

## Research Article

# A Gaussian Process Latent Variable Model for Subspace Clustering

Shangfang Li 

*School of Mathematics and Statistics, Yulin Normal University, Yulin 537000, Guangxi, China*

Correspondence should be addressed to Shangfang Li; [monklsf@126.com](mailto:monklsf@126.com)

Received 27 September 2020; Revised 6 December 2020; Accepted 19 December 2020; Published 13 January 2021

Academic Editor: Zhouchao Wei

Copyright © 2021 Shangfang Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Effective feature representation is the key to success of machine learning applications. Recently, many feature learning models have been proposed. Among these models, the Gaussian process latent variable model (GPLVM) for nonlinear feature learning has received much attention because of its superior performance. However, most of the existing GPLVMs are mainly designed for classification and regression tasks, thus cannot be used in data clustering task. To address this issue and extend the application scope, this paper proposes a novel GPLVM for clustering (C-GPLVM). Specifically, by combining GPLVM with the subspace clustering method, our C-GPLVM can obtain more representative latent variable for clustering. Moreover, it can directly predict the new samples by introducing a back constraint in the model, thus being more suitable for big data learning tasks such as analysis of chaotic time series and so on. In the experiment, we compare it with the related GPLVMs and clustering algorithms. The experimental results show that the proposed model not only inherits the feature learning ability of GPLVM but also has superior clustering accuracy.

## 1. Introduction

In machine learning tasks, data are often distributed in a high-dimensional space and have many redundant features. Training machine learning models in such high-dimensional data may result in not only higher computational and storage complexities but also the model overfitting problem [1]. Existing research studies have shown that high-dimensional data are often embedded in low-dimensional manifold. We therefore can utilize the dimension reduction and feature learning method to learn the low-dimensional manifold and obtain more representative feature for the improvements of machine learning model accuracy and efficiency. Thus, effective feature representation is the key to success of machine learning applications.

In the past decade, many related methods have been proposed, such as dictionary learning [2], autoencoder [3], Gaussian process latent variable model (GPLVM) [4], Iso-map [5], and locally linear embedding [6]. Among these models, the GPLVM for nonlinear feature learning has received extensive attention because of its superior feature learning ability and has been used in many applications such

as dynamical system [7], modelling and control of nonlinear system [8]. Given a few training samples, it can effectively learn the low-dimensional manifold that is embedded in the high-dimensional space, thus has been widely used in the dimension reduction and data visualization tasks [9, 10].

Although has the abovementioned advantages, the conventional GPLVM is just a fully unsupervised feature learning model, thus cannot meet the demands of real-world applications, when dealing with specific machine learning tasks such as analysis of chaotic time series, dynamical system [7], and modelling and control of nonlinear system in which we also observe the response values of the inputs. How to modify the GPLVM and improve its performance is the key content of the related research studies. To date, the extensions of this model mainly focus on the supervised and unsupervised learning methods [9, 11, 12]. These methods assume that apart from the input features, we also observe labels of the samples. By their extensions, the GPLVM can effectively utilize the supervised information to improve the classification accuracy of the learned latent variables. However, in the real-world applications, we may also deal with unsupervised clustering tasks in which we cannot

obtain the label information or any other auxiliary information, thus bringing more challenges to application of the GPLVM in clustering tasks.

In order to address the abovementioned issues, this paper proposes a fusion model that combines the GPLVM with the subspace clustering model [13] to simultaneously obtain more representative features and accurate clustering results. Moreover, we also use the back constraint trick [14] in the model, which makes the model predict new samples directly and more suitable for big data learning tasks such as analysis of chaotic time series. In the experiment, we verify the performance of the proposed model on multiple datasets. The experimental results show that our model has much superior clustering performance than the other related models.

## 2. Related Work

**2.1. Gaussian Process Latent Variable Model.** The GPLVM is a fully unsupervised and nonlinear latent variable model. In this model, given  $N$  observed samples  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times D}$  (where  $\mathbf{x}_n \in \mathbb{R}^D$  denotes the  $n^{\text{th}}$  training sample), our objective is to learn the corresponding low-dimensional latent variable. In this paper, we use  $\mathbf{h}_n \in \mathbb{R}^Q$  ( $Q \ll D$ ) to denote the latent variable of  $\mathbf{x}_n$ . Obviously, the GPLVM can realize dimension reduction by learning the latent variables. Specifically, the GPLVM assumes that the generation process of  $\mathbf{x}_n$  as follows:

