33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

54

Multi-View Clustering Based on Deep Non-Negative Tensor Factorization

Paper ID: 3857

ABSTRACT

Multi-view clustering (MVC) methods based on non-negative matrix factorization (NMF) have gained popularity owing to their ability to provide interpretable clustering results. However, these NMF-based MVC methods generally process each view independently and thus ignore the potential relationship between views. Besides, they are limited in the ability to capture the nonlinear data structures. To overcome these weaknesses and inspired by deep learning, we propose a multi-view clustering method based on deep non-negative tensor factorization (MVC-DNTF). With deep tesnor factorization, our method can well exploit the spatial structure of the original data and is capable of extracting more deep and nonlinear features embedded in different views. To further extract the complementary information of different views, we adopt the weighted tensor Schatten p-norm regularization term. An optimization algorithm is developed to effectively solves the MVC-DNTF objective. Extensive experiments are performed to demonstrate the effectiveness and superiority of our method.

CCS CONCEPTS

• Computing methodologies → Machine learning; Artificial intelligence.

KEYWORDS

Multi-view clustering, Multi-view learning, Non-negative tensor factorization, Schatten *p*-norm

1 INTRODUCTION

In recent years, the advancement of information technology has led to an exponential increase in data generated across various domains. This data usually comes from different sources [1]. Multiview data processing have demonstrated significant applications in scenarios such as image processing, social network analysis, and bioinformatics. As an unsupervised learning method, multi-view clustering gains increasing attention due to its ability to utilize the multiple data views to improve the clustering performance by exploiting the consistent and complementary information among them [15]. So far, numerous multi-view clustering methods are proposed, which can be roughly divided in four categories [41]: Cotraining based methods [20], multi-kernel learning [34], multi-view graph clustering [39], and multi-view subspace clustering [45].

51 for profit or commercial advantage and that copies bear this notice and the full citation 52 on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or 53 republic to note on servers not reclistificate to lists requires prior precific permission.

58

Among these methods, the co-training approach [22] leverages the co-training framework to ensure consistency in either prior knowledge or clustering outcomes. Furthermore, Tzortzis et al [34] proposed a multi-view clustering technique based on kernel functions, which weights all views using kernel functions. This approach assigns different weights to various views, achieving improved clustering results. Pan et al [30] proposed a generalized framework for clustering multi-view attribute graph data and introduced the Multi-view Comparison Graph Clustering (MCGC) method, which enhances graph clustering by filtering noise and learning consensus graphs with contrast loss.Additionally, some approaches focus on individually optimizing each view to identify the most informative subspaces, enabling clustering within these subspaces. This strategy enhances the handling of high-dimensional data and noise, thereby improving the robustness and effectiveness of clustering algorithms [6, 35].

Notably, non-negative matrix factorization (NMF) has received widespread attention owing to its outstanding clustering performance and interpretability [8, 23], and many improved methods were developed based on NMF [3, 7, 9, 10, 37, 38], *etc.* For example, Cai et al [4] proposed a graph-regularized NMF (GNMF) method, which mines the flow structure of NMF data for effective clustering. Its effectiveness motivates researchers to extend NMF to multi-view NMF [46]. For example, Ou et al [29] introduced NMF into multiview clustering by considering the geometric structure of each view to mine the potential information of each view; Semi-supervised multi-view clustering method based on NMF was proposed in [28], which adopts semi-supervised approach for clustering, and the effectiveness of the method is verified by several sets of experiments. Effective as they, these methods are incapable of extracting nonlinear features.

To overcome this weakness, Zhao et al. proposed a multi-view clustering via deep matrix factorization, which fully explores the potential attributes of the data in a layer-wise fashion and explore the geometrical structure with a graph-based regularizer [43]. Similarly, Zhang et al. developed a deep NMF based multi-view clustering method with partition alignment (MVC-DMF-PA), which employs deep NMF to learn the clustering partition matrix for each view and then fuse them into the final partition result with a partition algiment [42]. These deep NMF-based MVC methods exhibit impressive feature extraction ability. Besides, they realize better efficiency when compared with other MVC methods (e.g., graph-based methods or deep methods). Nonetheless, they essentially apply NMF on each single view and then integrate them into a consistent clustering assignment matrix, resulting in the the loss of the original inter-view spatial information [24]. For another, since they directly decompose the original data, their efficiency decreases dramatically when the original data dimensions are large.

Kilmer *et al.* introduced the factorization of nonnegative tensors [21], which has been proven to be an effective to learn the

115

116

59

60

61 62

63 64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

Unpublished working draft. Not for distribution.

and/or a fee Request permissions from permissions@acm.org

⁵⁵ ACM MM, 2024, Melbourne, Australia

^{56 © 2024} Copyright held by the owner/author(s). Publication rights licensed to ACM.

⁵⁷ https://doi.org/10.1145/nnnnnn.nnnnn

inter-view information from multi-view data and numerious ten-sorial MVC methods are developed [24, 27, 31, 44]. Inspired by their impressive performance, we in this paper extend deep NMF for multi-view clustering to deep non-negative tensor factorization (DNTF) and propose a novel multi-View clustering with deep tensor factorization (MVC-DTF). MVC-DTF not only takes into account the differences and complementarities between views but also en-ables the learning of deeper representations of the data through the deep tensor factorization method, so as to capture and utilize the complex structural information in the multi-view data in a more effective way. In addition to this, we use weighted tensor Schatten *p*-norm to describe the clustering structure of the view data, which fully utilizes the complementary information between views and avoids the problem that the use of Schatten *p*-norm in [24] leads to the same contribution of singular values across views. To further improve the efficiency, we perform tensor factorizaton on the anchor graph instead of the original data tensor. The main contributions of this paper are as follows:

- We extend deep non-negative matrix factorization to deep tensor factorization, based on which we propose a novel multi-view clustering method, called MVC-DTF. By further introducing a Schatten *p*-norm based regularizer, MVC-DTF can better exploit the inter-view spatial relationship and the complementary information of different/views.
- We adopt anchor graph to construct the tensor to be factorized to reduce the computational complexity. Besides, we develop an optimization algorithm to solve the MVC-DTF optimization problem.
 - Extensive experiments are conducted on several multi-view datasets to evaluate the prformance of our method, and the comparison with existing MVC methods demonstrate its effectiveness and superiority.

2 RELATED WORK

In this section, we introduce the concepts and formulas of multiview clustering using non-negative matrix factorization and nonnegative tensor factorization respectively, which is tightly related to our method.

