[Pattern Recognition 117 (2021) 107996](https://doi.org/10.1016/j.patcog.2021.107996)



Contents lists available at [ScienceDirect](http://www.ScienceDirect.com)

Pattern Recognition

journal homepage: [www.elsevier.com/locate/patcog](http://www.elsevier.com/locate/patcog)

|  |  |  |  |
| --- | --- | --- | --- |
| Robust deep *k*-means: An effective and simple method for data clustering[☆](#bookmark2)  Shudong Huang[a,](#bookmark1) Zhao Kang[b,](#bookmark1) Zenglin Xu[c,b,d,](#bookmark1) Quanhui Liu[a,\*](#bookmark1) a *College of Computer Science, Sichuan University, Chengdu 610065, China*  b *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China* c *School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China*  d *Centre for Artiﬁcial Intelligence, Peng Cheng Lab, Shenzhen 518055, China* | | | [check for](http://crossmark.crossref.org/dialog/?doi=10.1016/j.patcog.2021.107996&domain=pdf)  [updates](http://crossmark.crossref.org/dialog/?doi=10.1016/j.patcog.2021.107996&domain=pdf) |
| a r t i c l e i n f o |  | a b s t r a c t | |
| *Article history:*  Received 13 October 2020 Revised 10 March 2021  Accepted 18 April 2021  Available online 28 April 2021 | Clustering aims to partition an input dataset into distinct groups according to some distance or similar- ity measurements. One of the most widely used clustering method nowadays is the *k*-means algorithm because of its simplicity and eﬃciency. In the last few decades, *k*-means and its various extensions have been formulated to solve the practical clustering problems. However, existing clustering methods are of- ten presented in a single-layer formulation (i.e., shallow formulation). As a result, the mapping between the obtained low-level representation and the original input data may contain rather complex hierarchi- cal information. To overcome the drawbacks of low-level features, deep learning techniques are adopted to extract deep representations and improve the clustering performance. In this paper, we propose a ro- bust deep *k*-means model to learn the hidden representations associate with different implicit lower-level attributes. By using the deep structure to hierarchically perform *k*-means, the hierarchical semantics of data can be exploited in a layerwise way. Data samples from the same class are forced to be closer layer by layer, which is beneﬁcial for clustering task. The objective function of our model is derived to a more trackable form such that the optimization problem can be tackled more easily and the ﬁnal robust re- sults can be obtained. Experimental results over 12 benchmark data sets substantiate that the proposed model achieves a breakthrough in clustering performance, compared with both classical and state-of-the- art methods.  © 2021 Elsevier Ltd. All rights reserved. | |
| *Keywords:*  *k*-means algorithm Robust clustering Deep learning |

**1. Introduction**

Clustering, the goal of which focuses on dividing a dataset into homogeneous groups, is no doubt one of the most funda- mental techniques in statistic and machine learning [[1,2].](#bookmark4) Cus- tering has been found to conduct surprisingly well, especially in unsupervised scenarios [[3].](#bookmark5) Myriads of applications can be for- mulated as a clustering problem, text mining [[4],](#bookmark6) voice recogni- tion [[5],](#bookmark7) image segmentation [[6],](#bookmark8) to name a few. In recent years, a number of clustering methods have been investigated based on different methodologies and statistical theories [[7],](#bookmark9) such as *k*-means clustering [[8],](#bookmark10) spectral clustering [[9],](#bookmark11) nonnegative ma- trix factorization-based clustering [[10],](#bookmark12) information theoretic clus- tering [[11,12],](#bookmark13) multi-view clustering [[13,14],](#bookmark14) etc.. Among them, *k*-

☆ This paper is an extended version of [[1],](#bookmark4) which has been accepted for presen- tation at the International Conference on Neural Computing for Advanced Applica- tions (NCAA-2020).

\* Corresponding author.

*E-mail address:* [quanhuiliu@scu.edu.cn](mailto:quanhuiliu@scu.edu.cn) (Q. Liu).

means successfully attracted extensive attention because of its effectiveness and simplicity since it was presented in 1967 [[8].](#bookmark10) What’s more, it has been widely recognized as one of the top ten data mining algorithms, for contributions to the usage and cluster- ing performance in various practical problems [[15].](#bookmark16)

Recent development on machine learning has illustrated that one can process explosively increase of data more effectively with deep learning techniques, especially in unsupervised settings. For example, the deep neural networks have been commonly used in clustering tasks [[16,17].](#bookmark17) Ji et al. [[16]](#bookmark17) presented a deep neu- ral network architecture for subspace clustering by introducing a self-expressive layer between the encoder and the decoder. Zhou et al. [[17]](#bookmark18) further extended this architecture to a deep adversar- ial subspace clustering model by utilizing the adversarial learning. That is, a subspace clustering generator is used to learn the sam- ple representations, while a quality-verifying discriminator is used to evaluate current clustering performance by estimating whether the re-sampled data have consistent properties. Guo et al. [[18]](#bookmark19) pro- posed to jointly optimize cluster labels assignment and learn fea-

<https://doi.org/10.1016/j.patcog.2021.107996>

0031-3203/© 2021 Elsevier Ltd. All rights reserved.

*S. Huang, Z. Kang, Z. Xu et al. Pattern Recognition 117 (2021) 107996*

tures by making use of local structure preservation and applying the under-complete autoencoder. Dizaji et al. [[19]](#bookmark20) deﬁned a clus- tering model using KL divergence minimization, which can map the raw data into a discriminative subspace and predict cluster assignments. Peng et al. [[20]](#bookmark21) introduced a structured autoencoder for subspace clustering, where both the local and global subspace structure can be preserved by minimizing reconstruction error and incorporating a prior structured information. Peng et al. [[21]](#bookmark22) dis- covered a common invariance based on the assumption that dif- ferent distance metrics would result in similar clustering assign- ments on the manifold. Based on such a common invariance, a deep clustering method is designed by minimizing the discrep- ancy between pairwise sample assignments for each data point. Zhang et al. [[22]](#bookmark23) proposed an end-to-end self-supervised convolu- tional clustering network, which can accomplish the convolutional network module, self-expression module, and spectral clustering module into a joint optimization framework. However, the train- ing process of these methods are typically time-consuming and un- stable. Besides, there are myriads of parameters need to be tuned which makes them impractical for unsupervised tasks. More im- portant, these methods require a vast amount of data to train, which in turn needs extensive computational power. Combining deep learning architecture and classical clustering models into a uniﬁed framework provides a better potential solution for cluster- ing tasks.

In the past decades, numerous variants of classical *k*-means al- gorithm have been studied to boost the clustering performance. Ding and He [[23]](#bookmark24) proved that principal components essentially af- ford the continuous solutions, which can be seen as discrete in- dicators for *k*-means clustering. Buchta et al. [[24]](#bookmark25) employed co- sine similarities to conduct prototype-based partitioning, on which a spherical *k*-means algorithm was proposed for text cluster- ing. Khanmohammadi et al. [[25]](#bookmark26) tried to overcome the model sen- sitivity to initial cluster centroids by combining overlapping *k*- means and *k*-harmonic means algorithms. It uses the output of *k*- harmonic means method to initialize the cluster centers of over- lapping *k*-means method. By locating the seed points at dense areas of the dataset, Kumar and Reddy [[26]](#bookmark27) improved the per- formance of *k*-means ﬁltering method and separated the data points well. The dense areas, unlike other methods, are identi- ﬁed by representing the data points in a *kd*-tree. Considering ex- isting methods lack statistical guarantees when many features are irrelevant, Chakraborty et al. [[27]](#bookmark29) addressed the problem by ap- plying entropy regularization to learn feature relevance while an- nealing. The convergence and consistency of the model was guar- anteed, and a scalable majorization-minimization algorithm was proposed to optimize the model. This model yields signiﬁcant improvements over *k*-means, yet retains the same computational complexity. Capó et al. [[28]](#bookmark30) intended to solve the bottleneck of massive data by introducing an eﬃcient approximation to the *k*- means problem. This method partitions the dataset into a number of subsets, each of which is generally characterized by its repre- sentative and weight. The *k*-means algorithm is then performed on such local representation, which reduces the number of computed distances.

Despite the remarkable progress made by the aforementioned *k*-means approaches, these methods are conventionally devised in a single-layer formulation. Thus the mapping between the ob- tained low-dimensional representation and the original input data may contain rather complex hierarchical information. Considering the the development of deep learning that forces on adopting mul- tiple processing layers to extract the hierarchical information of data [[29],](#bookmark32) in this paper, a novel robust deep *k*-means model is pro- posed to exploit the hierarchical information of multiple level at- tributes. The overall framework of our model is shown in [Fig. 1.](#bookmark33) As we can see, by using the deep structure to hierarchically perform

*k*-means, the hierarchical semantics of data can be exploited in a layerwise way. That is, the data samples from the same class are gathered closer layer by layer, which is greatly beneﬁcial for the clustering task.

The main contributions of this work are three aspects:

• A novel robust deep model is proposed to perform *k*-means hi- erarchically, thus the hierarchical semantics of data can be ex- plored in a layerwise way. As a result, data samples from the same class are effectively gathered closer layer by layer, which provides a clear clustering structure.

• To solve the optimization problem of our model, the corre- sponding objective function is derived to a more trackable form and an alternative updating algorithm is presented to solve the optimization problem.

• Experiments over 12 benchmark data sets are conducted and show promising results, compared to both classical and state- of-the-art methods.

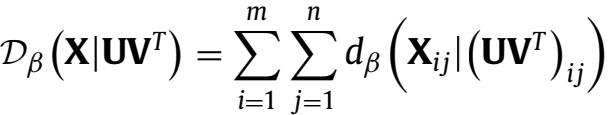
The groundwork of this paper is conceived as follows. We brief introduce the closely related works in [Section 2.](#bookmark28) The details of our model are given in [Section 3.](#bookmark34) Experimental results are described in [Section 4.](#bookmark35) Finally, we give a conclusion of this paper in [Section 5.](#bookmark36) The differences between this work and our earlier paper [[1]](#bookmark4) are threefold. First, we propose a general form of our robust deep *k*- kmeans model. In detail, we deploy a series of divergence func- tions to measure the reconstruction errors [(Section 3](#bookmark34)), instead of merely the Frobenius norm which is sensitive to noisy data and outliers. Hence our earlier work [[1]](#bookmark4) is simply a special case of this paper. Second, more comprehensive experiments over 12 benchmark datasets are conducted, substantiating the robustness and effectiveness of our model: (i) the clustering results on more datasets and advanced baselines are recorded [(Section 4.3](#bookmark37)); (ii) adding the experiment results with respect to different parameter settings [(Section 4.4](#bookmark38)); (iii) showcasing the experiments of conver- gence analysis [(Section 4.5](#bookmark39)); (iv) investigating the inﬂuence of dif- ferent divergence functions to clustering performance [(Section 4.6)](#bookmark40).

Third, we have introduced more closely related literatures (in [Sections 1](#bookmark3) and [2](#bookmark28)), clarifying the connections and differences with the state-of-the-arts. This helps better position the proposed work in the community.

**2. Preliminaries**

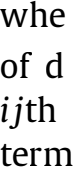
Nonnegative Matrix Factorization (NMF) has received much at- tention in data clustering due to its intuitive parts-based interpre- tation [[30,31].](#bookmark41) Previous studies have shown that NMF is essentially equal to *k*-means with a relaxed condition [[30].](#bookmark41) Here we start with an introduction to NMF [[32].](#bookmark42) Suppose **X** = [**x**1, **x**2, . . . , **x***n*] ∈ R*m* ×*n* is a nonnegative data matrix with *n* data samples and *m* features. NMF aims to ﬁnd two nonnegative matrices **U** ∈ R*m* ×*c* and **V** ∈ R*n* ×*c*

such that **X** ≈ **UV***T* , and the general form of NMF is



(1)

