


Nonparametric Bayesian Deep Visualization

Haruya Ishizuka ¹ and Daichi Mochihashi²

¹ Bridgeston Corporation haruya.ishizuka@bridgestone.com

² The Institute of Statistical Mathematics daichi@ism.ac.jp

Abstract. Visualization methods such as *t*-SNE [1] have helped in knowledge discovery from high-dimensional data; however, their performance may degrade when the intrinsic structure of observations is in low-dimensional space, and they cannot estimate clusters that are often useful to understand the internal structure of a dataset. A solution is to visualize the latent coordinates and clusters estimated using a neural clustering model. However, they require a long computational time since they have numerous weights to train and must tune the layer width, the number of latent dimensions and clusters to appropriately model the latent space. Additionally, the estimated coordinates may not be suitable for visualization since such a model and visualization method are applied independently. We utilize neural network Gaussian processes (NNGP) [2] equivalent to a neural network whose weights are marginalized to eliminate the necessity to optimize weights and layer widths. Additionally, to determine latent dimensions and the number of clusters without tuning, we propose a latent variable model that combines NNGP with automatic relevance determination [3] to extract necessary dimensions of latent space and infinite Gaussian mixture model [4] to infer the number of clusters. We integrate this model and visualization method into nonparametric Bayesian deep visualization (NPDV) that learns latent and visual coordinates jointly to render latent coordinates optimal for visualization. Experimental results on images and document datasets show that NPDV shows superior accuracy to existing methods, and it requires less training time than the neural clustering model because of its lower tuning cost. Furthermore, NPDV can reveal plausible latent clusters without labels.

Keywords: Data visualization · Gaussian processes · Nonparametric Bayesian models · Neural network

1 Introduction

Visualization methods such as *t*-SNE [1], which compress input to two- or three-dimensional visual coordinates to be mapped on a scatter plot, provide a useful overview of high-dimensional data, and a number of methods have been proposed. These methods estimate visual coordinates based on the similarity of data points and fall into two categories. The first category is local methods to preserve neighbor structures in the original space [5–9]. These methods utilize nearest neighbor graph that represents pairwise similarity to retain distances between neighbors. Among them, *t*-SNE [1] and UMAP [10] are the most popular

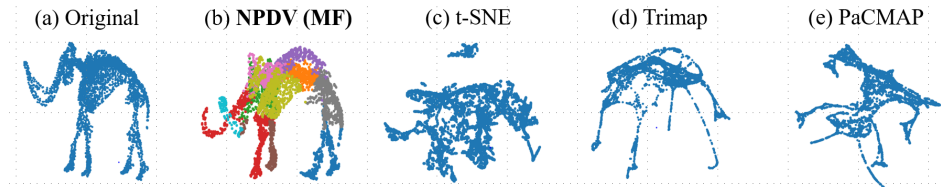


Fig. 1: 3D Visualization of 100-dimensional data generated by transforming the mammoth data in (a). Since NPDV(MF) estimates not only visual coordinates but also latent clusters, the associated plot in (b) can be colored by the cluster assignments, unlike existing methods in (c)–(e). See section 5 for details.

algorithms. The second is global methods to exploit relationships among three points, and preserve distances between data points distant from one another [11–13]. Particularly, Trimap [14] arguably shows comparable accuracy to *t*-SNE and UMAP. There is a trade-off between preserving the local or global structures. PaCMAP [15] exceptionally preserves both by combining the loss functions of local and global methods. They have improved knowledge discovery in various domains, such as bioinformatics [16] and audio processing [17].

However, they have several drawbacks to reveal hidden structures behind datasets. First, their performance may degrade when the observations are distributed on a low-dimensional manifold embedded in the observation space. In this case, the similarity between observations may differ from that in the manifold, resulting in an inaccurate visualization. This problem worsens when the manifold is embedded by a highly nonlinear function. Second, they cannot estimate clusters. Visualization together with clusters provides an intuitive understanding of the internal structure of a dataset; however, most of these methods only estimate visual coordinates. Supervised dimensionality-reduction [18–20] utilize labels to address this drawback; however, these are not always available. We present these problems through a simulation example. In this example, the 3D mammoth data shown in Fig.1 (a) is embedded into a 100-dimensional space nonlinearly by a neural network; then, this data is visualized in 3D space using several methods³. Existing methods in Fig.1 (c)–(e) evidently fail to recover the original mammoth shape. In addition, the lack of estimating clusters make it difficult to interpret the internal structure of resulting plot.

A solution here is to visualize latent coordinates and clusters estimated using a neural clustering model [21–23]; however, this approach presents other issues. As these models can accurately model a low-dimensional manifold by leveraging a neural autoencoder and perform clustering simultaneously, this approach addresses the aforementioned issues. However, they often require a long computational time to optimize model performance since they have numerous neural weights to train and need to search their appropriate hyperparameter settings, which is not suitable for scientific visualization. Particularly, the width of hidden layers, the number of latent dimensions, and clusters that are criti-

³ Rotatable plots are provided as an html file in the supplemental material.

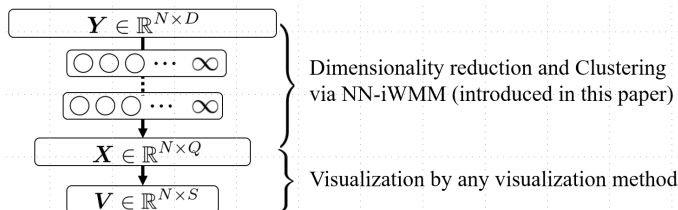


Fig. 2: Visualization flow of NPDV. NPDV integrates the estimation of latent coordinates \mathbf{X} from observations \mathbf{Y} using NN-iWMM and that of visual coordinates \mathbf{V} from \mathbf{X} . Clustering is also performed by NN-iWMM.

cal hyperparameters to reveal the latent structure. Furthermore, the estimated latent coordinates may not be suitable for visualization as such a model and visualization method are independently applied.

We utilize neural network Gaussian processes (NNGP) [2], which are equivalent to a neural network whose weights are marginalized, to address these issues. The marginalization greatly reduces the computational time by eliminating the necessity to optimize numerous weights and layer width while exploiting the power of neural networks. Additionally, to determine the number of latent dimensions and clusters without tuning, we propose the neural network infinite warped mixture model (NN-iWMM) by combining NNGP with automatic relevance determination [3] to extract the necessary dimensions of latent space and the infinite Gaussian mixture model [4] to infer the number of clusters. Finally, NN-iWMM and visualization methods are integrated as shown in Fig. 2, into nonparametric Bayesian deep visualization (NPDV) that jointly infers latent and visual coordinates to render latent coordinates optimal for visualization. Based on NPDV, we introduce NPDV(MF), which employs matrix factorization to linearly reduce the dimensionality, and NPDV(t -SNE) based on t -SNE; both methods enable to visualize the internal structure of dataset by utilizing the estimated clusters. As shown in Fig. 1 (b), we can observe from the simulation study that NPDV(MF) could accurately recover the mammoth shape. NPDV(t -SNE) achieves better accuracy than existing methods and NPDV(MF) for real-world data. Furthermore, NPDV(t -SNE) shows two preferable properties in unsupervised settings: (1) it takes considerably less training time than the neural clustering model and (2) has the ability to reveal plausible clusters without label information..

