Hierarchically Clustered Representation Learning

Su-Jin Shin,[1] **Kyungwoo Song,**[2] **Il-Chul Moon***[2]
[1]Institute of Defense Advanced Technology Research, Agency for Defense Development
[2]Department of Industrial and Systems Engineering, KAIST
[1,2]Yuseong-gu, Daejeon, Republic of Korea
sujinshin@add.re.kr, {gtshs2, icmoon}@kaist.ac.kr

Abstract

The joint optimization of representation learning and clustering in the embedding space has experienced a breakthrough in recent years. In spite of the advance, clustering with representation learning has been limited to flat-level categories, which often involves cohesive clustering with a focus on instance relations. To overcome the limitations of flat clustering, we introduce *hierarchically-clustered* representation learning (HCRL), which simultaneously optimizes representation learning and hierarchical clustering in the embedding space. Compared with a few prior works, HCRL firstly attempts to consider a generation of deep embeddings from every component of the hierarchy, not just leaf components. In addition to obtaining hierarchically clustered embeddings, we can reconstruct data by the various abstraction levels, infer the intrinsic hierarchical structure, and learn the level-proportion features. We conducted evaluations with image and text domains, and our quantitative analyses showed competent likelihoods and the best accuracies compared with the baselines.

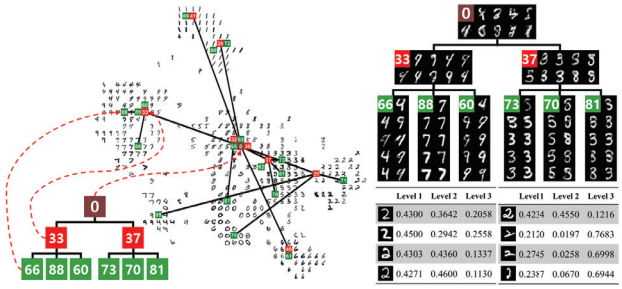


Figure 1: Example of hierarchically clustered embeddings on MNIST with three levels of hierarchy (left), the generated digits from the hierarchical Gaussian mixture components (top right), and the extracted level proportion features (bottom right). We marked the mean of a Gaussian mixture component with the colored square, and the digit written inside the square refers to the unique index of the mixture component.

Introduction

Clustering is one of the most traditional and frequently used machine learning tasks. Clustering models are designed to represent intrinsic data structures, such as latent Dirichlet allocation (Blei, Ng, and Jordan 2003). The recent development of *representation learning* has contributed to generalizing model feature engineering, which also enhances data representation (Bengio, Courville, and Vincent 2013). Therefore, representation learning has been merged into the clustering models, e.g., variational deep embedding (VaDE) (Jiang et al. 2017). Besides merging representation learning and clustering, another critical line of research is structuring the clustering result, e.g., hierarchical clustering.

Autoencoder (Rumelhart, Hinton, and Williams 1985) is a typical neural network for unsupervised representation learning and achieves a non-linear mapping from a input space to a embedding space by minimizing reconstruction errors. To turn the embeddings into random variables, a variational autoencoder (VAE) (Kingma and Welling 2014) places a Gaussian prior on the embeddings. The autoencoder, whether it is probabilistic or not, has a limitation in reflecting the intrinsic hierarchical structure of data. For instance, VAE assuming a single Gaussian prior needs to be expanded to suggest an elaborate clustering structure.

Due to the limitations of modeling the cluster structure with autoencoders, prior works combine the autoencoder and the clustering algorithm. While some early cases pipeline just two models, e.g., (Huang et al. 2014), a typical merging approach is to model an additional loss, such as a clustering loss, in the autoencoders (Xie, Girshick, and Farhadi 2016; Guo et al. 2017; Yang et al. 2017; Nalisnick, Hertel, and Smyth 2016; Chu and Cai 2017; Jiang et al. 2017). These suggestions exhibit gains from unifying the encoding and the clustering, yet they remain at the parametric and flat-structured clustering. A more recent development releases the previous constraints by using the nonparametric Bayesian approach. For example, the infinite mixture of VAEs (IMVAE) (Abbasnejad, Dick, and van den Hengel 2017) explores the infinite space for VAE mixtures by looking for an adequate embedding space through sampling, such as the Chinese restaurant process (CRP). Whereas

*Corresponding author
Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

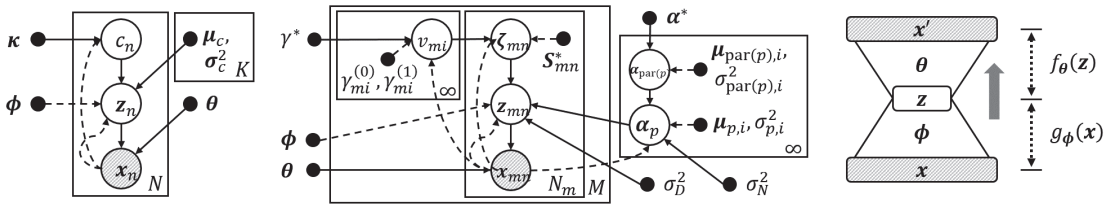


Figure 2: Graphical representation of VaDE (Jiang et al. 2017) (left), VAE-nCRP (Goyal et al. 2017) (center), and neural architecture of both models (right). In the graphical representation, the white/shaded circles represent latent/observed variables. The black dots indicate hyper or variational parameters. The solid lines represent a generative model, and dashed lines represent a variational approximation. A rectangle box means a repetition for the number of times denoted by the bottom right of the box.

IMVAE remains at the flat-structured clustering, VAE-nested CRP (VAE-nCRP) (Goyal et al. 2017) captures a more complex structure, i.e., a hierarchical structure of the data, by adopting the nested Chinese restaurant process (nCRP) prior (Griffiths et al. 2004) into the cluster assignment of the Gaussian mixture model.

Hierarchical mixture density estimation (Vasconcelos and Lippman 1999), where all internal and leaf components are directly modeled to generate data, is a flexible framework for hierarchical mixture modeling, such as hierarchical topic modeling (Mimno, Li, and McCallum 2007; Griffiths et al. 2004), with regard to the learning of the internal components. This paper proposes hierarchically clustered representation learning (HCRL) that is a joint model of 1) nonparametric Bayesian hierarchical clustering, and 2) representation learning with neural networks. HCRL extends a previous work on merging flat clustering and representation learning, i.e., VaDE, by incorporating inter-cluster relation modelings.

