



Deep K -Means: A Simple and Effective Method for Data Clustering

Shudong Huang^{1(✉)}, Zhao Kang², and Zenglin Xu^{2,3,4}

¹ College of Computer Science, Sichuan University, Chengdu 610065, China
huangsd@scu.edu.cn

² School of Computer Science and Engineering,
University of Electronic Science and Technology of China,
Chengdu 611731, China
zkang@uestc.edu.cn

³ School of Computer Science and Technology,
Harbin Institute of Technology, Shenzhen 518055, China
xuzenglin@hit.edu.cn

⁴ Center for Artificial Intelligence, Peng Cheng Lab, Shenzhen 518055, China

Abstract. Clustering is one of the most fundamental techniques in statistic and machine learning. Due to the simplicity and efficiency, the most frequently used clustering method is the k -means algorithm. In the past decades, k -means and its various extensions have been proposed and successfully applied in data mining practical problems. However, previous clustering methods are typically designed in a single layer formulation. Thus the mapping between the low-dimensional representation obtained by these methods and the original data may contain rather complex hierarchical information. In this paper, a novel deep k -means model is proposed to learn such hidden representations with respect to different implicit lower-level characteristics. By utilizing the deep structure to conduct k -means hierarchically, the hierarchical semantics of data is learned in a layerwise way. The data points from same class are gathered closer layer by layer, which is beneficial for the subsequent learning task. Experiments on benchmark data sets are performed to illustrate the effectiveness of our method.

Keywords: Clustering · Matrix factorization · Deep learning

1 Introduction

The goal of clustering is to divide a given dataset into different groups such that similar instances are allocated into one group [22, 33, 34]. Clustering is one of the most classical techniques that has been found to perform surprisingly well [15, 19, 21]. Clustering has been successfully utilized in various application areas, text mining [16, 36], voice recognition [1], image segmentation [38], to name a few. Up to now, myriads of clustering methods have been designed under the framework of different methodologies and statistical theories [30, 31],

like k -means clustering [26], spectral clustering [29], information theoretic clustering [9], energy clustering [37], discriminative embedded clustering [11], multi-view clustering [12, 20, 23, 32], *etc.* Among them, k -means, as one of the most popular clustering algorithms, has received considerable attention due to its efficiency and effectiveness since it was introduced in 1967 [26]. Furthermore, it has been categorized as one of the top ten data mining algorithms in term of usage and clustering performance [39]. There is no doubt that k -means is the most popularly used clustering method in various practical problems [13].

Recently, the Nonnegative Matrix Factorization (NMF) draws much attention in data clustering and achieves promising performance [2, 14, 18]. Previous work indicated that NMF is identical to k -means clustering with a relaxed condition [8]. Until now several variants of k -means have been presented to improve the clustering accuracy. Inspired by the principal component analysis (PCA), [6] shown that principal components actually provide continuous solutions, which can be treated as the discrete class indicators for k -means clustering. Moreover, the subspace separated by the cluster centroids can be obtained by spectral expansion. [5] designed a spherical k -means for text clustering with good performance in terms of both solution quality and computational efficiency by employing cosine dissimilarities to perform prototype-based partitioning of weight representations. [10] extended a kernel k -means clustering to handle multi-view datasets with the help of multiple kernel learning. To explore the sample-specific attributes of data, the authors focused on combining kernels in a localized way. [27] proposed a fast accelerated exact k -means, which can be considered as a general improvement of existing k -means algorithms with better estimates of the distance bounds. [28] assumed that incorporates distance bounds into the mini-batch algorithm, data should be preferentially reused. That is, data in a mini-batch at current iteration is reused at next iteration automatically by utilizing the nested mini-batches.

Although the aforementioned k -means methods have shown their effectiveness in many applications, they are typically designed in a single layer formulation. As a result, the mapping between the low-dimensional representation obtained by these methods and the original data may still contain complex hierarchical information. Motivated by the development of deep learning that employs multiple processing layers to explore the hierarchical information hidden in data [3], in this paper, we propose a novel deep k -means model to learn the hidden information with respect to multiple level characteristics. By utilizing the deep structure to conduct k -means hierarchically, the data hierarchical semantics is learned in a layerwise way. Through the deep k -means structure, instances from same class are pushed closer layer by layer, which is beneficial for the subsequent learning task. Furthermore, we introduce an alternative updating algorithm to address the corresponding optimization problem. Experiments are conducted on benchmark data sets and show promising results of our model compared to several state-of-the-art algorithms.