The remainder of this paper is organized as follows. We introduce the preliminaries in Section 2, and the proposed models and their training algorithm in Sections 3 and 4, respectively. Subsequently, we demonstrate their advantages through simulation and real data experiments in Sections 5 and 6.

2 Infinite Warped Mixture Model

The proposed models are based on the infinite warped mixture model (iWMM) [24], an extension of the Gaussian process latent variable model (GPLVM) [25]

that uses Gaussian processes [26] for dimensionality reduction. We introduce the notation and iWMM as preliminary. Let D , Q , and S denote the dimensionalities of observations $\mathbf{Y} = \{\mathbf{y}_i \in \mathbb{R}^D\}_{i=1}^N$, latent coordinates $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^Q\}_{i=1}^N$, and visual coordinates $\mathbf{V} = \{\mathbf{v}_i \in \mathbb{R}^S\}_{i=1}^N$, respectively. Q is smaller than D and larger than S , and S is typically set to two or three. N represents the number of observations. $\mathcal{N}(\mathbf{m}, \mathbf{C})$ is a multivariate Gaussian distribution with a mean \mathbf{m} and covariance \mathbf{C} . \mathbf{I}_N represents an N -dimensional identity matrix.

GPLVM independently draws a latent coordinate \mathbf{x}_i from $\mathcal{N}(\mathbf{0}, \mathbf{I}_Q)$ and draws the d th column of observations $\mathbf{y}_{\cdot d} = \{y_{id}\}_{i=1}^N$ from the distribution below:

$$p(\mathbf{y}_{\cdot d} | \mathbf{X}) = \mathcal{N}(\mathbf{y}_{\cdot d} | \mathbf{0}, K + \beta^{-1} \mathbf{I}_N), \quad (1)$$

where K is the Gram matrix, each component is given by the kernel function $k(\mathbf{x}, \mathbf{x}')$ evaluated at two coordinates \mathbf{x} and \mathbf{x}' , and $\beta > 0$ is the precision.

Latent space sometimes has clusters; however, GPLVM fails to capture them as it assumes a unimodal Gaussian distribution as prior to \mathbf{X} . iWMM assumes the infinite Gaussian mixture model (∞ -GMM) [4] defined under Dirichlet process theory [27] as prior to \mathbf{X} to model latent clusters. Each coordinate \mathbf{x}_i is drawn from the following distribution:

$$p(\mathbf{x}_i) = \sum_{k=1}^{\infty} \pi_k \mathcal{N}(\mathbf{x}_i | \mathbf{m}_k, \mathbf{R}_k^{-1}), \quad (2)$$

where π_k , \mathbf{m}_k , and \mathbf{R}_k represent the mixing weight, mean, and precision matrix of the k th Gaussian distribution, respectively. For simplicity, \mathbf{R}_k is assumed to be a diagonal matrix in this study.

3 Proposed Methods

We utilize neural network Gaussian processes (NNGP) [2], which are equivalent to a neural network whose weights and biases are marginalized, to eliminate the necessity to optimize neural weights, biases and the width of hidden layer while implicitly utilizing a neural network. Subsequently, we explain NNGP and introduce a latent variable model that combines iWMM and NNGP. Finally, we propose the NPDV.

3.1 Neural Network Gaussian Processes

In an L -layer fully connected neural network, let $\phi(\cdot)$ and N_ℓ denote the activation and width of the ℓ th layer, respectively. It is assumed that the weights and biases of the ℓ th layer, W_{ij}^ℓ and b_i^ℓ for $i = 1, 2, \dots, N_\ell$ and $j = 1, 2, \dots, N_{\ell-1}$, are independently drawn from $\mathcal{N}(0, \sigma_w/N_\ell)$ and $\mathcal{N}(0, \sigma_b/N_\ell)$, respectively. A pre-activation of the ℓ th layer, $a_i^\ell(\mathbf{x}_n)$, is then computed as a linear combination of the post-activations of the $(\ell - 1)$ th layer $\{\phi(a_j^{\ell-1}(\mathbf{x}_n))\}_{j=1}^{N_{\ell-1}}$:

$$a_i^\ell(\mathbf{x}_n) = b_i^\ell + \sum_{j=1}^{N_{\ell-1}} W_{ij}^\ell \phi(a_j^{\ell-1}(\mathbf{x}_n)). \quad (3)$$

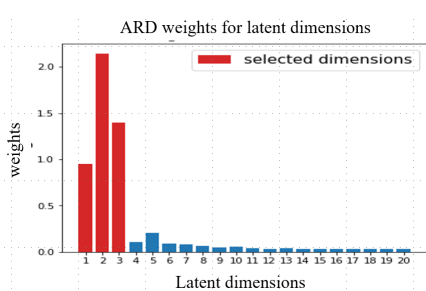


Fig. 3: ARD weights estimated by NPDV on simulation data. The three dimensions in red are selected as the necessary dimensions.

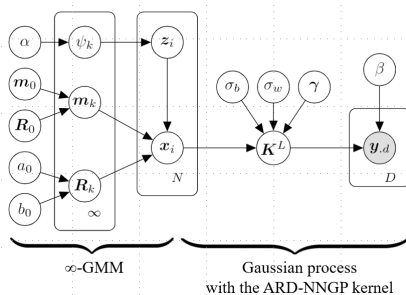


Fig. 4: Generative process of NN-iWMM. \mathbf{X} is drawn from the ∞ -GMM, and \mathbf{Y} is then drawn from Gaussian processes with the ARD-NNGP kernel.

$a_i^\ell(\mathbf{x}_n)$ is distributed with a Gaussian distribution from the central limit theorem when $N_\ell \rightarrow \infty$ because the summation in (3) is the sum of i.i.d. random variables. Because this result holds for any n , the output of the ℓ th layer $\{a_i^\ell(\mathbf{x}_n)\}_{n=1}^N$ is jointly distributed with the Gaussian process with the Gram matrix K^ℓ . Each component of K^ℓ , $k^\ell(\mathbf{x}, \mathbf{x}')$, is computed as follows:

$$k^\ell(\mathbf{x}, \mathbf{x}') = \sigma_b^2 + \sigma_w^2 \mathbb{E}_{a_i^{\ell-1} \sim \mathcal{GP}(\mathbf{0}, K^{\ell-1})} [\phi(a_i^{\ell-1}(\mathbf{x})) \phi(a_i^{\ell-1}(\mathbf{x}'))]. \quad (4)$$

