Available online at www.sciencedirect.com

ScienceDirect

CAAI Transactions on Intelligence Technology 1 (2016) 366–376
http://www.journals.elsevier.com/caai-transactions-on-intelligence-technology/

Review Article

A review on Gaussian Process Latent Variable Models

Ping Li, Songcan Chen*

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Available online 14 November 2016

Abstract

Gaussian Process Latent Variable Model (GPLVM), as a flexible bayesian non-parametric modeling method, has been extensively studied and applied in many learning tasks such as Intrusion Detection, Image Reconstruction, Facial Expression Recognition, Human pose estimation and so on. In this paper, we give a review and analysis for GPLVM and its extensions. Firstly, we formulate basic GPLVM and discuss its relation to *Kernel Principal Components Analysis*. Secondly, we summarize its improvements or variants and propose a taxonomy of GPLVM related models in terms of the various strategies that be used. Thirdly, we provide the detailed formulations of the main GPLVMs that extensively developed based on the strategies described in the paper. Finally, we further give some challenges in next researches of GPLVM.

Keywords: GPLVM; Non-parametric method; Gaussian process

1. Introduction

In many machine learning tasks, we are often faced with various complex, particularly high dimensional, data/observations [1−3] for which our goal is to learn the low dimensional underlying patterns from those observations [4,5]. For example, in classification task [6−9], we want to identify a category of a new observation by a classifier learned from a set of training data. In clustering task, the goal is to group a set of observations in such a way that observations in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups [10,11], achieving to understand inherent (low dimensional) structure of a given data set.

Recently, many machine learning models have been proposed to address the above problems [1,3,12,13]. Among those models, latent variable models (LVMs) [3,14,15] as a kind of underlying patterns extraction methods, have been widely used in image recognition [16], information retrieval [17], speech recognition [18] and recommender systems [19]. A latent variable model generally refers to a statistical model that relates a set of variables (so-called manifest variables) to a set of latent variables under the assumption that the responses on the manifest variables are controlled by the latent variables. Furthermore, we can provide latent variables with various meanings for specific tasks. In *Dimension Reduction* (DR), we assume that the latent variables are the low dimensional representations of high dimensional samples. In clustering, the latent variables can be defined to represent the clustering membership of samples [20]. This flexible definition of latent variables has made LVMs widely be used in many machine learning tasks.

LVMs have a history of several decades and many machine learning models can be actually considered as its special cases or variants, e.g., neural networks [18], PCA [21], latent graphical models [3] and so on. Among these models, *Gaussian Process Variable Models* (GPLVMs) [15], as a large class of LVMs, have been explored and applied in many machine scenarios. They can be considered as the combination of LVM and a Bayesian non-parametric *Gaussian Process* (GP) [22] model. GP is a probabilistic model and has been extensively applied in many machine learning tasks such as regression [23−25], classification [6−8] and clustering [26].

* Corresponding author.
E-mail addresses: ping.li.nj@nuaa.edu.cn (P. Li), s.chen@nuaa.edu.cn (S. Chen).
Peer review under responsibility of Chongqing University of Technology.

http://dx.doi.org/10.1016/j.trit.2016.11.004

In general, we can consider these GP-based models to be a set of LVMs where each observed variable is the sum of the corresponding latent variable and noise. Different from other LVMs, these latent variables can be thought of as functional variables which are the noise-free form of observed variables. In LVMs, our goal is to learn the latent variables or the underlying pattern of data. While the above GP-based models try to infer the target variable of new sample by integrating out the latent variables. This is a major difference between GP-based model and other LVMs.

In order to infer the latent variables, GPLVM assumes that the functional variables are generated by GP from some low dimensional latent variables. It is these latent variables that we should inferred from data. In model inference, we can learn the latent variables by integrating out the functional variables and maximizing the log marginal likelihood. Although originally proposed for dimension reduction, GPLVM has been extended and widely used in many machine learning scenarios, such as Intrusion Detection [27], Image Reconstruction [28], Facial Expression Recognition [29], Human pose estimation [30], Age Estimation [31] and Image-Text Retrieval [32].