2 Preliminaries

As mentioned before, NMF is essentially identical to relaxed k -means algorithm [8]. Before introducing our deep k -means, first we briefly review NMF [25]. Denote a nonnegative data matrix as $\mathbf{X} = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{m \times n}$, where n is the number of instances and m is the feature dimension. NMF tries to search two nonnegative matrices $\mathbf{U} \in \mathbb{R}^{m \times c}$ and $\mathbf{V} \in \mathbb{R}^{n \times c}$ such that

$$J_{NMF} = \sum_{i=1}^n \sum_{j=1}^m \left(\mathbf{X}_{ij} - (\mathbf{U}\mathbf{V}^T)_{ij} \right)^2 = \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 \quad (1)$$

s.t. $\mathbf{U} \geq 0, \mathbf{V} \geq 0,$

where $\|\cdot\|_F$ indicates a Frobenius norm and \mathbf{X}_{ij} is the (i, j) -th element of \mathbf{X} . [25] proved that Eq. (1) is not jointly convex in \mathbf{U} and \mathbf{V} (i.e., convex in \mathbf{U} or \mathbf{V} only), and proposed the following alternative updating rules to search the local minimum:

$$\begin{aligned} \mathbf{U}_{ij} &\leftarrow \mathbf{U}_{ij} \frac{(\mathbf{X}\mathbf{V})_{ij}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ij}}, \\ \mathbf{V}_{ij} &\leftarrow \mathbf{V}_{ij} \frac{(\mathbf{X}^T\mathbf{U})_{ij}}{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ij}}, \end{aligned}$$

where \mathbf{V} denotes the class indicator matrix in unsupervised setting [17], \mathbf{U} denotes the centroid matrix, and c is cluster number. Since $c \ll n$ and $c \ll m$, NMF actually tries to obtain a low-dimensional representation \mathbf{V} of the original input \mathbf{X} .

Real-world data sets are rather complex that contain multiple hierarchical modalities (i.e., factors). For instance, face data set typically consists several common modalities like pose, scene, expression, etc. Traditional NMF with single layer formulation is obviously unable to fully uncover the hidden structures of the corresponding factors. Therefore, [35] proposed a multi-layer deep model based on semi-NMF to exploit hierarchical information with respect to different modalities. And models a multi-layer decomposition of an input data \mathbf{X} as

$$\begin{aligned} \mathbf{X} &\approx \mathbf{U}_1 \mathbf{V}_1^T, \\ \mathbf{X} &\approx \mathbf{U}_1 \mathbf{U}_2 \mathbf{V}_2^T, \\ &\vdots \\ \mathbf{X} &\approx \mathbf{U}_1 \mathbf{U}_2 \dots \mathbf{U}_r \mathbf{V}_r^T, \end{aligned} \quad (2)$$

where r means the number of layers, \mathbf{U}_i and \mathbf{V}_i respectively denote the i -th layer basis matrix and representation matrix. It can be seen that deep semi-NMF model also focuses on searching a low-dimensional embedding representation that targets to a similar interpretation at the last layer, i.e., \mathbf{V}_r . The deep model in Eq. (2) is able to automatically search the latent hierarchy by further

factorizing $\mathbf{V}_i (i < r)$. Furthermore, this model is able to discover representations suitable for clustering with respect to different modalities (e.g., for face data set, \mathbf{U}_3 corresponds to the attributes of expressions, $\mathbf{U}_2\mathbf{U}_3$ corresponds to the attributes of poses, and $\mathbf{U} = \mathbf{U}_1\mathbf{U}_2\mathbf{U}_3$ finally corresponds to identities mapping of face images). Compared to traditional single layer NMF, deep semi-NMF provides a better ability to exploit the hidden hierarchical information, as different modalities can be fully identified by the obtained representations of each layer.

3 The Proposed Method

In this section, we introduce a novel deep k -means model for data clustering, followed with its optimization algorithm.