2.1 Non-negative Matrix Factorization (NMF)

NMF has become an important feature dimensionality reduction tool in clustering. NMF provides an intuitive way to explore the low-dimensional structure of the data by decomposing the nonnegative data matrix **X** into a weight matrix **W** and an indicator matrix **H**. Its formula is expressed as follows:

$$\min_{\mathbf{H},\mathbf{W}} \|\mathbf{X} - \mathbf{H}\mathbf{W}\|_F^2 \text{, s.t. } \mathbf{W} \ge 0, \mathbf{H} \ge 0 \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{H} \in \mathbb{R}^{n \times k}$ and $\mathbf{W} \in \mathbb{R}^{k \times d}$. Furthermore, Semi-NMF improvies NMF by relaxing the non-negative constraint of \mathbf{W} , which has been proven to be a relaxation of the K-means clustering approach [10]. Specifically, Semi-NMF can be expressed as follows:

$$\min_{\mathbf{H},\mathbf{W}} \|\mathbf{X} - \mathbf{H}\mathbf{W}\|_F^2 \text{, s.t. } \mathbf{H} \ge 0$$
(2)

2.2 Tensor Factorization

Kilmer *et al* [21] extended NMF to third-order tensors. Specifically, they present a new third-order tensors representation and introduce a closed tensor multiplication operation, based on which they defined tensor transpose, inverse, and unit tensor.

DEFINITION 1 ([21]). For a 3-way tensor $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the Frobenius norm of \mathbf{A} is defined as $\|\mathbf{A}\|_F = \sqrt{\sum_{ijk} |a_{ijk}|^2}$, and the conjugate transpose of $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is $\mathbf{A}^T \in \mathbb{R}^{n_2 \times n_1 \times n_3}$.

DEFINITION 2 ([21]). For a 3-way tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we denote \overline{A} as a block diagonal matrix with each block on the diagonal as the frontal slice $\overline{A}^{(i)}$ of $\overline{\mathcal{A}}$. \overline{A} has the following form:

$$\overline{\mathbf{A}} = \mathrm{bdiag}(\overline{\boldsymbol{\mathcal{A}}}) = \begin{bmatrix} \overline{\mathbf{A}}^{(1)} & & \\ & \overline{\mathbf{A}}^{(2)} & & \\ & & \ddots & \\ & & & \overline{\mathbf{A}}^{(n_3)} \end{bmatrix}$$

DEFINITION 3 (T-PRODUCT [21]). : The t-product between two 3 -order tensors with matched dimensions, $\mathcal{M} \in \mathbb{R}^{n_1 \times m \times v}$ and $\mathcal{N} \in \mathbb{R}^{m \times n_2 \times v}$, is defined as $\mathcal{M} * \mathcal{N} \in \mathbb{R}^{n_1 \times n_2 \times v}$, i.e.,

$$\mathcal{M} * \mathcal{N} = \operatorname{ifft}(\operatorname{bdiag}(\overline{MN}))$$

where $\overline{\mathcal{M}}$ is the discrete Fourier transform of \mathcal{M} along the third dimension, $\overline{\mathcal{M}} = \operatorname{fft}(\mathcal{M}, [], 3)$, $\overline{M} = \operatorname{bdiag}(\overline{\mathcal{M}})$ and $\operatorname{bdiag}(\cdot)$ denotes the block diagonal matrix.



Figure 1: Tensor Construction and Weighted Tensor Schatten *p*-norm

3 THE PROPOSED METHOD

In this section, we first present the motivation for our proposed scheme. After that, we present the details of our proposed multiview clustering scheme based on deep tensor factorization and the optimization method of the scheme. Finally, we summarize our scheme and give the time complexity analysis of the scheme. We show some key notations used in the scheme in Table 1 for easier presentation.

3.1 Objective

In recent years, tensor factorization has emerged as a powerful tool for multi-view data processing, since it can better explore Multi-View Clustering Based on Deep Non-Negative Tensor Factorization

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

Notations Description Μ Matrix with upper bold case letters \mathcal{M} Tensor with bold calligraphy letters $\overline{\mathcal{M}}$ Discrete Fourier transform of \mathcal{M} along the third dimension A (v) th frontal alian of A matrix MМ or processing iyer i • 1 r *i* > 1 $\cdot_2 * \ldots * \mathcal{W}_1$ ten *p*-norm

the ne spatial structure wit y to deal with the NMF, the objective cor d as follows: fun

$$\min_{\boldsymbol{\mathcal{H}},\boldsymbol{\mathcal{W}}} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{H}} * \boldsymbol{\mathcal{W}}\|_{F}^{2}$$
s.t. $\boldsymbol{\mathcal{H}} \ge 0, \boldsymbol{\mathcal{H}}^{\mathrm{T}} * \boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{I}}$
(3)

where $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n \times d \times v}$ is the original data tensor obtained by combining the data matrices of the different views. $\mathcal{H} \in \mathbb{R}^{n \times k \times v}$ is the cluster indicator tensor. $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{k \times d \times v}$ is the cluster weight tensor. This fusion of data from different views into a tensor for factorization can naturally leverage the relationships between views [24]. Fig. 1 depicts the construction of tensor by taking \mathcal{H} as an exmple. However, despite tensor factorization's theoretical superiority, its ability to handle nonlinear structures and learn deep representations is still limited. Inspired by deep learning, this paper presents an innovative deep tensor factorization framework to overcome the above challenges. By incorporating the concepts of deep learning into the tensor factorization process, our approach is able to not only capture the multidimensional structure of data but also learn deep and nonlinear representations of data.

$$\min_{\mathcal{H}_{l}, \mathcal{W}_{i}} \| \mathcal{X} - \mathcal{H}_{l} * \mathcal{W}_{l} * \mathcal{W}_{l-1} * \dots * \mathcal{W}_{1} \|_{F}^{2}$$

s.t. $\mathcal{H}_{l} \ge 0, \mathcal{H}_{I}^{\mathrm{T}} * \mathcal{H}_{l} = \mathcal{I}$ (4)

The deep tensor factorization method enables us to more effectively process multi-view data by extracting the deep and potential correlations between different views. The following experiments section proves that our method can significantly improve the performance of multi-view clustering.

Considering that directly performing deep tensor factorization on the original data matrix incurs high computational complexity, we choose to construct a anchor graph for each view, and a anchor tensor $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{n \times m \times v}$ constructed from the anchor graphs of different views to replace the original data tensor $\mathcal{X} \in \mathbb{R}^{n \times d \times v}$

as the input to the tensor factorization, and *m* is the number of anchors. Since the number of anchors is much smaller than the original data dimension (i.e., $m \ll d$), it can effectively reduce the computational overhead.