The NNGP kernel is computed by iterating the recursion (4) L times, where Gaussian processes with this kernel are equivalent to an L -layer ∞ -width neural network whose weights and biases are marginalized. The first step of the original NNGP kernel is the inner product of the input: $k^0(\mathbf{x}, \mathbf{x}') = \sigma_b^2 + Q^{-1} \sigma_w^2 \mathbf{x}^T \mathbf{x}'$. In contrast, we introduce automatic relevance determination (ARD) [3] weights $\gamma = \{\gamma_q\}_{q=1}^Q$ into $k^0(\mathbf{x}, \mathbf{x}')$ to estimate the importance of each dimension:

$$k^0(\mathbf{x}, \mathbf{x}') = \sigma_b^2 + Q^{-1} \sigma_w^2 \sum_{q=1}^Q \gamma_q x_q x'_q. \quad (5)$$

An ARD weight γ_q increases if the q th dimension is highly related to the observations, and becomes zero if it is totally irrelevant. Fig. 3 shows the ARD weights estimated by NPDV and the selected dimensions when conducting the simulation study. In the simulation, 20 dimensional latent coordinates were estimated from a 100-dimensional data whose intrinsic structure is in three-dimensional space. We can observe from Fig. 3 that ARD correctly determines the dimensionality of the intrinsic space. Hereafter, the NNGP kernel with ARD weights is referred to as the ARD-NNGP kernel and is denoted as $k^L(\mathbf{x}_i, \mathbf{x}_j)$. The Gram matrix computed using $k^L(\mathbf{x}, \mathbf{x}')$ is denoted as K^L .

K^L is generally intractable due to the nonlinearity of \mathbf{X} ; however, it can be analytically computed when the activation is an identity map or a nonlinear map belonging to the polynomial rectified nonlinear function family [28], including

ReLU. We use kernel functions compatible with an identity map or ReLU to compute K^L . Appendix A presents the concrete forms ⁴.

3.2 NN-iWMM

NN-iWMM is formulated by substituting the ARD-NNGP kernel into iWMM and infers the latent coordinates, their cluster assignments and the number of latent clusters. Fig. 4 shows its generative process consisting of generating latent coordinates \mathbf{X} from ∞ -GMM and mapping \mathbf{X} to the observation space using Gaussian processes with the ARD-NNGP kernel.

In ∞ -GMM, the mixing weights $\{\pi_k\}_{k=1}^\infty$ are drawn from the stick-breaking process GEM(α) [29], which defines a distribution equivalent to the Dirichlet process. Specifically, π_k is computed as $\pi_k = \psi_k \prod_{j=1}^{k-1} (1 - \psi_j)$, where each ψ_j is drawn from the beta distribution Beta(1, α). Subsequently, the mean \mathbf{m}_k and diagonal components of the precision matrix $\{r_{kq}\}_{q=1}^Q$ of the k th Gaussian distribution are drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and the gamma distribution Gam(1, 1), respectively.

For $i = 1, 2, \dots, N$, the cluster assignment z_i is drawn from the categorical distribution $\text{Cat}(\{\pi_k\}_{k=1}^\infty)$ and \mathbf{x}_i is generated from the z_i th Gaussian distribution $\mathcal{N}(\mathbf{m}_{z_i}, \mathbf{R}_{z_i})$. Then, the Gram matrix K^L is computed from \mathbf{X} using (4) and (5). \mathbf{y}_d is drawn from $\mathcal{N}(\mathbf{0}, K^L)$ for $d = 1, 2, \dots, D$.

Due to the absence of weights and biases, it is unnecessary for NN-iWMM to optimize the weights and layer widths while leveraging a neural network. Additionally, unlike neural clustering models, using ∞ -GMM and ARD weights allows to determine the number of latent dimensions and clusters, which are critical to model the latent space appropriately, without tuning.

3.3 Nonparametric Bayesian Deep Visualization

NPDV jointly estimates \mathbf{X} and the visual coordinates \mathbf{V} by integrating the two reduction steps, as shown in Fig. 2. We use weighted latent coordinates $\mathbf{X}\boldsymbol{\gamma}^T$ as the input to prioritize the necessary dimensions of \mathbf{X} in terms of ARD weights when estimating \mathbf{V} . Denoting $\mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})$ and $\mathbf{L}(\mathbf{Y}, \mathbf{X})$ as the visualization loss to estimate \mathbf{V} and the loss of NN-iWMM to estimate \mathbf{X} , respectively, we introduce how to integrate these two losses into a Bayesian model using regularized Bayesian inference (regBayes) [30].

The regBayes framework enables the design of Bayesian models while considering the appropriate constraints on its posterior. This framework is built on a variational formulation of the Bayesian posterior $p(\boldsymbol{\theta}|\mathbf{Y}) \propto p(\mathbf{Y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$, where \mathbf{Y} and $\boldsymbol{\theta}$ are the observations and parameters, respectively. $p(\boldsymbol{\theta}|\mathbf{Y})$ can be viewed as a solution to the following variational optimization problem [31]:

$$\begin{cases} \min_{q(\boldsymbol{\theta})} & \text{KL}[q(\boldsymbol{\theta})\|p(\boldsymbol{\theta})] - \int q(\boldsymbol{\theta}) \log p(\mathbf{Y}|\boldsymbol{\theta}) d\boldsymbol{\theta} \\ \text{s.t.} & q(\boldsymbol{\theta}) \in \mathcal{P}, \end{cases} \quad (6)$$

⁴ All appendices are provided in the Supplemental Materials.

where \mathcal{P} is a set of probability distributions. In the regBayes framework, we consider the following optimization problem with constraints regarding the expectation of the regularizer $E_{q(\boldsymbol{\theta})}[\mathcal{R}(\boldsymbol{\theta}, \mathbf{Y})]$:

$$\begin{cases} \min_{q(\boldsymbol{\theta})} & \text{KL}[q(\boldsymbol{\theta})\|p(\boldsymbol{\theta})] - \int q(\boldsymbol{\theta}) \log p(\mathbf{Y}|\boldsymbol{\theta})d\boldsymbol{\theta} \\ \text{s.t.} & E_{q(\boldsymbol{\theta})}[\mathcal{R}(\boldsymbol{\theta}, \mathbf{Y})] \leq 0, q(\boldsymbol{\theta}) \in \mathcal{P}. \end{cases} \quad (7)$$

The optimal solution of (7) is obtained by

$$q^*(\boldsymbol{\theta}) \propto p(\mathbf{Y}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \exp(-\lambda\mathcal{R}(\boldsymbol{\theta}, \mathbf{Y})), \quad (8)$$

where λ is the Lagrange multiplier. From (8), the optimal posterior of $\boldsymbol{\theta}$ that satisfies the constraints is obtained by its right-hand term. For NPDV, $p(\mathbf{Y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ corresponds to the likelihood of NN-iWMM $\mathcal{L}(\mathbf{Y}, \mathbf{X})$:

$$\mathcal{L}(\mathbf{Y}, \mathbf{X}) = p(\mathbf{Y}|\mathbf{X})p(\mathbf{X}|\mathbf{z})p(\mathbf{z}|\{\mathbf{m}_k, \mathbf{r}_k, \psi_k\}_{k=1}^{\infty})p(\{\mathbf{m}_k, \mathbf{r}_k, \psi_k\}_{k=1}^{\infty}). \quad (9)$$