s.t.**U** ≥ 0, **V** ≥ 0,

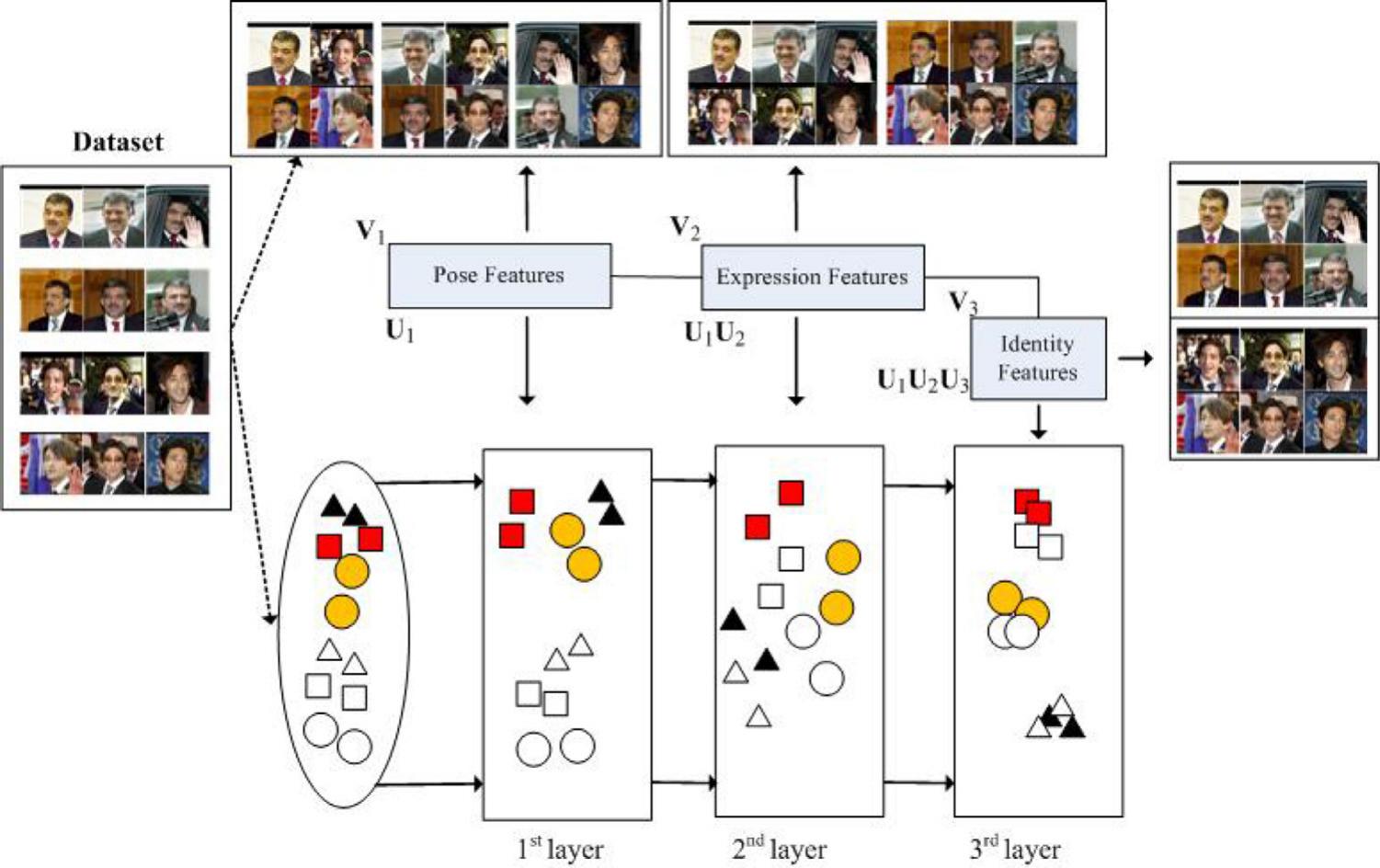
ee)[e[ ]aala,rs](#bookmark43)..,) *i*nei-

gence functions most widely used in NMF:

∗ β = 2 (Euclidean Distance): *d*2 (*a*|*b*) = (*a* − *b*)2

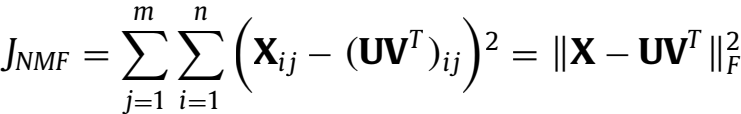
∗ β = 1 (Kullback–Leibler Divergence): *d*1 (*a*|*b*) = *a*log  − *a* + *b* ∗ β = 0 (Itakura–Saito Divergence): *d*0 (*a*|*b*) =  − log  − 1

*S. Huang, Z. Kang, Z. Xu et al. Pattern Recognition 117 (2021) 107996*



**Fig. 1.** A three-layer deep *k*-means structure is used to depict the framework of our model. Same shape indicates the data samples belong to the same class. It is clear the variability in face data can be distinguished based on the attributes such as the facial expression (with or without a smile) or the pose (eyes left, right or front) of the subject. The proposed deep model focuses on exploiting the hierarchical information layer by layer. As a result, a more discriminative representation can be expected.

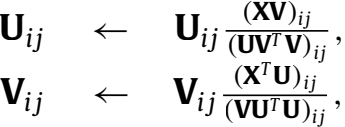
[[32]](#bookmark42) adopted the Euclidean distance in [Eq. (1),](#bookmark31) i.e.,



(2)

s.t.**U** ≥ 0, **V** ≥ 0,

where Ⅱ · Ⅱ*F* means the Frobenius norm. [[32]](#bookmark42) further pointed out that [Eq. (2)](#bookmark44) is a bi-convex formulation (convex in **U** or **V** only), and searched the local minimum by applying the updating rules as follows:



where **V** can be seen as the cluster indicator matrix [[30],](#bookmark41) **U** rep- resents the centroid matrix, and *c* denotes the number of clusters. Usually, we have *c* 冬 *n* and *c* 冬 *m*, which means [Eq. (2)](#bookmark44) actually searches for a low-dimensional representation **V** of **X**.

But in reality, datasets are often complex and always con- tain multiple hierarchical modalities (i.e., factors). Taking the face dataset as an example, it typically consists of some common modalities, such as expression, pose, scene, and so on. Thus it is obvious that single-layer-based NMF cannot fully exploit the hidden information with respect to different factors. To ﬁll this gap, [[34]](#bookmark45) studied a multi-layer deep model that innovatively ex- plores the hierarchical information of data by conducting the semi- NMF hierarchically. And the basic formulation of the deep semi-

NMF model is deﬁned as

**X** ≈ **U**1**V** ,

**X** ≈ **U**1**U**2**V** ,

.

(3)

.

.

**X** ≈ **U**1**U**2 . . . **U***r***V**,

where *r* is the layer number, **U***i* and **V***i* are basis matrix and rep- resentation matrix of the *i*th layer, respectively. It is clear that deep semi-NMF also targets to search a low-dimensional embed- ding representation, i.e., the last layer **V***r*. By hierarchically decom- posing each layer **V***i* (*i* < *r*), [Eq. (3)](#bookmark46) is able to discover the latent hierarchy. Compared with existing single-layer NMF models, deep

semi-NMF allows for a better ability to uncover the hierarchical in- formation of data, as different modalities can be can be recognized by the low-dimensional representations of different layers. Hence our model can fully achieve representations suitable for the sub- sequent clustering in terms of different modalities. For instance, as shown in [Fig. 1,](#bookmark33) **U**3 corresponds to the characteristics of expres- sions, **U**2**U**3 corresponds to the characteristics of poses, and ﬁnally, **U** = **U**1**U**2**U**3 corresponds to identities mapping of face images. In this way, a better high-identiﬁability, ﬁnal-layer representation for clustering according to the characteristic with the lowest variabil- ity can be obtained.

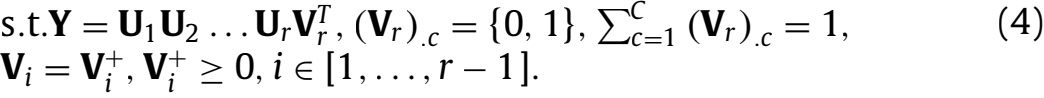
**3. The proposed model**

In this section, we present a novel deep *k*-means model termed robust deep *k*-means (RDKM). We introduce an eﬃcient updating algorithm to solve the corresponding optimization problem. The convergence of the proposed algorithm is also analysed.

*3.1. Robust deep k-means*

To explore the low-dimensional representations with respect to different modalities, a novel robust deep *k*-means model is inves- tigated by utilizing the deep structure to conduct *k*-means hierar- chically. In this paper, to enlarge the applicable range of our model (i.e., deal with both the negative and nonnegative data), the non- negative constraint on **U***i* is omitted. Considering that the nonneg- ativity constraints on *Vi* make the optimization problem diﬃcult to solve, we transform the objective function into a more track- able form by introduce new variables *Vi*+. In this way, the nonneg- ativity constraints are separated and equivalently adopted, with the constraints *Vi* = *Vi*+. Hence we not only extend the application, but also preserve the strong interpretability of our model. Here we use alternating direction method of multipliers (ADMM) [[35]](#bookmark48) to solve the optimization problem. Mathematically, the proposed RDMK

model is formulated as *J* = Dβ(**X**|**Y**)



*S. Huang, Z. Kang, Z. Xu et al. Pattern Recognition 117 (2021) 107996*

In [Eq. (4),](#bookmark47) we can see that the 1-of-*C* coding scheme is employed on each row of **V***r*. The primary goal of the 1-of-*C* coding scheme is to guarantee the uniqueness of **V***r*. Moreover, based on **V***r* , we can obtain the ﬁnal discrete partition result directly without any postprocessing.

Similar to [Eq. (2),](#bookmark44) if the Euclidean distance (i.e., β = 2) is

adopted in [Eq. (4),](#bookmark47) we have *J* = Ⅱ**X** − **Y**Ⅱ

s.t.**Y** = **U**1**U**2 . . . **U***r***V**, (**V***r* ).*c* = {0, 1}, Σ=1 (**V***r* ).*c* = 1, (5)

**V***i* = **V**, **V**≥ 0, *i* ∈ [1, . . . , *r* − 1].

However, it has been proven that Frobenius norm is sensitive to noisy data and outliers [[36,37].](#bookmark50) To enhance the robustness of the proposed model, the sparsity-inducing norm (i.e., *l*2,1 -norm), is employed in our model. According to [[36],](#bookmark50) *l*2,1 -norm is able to re- duce the inﬂuence of outliers as it performs the *l*2 -norm within a data point and the *l*1 -norm among data points. Finally, our robust deep *k*-means (RDKM) model is written as

*JRDKM* = Ⅱ**X** − **Y**Ⅱ2,1

s.t.**Y** = **U**1**U**2 . . . **U***r***V**, (**V***r* ).*c* = {0, 1}, Σ=1 (**V***r* ).*c* = 1, (6)

**V***i* = **V**, **V**≥ 0, *i* ∈ [1, . . . , *r* − 1].

As we can see, Ⅱ**X** − **Y**Ⅱ2,1 is simple to minimize with respect to **Y**,

while Ⅱ**X** − **U**1**U**2 . . . **U***r***V**Ⅱ2,1 is not that simple to minimize with

respect to **U***i* or **V***i*. Multiplicative updating rules implicitly address the problem such that **U***i* and **V***i* decouple. In ADMM context, a natural formulation is to optimize Ⅱ**X** − **Y**Ⅱ2,1 with the constraint

**Y** = **U**1**U**2 . . . **U***r***V**. **That’s the reason why we consider solving the**

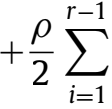
**problem like** [Eq. (6).](#bookmark51)

To verify the robustness and effectivenss of *l*2,1 -norm used in our model, the cases of different divergence functions (i.e., β = 2, β = 1, and β = 0) will be discussed in a later section. The opti- mization algorithms with respect to different divergence functions will also be described in [Appendix A.](#bookmark52)

For [Eq. (6),](#bookmark51) we introduce an effective optimization algorithm based on ADMM [[35].](#bookmark48) The Lagrangian function of [Eq. (6)](#bookmark51) is

L(**Y**, **U***i*,**V***i* , **V**, **μ** , **λ***i* ) = Ⅱ**X** − **Y**Ⅱ2,1 + (**μ** , **Y** − **U**1**U**2 . . . **U***r***V**〉

+  Ⅱ**Y** − **U**1**U**2 . . . **U***r***V**Ⅱ+Σ (**λ***i*, **V***i* − **V**〉

 Ⅱ**V***i* − **V**Ⅱ (7)

where ρ denotes a penalty parameter, **μ** and **λ***i* both represent the Lagrangian multipliers, and (· , ·〉means inner product operation.

The alternating algorithm for [Eq. (7)](#bookmark53) is derived by minimizing

L with respect to **Y**, **U***i*, **V***i* , **V**, one at a time while ﬁxing others,

i.e., the solution can be drawn by repeating the steps as follows,

**U**+1 = arg min**U***i* L (**Y***t* , **U***i* , **V**, (**V**)*t* , **μ***t* , **λ**),

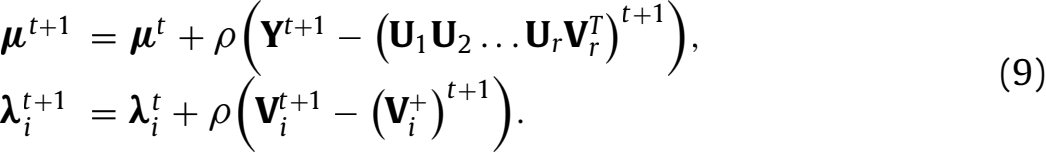
**V**+1 = arg min**V***i* L (**Y***t* , **U**+1 , **V***i*, (**V**)*t* , **μ***t* , **λ**),

(8)

**Y***t*+1 = arg min**Y** L (**Y**, **U**+1 , **V**+1, (**V**)*t* , **μ***t* , **λ**),

(**V**)*t*+1 = arg min**V**L (**Y***t*+1 , **U**+1 , **V**+1 , **V**, **μ***t* , **λ**),

and then updating the multipliers **μ** and **λ***i* with the following for- mulas [[35] :](#bookmark48)



It is noteworthy that our model is different from the deep semi- NMF model [[34].](#bookmark45) For one thing, deep semi-NMF is based on Frobe- nius norm, which is well-known to be sensitive to noisy data and outliers. For another, the target data representation in deep semi- NMF cannot directly assign the discrete clustering result, therefore

requires a postprocessing to allocate the class label to each data sample. Thus it is diﬃcult to achieve a stable result. Moreover, the optimization algorithm of our model is totally different from [[34],](#bookmark45) which will be introduced below.