Specifically, HCRL jointly optimizes soft-divisive hierarchical clustering in an embedding space from VAE via two mechanisms. First, HCRL includes a hierarchical-versioned Gaussian mixture model (HGMM) with a mixture of hierarchically organized Gaussian distributions. Then, HCRL sets the prior of embeddings by adopting the generative processes of HGMM. Second, to handle a dynamic hierarchy structure dealing with the clusters of unequal sizes, we explore the infinite hierarchy space by exploiting an nCRP prior. These mechanisms are fused as a unified objective function; this is done rather than concatenating the two distinct models of clustering and autoencoding.

We developed two variations of HCRL, called HCRL1 and HCRL2, where HCRL2 extends HCRL1 by the flexible modeling on the level proportion. The quantitative evaluations focus on density estimation quality and hierarchical clustering accuracy, which shows that HCRL2 have competent likelihoods and the best accuracies compared with the baselines. When we observe our results qualitatively, we visualize 1) the hierarchical clusterings, 2) the embeddings under the hierarchy modeling, and 3) the generated images from each Gaussian mixture component, as shown in Figure 1. These experiments were conducted by crossing the data domains of texts and images, so our benchmark datasets include MNIST, CIFAR-100, RCV1_v2, and 20Newsgroups.

Preliminaries

Variational Deep Embedding

VaDE, see Figure 2, is a combination of representation learning and mixture modeling (Jiang et al. 2017). VaDE defines the model parameters of $\boldsymbol{\kappa}$, $\boldsymbol{\mu}_{1:K}$, and $\boldsymbol{\sigma}^2_{1:K}$, which are a proportion, means, and covariances of mixture components, respectively. VaDE uses the Gaussian mixture model (GMM) as the prior, whereas VAE assumes a single standard Gaussian distribution on embeddings. VaDE uses an amortized inference as VAE, with a generative and inference networks; $\mathcal{L}(\boldsymbol{x})$ in Equation 1 denotes the evidence lower bound (ELBO), which is the lower bound on the log likelihood.

$$\log p(\boldsymbol{x}) \geq \mathcal{L}(\boldsymbol{x}) = \mathbb{E}_q\left[\log \frac{p(\boldsymbol{c}, \boldsymbol{z}, \boldsymbol{x})}{q(\boldsymbol{c}, \boldsymbol{z}|\boldsymbol{x})}\right]$$
$$= \mathbb{E}_q\left[\log \prod_{c=1}^{K} \frac{\kappa_c \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2 \boldsymbol{I}_J)}{p(c|\boldsymbol{z})\mathcal{N}(\boldsymbol{z}|\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\sigma}}^2 \boldsymbol{I}_J)} + \log p(\boldsymbol{x}|\boldsymbol{z})\right]. \quad (1)$$

Variational Autoencoder nested Chinese Restaurant Process

VAE-nCRP uses the nonparametric Bayesian prior for learning tree-based hierarchies, the nCRP (Griffiths et al. 2004), so the representation could be hierarchically organized. The nCRP prior is the nested version of CRP, and a nonparametric Bayesian prior for learning a tree structure from data providing the distribution over hierarchical partitions, i.e., defines the distributions over children components for each parent component, recursively in a top-down way. The variational inference of the nCRP can be formalized by the nested stick-breaking construction (Wang and Blei 2009), which is also kept in the VAE setting. The weight, π_i, for the i-th node follows the Griffiths-Engen-McCloskey (GEM) distribution (Pitman and others 2002), where π_i is constructed as $\pi_i = v_i \prod_{j=1}^{i-1}(1 - v_j), v_i \sim \text{Beta}(1, \gamma)$ by a stick-breaking process. Since the nCRP provides the ELBO with the nested stick-breaking process, VAE-nCRP has a unified ELBO of VAE and the nCRP as Equation 2.

Given the ELBO of VAE-nCRP, we recognized the potential improvements. First, term (3.1) is for modeling the hierarchical relationship among clusters, i.e., each child is generated from its parent. VAE-nCRP trade-off is the direct dependency modeling among clusters against the mean-field

approximation. This modeling may reveal that the higher clusters in the hierarchy are more difficult to train. Second, in term (3.2), leaf mixture components generate embeddings, which implies that only leaf clusters have direct summarization ability for sub-populations. Additionally, in term (3.2), variance parameter σ_D^2 is modeled as the hyperparameter shared by all clusters. In other words, only with J-dimensional parameters, $\boldsymbol{\alpha}$, for the leaf mixture components, the local density modeling without variance parameters has a critical disadvantage.

$$\mathcal{L}(\boldsymbol{x}) = \mathbb{E}_q\Bigg[\log\frac{p(\boldsymbol{v})}{q(\boldsymbol{v}|\boldsymbol{x})} + \log p(\boldsymbol{x}|\boldsymbol{z}) + \log\Bigg\{\frac{p(\boldsymbol{\zeta}|\boldsymbol{v})}{q(\boldsymbol{\zeta}|\boldsymbol{x})} \underbrace{\frac{p(\boldsymbol{\alpha}_{\text{par}(p)}|\boldsymbol{\alpha}^*)p(\boldsymbol{\alpha}_p|\boldsymbol{\alpha}_{\text{par}(p)},\sigma_N^2)}{q(\boldsymbol{\alpha}_p,\boldsymbol{\alpha}_{\text{par}(p)}|\boldsymbol{x})}}_{(3.1)} \underbrace{\frac{p(\boldsymbol{z}|\boldsymbol{\alpha}_p,\boldsymbol{\zeta},\sigma_D^2)}{q(\boldsymbol{z}|\boldsymbol{x})}}_{(3.2)}\Bigg\}\Bigg]. \quad (2)$$

For all of these weaknesses, we were able to compensate with the level proportion modeling and HGMM prior. The level assignment generated from the level proportion allows a data instance to select among all mixture components. We do not need direct dependency modeling between the parents and their children because all internal mixture components also generate embeddings.

Proposed Models

Generative Process

We developed two models for the hierarchically clustered representation learning; HCRL1 and HCRL2. The generative processes of the presented models resemble the generative process of hierarchical clusterings, such as the hierarchical latent Dirichlet allocation (Griffiths et al. 2004). In detail, the generative process departs from selecting a path $\boldsymbol{\zeta}$, from the nCRP prior (Phase 1). Then, we sample a level proportion (Phase 2) and a level, l (Phase 3), from the sampled level proportion to find the mixture component in the path, and this component of ζ_l provides the Gaussian distribution for the latent representation (Phase 4). Finally, the latent representation is exploited to generate an observed datapoint (Phase 5). The first subfigure of Figure 3 depicts the generative process with the specific notations.

The level proportion of Phase 2 is commonly modeled as the group-specific variable in the topic modeling. To adapt the level proportion for our non-grouped setting, we considered two modeling assumptions on the level proportion: 1) globally defined the level proportion which is shared by all data instances, which characterizes HCRL1, and 2) locally defined, i.e., data-specific level proportion, which is a distinction of HCRL2 from HCRL1. Similar to the latter assumption, several recently proposed models also define a data-specific mixture membership over the mixture components (Zhang et al. 2018; Ji et al. 2016).