$\mathcal{R}(\mathbf{Y}, \boldsymbol{\theta})$ is given by $\mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})$. Therefore, the optimal posterior of the NPDV is obtained by

$$q^*(\mathbf{X}, \mathbf{V}) \propto \mathcal{L}(\mathbf{Y}, \mathbf{X}) \times \exp(-\lambda\mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})). \quad (10)$$

Hence, the joint optimization of $\mathcal{L}(\mathbf{Y}, \mathbf{X})$ and $\mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})$ is the posterior inference of $q^*(\mathbf{X}, \mathbf{V})$. NPDV makes the posterior of \mathbf{X} suitable for visualization because $\mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})$ in (10) serves as a regularizer to infer the posterior of NN-iWMM. λ is a hyperparameter that balances $\mathcal{L}(\mathbf{Y}, \mathbf{X})$ and $\mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})$, and NPDV degenerates to NN-iWMM when $\lambda = 0$. \mathbf{V} is treated as a deterministic parameter, as are the NNGP parameters $\{\sigma_w, \sigma_b, \boldsymbol{\gamma}\}$.

The *any* dimensionality-reduction method can be used for \mathcal{R}_{DR} . In this paper we combine matrix factorization (MF) and *t*-SNE, which are widely used in many different domains, with NPDV. We call these methods NPDV(MF) and NPDV(*t*-SNE), respectively. Notably, $\lambda = ND$ practically works well for both methods.

NPDV(MF): MF approximates a matrix using a product of two low-rank matrices, and is one of the widely used linear dimensionality-reduction methods. For NPDV(MF), the weighted coordinates $\mathbf{X}\boldsymbol{\gamma}^T$ are factorized as the rank S matrix \mathbf{W} and visual coordinates \mathbf{V} : $\mathbf{X}\boldsymbol{\gamma}^T \approx \mathbf{W}\mathbf{V}^T$. Denoting $\|\cdot\|_F^2$ as the Frobenius norm, $\|\mathbf{A}\|_F^2 \equiv \sum_{i,j} a_{ij}^2$, $\mathbf{A} \in \mathbb{R}^{I \times J}$, and the associated visualization loss $\mathcal{R}_{\text{MF}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})$ can be computed as

$$\mathcal{R}_{\text{MF}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V}) = \|\mathbf{X}\boldsymbol{\gamma}^T - \mathbf{W}\mathbf{V}^T\|_F^2. \quad (11)$$

NPDV(*t*-SNE): NPDV(MF) may fail to capture the nonlinear pattern as it linearly reduces the dimensionality of $\mathbf{X}\boldsymbol{\gamma}^T$. We then introduce NPDV(*t*-SNE), which uses the *t*-SNE loss for $\mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})$ as a nonlinear counterpart. For NPDV(*t*-SNE), $\mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})$ is computed as the divergence of similarities in

the latent and visual spaces. Denoting \odot as the element-wise Hadamard product, the similarity of weighted latent coordinates, $\mathbf{x}_i \odot \boldsymbol{\gamma}$ and $\mathbf{x}_j \odot \boldsymbol{\gamma}$, is computed from the conditional probability based on the Gaussian kernel:

$$p_{j|i}^X = \frac{\exp(-\|\boldsymbol{\gamma} \odot \mathbf{x}_i - \boldsymbol{\gamma} \odot \mathbf{x}_j\|^2 / 2\tau_i^2)}{\sum_{\ell \neq i} \exp(-\|\boldsymbol{\gamma} \odot \mathbf{x}_i - \boldsymbol{\gamma} \odot \mathbf{x}_\ell\|^2 / 2\tau_i^2)}, \quad p_{ij}^X = \frac{p_{i|j}^X + p_{j|i}^X}{2N}, \quad (12)$$

where τ_i^2 is the variance of the Gaussian distribution and is computed from the neighbors of $\boldsymbol{\gamma} \odot \mathbf{x}_i$ using a binary search with perplexity ρ , which is a hyperparameter that controls the number of neighbors.

The similarity of two visual coordinates, \mathbf{v}_i and \mathbf{v}_j , is evaluated using Student's t -distribution kernel as follows:

$$p_{ij}^V \equiv \frac{(1 + \|\mathbf{v}_j - \mathbf{v}_i\|^2)^{-1}}{\sum_k \sum_{\ell \neq k} (1 + \|\mathbf{v}_k - \mathbf{v}_\ell\|^2)^{-1}}. \quad (13)$$

$\mathcal{R}_{t\text{-SNE}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})$ is computed as the KL divergence between $\{p_{ij}^X\}_{i,j}$ and $\{p_{ij}^V\}_{i,j}$:

$$\mathcal{R}_{t\text{-SNE}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V}) \equiv \sum_{i,j,i \neq j} p_{ij}^X \log \frac{p_{ij}^Y}{p_{ij}^V}. \quad (14)$$

4 Bayesian Training

We employ variational inference to train the NPDV. NN-iWMM is a special case of NPDV when $\lambda = 0$; hence, we focus on the training algorithm for NPDV. The parameters of NPDV are estimated by maximizing the evidence lower bound (ELBO) \mathcal{L} of NPDV. \mathcal{L} is derived from Jensen's inequality and the variational distribution \mathcal{Q} that is used to approximate the true posterior [32]:

$$\mathcal{L} = \mathbb{E}_{\mathcal{Q}} \left[\log \frac{\mathcal{L}(\mathbf{Y}, \mathbf{X}) \times \exp(-\lambda \mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V}))}{\mathcal{Q}} \right]. \quad (15)$$

Following [33], the ∞ -GMM is approximated by a finite Gaussian mixture model whose maximum number of mixtures is K . Additionally, we impose the following mean-field assumption on \mathcal{Q} .

$$\mathcal{Q} = \prod_{i=1}^N q(\mathbf{x}_i) q(z_i) \prod_{k=1}^K q(\psi_k) q(\mathbf{m}_k) q(\mathbf{r}_k), \quad (16)$$

where $q(\mathbf{x}_i) = \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{S}_i)$. The variational distributions for the mixture model $q(\psi_k)$, $q(\mathbf{m}_k)$, $q(\mathbf{r}_k)$, and $q(z_i)$ have the same form as in [33]. Using the mean-field assumption on \mathcal{Q} and conditional independence of NN-iWMM, \mathcal{L} is decomposed into the four terms:

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{q(\mathbf{X})} [\log p(\mathbf{Y} | \mathbf{X})] - \mathbb{E}_{q(\mathbf{X})} [\log q(\mathbf{X})] \\ &+ \mathbb{E}_{q(\mathbf{X}, \mathbf{z}, \mathbf{m}, \mathbf{r}, \phi)} \left[\log \frac{p(\mathbf{X}, \mathbf{z}, \{\mathbf{m}_k, \mathbf{r}_k, \pi_k\}_{k=1}^K)}{q(\mathbf{z}, \{\mathbf{m}_k, \mathbf{r}_k, \pi_k\}_{k=1}^K)} \right] - \lambda \mathbb{E}_{q(\mathbf{X})} [\mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})] \\ &= \mathcal{L}_1 + \sum_{i=1}^N \mathcal{H}(q(\mathbf{x}_i)) + \mathcal{L}_2 - \lambda \mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V}), \end{aligned} \quad (17)$$

