	59.23±4.49	60.71±4.42	14.92±9.54
MSPL	56.68±3.85	58.97±2.94	11.30±6.30
SAMVC	56.81±1.38	57.68±1.35	8.54±4.54
DMVC	56.90±4.29	59.84±3.24	16.32±7.15
NSMVC	57.81±4.93	67.17±1.45	25.23±2.60

Table 6: Results on Washington.

Methods	ACC(%)	Purity(%)	NMI(%)
KM(1)	49.80±6.84	51.46±6.86	8.44±7.84
KM(2)	57.54±9.77	61.39±9.64	25.36±14.50
KM(All)	58.75±9.40	66.29±8.68	26.23±12.89
Co-train	53.99±2.25	62.93±1.22	19.30±1.78
Co-reg	55.97±2.95	58.64±4.19	16.68±3.63
MVKKM	48.39±1.85	49.49±1.83	8.65±4.53
AMGL	47.26±0.20	48.26±0.00	3.58±0.32
CAMVC	58.97±10.57	60.81±10.62	22.53±14.99
MSPL	52.67±7.99	54.16±7.95	13.75±11.15
SAMVC	52.93±7.95	53.70±8.12	11.39±9.43
DMVC	58.45±6.96	62.44±7.54	22.09±9.38
NSMVC	57.96±5.82	71.13±3.20	36.20±3.79

converges quickly in the vicinity of the minimum, i.e., only around 15 iterations. At the same time, it is clear that even in the beginning of the training process, NSMVC converges quickly. As the training forwards, NSMVC learns more available knowledge for clustering and converges faster.

Table 7: Results on Wisconsin.

Methods	ACC(%)	Purity(%)	NMI(%)
KM(1)	46.44±2.18	48.67±1.82	5.69±2.30
KM(2)	59.81±7.92	62.60±8.70	28.97±12.41
KM(All)	58.57±6.93	60.33±7.68	25.95±11.35
Co-train	42.58±1.89	52.57±1.10	8.28±0.83
Co-reg	47.35±0.24	47.76±0.21	4.06±0.37
MVKKM	45.62±2.88	48.03±1.34	6.29±2.22
AMGL	47.09±0.16	47.55±0.00	4.03±0.31
CAMVC	56.49±7.30	59.58±7.88	21.57±9.36
MSPL	55.50±6.85	57.66±6.79	21.87±9.27
SAMVC	49.93±3.13	48.93±3.15	7.04±4.69
DMVC	53.01±6.93	58.73±7.24	19.68±10.17
NSMVC	60.30±4.89	73.70±1.73	40.48±1.85

4.4 Parameter Sensitivity

In our NSMVC, there are two parameters for self-paced learning in Eq. (10), i.e., the starting point α and the total iteration number T . Taking Handwritten Numerals and MSRCv1 data sets as examples, we examine the influence of these parameters to the clustering performance. Figures 3 and 4 show the variation of ACC, Purity and NMI over different α and T on two data sets. We can observe that the clustering performance of the proposed NSMVC is relatively stable in a wide range of α and T values, which may provide a good guidance for the parameter setting.

5 CONCLUSION

In this paper, a self-paced learning-based non-linear fusion method (NSMVC) is proposed to enhance the clustering performance of conventional multi-view clustering methods in both view level and instance level. In the view level, a novel and effective non-linear fusion paradigm for multi-view clustering is proposed to exploit the complementary information from different views. Meanwhile, to fit the non-linear model, we further design a novel modality of self-paced learning without the regularizer term. By training the MVC model from simplicity to complexity progressively in the instance level, the non-convex issue is significantly alleviated and the robustness of the MVC model is further enhanced. Extensive experiments on various multi-view data sets demonstrate the effectiveness of the proposed NSMVC.

REFERENCES

- [1] Banfield, Adrián, and Adrian E. Raftery. 1993. Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics* 49 (1993), 803–821.
- [2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*. 41–48.
- [3] Xiao Cai, Feiping Nie, and Heng Huang. 2013. Multi-View K-Means Clustering on Big Data. In *Proceedings of International Joint Conference on Artificial Intelligence*. 2598–2604.
- [4] Dorin Comaniciu and Peter Meer. 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 5 (2002), 603–619.
- [5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*. 226–231.
- [6] John A. Hartigan. 1975. *Clustering algorithms*. John Wiley & Sons, Inc.