*3.2. Optimization*

In this paper, an iterative updating algorithm is intro- duced to solve the optimization problem. Speciﬁcally, we update [Eq. (7)](#bookmark53) with respect to each distinct variable while keeping the other variables ﬁxed.

Before the optimization, we adopt a pre-training by decompos-

ing the input data matrix **X** ≈ **U**1**V** , where **V**1 ∈ R*n* ×*c*1 and **U**1 ∈

R*m* ×*c*1 . The obtained representation matrix **V**1 is then further de-

composed as **V**1 ≈ **U**2**V** , where **V**2 ∈ R*n* ×*c*2 and **U**2 ∈ R*c*1 ×*c*2 . We

respectively denote *c*1 and *c*2 as the dimensionalities of the ﬁrst layer and the second layer[.1](#bookmark54) Continue to do so, all layers will be pre-trained, which would greatly boost the effectiveness and re- duce the training time of our model. This trick has been success- fully employed in deep autoencoder networks [[35].](#bookmark48)

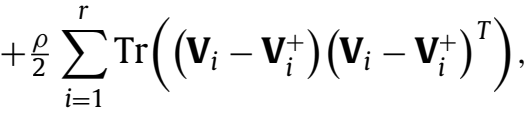
With simple algebra, [Eq. (7)](#bookmark53) can be rewritten as

L(**Y**, **U***i*,**V***i* , **V**, **μ** , **λ***i* ) = Tr ((**X** − **Y**)**D**(**X** − **Y**)*T* )

+ Tr ( (**Y** − **U**1**U**2 . . . **U***r***V**)(**Y** − **U**1**U**2 . . . **U***r***V**)*T* )

+(**μ** , **Y** − **U**1**U**2 . . . **U***r***V**〉+ Σ*i*1 (**λ***i*, **V***i* − **V**〉

(10)



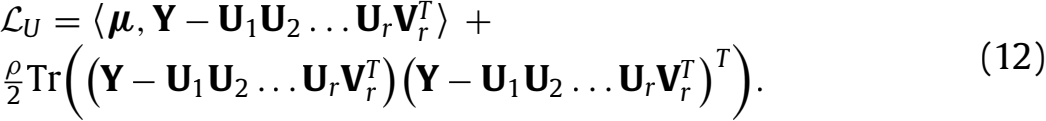
where **D** denotes a diagonal matrix with the *j*th diagonal element being

 (11)

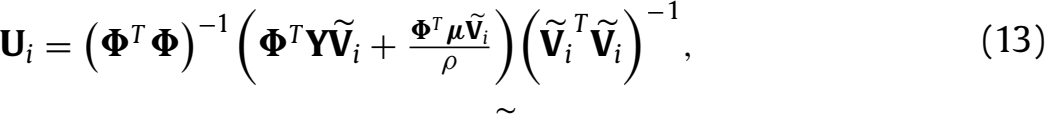
and **e***j* is the *j*th column of **E** = **X** − **Y**.

*3.2.1. Updating* **U***i*

The optimization problem w.r.t. **U***i* is



Setting  = 0, we have

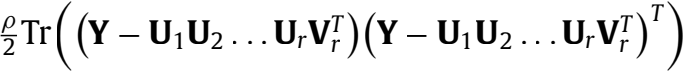


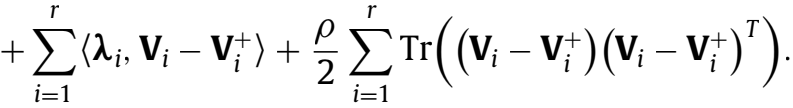
where **Φ** = **U**1**U**2... **U***i*−1 , and **V***i* is the reconstruction of the *i*th layer’s centroid matrix.

*3.2.2. Updating* **V***i* (*i* < *r*)

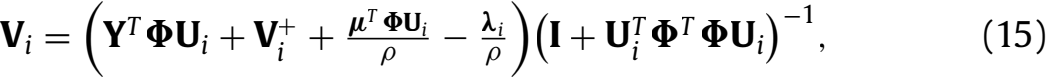
The optimization problem w.r.t. **V** is

L*V* = (**μ** , **Y** − **U**1**U**2 . . . **U***r***V**〉+

 (14)



Similarly, setting  = 0, we obtain



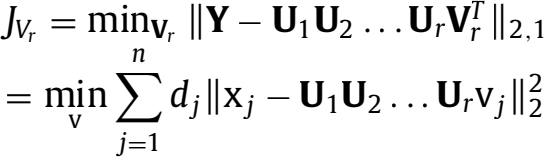
where **I** denotes an identity matrix.

1 For simplicity, the layer size (dimensionalities) of layer 1 to layer *r* is written as [*c*1, . . . , *cr* ] in this paper.

*S. Huang, Z. Kang, Z. Xu et al.*

*3.2.3. Updating* **V***r (i.e.,* **V***i* , (*i* = *r*)*)*

The optimization problem w.r.t **V***r* can be stated as



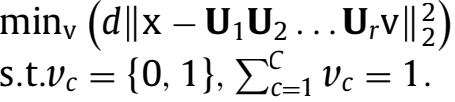
(16)

s.t.(**V***r* ).*c* = {0, 1}, Σ=1 (**V***r* ).*c* = 1,

where x*j* is the *j*th data point of **X**, and v*j* denotes the *j*th col-

umn of **V**.It is obvious that [Eq. (16)](#bookmark56) is independent for each *j*.

Accordingly, for a speciﬁc *j*, it can be independently solved by

 (17)

Since v satisﬁes the 1-of-*C* coding scheme, there are obvious *C* possible solutions for [Eq. (17).](#bookmark57) We can ﬁnd that each individ- ual solution is exactly the *c*th column of the identity matrix **I***C* = [**f**1, **f**2, . . . , **f***C* ]. Hence we can obtain the optimal solution, v∗ , by conducting an exhaustive search, i.e.,

v∗ = **f**c , (18)

where

*c* = arg min (*d*Ⅱx − **U**1**U**2... **U***r***f***c* Ⅱ). (19)

*c*

*3.2.4. Updating* **Y**

The optimization problem w.r.t. **Y** is L*Y* = Tr ((**X** − **Y**)**D**(**X** − **Y**)*T* )+

Tr ((**Y** − **U**1**U**2 . . . **U***r***V**)(**Y** − **U**1**U**2 . . . **U***r***V**)*T* )

(20)

+(**μ** , **Y** − **U**1**U**2 . . . **U***r***V**〉.

Setting  = 0, we get

**Y** = (2**XD** + ρ**U**1**U**2 . . . **U***r***V**− **μ**)(2**D** + ρ**I**)−1 .

(21)

[Eq. (21)](#bookmark58) gives us an update rule for **Y** when *l*2,1 -norm is adopted to measure the reconstruction error. One may ask what if adopting other divergence functions. Here we give the updating rules for **Y** with different divergence functions. Thus we need to

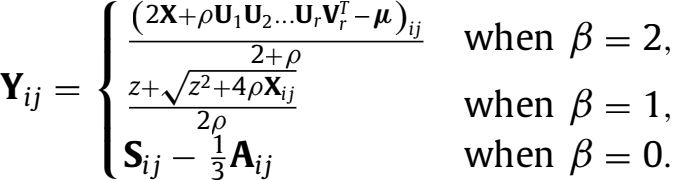
solve the general optimization problem as follows

L*Y*β = Dβ(**X**|**Y**) + (**μ** , **Y** − **U**1**U**2 . . . **U***r***V**〉+

Tr ((**Y** − **U**1**U**2 . . . **U***r***V**)(**Y** − **U**1**U**2 . . . **U***r***V**)*T* ).

(22)

According to [Eq. (22),](#bookmark59) the updating rules for **Y** are given by

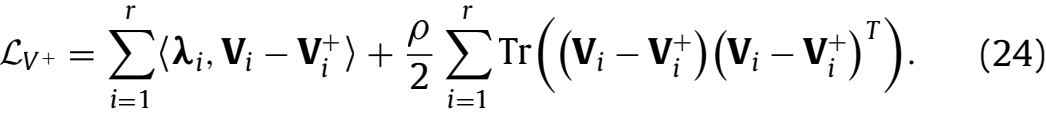


(23)

The deﬁnition of *z*, **S**, **A** as well as the speciﬁc details about [Eq. (23)](#bookmark60) are given in [Appendix A.](#bookmark52)

*3.2.5. Updating* **V**

The optimization problem w.r.t. **V**is



Similarly, setting  = 0, we obtain

 = **V***i* +  (25)

For clarity, the proposed algorithm is summarized step by step in [Algorithm 1.](#bookmark55)

*Pattern Recognition 117 (2021) 107996*

|  |
| --- |
| **Algorithm 1** Robust deep *k*-means for data clustering (RDKM). |
| **Input:** Input data matrix **X**, layer size *p*; **Output: U***i* and **V***i* for each of the layers;  **Iteration:**  **for** all layers **do**  **U***i* , **V***i* ← *k*-means (**V***i*−1,layers(*i*)) **end for**  Initialize **D** as deﬁned in Eq.~(11). **repeat**  **for** all layers **do**    2. Compute **Φ** = Π **U***j*.  3. Update **U***i* according to Eq.~(13).  4. Update **V***i* according to Eq.~(15).  5. Update **Y** according to Eq.~(21) or Eq.~(23).  6. Update **V**according to Eq.~(25).  7. Compute **D** according to Eq.~(11).  8. Update **μ** ← **μ** + ρ (**Y** − **U**1**U**2...**U***r***V**).  9. Update **λ***i* ← **λ***i*+ ρ (**V***i* − **V**).  **end for**  **until** Converges |

*3.3. Convergence analysis*

We prove the convergence of [Algorithm 1](#bookmark55) as follows: the objec- tive function in [Eq. (6)](#bookmark51) can be divided into four subproblems and each one is a convex problem with respect to the corresponding variable. By iteratively solving the subproblems, it can be guaran- teed that we can search the optimal solution to each subproblem. Finally, [Algorithm 1](#bookmark55) will converge to a local minima.

For different divergence functions, the main difference of the corresponding optimization algorithms is the updating rules for **Y**. As we introduced in [Appendix A,](#bookmark52) the optimization subproblems for **Y** with respect to different divergences are all convex prob- lems, and we can obtain the corresponding closed-form solutions as shown in [Eqs. (A.2),](#bookmark61) [(A.5)](#bookmark62) and [(A.14).](#bookmark63) Thus the optimization al- gorithms with respect to different divergences will also converge to a local minima.

**4. Experiments**

In this paper, we experimentally evaluate the effectiveness of our method. We compare the proposed RDKM on 12 bench- mark datasets against six baselines: the standard *k*-means [[8],](#bookmark10) NMF [[32],](#bookmark42) Orthogonal NMF (ONMF) [[30],](#bookmark41) Semi-NMF (SNMF) [[38],](#bookmark64) *l*2,1 -NMF [[36]](#bookmark50) and the deep Semi-NMF (DeepSNMF) [[34].](#bookmark45)

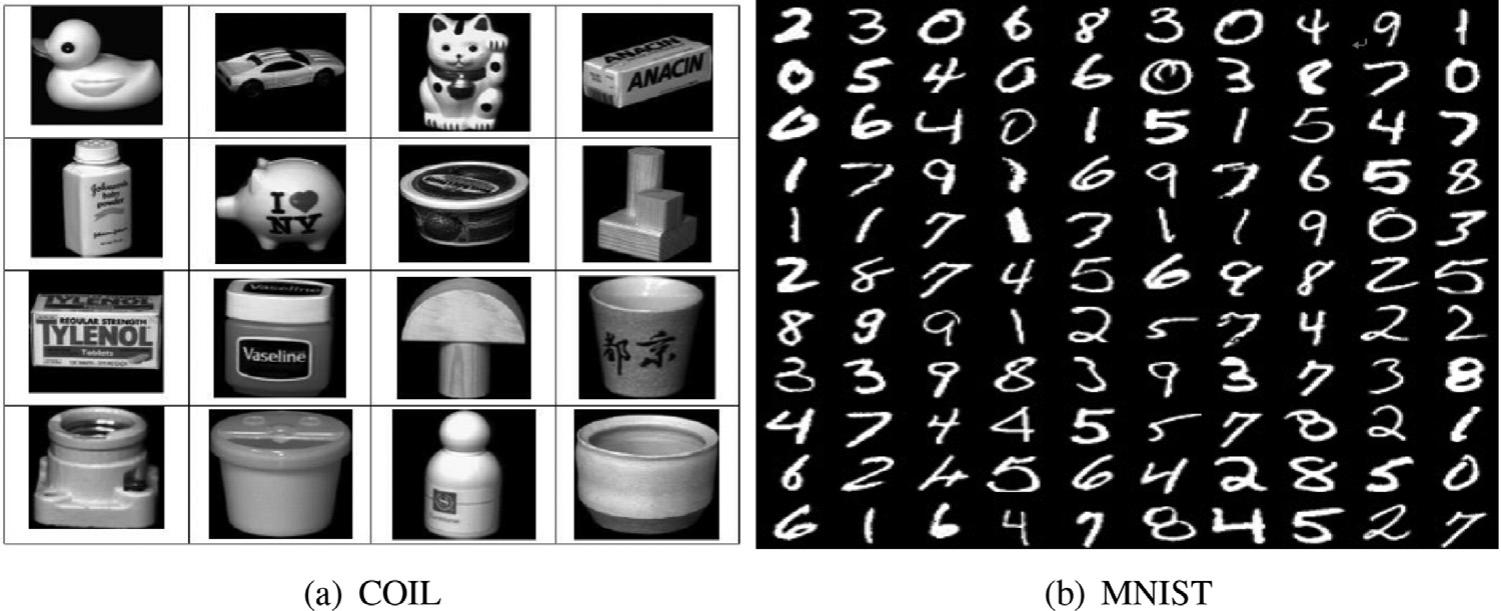
*4.1. Data sets*

In our experiment, we adopt 12 benchmark datasets including two gene expression datasets, four textual datasets, and six im- age datasets. As an illustration, [Fig. 2](#bookmark65) shows the sample images of datasets COIL and MNIST. [Table 1](#bookmark66) summarizes the speciﬁc de- tails of all datasets, from which we can see the instance number is ranged from 102 to 7094, and the number of features is from 256 to 7511, covering a broad range of properties.

