Front. Comput. Sci., 2023, 17(3): 173322
https://doi.org/10.1007/s11704-022-1376-2

RESEARCH ARTICLE

Heterogeneous clustering via adversarial deep Bayesian generative model

Xulun YE (✉), Jieyu ZHAO

Institute of Computer Science and Technology, Ningbo University, Ningbo 315211, China

Abstract This paper aims to study the deep clustering problem with heterogeneous features and unknown cluster number. To address this issue, a novel deep Bayesian clustering framework is proposed. In particular, a heterogeneous feature metric is first constructed to measure the similarity between different types of features. Then, a feature metric-restricted hierarchical sample generation process is established, in which sample with heterogeneous features is clustered by generating it from a similarity constraint hidden space. When estimating the model parameters and posterior probability, the corresponding variational inference algorithm is derived and implemented. To verify our model capability, we demonstrate our model on the synthetic dataset and show the superiority of the proposed method on some real datasets. Our source code is released on the website: Github.com/yexlwh/Heterogeneousclustering.

Keywords dirichlet process, heterogeneous clustering, generative adversarial network, laplacian approximation, variational inference

1 Introduction

As a classical and popular tool for data analysis, clustering methods are widely studied in the computer vision and machine learning community [1−3]. Nowadays, due to the high efficiency of Deep Neural Network (DNN), many works try to extend the supervised DNN model to the clustering task [4,5]. Although these DNN based clustering methods are quite efficiency in different applications, there still exists some limitations due to the facts that: (1) Existing deep learning methods always assume that dataset contains the same feature and data is extracted from the same conditions. Real applications are complicated; a sample with different features may act like different clusters in the feature space; (2) Many DNN based methods require a predefined cluster number which is usually not available in many real applications.

To address these problems, in this paper, a novel heterogeneous metric is first constructed to capture the similarity between different features. Then, Dirichlet Process (DP) is exploited to model the unknown class number. When clustering the heterogeneous dataset, we assume that the class from different features is generated from a same hidden space. Due to the limitation that conventional Bayesian model is unable to characterize the complicated distribution in the real world, we treat variables generated from the hidden space as the pseudo features, and the real world observations as the generating samples from the pseudo features with a Generative Adversarial Network (GAN). Additionally, we constrain the pseudo features generated from a same hidden space with the heterogeneous metric, which assures that features from a same class are generated from a same hidden space. We conclude our main contributions as:

1) A novel deep Bayesian generative model is proposed to model the unknown cluster number and cluster the heterogeneous feature samples.

2) A heterogeneous metric is proposed and integrated with the generation process to capture the similarity between different feature spaces.

3) For the non-conjugated property and the use of GAN, we derive a Laplacian approximation boosted mean field variational inference for the model inference and optimization.

Related work In the field of deep neural network based clustering, recently, Jiang et al. [1] construct a clustering method by combining the Variational Auto-Encoders (VAE) and Gaussian Mixture Model (GMM). Xie et al. [6] propose the Deep Embedded Clustering (DEC) by incorporating the K-means with a deep neural network based t-SNE. Pan et al. [7−10] extend the conventional subspace clustering method to the non-linear manifold clustering algorithm by learning a deep embedding. Dizaji et al. [4,11] extend the GAN to the clustering task with a mixture model and a learned discriminative embedding. Tian et al. [12,13] seek a deep spectral clustering method by requiring the consistence of the DNN embedding and the given graph. Cheng et al. [5,14] exploit the DNN model in the multi-view clustering task. Menapace et al. extend the DNN model to the clustering task with the domain shift [15]. Despite various advantages they have, they need to specify the cluster number in a prior. To overcome this problem, Tapaswi et al. [16] and Yang et al. [17] exploit the supervised clustering method. Although these methods are effective in many real applications, unfortunately, they are unable to cluster the heterogeneous data.

Received July 10, 2021; accepted April 27, 2022

E-mail: yexlwh@163.com

When handling the heterogeneous clustering task, Heterogeneous Domain Adaption (HDA) is a method which relates to our approach. Generally, HDA transforms the dataset with different features to a same feature space, and can be roughly categorized into semi-supervised method and unsupervised method [18]. In the first method, many works seek to exploit both the source and target labeled data to adapt the heterogeneous features [18,19]. Compared to the semi-supervised method, although label is no more a requirement in the unsupervised method, labeled source domain is sometimes still a necessary information [20−23]. For a much more detailed summarization, please refer to [18,23,24]. Different to these HDA methods, our framework offers a unsupervised end-to-end heterogeneous clustering method, and exhibit another superiority of real sample generation. In addition, multi-view clustering methods [25−29] are also related to our heterogeneous clustering, in which the clustering methods assume that each sample has multiple types of features, and clustering accuracy can be improved by exploiting the complementarity of different feature information. But, compared to their assumption that samples share the same feature, our task assumes that each sample has only one type feature and features from different samples have no intersection.

Dirichlet process is a widely studied model in the model selection tasks. We summarize these models in two different fields, unsupervised learning and supervised learning. In the unsupervised task, many researches exploit the DP to estimate the cluster number, such as DP-space [30], temporal subspace clustering model [31], geodesic mixture model [32], sphere mixture model [33] and our previously proposed manifold clustering method [34,35]. For the supervised task, DP is employed to model the underlying data distribution, e.g., infinite mixture of Gaussian processes, DP mixture of generalized linear models (GLMs) and Infinite SVM use the DP to split the input space into a number of subregions and learn an conventional supervised model within each region [36,37]. Unlike the conventional DP based methods, those exploit the DP in the homogenous dataset, our application scenarios are heterogeneous.

2 Proposed approach

In this section, we construct our Bayesian Heterogeneous Adversarial Clustering (BHAC) method, which we divide it into the heterogeneous metric construction method and the heterogeneous clustering method. Then, we derive the corresponding optimization algorithm for the model inference. The main notations are summarized in Table 1.

2.1 Problem formulation

Given the dataset X, the features come from different feature spaces, and the samples belong to $\hat{K}$ different classes. Note that $\hat{K}$ is unknown. In our paper, we consider a special case which can be easily extended to the general problem. That is, our samples come from two different feature space X^s, X^t ($X = \{X^s, X^t\}$), where $X^s = \{x_n^s\}_{n=1}^{N^s}$ and $X^t = \{x_n^t\}_{n=1}^{N^t}$ denote the dataset with the feature s and t, N^s and N^t are the number with feature s and t. For the derivation convenience, another representation $X = \{x_n\}_{n=1}^{N}$ is also used, note that $N = N^s + N^t$, and $X^s \cup X^t = \{x_n\}_{n=1}^{N}$. Our heterogeneous clustering task aims to cluster heterogeneous data X and estimate the cluster number $\hat{K}$.