To further exploit the complementary information of different views, we further minimize the tensor schatten p norm of the tensorial form of the cluster indicator matrix, and Eq. (4) becomes:

$$\min_{\mathcal{H}_{l}, \mathcal{W}_{l}} \| \mathcal{S} - \mathcal{H}_{l} * \mathcal{W}_{l} * \mathcal{W}_{l-1} * \dots * \mathcal{W}_{l} \|_{F}^{2} + \lambda \| \mathcal{H}_{l} \|_{\omega, Sp}$$
s.t. $\mathcal{H}_{l} \ge 0, \mathcal{H}_{l}^{\mathrm{T}} * \mathcal{H}_{l} = \mathcal{I}$
(5)

The second term of objective function is regularization term based on weighted tensor Schatten *p*-norm of \mathcal{H}_l , and λ is a balancing hyperparameter. The weighted tensor Schatten p-norm is defined as follows

DEFINITION 4 (WEIGHTED TENSOR SCHATTEN p-NORM [16]). Given $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, Weighted tensor Schatten p-norm of \mathcal{X} is defined as

$$\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{\omega},S_{p}} = \left(\sum_{i=1}^{n_{3}} \left\|\boldsymbol{\overline{\mathcal{X}}}^{(i)}\right\|_{\boldsymbol{\omega},S_{p}}^{p}\right)^{\frac{1}{p}}$$
$$= \left(\sum_{i=1}^{n_{3}} \sum_{j=1}^{\min(n_{1},n_{2})} \omega_{j} * \sigma_{j} \left(\boldsymbol{\overline{\mathcal{X}}}^{(i)}\right)^{p}\right)^{\frac{1}{p}}.$$
(6)

where $0 \le p \le 1$, $\sigma_i(\overline{\mathcal{X}}^{(i)})$ is the *j*-th singular value of $\overline{\mathcal{X}}^{(i)}$, ω_i is the weight of $\sigma_i(\overline{\boldsymbol{\mathcal{X}}}^{(i)})$.

It is worth noting that the aim of weighted tensor Schatten pnorm for the regularization is to avoid the problem of all singular values contributing equally in ordinary tensor Schatten p-norm, which provides better robustness to data noise or outliers. Based which provides better robustices to an on [27], we update the weights by $\left(\omega_j^k\right)^{t+1} = \frac{1}{\left(\delta_j^p(\mathbf{x}_f^k)\right)^{t} + \varepsilon}$, where tand ε denote the number of iteration and a small constant.

3.2 Initialization

Before optimizing the objective equation, following the work on deep non-negative matrix factorization for multi-view clustering [42, 43], we first initialize layer after layer of views. We first get $\mathcal{W}_1^{(v)}$ by $\mathcal{S}^{(v)} = \mathcal{H}_1^{(v)} * \mathcal{W}_1^{(v)}$ and then factorize $\mathcal{H}_1^{(v)} = \mathcal{H}_2^{(v)} * \mathcal{W}_2^{(v)}$ layer by layer up to \mathcal{W}_l . Besides, We initialize $\mathcal{Y}_1 = \mathcal{Y}_2 = 0$ and \overline{N} as the unit matrix.

3.3 Optimization

We use the Augmented Lagrange Multiplier (ALM) to optimize the objective Eq. (5), To perform an alternating direction minimization strategy, we introduce two auxiliary tensor variables $\mathcal{H}_l = \mathcal{N}, \mathcal{H}_l = \mathcal{M}$ where $\mathcal{N} \ge 0$ are used, and then, the objective function can be reformularized to the following unconstrained optimization problem:

348

Table 1: Notations

	v-th frontal slice of JVI						
$tr(M), M^T$	Transpose and trace of						
$\ \mathcal{M}\ _F$	The F-norm of tensor						
$oldsymbol{\mathcal{S}} \in \mathbb{R}^{n imes m imes v}$	Data tensor after anch						
$oldsymbol{\mathcal{H}}_i \in \mathbb{R}^{n imes l_i imes v}$	Indication tensor for la						
$\mathcal{W}_1 \in \mathbb{R}^{l_1 imes m imes v}$	Weight tensor for laye						
$\boldsymbol{\mathcal{W}}_i \in \mathbb{R}^{l_i imes l_{i-1} imes v}$	Weight tensor for laye						
ψ_i	Short for $\mathcal{W}_{i-1} * \mathcal{W}_{i-1}$						
$\ \mathcal{X}\ _{\omega,S_p}$	Weighted tensor Schat						
\mathcal{M},\mathcal{N}	Auxiliary Variables						
$oldsymbol{\mathcal{Y}}_1,oldsymbol{\mathcal{Y}}_2$	Lagrange multipliers						
relationship betwe	en different views and th						
hin the view, whicl	h provides the possibilit						
relation between m	ultiple views. Similar to I						
ction of tensor facto	orization can be describe						

$$\min \mathcal{L}(\mathcal{H}_{l}, \mathcal{W}_{1}, \mathcal{W}_{2}, \dots, \mathcal{W}_{l}, \mathcal{N}, \mathcal{M})$$

$$= \min_{\mathcal{N} \ge 0, \mathcal{H}_{l}^{\mathrm{T}} \mathcal{H}_{l} = \mathcal{I}} \|\mathcal{S} - \mathcal{H}_{l} * \mathcal{W}_{l} * \mathcal{W}_{l-1} * \dots * \mathcal{W}_{1}\|_{F}^{2}$$

$$+ \lambda \|\mathcal{M}\|_{\omega, Sp} + \frac{\mu}{2} \|\mathcal{H}_{l} - \mathcal{N} + \frac{\mathcal{Y}_{1}}{\mu}\|_{F}^{2} + \frac{\rho}{2} \|\mathcal{H}_{l} - \mathcal{M} + \frac{\mathcal{Y}_{2}}{\rho}\|_{F}^{2}$$
(7)

where the tensors $\mathcal{Y}_1 \in \mathbb{R}^{n \times k \times v}$ and $\mathcal{Y}_2 \in \mathbb{R}^{n \times k \times v}$ are Lagrange multipliers and μ , ρ are the penalty parameters. The optimization process can therefore be separated into five steps:

 \mathcal{W}_i -subproblem: By fixing $\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_{i-1}, \mathcal{N}, \mathcal{H}_i, \mathcal{M}$, the optimization Eq.(7) becomes:

$$\min \|\boldsymbol{\mathcal{S}} - \boldsymbol{\mathcal{H}}_i * \boldsymbol{\mathcal{W}}_i * \psi_{i-1}\|_F^2 \tag{8}$$

where $\psi_{i-1} = \mathcal{W}_{i-1} * \mathcal{W}_{i-2} * \ldots * \mathcal{W}_1$. First, implement the discrete Fourier transform (DFT) along the third dimension, the equivalent representation of Eq.(8) in the frequency domain becomes:

$$\min \sum_{v=1}^{V} \|\overline{\boldsymbol{\mathcal{S}}}^{(v)} - \overline{\boldsymbol{\mathcal{H}}_i}^{(v)} \overline{\boldsymbol{\mathcal{W}}_i}^{(v)} \overline{\boldsymbol{\mathcal{W}}_i}^{(v)} \|_{j=1}^{2}$$

where $\overline{\mathcal{W}} = \operatorname{ftt}(\mathcal{W}, [], 3)$, and the others in the same way. So Making the equation zero gets the solution of Eq.(8) is:

$$\overline{\mathcal{W}_{i}}^{(v)} = \overline{\mathcal{H}_{l}}^{(v)\dagger} \overline{\mathcal{S}}^{(v)} \overline{\psi_{i-1}}^{(v)\dagger}$$
(9)

where $\overline{\mathcal{H}_l}^{(v)\dagger}$ denotes MP inverse of $\overline{\mathcal{H}_l}^{(v)}$ [10].