$$x_{nd} = f_d(\mathbf{h}_n) + \epsilon_{nd}, \quad (1)$$

where  $x_{nd}$  is the  $d^{\text{th}}$  feature of the  $n^{\text{th}}$  training sample,  $\epsilon_{nd}$  is the noise term that follows a Gaussian distribution  $p(\epsilon_{nd}) = N(\epsilon_{nd}|0, \sigma^2)$ , and  $f_d(\cdot)$  is a function that follows a Gaussian process prior. We use  $\mathbf{f}_d$  to denote the  $N$  outputs of  $f_d(\cdot)$  with inputs  $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_N]^T$ . Thus, we have  $\mathbf{f}_d \sim N(\mathbf{0}, \mathbf{K})$ , where  $\mathbf{K}$  is the kernel matrix that is computed by using kernel function  $k(\cdot, \cdot)$  on the latent variables in  $\mathbf{H}$ . The  $i^{\text{th}}$  row and  $j^{\text{th}}$  column element of  $\mathbf{K}$  is computed as  $K_{ij} = k(\mathbf{h}_i, \mathbf{h}_j)$ . By integrating out the intermediate variable  $\mathbf{f}_d$ , we can obtain the following marginal likelihood function:

$$p(\mathbf{X}|\mathbf{H}, \theta) = \prod_{d=1}^D p(\mathbf{x}_{:,d}|\mathbf{H}, \theta) = \prod_{d=1}^D \frac{\exp\left(-1/2\mathbf{x}_{:,d}^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{x}_{:,d}\right)}{(2\pi)^{N/2}|\mathbf{K} + \sigma^2\mathbf{I}|^{1/2}}, \quad (2)$$

where  $\theta$  denotes the hyperparameters involved in the kernel function and noise distribution. In the model optimization process, the GPLVM learns the latent variable and hyperparameters jointly by maximizing the above likelihood function and obtains the low-dimensional representation finally.

From the abovementioned generation process, as a fully unsupervised dimension reduction model, the GPLVM cannot embed auxiliary information when dealing with specific machine learning tasks, thus cannot meet demands of real-world application. For example, in analysis of chaotic time series, data of similar time will have similar features. If it can utilize this knowledge, the GPLVM will learn more

representative features for the task and significantly improve the prediction accuracy. The existing methods for the extension of GPLVM mainly focus on embedding supervised information to improve its classification and regression accuracy, for example, the discriminative GPLVM (D-GPLVM) and supervised GPLVM (S-GPLVM). For the extension to the clustering task, the related works is much fewer. The existing unsupervised GPLVM just focuses on how to preserve the local distance and learn better latent variables or features. For example, local preserving projection GPLVM (LPP-GPLVM) combines the objective of local preserving projection with that of the GPLVM, thus simultaneously learning the low-dimensional representation and preserving the local structures [15]. The GPLVM with back constraints (B-GPLVM) introduces a back-constraint (from observed space to latent space) into the GPLVM. By this way, it can also realize the preservation of local distance.

**2.2. Subspace Clustering.** The goal of subspace clustering is to segment a set of data samples into different subspaces; thus, similar samples are in the same subspace, while dissimilar samples are in different subspaces. Over the past decade, subspace clustering has been used in various clustering tasks and many well-designed algorithms have been proposed such as Gaussian mixture model- (GMM-) based methods [16, 17], matrix factorization- (MF-) based methods [18, 19], algebra-based methods [20], and spectral clustering methods [13, 21, 22]. Among these models, the subspace clustering method based on spectral clustering has been widely applied because of its concise implementation process and reliable performance. It uses low-rank representation to construct the affinity matrix of the spectral clustering. Its objective is to find the low-rank representation of input data  $\mathbf{X}$  by optimizing the following function:

$$\begin{aligned} & \operatorname{argmin}_{\mathbf{Z}} \operatorname{rank}(\mathbf{Z}), \\ & \text{s.t. } \mathbf{X} = \mathbf{Z}\mathbf{X}, \end{aligned} \quad (3)$$

where we assume that each sample can be expressed by the linear combination of other samples. The above low-rank penalty term can be considered as a global constraint on the subspace structure of samples and makes similar samples have similar weights. In general, we can use the following nuclear norm to replace the penalty term:

$$\begin{aligned} & \operatorname{argmin}_{\mathbf{Z}} \|\mathbf{Z}\|_*, \\ & \text{s.t. } \mathbf{X} = \mathbf{Z}\mathbf{X}, \end{aligned} \quad (4)$$

where we use the nuclear norm  $\|\mathbf{Z}\|_*$  to approximate the rank of  $\mathbf{Z}$ . Considering that the data often contain noise, we use the following formulation to learn the self-representation matrix  $\mathbf{Z}$ :

$$\operatorname{argmin}_{\mathbf{Z}} \frac{1}{2} \|\mathbf{X} - \mathbf{Z}\mathbf{X}\|_F^2 + \lambda \|\mathbf{Z}\|_*. \quad (5)$$