The below formulas are the generative process of HCRL2 with its density functions, where the level proportion is generated by a data instance. In addition, Figure 3 illustrates graphical representations of HCRL1 and HCRL2, respectively, and the graphical representations are corresponding to the described generative process. The generative process also presents our formalization of our prior distributions, denoted as $p(\cdot)$, and variational distributions, denoted as $q(\cdot)$, by generation phases. The variational distributions are used for the mean-field variational inference (Jordan et al. 1999) as detailed in Section .

1. Choose a path $\boldsymbol{\zeta} \sim \text{nCRP}(\boldsymbol{\zeta}|\gamma)$
   - $p(\boldsymbol{\zeta}) = \prod_{l=1}^{L}\pi_{1,\zeta_2,\ldots,\zeta_l}$ where $\pi_{1,\zeta_2,\ldots,\zeta_l} = \prod_{l'=1}^{l}\{v_{1,\zeta_2,\ldots,\zeta_{l'}}(\prod_{j=1}^{\zeta_{l'}-1}(1-v_{1,\zeta_2,\ldots,j}))\}$
   - $q(\boldsymbol{\zeta}|\boldsymbol{x}) \propto S_{\overline{\boldsymbol{\zeta}}} \triangleq \sum_{\boldsymbol{\zeta}\in\text{child}(\overline{\boldsymbol{\zeta}})} S_{\boldsymbol{\zeta}}$
2. Choose a level proportion $\boldsymbol{\eta} \sim \text{Dirichlet}(\boldsymbol{\eta}|\boldsymbol{\alpha})$
   - $p(\boldsymbol{\eta}) = \text{Dirichlet}(\boldsymbol{\eta}|\boldsymbol{\alpha})$
   - $q_{\boldsymbol{\phi}_{\boldsymbol{\eta}}}(\boldsymbol{\eta}|\boldsymbol{x}) = \text{Dirichlet}(\boldsymbol{\eta}|\widetilde{\boldsymbol{\alpha}})$
     $\approx \text{LogisticNormal}(\boldsymbol{\eta}|\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\eta}}, \widetilde{\boldsymbol{\sigma}}_{\boldsymbol{\eta}}^2\boldsymbol{I}_L)$
     where $[\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\eta}}; \log\widetilde{\boldsymbol{\sigma}}_{\boldsymbol{\eta}}^2] = g_{\boldsymbol{\phi}_{\boldsymbol{\eta}}}(\boldsymbol{x})$,
     $\widetilde{\alpha}_l = \frac{1}{\widetilde{\sigma}_{\eta_l}^2}(1-\frac{2}{L}+\frac{e^{-\widetilde{\mu}_{\eta_l}}}{L^2}\sum_{l'}e^{-\widetilde{\mu}_{\eta_{l'}}})$
3. Choose a level $l \sim \text{Multinomial}(l|\boldsymbol{\eta})$
   - $p(l) = \text{Multinomial}(\boldsymbol{\eta})$
   - $q(l|\boldsymbol{x}) = \text{Multinomial}(l|\boldsymbol{\omega})$
     where $\omega_l \propto \exp\Big\{\sum_{\boldsymbol{\zeta}} S_{\boldsymbol{\zeta}}\Big(\sum_{j=1}^{J} -\frac{1}{2}\log(2\pi\sigma_{\zeta_l,j}^2) - \frac{\widetilde{\sigma}_{z_j}^2}{2\sigma_{\zeta_l,j}^2} - \frac{(\widetilde{\mu}_{z_j}-\mu_{\zeta_l,j})^2}{2\sigma_{\zeta_l,j}^2}\Big) + \psi(\widetilde{\alpha}_l) - \psi(\widetilde{\alpha}_0)\Big\}$
4. Choose a latent representation $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_{\zeta_l}, \boldsymbol{\sigma}_{\zeta_l}^2\boldsymbol{I}_J)$
   - $p(\boldsymbol{z}) = \sum_{\boldsymbol{\zeta},l} p(\boldsymbol{\zeta}|\gamma)\cdot\eta_l\cdot\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_{\zeta_l}, \boldsymbol{\sigma}_{\zeta_l}^2\boldsymbol{I}_J)$
   - $q_{\boldsymbol{\phi}_{\boldsymbol{z}}}(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}|\widetilde{\boldsymbol{\mu}}_{\boldsymbol{z}}, \widetilde{\boldsymbol{\sigma}}_{\boldsymbol{z}}^2\boldsymbol{I}_J)$
     where $[\widetilde{\boldsymbol{\mu}}_{\boldsymbol{z}}; \log\widetilde{\boldsymbol{\sigma}}_{\boldsymbol{z}}^2] = g_{\boldsymbol{\phi}_{\boldsymbol{z}}}(\boldsymbol{x})$
5. Choose an observed datapoint $\boldsymbol{x} \sim \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\sigma}_{\boldsymbol{x}}^2\boldsymbol{I}_D\right)$
   where $[\boldsymbol{\mu}_{\boldsymbol{x}}; \log\boldsymbol{\sigma}_{\boldsymbol{x}}^2] = f_{\boldsymbol{\theta}}(\boldsymbol{z})$[1]

Neural Architecture

The discrepancy in prior assumptions on the level assignment leads to the different neural architectures. The neural architecture of HCRL1 is a standard variational autoencoder, while the neural architecture of HCRL2 consists of two probabilistic encoders on $\boldsymbol{z}$ and $\boldsymbol{\eta}$, and one probabilistic decoder on $\boldsymbol{z}$ as shown in the right part of Figure 3. We designed the probabilistic encoder on $\boldsymbol{\eta}$ for inferring the variational posterior of data-specific level proportion. The unbalanced architecture originates from our modeling assumption of $p(\boldsymbol{x}|\boldsymbol{z})$, not $p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\eta})$.

One may be puzzled by the lack of the generative network of $\boldsymbol{\eta}$, but $\boldsymbol{\eta}$ is used for the hierarchy construction in the nCRP that is a part of the previous section. In detail, $\boldsymbol{\eta}$ is a random variable of the level proportion in Phase 2 of the generative process. The sampling of $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ reflects in the

[1]We introduce the sample distribution for the real-valued data instances, and supplementary material Section 6 provides the binary case as well, which we use for MNIST. The supplementary material is available at https://github.com/sujin6003/HCRL.