3.1 Deep K -Means

Traditional k -means methods are typically designed in a single layer formulation. Thus the mapping between the obtained low-dimensional representation and the original data may contain complex hierarchical information corresponding to the implicit modalities. To exploit such hidden representations with respect to different modalities, we propose a novel deep k -means model by utilizing the deep structure to conduct k -means hierarchically. The hierarchical semantics of the original data in our model is comprehensively learned in a layerwise way. To improve the robustness of our model, the sparsity-inducing norm, $l_{2,1}$ -norm, is used in the objective. Since $l_{2,1}$ -norm based residue calculation adopts the l_2 -norm within a data point and the l_1 -norm among data points, the influence of outliers is reduced by the l_1 -norm [24]. Moreover, the non-negativity constraint on \mathbf{U}_i is removed such that the input data can consist of mixed signs, thus the applicable range of the proposed model is obviously enlarged. Since the non-negativity constraints on V_i make them more difficult to be optimized, we introduce new variables V_i^+ to which the non-negativity constraints are applied, with the constraints $V_i = V_i^+$. In this paper, we utilize the alternating direction method of multipliers (ADMM) [4] to handle the constraint with an elegant way, while maintain the separability of the objective. As a result, the non-negativity constraints are effectively incorporated to our deep k -means model. Our deep k -means model (DKM) is stated as

$$\begin{aligned}
 & J_{DKM} = \|\mathbf{X} - \mathbf{Y}\|_{2,1} \\
 & \text{s.t. } \mathbf{Y} = \mathbf{U}_1\mathbf{U}_2 \cdots \mathbf{U}_r\mathbf{V}_r^T, (\mathbf{V}_r)_{.c} = \{0, 1\}, \sum_{c=1}^C (\mathbf{V}_r)_{.c} = 1, \\
 & \mathbf{V}_i = \mathbf{V}_i^+, \mathbf{V}_i^+ \geq 0, i \in [1, \dots, r-1].
 \end{aligned} \tag{3}$$

While $\|\mathbf{X} - \mathbf{Y}\|_{2,1}$ is simple to minimize with respect to \mathbf{Y} , $\|\mathbf{X} - \mathbf{U}_1\mathbf{U}_2 \cdots \mathbf{U}_r\mathbf{V}_r^T\|_{2,1}$ is not simple to minimize with respect to \mathbf{U}_i or \mathbf{V}_i . Multiplicative updates implicitly address the problem such that \mathbf{U}_i and \mathbf{V}_i decouple.

In ADMM context, a natural formulation would be to minimize $\|\mathbf{X} - \mathbf{Y}\|_{2,1}$ with the constraint $\mathbf{Y} = \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T$. This is the core reason why we choose to solve such a problem like Eq. (3). In addition, each row of \mathbf{V}_r in Eq. (3) is enforced to satisfy the 1-of- C coding scheme. Its primary goal is to ensure the uniqueness of the final solution \mathbf{V}_r .

The augmented Lagrangian of Eq. (3) is

$$\begin{aligned} \mathcal{L}(\mathbf{Y}, \mathbf{U}_i, \mathbf{V}_i, \mathbf{V}_i^+, \boldsymbol{\mu}, \boldsymbol{\lambda}_i) &= \|\mathbf{X} - \mathbf{Y}\|_{2,1} + \langle \boldsymbol{\mu}, \mathbf{Y} - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T \rangle \\ &+ \frac{\rho}{2} \|\mathbf{Y} - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T\|_F^2 + \sum_{i=1}^{r-1} \langle \boldsymbol{\lambda}_i, \mathbf{V}_i - \mathbf{V}_i^+ \rangle + \frac{\rho}{2} \sum_{i=1}^{r-1} \|\mathbf{V}_i - \mathbf{V}_i^+\|_F^2, \end{aligned} \quad (4)$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}_i$ are Lagrangian multipliers, ρ is the penalty parameter, and $\langle \cdot, \cdot \rangle$ represents the inner product operation.

The alternating direction method for Eq. (4) is derived by minimizing \mathcal{L} with respect to $\mathbf{Y}, \mathbf{U}_i, \mathbf{V}_i, \mathbf{V}_i^+$, one at a time while fixing others, which will be discussed below.

3.2 Optimization

In the following, we propose an alternative updating algorithm to solve the optimization problem of the proposed objective. We update the objective with respect to one variable while fixing the other variables. This procedure repeats until convergence.