 \mathcal{H}_i -subproblem: By fixing $\psi_i, \mathcal{N}, \mathcal{M}$, the optimization Eq.(7) becomes:

$$\min \|\boldsymbol{\mathcal{S}} - \boldsymbol{\mathcal{H}}_i * \boldsymbol{\psi}_i\|_F^2 \tag{10}$$

Same as \mathcal{W}_i . First, perform the DFT of the equivalent representation of Eq.(10) in the frequency domain becomes:

$$\min \sum_{v=1}^{V} \|\overline{\boldsymbol{\mathcal{S}}}^{(v)} - \overline{\boldsymbol{\mathcal{H}}_{i}}^{(v)} \overline{\psi_{i}}^{(v)}\|_{F}^{2}$$

According to [10] the update rule for $\overline{\mathcal{H}_i}^{(v)}$ (i < l) is

$$\overline{\mathcal{H}_{i}}^{(v)} = \overline{\mathcal{H}_{i}}^{(v)}$$

$$\odot \sqrt{\frac{\left[\overline{\psi_{i}}^{(v)}(\overline{\mathcal{S}}^{(v)})^{T}\right]^{+} + \left[\overline{\psi_{i}}^{(v)}(\overline{\psi_{i}}^{(v)})^{T}(\overline{\mathcal{H}_{i}}^{(v)})^{T}\right]^{-}}{\left[\overline{\psi_{i}}^{(v)}(\overline{\mathcal{S}}^{(v)})^{T}\right]^{-} + \left[\overline{\psi_{i}}^{(v)}(\overline{\psi_{i}}^{(v)})^{T}(\overline{\mathcal{H}_{i}}^{(v)})^{T}\right]^{+}}}$$
(11)

Where $[A]^+ = (|A| + A)/2, [A]^- = (|A| - A)/2.$ \mathcal{H}_l -subproblem: By fixing $\mathcal{N}, \psi_l, \mathcal{M}$ Eq.(7) becomes:

$$\min_{\boldsymbol{\mathcal{H}}_{l}^{\mathrm{T}}\boldsymbol{\mathcal{H}}_{l}=\boldsymbol{\mathcal{I}}} \|\boldsymbol{\mathcal{S}}-\boldsymbol{\mathcal{H}}_{l}\boldsymbol{\psi}_{l}\|_{F}^{2} + \frac{\mu}{2} \|\boldsymbol{\mathcal{H}}_{l}-\boldsymbol{\mathcal{N}}+\frac{\boldsymbol{\mathcal{Y}}_{l}}{\mu}\|_{F}^{2} + \frac{\rho}{2} \|\boldsymbol{\mathcal{H}}_{l}-\boldsymbol{\mathcal{M}}+\frac{\boldsymbol{\mathcal{Y}}_{2}}{\rho}\|_{I}^{2}$$
(12)

Eq.(12) can be reduced to

$$\max_{\overline{\mathcal{H}_{l}}^{(v)}} \operatorname{tr}\left(\left(\overline{\mathcal{H}_{l}}^{(v)}\right)^{\mathrm{T}} \overline{\mathcal{B}}^{(v)}\right)$$
(13)

where $\overline{\mathcal{B}}^{(v)} = 2\overline{\mathcal{S}}^{(v)} \left(\overline{\psi}_{l}^{(v)}\right)^{\mathrm{T}} + \mu \overline{\mathcal{Q}}_{1}^{(v)} + \rho \overline{\mathcal{Q}}_{2}^{(v)}, \overline{\mathcal{Q}}_{1}^{(v)} = \overline{\mathcal{N}}^{(v)} - \overline{\mathcal{Q}}_{1}^{(v)}$ $\frac{\overline{\boldsymbol{\mathcal{Y}}}_{1}^{(v)}}{\underline{\boldsymbol{\mathcal{U}}}}$ and $\overline{\boldsymbol{\mathcal{Q}}}_{2}^{(v)} = \overline{\boldsymbol{\mathcal{M}}}^{(v)} - \frac{\overline{\boldsymbol{\mathcal{Y}}}_{2}^{(v)}}{\rho}$

According to [24], the solution of Eq.(13)is

$$\overline{\mathcal{H}_{l}}^{(v)} = \overline{\Lambda}^{(v)} \left(\overline{V}^{(v)}\right)^{\mathrm{I}}$$
(14)

where $\overline{\Lambda}^{(v)}$ and $\overline{V}^{(v)}$ can be obtained by SVD $\overline{\mathcal{B}}^{(v)} = \overline{\Lambda}^{(v)} \mathbf{X} \left(\overline{V}^{(v)} \right)^{\mathrm{T}}$ \mathcal{N} -subproblem: By fixing $\mathcal{H}, \mathcal{G}, \mathcal{M}$. Eq.(7) becomes:

$$\min_{\mathcal{N} \ge 0} \frac{\mu}{2} \left\| \mathcal{N} - \left(\mathcal{H} + \frac{\mathcal{Y}_1}{\mu} \right) \right\|_F^2$$
(15)

According to [40], the solution of Eq.(15) is:

$$\mathcal{N} = \left(\mathcal{H} + \frac{\mathcal{Y}_1}{\mu}\right)_+ \tag{16}$$

 \mathcal{M} -subproblem: By fixing $\mathcal{N}, \mathcal{H}, \mathcal{G}$. Eq.(7) becomes:

$$\min \lambda \|\mathcal{M}\|_{\omega,Sp} + \frac{\rho}{2} \left\|\mathcal{H} - \mathcal{M} + \frac{\mathcal{Y}_2}{\rho}\right\|_F^2$$

A simple transformation gives us

Л

$$\boldsymbol{\mathcal{A}}^* = \arg\min\frac{1}{2} \|\boldsymbol{\mathcal{H}} + \boldsymbol{\mathcal{Z}}\|_F^2 + \frac{\lambda}{\rho} \|\boldsymbol{\mathcal{M}}\|_{\omega,Sp}, \qquad (17)$$

where $\mathbf{\mathcal{Z}} = \frac{\mathbf{\mathcal{Y}}_2}{\rho} - \mathbf{\mathcal{M}}$. According to [17], the following lemma can give the solution of Eq.(17)

LEMMA 1. Let $\mathbf{Z} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ satisfy an increasing sequence of $0 \leq \omega_1 \leq \omega_2 \leq \cdots \leq \omega_{\min}$ and have a t-SVD $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}^T$. For the following optimization model:

$$\underset{\boldsymbol{\mathcal{X}}}{\operatorname{argmin}} \frac{1}{2} \| \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Z}} \|_{F}^{2} + \tau \| \boldsymbol{\mathcal{X}} \|_{\omega, S_{p}}^{p}.$$
(18)

a global optimal solution to the 18 is

$$\mathcal{X}^* = \Upsilon_{\tau * \omega}(\mathcal{Z}) = \mathcal{U} * ifft\left(\mathbf{P}_{\tau * \omega}(\overline{\mathcal{Z}})\right) * \mathcal{V}^T,$$

where, $\mathbf{P}_{\tau*\omega}(\overline{\mathbf{Z}})$ is a tensor and $\mathbf{P}_{\tau*\omega}(\overline{\mathbf{Z}}^{(i)})$ is the *i*-th frontal slice of $\mathbf{P}_{\tau*\omega}(\overline{\mathbf{Z}}) = \operatorname{diag}(\gamma_1, \gamma_2, \dots, \gamma_l)$ and $\gamma_i = T_{\mathbf{D}}^{GST}(\sigma_i(\overline{\mathbf{Z}}), \tau* \omega_i)$, the introduction of GST can be found in [17].