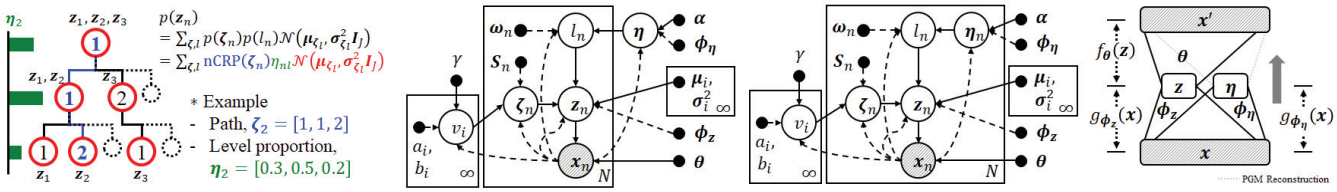


Figure 3: A simple depiction (far left) of the key notations, where each numbered circle refers to the corresponding Gaussian mixture component. The graphical representation of HCRL1 (center left), the graphical representation of HCRL2 (center right), and the neural architecture (far right) of our proposed model, HCRL2. The neural architecture of HCRL2 consists of two probabilistic encoder networks, g_{ϕ_η} and g_{ϕ_z}, and one probabilistic decoder network, f_θ.

selecting a Gaussian mixture component in Phase 4, and the latent vector $\boldsymbol{z}$ becomes an indicator of a data instance, $\boldsymbol{x}$. Therefore, the sampling of $\boldsymbol{\eta}$ from the neural network is linked to the probabilistic modeling of $\boldsymbol{x}$, so the probabilistic model substitutes for creating a generative network from $\boldsymbol{\eta}$ to $\boldsymbol{x}$.

Considering $\boldsymbol{\eta}$ in HCRL, the inference network is given, but the generative network was replaced by the generative process of the graphical model. If we imagine a balanced structure, then the generative process needs to be fully described by the neural network, but the complex interaction within the hierarchy makes a complex neural network structure. Therefore, the neural network structure in Figure 3 may disguise that the structure misses the reconstruction learning on $\boldsymbol{\eta}$, but the reconstruction has been reflected in the probabilistic graphical model (PGM) side of learning. This is also a difference between (VaDE, VAE-nCRP) and HCRL because VaDE and VAE-nCRP adhere to the balanced autoencoder structure. We call this reconstruction process, which is inherently a generative process of the traditional PGM, *PGM reconstruction* (see the decoding neural network part of Figure 3).

Mean-Field Variational Inference

The formal specification can be a factorized probabilistic model as Equation 3, which is based on HCRL2. In the case of HCRL1, $\boldsymbol{\eta}_n$ should be changed to $\boldsymbol{\eta}$ and be placed outside the product over n.

$$p(\boldsymbol{\Phi}, \boldsymbol{x}) = \prod_{j \notin \mathcal{M}_T} p(v_j|\gamma) \times \prod_{i \in \mathcal{M}_T} p(v_i|\gamma) \times \prod_{n=1}^{N} p(\boldsymbol{\zeta}_n|\boldsymbol{v}) p(\boldsymbol{\eta}_n|\boldsymbol{\alpha}) p(l_n|\boldsymbol{\eta}_n) p(\boldsymbol{z}_n|\boldsymbol{\zeta}_n, l_n) p_{\boldsymbol{\theta}}(\boldsymbol{x}_n|\boldsymbol{z}_n). \tag{3}$$

where $\boldsymbol{\Phi} = \{\boldsymbol{v}, \boldsymbol{\zeta}, \boldsymbol{\eta}, \boldsymbol{l}, \boldsymbol{z}\}$ denotes the set of latent variables, $\mathcal{M}_T$ denotes the set of all nodes in tree T, and N is the total number of data instances. The proportion and assignment on the mixture components for the n-th data instance are modeled by $\boldsymbol{\zeta}_n$ as a path assignment; $\boldsymbol{\eta}_n$ as a level proportion; and l_n as a level assignment. v is a Beta draw used in the stick-breaking construction. We assume that the variational distributions of HCRL2 are as Equation 4 by the mean-field approximation. In HCRL1, we also assume the mean-field variational distributions, and therefore, $\boldsymbol{\eta}_n$ should be replaced by $\boldsymbol{\eta}$ and be outside the product over n.

$$q(\boldsymbol{\Phi}|\boldsymbol{x}) = \prod_{j \notin \mathcal{M}_T} p(v_j|\gamma) \times \prod_{i \in \mathcal{M}_T} q(v_i|a_i, b_i) \times \prod_{n=1}^{N} q(\boldsymbol{\zeta}_n|\boldsymbol{x}_n) q_{\boldsymbol{\phi}_\eta}(\boldsymbol{\eta}_n|\boldsymbol{x}_n) q(l_n|\boldsymbol{\omega}_n, \boldsymbol{x}_n) q_{\boldsymbol{\phi}_z}(\boldsymbol{z}_n|\boldsymbol{x}_n). \tag{4}$$

where a_i, b_i are the parameters of the Beta distribution that was used for the stick-breaking process of nCRP. $q_{\boldsymbol{\phi}_\eta}(\boldsymbol{\eta}_n|\boldsymbol{x}_n)$ and $q_{\boldsymbol{\phi}_z}(\boldsymbol{z}_n|\boldsymbol{x}_n)$ should be noted because these two variational distributions follow the amortized inference of VAE. $q(\boldsymbol{\zeta}|\boldsymbol{x}) \propto S_{\overline{\boldsymbol{\zeta}}} \triangleq \sum_{\boldsymbol{\zeta} \in \text{child}(\overline{\boldsymbol{\zeta}})} S_{\boldsymbol{\zeta}}$ is the variational distribution over path $\boldsymbol{\zeta}$, where $\text{child}(\overline{\boldsymbol{\zeta}})$ means the set of all full paths that are not in T but include $\overline{\boldsymbol{\zeta}}$ as a sub path. Because we specified both generative and variational distributions, we define the ELBO of HCRL2, $\mathcal{L} = \mathbb{E}_q \left[\log \frac{p(\boldsymbol{\Phi}, \boldsymbol{x})}{q(\boldsymbol{\Phi}|\boldsymbol{x})} \right]$, in Equation 5. Supplementary material Section 6 enumerates the full derivation in detail. We report that the Laplace approximation with the logistic normal distribution is applied to model the prior, $\boldsymbol{\alpha}$, of the level proportion, $\boldsymbol{\eta}$. We choose a conjugate prior of a multinomial, so $p(\boldsymbol{\eta}_n|\boldsymbol{\alpha})$ follows the Dirichlet distribution parametrized by $\boldsymbol{\alpha}$. To configure the inference network on the Dirichlet prior, the Laplace approximation is used (MacKay 1998; Srivastava and Sutton 2017; Hennig et al. 2012).