Table 1 The main notations and descriptions

Notations	Descriptions
$\hat{K}$	Ground truth of the cluster number
X^s	Samples in the s space
X^t	Samples in the t space
X	Samples which ignores the feature space
$\delta_i^s(), \delta_i^t()$	Permutation function defined at x_i^s and x_i^t
T^{st}	Metric measures feature space s and t
T^{ss}, T^{tt}	Metric of feature space s (t) and s (t)
T	Heterogeneous metric
L	Graph Laplacian of T
$Seq()$	Structure order sequence
c_0, u_0, v_0, B_0	Hyper parameter for G_0
Ω	Hidden variables for the BHAC
α	Hyper parameter for DP
$\Psi()$	Digamma function
$DP()$	Dirichlet process (DP)
$GN()$	GAN generative network
$DN()$	GAN discriminative network
K	Maximum cluster number
z_n	Cluster indicator
γ_k, τ_k, ϕ_n	Variational parameters
$\tilde{w}_n$	Laplacian approximation parameters

2.2 Heterogeneous metric construction

Motivation of the heterogeneous metric In order to cluster the heterogeneous dataset, we first construct a metric to measure the similarity between the features in different spaces. We start this metric from an observation that data with different feature spaces shares a same data relationship structure. We illustrate this observation in Fig. 1. This structure may not preserve in the high dimension or other real datasets. But, it motivates us to exploit a relaxed intuition that, when we describe a same pattern with different features, their relationship within a feature space is very similar. For example, let us say class A with three samples which are described with two different features, x_1^t, x_2^t, x_3^t and x_1^s, x_2^s, x_3^s. Then, we know that, if x_1^t is closer to x_2^t than x_3^t, this relationship will pass to feature x_1^s, x_2^s, x_3^s with high probability.

With the above motivations, we derive our heterogeneous metric by making the following definition and assumption (For the debate convenience, we first assume X^s and X^t are the same samples with different features. That is, x_i^s and x_i^t are the same sample with different features):

Definition Structure order and structure order sequence. Given observation samples $X^s = \{x_n^s\}_{n=1}^{N^s}$ and a point x_i^s, we define structure order $x_j^s \geqslant x_l^s$ at point x_i^s as the distance between x_i^s and x_j^s (denoted as $D(x_i^s, x_j^s)$, $D(x_i^s, x_j^s) \geqslant 0$) is less than x_i^s and x_l^s. Then, for every point x_i^s, we can define a structure order sequence $Seq(x_i^s) = \{x^s_{\delta_i^s(n)}\}_{n=1}^{N^s}$ ($\delta_i^s(n)$ is the permutation function), in which, $\forall n > m$, $x_{\delta_i^s(n)} \geq x_{\delta_i^s(m)}$. For

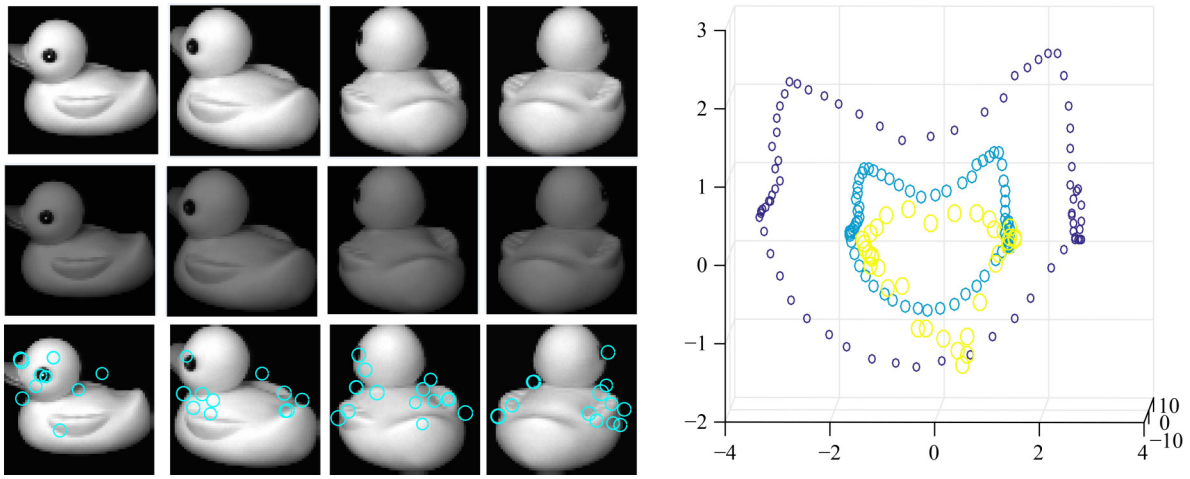

Fig. 1 Illustration of data structure with different features. The 1st row is the original images. The 2nd row shows the images with different light conditions. The 3rd row demonstrates the images with sift feature. Right figure illustrates the feature distribution in the 3D space. From the figure, we know that data with different feature share the same data structure

$X^t = \{x_i^t\}_{i=1}^{N_t}$, we have the same definition for the structure order, structure order sequence and the permutation function $\delta_i^t()$.

Lemma 1 Given the point x_i^s and $Seq(x_i^s)$, we can conclude that $x^s_{\delta_i^s(1)}$ is x_i^s. Same as point x_i^t.

Proof We know that any defined distance $D(x_i^s, x_i^s)$ is 0. Then, $\forall n > 1$, $D(x^s_{\delta^s i(1)}, x^s_{\delta_i(1)}) < D(x^s_{\delta^s i(1)}, x^s_{\delta^s i(n)})$. With the definition, we have $x^s_{\delta_i^s(1)} = x_i^s$. For the point x_i^t, we have the same proof. □

Assumption Given the data sample x_i^s in the feature space s and the corresponding feature x_i^t in feature space t, then, we assume that $\forall j, j \neq i$, structure order sequence $Seq(x_i^s) = \{x^s_{\delta_i^s(n)}\}_{n=1}^{N^s}$, $Seq(x_i^t) = \{x^t_{\delta_i^t(n)}\}_{n=1}^{N^t}$ and $Seq(x_j^t) = \{x^t_{\delta_j^t(n)}\}_{n=1}^{N^t}$ have the probability that, $\forall n > 1$, $p([D(x^s_{\delta_i^s(1)}, x^s_{\delta_i^s(n)}) - D(x^t_{\delta_i^t(1)}, x^t_{\delta_i^t(n)})]^2 < [D(x^s_{\delta_i^s(1)}, x^s_{\delta_i^s(n)}) - D(x^t_{\delta_j^t(1)}, x^t_{\delta_j^t(n)})]^2) = \epsilon$, $\epsilon > 1 - \epsilon$.

Lemma 2 Given the samples x_i^s, x_j^t and a positive integer $M \leqslant \min(N^s, N^t)$, if $\epsilon < M/\min(N^s, N^t)$, $\forall l \neq j$, there are M pairs $[D(x^s_{\delta_i^s(1)}, x^s_{\delta_i^s(m)}) - D(x^t_{\delta_j^t(1)}, x^t_{\delta_j^t(m)})]^2$ is less than $[D(x^s_{\delta_i^s(1)}, x^s_{\delta_i^s(m)}) - D(x^t_{\delta_l^t(1)}, x^t_{\delta_l^t(m)})]^2$. Then, with high probability, x_i^s and x_j^t are same points with different features.