Before the minimization, first we perform a pre-training by decomposing the data matrix $\mathbf{X} \approx \mathbf{U}_1 \mathbf{V}_1^T$, where $\mathbf{V}_1 \in \mathbb{R}^{n \times k_1}$ and $\mathbf{U}_1 \in \mathbb{R}^{m \times k_1}$. The obtained representation matrix \mathbf{V}_1 is then further decomposed as $\mathbf{V}_1 \approx \mathbf{U}_2 \mathbf{V}_2^T$, where $\mathbf{V}_2 \in \mathbb{R}^{n \times k_2}$ and $\mathbf{U}_2 \in \mathbb{R}^{k_1 \times k_2}$. We respectively denote k_1 and k_2 as the dimensionalities of the first layer and the second layer¹. Continue to do this, finally all layers are pre-trained, which would greatly improve the training time as well as the effectiveness of our model. This trick has been applied favourably in deep autoencoder networks [4].

Optimizing Eq. (4) is identical to minimizing the formulation as follows

$$\begin{aligned} \mathcal{L}(\mathbf{Y}, \mathbf{U}_i, \mathbf{V}_i, \mathbf{V}_i^+, \boldsymbol{\mu}, \boldsymbol{\lambda}_i) &= \text{Tr} \left((\mathbf{X} - \mathbf{Y}) \mathbf{D} (\mathbf{X} - \mathbf{Y})^T \right) \\ &+ \frac{\rho}{2} \text{Tr} \left((\mathbf{Y} - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T) (\mathbf{Y} - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T)^T \right) \\ &+ \langle \boldsymbol{\mu}, \mathbf{Y} - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T \rangle + \sum_{i=1}^r \langle \boldsymbol{\lambda}_i, \mathbf{V}_i - \mathbf{V}_i^+ \rangle \\ &+ \frac{\rho}{2} \sum_{i=1}^r \text{Tr} \left((\mathbf{V}_i - \mathbf{V}_i^+) (\mathbf{V}_i - \mathbf{V}_i^+)^T \right), \end{aligned} \quad (5)$$

¹ For simplicity, the layer size (dimensionalities) of layer 1 to layer r is denoted as $[k_1 \cdots k_r]$ in the experiments.

where \mathbf{D} is a diagonal matrix and its j -th diagonal element is

$$d_j = \frac{1}{2\|\mathbf{e}_j\|_2}. \quad (6)$$

and \mathbf{e}_j is the j -th column of the following matrix

$$\mathbf{E} = \mathbf{X} - \mathbf{Y}. \quad (7)$$

Updating \mathbf{U}_i Minimizing Eq. (4) w.r.t. \mathbf{U}_i is identical to solving

$$\begin{aligned} \mathcal{L}_U &= \langle \boldsymbol{\mu}, \mathbf{Y} - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T \rangle \\ &+ \frac{\rho}{2} \text{Tr} \left((\mathbf{Y} - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T) (\mathbf{Y} - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T)^T \right). \end{aligned} \quad (8)$$

Calculating the derivative of \mathcal{L}_U w.r.t. \mathbf{U}_i and setting it to zero, then we have

$$\mathbf{U}_i = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \left(\boldsymbol{\Phi}^T \mathbf{Y} \widetilde{\mathbf{V}}_i + \frac{\boldsymbol{\Phi}^T \boldsymbol{\mu} \widetilde{\mathbf{V}}_i}{\rho} \right) (\widetilde{\mathbf{V}}_i^T \widetilde{\mathbf{V}}_i)^{-1}, \quad (9)$$

where $\boldsymbol{\Phi} = \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_{i-1}$ and $\widetilde{\mathbf{V}}_i$ denotes the reconstruction of the i -th layer's centroid matrix.

Updating \mathbf{V}_i ($i < r$) Minimizing Eq. (4) w.r.t. \mathbf{V} is identical to solving

$$\begin{aligned} \mathcal{L}_V &= \langle \boldsymbol{\mu}, \mathbf{Y} - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T \rangle \\ &+ \frac{\rho}{2} \text{Tr} \left((\mathbf{Y} - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T) (\mathbf{Y} - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T)^T \right) \\ &+ \sum_{i=1}^r \langle \boldsymbol{\lambda}_i, \mathbf{V}_i - \mathbf{V}_i^+ \rangle + \frac{\rho}{2} \sum_{i=1}^r \text{Tr} \left((\mathbf{V}_i - \mathbf{V}_i^+) (\mathbf{V}_i - \mathbf{V}_i^+)^T \right). \end{aligned} \quad (10)$$

Similarly, calculating the derivative of \mathcal{L}_V w.r.t. \mathbf{V}_i , and setting it to zero, we obtain

$$\mathbf{V}_i = \left(\mathbf{Y}^T \boldsymbol{\Phi} \mathbf{U}_i + \mathbf{V}_i^+ + \frac{\boldsymbol{\mu}^T \boldsymbol{\Phi} \mathbf{U}_i}{\rho} - \frac{\boldsymbol{\lambda}_i}{\rho} \right) (\mathbf{I} + \mathbf{U}_i^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} \mathbf{U}_i)^{-1}, \quad (11)$$

where \mathbf{I} represents an identity matrix.