$$\mathcal{L}(\boldsymbol{x}) = \mathbb{E}_q \left[\log \frac{p(\boldsymbol{v})}{q(\boldsymbol{v}|\boldsymbol{x})} + \log \frac{p(\boldsymbol{\eta})}{q(\boldsymbol{\eta}|\boldsymbol{x})} + \log \prod_{\boldsymbol{\zeta}, \boldsymbol{l}} \frac{p(\boldsymbol{\zeta}|\boldsymbol{v})}{q(\boldsymbol{\zeta}|\boldsymbol{x})} \frac{p(\boldsymbol{l}|\boldsymbol{\eta})}{q(\boldsymbol{l}|\boldsymbol{x})} \frac{p(\boldsymbol{z}|\boldsymbol{\mu}_{\boldsymbol{\zeta}_l}, \boldsymbol{\sigma}^2_{\boldsymbol{\zeta}_l})}{q(\boldsymbol{z}|\boldsymbol{x})} + \log p(\boldsymbol{x}|\boldsymbol{z}) \right]. \tag{5}$$

Training Algorithm of Clustering Hierarchy

HCRL is formalized according to the stick-breaking process scheme. Unlike the CRP, the stick-breaking process does not represent the direct sampling of the mixture component at the data instance level. Therefore, it is necessary to devise a

heuristic algorithm for operations, such as *GROW*, *PRUNE*, and *MERGE*, to refine the hierarchy structure. Section 3 of the supplementary material provides details about each operation. In the below description, an *inner* path and a *full* path refer to the path ending with an internal node and a leaf node, respectively.

Algorithm 1 Training for Hierarchically Clustered Representation Learning

Require: Training data $\boldsymbol{x}$; number of epochs, E; tree-based hierarchy depth, L; period of performing GROW, t_{grow}; minimum number of epochs locking the hierarchy, t_{lock}
Ensure: $T^{(E)}, \boldsymbol{\omega}, \{a_i, b_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\}_{i \in \mathcal{M}_{T^{(E)}}}$

$\boldsymbol{\mu}_{\overline{\boldsymbol{\zeta}}_{1:L}}, \boldsymbol{\sigma}^2_{\overline{\boldsymbol{\zeta}}_{1:L}} \leftarrow$ Initialize L Gaussian of a single path $\overline{\boldsymbol{\zeta}}$
$T^{(0)} \leftarrow$ Initialize the tree-based hierarchy having $\overline{\boldsymbol{\zeta}}$
$t \leftarrow 0$
for each epoch $e = 1, \cdots, E$ **do**
  Update the weight parameters using $\nabla \mathcal{L}(\boldsymbol{x})$
  $\{a_i, b_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\}_{i \in \mathcal{M}_{T^{(e-1)}}} \leftarrow$ Update node-specific parameters using $\nabla_{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2} \mathcal{L}(\boldsymbol{x})$
  Update other variational parameters using $\nabla \mathcal{L}(\boldsymbol{x})$
  if $\text{mod}(e, t_{\text{grow}}) = 0$ **then**
    $T^{(e)}, \mathbb{Q} \leftarrow$ GROW
  end if
  if $T^{(e)} = T^{(e-1)}$ and $t \geq t_{\text{lock}}$ **then**
    $T^{(e)}, \mathbb{Q} \leftarrow$ PRUNE
    if $T^{(e)} = T^{(e-1)}$ **then** $T^{(e)}, \mathbb{Q} \leftarrow$ MERGE
  end if
  if $T^{(e)} \neq T^{(e-1)}$ **then** $t \leftarrow 0$ **else** $t \leftarrow t + 1$
end for

- **GROW** expands the hierarchy by creating a new branch under the heavily weighted internal node. Compared with the work of (Wang and Blei 2009), we modified GROW to first sample a path, $\overline{\boldsymbol{\zeta}}^*$, proportional to $\sum_n q(\boldsymbol{\zeta}_n = \overline{\boldsymbol{\zeta}}^*)$, and then to grow the path if the sampled path is an inner path.
- **PRUNE** cuts a randomly sampled minor full path, $\overline{\boldsymbol{\zeta}}^*$, satisfying $\frac{\sum_n q(\boldsymbol{\zeta}_n = \overline{\boldsymbol{\zeta}}^*)}{\sum_{n, \overline{\boldsymbol{\zeta}}} q(\boldsymbol{\zeta}_n = \overline{\boldsymbol{\zeta}})} < \delta$, where δ is the pre-defined threshold. If the removed leaf node of the full path is the last child of the parent node, we also recursively remove the parent node.
- **MERGE** combines two full paths, $\overline{\boldsymbol{\zeta}}^{(i)}$ and $\overline{\boldsymbol{\zeta}}^{(j)}$, with similar posterior probabilities, measured by $J(\overline{\boldsymbol{\zeta}}^{(i)}, \overline{\boldsymbol{\zeta}}^{(j)}) = \boldsymbol{q}_i \boldsymbol{q}_j^T / |\boldsymbol{q}_i||\boldsymbol{q}_j|$, where $\boldsymbol{q}_i = [q(\boldsymbol{\zeta}_1 = \overline{\boldsymbol{\zeta}}^{(i)}), \cdots, q(\boldsymbol{\zeta}_N = \overline{\boldsymbol{\zeta}}^{(i)})]$.

Algorithm 1 summarizes the overall algorithm for HCRL. The tree-based hierarchy T is defined as $(\mathbb{N}, \mathbb{P})$, where $\mathbb{N}$ and $\mathbb{P}$ denote a set of nodes and paths, respectively. We refer to the node at level l lying on path $\boldsymbol{\zeta}$, as $\text{N}(\boldsymbol{\zeta}_{1:l}) \in \mathbb{N}$. The defined paths, $\mathbb{P}$, consist of full paths, $\mathbb{P}_{\text{full}}$, and inner paths, $\mathbb{P}_{\text{inner}}$, as a union set. The GROW algorithm is executed for every specific iteration period, t_{grow}. After ellapsing t_{lock} iterations since performing the GROW operation, we begin to check whether the PRUNE or MERGE operation should be performed. We prioritize the PRUNE operation first, and if the condition of performing PRUNE is not satisfied, we check for the MERGE operation next. After performing any operation, we initialize t to 0, which is for locking the changed hierarchy during minimum t_{lock} iterations to be fitted to the training data.

Experiments

Datasets and Baselines

Datasets We used various hierarchically organized benchmark datasets, such as CIFAR-100, RCV1_v2, 20Newsgroups, as well as a flat structured benchmark dataset, MNIST. Supplementary Section 10 illustrates the details of the data pre-processing.