Proof We prove this by calculation and exploiting the assumption. For the convenience, we denote $\min(N^s, N^t)$ as $N_{\min}$. That is, when $\epsilon < M/N_{\min}$, $C_N^M \epsilon^M (1-\epsilon)^{N_{\min}-M}$ is increasing when ϵ is increasing, then, since we assume $\epsilon > 1-\epsilon, C^M_{N_{\min}} \epsilon^M (1-\epsilon)^{N_{\min}-M} > C^M_{N_{\min}} (1-\epsilon)^M \epsilon^{N_{\min}-M}$. Then, x_i^s and x_j^t are same point from different feature spaces with high probability. □

Metric construction In the above debates, we give the motivation that why we construct the metric. That is, the structure between different features is similar. Then, we derive a definition for the structure order sequence construction, and give the Lemma 1 and Lemma 2 to ensure that the constructed structure order sequence could be used to measure the structure similar between different features. From Lemma 2, we can conclude a metric: $\forall i$, given the order sequence $Seq(x_i^s)$, if there exists a point x_j^t with $Seq(x_j^t)$, and M pairs structure order distance $[D(x^s_{\delta_i^s(1)}, x^s_{\delta_i^s(m)}) - D(x^t_{\delta_j^t(1)}, x^t_{\delta_j^t(m)})]^2$ between $Seq(x_i^s)$ and $Seq(x_j^t)$ is the least, x_i^s and x_j^t is the the same sample point with high probability. To find more samples x_j^t belonged to the same sample with high probability, we increase M to N, and select all M samples that the structure order sequence has the least values.

In the above debate, we assume that the X^s and X^t are the same samples with different features. If X^s and X^t are not the same samples, following the previous debate, we know that x_i^s and x_j^t are with the same class at least. Then, given the observation samples $X = \{X^s, X^t\}$, we can achieve the following graph which measures the similarity between samples in X.

$$T = \begin{bmatrix} T^{ss} & T^{st} \\ T^{ts} & T^{tt} \end{bmatrix}, \tag{1}$$

where T^{ss} and T^{tt} are constructed by a standard k-nearest neighbor algorithm to measure the difference between samples within the same feature. T^{ts} and T^{st} are the similarity metrics to measure the difference between samples with the different features. When constructing the order sequence, we compute the distance between each point with the shortest path, which is used to capture the data structure. We summarize our algorithm in Algorithm 1.

2.3 Heterogeneous clustering

After constructing the metric, we now establish our heterogeneous clustering model, in which we exploit the Dirichlet process [30] to estimate the unknown cluster number and assume that the heterogeneous data x_n is generated from a same hidden space denoted as θ_n with probability $F(x|\theta_n)$ (following, we no more distinguish X^s and X^t, and use $X = \{x_n\}_{n=1}^N$ since we have constructed the similarity metric T):

$$G|G_0 \sim DP(G_0, \alpha), \theta_n|G \sim G, x_n \sim F(x|\theta_n). \tag{2}$$

Algorithm 1 Heterogeneous metric construction

Input: Unlabeled dataset X^s and X^t, M;
Output: Metric matrix T;

1: Construct T^{ss} and T^{tt} by k-nearest neighbor algorithm.
2: For every point, construct the structure order sequence $Seq(x_i^s)$ and $Seq(x_j^t)$ with the shortest path.
3: For $n < \min(N^s, N^t), \forall x_i^s \in X^s, x_j^t \in X^t$, $\hat{T}_{i,j}^n = [D(x^s_{\delta_i^s(1)}, x^s_{\delta_i^s(n)}) - D(x^t_{\delta_j^t(1)}, x^t_{\delta_j^t(n)})]^2$;
4: Increasing m from M to N, $T_{i,j}^{st} = 0$, repeat Step 5.
5: Fix x_i^s and find a sample x_j^t. $\forall l \neq j$ if there exists m values $\{\hat{T}_{i,j}^n\}_{n=1}^{\min(N^s,N^t)}$ that are the least among all values in $\{\hat{T}_{i,l}^n\}_{n=1}^{\min(N^s,N^t)}$. Set $T_{i,j}^{st} = \sum_{n=1}^{\min(N^s,N^t)} \hat{T}_{i,j}^n$;

According to the previous study [38], we know that θ_n demonstrates a clustering effect that a new sample x_N is either sampled from a novel class, or extracted from the existing clusters.

$$p(\theta_N|\{\theta_n\}_{n=1}^{N-1}) = A_N \alpha G_0 + A_N \sum_{i=1}^{I} \hat{n}_i \hat{\delta}(i),$$

where $A_N = 1/(\alpha + N - 1)$, $\hat{n}_i$ denotes the θ frequency of occurrence in $\{\theta_n\}_{n=1}^{N-1}$, $\hat{\delta}(j)$ represents the delta function. I denotes the number of unique value in $\{\theta_n\}_{n=1}^{N-1}$. By establishing the generating process with the conventional Bayesian DP framework, we get two issues: (1) Conventional distribution $F(x|\theta_n)$ cannot model the complicated high dimension real dataset. (2) Generated samples x_n cannot assure that samples from the same classes with different feature spaces have the similar representation.

To address the first issue, we exploit the GAN to model the complicated real dataset. For the second issue, we constrain the generation process with the proposed heterogeneous similarity metric. That is, when generating the observation samples, we employ a pseudo feature concept which is generated from the DP process, and is consistent with the heterogeneous similarity metric. With the previous debate, we can derive the following generation model:

1. For $n \in \{n\}_1^{\infty}$, $\theta_n|G \sim G, G|G_0 \sim DP(G_0(\lambda), \alpha)$,
2. For $n \in \{n\}_1^{\infty}$, $\hat{x}_n \sim F(\hat{x}|\theta_n)$,
3. For $n, \hat{n} \in \{n\}_1^{N}$, $h_{n,\hat{n}} \sim N(r_{n,\hat{n}}|T_{n,\hat{n}}(\hat{x}_n - \hat{x}_{\hat{n}})^2, \delta)$,
4. For $n \in \{n\}_1^{N}$, $x_n|\hat{x}_n \sim N(x|GN(\hat{x}_n))$,

where $GN(\hat{x}_i)$ is the GAN generative network. G_0 is the Normal-Wishart distribution with parameter $(\lambda) = \{c_0, u_0, v_0, B_0\}$. $F(\hat{x}|\theta)$ and $N(h_{n,\hat{n}}|T_{n,\hat{n}}(\hat{x}_n - \hat{x}_{\hat{n}})^2, \delta)$ are Gaussian distribution. In order to optimize the GAN, we introduce a discriminative net $D(X)$. When realizing the DP for the model optimization, we exploit stick break process.

$$G|G_0 \sim DP(G_0, \alpha) \Longleftrightarrow \left\{ \begin{array}{l} v_n|\alpha \sim Beta(1, \alpha) \\ \theta_n|G_0(\lambda) \sim G_0(\lambda) \\ G = \sum_{n=1}^{\infty} \pi(v_n) \hat{\delta}_{\theta_n}(\theta_n) \end{array} \right\},$$

where $Beta(1, \alpha)$ is the Beta distribution, $\pi(v_n)$ is defined as $v_n \prod(1 - v_{n-1})$, $\hat{\delta}_{\theta_n}(\theta_n)$ is the indicator function.