Through *Lemma*1, we get the solution of Eq.(17) as

$$\mathcal{M}^* = \Upsilon_{\frac{\lambda}{\rho} * \omega} \left(\mathcal{H} + \frac{\mathcal{Y}_2}{\rho} \right)$$
(19)

The optimization of our multi-view clustering method based on deep tensor factorization can be summarized as Algorithm 1. It is worth noting that we need to optimize \mathcal{H}_i and \mathcal{M}_i layer by layer before obtaining \mathcal{H}_l .

3.4 Computational Complexity

To simplify the analysis, we set the dimension of each layer to be uniformly d and assume that the number of iterations of the Initialization process and training process are t_{pre} and t_{train} . The computational complexity of constructing the anchor data ten-sor from the original data tensor is $\mathcal{O}(vnmd + vnmlog(m))$. The computational complexity of is $\mathcal{O}(vt_{pre}nmd^2l)$ after approximate simplicity. The computational complexity of the update process for $\mathcal{W}_i, \mathcal{H}_i, \mathcal{H}_l, \mathcal{M}, \mathcal{N}$ are $\mathcal{O}(vlt_{train}(dnm+d^2m+d^2n)), \mathcal{O}(vlt_{train}d^2n+d^2n))$ $vlt_{train}dnm), \mathcal{O}(t_{train}v(m^2k+mk^2)), \mathcal{O}(t_{train}vnk), \mathcal{O}(t_{train}(2vnklog(vk)+$ v^2kn)). As these parameters are small constants and m < n, k < d,

Multi-View Clustering Based on Deep Non-Negative Tensor Factorization

ACM MM, 2024, Melbourne, Australia

Algorithm 1 Multi-view Clustering based on Deep Non-negative 465 Tensor Factorization (MVC-DNTF) 466 467 **input:** multi-view data $\{\mathbf{X}^{(v)}\}_{v=1}^{V}$; anchor number *m*; cluster number *k*; number of layers and dimension for each layer; 468 469 output: Cluster labels Y of each sample. 470 1: Initialize: $\mathcal{H}_{i}^{(v)}, \mathcal{W}_{i}^{(v)}, \mu = 10^{-5}, \rho, \eta, \mathcal{Y}_{1}, \mathcal{Y}_{2}, \overline{N}^{(v)}$ according 471 to section 3.2. 472 2: Compute anchor graph matrix $S^{(v)}$ of each view. 473 while not convergence do 3: 474 for i < m do 4: 475 Update $\overline{\mathcal{W}_{i}}^{(v)}$ by solving Eq.(9) Update $\overline{\mathcal{H}_{i}}^{(v)}$ by solving Eq. (11). 5: 476 477 6: end for 478 7: Update $\overline{\mathcal{W}_l}^{(v)}$ by solving Eq.(9). Update $\overline{\mathcal{H}_l}^{(v)}$ by solving Eq.(14). 479 8: 480 9: 481 Update $\overline{\mathcal{N}^{(v)}}$ by solving Eq.(16). 10: 482 Update \mathcal{M} by using Eq.(19). 11: 483 Update $\mathcal{Y}_1, \mathcal{Y}_2, \mu$ and $\rho: \mathcal{Y}_1 = \mathcal{Y}_1 + \mu(\mathcal{H} - \mathcal{N}), \mathcal{Y}_2 = \mathcal{Y}_2 + \mathcal{Y}_2$ 12: 484 $\mu(\mathcal{H} - \mathcal{M}), \mu = \min(\eta \mu, 10^{13}), \rho = \min(\eta \rho, 10^{13}).$ 485 13: end while 486 14: Calculate the *K* clusters by using $\mathbf{H} = \sum_{v=1}^{V} \mathcal{H}^{(v)} / V$. 487 15: return Clustering result (In the resulting indicator matrix, the 488 489

index of the largest element in each row corresponds to the cluster label of the respective sample.).

and in general *n* is the largest number among them, the final computational complexity of our proposed scheme is $\mathcal{O}(vt_{pre}nmd^2l +$ $vlt_{train}dnm + t_{train}vm^2k$)

EXPERIMENTS 4

4.1 Dataset

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

522

We evaluate the performance of the proposed method on eight widely adapted multi-view learning benchmark datasets, which are 3-sources, BBCSport [18], HW [12], Sonar [32], Yale, Vehicle Sensor [11], Caltech-5V [14], SentencesNYU v2(RGB-D) [33]. Details of these datasets are shown in Table 2.

4.2 Compared Method

510 All experiments were conducted on a desktop computer equipped with a 13th Gen Intel(R) Core(TM) i5-13400 processor at 2.50 GHz, 511 512 and 32 GB RAM. The experiments were executed using MATLAB 513 2023a (64-bit) as the primary software environment. For experimental setup, we have selected the following nine representative multi-514 view clustering algorithms to compare with our proposed method: 515 516 Two multi-view clustering methods based on graph learning GMC [36] and UDBGL [13], Three sub-space multi-view learning meth-517 ods DiMSC [5], MvLRSSC [2] and RMSL [26], a fast calculation 518 method FastMICE [19] and two non-negative matrix factorization 519 520 methods MvDGNMF [25] and MVC-DMF-PA [42], and a method 521 for orthogonal non-negative tensor factorization Orth-NTF [24].

4.3 Experimental Setup

Before performing clustering, for all the methods including the proposed method and all the comparison methods, we first preprocess the data of all the datasets, i.e., we normalize the data of different views. For the method we proposed, λ and p are hyper-parameters. According to [42], when performing a two-layer tensor factorization the layer size is $[l_1, k]$, where l_1 is chosen from [4k, 5k, 6k], when performing a three-layer tensor factorization the layer size is $[l_1, l_2, k]$, where l_1 is chosen from [8k, 10k, 12k] and l_2 is chosen from [4k, 5k, 6k] and so on. We used three widely used metrics to evaluate clustering performance i.e., 1) Accuracy (ACC); 2)Normalized Mutual Information(NMI); 3) Purity(PUR). For all the metrics mentioned previously, a higher value indicates better clustering outcomes. For each experiment to avoid the effect of random initialization, we repeated each experiment 10 times.