Baselines We completed our evaluation in two aspects: 1) optimizing the density estimation, and 2) clustering the hierarchical categories.

First, we evaluated HCRL1 and HCRL2 from the density estimation perspective by comparing it with diverse flat clustered representation learning models, such as Variational Autoencoder (VAE) (Kingma and Welling 2014), Variational Deep Embedding (VaDE) (Jiang et al. 2017), Improved Deep Embedded Clustering (IDEC) (Guo et al. 2017), Deep Clustering Network (DCN) (Yang et al. 2017), Infinite Mixture of Variational Autoencoders (IMVAE) (Abbasnejad, Dick, and van den Hengel 2017); and VAE-nCRP.

Second, we tested HCRL1 and HCRL2 from the accuracy perspective by comparing it with multiple divisive hierarchical clusterings, such as VAE-nCRP, Hierarchical K-means (HKM) (Nister and Stewenius 2006), Mixture of Hierarchical Gaussians (MOHG) (Vasconcelos and Lippman 1999), Recursive Gaussian Mixture Model (RGMM), and Recursive Scalable Sparse Subspace Clustering by Orthogonal Matching Pursuit (RSSCOMP). More details can be found in Supplementary Section 11.

Quantitative Analysis

We used two measures to evaluate the learned representations in terms of the density estimations: 1) negative log likelihood (NLL), and 2) reconstruction errors (REs). Autoencoder models, such as IDEC and DCN, were tested only for the REs. The NLL is estimated with 100 samples. Table 1 indicates that HCRL is best in the NLL and is competent in the REs which means that the hierarchically clustered embeddings preserve the intrinsic raw data structure.

Additionally, we evaluated hierarchical clustering accuracies by following (Xie, Girshick, and Farhadi 2016), except for MNIST that is flat structured. Table 2 points out that HCRL2 has better micro-averaged F-scores compared with every baseline. HCRL2 is able to reproduce the ground truth hierarchical structure of the data, and this trend is consistent when HCRL2 compared with the pipelined model, such as VaDE with a clustering model. The result of the comparisons with the clustering models, such as HKM, MOHG, RGMM, and RSSCOMP, is interesting because it experimentally proves that the joint optimization of hierarchical

Table 1: Test set performance of the negative log likelihood (NLL) and the reconstruction errors (REs). Replicated ten times, and the best in bold. $P^{\dagger} < 0.05$ (Student's t-test). *Model-*$L\#$ means that the model trained with the $\#$-depth hierarchy.

	MNIST		CIFAR-100		RCV1_v2		20Newsgroups	
Model	**NLL**	**REs**	**NLL**	**REs**	**NLL**	**REs**	**NLL**	**REs**
VAE (Kingma and Welling 2014)	230.71	10.46	1960.06	57.54	2559.46	1434.59	2735.80	1788.22
VaDE (Jiang et al. 2017)	217.20	10.35	1921.85	53.60	2558.32	1426.38	2733.46	1782.86
IDEC (Guo et al. 2017)	N/A	12.75	N/A	64.09	N/A	1376.26	N/A	$\mathbf{1660.61}^{\dagger}$
DCN (Yang et al. 2017)	N/A	11.30	N/A	44.26	N/A	1361.98	N/A	1691.17
IMVAE (Abbasnejad, Dick, and van den Hengel 2017)	296.57	10.69	1992.83	$\mathbf{40.45}^{\dagger}$	2566.01	1387.02	2722.81	1718.08
VAE-nCRP-$L3$ (Goyal et al. 2017)	718.78	32.67	2969.62	198.66	2642.88	1538.42	2712.28	1680.56
VAE-nCRP-$L4$ (Goyal et al. 2017)	721.00	32.53	2950.73	198.97	2646.48	1542.81	2713.58	1680.71
HCRL1-$L3$	$209.59^{\dagger}$	$9.28^{\dagger}$	$1864.69^{\dagger}$	55.12	2562.79	1418.30	2732.10	1792.13
HCRL1-$L4$	$212.31^{\dagger}$	$8.31^{\dagger}$	$1860.22^{\dagger}$	55.56	2555.84	1404.23	2727.49	1754.94
HCRL2-$L3$	$\mathbf{203.24}^{\dagger}$	$8.70^{\dagger}$	$1843.40^{\dagger}$	50.44	$2554.50^{\dagger}$	1395.05	2726.75	1828.71
HCRL2-$L4$	$203.91^{\dagger}$	$\mathbf{8.16}^{\dagger}$	$\mathbf{1849.13}^{\dagger}$	50.47	$\mathbf{2535.43}^{\dagger}$	**1353.34**	**2702.88**	1711.30

Table 2: Hierarchical clustering accuracies with F-scores, on CIFAR-100 with a depth of three, RCV1_v2 with a depth of four, and 20Newsgroups with a depth of four. Replicated ten times, and a confidence interval with 95%. Best in bold.

Model	**CIFAR-100**	**RCV1_v2**	**20Newsgroups**
HKM (Nister and Stewenius 2006)	$0.162_{\pm 0.008}$	$0.256_{\pm 0.068}$	$0.410_{\pm 0.043}$
MOHG (Vasconcelos and Lippman 1999)	$0.085_{\pm 0.038}$	$0.103_{\pm 0.014}$	$0.040_{\pm 0.012}$
RGMM	$0.169_{\pm 0.012}$	$0.274_{\pm 0.052}$	$0.435_{\pm 0.037}$
RSSCOMP (You, Robinson, and Vidal 2016)	$0.146_{\pm 0.023}$	$0.266_{\pm 0.055}$	$0.295_{\pm 0.047}$
VAE-nCRP (Goyal et al. 2017)	$0.201_{\pm 0.008}$	$0.413_{\pm 0.024}$	$0.558_{\pm 0.027}$
VaDE (Jiang et al. 2017) + HKM	$0.164_{\pm 0.012}$	$0.331_{\pm 0.066}$	$0.485_{\pm 0.056}$
VaDE (Jiang et al. 2017) + MOHG	$0.166_{\pm 0.016}$	$0.423_{\pm 0.093}$	$0.492_{\pm 0.071}$
VaDE (Jiang et al. 2017) + RGMM	$0.181_{\pm 0.013}$	$0.386_{\pm 0.062}$	$0.410_{\pm 0.065}$
VaDE (Jiang et al. 2017) + RSSCOMP	$0.192_{\pm 0.021}$	$0.272_{\pm 0.044}$	$0.291_{\pm 0.043}$
HCRL1	$0.199_{\pm 0.016}$	$0.437_{\pm 0.029}$	$0.566_{\pm 0.048}$
HCRL2	$\mathbf{0.225}_{\pm 0.014}$	$\mathbf{0.455}_{\pm 0.030}$	$\mathbf{0.601}_{\pm 0.097}$

clustering in the embedding space improves hierarchical clustering accuracies. HCRL2 also presented better hierarchical accuracies than VAE-nCRP. We conjecture the reasons for the modeling aspect of VAE-nCRP: 1) the simplified prior modeling on the variance of the mixture component as just constants, and 2) the non-flexible learning of the internal components.