Updating \mathbf{V}_r (i.e., \mathbf{V}_i , ($i = r$)) We update \mathbf{V}_r by solving

$$\begin{aligned} J_{V_r} &= \min_{\mathbf{V}_r} \|\mathbf{Y} - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{V}_r^T\|_{2,1} = \min_{\mathbf{v}} \sum_{j=1}^n d_j \|x_j - \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_r \mathbf{v}_j\|_2^2 \\ &\text{s.t. } (\mathbf{V}_r)_{\cdot c} = \{0, 1\}, \sum_{c=1}^C (\mathbf{V}_r)_{\cdot c} = 1, \end{aligned} \quad (12)$$

where x_j denotes the j -th data sample of \mathbf{X} , and \mathbf{v}_j denotes the j -th column of \mathbf{V}_r^T . Taking a closer look at Eq. (12), we can see that it is independent between

different j . Thus we can independently solve it one by one:

$$\begin{aligned} & \min_{\mathbf{v}} (d\|\mathbf{x} - \mathbf{U}_1\mathbf{U}_2 \cdots \mathbf{U}_r\mathbf{v}\|_2^2) \\ & \text{s.t. } v_c = \{0, 1\}, \sum_{c=1}^C v_c = 1. \end{aligned} \tag{13}$$

Since \mathbf{v} is coded by 1-of- C scheme, there exists C candidates that could be the solution of Eq. (13). And each individual solution is exactly the c -th column of identity matrix $\mathbf{I}_C = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_C]$. Thus the optimal solution can be obtained by performing an exhaustive search, i.e.,

$$\mathbf{v}^* = \mathbf{f}_c, \tag{14}$$

where c is given by

$$c = \arg \min_{\bar{c}} (d\|\mathbf{x} - \mathbf{U}_1\mathbf{U}_2 \cdots \mathbf{U}_r\mathbf{f}_{\bar{c}}\|_2^2). \tag{15}$$

Updating \mathbf{Y} Minimizing Eq. (4) w.r.t. \mathbf{Y} is identical to solving

$$\begin{aligned} \mathcal{L}_Y &= \text{Tr} \left((\mathbf{X} - \mathbf{Y}) \mathbf{D} (\mathbf{X} - \mathbf{Y})^T \right) \\ &+ \frac{\rho}{2} \text{Tr} \left((\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2 \cdots \mathbf{U}_r\mathbf{V}_r^T) (\mathbf{Y} - \mathbf{U}_1\mathbf{U}_2 \cdots \mathbf{U}_r\mathbf{V}_r^T)^T \right) \\ &+ \langle \boldsymbol{\mu}, \mathbf{Y} - \mathbf{U}_1\mathbf{U}_2 \cdots \mathbf{U}_r\mathbf{V}_r^T \rangle. \end{aligned} \tag{16}$$

Calculating the derivative of \mathcal{L}_Y w.r.t. \mathbf{Y} and setting it to 0, we have

$$\mathbf{Y} = (2\mathbf{X}\mathbf{D} + \rho\mathbf{U}_1\mathbf{U}_2 \cdots \mathbf{U}_r\mathbf{V}_r^T - \boldsymbol{\mu}) (2\mathbf{D} + \rho\mathbf{I})^{-1}. \tag{17}$$

Updating \mathbf{V}_i^+ Minimizing Eq. (4) w.r.t. \mathbf{V}_i^+ is identical to solving

$$\mathcal{L}_{V^+} = \sum_{i=1}^r \langle \boldsymbol{\lambda}_i, \mathbf{V}_i - \mathbf{V}_i^+ \rangle + \frac{\rho}{2} \sum_{i=1}^r \text{Tr} \left((\mathbf{V}_i - \mathbf{V}_i^+) (\mathbf{V}_i - \mathbf{V}_i^+)^T \right). \tag{18}$$

Calculating the derivative of \mathcal{L}_{V^+} w.r.t. \mathbf{V}_i^+ and setting it to 0, we get

$$\mathbf{V}_i^+ = \mathbf{V}_i + \frac{\boldsymbol{\lambda}_i}{\rho}. \tag{19}$$

In summary, we optimize the proposed model by orderly performing the above steps. This procedure repeats until convergence.