Algorithm 1 Variational inference algorithm for NPDV

Input observations \mathbf{Y} and the number of layers L
1. Pre-training
 Initialize $\mathbf{\Pi}_0 = [\{\boldsymbol{\mu}_i, \mathbf{S}_i\}_{i=1}^N, \boldsymbol{\zeta}, \sigma_b^2, \sigma_w^2, \boldsymbol{\gamma}, \beta]$
for $i=1, 2, \dots$ **do**
 Update $\mathbf{\Pi}_0$ with a gradient-based method
end for
 Initialize \mathbf{V}
2. Training NPDV.
 Initialize $\mathbf{\Pi}_1 = [\mathbf{z}, \{\mathbf{m}_k, \mathbf{r}_k, \pi_k\}_{k=1}^K]$
for $i=1, 2, \dots$ **do**
 Generate $\widetilde{\mathbf{X}}$ with (18)
 Approximate \mathcal{L}_1 , \mathcal{L}_2 , and \mathcal{R}_{DR} with $\widetilde{\mathbf{X}}$
 Update $\mathbf{\Pi}_1$ with the EM algorithm in Appendix B
 Update $\mathbf{\Pi}_2 = \{\boldsymbol{\mu}_i, \mathbf{S}_i\}_{i=1}^N, \boldsymbol{\zeta}, \mathbf{V}, \sigma_b^2, \sigma_w^2, \boldsymbol{\gamma}, \beta]$ with a gradient-based method
end for

Unlike Gaussian entropy $\mathcal{H}(q(\mathbf{x}_i))$, \mathcal{L}_1 , \mathcal{L}_2 , and $\mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})$ cannot be evaluated analytically. Therefore, we adapt the reparameterization trick [34] to approximate these quantities. Using this trick, Monte Carlo samples of \mathbf{x}_i , $\tilde{\mathbf{x}}_i$, is generated by affine-transforming the standard Gaussian noise $\boldsymbol{\epsilon}$ with $\boldsymbol{\mu}_i$ and \mathbf{S}_i :

$$\tilde{\mathbf{x}}_i = \boldsymbol{\mu}_i + \mathbf{S}_i \boldsymbol{\epsilon}_i; \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_Q) \text{ for } i = 1, 2, \dots, N. \quad (18)$$

Hereafter, we outline the evaluation of \mathcal{L}_1 , \mathcal{L}_2 , and \mathcal{R}_{DR} using $\widetilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_i\}_{i=1}^N$.

The naive approximation of \mathcal{L}_1 requires $\mathcal{O}(N^3)$ complexity because it inverts $K^L \in \mathbb{R}^{N \times N}$ and is difficult to train over a large dataset. We exploit the inducing-point approach [35] to reduce the complexity. Using this approach, \mathcal{L}_1 given $\widetilde{\mathbf{X}}$ is approximated by $\mathcal{O}(M^3)$, ($M \ll N$) complexity based on M pseudo-inputs $\boldsymbol{\zeta} \in \mathbb{R}^{M \times Q}$ in the latent space, and the corresponding Gaussian process outputs $\mathbf{u}_d \in \mathbb{R}^M$. The cost of the variational mixture model \mathcal{L}_2 , given $\widetilde{\mathbf{X}}$, is of the same form as the ELBO for ∞ -GMM in [33]. Moreover, the parameters of the variational mixture model can be updated using the expectation-maximization (EM) algorithm. $\mathcal{R}_{\text{DR}}(\mathbf{X}\boldsymbol{\gamma}^T, \mathbf{V})$ is computed by substituting $\widetilde{\mathbf{X}}$ and ARD weights with $\boldsymbol{\gamma}$. Appendix B provides the details of evaluating \mathcal{L}_1 and \mathcal{L}_2 and the update formulae of the parameters of the variational mixtures.

Algorithm 1 summarizes the NPDV training algorithm. First, we pretrain the NN-iWMM that assumes $\mathcal{N}(\mathbf{0}, \mathbf{I}_Q)$ as the prior of \mathbf{X} to initialize $\mathbf{\Pi}_0 = [\{\boldsymbol{\mu}_i, \mathbf{S}_i\}_{i=1}^N, \boldsymbol{\zeta}, \sigma_w^2, \sigma_b^2, \boldsymbol{\gamma}, \beta]$. Subsequently, we initialize \mathbf{V} . After that, We generate Monte Carlo samples $\widetilde{\mathbf{X}}$ using (18) and approximate \mathcal{L}_1 , \mathcal{L}_2 , and \mathcal{R}_{DR} to train the NPDV. Then, we update the parameters of the variational mixture model $\mathbf{\Pi}_1 = [\mathbf{z}, \{\mathbf{m}_k, \mathbf{r}_k, \psi_k\}_{k=1}^K]$ and the others $\mathbf{\Pi}_2 = [\{\boldsymbol{\mu}_i, \mathbf{S}_i\}_{i=1}^N, \boldsymbol{\zeta}, \mathbf{V}, \sigma_b^2, \sigma_w^2, \boldsymbol{\gamma}, \beta]$ using the EM algorithm and a gradient-based method, respectively.

Less parametricity: Due to the absence of weights and biases, NPDV has significantly less parameters than autoencoder variants. A symmetric neural au-

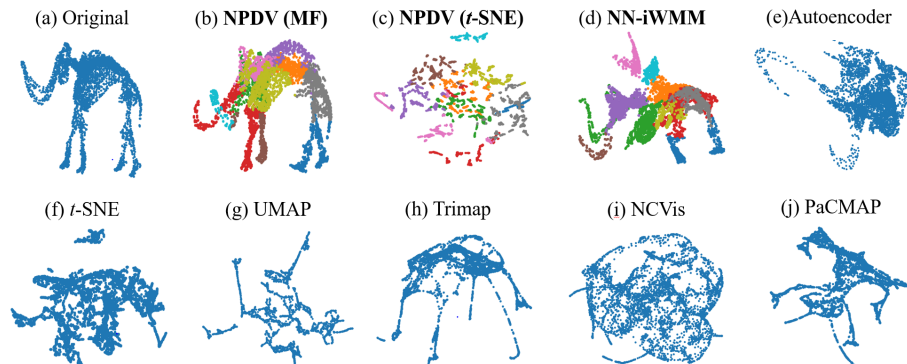


Fig. 5: 3D Visualization of a 100-dimensional data that are generated by transforming the mammoth data in (a) using a neural network. (b)–(d) are obtained by proposed methods and colored by the estimated clusters. (e)–(j) are obtained by existing methods,

toencoder has $2 \sum_{\ell=1}^L (N_{\ell} + N_{\ell} N_{\ell-1})$ weights and biases, where N_{ℓ} and N_0 represent the width of ℓ th hidden layer and dimensionality of the observations, respectively. It is not uncommon for them to have more than 10^7 parameters as N_{ℓ} often exceeds thousands. Conversely, because NNGP has no need to estimate weights and biases, NPDV has much less parameters than neural models.