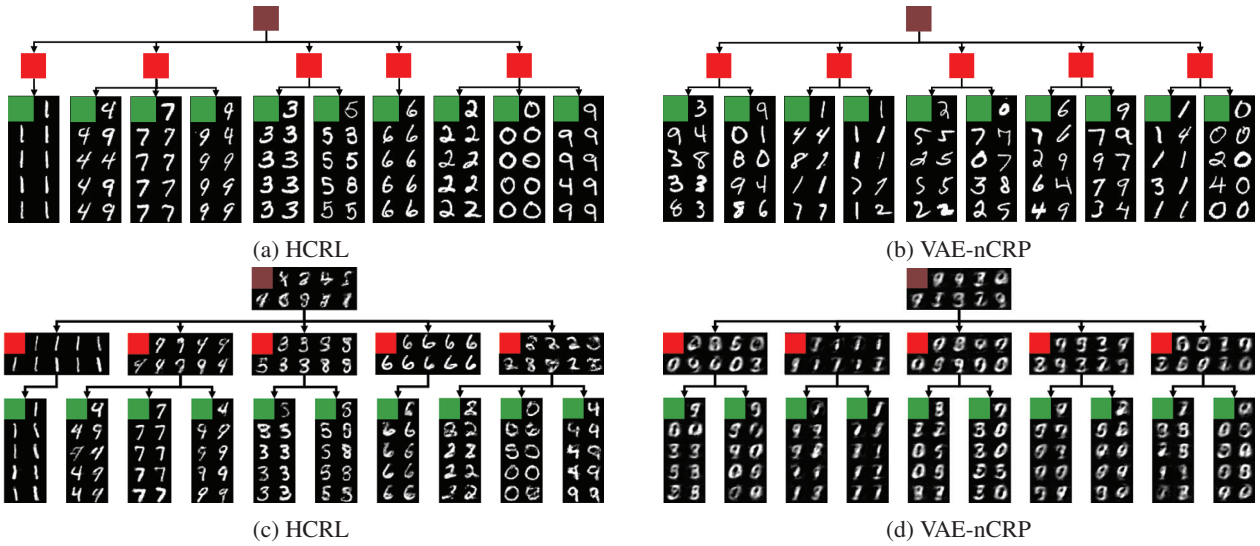
The performance gain of HCRL2 compared to HCRL1 arises from the detailed modeling of the level proportion. The prior assumption that the level proportion is shared by all data may give rise to the optimization biased towards the learning of leaf components. Specifically, a lot of data would be generated from the leaf components with the high probability since the leaf components have small variance, which causes the global level proportion to focus the high probability on the leaf level.

Qualitative Analysis

MNIST In Figure 1, the digits {4, 7, 9} and the digits {3, 8} are grouped together with a clear hierarchy, which was consistent between HCRL2 and VaDE. Also, some digit images {0, 4, 2} in a round form are grouped, together, in HCRL2. In addition, among the reconstructed digit images from the hierarchical mixture components, the digit images generated from the root have blended shapes from 0 to 9, which is natural considering the root position. As shown in Figure 1, similarly shaped digit images are hierarchically clustered, and the specific hierarchical clustering results are partly visualized in Figure 4. Additionally, we reconstructed images by feeding the mean embedding vectors for the Gaussian mixture components, and the images are enumerated with the corresponding mixture components like the right top subfigure of Figure 1. Figure 4 suggests that HCRL captures the more intrinsic structure of MNIST; and HCRL shows that the branches in the hierarchy learned more distinguishing features than VAE-nCRP.

CIFAR-100 Figure 5 shows the hierarchical clustering results on CIFAR-100, which are inferred from HCRL2. Given that there were no semantic inputs from the data, the color was dominantly reflected in the clustering criteria. However, if one observes the second hierarchy, the scene images of the same sub-hierarchy are semantically consistent, although the background colors are slightly different.

RCV1_v2 and 20Newsgroups Figure 6 shows the embedding of RCV1_v2. VAE and VaDE show no hierarchy, and close sub-hierarchies are distantly embedded. Since the flat clustered representation learning focuses on isolating clusters from each other, the distances between different clusters tend to be uniformly distributed. However, when constructing a hi-



(a) HCRL (b) VAE-nCRP

(c) HCRL (d) VAE-nCRP

Figure 4: Comparison of the hierarchical clustering (a, b) and the reconstructed images (c, d) on MNIST. The result learned from HCRL shows that digit images of similar shape are hierarchically clustered without any help of external knowledge. We obtained the reconstructed images by feeding the mean vectors for each Gaussian mixture component into the bottleneck z.

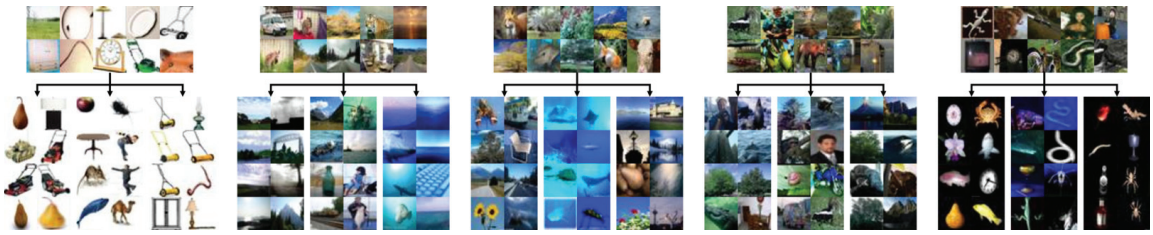
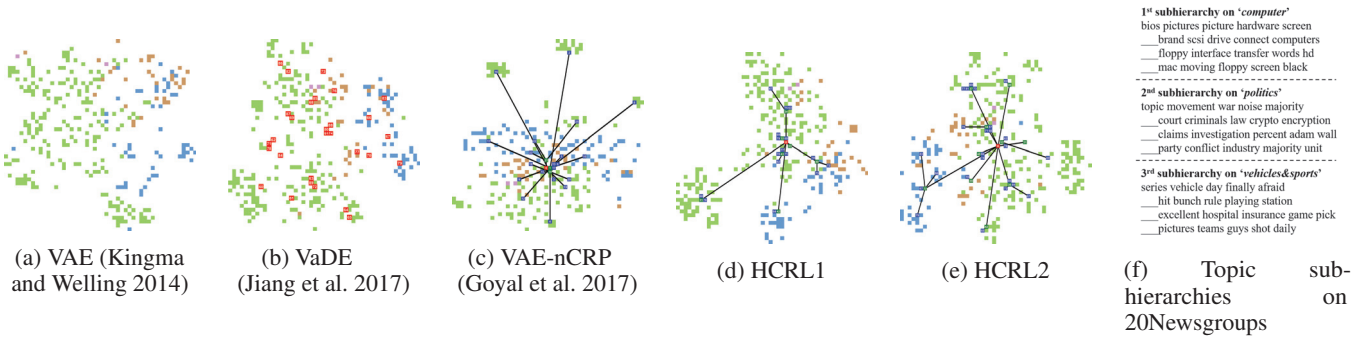