Fig. 1. COIL dataset.

4 Experiments

To validate the effectiveness of our method. We compare it with the classical k -means [26], NMF [25], Orthogonal NMF (ONMF) [8], Semi-NMF (SNMF) [7], $l_{2,1}$ -NMF [24] and the deep Semi-NMF (DeepSNMF) [35].

4.1 Data Sets

We empirically evaluate the proposed method on six benchmark data sets². As a demonstration, Fig. 1 shows the dataset COIL. Table 1 shows the specific characteristics of all datasets. The number of instances is ranged from 400 to 4663, and feature number is ranged from 256 to 7511.

Table 1. Characteristics of experimental data sets.

Data sets	# samples	# features	# classes
ORL32	400	1024	40
COIL	1440	1024	20
Semeion	1593	256	10
Text	1946	7511	2
Cranmed	2431	462	2
Cacmcisi	4663	348	2

² <https://archive.ics.uci.edu/ml/datasets.html>.

4.2 Parameter Setting

For k -means algorithm, we perform k -means on each data set until it convergence. And the corresponding results are treated as the final result of k -means. We also use this result as the initialization of all other compared methods for fairness. For each compared method, the optimal value is solved based on the parameter setting range recommended by the relevant literature, and then the result under this condition is regarded as the final result output. For the proposed deep method, the layer sizes (as described in Sect. 3.2) are set as $[100\ c]$, $[50\ c]$ and $[100\ 50\ c]$ for simplicity. As for parameter ρ , we search it from $\{1e-5, 1e-4, 1e-3, 0.01, 0.1, 1, 10, 100\}$.

Under each parameter setting, we repeat the experiments 10 times for all methods, and the average results are reported for fair comparison.

4.3 Results and Analysis

The clustering performance measured by clustering accuracy (ACC) and normalized mutual information (NMI) of all methods are given in Tables 2-3. It is obvious that our method has better results than other algorithms. The superiority of DKM verifies that it could better explore cluster structure by uncovering the hierarchical semantics of data. That is, by utilizing the deep framework to conduct k -means hierarchically, the hidden structure of data is learned in a layerwise way, and finally, a better high-level, final-layer representation can be obtained for clustering task. By leveraging the deep framework and k -means model, the proposed DKM can enhance the performance of data clustering in general cases.

Table 2. Clustering results of ACC on all data sets.

Datasets	Kmeans	NMF	ONMF	L21NMF	SNMF	DeepSNMF	DKM
ORL32	50.30 \pm 2.2	51.97 \pm 2.8	49.90 \pm 3.1	53.40 \pm 4.1	51.78 \pm 3.5	49.86 \pm 2.0	54.50 \pm 1.2
COIL	59.43 \pm 6.8	62.24 \pm 3.1	58.35 \pm 6.0	63.49 \pm 4.4	63.78 \pm 5.9	66.36 \pm 6.2	68.03 \pm 3.8
Semeion	51.93 \pm 2.8	42.46 \pm 3.5	48.50 \pm 2.8	42.47 \pm 2.0	43.95 \pm 2.8	44.77 \pm 1.2	53.97 \pm 3.3
Text	91.84 \pm 2.1	93.85 \pm 3.9	92.47 \pm 2.9	90.21 \pm 4.0	90.99 \pm 2.0	90.67 \pm 4.5	93.88 \pm 5.7
Cranmed	74.58 \pm 0.1	80.13 \pm 8.8	77.31 \pm 1.2	77.39 \pm 2.5	76.49 \pm 4.9	80.23 \pm 3.1	82.31 \pm 3.9
Cacmcisi	91.99 \pm 0.2	89.75 \pm 5.4	94.96 \pm 0.6	95.37 \pm 0.8	92.22 \pm 0.3	92.80 \pm 3.5	97.12 \pm 7.7

Table 3. Clustering results of NMI of on all data sets.