2.4 Optimization

In the previous section, we have developed the deep generative model for the heterogeneous datasets. We now derive the corresponding optimization algorithm for the model inference. Given the observation samples generation process, our model can achieve the class indicator by calculating the posterior probability. In order to calculate the posterior probability of the given model, we exploit the variational inference framework, in which we truncate the relation and approximate the posterior probability of the hidden parameters:

$$q(V, \theta, Z, \hat{X}) = \prod_{k=1}^{K-1} q_{\gamma_k}(v_k) \prod_{k=1}^{K} q_{\tau_k}(\theta_k) \prod_{n=1}^{N} q_{\phi_n}(z_n) \prod_{n=1}^{N} q_{\tilde{w}_n}(\hat{x}_n), \quad (3)$$

where $q_{\gamma_k}(v_k)$, $q_{\tau_k}(\theta_k)$, $q_{\phi_n}(z_n)$ are the Beta, Wishart-Normal and categorical distributions with parameter $\gamma_k = \{\gamma_{k,1}, \gamma_{k,2}\}$, $\tau_k = \{c_k, u_k, v_k, B_k\}$ and $\phi_n = \{\phi_{n,k}\}_{k=1}^K$, which are the conjugate distribution of the given Bayesian priors. For $\hat{x}_n$, it is a little different since the GAN is deep neural network. We have no closed form of $q_{\tilde{w}_n}(\hat{x}_n)$. We thus use the Laplacian approximation to the $\hat{x}_n$ and set the distribution $q_{\tilde{w}_n}(\hat{x}_n)$ as a Gaussian distribution with parameter $\tilde{w}_n = \{u_n^x, \Sigma_n^x\}_{n=1}^N$. From the generation constructed, we can also derive the corresponding likelihood probability lower bound given observations.

$$\log \prod\nolimits_{n=1}^{N} p(x_n|\alpha, \lambda) \geqslant \int q(\Omega) \log \frac{p(X, \Omega, \alpha, \lambda)}{q(\Omega)} \mathrm{d}(\Omega), \quad (4)$$

where $\Omega = \{V, \theta, Z, \hat{X}\}$, $q(\Omega)$ is the variational posterior probability, $p(X, \Omega, \alpha, \lambda)$ is the joint probability of the hidden variables and observations.

Update of the γ_k: Differentiating the variational lower bound of the likelihood function, we can derive the following updating rules:

$$\gamma_{k,1} = 1 + \sum_{n=1}^{N} \phi_{n,k}, \gamma_{k,2} = \alpha + \sum_{n=1}^{N} \sum_{j>k} \phi_{n,j}. \quad (5)$$

Update of the τ_k: The updating rule can be achieved by taking the partial derivative and setting the derivative to zero:

$$c_k = c_0 + \sum_{n=1}^{N} \phi_{n,k}, \quad v_k = v_0 + \sum_{n=1}^{N} \phi_{n,k}, \quad (6)$$

$$u_k = \frac{1}{c_k} \Big(c_0 u_0 + \sum_{n=1}^{N} \phi_{n,k} u_n^x \Big), \quad (7)$$

$$B_k^{-1} = \Big(u_k - u_0) c_0 (u_k - u_0\Big)^T + \sum_{n=1}^{N} \phi_{n,k} \Sigma_n^x + B_0^{-1} + \sum_{n=1}^{N} \phi_{n,k} \Big(u_n^x - u_k\Big)\Big(u_n^x - u_k\Big)^T. \quad (8)$$

Update of the ϕ_n: Similar to the τ_k, we have:

$$\log\phi_{n,k} \propto \Psi(\gamma_{k,1}) - \Psi(\gamma_{k,1}+\gamma_{k,2}) + Dc_k^{-1}$$
$$+\sum_{i=1}^{D}\psi\left(\frac{\nu_k+1-i}{2}\right)+\sum\nolimits_{j<k}\left\{\Psi(\gamma_{j,2})-\Psi(\gamma_{j,1}+\gamma_{j,2})\right\}$$
$$+\phi_{n,k}\times\left\{-\nu_k Tr(B_k\Sigma_n^x)-\nu_k(u_n^x-u_k)^T B_k(u_n^x-u_k)\right\}, \quad (9)$$

where $\Psi()$ is a digamma function. $Tr()$ stands for the trace sum of matrix.

Learning adversarial parameters Different from the updating rules used in the variational parameters, since we adopt the generative adversarial network to model the complicated high dimension real dataset, the variational inference can not derive a closed form solution to update some variational and network parameters. We then exploit a sampling based method:

$$\arg\min_{\{u_n^x,\Sigma_n^x\}} E[\log p(x_n|\hat{x}_n)] = \arg\min_{\{u_n^x,\Sigma_n^x\}} E_q[\log F(\hat{x}_n|\theta_n)]$$

$$\min_{\{u_n^y,\Sigma_n^y\}}\max\sum_{s=1}^{S}\left\{E[\log DN(x_n)]+E[\log(1-DN(GN(h_{(n,s)})))]\right\},$$

where $h_{(n,s)}$ is sampled from $q_{\tilde{w}_n}(\hat{x}_n)$, S is the sampling times. For the variational parameters u_n^x, we exploit an auxiliary variable $\hat{h}_n = u_n^x$ and a constraint. Then, we expand the above formulations and alter it into:

$$\min 1/\hat{\sigma}H(\hat{D}-T)H^T - \frac{1}{S}\sum_{s,n=1}^{S,N}\log N(x_n|GN(h_{(n,s)}))$$
$$+\sum_{n,k=1}^{N,K}\left\{\phi_{n,k}(\nu_k(\hat{h}_n-u_k)^T B_k(\hat{h}_n-u_k)+Tr(B_k\Sigma_n^x))\right.$$
$$\left.+\log(\pi)^D-\log|B_k|+Dc_k^{-1}+\sum_{i=1}^{D}\psi\left(\frac{\nu_k+1-i}{2}\right)\right\}$$
$$s.t.\ \hat{H}\hat{H}^T = \mathrm{Diag}(1/\epsilon_1 F_1,\ldots,1/\epsilon_D F_D), \hat{h}_n = u_n^x, \quad (10)$$

where $\hat{D}$ is the sum of diagonal matrix, row or column elements of T. Note that $h_{(n,s)}$ is sampled from $q_{\tilde{w}_n}(\hat{x}_n)$, $q_{\tilde{w}_n}(\hat{x}_n)$ contains the variable u_n^x which should be optimized. We achieve this in the network optimization process. For the derivation convenience, we give a collection $\hat{H} = \{\hat{h}_1,\ldots,\hat{h}_N\}$. Also, we set the constraint as:

$$\sum_{d=1}^{D}F_d = \sum_{n,i,k=1}^{N,D,K}\left\{\psi(\frac{\nu_k+1-i}{2})+\phi_{n,k}\left(\nu_k(\hat{h}_n-u_k)^T B_k(\hat{h}_n-u_k)\right.\right.$$
$$\left.\left.+Tr(B_k\Sigma_n^x)\right)-\log|B_k|+Dc_k^{-1}\right\}.$$

To solve this problem, we utilize the Lagrangian multiplier method. We then can achieve the solution of $\hat{H}$ via L (graph Laplacian of T) eigenvalue decomposition. For the parameter Σ_n^x, we optimize it within the network with the following loss function.

$$\arg\min_{\{u_n^x,\Sigma_n^x\}} E[\log p(x_n|\hat{x}_n)] =$$
$$\min_{\{u_n^y,\Sigma_n^y\}}\max\sum_{s=1}^{S}\left\{E[\log DN(x_n)]+E[\log(1-DN(GN(h_{(n,s)})))]\right\}, \quad (11)$$

where $h_{(n,s)} = u_n^x + \Sigma_n^x\hat{\epsilon}$, $\hat{\epsilon}\sim N(\cdot|0,I)$. We summarize the full algorithm in Algorithm 2 and the network architecture in Fig. 2.