Figure 5: Example extracted sub-hierarchies on CIFAR-100



(a) VAE (Kingma and Welling 2014) (b) VaDE (Jiang et al. 2017) (c) VAE-nCRP (Goyal et al. 2017) (d) HCRL1 (e) HCRL2 (f) Topic sub-hierarchies on 20Newsgroups

Figure 6: Hierarchical clustering results on text datasets. (a,b,c,d,e) Comparison of embeddings on RCV1_v2, plotted using t-SNE (Maaten and Hinton 2008). We mark the mean of a mixture component with a numbered square, colored by {red (root), green (internal), blue (leaf)}. The edges between components are connected if the components are having the direct parent-child relationship. (f) the topic hierarchy from 20Newsgroup

erarchy, the distance of two clusters in the same sub-hierarchy needs to be reduced. VAE-nCRP guides the internal mixture components to be agglomerated at the center, and the cause of agglomeration is the generative process of VAE-nCRP, where the parameter of the internal components are inferred without direct information from data, which is a key weakness of VAE-nCRP. HCRL1 and HCRL2 show a relatively clear separation without the agglomeration. Figure 6 also shows the example sub-hierarchies on 20Newsgroups. We observe relatively more general contents in the internal clusters than in the leaf clusters of each internal cluster.

Conclusion

In this paper, we have presented a hierarchically clustered representation learning framework for the hierarchical mixture density estimation on deep embeddings. HCRL aims at encoding the relations among clusters as well as among instances to preserve the internal hierarchical structure of data. We have introduced two models called HCRL1 and HCRL2, whose the main differentiated features are 1) the crucial assumption regarding the internal mixture components for having the ability to generate data directly, and 2) the level selection modeling. HCRL2 improves the performance of HCRL1 by inferring the data-specific level proportion through the unbalanced autoencoding neural architecture. From the modeling and the evaluation, we found that our proposed models enable the improvements due to the high flexibility modeling compared with the baselines.

Acknowledgments This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2019M3F2A1072239)

References

Abbasnejad, M. E.; Dick, A.; and van den Hengel, A. 2017. Infinite variational autoencoder for semi-supervised learning. In *CVPR*, 781–790. IEEE.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8):1798–1828.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3(Jan):993–1022.

Chu, W., and Cai, D. 2017. Stacked similarity-aware autoencoders. In *IJCAI*, 1561–1567. AAAI Press.

Goyal, P.; Hu, Z.; Liang, X.; Wang, C.; and Xing, E. P. 2017. Nonparametric variational auto-encoders for hierarchical representation learning. In *ICCV*, 5094–5102.

Griffiths, T. L.; Jordan, M. I.; Tenenbaum, J. B.; and Blei, D. M. 2004. Hierarchical topic models and the nested chinese restaurant process. In *NIPS*, 17–24.

Guo, X.; Gao, L.; Liu, X.; and Yin, J. 2017. Improved deep embedded clustering with local structure preservation. In *IJCAI*, 1753–1759.

Hennig, P.; Stern, D.; Herbrich, R.; and Graepel, T. 2012. Kernel topic models. In *Artificial Intelligence and Statistics*, 511–519.

Huang, P.; Huang, Y.; Wang, W.; and Wang, L. 2014. Deep embedding network for clustering. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, 1532–1537. IEEE.

Ji, Z.; Huang, Y.; Sun, Q.; and Cao, G. 2016. A spatially constrained generative asymmetric gaussian mixture model for image segmentation. *Journal of Visual Communication and Image Representation* 40:611–626.

Jiang, Z.; Zheng, Y.; Tan, H.; Tang, B.; and Zhou, H. 2017. Variational deep embedding: An unsupervised and generative approach to clustering. In *IJCAI*, 1965–1972. AAAI Press.

Jordan, M. I.; Ghahramani, Z.; Jaakkola, T. S.; and Saul, L. K. 1999. An introduction to variational methods for graphical models. *Machine Learning* 37(2):183–233.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. In *ICLR*.

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of machine learning research* 9(Nov):2579–2605.

MacKay, D. J. 1998. Choice of basis for laplace approximation. *Machine learning* 33(1):77–86.

Mimno, D.; Li, W.; and McCallum, A. 2007. Mixtures of hierarchical topics with pachinko allocation. In *ICML*, 633–640. ACM.

Nalisnick, E.; Hertel, L.; and Smyth, P. 2016. Approximate inference for deep latent gaussian mixtures. In *NIPS Workshop on Bayesian Deep Learning*, volume 2.

Nister, D., and Stewenius, H. 2006. Scalable recognition with a vocabulary tree. In *CVPR*, volume 2, 2161–2168. Ieee.

Pitman, J., et al. 2002. Combinatorial stochastic processes. Technical report, Technical Report 621, Dept. Statistics, UC Berkeley, 2002. Lecture notes for St. Flour course.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1985. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Srivastava, A., and Sutton, C. 2017. Autoencoding variational inference for topic models. In *ICLR*.

Vasconcelos, N., and Lippman, A. 1999. Learning mixture hierarchies. In *NIPS*, 606–612.

Wang, C., and Blei, D. M. 2009. Variational inference for the nested chinese restaurant process. In *NIPS*, 1990–1998.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *ICML*, 478–487.

Yang, B.; Fu, X.; Sidiropoulos, N. D.; and Hong, M. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *ICML*, 3861–3870.

You, C.; Robinson, D.; and Vidal, R. 2016. Scalable sparse subspace clustering by orthogonal matching pursuit. In *CVPR*, 3918–3927.

Zhang, H.; Jin, X.; Wu, Q. J.; Wang, Y.; He, Z.; and Yang, Y. 2018. Automatic visual detection system of railway surface defects with curvature filter and improved gaussian mixture model. *IEEE Transactions on Instrumentation and Measurement* 67(7):1593–1608.