4.4 Experiment Results

The experimental results in Tables 3 and 4 demonstrate the performance of our method MVC-DNTF on eight datasets and are compared with several benchmark algorithms. The best results are bolded and the second best results are underlined. The main results of our analysis of these results are as follows:

- (1) Our method performs better than the benchmark methods on most of the datasets. We believe that this superior performance is largely attributable to the unique design of our proposed method, which performs deep factorization based on a data tensor consisting of multi-view data anchor graphs. This design allows our method to effectively utilize spatial structure information and depth and hidden features from different views. Meanwhile the imposition of orthogonal and non-negative constraints enhances the interpretability of our clustering, and each row of the metrics matrix for each view is explicitly mapped to a cluster without the need for post-processing for label determination.
- (2) Compared with DMVC and AwDMVC methods that rely on the deep semi-NMF framework, our method achieves superior results. This demonstrates that the tensor factorization method we adopt better preserves the structural information of the data in multiple dimensions and effectively captures the complementary information and intrinsic connections among different views. Compared to Orth-NTF, which also employs tensor factorization, our method also achieves superior performance, which suggests that our proposed deep tensor factorization can automatically learn deeper feature representations from multi-view data, and thus better capture the essential properties and intrinsic structure of the data.

In conclusion, the experimental results highlight the effectiveness of our multi-view clustering method based on deep tensor factorization.

4.5 Convergence

We performed convergence experiments on four different datasets. These experiments were aimed at observing the evolution of the clustering accuracy of the algorithm and the convergence behavior of the algorithm's objective function during successive iterations.

Caltech-5V

SentencesNYU v2 (RGB-D)

Dataset	Туре	Views number	View Dimension	Sample number	Cluster number
3-sources	text	3	3056/3631/3068	169	3
BBCSport	text	2	3283/3183	544	5
ΗŴ	handwritten	6	216/76/64/6/240/47	2000	10
Sonar	signal	3	20/20/20	208	2
Yale	image	2	1024/4096	165	11
Vehicle Sensor	sensor	4	5/5/7/5	1594	2

40/254/1984/512/928

2048/300

Table 3: Clustering performance comparison in terms of ACC(%), NMI(%), and PUR(%) on 3-sources, BBCSport, HW, and Sonar datasets.

image

image

Datasets 3-sources]	3BCSpor	rt	HW			Sonar			
Metrics	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
DiMSC	69.23	63.13	74.56	85.85	70.62	85.85	24.50	12.19	25.85	55.77	1.25	55.77
MvLRSSC	55.92	49.81	70.59	62.87	40.47	64.63	66.88	68.88	70.56	50.48	3.12	53.37
RMSL	31.95	14.46	41.42	76.63	72.36	76.63	81.38	78.82	81.50	62.02	4.29	62.02
GMC	70.74	65.15	79.29	80.33	73.89	84.01	84.80	89.13	87.25	50.48	4.50	53.37
MvDGNMF	30.77	21.11	33.14	82.54	67.32	82.54	75.60	63.83	75.60	62.50	4.66	62.50
UDGBL	34.91	5.60	35.50	36.40	2.43	36.58	67.10	57.42	80.69	57.21	1.61	57.21
FastMICE	55.62	50.25	71.01	43.93	11.16	45.40	85.65	85.04	85.65	58.00	3.23	58.17
MVC-DMF-PA	53.84	20.02	56.21	73.34	52.68	76.28	86.90	76.58	86.90	53.84	53.27	53.84
Orth-NTF	72.78	72.78	77.50	89.15	79.49	89.52	89.35	86.16	89.35	97.11	83.80	97.1
Ours	83.43	76.66	88.16	94.85	87.85	94.85	97.60	95.24	97.60	97.11	83.80	97.1 1

Table 4: Clustering performance comparison in terms of ACC(%), NMI(%), and PUR(%) on Yale, Vehicle Sensor, HAR, and RGB-D datasets.

Datasets	Yale		Vel	nicle Sen	sor	C	altech-5	V	RGB-D			
Metrics	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
DiMSC	44.85	52.84	44.85	68.95	22.29	68.95	57.57	39.76	61.43	39.61	32.67	49.76
MvLRSSC	44.06	48.02	45.09	56.78	6.12	56.78	46.15	34.81	46.79	39.00	32.40	50.59
RMSL	78.78	78.23	79.39	67.50	11.90	67.50	55	52.18	59.07	12.63	2.85	26.98
GMC	21.21	27.51	24.24	80.43	28.75	80.43	34.07	48.4	36.07	40.23	33.06	46.51
MvDGNMF	36.36	42.70	38.79	50.06	0.60	50.06	49.57	38.24	53.86	26.50	0.78	27.26
UDGBL	34.91	5.60	35.50	36.40	2.43	36.58	31.8	23.54	19.28	57.21	1.61	57.21
FastMICE	62.42	57.01	65.46	51.49	0.085	51.69	77.58	69.6	79.57	41.81	32.61	49.53
MVC-DMF-PA	15.75	16.10	20.00	50.37	62.76	50.37	71.64	56.49	71.64	16.83	72.25	33.12
Orth-NTF	78.18	81.90	80.00	98.05	86.23	98.05	89.35	81.64	89.35	59.07	65.78	75.56
Ours	84.24	86.39	82.42	99.62	96.79	99.62	94.21	89.08	94.21	63.21	71.28	82.95

Fig 2 illustrates the convergence process of the algorithm on the four datasets. It is noteworthy that the algorithm converges rapidly after 60 iterations, with a significant decrease in the difference between the objective matrices \mathcal{H}_l , \mathcal{M} , and \mathcal{N} . At the same time, the clustering accuracy improves dramatically, which is consistent with the convergence of the objective values. This proves the effectiveness of our proposed method since the convergence of the

objective function is equivalent to the improvement of the clustering performance.

4.6 Parameter Analysis

Our proposed algorithm depends on three key variable parameters: the number of anchors *m*, λ and *p*. λ and *p* regulate the sparsity and orthology of the potential representation \mathcal{H}_l . The hyperparameter λ plays an important role in the trade-off between the accuracy

Multi-View Clustering Based on Deep Non-Negative Tensor Factorization

ACM MM, 2024, Melbourne, Australia



Figure 3: The influence of λ and p on clustering results on 3-sources, BBCSport, HW, Sonar, Yale , Vehicle Sensor Caltech-5V and RGB-D

of the data reconstruction and the sparsity of the representation. The hyperparameter *p* fine-tunes this sparsity constraint by adjusting the distribution of non-zero values in \mathcal{H}_l , thus affecting the granularity of the representation. In order to study the impact of these hyperparameters in depth, we employ a grid search strategy to study values of *p* in the range [0.1, 1.0] and values of λ in the range [1, 1000]. We performed systematic experiments on eight different datasets to ensure the generalizability of our findings. Fig 3 illustrates the results of our experiments, which show that λ has a significant effect on the efficacy of the algorithm, probably due to its key role in preventing overfitting through regularization, while p has a relatively stable effect on performance, but tends to perform better when *p* takes smaller values than when *p* takes larger values. Empirical results from various datasets indicate that the best results are obtained when λ is calibrated in the middle range 10 to 100 and *p* is taken to be 0.1. This range allows the algorithm to achieve sufficient sparsity for generalization without compromising the ability to capture important data structures.