Hyperparameter settings: NPDV has several hyperparameters; however, not all of them are considered in practice. For the maximum latent dimensions and cluster, Q and K , if they are set to sufficiently large values, the necessary latent dimensions and number of clusters are estimated by the ARD mechanism and ∞ -GMM. For the perplexity of NPDV(t -SNE), ρ , the number of inducing points M , learning rate η , and balance term λ , NPDV(t -SNE) achieves higher accuracy than the existing methods with $M=100$, $\rho=30$, $\eta=0.01$, and $\lambda=ND$. Therefore, we only focus on a single hyperparameter, the layer depth of the NN-iWMM L .

5 Simulation Study

We present the qualitative properties of the NPDV through a simulation study. As mentioned in Section 1, the visualization accuracy may degrade when observations are distributed on a lower dimensional manifold. To imitate such situation, the 3D mammoth data in Fig. 5(a) is embedded into a 100-dimensional space by a neural network. Then, we visualize this 100-dimensional data in 3D space using several methods. Besides NN-iWMM, NPDV(MF) and NPDV(t -SNE), we apply six existing methods to the data as baselines: Autoencoder⁵, t -SNE, UMAP [10], Trimap [14], NCvis [9], and PaCMAP [15]. Appendix C provides the details of the experimental settings.

⁵ The network architecture is the same as that of the data generation network

Table 1: Summary of the datasets used in Section 6. C and D are the number of labels and dimensionality of observations, respectively. For **20 news**, 20 labels are converted to 6 meta-labels according to the dataset guideline.

| Dataset | type | C | D |
|---------------|-----------|-----|-------|
| MNIST | Images | 10 | 784 |
| Fashion-MNIST | Images | 10 | 784 |
| 20 news | Documents | 6 | 1,000 |

Table 2: Average k -nearest neighbors classification accuracy with five different random seeds. The highest scores are in bold font.

| Method | MNIST | | | Fashion-MNIST | | | 20 news | | |
|-----------------|--------------|--------------|--------------|---------------|--------------|--------------|--------------|--------------|--------------|
| | $k=10$ | $k=20$ | $k=30$ | $k=10$ | $k=20$ | $k=30$ | $k=10$ | $k=20$ | $k=30$ |
| t -SNE | 0.930 | 0.920 | 0.915 | 0.819 | 0.794 | 0.783 | 0.726 | 0.700 | 0.690 |
| UMAP | 0.921 | 0.916 | 0.913 | 0.777 | 0.763 | 0.757 | 0.729 | 0.705 | 0.695 |
| Trimap | 0.902 | 0.897 | 0.891 | 0.774 | 0.760 | 0.757 | 0.740 | 0.720 | 0.713 |
| NCVis | 0.891 | 0.886 | 0.884 | 0.776 | 0.764 | 0.759 | 0.405 | 0.358 | 0.338 |
| PaCMAP | 0.902 | 0.896 | 0.894 | 0.778 | 0.766 | 0.759 | 0.741 | 0.724 | 0.717 |
| VSB-DVM | 0.931 | 0.920 | 0.915 | 0.837 | 0.819 | 0.806 | 0.778 | 0.757 | 0.749 |
| NN-iWMM | 0.893 | 0.884 | 0.881 | 0.765 | 0.748 | 0.741 | 0.725 | 0.706 | 0.698 |
| NPDV(MF) | 0.529 | 0.495 | 0.484 | 0.650 | 0.629 | 0.619 | 0.636 | 0.618 | 0.610 |
| NPDV(t -SNE) | 0.928 | 0.917 | 0.911 | 0.834 | 0.820 | 0.808 | 0.786 | 0.761 | 0.750 |

Fig. 5 shows the resulting plot of each method. In contrast to conventional visualization methods in Fig. 5 (f)–(j), since NN-iWMM based methods infers clusters in addition to latent coordinates, we colored the associated plots by estimated clusters. Especially, Fig.5 (b) shows that NPDV(MF) recovers the original mammoth shape more accurately than the existing methods in Fig.5 (e)–(j) and other NN-iWMM based methods in Fig.5 (c) and (d). This means NPDV(MF) accurately recovered the intrinsic manifold embedded in a high-dimensional space in this simulation. Furthermore, it enables to find the specific parts of the mammoth body, such as paws and horns, as clusters by coloring with the cluster assignments.

The plot of NPDV(t -SNE) in Fig.5 (c) is blurred because the t -SNE occasionally fails to capture the global structure. However, as shown in the next section, NPDV(t -SNE) achieves superior performance to NPDV(MF) on real-world data.

6 Experiments on Real-World Data

We demonstrate several advantages of NPDV(t -SNE) through real-world data experiments on three datasets. **MNIST** contains hand-written digit images, where each image is labeled one of 0–9. **Fashion-MNIST** contains images of clothing, where each image is labeled with one of 10 categories, such as T-shirts or shoes. **20 news** corpus records English articles, where each article is classified into one of 20 labels. Table 1 summarises these datasets. We randomly

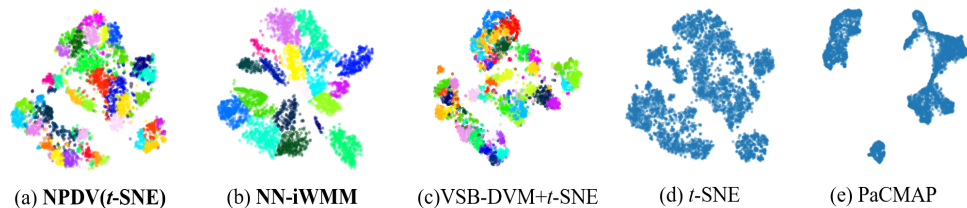


Fig. 6: Visualization of **Fashion-MNIST**. (a), (b) and (c) are colored by latent clusters that can be discovered by each method.

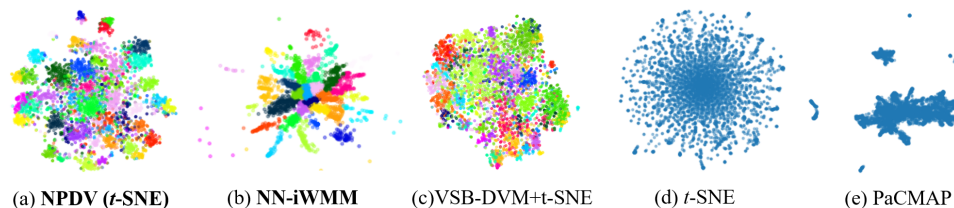


Fig. 7: Visualization of **20 news**. (a), (b) and (c) are colored by latent clusters that can be discovered by each method.

extracted 5,000 samples from each dataset for evaluation. For **MNIST** and **Fashion-MNIST**, all images are scaled within $[0, 1]$. For **20 news**, the original 20 labels were converted into six meta-labels, e.g., *rec* and *sci* for recreation and science, respectively, according to the dataset guideline⁶, and documents were transformed into 1,000-dimensional tf.idf vectors after removing stopwords and performing lemmatization.