Algorithm 2 Heterogeneous clustering (BHAC)

Input: Unlabeled dataset X;
Output: Cluster indicator $Z = \{z_n\}_{n=1}^N$

1: Construct the heterogeneous metric T with Algorithm. 1.
2: For every point and every k, estimate the following variational parameters.
3: For γ_k, Update it using Eq. (5)
4: For τ_k, Update it using Eqs. (6), (7), and (8)
5: For ϕ_n, Update it using Eq. (9)
6: For Laplacian approximation parameter $\{u_n^x\}_{n=1}^N$, we first estimate $\hat{H}$ with L eigenvalue decomposition.
7: For $\{u_n^x\}_{n=1}^N$, set $\hat{H}$ to $\{u_n^x\}_{n=1}^N$.
8: For Laplacian approximation parameter $\{\Sigma_n^x\}_{n=1}^N$, $\{u_n^x\}_{n=1}^N$ and the GAN parameter, estimate it with Eq. (11).
9: Achieve the cluster indicator z_n by $\arg\max_k \phi_{n,k}$.

2.5 Computational complexity

Before deriving the computational complexity, we assume that our clustering algorithm runs Ti iterations. For the GAN network, we assume that the generator has L_g layers, and each layer has U_i^g input nodes with V_i^g outputs. The discriminator has L_d layers, and each layer has U_i^d input nodes with V_i^d outputs. Generator and discriminator runs for t_g and t_d times.

For the metric, we use the k-nearest neighbor graph and the shortest path algorithm for the metric construction. When computing the shortest path, we exploit the Floyd's algorithm.

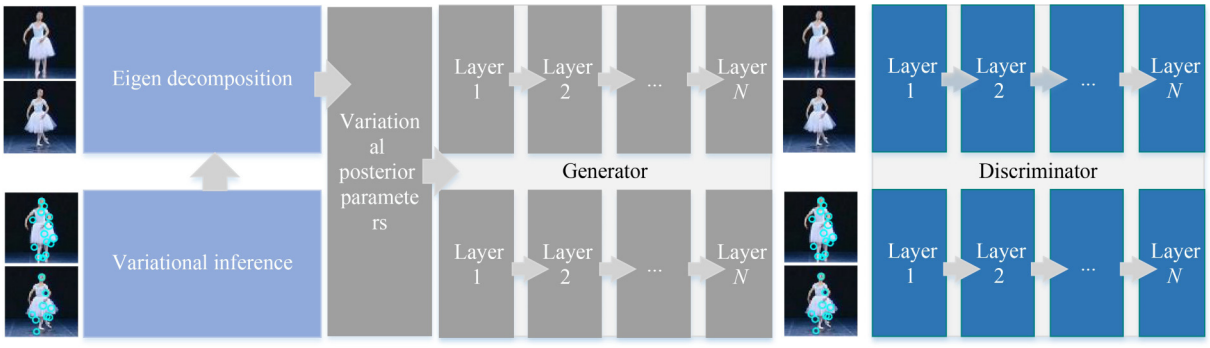

Fig. 2 Overview of the optimization process of the proposed deep Bayesian generative network

Thus, the computational complexity is $O(N^3)$. After computing the shortest path, we exploit the k-nearest neighbor graph algorithm. In the k-nearest neighbor graph algorithm, we use the quick sort algorithm for each sample point. Thus, the computational complexity for the k-nearest neighbor graph is $O(N^2\log_2 N)$. We then deduce the computational complexity for the whole metric algorithm which is $O(N^3 + N^2\log_2 N)$

For the clustering model, the major computation complexity lies on the inverse and determinant of the covariance matrix those need $O(D^3)$. Another major computation complexity is Eq. (8), Eq. (9) and eigenvalue decomposition Eq. (10) those cost $O(ND^2)$ and $O(N^3)$. In addition to these computation costs, the GAN optimization also leads to a computation complexity which should also be considered. That is $O(t_g \times \sum_{i=1}^{L_g} U_i^g \times V_i^g + t_d \times \sum_{i=1}^{L_d} U_i^d \times V_i^d)$. Then, the whole computation complexity will be $O(Ti \times (KD^3 + KND^2) + (N^3 + N^2\log_2 N) + t_g \times \sum_{i=1}^{L_g} U_i^g V_i^g + t_d \times \sum_{i=1}^{L_d} U_i^d V_i^d)$

3 Experiments

In this section, we show the experimental results on synthetic dataset firstly. Then, we validate our model on the real-world dataset, and compare it with some other related clustering methods. The effect of the model parameters is also demonstrated in this section.

Synthetic dataset We first validate our theory on the synthetic dataset (Fig. 3). Heterogeneous data samples are generated from the same dataset with translation transform. In the dataset, we define that the cluster with the same structure as the same class. From the results, we can validate our theoretic analysis, and observe that our model can cluster the heterogeneous dataset, where our method can cluster the class with same structure as the same class. In the following, we validate our method on the real dataset.

Benchmark datasets We use the following datasets for the model theory and performance validation.

COIL20 dataset: COIL20 dataset contains the objects which are rotating on a table. It has 20 objects and each object in each class contains 72 images.

COIL100 dataset: COIL100 is an extended version of COIL20 dataset, which extends the 20 objects to 100 objects, and each object contains 72 images same as COIL20 dataset.

MNIST dataset: MNIST is a well-known dataset of handwritten digits which contains 10 objects and 70000 images.

USPS dataset: USPS dataset is a widely used handwritten dataset which contains 10 classes and 20000 samples. We use a popular subset which contains 9298 images.

BALLET dataset: The BALLET data set contains 44 real video sequences of eight actions collected from a ballet DVD. There are 9594 images in the BALLET image dataset.

Baseline To demonstrate the usefulness of the proposed clustering model, we compare our method with the following two different algorithms. Algorithms those do not need to specify the cluster number: Geodesic Finite Mixture Model (GFMM) [32]. Dirichlet Process Mixture(DPM) model [38]. SCAMS [39]. autoSc-N [40]. Density based Clustering algorithm by Fast Search and Find of Density Peaks (CFSFDP) [41]. In this algorithm, clustering result can be determined by human interaction with the decision graph. A Dirichlet process based linear manifold clustering method, DP-space [30]. Our previously proposed low-rank based clustering method, BLRASC [42]. Three deep learning (ACIDS, ClusterGAN and Spectral-Net) based clustering methods are also considered as the baseline, although they require the cluster number in advance [13,15,43].

For the comparison, three heterogeneous domain adaption methods are also exploited to show the effective of our proposed method. An unsupervised method, DAMA [20], two semi-supervised methods, TNT [44] and CDLS [21]. When exploiting the semi-supervised method, few labeled samples for each class are given, which are used to make the method work.

Experimental setup We evaluate our method on three different data representations: Original images, the noise and rotated images (noise the image with Gaussian distribution and rotate the image 90 degrees), and the SIFT features from the original dataset. To make the baseline algorithms work on images and SIFT feature space, we exploit PCA to project the SIFT and image into 128 dimensions.