We can analyze the advantages of GPLVM from two aspects. Firstly, GPLVM can greatly benefit from the non-linear learning characteristic of GP which uses a non-linear kernel to replace the covariance matrix. Moreover, as a non-linear DR model, GPLVM has a strong link with *Kernel Principal Components Analysis* (KPCA) [33] (a popular DR method) and can be considered as a *Probabilistic Kernel Principal Components Analysis* (PKPCA). For such a link, we will discuss in next section. Secondly, most of the existing LVMs are parametric models in which there is a strong assumption on the projection function or data distribution. Such a parametric construction form partly loses flexibility in modeling. Therefore, in the past decades, many non-parametric machine learning methods have successively been proposed, such as Nearest Neighbor methods [34,35] and kernel estimates of probability densities [36,37]. GP and GPLVM can be treated as a class of Bayesian non-parametric model whose distribution-free form makes the models have a flexible structure which can grow in size to accommodate the complexity of the data.

Besides widely used in DR, GPLVM can also be extended to other machine learning tasks due to its characteristics below. Firstly, its distribution-free assumption on prior of latent variables provides us a lot of opportunities to improve it. Secondly, its generation process can be amenable to different tasks. Thirdly, we can also exert classical kernel methods for a further expansion of GPLVM, such as enhancing the scalability of the model, automatic selection of the feature dimension and so on. Despite GPLVM has been widely studied and extended, to our best knowledge, there has actually had no survey for those related models. So in this paper, we try to present a review and analysis of both GPLVM and its extensions.

The rest of this paper is organized as follows: In Section 2, we formulate the GPLVMs and discuss its relation to *Kernel Principal Components Analysis* (KPCA). In Section 3, we summarize its improvements or variants and propose a taxonomy of GPLVM related models. A specific review of GPLVM that extensively developed in the past decade is given in Section 4. Finally, in Section 5, we further give some challenges in next researches of GPLVM.

2. Gaussian process and Gaussian Process Latent Variable Model

2.1. Gaussian process

GP, as the a flexible Bayesian nonparametric model and the building block for GPLVM, has been widely used in many machine learning applications [38–40] for data analysis. In GP, we model a finite set of random function variables $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]^T$ as a joint Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{K}$, where $\boldsymbol{x}_i$ is the ith input. If the function f has a GP prior, we can write it as

$$f \sim \mathcal{GP}(\boldsymbol{\mu}, \boldsymbol{K}) \tag{1}$$

where in many cases we can specify a zero mean ($\boldsymbol{\mu} = \boldsymbol{0}$) and a kernel matrix $\boldsymbol{K}$ (with hyper-parameter $\boldsymbol{\theta}$) as covariance matrix. GP has been widely used in various machine learning scenarios such as regression, classification, clustering. In this section, we detailed the formulation of *Gaussian Process Regression* (GPR) to demonstrate the use of GP.

In GPR, our goal is to predict the response y^* of a new input $\boldsymbol{x}^*$, given a training dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ of N training samples, where $\boldsymbol{x}_i$ is the input variable and y_i is the corresponding continuous response variable. We model the response variable y_i as a noise-version of the function value $f(\boldsymbol{x}_i)$

$$y_i \sim \mathcal{N}\left(f(\boldsymbol{x}_i), \sigma^2\right) \tag{2}$$

where the distribution of noise is Gaussian $\mathcal{N}(0, \sigma^2)$ with variance σ^2. From the above definition, we can get the joint probability of the response variables and latent function variables $p(\boldsymbol{y},\boldsymbol{f}) = p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f})$. Then we can know that the distribution of the latent function value f^* is a Gaussian distribution with mean $\mu(\boldsymbol{x}^*)$ and variance $var(\boldsymbol{x}^*)$:

$$\begin{aligned} \mu(\boldsymbol{x}^*) &= \boldsymbol{k}_{\boldsymbol{x}^*\boldsymbol{X}}\left(\sigma^2 I + \boldsymbol{K}_{\boldsymbol{XX}}\right)^{-1}\boldsymbol{y} \\ var(\boldsymbol{x}^*) &= k_{\boldsymbol{x}^*\boldsymbol{x}^*} - \boldsymbol{k}_{\boldsymbol{x}^*\boldsymbol{X}}\left(\sigma^2 I + \boldsymbol{K}_{\boldsymbol{XX}}\right)^{-1}\boldsymbol{k}_{\boldsymbol{X}\boldsymbol{x}^*} \end{aligned} \tag{3}$$