In low-rank subspace clustering, we can first construct the affinity matrix  $\mathbf{W}$  and the Laplace matrix  $\mathbf{L}$  and then use spectral clustering to cluster the data.  $\mathbf{W}$  and  $\mathbf{L}$  can be constructed as follows:

$$\begin{aligned}\mathbf{W} &= |\mathbf{Z}| + |\mathbf{Z}|^T, \\ \mathbf{L} &= \mathbf{D} - \mathbf{W},\end{aligned}\quad (6)$$

where  $\mathbf{D}$  denotes a diagonal matrix and  $D_{ii} = \sum_{j=1}^N W_{ij}$ . After obtaining the Laplace matrix, we can optimize the following objective function to obtain the latent variable  $\mathbf{H} \in \mathbb{R}^{N \times Q}$ :

$$\begin{aligned}\arg \min_{\mathbf{H}} \quad & tr(\mathbf{H}^T \mathbf{L} \mathbf{H}), \\ \text{s.t.} \quad & \mathbf{H}^T \mathbf{H} = \mathbf{I}.\end{aligned}\quad (7)$$

Obviously,  $\mathbf{H}$  is composed of the eigenvectors corresponding to the  $Q$  smallest eigenvalues. At last, we can run the k-means algorithm on the learned  $\mathbf{H}$  and obtain the clustering result.

### 3. Model Construction and Optimization

*3.1. Designing of the Gaussian Process Latent Variable Clustering Model.* Assuming that there are  $N$  observed samples denoted as  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$ , our goal is to learn the low-dimensional latent variable  $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_N]^T$  corresponded to these observed variables and make the latent variable have more superior clustering performance (i.e., make the common clustering algorithms obtain accurate clustering result on the learned  $\mathbf{H}$ ).

In order to achieve the above goal, we assume that the latent variable  $\mathbf{H}$  has the following prior distribution:

$$p(\mathbf{H}) = \frac{1}{Z_0} \exp(-\lambda_1 R(\mathbf{H})), \quad (8)$$

where  $p_0$  is a constant that makes  $\int_{\mathbf{H}} p(\mathbf{H}) d\mathbf{H} = 1$  and  $R(\mathbf{H})$  has the following form:

$$R(\mathbf{H}) = \sum_{i=1}^N \sum_{j=1}^N \|\mathbf{h}_i - \mathbf{h}_j\|_2^2 W_{ij}, \quad (9)$$

where  $W_{ij}$  is the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column element of the affinity matrix  $\mathbf{W}$ . Equation (9) often can be written as follows:

$$R(\mathbf{H}) = \mathbf{H}^T \mathbf{L} \mathbf{H}. \quad (10)$$

In this paper, we assume that the generation process observed variables from the latent variable can be constructed by conditional distribution  $p(\mathbf{X}|\mathbf{H})$ . Thus, from the Bayes formulation, we can obtain the posterior distribution of latent variable as

$$p(\mathbf{H}|\mathbf{X}) = \frac{p(\mathbf{X}|\mathbf{H})p(\mathbf{H})}{p(\mathbf{X})}. \quad (11)$$

Since  $p(\mathbf{X})$  is a constant, we therefore can obtain the optimal latent variable by maximizing the following joint marginal distribution:

$$\arg \max_{\mathbf{H}} p(\mathbf{X}|\mathbf{H}) p(\mathbf{H}). \quad (12)$$

To introduce the GPLVM into this model, we assume that  $\mathbf{X}$  is generated by latent function which follows a Gaussian process prior. Thus, equation (12) can be written as

$$\arg \min_{\mathbf{H}, \theta, \sigma} \frac{DN}{2} \ln |\mathbf{K} + \sigma^2 \mathbf{I}| + \frac{1}{2} tr(\mathbf{X}(\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{X}^T) + \lambda_1 R(\mathbf{H}), \quad (13)$$

where  $\theta$  denotes the hyperparameter that is involved in the kernel function and  $\sigma^2$  denotes the variance of Gaussian noise distribution.