Hyper parameters for Bayesian model are set to tiny values to make them affect as little as possible to the model inference and are set as follows: $\alpha = 1$, $u_0 = 0$ and $B_0 = I$, where I is an identity matrix. To initialize the value of the variational parameters, we set them randomly. We use the NMI to measure the clustering accuracy. The hidden dimension of the pseudo features and the maximum cluster number we used in the experiments are summarized in Table 2. M is set at 5. We select the parameters by using some ground-truth labels

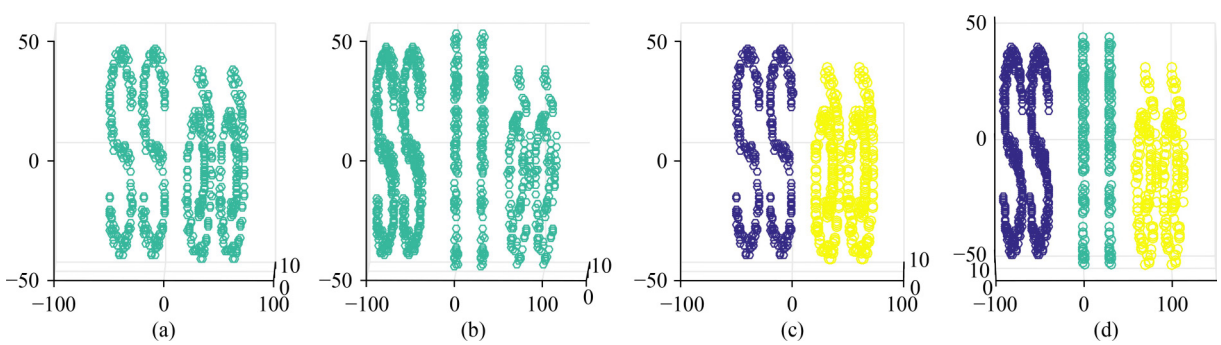

Fig. 3 Illustration of BHAC clustering result on the synthetic dataset. (a) and (b) demonstrate the original dataset without class labels; (c) and (d) are the BHAC results

Table 2 Parameters used in the five real datasets

Data	COIL20	COIL100	BALLET	USPS	MNIST
Feature	Original image+noise rotated image				
D	100	100	130	70	30
K	70	130	60	60	50
Feature	Original image+SIFT				
D	90	90	60	90	40
K	70	130	110	70	40

according to NMI. The labelled data is less than 60% and is selected randomly. The maximum cluster number is selected from the range 30 to 140. The hidden dimension of the pseudo feature is 10 to 140.

GAN model has a backbone architecture. The others are largely altered from this backbone by adding or cutting down some layers. Generator with four layers, each layer has the following kernel number, size and activation function. Layer 1: 128, 3×3, ReLU. Layer 2: 64, 5×5, ReLU. Layer 3: 32, 5×5, ReLU. Layer 4: 1, 5×5, Sigmoid. Discriminator is its inverse. When fitting the SIFT features, we exploit a fully connected network with four layers.

Experimental result We show our experimental results on the real dataset in Table 3. From the table, we know that our method achieves the best on most datasets. We can also observe that our model can not only cluster the heterogeneous data samples, but also can be applied to estimate the cluster number.

From the experiments on MNIST and BALLET dataset (original image+noise rotated images and original image+SIFT features), we know that the deep clustering methods perform better than BHAC, the main reason is that: (1) ACIDS, ClusterGAN and Spectral-Net have the cluster number a priori, which makes them perform better than the BHAC method. Our model has the flexible model size, which means that the model should take the clustering task and the cluster number estimation task as a unified framework. This will make the model optimization much harder, and drop the clustering accuracy. (2) Our model relies on the heterogeneous metric, which is constructed via the data structure assumptions. But real dataset may not always follow the assumptions, this happens especially in the large-scale dataset (MNIST and BALLET has much more samples and clusters) where much more samples will not follow the metric assumptions. Another observation is that, the clustering accuracy drops in the SIFT and original image feature space. The reason may be that SIFT and image are much more complicated heterogeneous clustering task, which makes the GAN model hard to be optimized and some samples may not be consistent with the heterogeneous metric assumption.

In addition to the clustering task, we show another advantage of BHAC over the other clustering models. That is, BHAC is a nature generative clustering model and can generate highly realistic samples. In our experiment, we conduct a series of experiments on the COIL20 and BALLET dataset. Figure 4 illustrates the generated samples. From the figure, we know that our model can fit the complicated real image distribution, which is hard to be tackled by the conventional shallow clustering models. For the running time, we summarize it in Table 4.

Effect of the parameters There are several model parameters affecting the performance of our clustering algorithm (the main parameters affect the algorithm are pseudo feature dimension and maximum cluster number). In

Table 3 Clustering accuracy (NMI) and the estimated cluster number on the real world datasets

Method	COIL20	COIL100	BALLET	USPS	MNIST	COIL20	COIL100	BALLET	USPS	MNIST
Feature	Original image+noise rotated Image					Original image+SIFT+TNT				
BHAC	**0.51**	**0.54**	**0.45**	**0.49**	0.47	**0.61**	**0.55**	0.23	**0.41**	**0.41**
DP-space	0.07	0.02	0.07	0.02	0.01	0.02	0.02	0.01	0.05	0.02
SCAMS	0.26	0.24	0.15	0.07	0.24	0.08	0.03	0.04	0.05	0.11
AutoSC-N	0.22	0.18	0.17	0.18	0.21	0.01	0.12	0.13	0.09	0.13
CFSFDP	0.17	0.21	0.29	0.21	0.34	0.36	0.17	0.19	0.21	0.21
DPM	0.05	0.12	0.03	0.03	0.26	0.06	0.04	0.03	0.07	0.11
GFMM	0.01	0.06	0.04	0.08	0.04	0.04	0.02	0.02	0.06	0.03
BLRASC	0.41	0.03	0.11	0.30	0.37	0.29	0.22	0.27	0.24	0.03
ACIDS	0.32	0.39	0.26	0.32	0.42	0.33	0.13	**0.39**	0.10	0.17
ClusterGAN	0.27	0.17	0.04	0.23	0.47	0.11	0.13	0.01	0.14	0.04
Spectral-Net	0.29	0.32	0.35	0.28	0.39	0.34	0.17	0.36	0.22	0.21
Feature	Original image+noise rotated image+DAMA					Original image+SIFT+CDLS				
DP-space	0.04	0.12	0.02	0.07	0.03	0.06	0.01	0.02	0.02	0.04
SCAMS	0.26	0.25	0.14	0.21	0.27	0.11	0.02	0.02	0.01	0.02
AutoSC-N	0.31	0.37	0.11	0.23	0.31	0.31	0.07	0.06	0.14	0.12
CFSFDP	0.21	0.24	0.17	0.31	0.37	0.39	0.12	0.27	0.21	0.17
DPM	0.17	0.09	0.04	0.06	0.31	0.31	0.02	0.03	0.05	0.13
GFMM	0.12	0.16	0.02	0.06	0.02	0.03	0.01	0.01	0.01	0.04
BLRASC	0.39	0.42	0.27	0.34	0.44	0.37	0.21	0.11	0.17	0.12
ACIDS	0.41	0.44	0.27	0.27	0.51	0.42	0.11	0.36	0.13	0.14
ClusterGAN	0.36	0.32	0.05	0.38	**0.58**	0.07	0.04	0.07	0.08	0.02
Spectral-Net	0.31	0.40	0.22	0.35	0.54	0.34	0.09	0.29	0.12	0.17
Ground truth	20.0	100.0	44.0	10.0	10.0	20.0	100.0	44.0	10.0	10.0
Estimated number	31.2	127.6	60.4	20.2	21.2	25.1	125.2	7.1	30.1	17.1