*4.2. Parameter setting*

For *k*-means algorithm, on all datasets, we conduct *k*-means un- til it convergence. For a fair comparison, the results of *k*-means are also used as the initialization of other compared methods. For the compared methods, we set the parameters just as reported in each

*S. Huang, Z. Kang, Z. Xu et al. Pattern Recognition 117 (2021) 107996*



**Fig. 2.** Sample images of (a) COIL and (b) MNIST.

**Table 1**

The details of our experimental data sets.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Data sets | # samples | # features | # classes | # type |
| 1 | PROSTATEML | 102 | 5966 | 2 | gene |
| 2 | Yale32 | 165 | 1024 | 15 | image |
| 3 | Lungml | 203 | 3312 | 5 | gene |
| 4 | ORL32 | 400 | 1024 | 40 | image |
| 5 | COIL | 1440 | 1024 | 20 | image |
| 6 | Semeion | 1593 | 256 | 10 | image |
| 7 | MSRA | 1799 | 256 | 12 | image |
| 8 | Text | 1946 | 7511 | 2 | text |
| 9 | Cranmed | 2431 | 462 | 2 | text |
| 10 | MNIST05 | 3495 | 784 | 10 | image |
| 11 | Cacmcisi | 4663 | 348 | 2 | text |
| 12 | Classic | 7094 | 462 | 4 | text |

paper. If there are no suggested values, we searched the param- eters exhaustively, and used the ones producing the best perfor- mance. For our RDKM, the layer sizes (as described in [3.2)](#bookmark49) are set as [50 *C*], [100 *C*] and [100 50 *C*] according to [[14,39].](#bookmark67) As for the parameter ρ , we search it from {1e5, 1e4, 1e3, 1e2, 0.1, 1, 10, 100}.

To reduce the inﬂuence of the initialization, we repeat the ex- periments 20 times, and reported the average performance over the 20 repetitions.

*4.3. Results and analysis*

[Table 2](#bookmark68) displays the clustering performance in terms of clus- tering accuracy (ACC), normalized mutual information (NMI) and purity of all methods over 12 datasets. It can be seen that the pro- posed method outperforms other algorithms in most cases. In de- tail, for ACC, our model achieves the best results 11 times among the 12 datasets. For NMI, our model achieves the best results 10 times. For purity, the number is also 10. In a word, the cluster- ing performance is suﬃcient to validate the effectiveness of the proposed model. The superiority of RDKM demonstrates that it is beneﬁcial to discover a better cluster structure by exploring the hierarchical semantics of data. The reason is that, by applying the deep framework to perform *k*-means hierarchically, the hierarchi- cal information of data can be exploited in a layerwise way, and ﬁnally, a better high-identiﬁability, ﬁnal-layer representation is ob- tained for clustering task. By skillfully combining the deep frame- work and *k*-means model, our model is able to boost the clustering performance in general cases.

According to the theoretical analysis and empirical results re- ported in this paper, it can be concluded that combining the deep structure learning and classical machine learning models into a uniﬁed framework would be an interesting research trend.

*4.4. Parameter discussion*

There are two parameters, the layer size and the penalty pa- rameter ρ , need to be tuned in our model. Here we study the clus- tering performance with respect to different parameter settings. As shown in [Fig. 3,](#bookmark69) we can ﬁnd that the clustering performance is sta- ble with respect to different settings of layer size, while the per- formance is a little sensitive with respect to the penalty parameter ρ . For image datasets (e.g., Yale32, ORL32, COIL and MSRA), bet- ter results can be obtained when ρ is in the range [1e3,1e1]. For gene expression dataset (e.g., Lungml), better results are obtained when ρ is in the range [1e2,1]. While for textual datase (e.g., Cran- med), searching ρ in the range [1e5,1e3] would be a better choice. In general, we can search the parameter ρ in the range [1e3,1] for a relatively good performance.

*4.5. Convergence analysis*

In this subsection, we empirically show how fast our method will converge. [Fig. 4](#bookmark70) shows the convergence curves of our RDKM, where *x*-axis denotes the number of iterations, while *y*-axis de- notes the objective value. It can be observed that the updating rules for our RDKM converge very fast, usually within 100 itera- tions. For data set MSRA, it even converges within 10 iterations, which further demonstrates the effectiveness of the proposed op- timization algorithm.

*4.6. Divergence function selection*

As mentioned in [Section 2,](#bookmark28) several widely used divergence functions can be adopted for residue calculation. We use *l*2,1 -norm in our previous experiment. In this section, we empirically inves- tigate the inﬂuence of different divergence functions to clustering performance.

As can be seen in [Eqs. (6)](#bookmark51) and [(7),](#bookmark53) only the step of updating **Y** will change when the divergence function is changed. That is, the updating of all variables except **Y** are still the same with respect to different divergence functions. We give the updating rules for **Y** when β = 2 (Euclidean Distance), β = 1 (Kullback–Leibler Diver- gence), and β = 0 (Itakura–Saito Divergence) in [Appendix A.](#bookmark52)

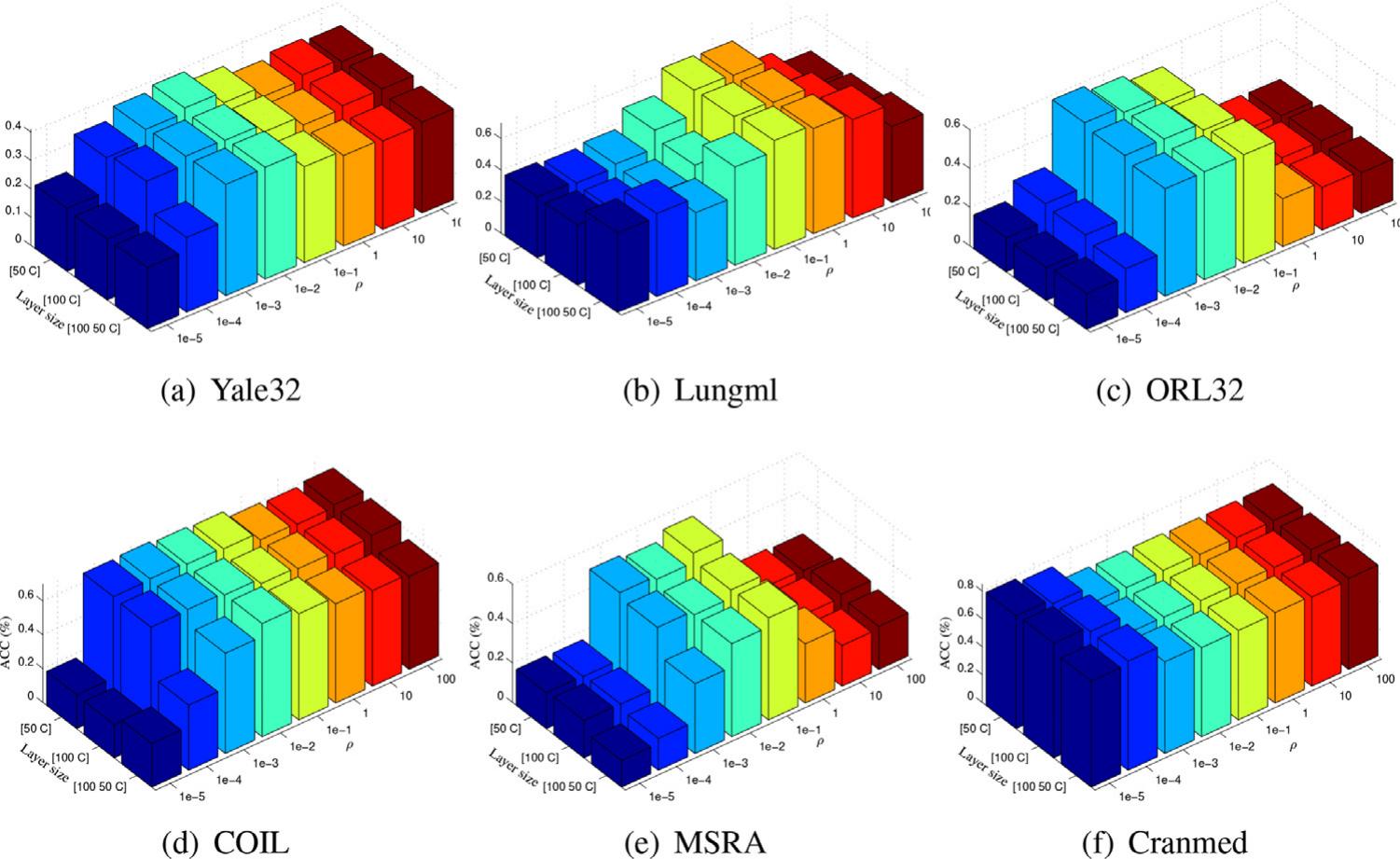
The clustering performance with respect to different divergence functions is shown in [Fig. 5.](#bookmark71) For comparison, the results of our model (i.e., *l*2,1 -norm based divergence function) are also recorded. It is clear that *l*2,1 -norm outperforms other divergence functions on all datasets, which once again veriﬁes the robustness of our model. For the three cases that β = 2, β = 1, and β = 0, we can see that β = 1 obtains good results on dataset Yale32, β = 0 obtains good results on dataset Lungml, while β = 2 achieves better results on