where $\boldsymbol{k}_{\boldsymbol{x}^*\boldsymbol{X}} = k(\boldsymbol{x}^*, \boldsymbol{X})$ is a n-dimensional row vector of the covariance between $\boldsymbol{x}^*$ and the N training samples ($\boldsymbol{k}_{\boldsymbol{x}^*\boldsymbol{X}} = \boldsymbol{k}_{\boldsymbol{X}\boldsymbol{x}^*}^T$), $\boldsymbol{K}_{\boldsymbol{XX}} = k(\boldsymbol{X}, \boldsymbol{X})$ denotes the kernel matrix of the N training samples.

2.2. Gaussian Process Latent Variable Model

GPLVM [15] is originally proposed for dimension reduction of high dimensional data. Its goal is to learn the low dimensional representation $\boldsymbol{X}^{N\times Q}$ of the data matrix $\boldsymbol{Y} \in \mathbb{R}^{N\times D}$, where N and D are the number and dimensionality of training

samples, respectively. GPLVM assumes that the observed data is generated from a lower dimensional data $\boldsymbol{X}$ where $Q \ll D$. The generation process of the ith training sample $\boldsymbol{y}_i$ is

$$\boldsymbol{y}_i = f(\boldsymbol{x}_i) + \varepsilon \tag{4}$$

where ε is the noise with gaussian distribution $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. f is a nonlinear function with GP prior $f \sim \mathcal{GP}(0, \boldsymbol{K})$. As a probabilistic model, GPLVM can also be represented by a directed graph as shown in Fig. 1. In this paper, we use gray and white circles to denote the observed and latent variables, respectively. From the this graphical representation, we can know that the marginal likelihood $p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta})$ can be obtained by using Bayesian theorem and integrating out f,

$$p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}) = \prod_{j=1}^{D} \frac{1}{(2\pi)^{\frac{1}{2}} \left|\boldsymbol{K}\right|^{\frac{1}{2}}} e^{-\frac{1}{2}\boldsymbol{y}_{:,j}^T \boldsymbol{K}^{-1} \boldsymbol{y}_{:,j}} \tag{5}$$

where $\boldsymbol{\theta}$ denotes the hyper-parameters of both kernel function and noise, $\boldsymbol{y}_{:,j}$ denotes the jth column of matrix $\boldsymbol{Y}$. Thus, we can maximize the marginal likelihood with respect to $\boldsymbol{X}$ and the hyper-parameter $\boldsymbol{\theta}$,

$$\left\{\widehat{\boldsymbol{X}}, \widehat{\boldsymbol{\theta}}\right\} = \arg\max_{\boldsymbol{X}, \boldsymbol{\theta}} p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}) \tag{6}$$

where $\widehat{\boldsymbol{X}}$ and $\widehat{\boldsymbol{\theta}}$ denote the optimal values of $\boldsymbol{X}$ and $\boldsymbol{\theta}$, respectively.

The object function of GPLVM can also be derived from *Probabilistic Kernel Principal Components Analysis*. In PKPCA, we aim to learn a low-dimensional representation $\boldsymbol{X}$ of original data $\boldsymbol{Y}$ and assume that the projection (parameter) matrix $\boldsymbol{W}$ follows a spherical Gaussian distribution prior below:

$$p(\boldsymbol{W}) = \prod_{i=1}^{D} \mathcal{N}(\boldsymbol{w}_i|0, \boldsymbol{I}) \tag{7}$$

Then, we can get the marginal likelihood by integrating out $\boldsymbol{W}$:

$$\begin{aligned} p(\boldsymbol{Y}|\boldsymbol{X}, \beta) &= \int p(\boldsymbol{Y}|\boldsymbol{X}\boldsymbol{W}, \beta) p(\boldsymbol{W}) d\boldsymbol{W} \\ &= \prod_{d=1}^{D} \mathcal{N}\left(\boldsymbol{y}_{:,d}|0, \boldsymbol{X}\boldsymbol{X}^T + \beta^{-1}\boldsymbol{I}\right) \end{aligned} \tag{8}$$

As we can see from Eq. (8), GPLVM can be equivalent to PKPCA by replacing $\boldsymbol{X}\boldsymbol{X}^T$ in Eq. (8) with a kernel matrix $\boldsymbol{K}$.