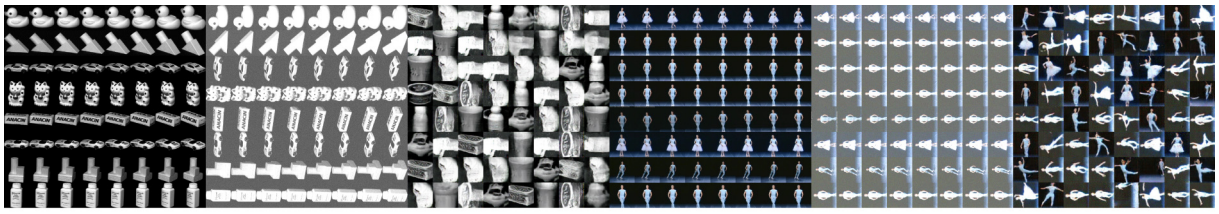

Fig. 4 Illustration of generated images with the COIL20 and BALLET dataset. (a) and (d) demonstrate the samples from original data. (b) and (e) show the samples with the noise and rotation. (c) and (f) are the images generated from our model

Table 4 Running time on the five real datasets (seconds)

Data	COIL20	COIL100	BALLET	USPS	MNIST	COIL20	COIL100	BALLET	USPS	MNIST
Feature	Original image+noise rotated image					Original image+SIFT				
Time	4813.5	22620.6	78160.3	67296.3	222426.7	3210.2	13108.1	34047.5	35915.9	110821.2

this subsection, we conduct some experiments on these parameters to analyze the effect (Fig. 5).

From the figure, we can know that clustering accuracy is decreasing along with the increasing of the maximum cluster number on the COIL20 dataset. The reason may be that large maximum cluster number leads to a much more difficult task for the model to select the right cluster number. For the COIL100 and BALLET dataset, the clustering accuracy is increasing along with the increasing of the maximum cluster number. The reason may be that COIL100 and BALLET dataset have 100 and 44 classes, smaller maximum cluster number doesn't have the capability to model this complicated distribution. The second experiment on the hidden dimension of the pseudo features indicates that high dimension increases the clustering accuracy on the COIL100 and COIL20 dataset. But, this improvement is limited in the BALLET dataset.

For the estimated cluster number, we can observe that, it increases when the maximum cluster number and the hidden dimension is increasing. The reason may be that, (1) the large maximum cluster number enlarges the model capability to find much more clusters; (2) the large hidden dimension indicates that the dataset enlarges its features space, which leads to a result that many samples will form as a outlier in the feature space. This makes the BHAC model increase the cluster number when the hidden space is increasing.

In addition to these effects, we also conduct some experiments on our method to measure the stable of the estimated cluster number (Table 5). We exploit the variance of the estimated cluster number. In our experiments, we test our model on the COIL20, COIL100 and BALLET dataset. There are three parameters (or trick) which could affect the stable of the model, maximum cluster number, the hidden dimension of the pseudo features (D) and the trick we initialize the variational parameter u_k, B_k (That is, initializing the parameter with the estimated variational parameter of $u^{x}_{n\cdot}$). For the comparison, we also report the stable cluster number estimation experimental results. That is: (1) experiments with the same parameter but run at different times. (2) Experiments with different hyper parameter of the DP (we mainly test our method on the α; the other hyper parameters of the DP are the same). From the experiments, we can conclude that: (1) our method with the same parameters or different hyper

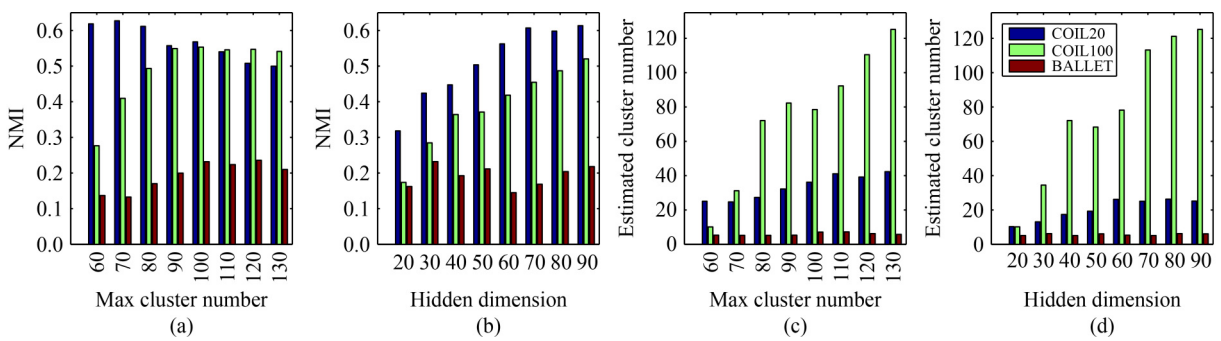

Fig. 5 Illustration of the clustering result of the BHAC parameter effect on the COIL20, COIL100 and BALLET dataset. The 1st two figures (a) and (b) show the clustering accuracy (NMI), while the last two figures (c) and (d) demonstrates the estimated cluster number

Table 5 Variance of the estimated cluster number

Method	COIL20	COIL100	BALLET
Same parameters	1.51	2.12	0.69
Hyper parameter α	1.67	2.02	0.68
Different stretagy	6.55	15.62	3.67
K	7.20	38.33	0.84
D	6.31	41.32	0.54

parameters of the DP will lead to a stable cluster number estimation. (2) The maximum cluster number, dimension of the hidden variable t_i and the strategy that we initialize the variational parameters u_k and B_k can affect the stable of the estimated cluster number.

4 Conclusion

In this paper, a deep heterogeneous clustering problem with unknown cluster number has been studied. We first construct a heterogeneous similarity metric to measure the difference between different features. Then, a hierarchical Bayesian deep generative model has been proposed to handle the deep clustering problem with unknown cluster number. Finally, we derive an efficient optimization method for the model inference and parameter estimation. Experimental results on different synthetic and real datasets validate our theoretic analysis, and show the effectiveness of our method.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 62006131, 62071260), the National Natural Science Foundation of Zhejiang Province (LQ21F020009, LQ18F020001).