We set the anchor point rate from 0.1 to 1 with a step size of 0.1, and conduct experiments on the four datasets 3-sources, BBCSport,

Yale , RGB-D to test the effect of anchor point rate on the clustering results. As shown in Fig. 4, we can find that the clustering performance is improved with the increase of the number of anchors, but the anchor rate has little effect on the experimental results after the anchor rate is greater than 0.3. The four datasets achieve the best clustering results on 1.0, 0.9, 1.0, 0.8 respectively.

4.7 Ablation Study

We primarily focused on exploring how different depths of deep tensor factorization affect clustering performance for multi-view data. We executed numerous experiments using eight benchmark datasets as described in Section 4.1. These experiments varied the number of layers in the models to examine their influence on clustering results. The depths ranged from one to four layers, with configurations from [k] to $[l_1, l_2, l_3, k]$. This setup allowed us to evaluate the effectiveness of deep tensor factorization at various layer depths. Results detailing the experimental accuracy rates are presented in Table 5.

p	Caltech-5V	BBCSport	Yale	Vehicle Sensor	RGB-D	Sonar	HW	3-sources
[<i>k</i>]	90.64	89.52	79.39	98.05	59.07	97.11	92.80	68.04
$[l_2, k]$	90.64	89.15	80.00	98.05	61.07	97.11	97.60	76.33
$[l_1, l_2, k]$	94.21	94.85	81.21	99.62	62.59	97.11	91.60	83.43
$[l_1, l_2, l_3, k]$	89.57	90.07	84.24	98.05	63.21	97.11	96.35	75.73

Table 5: ACC(%) of different layers on eight benchmark datasets.



Figure 4: Clustering results with different anchor rate on 3-sources, BBCSport, Yale, RGB-D

On most datasets, the accuracy rate shows different trends as the layers deepens. For example, on the Caltech-5V, BBCSport, Vehicle Sensor 3-sources and HW dataset, the highest correctness rate was achieved using a three-layer tensor factorization ($[l_1, l_2, k]$), while four-layer tensor factorization ($[l_1, l_2, k]$) performed best on the Yale and RGB-D datasets.

However, overfitting may occur as the number of factorization layers increases . On the Caltech-5V dataset, the accuracy using two layers was essentially the same as when using only one layer , increasing to three layers the accuracy went up but then declined when going to four layers and the same happens on the BBCSport and HW datasets.

On the Vehicle Sensor and Sonar datasets, the performance improvement with increasing layers is not significant, which we attribute to the simpler structure of these datasets, where the accuracy of the basic tensor factorization is already close to 100, and thus it is difficult to achieve a large improvement in the deeper factorization.

In general, the deep tensor factorization method significantly improves the experimental results, which proves that our proposed multi-view clustering method with deep tensor factorization can effectively extract the deep and hidden features of the data.

Besides, We conducted ablation experiments on deep tensor factorization and weighted tensor Schatten *p*-norm in Table 6. case1 indicates whether deep tensor factorization or single-layer tensor factorization was used. due to the lack of consistency constraints of the tensor factorization methods alone, it is difficult to learn consistent cluster representation matrices, and the effect is poor and difficult to compare, so we set case2 as the use of the weighted tensor Schatten *p*-norm or the use of the ordinary Schatten *p*norm.It can be found that the weighted Schatten *p*-norm is overall superior to the Schatten *p*-norm because the weighted Schatten *p*norm capitalizes on certain important information corresponding to larger singular values while portraying complementary information from different views [27].

Table 6: ACC(%) of ablation experiments

		Datasets							
case1	case2	BBCSport	Yale	3-sources	Caltech-5V				
×	×	89.15	78.18	72.78	89.35				
×	\checkmark	89.52	79.39	68.04	90.64				
\checkmark	×	91.91	78.18	81.06	90.00				
\checkmark	\checkmark	94.85	84.24	83.43	94.21				

CONCLUSION

We propose an novel multi-view clustering method based on deep tensor factorization named (MVC-DTF). By extending deem NMF to deep tensor factorization, MVC-DTF can effectively extract the deep and nonlinear features as well as the spatial informationof multi-view data . In addition, by introducing the weighted tensor Schatten *p*-norm of the clustering indicator matrix, the inter-view correlation and the complementary information of each views are well explored. We also develop an optimization algorithm for our proposed method. Extensive experiments and comparison with several state-of-the-art multi-view clustering method are conducted, whose results demonstrate the effectiveness and superiority of the method.

REFERENCES

- Steffen Bickel and Tobias Scheffer. 2004. Multi-view clustering.. In ICDM, Vol. 4. Citeseer. 19–26.
- [2] Maria Brbić and Ivica Kopriva. 2018. Multi-view low-rank sparse subspace clustering. Pattern Recognition 73 (2018), 247–258.
- [3] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. 2010. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence* 33, 8 (2010), 1548–1560.