Datasets	Kmeans	NMF	ONMF	L21NMF	SNMF	DeepSNMF	DKM
ORL32	71.06 \pm 1.3	72.10 \pm 1.3	70.11 \pm 1.7	72.70 \pm 1.8	71.76 \pm 1.9	68.83 \pm 1.3	72.94 \pm 1.5
COIL	74.53 \pm 2.8	73.12 \pm 1.7	72.84 \pm 2.6	74.04 \pm 2.3	74.91 \pm 3.0	77.52 \pm 7.4	78.99 \pm 1.6
Semeion	57.34 \pm 4.7	49.64 \pm 5.6	53.87 \pm 5.8	47.88 \pm 2.7	49.72 \pm 5.4	56.07 \pm 2.0	62.07 \pm 5.9
Text	61.31 \pm 6.5	61.21 \pm 1.5	60.88 \pm 1.9	60.81 \pm 3.0	57.85 \pm 5.2	60.01 \pm 4.2	61.55 \pm 5.4
Cranmed	18.79 \pm 0.3	31.67 \pm 5.6	20.74 \pm 0.3	20.84 \pm 1.2	24.66 \pm 5.4	25.05 \pm 2.4	32.89 \pm 2.3
Cacmcisi	58.47 \pm 0.1	60.01 \pm 2.5	70.42 \pm 0.2	72.05 \pm 0.4	70.52 \pm 0.2	70.07 \pm 3.3	73.06 \pm 2.4

Based on the theoretical analysis and empirical results presented in this paper, it would be interesting to combine deep structure learning and classical machine learning models into a unified framework. By taking advantages of the both learning paradigms, more promising results in various learning tasks can be expected.

5 Conclusion

In this paper, we propose a novel deep k -means model to learn the hidden information with respect to multiple level characteristics. By utilizing the deep structure to conduct k -means hierarchically, the data hierarchical semantics is learned in a layerwise way. Through the deep k -means structure, instances from same class are pushed closer layer by layer, which benefits the subsequent clustering task. We also introduce an alternative updating algorithm to address the corresponding optimization problem. Experiments are conducted on six benchmark data sets and show promising results of our model against several state-of-the-art algorithms.

Acknowledgments. This work was partially supported by the National Key Research and Development Program of China under Contract 2017YFB1002201, the National Natural Science Fund for Distinguished Young Scholar under Grant 61625204, the State Key Program of the National Science Foundation of China under Grant 61836006, and the Fundamental Research Funds for the Central Universities under Grant 1082204112364.

References

1. Ault, S.V., Perez, R.J., Kimble, C.A., Wang, J.: On speech recognition algorithms. *Int. J. Mach. Learn. Comput.* **8**(6) (2018)
2. Badea, L.: Clustering and metaclustering with nonnegative matrix decompositions. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) *ECML 2005. LNCS (LNAI)*, vol. 3720, pp. 10–22. Springer, Heidelberg (2005). https://doi.org/10.1007/11564096_7
3. Bengio, Y.: Learning deep architectures for AI. *Found. Trends® in Mach. Learn.* **2**(1), 1–127 (2009)
4. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends® Mach. Learn.* **3**(1), 1–122 (2011)
5. Buchta, C., Kober, M., Feinerer, I., Hornik, K.: Spherical k -means clustering. *J. Stat. Softw.* **50**(10), 1–22 (2012)
6. Ding, C., He, X.: K -means clustering via principal component analysis. In: *Proceedings of the Twenty-First International Conference on Machine Learning*, pp. 29–37 (2004)
7. Ding, C., Li, T., Jordan, M.I.: Convex and semi-nonnegative matrix factorizations. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(1), 45–55 (2010)
8. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t -factorizations for clustering. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 126–135 (2006)