References

1. Jiang Z, Zheng Y, Tan H, Tang B, Zhou H. Variational deep embedding: an unsupervised and generative approach to clustering. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. 2017, 1965–1972
2. Bhattacharjee P, Mitra P. A survey of density based clustering algorithms. Frontiers of Computer Science, 2021, 15(1): 151308
3. Xue H, Li S, Chen X, Wang Y. A maximum margin clustering algorithm based on indefinite kernels. Frontiers of Computer Science, 2019, 13(4): 813–827
4. Ghasedi K, Wang X, Deng C, Huang H. Balanced self-paced learning for generative adversarial clustering network. In: Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, 4386–4395
5. Wen J, Zhang Z, Xu Y, Zhang B, Fei L, Xie G S. CDIMC-net: cognitive deep incomplete multi-view clustering network. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence. 2021, 447
6. Xie J, Girshick R, Farhadi A. Unsupervised deep embedding for clustering analysis. In: Proceedings of the 3rd International Conference on Machine Learning. 2016, 478–487
7. Zhou P, Hou Y, Feng J. Deep adversarial subspace clustering. In: Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018, 1596–1604
8. Peng X, Xiao S, Feng J, Yau W Y, Yi Z. Deep subspace clustering with sparsity prior. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence. 2016, 1925–1931
9. Guo X, Gao L, Liu X, Yin J. Improved deep embedded clustering with local structure preservation. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. 2017, 1753–1759
10. Ji P, Zhang T, Li H, Salzmann M, Reid I. Deep subspace clustering networks. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017, 23–32
11. Yu Y, Zhou W J. Mixture of GANs for clustering. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. 2018, 3047–3053
12. Yang X, Deng C, Zheng F, Yan J, Liu W. Deep spectral clustering using dual autoencoder network. In: Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, 4061–4070
13. Shaham U, Stanton K P, Li H, Basri R, Nadler B, Kluger Y. SpectralNet: spectral clustering using deep neural networks. In: Proceedings of the 6th International Conference on Learning Representation. 2018
14. Cheng J, Wang Q, Tao Z, Xie D, Gao Q. Multi-view attribute graph convolution networks for clustering. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence. 2021, 411
15. Menapace W, Lathuilière S, Ricci E. Learning to cluster under domain shift. In: Proceedings of the 16th European Conference on Computer Vision. 2020, 736–752
16. Tapaswi M, Law M T, Fidler S. Video face clustering with unknown number of clusters. In: Proceedings of 2019 IEEE/CVF International Conference on Computer Vision. 2019, 5026–5035
17. Yang L, Zhan X, Chen D, Yan J, Boy C C, Lin D. Learning to cluster faces on an affinity graph. In: Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, 2293–2301
18. Li J, Lu K, Huang Z, Zhu L, Shen H T. Heterogeneous domain adaptation through progressive alignment. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(5): 1381–1391
19. Yang S, Song G, Jin Y, Du L. Domain adaptive classification on heterogeneous information networks. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence. 2021, 196
20. Wang C, Mahadevan S. Heterogeneous domain adaptation using manifold alignment. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence. 2011, 1541–1546
21. Tsai Y H H, Yeh Y R, Wang Y C F. Learning cross-domain landmarks for heterogeneous domain adaptation. In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. 2016, 5081–5090
22. Yeh Y R, Huang C H, Wang Y C F. Heterogeneous domain adaptation and classification by exploiting the correlation subspace. IEEE Transactions on Image Processing, 2014, 23(5): 2009–2018
23. Wang M, Deng W. Deep visual domain adaptation: a survey. Neurocomputing, 2018, 312: 135–153
24. Day O, Khoshgoftaar T M. A survey on heterogeneous transfer learning. Journal of Big Data, 2017, 4(1): 29
25. Wang H, Yang Y, Liu B. GMC: graph-based multi-view clustering. IEEE Transactions on Knowledge and Data Engineering, 2020, 32(6): 1116–1129
26. Shi S, Nie F, Wang R, Li X. Fast multi-view clustering via prototype graph. IEEE Transactions on Knowledge and Data Engineering, 2021, doi: 10.1109/TKDE.2021.3078728
27. Li Z, Nie F, Chang X, Yang Y, Zhang C, Sebe N. Dynamic affinity graph construction for spectral clustering using multiple features. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(12): 6323–6332
28. Yin J, Sun S. Incomplete multi-view clustering with reconstructed views. IEEE Transactions on Knowledge and Data Engineering, 2021, doi: 10.1109/TKDE.2021.3112114
29. Li L, Wan Z, He H. Incomplete multi-view clustering with joint partition and graph learning. IEEE Transactions on Knowledge and Data Engineering, 2021, doi: 10.1109/TKDE.2021.3082470
30. Wang Y, Zhu J. DP-space: Bayesian nonparametric subspace clustering with small-variance asymptotics. In: Proceedings of the 32nd International Conference on Machine Learning. 2015, 862–870
31. Gholami B, Pavlovic V. Probabilistic temporal subspace clustering. In: Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. 2017, 4313–4322
32. Simo-Serra E, Torras C, Moreno-Noguer F. 3D human pose tracking priors using geodesic mixture models. International Journal of

Computer Vision, 2017, 122(2): 388–408
33. Straub J, Freifeld O, Rosman G, Leonard J J, Fisher J W. The Manhattan frame model—Manhattan world inference in the space of surface normals. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(1): 235–249
34. Ye X, Zhao J. Multi-manifold clustering: a graph-constrained deep nonparametric method. Pattern Recognition, 2019, 93: 215–227
35. Ye X, Zhao J, Zhang L, Guo L. A nonparametric deep generative model for multimanifold clustering. IEEE Transactions on Cybernetics, 2019, 49(7): 2664–2677
36. Hannah L A, Blei D M, Powell W B. Dirichlet process mixtures of generalized linear models. The Journal of Machine Learning Research, 2011, 12: 1923–1953
37. Wang Y, Zhu J. Small-variance asymptotics for Dirichlet process mixtures of SVMs. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence. 2014, 2135–2141
38. Blei D M, Jordan M I. Variational inference for Dirichlet process mixtures. Bayesian Analysis, 2006, 1(1): 121–143
39. Li Z, Cheong L F, Yang S, Toh K C. Simultaneous clustering and model selection: algorithm, theory and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(8): 1964–1978
40. Liang J, Yang J, Cheng M M, Rosin P L, Wang L. Simultaneous subspace clustering and cluster number estimating based on triplet relationship. IEEE Transactions on Image Processing, 2019, 28(8): 3973–3985
41. Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science, 2014, 344(6191): 1492–1496
42. Ye X L, Zhao J, Chen Y, Guo L J. Bayesian adversarial spectral clustering with unknown cluster number. IEEE Transactions on Image Processing, 2020, 29: 8506–8518
43. Mukherjee S, Asnani H, Lin E, Kannan S. ClusterGAN: latent space clustering in generative adversarial networks. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence. 2019, 4610–4617
44. Chen W Y, Hsu T M H, Tsai Y H H, Wang Y C F, Chen M S. Transfer neural trees for heterogeneous domain adaptation. In: Proceedings of the 14th European Conference on Computer Vision. 2016, 399–414

Xulun Ye received the MSc and PhD degrees from Ningbo University, China in 2016 and 2019, respectively, where he is currently a lecturer. His research interests include Bayesian learning, deep learning, nonparametric clustering and convex analysis.

Jieyu Zhao received the BS and MSc degrees from Zhejiang University, China and the PhD degree from Royal Holloway University of London, UK in 1985, 1988 and 1995 respectively. He is currently a full professor at Ningbo University, China. His research interests include deep learning, and computer vision.