Besides be used in DR, GPLVM has also been extended to adapt to many other machine learning problems, such as Image Reconstruction [28], Human pose estimation [30], Age Estimation [31] and so on. In addition, there have been many open-source softwares that are available for the implementation of GPs and GPLVMs. We give a brief description as follows:

- GPML[1] is an excellent GP toolkit. It contains a large number of GP-related codes such as various kernel functions, likelihood functions and inference methods.
- GPy[2] is a Gaussian Process framework written in python, from the Sheffield machine learning group. It provides a set of tools for the implementation of GP-based methods.
- FGPLVM[3] is a toolkit that allows for larger GP-LVM models through using the sparse approximation.
- Matlab Toolbox for Dimensionality Reduction[4] is a dimension reduction toolkit. It provides the implementations of many techniques for dimensionality reduction and metric learning. It also contains an implementation of GPLVM.

3. Extensions of GPLVM

As shown in Section 2.2, due to its flexible structure in modeling, GPLVM has been adapted to various learning scenarios and led to corresponding learning methods. In this section, we first classify these methods into three types in terms of the various *strategies* that used and then provide a taxonomy of the main existing GPLVMs as shown in Fig. 2. For more typical examples of specific applications refer to Section 4.

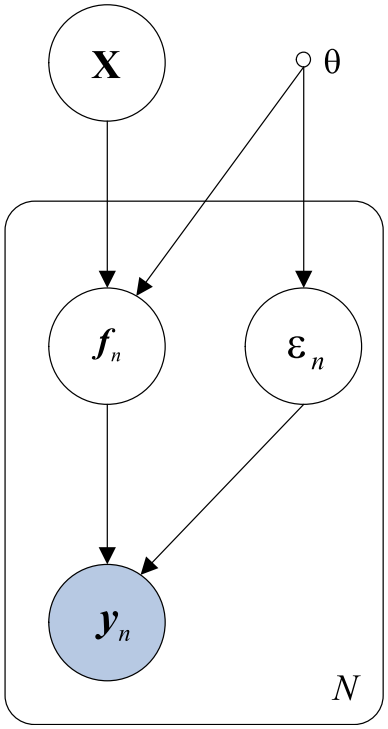

Fig. 1. Gaussian process as a latent variable model: we use arrows to denote the dependency relations between variables. The gray and white circles denote the observed and latent variables, respectively.

[1] http://www.gaussianprocess.org/gpml/code/matlab.
[2] https://github.com/SheffieldML/GPy.
[3] https://github.com/lawrennd/fgplvm.
[4] http://lvdmaaten.github.io/drtoolbox/.