*S. Huang, Z. Kang, Z. Xu et al. Pattern Recognition 117 (2021) 107996*

**Table 2**

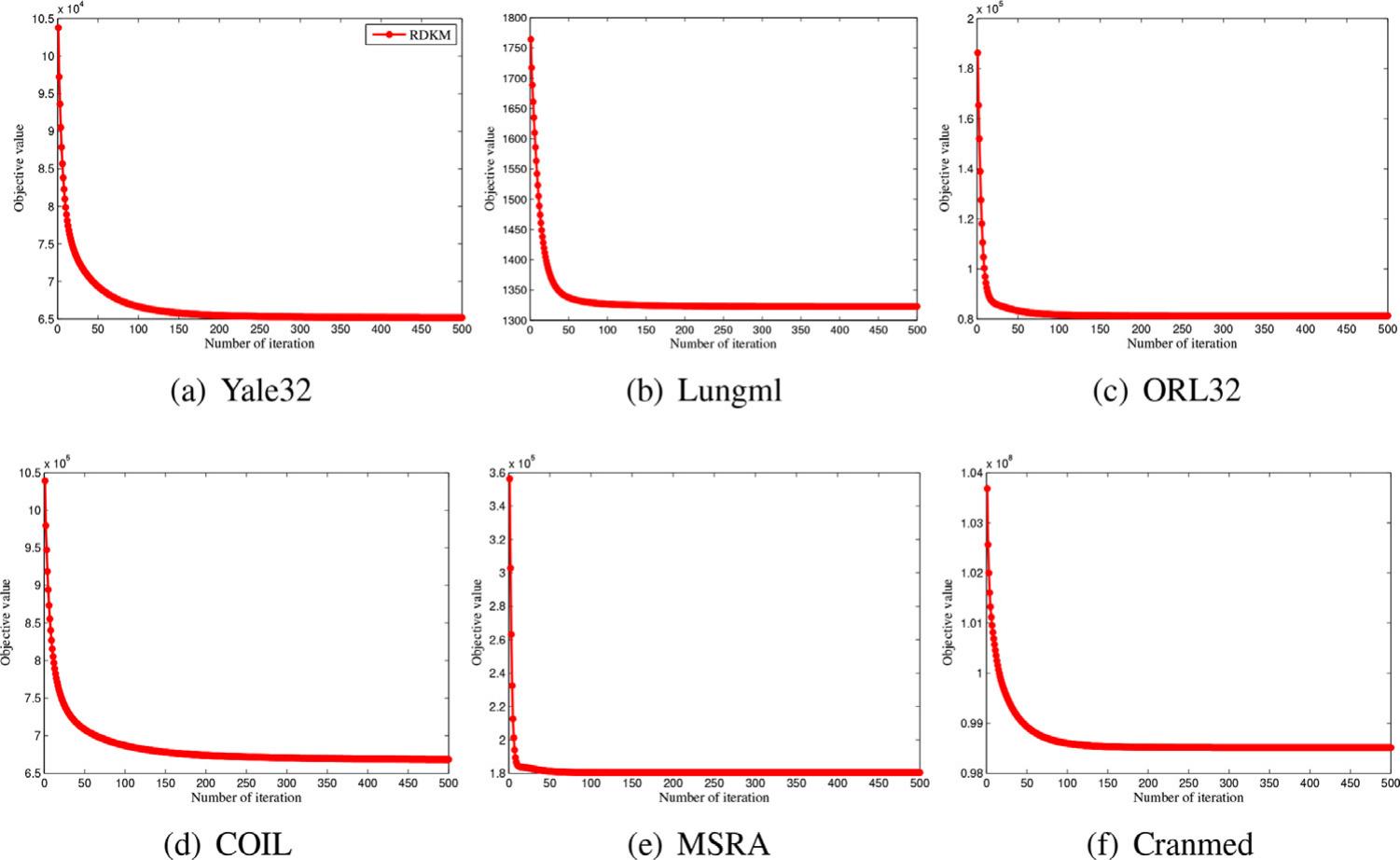
Clustering performance measured by Accuracy/NMI/Purity (mean ± std) of the compared methods. The best performance on each data set is bolded. •/◦ indicates our method is signiﬁcantly better/worse than compared methods (paired *t*-tests at 95% signiﬁcance level). The win/tie/loss (w/t/l) counts for our method are summarized in the last row.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data sets | Metrics | Kmeans | NMF | ONMF | L21NMF | SNMF | DeepSNMF | RDKM |
| Prostateml | ACC | 58.82 ± 0.5• | 58.63 ± 0.6 • | 57.65 ± 0.9 • | 58.14 ± 0.5 • | 58.82 ± 0.5• | 59.98 ± 6.9 • | **62.78** ± **2.1** |
| NMI | 12.39 ± 0.3 • | 12.33 ± 0.3 • | 11.78 ± 0.5 • | 11.98 ± 0.3 • | 12.39 ± 0.2 • | 11.93 ± 1.0 • | **13.36** ± **1.8** |
| Purity | 58.82 ± 0.5 • | 58.63 ± 0.6 • | 57.65 ± 0.9 • | 58.14 ± 0.5 • | 58.82 ± 0.5 • | 59.98 ± 6.9 • | **62.78** ± **2.1** |
| Yael32 | ACC | 37.39 ± 3.1 • | 35.88 ± 3.3 • | 35.58 ± 3.3 • | 38.67 ± 2.7 • | 37.64 ± 2.9 • | 29.09 ± 1.6 • | **40.06** ± **2.8** |
| NMI | 43.05 ± 3.1 • | 42.42 ± 3.1 • | 41.04 ± 3.0 | 45.02 ± 1.6 • | 43.44 ± 2.3 • | 28.92 ± 1.1 • | **48.22** ± **1.9** |
| Purity | 39.76 ± 3.1 • | 37.88 ± 2.7 • | 37.82 ± 3.1 • | 40.48 ± 1.7 • | 39.82 ± 2.6 • | 32.12 ± 1.8 • | **43.52** ± **1.7** |
| Lungml | ACC | 68.92 ± 11.1 | 62.27 ± 7.2 • | 68.92 ± 11.1 | 62.17 ± 5.9 • | 68.92 ± 11.1 | 58.62 ± 7.0 • | **69.01** ± **1.5** |
| NMI | 52.10 ± 8.1 | 47.22 ± 4.4 • | 52.10 ± 8.1 | 47.07 ± 3.3 • | 52.10 ± 8.1 | 16.17 ± 1.8 • | **52.72** ± **5.2** |
| Purity | 87.04 ± 3.0 • | 85.32 ± 4.0 • | 87.04 ± 3.0 • | 84.63 ± 3.2 • | 87.04 ± 3.0 • | 72.17 ± 2.4 • | **89.41** ± **2.9** |
| ORL32 | ACC | 50.30 ± 2.2 • | 51.97 ± 2.8 • | 49.90 ± 3.1 • | 53.40 ± 4.1 | 51.78 ± 3.5 • | 49.86 ± 2.0 • | **54.50** ± **1.2** |
| NMI | 71.06 ± 1.3 • | 72.10 ± 1.3 • | 70.11 ± 1.7 • | 72.70 ± 1.8 • | 71.76 ± 1.9 • | 68.83 ± 1.3 • | **72.94** ± **1.5** |
| Purity | 56.06 ± 2.4 • | 56.37 ± 2.2 • | 55.15 ± 2.9 • | 58.07 ± 3.5 • | 56.07 ± 3.1 • | 57.18 ± 2.1 • | **63.33** ± **2.0** |
| COIL | ACC | 59.43 ± 6.8 • | 62.24 ± 3.1 • | 58.35 ± 6.0 • | 63.49 ± 4.4 • | 63.78 ± 5.9 • | 66.36 ± 6.2 • | **68.03** ± **3.8** |
| NMI | 74.53 ± 2.8 • | 73.12 ± 1.7 • | 72.84 ± 2.6 • | 74.04 ± 2.3 • | 74.91 ± 3.0 • | 77.52 ± 7.4 • | **78.99** ± **1.6** |
| Purity | 64.62 ± 5.1 • | 66.24 ± 2.7 • | 62.95 ± 4.7 • | 67.03 ± 3.5 • | 67.10 ± 4.8 • | 66.94 ± 6.7 • | **71.01** ± **3.6** |
| Semeion | ACC | 57.34 ± 4.7 • | 49.64 ± 5.6 • | 53.87 ± 5.8 • | 47.88 ± 2.7 • | 49.72 ± 5.4 • | 56.07 ± 2.0 • | **62.07** ± **5.9** |
| NMI | 51.93 ± 2.8 • | 42.46 ± 3.5 • | 48.50 ± 2.8 • | 42.47 ± 2.0 • | 43.95 ± 2.8 • | 44.77 ± 1.2 • | **53.97** ± **3.3** |
| Purity | 59.69 ± 3.8 • | 51.65 ± 5.0 • | 56.69 ± 4.4 • | 50.89 ± 2.0 • | 52.32 ± 4.7 • | 58.27 ± 2.6 • | **67.93** ± **4.6** |
| MSRA | ACC | 48.73 ± 4.4 • | 48.93 ± 2.7 • | 48.73 ± 5.8 • | 51.63 ± 5.2 | 49.24 ± 4.4 • | 49.19 ± 1.5 • | **52.24** ± **2.2** |
| NMI | 55.85 ± 5.1 • | 55.86 ± 5.0 • | 55.64 ± 2.8 • | 59.06 ± 4.2 | 55.44 ± 4.9 • | **60.10** ± **0.2** | 59.83 ± 2.6 |
| Purity | 52.42 ± 3.8 • | 51.86 ± 3.3 • | 52.30 ± 4.4 • | 55.14 ± 4.2 | 51.77 ± 3.8 • | 54.75 ± 0.2 • | **55.79** ± **2.4** |
| Text | ACC | 91.84 ± 2.1 • | 93.85 ± 3.9 | 92.47 ± 2.9 • | 90.21 ± 4.0 • | 90.99 ± 2.0 • | 90.67 ± 4.5 • | **93.88** ± **5.7** |
| NMI | 61.31 ± 6.5 | 61.21 ± 1.5 • | 60.88 ± 1.9 • | 60.81 ± 3.0 • | 57.85 ± 5.2 • | 60.01 ± 4.2 • | **61.55** ± **5.4** |
| Purity | 91.84 ± 2.1 • | **94.08** ± **3.6** ◦ | 92.71 ± 2.7 • | 90.62 ± 3.2 • | 90.99 ± 2.0 • | 90.67 ± 4.5 • | 93.88 ± 5.7 |
| Cranmed | ACC | 74.58 ± 0.1 • | 80.13 ± 8.8 • | 77.31 ± 1.2 • | 77.39 ± 2.5 • | 76.49 ± 4.9 • | 80.23 ± 3.1 • | **82.31** ± **3.9** |
| NMI | 18.79 ± 0.3 • | 31.67 ± 5.6 | 20.74 ± 0.3 • | 20.84 ± 1.2 • | 24.66 ± 5.4 • | 25.05 ± 2.4 • | **32.89** ± **2.3** |
| Purity | 74.58 ± 0.1 • | 80.38 ± 8.0 • | 77.78 ± 3.4 • | 77.67 ± 3.2 • | 79.10 ± 6.1 • | **82.51** ± **3.0** ◦ | 82.31 ± 3.9 |
| MINIST | ACC | 54.84 ± 4.1 • | 52.30 ± 3.3 • | **58.84** ± **4.7** ◦ | 54.00 ± 5.1 • | 48.65 ± 2.2 • | 56.18 ± 1.2◦ | 55.46 ± 3.7 |
| NMI | 49.44 ± 1.6 • | 43.21 ± 2.1 • | 49.92 ± 1.4◦ | 45.26 ± 3.0 • | 41.26 ± 1.8 • | **49.81** ± **2.0** ◦ | 49.53 ± 2.4 |
| Purity | 58.38 ± 2.1 • | 54.22 ± 2.4 • | 59.84 ± 2.3 • | 57.18 ± 4.5 • | 52.89 ± 2.7 • | 60.73 ± 2.4 • | **62.10** ± **2.8** |
| Cacmcisi | ACC | 91.99 ± 0.2 • | 89.75 ± 5.4 • | 94.96 ± 0.6 • | 95.37 ± 0.8 • | 92.22 ± 0.3 • | 92.80 ± 3.5 • | **97.12** ± **7.7** |
| NMI | 58.47 ± 0.1 • | 60.01 ± 2.5 • | 70.42 ± 0.2 • | 72.05 ± 0.4 • | 70.52 ± 0.2 • | 70.07 ± 3.3 • | **73.06** ± **2.4** |
| Purity | 91.99 ± 0.2 • | 91.70 ± 3.2 • | 94.96 ± 0.6 • | 95.37 ± 0.8 • | 93.69 ± 0.7 • | 94.86 ± 4.1 • | **97.12** ± **7.7** |
| Classic | ACC | 67.45 ± 0.3 • | 70.31 ± 3.2 • | 76.52 ± 6.6 • | 76.19 ± 6.4 • | 74.44 ± 2.0 • | 72.40 ± 1.3 • | **76.98** ± **5.9** |
| NMI | 46.76 ± 1.3 • | 50.12 ± 2.2 • | 47.91 ± 6.4 • | 57.45 ± 7.9 • | 55.30 ± 2.6 • | 56.27 ± 2.5 • | **59.61** ± **4.1** |
| Purity | 70.48 ± 1.0 • | 74.09 ± 3.3 • | 77.74 ± 5.2 • | 79.73 ± 4.2 • | 76.42 ± 1.2 • | 78.41 ± 1.9 • | **81.07** ± **4.1** |
| RDKM: | ACC | 11/1/0 | 11/1/0 | 10/1/1 | 10/2/0 | 11/1/0 | 11/0/1 |  |
| w/t/l | NMI | 10/2/0 | 11/1/0 | 9/2/1 | 11/1/0 | 11/1/0 | 10/1/1 |  |
| Purity | 12/0/0 | 11/0/1 | 12/0/0 | 11/1/0 | 12/0/0 | 11/0/1 |  |