9. Gokcay, E., Principe, J.C.: Information theoretic clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(2), 158–171 (2002)
10. Gönen, M., Margolin, A.A.: Localized data fusion for kernel k-means clustering with application to cancer biology. In: *Advances in Neural Information Processing Systems*, pp. 1305–1313 (2014)
11. Hou, C., Nie, F., Yi, D., Tao, D.: Discriminative embedded clustering: a framework for grouping high-dimensional data. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(6), 1287–1299 (2015)
12. Huang, S., Kang, Z., Xu, Z.: Self-weighted multi-view clustering with soft capped norm. *Knowl. Based Syst.* **158**, 1–8 (2018)
13. Huang, S., Ren, Y., Xu, Z.: Robust multi-view data clustering with multi-view capped-norm k-means. *Neurocomputing* **311**, 197–208 (2018)
14. Huang, S., Wang, H., Li, T., Li, T., Xu, Z.: Robust graph regularized nonnegative matrix factorization for clustering. *Data Min. Knowl. Disc.* **32**(2), 483–503 (2018)
15. Huang, S., Xu, Z., Kang, Z., Ren, Y.: Regularized nonnegative matrix factorization with adaptive local structure learning. *Neurocomputing* **382**, 196–209 (2020)
16. Huang, S., Xu, Z., Lv, J.: Adaptive local structure learning for document co-clustering. *Knowl.-Based Syst.* **148**, 74–84 (2018)
17. Huang, S., Xu, Z., Wang, F.: Nonnegative matrix factorization with adaptive neighbors. In: *International Joint Conference on Neural Networks*, pp. 486–493 (2017)
18. Huang, S., Zhao, P., Ren, Y., Li, T., Xu, Z.: Self-paced and soft-weighted nonnegative matrix factorization for data representation. *Knowl.-Based Syst.* **164**, 29–37 (2018)
19. Kang, Z., Peng, C., Cheng, Q.: Kernel-driven similarity learning. *Neurocomputing* **267**, 210–219 (2017)
20. Kang, Z., et al.: Multi-graph fusion for multi-view spectral clustering. *Knowl. Based Syst.* **189**, 105102 (2020)
21. Kang, Z., Wen, L., Chen, W., Xu, Z.: Low-rank kernel learning for graph-based clustering. *Knowl.-Based Syst.* **163**, 510–517 (2019)
22. Kang, Z., Xu, H., Wang, B., Zhu, H., Xu, Z.: Clustering with similarity preserving. *Neurocomputing* **365**, 211–218 (2019)
23. Kang, Z., et al.: Partition level multiview subspace clustering. *Neural Netw.* **122**, 279–288 (2020)
24. Kong, D., Ding, C., Huang, H.: Robust nonnegative matrix factorization using l21-norm. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pp. 673–682. ACM (2011)
25. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562 (2001)
26. Macqueen, J.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297 (1967)
27. Newling, J., Fleuret, F.: Fast k-means with accurate bounds. In: *International Conference on Machine Learning*, pp. 936–944 (2016)
28. Newling, J., Fleuret, F.: Nested mini-batch k-means. In: *Advances in Neural Information Processing Systems*, pp. 1352–1360 (2016)
29. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: *Advances in Neural Information Processing Systems*, pp. 849–856 (2002)
30. Ren, Y., Domeniconi, C., Zhang, G., Yu, G.: Weighted-object ensemble clustering: methods and analysis. *Knowl. Inf. Syst.* **51**(2), 661–689 (2017)
31. Ren, Y., Hu, K., Dai, X., Pan, L., Hoi, S.C., Xu, Z.: Semi-supervised deep embedded clustering. *Neurocomputing* **325**, 121–130 (2019)

32. Ren, Y., Huang, S., Zhao, P., Han, M., Xu, Z.: Self-paced and auto-weighted multi-view clustering. *Neurocomputing* **383**, 248–256 (2020)
33. Ren, Y., Kamath, U., Domeniconi, C., Xu, Z.: Parallel boosted clustering. *Neurocomputing* **351**, 87–100 (2019)
34. Ren, Y., Que, X., Yao, D., Xu, Z.: Self-paced multi-task clustering. *Neurocomputing* **350**, 212–220 (2019)
35. Trigeorgis, G., Bousmalis, K., Zafeiriou, S., Schuller, B.W.: A deep matrix factorization method for learning attribute representations. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(3), 417–429 (2017)
36. Tunali, V., Bilgin, T., Camurcu, A.: An improved clustering algorithm for text mining: multi-cluster spherical k-means. *Int. Arab J. Inf. Technol.* **13**(1), 12–19 (2016)
37. Wang, J., et al.: Enhancing multiphoton upconversion through energy clustering at sublattice level. *Nat. Mater.* **13**(2), 157 (2014)
38. Wang, L., Pan, C.: Robust level set image segmentation via a local correntropy-based k-means clustering. *Pattern Recogn.* **47**(5), 1917–1925 (2014)
39. Wu, X., et al.: Top 10 algorithms in data mining. *Knowl. Inf. Syst.* **14**(1), 1–37 (2008)