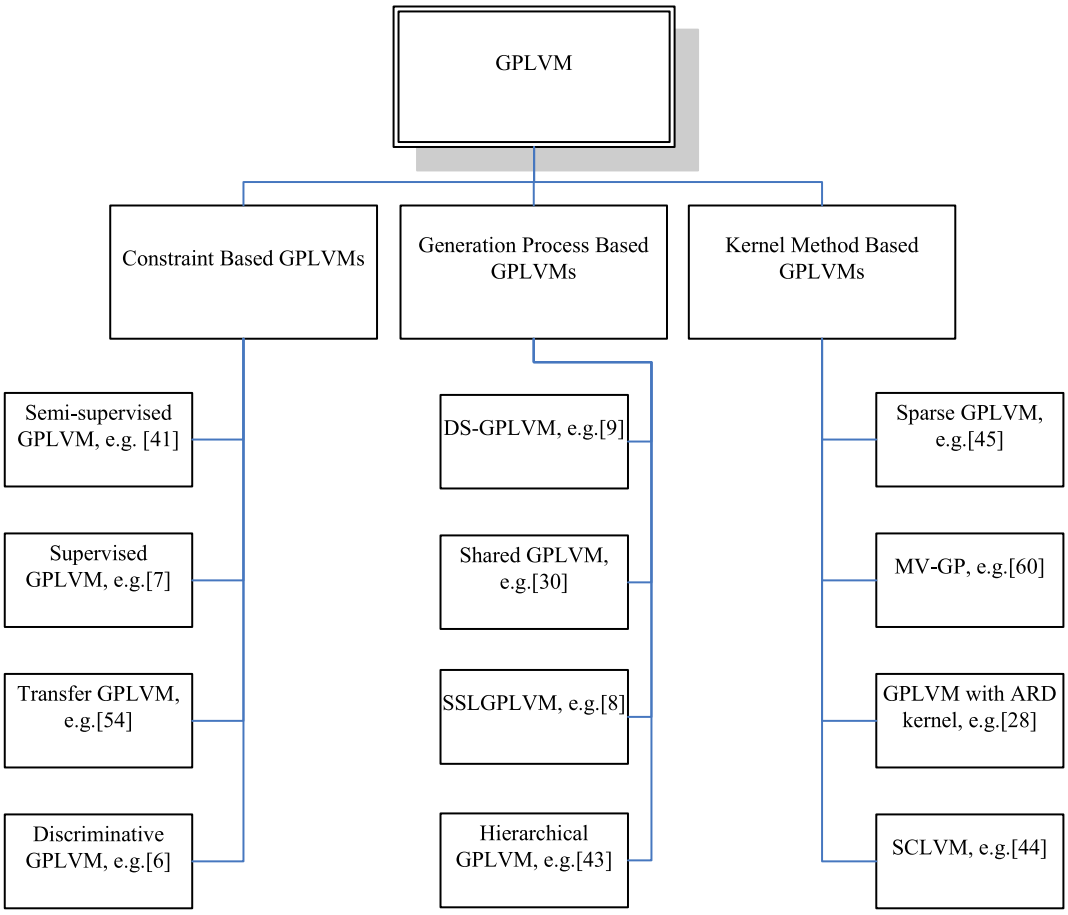

Fig. 2. A taxonomy of the GPLVMs.

3.1. Constraint based GPLVMs

From the generation process of GPLVM in Section 2.2, we can know that the conventional GPLVM needs not make any assumptions on the prior of latent variables. However, lack of such assumption makes the model inferred by just maximizing the log marginal likelihood in Eq. (6) prone to overfitting. To tackle this problem, one of effective approaches is to impose a specific prior onto the latent variables for a posterior estimation. Thus, we can introduce various constraints into the prior for the estimation the latent variables in different tasks [41,6]. Specifically, we assume that $p(\boldsymbol{X})$ denotes the imposed prior. By using the Bayesian theorem, we can formulate the posterior probability of the latent variables $\boldsymbol{X}$

$$p(\boldsymbol{X}|\boldsymbol{Y},\boldsymbol{\theta}) = \frac{p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})p(\boldsymbol{X})}{p(\boldsymbol{Y})} \propto p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})p(\boldsymbol{X}) \tag{9}$$

Thus, we get the posterior estimation of $\boldsymbol{X}$ by maximizing Eq. (9) instead of the marginal likelihood $p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})$.

In practical applications, many constraints can be introduced and embedded into the priors for problem at hand. In [41], pairwise constraints (which indicates whether two samples belong to the same class or different classes) are utilized to construct a specific prior of $\boldsymbol{X}$ for semi-supervised learning [6,7]. Construct a discriminative prior based on the label information of data and use this prior directly predict the label of a new sample. In general, all these priors described above can be considered as such a set of constraints derived from given problems, which can be used to learn the latent variables of GPLVM. In some other learning scenarios, we can even also impose explicit constraints on the latent variable [42] for construction of the proper prior.

3.2. Generation process based GPLVMs

The conventional GPLVM just defines the generation process of high dimensional data for dimension reduction. However, for more complex data, such as multi-view and/or multi-modal data, such a single generation process fails to fit the data. Therefore, we need to redefine the generation process to deal with different types of data. By this approach, GPLVM is extended to be capable to model various complex data.

In [30], a *Shared Gaussian Process Latent Variables Model* (Shared GPLVM) is proposed to define the generation process of data from multiple sources and learn a low dimensional shared representation for these data. Besides the Shared GPLVM, there are yet many other methods to define the generation process from inputs to outputs for different learning

tasks, for example [7] and [8] define the discriminant form generation processes to implement the prediction of labels [43]. Construct the hierarchical GPLVM to learn more complicated functions.