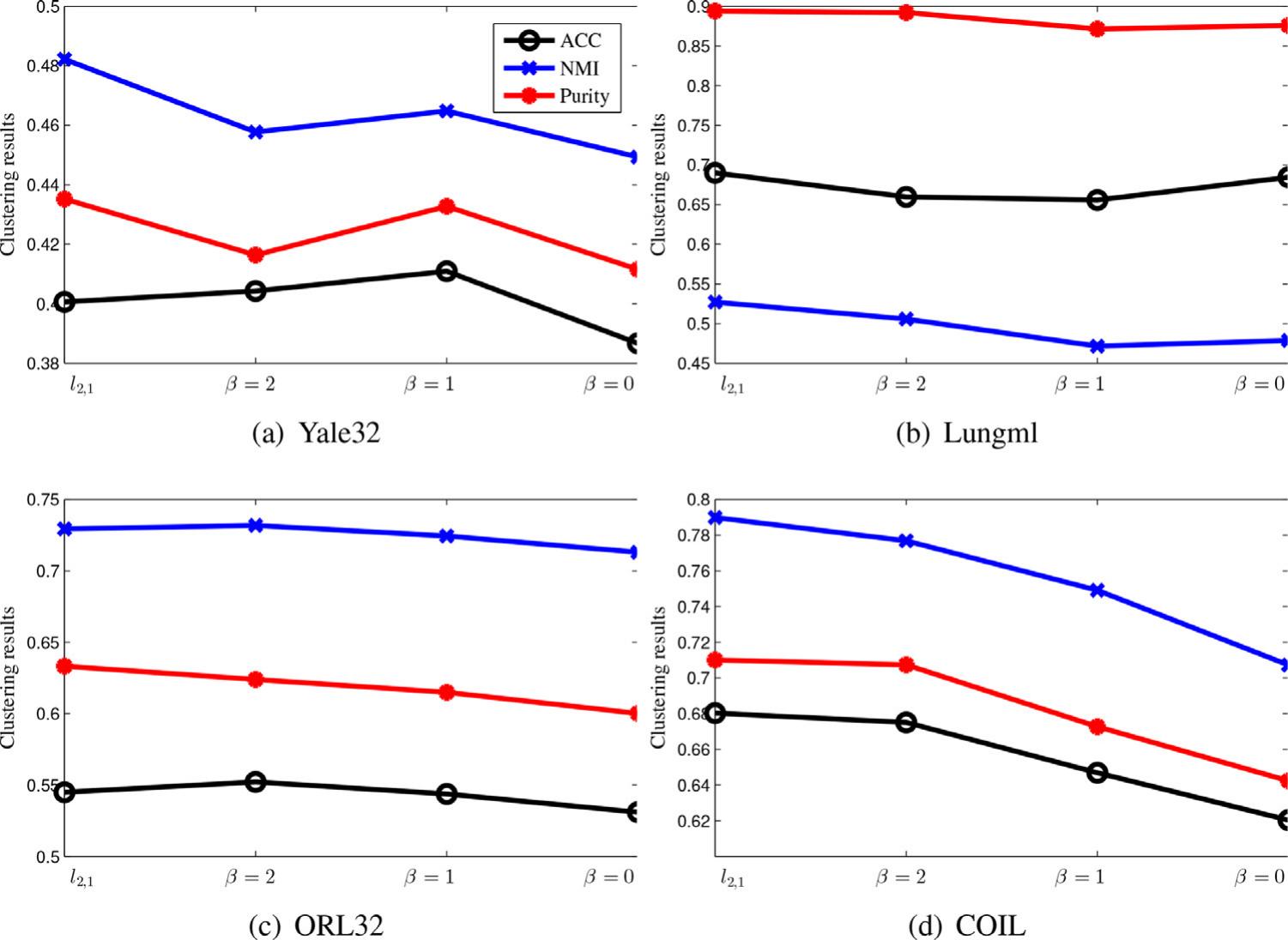


**Fig. 3.** Clustering results of RDKM w.r.t. layer size and parameter ρ .

*S. Huang, Z. Kang, Z. Xu et al. Pattern Recognition 117 (2021) 107996*



**Fig. 4.** Convergence speed of RDKM.



**Fig. 5.** Clustering performance of RDKM w.r.t. divergence functions.

datasets ORL32 and COIL. That is, different divergence functions get better results on different datasets. Generally, *l*2,1 -norm may be a better choice as it consistently achieves good performance.

**5. Conclusion**

In this paper, we introduced a robust deep *k*-means model to learn the hidden representations associate with different implicit lower-level attributes. By using the deep structure to hierarchically perform *k*-means, the hierarchical semantics of data can be ex- ploited in a layerwise way. Data samples from the same class are forced to be closer layer by layer, which is beneﬁcial for cluster-

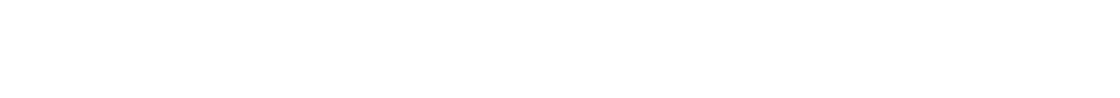
ing task. The objective function of our model is derived to a more trackable form such that the optimization problem can be tackled more easily and the ﬁnal robust results can be obtained. Experi- mental results over 12 benchmark data sets substantiate that: (i) the proposed model achieves a breakthrough in clustering perfor- mance, compared with both classical and state-of-the-art methods; (ii) the clustering performance is robust with respect to different settings of layer size as well as the divergence functions; (iii) the proposed optimization algorithm is effective and converges very fast. In our future work, it would be interesting to combine our deep model and other machine learning models (e.g., kernel learn- ing and classiﬁcation methods) into a uniﬁed framework.

*S. Huang, Z. Kang, Z. Xu et al.*

**Declaration of Competing Interest**

The authors declare that they have no known competing ﬁnan- cial interests or personal relationships that could have appeared to inﬂuence the work reported in this paper.

**Acknowledgments**

[](https://doi.org/10.13039/501100012226)This work was partially supported by the State Key Program of the National Science Foundation of China under Grant 61836006, the National Natural Science Fund for Distinguished Young Scholar under Grant 61625204, [Sichuan Science and Technology Program](https://doi.org/10.13039/100012542) [under Grant](https://doi.org/10.13039/501100012226) [2020YFG0323,](#bookmark73) [and the Fundamental Research Funds for the Central Universities under Grant](https://doi.org/10.13039/501100012226) [1082204112364.](#bookmark74)

**Appendix A. Updating rules for Y w.r.t different divergence** **functions**

For different divergence functions, the main difference of the corresponding optimization algorithms is the updating rules for **Y**. Here we introduce the updating rules for **Y** in terms of different divergence functions, particularly, when β = 2, β = 1, and β = 0.

**When** β = 2**(Euclidean Distance),** the objective function w.r.t **Y** is

L*Y*β =2 = D2 (**X**|**Y**) + (**μ** , **Y** − **U**1**U**2 . . . **U***r***V**〉+

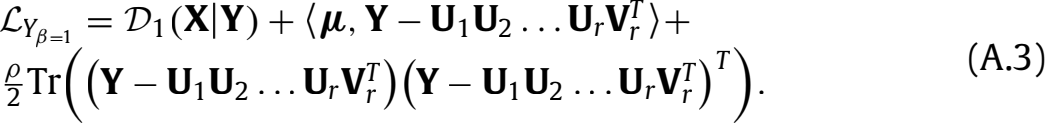
Tr ((**Y** − **U**1**U**2 . . . **U***r***V**)(**Y** − **U**1**U**2 . . . **U***r***V**)*T* ). (A.1)

Calculating the derivative of L*Y*β =2 w.r.t. **Y** and setting it to 0, we have

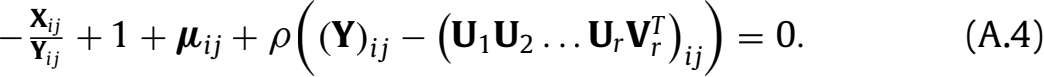
**Y** = (2**X** + ρ**U**1**U**2 . . . **U***r***V**− **μ**)/(2 + ρ). (A.2)

**When** β = 1 (**Kullback**–**Leibler Divergence),** the objective

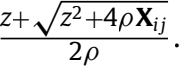
function w.r.t **Y** is



Calculating the derivative of L*Y*β =1 w.r.t. **Y** in an element-wise way, we have

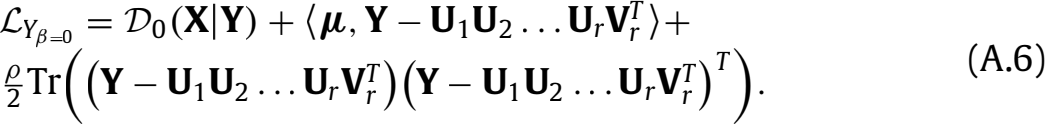


Thus the solution for **Y** is

**Y***ij* =  (A.5)

where *z* = ρ (**U**1**U**2 . . . **U***r***V**)*ij* − **μ***ij* − 1.

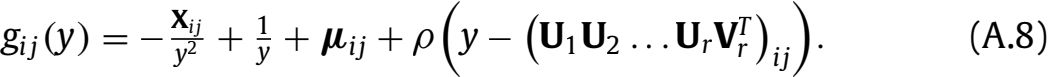
**When** β = 0 (**Itakura**–**Saito Divergence),** the objective function w.r.t **Y** is

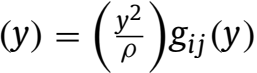


It is clear that **Y**∗ would be a minimizer if and only if the gradient vanishes at **Y**∗ and the the Hessian matrix is positive deﬁnite:

*gij* (**Y**) = 0, *g* (**Y**) = 0, ∀*i*, *j*, (A.7)

where *gij* represents the derivative w.r.t **Y***ij* and can be deﬁned as



Denote *pij* such that *pij* is a cubic polynomial. It is

clear that *pij* has the same roots as *gij* , and *p* has the same sign as *g* . *pij* can be explicitly expressed as

*pij* (*y*) = *y*3 + **A***ijy*2 + *y* − **X***ij* , (A.9)

*Pattern Recognition 117 (2021) 107996*

where **A** =  **μ** − **U**1**U**2 . . . **U***r***V**. Substituting *y* = *s* − **A***ij* in [Eq. (A.9),](#bookmark13) a depressed cubic can be obtained

*q*(*s*) = *s*3 + 3**B***ijs* − 2**R***ij* , (A.10)

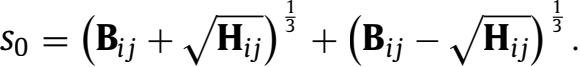
where **B***ij* =  − **A**, **R***ij* = − **A**+ **A***ij* + **X***ij*. We want to

search a positive root *s*0 > 0 of *q*(*s*) such that *q*/ (*s*0 ) < 0. When dealing with nonnegative data (i.e., **X***ij* ≥ 0), we can search at least one such root as *p*(0) ≤ 0 and *p*(∞ ) → ∞ . The roots of *qij* are re- lated to the roots of *pij* by *yt* = *st* − **A***ij* (*t* = 0, 1, 2, i.e., there are at most three roots for [Eq. (A.10))](#bookmark72). The discriminant **H** of *qij* can be deﬁned as

**H***ij* = **B**+ **R**, (A.11)

and there are three cases:

1) **H***ij* > 0, one real root:

 (A.12)

Correspondingly, the root *y*0 of *pij* must be positive and the mini- mizer of [Eq. (A.6).](#bookmark4)

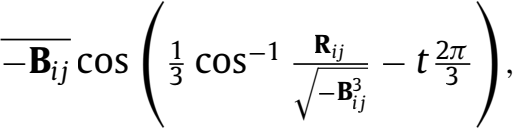
2) **H***ij* = 0, two distinct real roots:

A double root*s*1 = *s*2 = − *s*0 can be derived. However, the dou-

ble roots correspond to the point of inﬂections of *qij* , i.e., *g* (*y*1 ) =

0. Thus *y*1 is not a minimizer of [Eq. (A.6).](#bookmark4) As a result, the relevant root is still *s*0 .

3) **H***ij* < 0, three distinct real roots:

*st* = 2√ (A.13)

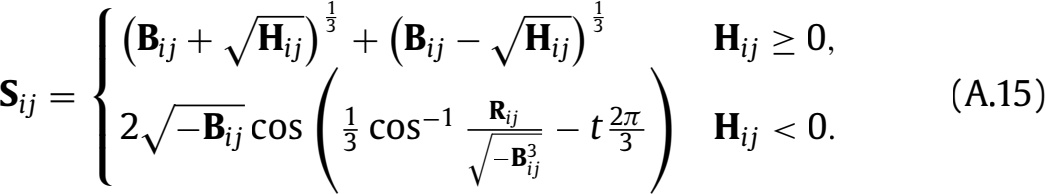
for *t* = 0, 1, 2. When there are three distinct roots, *p* can only be

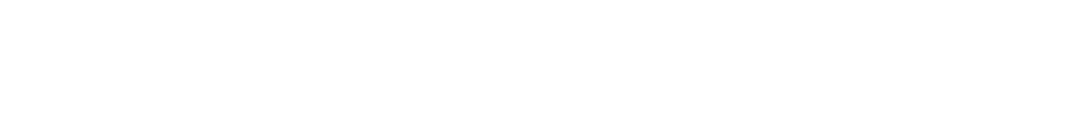
positive at the smallest and largest roots. Since *s*0 ≥ *s*1 ≥ *s*2 , we only need to check *y*0 and*y*2 (the latter only if *y*2 > 0). For simplic- ity, we always take *y*0 , which is guaranteed to be at least a local minima of [Eq. (A.6).](#bookmark4) We omit the solution derivation and for more details please refer to [[40].](#bookmark75)

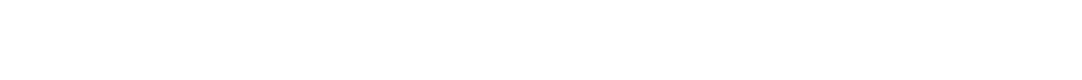
In summary, we update **Y** by

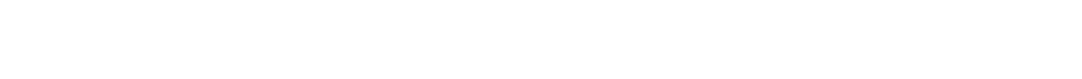
**Y***ij* = **S***ij* − **A***ij* , (A.14)