By the above modelling process, the GPLVM can effectively embed the sample similarity information when learning the latent variable, thus improving its latent variable clustering ability. However, how to learn the affinity matrix is still an urgent problem of this paper and other related algorithms such as self-representation learning and subspace clustering. In this paper, we borrow the idea of low-rank self-representation learning and introduce the following low-rank subspace constraints into the model:

$$\mathbf{X} = \mathbf{Z}\mathbf{X}, \quad \text{rank}(\mathbf{Z}) \leq c, \mathbf{Z} \geq 0. \quad (14)$$

It is worth noting that, in this paper, we assume that  $\mathbf{W} = \mathbf{Z}$ , i.e., we directly use matrix  $\mathbf{Z}$  as the affinity matrix. This setting is the same as that of [23] and its role is similar to the affinity matrix of the original subspace clustering. This C-GPLVM is very similar to the LPP-GPLVM. However, in the LPP-GPLVM, the Laplace matrix is fixed. Different from LPP-GPLVM, the Laplace matrix in our C-GPLVM can be learned in the training process. Thus, our C-GPLVM has more superior performance than the LPP-GPLVM.

One important limitation of GPLVM and self-representation is that they cannot effectively predict the new samples. To mitigate this problem, we introduce a back constraint on the proposed model. Thus, given a new sample, it can effectively predict the corresponding low-dimensional latent variable using the constraint function. Specifically, given an observed sample  $\mathbf{x}_n$ , we assume that we can use a function  $g(\cdot)$  to obtain latent variable  $\mathbf{h}_n$ :

$$\mathbf{h}_n = g(\mathbf{x}_n), \quad (15)$$

where  $g(\cdot)$  is the neural network function with learnable parameter  $\Phi$ . At last, we obtain the objective of the proposed model as follows:

$$\begin{aligned}\arg \min_{\Phi, \theta, \sigma} \quad & \frac{DN}{2} \ln |\mathbf{K} + \sigma^2 \mathbf{I}| + \frac{1}{2} tr(\mathbf{X}(\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{X}^T) + \lambda_1 R(\mathbf{H}), \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{Z}\mathbf{X}, \quad \text{rank}(\mathbf{Z}) \leq c, \mathbf{Z} \geq 0.\end{aligned}\quad (16)$$

The whole model structure is shown in Figure 1.

### 4. Model Optimization

In order to optimize (16), we transform it into the following optimization problem:

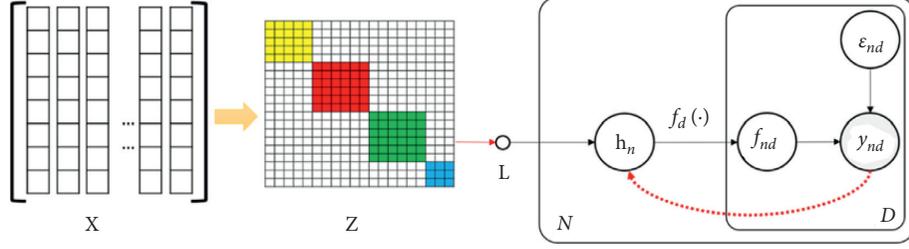


FIGURE 1: The structure of C-GPLVM.

$$\begin{aligned} \arg \min_{\Phi, \theta, \sigma} \frac{DN}{2} \ln |\mathbf{K} + \sigma^2 \mathbf{I}| + \frac{1}{2} \text{tr} \left( \mathbf{X} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{X}^T \right) + \lambda_1 R(\mathbf{H}) \\ + \lambda_2 \|\mathbf{X} - \mathbf{Z}\mathbf{X}\|_2^2 + \lambda_3 \|\mathbf{Z}\|_*, \\ \text{s.t. } \mathbf{Z} \geq 0, \end{aligned} \quad (17)$$

where  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are regularization terms. By this formulation, we can use the alternating iterative optimization method to learn all the parameters. First, we fix  $\mathbf{Z}$  and write (17) as

$$\begin{aligned} \arg \min_{\Phi, \theta, \sigma} L_p = \arg \min_{\Phi, \theta, \sigma} \frac{DN}{2} \ln |\mathbf{K} + \sigma^2 \mathbf{I}| + \frac{1}{2} \text{tr} \left( \mathbf{X} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{X}^T \right) \\ + \lambda_1 R(\mathbf{H}). \end{aligned} \quad (18)$$

This problem can be solved effectively by using gradient-based methods, and its gradient with respect to  $\Phi$  can be computed as

$$\frac{\partial L_p}{\partial \Phi} = \frac{\partial L_p}{\partial \mathbf{K}} \frac{\partial \mathbf{K}}{\partial \Phi} + \lambda_1 \frac{\partial R(\mathbf{H})}{\partial \Phi}. \quad (19)$$