We used the k -nearest neighbor classification accuracy for $k = [10, 20, 30]$ as a metric. This metric increases when coordinates with the same label are close to one another and measures how accurately they can capture label differences. Furthermore, we qualitatively compared the resulting plots. For model comparison, in addition to the methods used in section 5, we built VSB-DVM+t-SNE that applies t -SNE to latent coordinates estimated by the latest neural clustering model, VSB-DVM [23]. The hyperparameters of VSB-DVM are tuned by minimizing the loss of held-out 1,000 samples using Optuna [36] with 50 trials. Note that tuning VSB-DVM is time-consuming due to iterative model fitting. For NPDV(MF) and NPDV(t -SNE), the layer depth, L , the maximum number of latent dimensions and clusters, Q and K are set to $L = 6$, $Q = 100$ and $K = 50$, respectively. Appendix D provides The details of experimental settings and all visualization results.

Quantitative and qualitative comparison Table 2 lists the k -nearest neighbor classification accuracies. VSB-DVM+t-SNE and NPDV(t -SNE) outperform other methods for **Fashion-MNIST** and **20 news**. Notably, NPDV(t -SNE) shows comparable accuracies with well-tuned VSB-DVM+t-SNE. Fig. 5 and 6 show the

⁶ <http://qwone.com/~jason/20Newsgroups/>

Table 3: Number of parameters of optimized models.

| Method | MNIST | Fashion-MNIST | 20 news |
|-------------------|---------------------|--------------------|--------------------|
| VSB-DVM+ t -SNE | 133.6×10^6 | 29.1×10^6 | 2.81×10^6 |
| NPDV(t -SNE) | 1.04×10^6 | 1.04×10^6 | 1.04×10^6 |

Table 4: Elapsed time for tuning. h and m is hours and minutes, respectively.

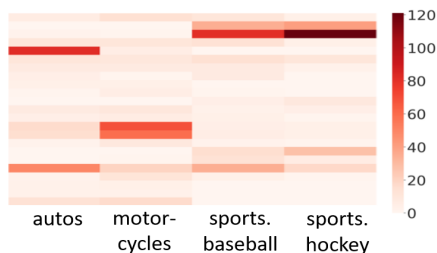
| Method | MNIST | Fashion-MNIST | 20 news |
|-------------------|----------------|----------------|---------------|
| VSB-DVM+ t -SNE | 5 days 13h 47m | 4 days 23h 23m | 6 days 0h 13m |
| NPDV(t -SNE) | 8h 32m | 8h 6m | 8h 19m |

resulting plots on **Fashion-MNIST** and **20 news** obtained by the five methods. For NPDV(t -SNE), NN-iWMM and VSB-DVM+ t -SNE, the points are colored by the estimated clusters. The coloring helps to understand cluster structures more easily than t -SNE and PaCMAP, as in the simulation study. For **20 news**, NPDV(t -SNE) shows better cluster separation compared to VSB-DVM+ t -SNE. We guess the reason why cluster structures are taken over to visual coordinates due to the joint training of NN-iWMM and t -SNE.

Computational cost comparison with neural clustering model NPDV(t -SNE) incurs significantly less computational cost compared to VSB-DVM+ t -SNE. VSB-DVM needs to estimate numerous weights. Consequently, it has 2.8–133 times more parameters than NPDV(t -SNE), as shown in Table 3. Additionally, VSB-DVM must tune several hyperparameters, which increases the computational time. Specifically, it took multiple days with 50 trials, as shown in Table 4. Conversely, we only focus on layer depth L to train NPDV(t -SNE). NPDV(t -SNE) finishes computation in a considerably shorter time than VSB-DVM even if we try all candidates $L = \{4, 5, 6, 7\}$.

Latent cluster discovery We investigated the characteristics of the clusters estimated using the NPDV(t -SNE). In addition to the **20 news**, we used the **Brown** corpus⁷. For the **Brown** corpus, we randomly extracted 5,000 sentences and converted them into sentence vectors using SIF weighting [37].

For **20 news**, we investigated the relationships between the four lower labels in *rec* and estimated clusters. The articles in *rec* belong to one of 23 clusters. Fig. 8 is a cross table of these labels and clusters. Evidently, the same labeled articles belong to specific clusters, while these labels were not used during training. The link between labels and clusters indicates that a common topic exists

Fig. 8: Cross table of lower labels in *rec* (x -axis) and the clusters (y -axis).⁷ <http://korpus.uib.no/icame/manuals/BROWN/INDEX.HTML>

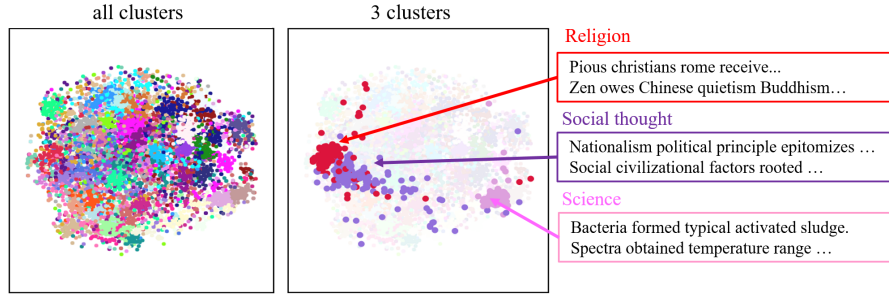


Fig. 9: Visualization of **Brown** corpus using NPDV(t -SNE). The points are colored by the estimated clusters. The right is an entire plot, and the left shows three clusters and their sample sentences.

within a cluster because each label represents a single topic. Fig. 9 shows a scatter plot of the **Brown** corpus. We discovered that some clusters shared topics, such as religion, social thought, and science. Furthermore, similar themes (religion and social thought) were placed close, whereas different themes (science) were distant, based on the first two themes. Therefore, the distance between clusters reflects the similarity of themes.

From these visualizations, we found that a common topic exists within a cluster and the distance between clusters reflects the similarity of topics. As topics can be considered as intrinsic clusters in a dataset, NPDV(t -SNE) can help to reveal clusters and grasp their similarities.

7 Conclusion

We proposed a nonparametric Bayesian latent variable model, NN-iWMM, and an associated visualization method, NPDV. NN-iWMM determines the layer widths, the dimensionality of latent space, and the number of clusters that are critical to model the latent space without tuning, while leveraging the power of neural networks implicitly. NPDV estimates the optimal latent coordinates to learn visual coordinates by integrating NN-iWMM and a visualization method. Additionally, we introduced NPDV(MF) and NPDV(t -SNE). Both methods enable to visualize the internal structure of dataset by utilizing the estimated clusters. Simulation studies demonstrated that NPDV(MF) infers the intrinsic latent manifold better than the existing methods. Real data experiments demonstrated that NPDV(t -SNE) outperforms conventional methods and shows comparable accuracy with a well-tuned neural clustering model. Furthermore, it shows two preferable properties in unsupervised settings: (1) NPDV(t -SNE) takes considerably less training time than the neural clustering model and (2) it has the ability to revealing plausible clusters without label information..