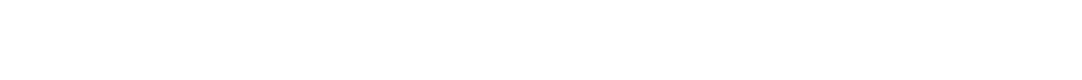
where

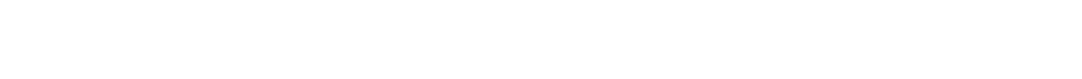


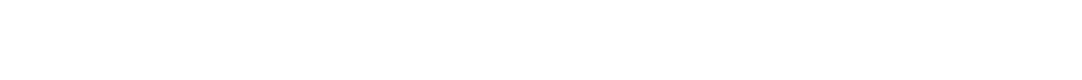
[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0001)**References**

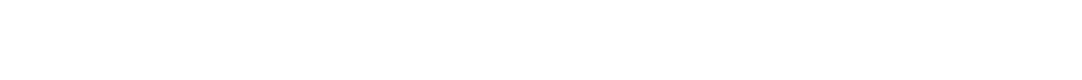
[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0002)[[1] S. Huang,Z. Kang,Z. Xu, Deep *k*-means: a simple and effective method for data clustering, in: Proceedings of the International Conference on Neural Comput- ing for Advanced Applications, Springer, 2020, pp. 272–283.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0001)

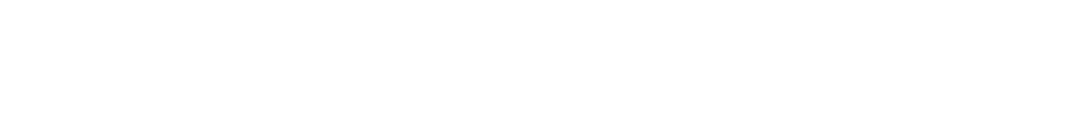
[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0003)[[2] A.K. Jain, Data clustering: 50 years beyond *k*-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0002)

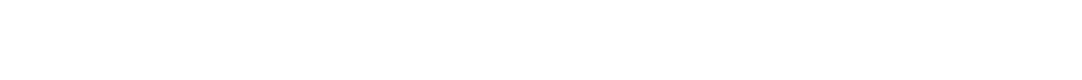
[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0004)[[3] A. Banerjee, S. Merugu, I.S. Dhillon, J. Ghosh, Clustering with Bregman diver- gences, J. Mach. Learn. Res. 6 (2005) 1705–1749.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0003)

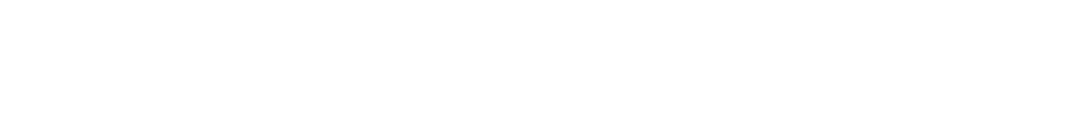
[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0005)[[4] V. Tunali,T. Bilgin,A. Camurcu, An improved clustering algorithm for text min- ing: multi-cluster spherical *k*-means, Int. Arab J. Inf. Technol. 13 (1) (2016).](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0004)

[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0006)[[5] S.V. Ault, R.J. Perez, C.A. Kimble, J. Wang, On speech recognition algorithms, Int. J. Mach. Learn. Comput. 8 (6) (2018) 518–523.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0005)

[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0007)[[6] L. Wang, C. Pan, Robust level set image segmentation via a local correntropy- -based *k*-means clustering, Pattern Recognit. 47 (5) (2014) 1917–1925.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0006)

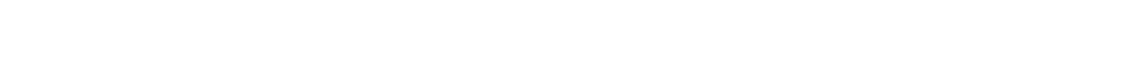
[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0008)[[7] Y. Ren, C. Domeniconi, G. Zhang, G. Yu, Weighted-object ensemble clustering: methods and analysis, Knowl. Inf. Syst. 51 (2) (2017) 661–689.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0007)

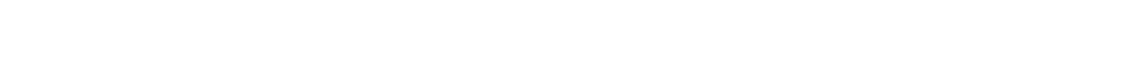
[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0009)[[8] J. Macqueen, Some methods for classiﬁcation and analysis of multivariate ob- servations, in: Proceedings of Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0008)

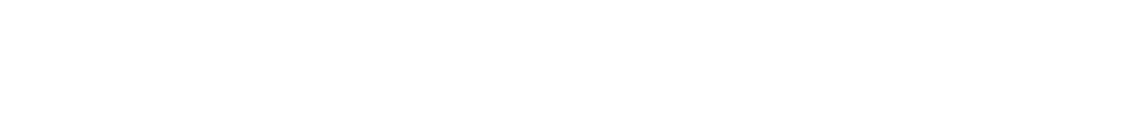
[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0010)[[9] A.Y. Ng,M.I. Jordan,Y. Weiss, On spectral clustering: analysis and an algorithm, in: Advances in Neural Information Processing Systems, 2002, pp. 849–856.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0009)

[[10] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factor- ization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2010) 1548–1560.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0010)

*S. Huang, Z. Kang, Z. Xu et al.*

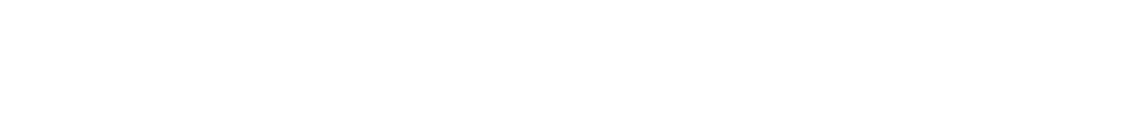
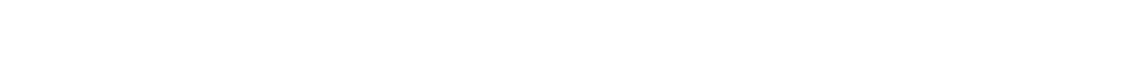
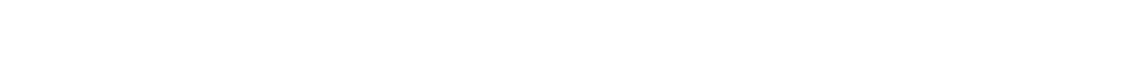
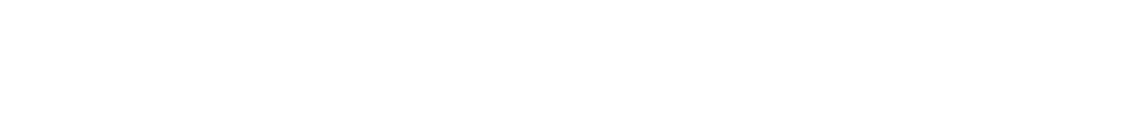
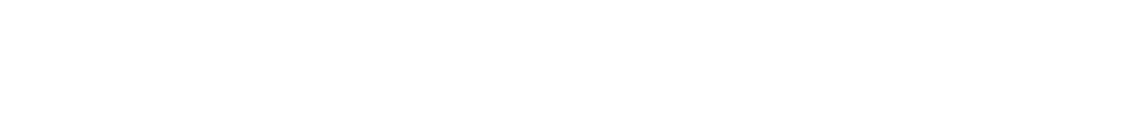
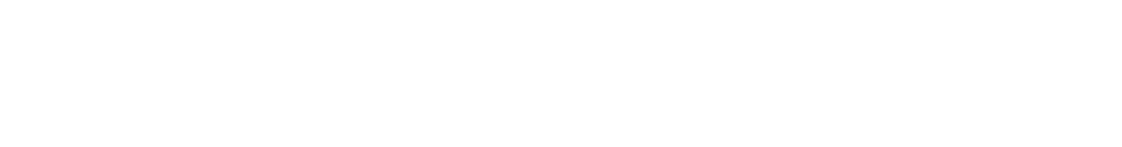
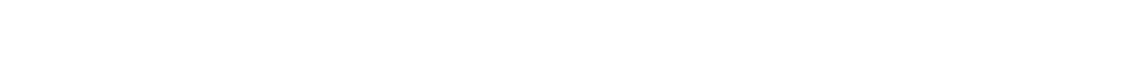
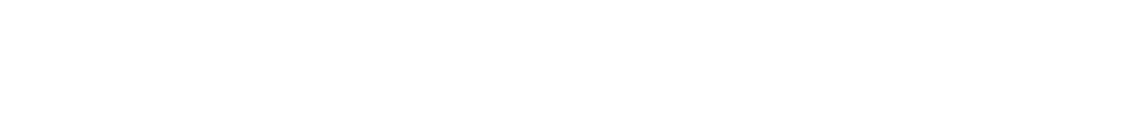
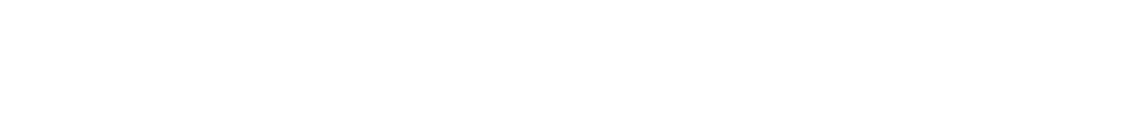
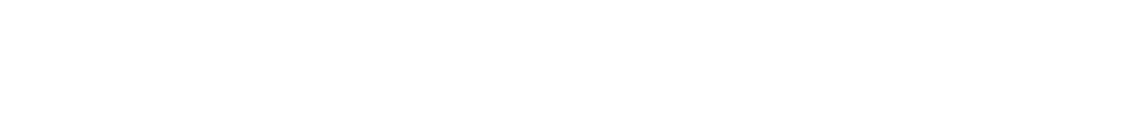
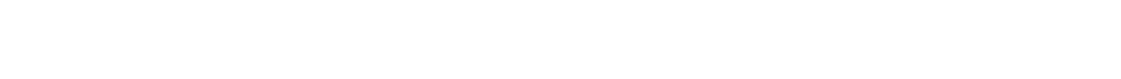
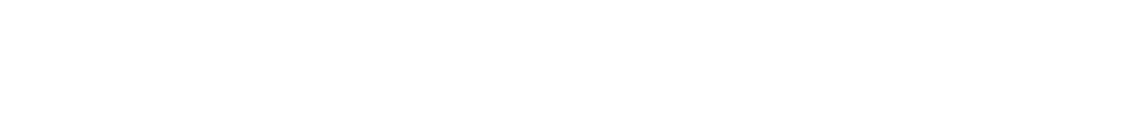
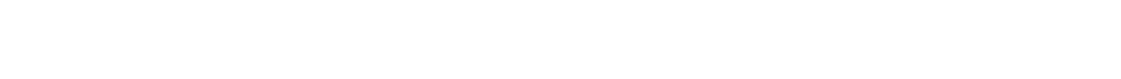
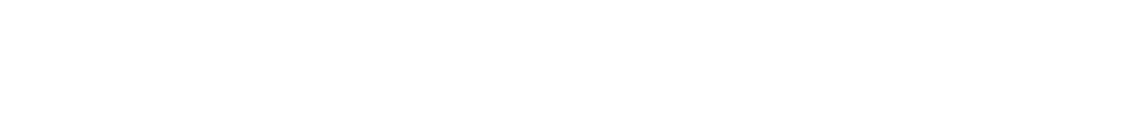
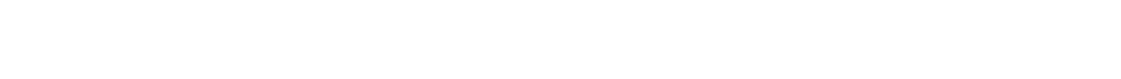
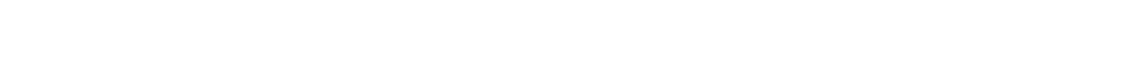
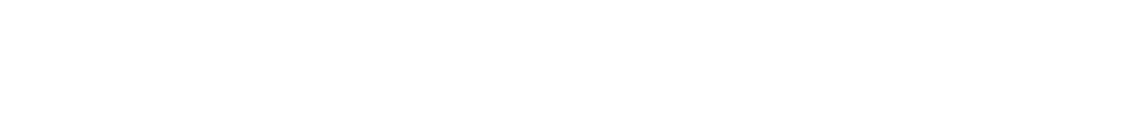
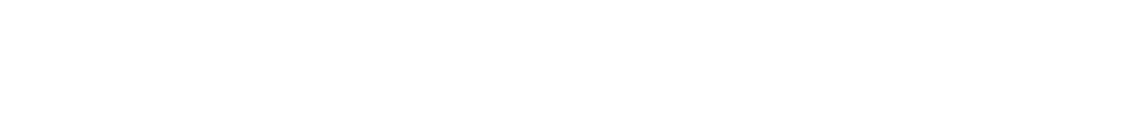
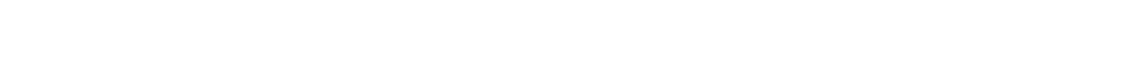
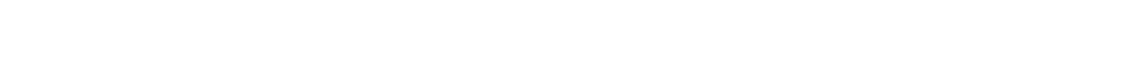
[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0011)