The gradients with respect to  $\theta$  and  $\sigma$  are similar to the above formulation. For the sake of brevity, we have omitted their derivation processes. We then can fix  $\Phi$ ,  $\theta$  and  $\sigma$  and write (17) as

$$\begin{aligned} \arg \min_{\mathbf{Z}} \lambda_1 R(\mathbf{H}) + \lambda_2 \|\mathbf{X} - \mathbf{Z}\mathbf{X}\|_2^2 + \lambda_3 \|\mathbf{Z}\|_*, \\ \text{s.t. } \mathbf{Z} \geq 0. \end{aligned} \quad (20)$$

The gradient of the first term with respect to  $\mathbf{Z}$  can be computed as

$$\frac{\partial \lambda_1 R(\mathbf{H})}{\partial Z_{n,:}} = \mathbf{e}^T \sum_{j=1}^N \|\mathbf{h}_n - \mathbf{h}_j\|_2^2, \quad (21)$$

where  $Z_{n,:}$  denotes the  $n^{\text{th}}$  row of matrix  $\mathbf{Z}$  and  $\mathbf{e} \in \mathbb{R}^N$  denotes the vector whose element is 1. The gradient of the second term can be computed as

$$\frac{\partial \lambda_2 \|\mathbf{X} - \mathbf{Z}\mathbf{X}\|_2^2}{\partial \mathbf{Z}} = \lambda_2 (\mathbf{X} - \mathbf{Z}\mathbf{X})\mathbf{X}^T. \quad (22)$$

The subgradient of the third term is

$$\frac{\partial \|\mathbf{X}\|_*}{\partial \mathbf{X}} = \mathbf{V} \text{sgn}(\mathbf{\Sigma}) \mathbf{U}^T. \quad (23)$$

By the above derivations, we can learn the whole model, as described in Algorithm 1. The main computation complexity of C-GPLVM is the inversion of the kernel matrix, which has a complexity of  $O(N^3)$ , where  $N$  is the number of training samples. The main storage complexity of this is the storage of kernel matrix, which has a complexity of  $O(N^2)$ . Thus, both the computation and storage complexities are the same as those of the conventional GPLVM.

## 5. Experiments and Analysis

*5.1. Experimental Setup.* To verify the effectiveness of C-GPLVM, we use 8 datasets in the experiments. The detailed information of these datasets is given in Table 1.

The YEAST is a dataset for the prediction of protein localization sites. The USPS is a digits dataset that was gathered at the Center of Excellence in Document Analysis and Recognition at SUNY Buffalo, as part of a project sponsored by the US Postal Service. YALE, JAFFE, and ORL are three face recognition datasets, as shown in Figure 2. TR11, TR41, and TR45 are three textual datasets.

In order to fully verify the advantage of C-GPLVM, we compare it to the related Gaussian process latent variable model (i.e., GPLVM, B-GPLVM, and LPP-GPLVM) and clustering methods, such as spectral clustering method (SC) [24], kernel spectral clustering (KSC) [25], and simplex sparse representation learning (SSR) [21]. All the kernel-based models (GPLVM, LPP-GPLVM, KSC, and C-GPLVM) used Radial Basis Function (RBF) as the kernel function. It is worth noting that some other kernel functions can also be used in the proposed model such as linear kernel, Laplacian kernel, and circular kernel. Furthermore, all the hyperparameters in these kernel functions can also be learned in the same form as described in the paper. During the experiment process, the hyperparameters  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are chosen from  $\{0.01, 0.1, 1, 10, 100\}$ . The hyperparameters involved in other models are set to be the same as those of original paper. In the experiment process, we use the Gaussian process toolkit (GPFlow) <sup>1</sup> to implement the GP-based model. Other related models are all implemented with python. All the algorithms are tested on the Windows computer with i7 9700 CPU, 16G RAM.

Input: training set  $\mathbf{X}$ , dimension  $Q$  of latent variable, hyperparameters  $\lambda_1, \lambda_2, \lambda_3$ .  
Output: parameter  $\Phi$  and the clustering result.