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

ACM MM, 2024, Melbourne, Australia

- [4] Deng Cai, Xiaofei He, Xiaoyun Wu, and Jiawei Han. 2008. Non-negative Matrix Factorization on Manifold. In 2008 Eighth IEEE International Conference on Data Mining. 63–72. https://doi.org/10.1109/ICDM.2008.57
- [5] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. 2015. Diversity-induced multi-view subspace clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition. 586–594.
- [6] Man-Sheng Chen, Ling Huang, Chang-Dong Wang, Dong Huang, and Jian-Huang Lai. 2021. Relaxed multi-view clustering in latent embedding space. *Information Fusion* 68 (2021), 8–21.
- [7] Andrzej Cichocki, Rafal Zdunek, et al. 2006. Multilayer nonnegative matrix factorisation. ELECTRONICS LETTERS-IEE 42, 16 (2006), 947.
- [8] Chris Ding, Xiaofeng He, and Horst D Simon. 2005. On the equivalence of nonnegative matrix factorization and spectral clustering. In Proceedings of the 2005 SIAM international conference on data mining. SIAM, 606–610.
- [9] Chris Ding, Tao Li, Wei Peng, and Haesun Park. 2006. Orthogonal nonnegative matrix t-factorizations for clustering. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 126–135.
- [10] Chris HQ Ding, Tao Li, and Michael I Jordan. 2008. Convex and semi-nonnegative matrix factorizations. IEEE transactions on pattern analysis and machine intelligence 32, 1 (2008), 45–55.
- [11] Marco F Duarte and Yu Hen Hu. 2004. Vehicle classification in distributed sensor networks. J. Parallel and Distrib. Comput. 64, 7 (2004), 826–838.
- [12] Robert Duin. [n. d.]. Multiple Features. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5HC70.
- [13] Si-Guo Fang, Dong Huang, Xiao-Sha Cai, Chang-Dong Wang, Chaobo He, and Yong Tang. 2023. Efficient multi-view clustering via unified and discrete bipartite graph learning. *IEEE Transactions on Neural Networks and Learning Systems* (2023).
- [14] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In 2004 conference on computer vision and pattern recognition workshop. IEEE, 178–178.
- [15] Lele Fu, Pengfei Lin, Athanasios V Vasilakos, and Shiping Wang. 2020. An overview of recent multi-view clustering. *Neurocomputing* 402 (2020), 148–161.
- [16] Quanxue Gao, Pu Zhang, Wei Xia, Deyan Xie, Xinbo Gao, and Dacheng Tao. 2020. Enhanced tensor RPCA and its application. *IEEE transactions on pattern analysis and machine intelligence* 43, 6 (2020), 2133–2140.
- [17] Quanxue Gao, Pu Zhang, Wei Xia, Deyan Xie, Xinbo Gao, and Dacheng Tao. 2021. Enhanced Tensor RPCA and its Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 6 (2021), 2133–2140. https://doi.org/10. 1109/TPAMI.2020.3017672
- [18] Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In Proceedings of the 23rd international conference on Machine learning. 377–384.
- [19] Dong Huang, Chang-Dong Wang, and Jian-Huang Lai. 2023. Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity. *IEEE Transactions on Knowledge and Data Engineering* (2023).
- [20] Yu Jiang, Jing Liu, Zechao Li, Peng Li, and Hanqing Lu. 2013. Co-regularized PLSA for multi-view clustering. In Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part II 11. Springer, 202–213.
- [21] Misha E Kilmer and Carla D Martin. 2011. Factorization strategies for third-order tensors. Linear Algebra Appl. 435, 3 (2011), 641–658.
- [22] Abhishek Kumar and Hal Daumé. 2011. A co-training approach for multi-view spectral clustering. In Proceedings of the 28th international conference on machine learning (ICML-11). 393–400.
- [23] Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *nature* 401, 6755 (1999), 788–791.
- [24] Jing Li, Quanxue Gao, Qianqian Wang, Ming Yang, and Wei Xia. 2024. Orthogonal non-negative tensor factorization based multi-view clustering. Advances in Neural Information Processing Systems 36 (2024).
- [25] Jianqiang Li, Guoxu Zhou, Yuning Qiu, Yanjiao Wang, Yu Zhang, and Shengli Xie. 2020. Deep graph regularized non-negative matrix factorization for multi-view clustering. *Neurocomputing* 390 (2020), 108–116.
- [26] Ruihuang Li, Changqing Zhang, Huazhu Fu, Xi Peng, Tianyi Zhou, and Qinghua Hu. 2019. Reciprocal multi-layer subspace learning for multi-view clustering. In Proceedings of the IEEE/CVF international conference on computer vision. 8172– 8180.
- [27] Xingfeng Li, Zhenwen Ren, Quansen Sun, and Zhi Xu. 2023. Auto-weighted tensor schatten p-norm for robust multi-view graph clustering. *Pattern Recognition* 134 (2023), 109083.
- [28] Yun Liu, Feiping Nie, Jigang Wu, and Lihui Chen. 2013. Efficient semi-supervised feature selection with noise insensitive trace ratio criterion. *Neurocomputing* 105 (2013), 12–18.
- [29] Weihua Ou, Shujian Yu, Gai Li, Jian Lu, Kesheng Zhang, and Gang Xie. 2016. Multi-view non-negative matrix factorization by patch alignment framework with view consistency. *Neurocomputing* 204 (2016), 116–124.

- [30] Erlin Pan and Zhao Kang. 2021. Multi-view contrastive graph clustering. Advances in neural information processing systems 34 (2021), 2148–2159.
- [31] Zhenwen Ren, Mithun Mukherjee, Mehdi Bennis, and Jaime Lloret. 2020. Multikernel clustering via non-negative matrix factorization tailored graph tensor over distributed networks. *IEEE Journal on Selected Areas in Communications* 39, 7 (2020), 1946–1956.
- [32] Terry Sejnowski and R. Gorman. [n.d.]. Connectionist Bench (Sonar, Mines vs. Rocks). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5T01Q.
- [33] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12. Springer, 746–760.
- [34] Grigorios Tzortzis and Aristidis Likas. 2012. Kernel-based weighted multi-view clustering. In 2012 IEEE 12th international conference on data mining. IEEE, 675– 684.
- [35] Boyue Wang, Yongli Hu, Junbin Gao, Yanfeng Sun, Fujiao Ju, and Baocai Yin. 2020. Learning adaptive neighborhood graph on Grassmann manifolds for video/imageset subspace clustering. *IEEE Transactions on Multimedia* 23 (2020), 216–227.
- [36] Hao Wang, Yan Yang, and Bing Liu. 2019. GMC: Graph-based multi-view clustering. IEEE Transactions on Knowledge and Data Engineering 32, 6 (2019), 1116–1129.
- [37] Jing Wang, Feng Tian, Hongchuan Yu, Chang Hong Liu, Kun Zhan, and Xiao Wang. 2017. Diverse non-negative matrix factorization for multiview data representation. *IEEE transactions on cybernetics* 48, 9 (2017), 2620–2632.
- [38] Yu-Xiong Wang and Yu-Jin Zhang. 2012. Nonnegative matrix factorization: A comprehensive review. IEEE Transactions on knowledge and data engineering 25, 6 (2012), 1336–1353.
- [39] Xiaokai Wei, Bokai Cao, and S Yu Philip. 2017. Multi-view unsupervised feature selection by cross-diffused matrix alignment. In 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 494–501.
- [40] Ben Yang, Xuetao Zhang, Feiping Nie, Fei Wang, Weizhong Yu, and Rong Wang. 2020. Fast multi-view clustering via nonnegative and orthogonal factorization. *IEEE Transactions on Image Processing* 30 (2020), 2575–2586.
- [41] Yan Yang and Hao Wang. 2018. Multi-view clustering: A survey. Big Data Mining and Analytics 1, 2 (2018), 83–107.
- [42] Chen Zhang, Siwei Wang, Jiyuan Liu, Sihang Zhou, Pei Zhang, Xinwang Liu, En Zhu, and Changwang Zhang. 2021. Multi-view clustering via deep matrix factorization and partition alignment. In Proceedings of the 29th ACM international conference on multimedia. 4156–4164.
- [43] Handong Zhao, Zhengming Ding, and Yun Fu. 2017. Multi-view clustering via deep matrix factorization. In Proceedings of the AAAI conference on artificial intelligence, Vol. 31.
- [44] Yujiao Zhao, Yu Yun, Xiangdong Zhang, Qin Li, and Quanxue Gao. 2022. Multiview spectral clustering with adaptive graph learning and tensor schatten p-norm. *Neurocomputing* 468 (2022), 257–264.
- [45] Xiao Zheng, Chang Tang, Xinwang Liu, and En Zhu. 2023. Multi-view clustering via matrix factorization assisted k-means. *Neurocomputing* 534 (2023), 45–54.
- [46] Linlin Zong, Xianchao Zhang, and Xinyue Liu. 2018. Multi-view clustering on unmapped data via constrained non-negative matrix factorization. *Neural Networks* 108 (2018), 155–171.

1043 1044

987

988

989

990

991

992

993

994

995

996

997