In this paper, we limit ourselves to study the properties of NPDV in the case of using matrix factorization or t -SNE. By considering combination with other methods, we expect to improve the accuracy and computing efficiency of NPDV.

References

1. van der Maaten and Hinton, G.: Visualizing data using *t*-SNE, *Journal of Machine Learning Research* **9**(1), 2579–2605, (2008)
2. Lee, J. H., Bahri, Y., Novak, R., Schoenholz, S. , Pennington, J. and Sohl-Dickstein, J.: Deep neural networks as gaussian processes, *International Conference on Learning Representation* 2018 **48** 478–487 (2018)
3. Mackay, D. J. C: Bayesian Non-Linear Modeling for the Prediction Competition, *ASHRAE Transaction***100**(2) 1053–1062 (1994)
4. Rasmussen, C. E.: The infinite Gaussian mixture model, *Proceedings of the 12th International Conference on Neural Information Processing Systems*, 554–560.
5. Kruscal, J. B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, *Psychometrika* **29** 1–27 (1964)
6. Tenebaum, J. B., de Silva, J. C., Langford, A: A global geometric framework for nonlinear dimensionality reduction, *Science* **290**(5500) 2319–2323 (2000)
7. Hinto, G. E. and Roweis S.:Stochastic Neighbor Embedding, *Advances in Neural Information Processing Systems* 15 (2002)
8. Tang, J., Liu, J., Zhang, M. and Mei, Q.: Visualizing large-scale and highdimensional data. the 25th International Conference on the World Wide Web 287–297 (2016)
9. Aleksandr, A. and Maxim, P.: NCVis: Noise Contrastive Approach for Scalable Visualization, *Proceedings of The Web Conference 2020* 2941–2947 (2020)
10. McInnes, L., Healy, J., Saul, N. and Großberge, L:UMAP : Uniform Manifold Approximation and Projection, *The Journal of Open Source Software* **3**(29), 2579–2605 (2018)
11. Hadsell, R., Chopra, S. and LeCun, Y: Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2** 1735–1742 (2006)
12. van der Maaten and Weinberger, K.: Stochastic triplet embedding. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*, 1–6 (2012).
13. Wilber, M.J., Kwak, I. S., Kriegman D. J., and Belongie, S.] Learning concept embeddings with combined human-machine expertise. In *Proceedings of the IEEE International Conference on Computer Vision***2**, pages 981–989. (2015).
14. Ehsan, A. and Manfred K. W: TriMap: Large-scale Dimensionality Reduction Using Triplets, *arXiv preprint arXiv:1910.00204* (2019)
15. Wang, Y., Huang, H. M., Rudin, C. and Shaposhnik, Y.: Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap, and PaCMAP for Data Visualization, *Journal of Machine Learning Research* **2021**(201) 1–73 (2021)
16. Wallach, I. and Liliean, R. : The Protein-Small-Molecule Database, A Non-Redundant Structural Resource for the Analysis of Protein-Ligand Binding, *Bioinformatics***25**(5) 615–620 (2010)
17. Hamel, P. and Eck, D.: Learning Features from Music Audio with Deep Belief Networks, *Proceedings of the International Society for Music Information Retrieval Conference*: 339–344 (2010)
18. Geng, X.; Zhan, De-Chuan. and Zhou, Zhi-Hua: Supervised nonlinear dimensionality reduction for visualization and classification, *IEEE Transactions on Systems, Man, and Cybernetics* **35**(6) 1098–1107 (2005)
19. Venna, A.; Peltonen, J.; Nybo, K.; Aidos, H. and Kaski, S.:Information Retrieval Perspective to Nonlinear Dimensionality Reduction for Data Visualization, *Journal of Machine Learning Research* **11**(13) 451–490 (2010)

20. Zheng, J.; Hangke Zhang, H.; Cattani, C. and Wang, W.: Dimensionality Reduction by Supervised Neighbor Embedding Using Laplacian Search. *Biomedical Signal Processing and Modeling Complexity of Living Systems 2014* (2014)
21. Xie, J., d Girshick, R. and Farhadi, A.: Unsupervised deep embedding for clustering analysis, *Proceedings of the 33rd International Conference on International Conference on Machine Learning* **48** 478–487 (2016)
22. Fard, M. N. and Thonet, T. and Gaussier, E.: Deep k -means: Jointly clustering with k -means and learning representations, *ArXiv:1806.10069*, (2018)
23. Yang, X., Yan, Y., Huang, K. and Zhang, R.: VSB-DVM: An end-to-end Bayesian nonparametric generalization of deep variational Mixture Model, *2019 IEEE International Conference on Data Mining* (2019)
24. Iwata, T., Duvenaud, D. and Ghahramani, Z.: Warped mixtures for nonparametric cluster shapes, *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence* 311–320 (2013)
25. Lawrence, N. D.: Probabilistic non-linear principal component analysis with Gaussian process latent variable models, *Journal of Machine Learning Research* **6** 1783–1816 (2004)
26. Rasmussen, C. E. and Williams C. K. I.: *Gaussian Processes for Machine Learning*, The MIT Press, (2006)
27. Ferguson, T. S.: A Bayesian analysis of some nonparametric problems. **1** 209–230, (1973)
28. Cho, Y. and Saul, L. K.: Kernel methods for deep learning, *Advances in Neural Information Processing Systems* **22**, 342–350 (2009)
29. J. Sethuraman: Constructive definition of Dirichlet process, *Statistical Sinica* **4**(2) 639–650 (1994)
30. Zhu, J., Chen, N. and Xing, E. P.: Bayesian inference with posterior regularization and applications to infinite latent SVMs, *Journal of Machine Learning* **15**(1) 1799–1847 (2014)
31. Zellner, A.: Optimal information processing and Bayestheorem, *American Statistician* **42**(4) 278–280 (1988)
32. Bishop, C. M.: *Pattern recognition and machine learning*, Springer (2006)
33. Blei, D. and Jordan, M.: Variational inference for Dirichlet process mixtures, *Journal of Bayesian Analysis* **1**(1) 121–144 (2006)
34. Kingma, D. P. and Welling, M.: Auto-encoding variational Bayes, *Proceedings of the 2nd International Conference on Learning Representations* (2013)
35. Titsias, M. K.; and Lawrence, N. D.: Bayesian Gaussian process latent variable model, *Proceedings of the 13th International Workshop on Artificial Intelligence and Statistics* **9** 844–851 (2010)
36. Akiba, T., Sano, S., Yanase, T., Ohta, T. and Koyama, M: Optuna: A next-generation hyperparameter optimization framework, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, KDD '19* 2623–2631 (2019)
37. Arora, S., Liang, Y. and Ma, T. A simple but tough-to-beat baseline for sentence embeddings, *International Conference on Learning Representations* (2017)