- (1) Pretrain the following model to initialize the latent variable  $\mathbf{Z}$ 

$$\mathbf{Z} = \operatorname{argmin}_{\mathbf{Z}} \lambda_2 \|\mathbf{X} - \mathbf{Z}\mathbf{X}\|_2^2 + \lambda_3 \|\mathbf{Z}\|_*$$
s.t.  $\mathbf{Z} \geq 0$
- (2) while ( $l_{\text{tol}} \geq 10^{-3}$ )
- (3) With fixed  $\mathbf{Z}$ , minimize (18) to obtain  $\Phi, \theta$ , and  $\sigma$ .
- (4) With fixed  $\Phi, \theta$ , and  $\sigma$ , minimize (20) to obtain  $\mathbf{Z}$ .
- (5) Calculate the difference  $l_{\text{tol}}$  of the objective function between the last two iterations
- (6) end while
- (7) Compute the latent variable  $\mathbf{H}$  based on the learned parameter  $\Phi$ .
- (8) Run  $k$ -means algorithm on  $\mathbf{H}$  to get the final clustering result.

ALGORITHM 1: The optimization process of C-GPLVM.

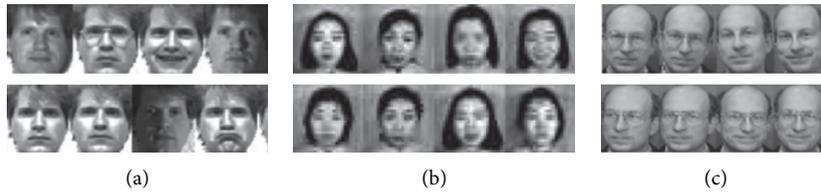


FIGURE 2: Samples in YALE, JAFFE, and ORL datasets.

TABLE 1: The detailed information of the datasets.

Dataset	# of sample	# of feature	# of classes
YALE	165	1024	15
JAFFE	213	676	10
ORL	400	1024	40
YEAST	1484	1470	10
USPS	1854	256	20
TR11	414	6429	9
TR41	878	7454	10
TR45	690	8261	10

TABLE 2: Accuracy.

	YALE	JAFFE	ORL	YEAST	USPS	TR11	TR41	TR45
GPLVM	29.80 ± 1.01	55.91 ± 2.94	21.58 ± 1.45	19.48 ± 1.49	23.14 ± 2.41	30.01 ± 2.58	42.87 ± 2.94	33.54 ± 2.31
B-GPLVM	32.12 ± 1.58	59.66 ± 2.14	25.17 ± 2.18	23.94 ± 1.54	28.14 ± 1.19	35.47 ± 1.50	48.99 ± 1.37	37.93 ± 2.54
LPP-GPLVM	36.14 ± 1.25	71.49 ± 2.75	38.97 ± 1.94	29.88 ± 1.73	61.71 ± 2.21	51.44 ± 2.20	49.79 ± 1.40	51.83 ± 2.91
SC	50.32 ± 1.23	73.55 ± 2.33	57.96 ± 1.45	<b>35.91 ± 1.34</b>	36.45 ± 2.90	52.69 ± 3.32	61.74 ± 1.8	55.91 ± 1.96
KSC	37.21 ± 1.55	72.19 ± 1.98	39.74 ± 2.39	31.18 ± 1.83	63.78 ± 3.87	53.74 ± 1.86	51.12 ± 1.11	53.44 ± 1.99
SSR	53.59 ± 1.09	85.92 ± 2.34	68.22 ± 1.98	27.62 ± 1.92	<b>64.83 ± 2.39</b>	41.06 ± 2.33	65.78 ± 2.58	73.45 ± 1.36
C-GPLVM	<b>55.91 ± 1.38</b>	<b>86.14 ± 1.81</b>	<b>70.43 ± 1.30</b>	33.71 ± 1.42	63.16 ± 2.01	<b>55.89 ± 2.59</b>	<b>67.16 ± 2.14</b>	<b>75.25 ± 1.90</b>

TABLE 3: Purity.

	YALE	JAFFE	ORL	YEAST	USPS	TR11	TR41	TR45
GPLVM	30.48 ± 1.00	53.41 ± 1.84	38.14 ± 1.09	8.47 ± 1.71	21.10 ± 1.66	19.41 ± 1.10	30.11 ± 2.09	34.57 ± 1.49
B-GPLVM	33.19 ± 1.44	56.49 ± 1.58	49.18 ± 1.41	8.56 ± 2.61	25.17 ± 2.17	23.61 ± 1.97	34.17 ± 2.37	38.44 ± 1.56
LPP-GPLVM	41.57 ± 1.11	67.18 ± 2.10	67.18 ± 1.94	12.80 ± 1.54	26.49 ± 1.43	28.15 ± 2.15	53.91 ± 2.34	45.61 ± 2.01
SC	53.24 ± 1.25	82.32 ± 1.02	74.41 ± 1.47	21.47 ± 3.14	28.91 ± 1.84	42.89 ± 1.58	61.84 ± 2.14	48.17 ± 2.17
KSC	45.91 ± 2.17	71.82 ± 1.84	55.91 ± 1.19	11.14 ± 2.14	49.83 ± 1.36	45.77 ± 1.91	44.77 ± 2.91	44.18 ± 2.22
SSR	54.66 ± 2.14	91.10 ± 1.24	83.16 ± 1.84	16.91 ± 2.17	71.99 ± 1.47	28.67 ± 1.43	56.90 ± 2.79	65.10 ± 1.97
C-GPLVM	55.17 ± 2.38	91.84 ± 1.33	82.71 ± 1.94	21.04 ± 2.38	70.61 ± 1.89	47.94 ± 1.95	62.85 ± 2.57	66.73 ± 2.32