- [7] Shudong Huang, Zhao Kang, and Zenglin Xu. 2020. Auto-weighted multi-view clustering via deep matrix decomposition. *Pattern Recognition* 97 (2020), 107015.
- [8] Shudong Huang, Yazhou Ren, and Zenglin Xu. 2018. Robust multi-view data clustering with multi-view capped-norm K-means. *Neurocomputing* 311 (2018), 197–208.
- [9] Zongmo Huang, Yazhou Ren, Xiaorong Pu, Lili Pan, Dezhong Yao, and Guoxian Yu. 2021. Dual self-paced multi-view clustering. *Neural Networks* 140 (2021), 184–192.
- [10] A. K. Jain, M. N. Murty, and P. J. Flynn. 1999. Data Clustering: A Review. *Comput. Surveys* 31, 3 (September 1999), 264–323.
- [11] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. 2015. Self-Paced Curriculum Learning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 2694–2900.
- [12] Yangbangyan Jiang, Qianqian Xu, Zhiyong Yang, and Xiaochun Cao. 2019. Duet Robust Deep Subspace Clustering. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1596–1604.
- [13] Wen Jie, Zheng Zhang, Zhao Zhang, Zhihao Wu, Lunke Fei, Yong xu, and Bob Zhang. 2020. DIMC-net: Deep Incomplete Multi-view Clustering Network. In *Proceedings of the 28th ACM International Conference on Multimedia*. 3753–3761.
- [14] Zhao Kang, Xinjia Zhao, Chong Peng, Hongyuan Zhu, Joey Tianyi Zhou, Xi Peng, Wenyu Chen, and Zenglin Xu. 2020. Partition level multiview subspace clustering. *Neural Networks* 122 (2020), 279 – 288.
- [15] Deguang Kong, Chris Ding, and Heng Huang. 2011. Robust Nonnegative Matrix Factorization using L21-norm. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. 673–682.
- [16] Abhishek Kumar and Hal Daumé. 2011. A Co-training Approach for Multi-view Spectral Clustering. In *Proceedings of International Conference on Machine Learning*. 393–400.
- [17] Abhishek Kumar, Piyush Rai, and Hal Daumé, III. 2011. Co-regularized Multi-view Spectral Clustering. In *Advances in Neural Information Processing Systems* (Granada, Spain). 1413–1421.
- [18] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*. 1189–1197.
- [19] Daniel D. Lee and Hyunjune Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *14th Annual Neural Information Processing Systems Conference*. 556–562.
- [20] J. MacQueen. 1967. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. 281–297.
- [21] F. Nie, G. Cai, J. Li, and X. Li. 2018. Auto-Weighted Multi-View Learning for Image Clustering and Semi-Supervised Classification. *IEEE Transactions on Image Processing* 27, 3 (2018), 1501–1511.
- [22] Feiping Nie, Heng Huang, Xiao Cai, and Chris Ding. 2010. Efficient and Robust Feature Selection via Joint l2,1-Norms Minimization. In *Advances in Neural Information Processing Systems*. 1813–1821.
- [23] 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 Twenty-Fifth International Joint Conference on Artificial Intelligence* (New York, New York, USA). 1881–1887.
- [24] Yazhou Ren, Shudong Huang, Peng Zhao, Minghao Han, and Zenglin Xu. 2020. Self-paced and auto-weighted multi-view clustering. *Neurocomputing* 383 (2020), 248 – 256.
- [25] Shaojun Shi, Feiping Nie, Rong Wang, and Xuelong Li. 2020. Auto-weighted multi-view clustering via spectral embedding. *Neurocomputing* 399 (2020), 369–379.
- [26] Alexander Strehl and Joydeep Ghosh. 2002. Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research* 3 (2002), 583–617.
- [27] Chang Tang, Xinwang Liu, Xinzhou Zhu, En Zhu, Zhigang Luo, Lizhe Wang, and Wen Gao. 2020. CGD: Multi-View Clustering via Cross-View Graph Diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 5924–5931.
- [28] Grigoris Tzortzis and Aristidis Likas. 2012. Kernel-Based Weighted Multi-view Clustering. *2012 IEEE 12th International Conference on Data Mining* (2012), 675–684.
- [29] Hongxing Wang, Chaoqun Weng, and Junsong Yuan. 2014. Multi-feature spectral clustering with minimax optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4106–4113.
- [30] 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.
- [31] Yang Wang and Lin Wu. 2018. Beyond Low-Rank Representations: Orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering. *Neural Networks* 103 (2018), 1 – 8.
- [32] Chang Xu, Dacheng Tao, and Chao Xu. 2015. Multi-view self-paced learning for clustering. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. 3974–3980.
- [33] 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.
- [34] Nan Xu, Yanqing Guo, Xin Zheng, Qianyu Wang, and Xiangyang Luo. 2018. Partial Multi-view Subspace Clustering. In *Proceedings of the 26th ACM international conference on Multimedia*. 1794–1801.
- [35] Yu-Meng Xu, Chang-Dong Wang, and Jian-Huang Lai. 2016. Weighted Multi-view Clustering with Feature Selection. *Pattern Recognition* 53 (2016), 25–35.
- [36] Guang-Yu Zhang, Chang-Dong Wang, Dong Huang, and Wei-Shi Zheng. 2017. Multi-view collaborative locally adaptive clustering with Minkowski metric. *Expert Systems with Applications* 86 (2017), 307–320.
- [37] Linlin Zong, Xianchao Zhang, Long Zhao, Hong Yu, and Qianli Zhao. 2017. Multi-view clustering via multi-manifold regularized non-negative matrix factorization. *Neural Networks* 88 (2017), 74 – 89.