[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0012)[[11] E. Gokcay, J.C. Principe, Information theoretic clustering, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2) (2002) 158–171.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0011)

[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0013)[[12] F. Gullo, G. Ponti, A. Tagarelli, S. Greco, An information-theoretic approach to hierarchical clustering of uncertain data, Inf. Sci. 402 (2017) 199–215.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0012)

[[13] S. Huang, I. Tsang, Z. Xu, J.C. Lv, Measuring diversity in graph learning: a uni- ﬁed framework for structured multi-view clustering, IEEE Trans. Knowl. Data Eng. (2021) 1–15.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0013)

[[14] H. Zhao, Z. Ding, Y. Fu, Multi-view clustering via deep matrix factorization, in: Proceedings of the 21st AAAI Conference on Artiﬁcial Intelligence, 2017, pp. 2921–2927.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0014)

[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0014)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0015)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0016)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0017)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0018)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0019)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0020)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0021)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0022)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0023)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0024)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0025)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0026)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0027)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0028)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0029)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0030)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0031)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0032)[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0033)[15] [X. Wu,](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0015) [V. Kumar,](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0015) [J.R. Quinlan,](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0015) [J. Ghosh,](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0015) [Q. Yang,](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0015) [H. Motoda,](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0015) [G.J. McLachlan,](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0015)

[A. Ng, B. Liu, S.Y. Philip, et al., Top 10 algorithms in data mining, Knowl. Inf. Syst. 14 (1) (2008) 1–37.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0015)

[[16] P. Ji, T. Zhang, H. Li, M. Salzmann, I. Reid, Deep subspace clustering networks, in: Advances in Neural Information Processing Systems, 2017, pp. 24–33.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0016)

[[17] P. Zhou, Y. Hou, J. Feng, Deep adversarial subspace clustering, in: Proceed- ings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1596–1604.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0017)

[[18] X. Guo, L. Gao, X. Liu, J. Yin, Improved deep embedded clustering with local structure preservation, in: Proceedings of the 26th International Joint Confer-](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0018) [ence on Artiﬁcial Intelligence, 2017, pp. 1753–1759.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0018)

[[19] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, H. Huang, Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5736–5745.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0019)

[[20] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J.T. Zhou, S. Yang, Structured autoencoders for subspace clustering, IEEE Trans. Image Process. 27 (10) (2018) 5076–5086.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0020)

[[21] X. Peng,H. Zhu,J. Feng,C. Shen,H. Zhang,J.T. Zhou, Deep clustering with sam- ple-assignment invariance prior, IEEE Trans. Neural Netw. Learn. Syst. (2019) 1–12.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0021)

[[22] J. Zhang, C.-G. Li, C. You, X. Qi, H. Zhang, J. Guo, Z. Lin, Self-supervised convo- lutional subspace clustering network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5473–5482.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0022)

[[23] C. Ding, X. He, K-means clustering via principal component analysis, in: Pro- ceedings of the 21st International Conference on Machine Learning, 2004, pp. 29–37.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0023)

[[24] C. Buchta, M. Kober, I. Feinerer, K. Hornik, Spherical *k*-means clustering,J. Stat. Softw. 50 (10) (2012) 1–22.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0024)

[[25] S. Khanmohammadi, N. Adibeig, S. Shanehbandy, An improved overlapping *k*-means clustering method for medical applications, Expert Syst. Appl. 67 (2017) 12–18.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0025)

[[26] K.M. Kumar, A.R.M. Reddy, An eﬃcient *k*-means clustering ﬁltering algorithm using density based initial cluster centers, Inf. Sci. 418 (2017) 286–301.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0026)

[[27] S. Chakraborty, D. Paul, S. Das, J. Xu, Entropy weighted power *k*-means cluster- ing, in: International Conference on Artiﬁcial Intelligence and Statistics, 2020, pp. 691–701.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0027)

[[28] M. Capó, A. Pérez, J.A. Lozano, An eﬃcient approximation to the *k*-means clus- tering for massive data, Knowledge-Based Syst. 117 (2017) 56–69.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0028)

[[29] Y. Bengio, Learning deep architectures for AI, Found. Trends® Mach. Learn. 2 (1) (2009) 1–127.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0029)

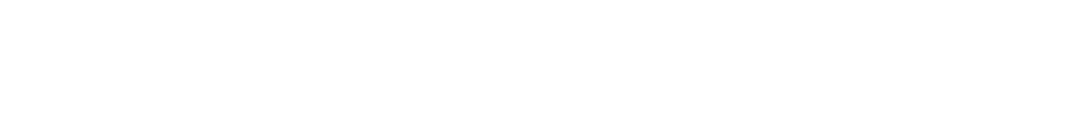
[[30] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix *t*-factorizations for clustering, in: Proceedings of the 12th ACM SIGKDD International Confer- ence on Knowledge Discovery and Data Mining, 2006, pp. 126–135.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0030)

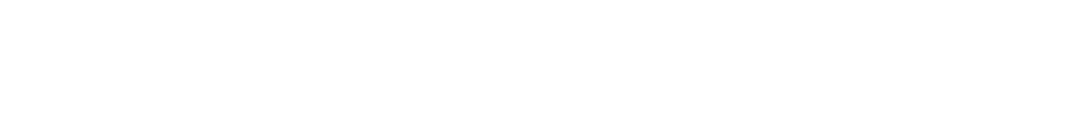
[[31] S. Huang, H. Wang, T. Li, T. Li, Z. Xu, Robust graph regularized nonnegative](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0031) [matrix factorization for clustering, Data Min. Knowl. Discov. 32 (2) (2018) 483–503.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0031)

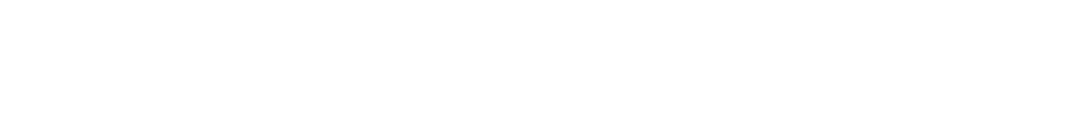
[[32] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Ad- vances in Neural Information Processing Systems, 2001, pp. 556–562.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0032)

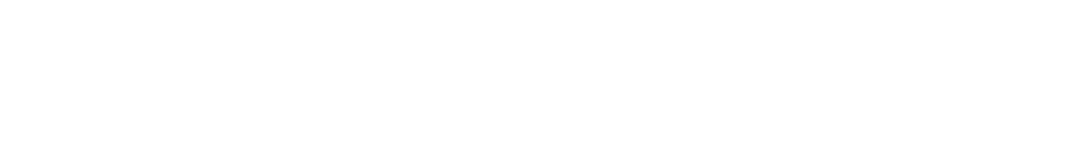
[[33] C. Févotte, J. Idier, Algorithms for nonnegative matrix factorization with the β-divergence, Neural Comput. 23 (9) (2011) 2421–2456.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0033)

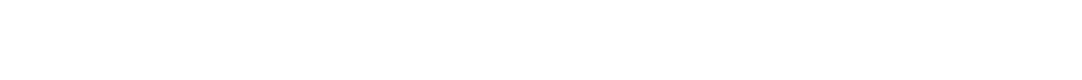
*Pattern Recognition 117 (2021) 107996*

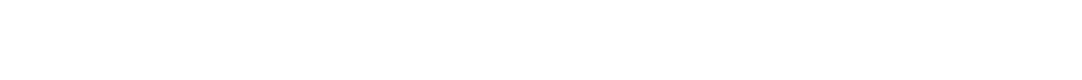
[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0034)

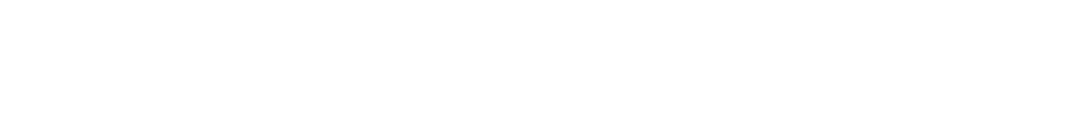
[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0035)[[34] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, B.W. Schuller, A deep matrix factor- ization method for learning attribute representations, IEEE Trans. Pattern Anal. Mach. Intell. 39 (3) (2017) 417–429.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0034)

[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0036)[[35] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimiza- tion and statistical learning via the alternating direction method of multipliers, Found. Trends® Mach. learn. 3 (1) (2011) 1–122.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0035)

[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0037)[[36] D. Kong, C. Ding, H. Huang, Robust nonnegative matrix factorization using l21-norm, in: Proceedings of the 20th ACM International Conference on In- formation and Knowledge Management, 2011, pp. 673–682.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0036)

[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0038)[[37] H. Gao, F. Nie, W. Cai, H. Huang, Robust capped norm nonnegative ma- trix factorization: capped norm NMF, in: Proceedings of the 24th ACM In- ternational on Conference on Information and Knowledge Management, 2015, pp. 871–880.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0037)

[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0039)[[38] C. Ding, T. Li, M.I. Jordan, Convex and semi-nonnegative matrix factorizations, IEEE Trans. Pattern Anal. Mach. Intell. 32 (1) (2010) 45–55.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0038)

[](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0040)[[39] S. Huang, Z. Kang, Z. Xu, Auto-weighted multi-view clustering via deep matrix decomposition, Pattern Recognit. 97 (2020) 1–11.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0039)

[[40] D.L. Sun, R. Mazumder, Non-negative matrix completion for bandwidth exten- sion: a convex optimization approach, in: Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing, 2013, pp. 1–6.](http://refhub.elsevier.com/S0031-3203(21)00183-7/sbref0040)

**Shudong Huang:** received his Ph.D. degree in computer science and engineer- ing from the University of Electronic Science and Technology of China (UESTC), Chengdu, China. He is currently an associate research professor with the College of Computer Science, Sichuan University, Chengdu, China. His research interests in- clude machine learning, deep learning, pattern recognition and data mining. He has published nearly 20 research papers in several related conferences and journals, including IEEE TCYB, Pattern Recognition, Data Mining and Knowledge Discovery, Information Sciences, Knowledge-Based Systems, Neurocomputing, IJCAI, IJCNN, etc. He has been a PC member or reviewer to a number of top conferences and journals such as AAAI, IJCAI, IEEE TNNLS, IEEE TFS, Neurocomputing, etc.

**Zhao Kang:** Dr. Kang received his Ph.D. degree in Computer Science from Southern Illinois University Carbondale, USA, in 2017. Currently, he is an assistant professor at the School of Computer Science and Engineering at University of Electronic Sci- ence and Technology of China (UESTC). His research interests are machine learn- ing, data mining, pattern recognition, and deep learning. He has published over 40 research papers in top-tier conferences and journals, including AAAI, IJCAI, ICDE, CVPR, SIGKDD, ICDM, CIKM, SDM, ACML, IEEE Transactions on Cybernetics, ACM TIST, ACM TKDD, Pattern Recognition, Neurocomputing, Knowledge-Based Systems. He has been a PC member or reviewer to a number of top conferences such as AAAI, IJCAI, CVPR, ICCV, MM, ICDM, CIKM, etc. He regularly servers as a reviewer to IEEE TNNLS, IEEE Transactions on Cybernetics, IEEE TKDE, Neurocomputing, etc.

**Zenglin Xu:** received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong. He is currently a full professor in University of Electronic Science & Technology of China. He has been working at Michigan State University, Cluster of Excellence at Saarland University and Max Planck Institute for Informatics, and later Purdue University. Dr. Xu’s research interests include machine learning and its applications in information retrieval, health informatics, and social network analysis. He has been elected in the 2013’s China Youth 1000-talent Pro- gram. He is the recipient of the outstanding student paper honorable mention of AAAI 2015, the best student paper runner up of ACML 2016, and the 2016 young researcher award from APNNS.

**Quanhui Liu:** received his Ph.D. degree in computer science and engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China. He is currently an associate research professor with the College of Computer Science, Sichuan University, Chengdu, China. His research interests include social networks, deep learning, and complex network.