TABLE 4: Normalized mutual information.

	YALE	JAFFE	ORL	YEAST	USPS	TR11	TR41	TR45
GPLVM	30.41 ± 1.40	59.17 ± 1.09	31.90 ± 1.29	19.80 ± 1.41	21.36 ± 1.00	34.82 ± 1.06	39.40 ± 1.61	37.21 ± 1.27
B-GPLVM	32.49 ± 1.66	63.99 ± 1.81	32.81 ± 1.11	28.39 ± 1.58	30.58 ± 1.07	40.59 ± 1.40	49.55 ± 2.06	46.79 ± 1.41
LPP-GPLVM	40.85 ± 1.29	68.46 ± 1.43	40.90 ± 1.09	31.63 ± 1.47	34.83 ± 1.06	48.97 ± 1.38	51.63 ± 2.44	52.88 ± 1.67
SC	50.49 ± 1.84	75.91 ± 2.41	60.49 ± 1.84	<b>53.71 ± 2.17</b>	36.49 ± 1.61	57.39 ± 1.20	72.63 ± 2.19	61.98 ± 1.94
KSC	43.50 ± 1.32	77.35 ± 2.01	41.53 ± 1.44	32.59 ± 2.98	61.29 ± 1.65	51.09 ± 1.69	54.81 ± 2.81	55.71 ± 2.11
SSR	57.39 ± 1.09	95.50 ± 1.86	76.41 ± 1.00	43.16 ± 2.73	<b>73.24 ± 1.10</b>	83.77 ± 1.43	75.63 ± 2.37	82.69 ± 2.43
C-GPLVM	<b>58.41 ± 2.17</b>	<b>96.01 ± 2.38</b>	<b>77.25 ± 1.53</b>	51.42 ± 1.85	71.11 ± 1.83	<b>85.64 ± 1.82</b>	<b>77.31 ± 2.82</b>	<b>83.11 ± 2.28</b>

5.2. *Clustering Results and Analysis.* In the experiments, we use clustering accuracy, purity, and normalized mutual information (NMI) as the clustering measurement. At the clustering stage, the latent variables learned by different methods are used as inputs and the k-means algorithm is used to obtain the final clustering methods. The dimension and the number of clusters are set to be the same as the number of classes. At the same time, in order to mitigate the initial value sensitivity problem of k-means method, we randomly initialize and run the k-means method 20 times. Finally, we calculate the mean and standard deviation of these 20 experiments. The experimental results are shown in Tables 2–4, where the best results are given in bold.

From Tables 2–4, we can observe that the GPLVM, as an unsupervised dimension reduction model, usually obtain latent variables that have poor clustering performance. The B-GPLVM and LPP-GPLVM can preserve the local distance of samples during the feature learning process, thus obtaining more representative latent variables. Meanwhile, the LPP-GPLVM obtains much better result than the B-GPLVM which indicates that graph Laplace regularization is more suitable for clustering than back constraints. In general, spectral clustering and subspace clustering methods have better performance than the GPLVM. As we can see, SC, KSC, and SSR outperform GPLVM, B-GPLVM, and LPP-GPLVM. In this paper, the proposed C-GPLVM combines the subspace clustering with GPLVM, thus effectively improving the clustering performance of the GPLVM. As shown in the experimental results, the C-GPLVM has more superior clustering result than other related models in most cases.

## 6. Conclusion and Future Work

This paper proposes a joint model by combining the low-rank subspace with the back constraint GPLVM to address the poor clustering performance problem of the conventional GPLVM. The proposed C-GPLVM can not only obtain low-dimensional latent variables but also directly predict the new samples, thus effectively extending the application scope of GPLVM on tasks such as analysis of chaotic time series. The experimental results show that the C-GPLVM has much better latent variable learning ability and superior clustering performance. In the future work, we will further extend the C-GPLVM to make it suitable for much bigger dataset and supervise tasks such as classification and regression, improving its efficiency and application scope.

## Data Availability

The experimental data used to support the findings of this study have been deposited in the UCI repository (<https://archive.ics.uci.edu/ml/index.php>).

## Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the Project of Yulin Normal University Research under Grant 2015YJYB03.

## References

- [1] X. Li and L. Shu, “Kernel based nonlinear dimensionality reduction for microarray gene expression data analysis,” *Expert Systems with Applications*, vol. 36, no. 4, pp. 7644–7650, 2009.
- [2] K. Kreutz-Delgado, B. D. Murray, K. Engan, T.-W. Lee, and T. J. Sejnowski, “Dictionary learning algorithms for sparse representation,” *Neural Computation*, vol. 15, no. 2, pp. 349–396, 2003.
- [3] L. Fengfu, Q. Hong, and Z. Bo, “Discriminatively boosted image clustering with fully convolutional auto-encoders,” *Pattern Recognition*, vol. 83, pp. 161–173, 2018.
- [4] N. D. Lawrence, “Probabilistic non-linear principal component analysis with Gaussian process latent variable models,” *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.
- [5] J. B. Tenenbaum, V. De Silva, J. Langford et al., “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [6] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [7] J. M. Wang, A. Hertzmann, D. M. Blei et al., “Gaussian process dynamical models,” in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, pp. 1441–1448, Vancouver, Canada, December 2005.
- [8] J. Hall, C. E. Rasmussen, J. M. Maciejowski et al., “Modelling and control of nonlinear systems using gaussian processes with partial model information,” in *Proceedings of the Conference on Decision and Control*, pp. 5266–5271, Orlando, FL, USA, December 2012.
- [9] L. Cai, L. Huang, C. Liu et al., “Age estimation based on improved discriminative Gaussian process latent variable model,” *Multimedia Tools and Applications*, vol. 75, no. 19, pp. 11977–11994, 2016.

- [10] C. Lu and X. Tang, "Surpassing human-level face verification performance on LFW with gaussianface," in *Proceedings of the Computer Vision and Pattern Recognition*, Columbus, OH, USA, June 2014.
- [11] X. Gao, X. Wang, D. Tao et al., "Supervised Gaussian process latent variable model for dimensionality reduction," *Systems Man and Cybernetics*, vol. 41, no. 2, pp. 425–434, 2011.
- [12] S. Eleftheriadis, O. Rudovic, M. Pantic et al., "Discriminative shared Gaussian processes for multiview and view-invariant facial expression recognition," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 189–204, 2015.
- [13] G. Liu, Z. Lin, Y. Yu et al., "Robust subspace segmentation by low-rank representation," in *Proceedings of the International Conference on Machine Learning*, pp. 663–670, Haifa, Israel, June 2010.
- [14] N. D. Lawrence and J. Quinonero-candela, "Local distance preservation in the GP-LVM through back constraints," in *Proceedings of the International Conference on Machine Learning*, pp. 513–520, New York, NY, USA, June 2006.
- [15] W. Xiu, "A latent variable model based on local preservation," *Pattern Recognition and Artificial Intelligence*, vol. 23, no. 3, pp. 369–375, 2010.
- [16] P. S. Bradley and O. L. Mangasarian, "K-plane clustering," *Journal of Global Optimization*, vol. 16, no. 1, pp. 23–32, 2000.
- [17] P. Tseng, "Nearest q-Flat to m Points," *Journal of Optimization Theory and Applications*, vol. 105, no. 1, pp. 249–252, 2000.
- [18] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 159–179, 1998.
- [19] C. W. Gear, "Multibody grouping from motion images," *International Journal of Computer Vision*, vol. 29, no. 2, pp. 133–150, 1998.
- [20] R. Vidal, Y. Yi Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1945–1959, 2005.
- [21] J. Huang, F. Nie, H. Huang et al., "A new simplex sparse learning model to measure data similarity for clustering," in *Proceedings of the International conference on Artificial Intelligence*, pp. 3569–3575, New Delhi, India, February 2015.
- [22] S. Luo, C. Zhang, W. Zhang et al., "Consistent and specific multi-view subspace clustering," in *Proceedings of the National Conference on Artificial Intelligence*, pp. 3730–3737, Vancouver, Canada, February 2018.
- [23] Z. Kang, H. Xu, B. Wang, H. Zhu, and Z. Xu, "Clustering with similarity preserving," *Neurocomputing*, vol. 365, pp. 211–218, 2019.
- [24] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [25] C. Alzate and J. A. K. Suykens, "Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 32, no. 2, pp. 335–347, 2009.