3.3. Kernel method based GPLVMs

As described in Section 2.2, GPLVM can in fact also be considered as a kernel method, in which the selection of kernel can greatly influence its performance. In general, we can select various kernel to meet the demands of different tasks. For example, in order to automatically select the subset of the latent space, we can use the automatic relevance determination (ARD) kernel in the construction of GPLVM [28]. The definition of the ARD kernel is as follows,

$$k(\boldsymbol{x},\boldsymbol{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} \alpha_q \left(x_q - x_q'\right)^2 \right) \tag{10}$$

where σ_f^2 and $\{\alpha_q\}_{q=1}^{Q}$ are nonnegative hyper-parameters. By learning the hyper-parameters $\{\alpha_q\}_{q=1}^{Q}$, we can automatically remove the dimensions with the corresponding hyper-parameters $\alpha = 0$.

Besides ARD kernel, many other kernels can also be used to handle different tasks. In the *Structure Consolidation Latent Variable Model* (SCLVM) [44], a compositional kernel is used to solve the problem of label imbalance. In some situations, we can even learn the kernel matrix from data directly. Sparse GPLVMs [45–49] try to learn a *reduced-rank approximations* of the kernel matrix to improve the scalability of GPLVM. Specifically, they use the following equation to estimate the kernel matrix $\boldsymbol{K}$,

$$\boldsymbol{K} \approx \boldsymbol{K}_{NM}\boldsymbol{K}_{MM}\boldsymbol{K}_{MN} \tag{11}$$

where N denotes the number of training samples and $M \ll N$. With this method, the computation of GPLVMs when inverting the kernel matrix has a time complexity of $\mathcal{O}(M^2N)$ other than $\mathcal{O}(N^3)$, which makes GPs and GPLVMs able to effectively solve problems with large scale data.

4. Typical examples

In Section 3, we summarized the main strategies for the extension of GPLVM. In this section, we will give a review and detailed formulations of the typical GPLVMs that extensively developed based on these strategies. Moreover, we also give a brief description of some other GPLVMs that proposed for special application scenarios.

4.1. GPLVMs with various constraints

As described in Section 3.1, various constraints can be imposed into the prior of latent variables according to the specific tasks. In general, the existing GPLVMs mainly utilize the following constraints: semi-supervised constraints [41,50], supervised constraints [6,7], cross-task constraints [51].

4.1.1. Semi-supervised GPLVM

In some machine learning scenarios, we assume that user can get both input data described in Section 2.2 and some semi-supervised information, such as pairwise constraints [41]. Proposes the Semi-supervised GPLVM which utilizes such pairwise information to construct a specific prior of $\boldsymbol{X}$. Firstly, it defines a weight matrix $\boldsymbol{W} \in \mathbb{R}^{N\times N}$

$$W_{ij} = \begin{cases} \dfrac{e^t}{1+e^t}, & \text{if } y_i \text{ and } y_j \text{ belong to different classes} \\ -\dfrac{e^t}{1+e^t}, & \text{if } y_i \text{ and } y_j \text{ belong to the same class} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where $t = \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2$ represents the Euclidean distance between two latent variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Then, based on the matrix $\boldsymbol{W}$, it constructs a priori probability of the latent variables $\boldsymbol{X}$ as

$$p(\boldsymbol{X}) = \frac{1}{Z}\exp\left(-\sum_{i,j=1}^{N} d(\boldsymbol{x}_i,\boldsymbol{x}_j) \right) \tag{13}$$

where $d(\boldsymbol{x}_i,\boldsymbol{x}_j) = W_{i,j}\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2$ and Z is normalization constant. From the above process, it is clear that the pairwise constraints can enforce similar samples to be close and dissimilar samples to be far. In order to derive the posterior, we can first write the marginal likelihood as

$$p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta}) = \prod_{d=1}^{D} \frac{1}{(2\pi)^{N/2}\left|\boldsymbol{K}\right|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{y}_{:,d}^T \boldsymbol{K}^{-1}\boldsymbol{y}_{:,d} \right) \tag{14}$$

Then, we use the Bayesian theorem in Eq. (9) to get the posterior probability of the latent variables $\boldsymbol{X}$. Thus, maximizing marginal likelihood can be replaced by maximizing the log posterior given by

$$\mathcal{L} = \ln p(\boldsymbol{X}|\boldsymbol{Y},\boldsymbol{\theta}) \approx \ln p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta}) + \ln p(\boldsymbol{X}) \tag{15}$$

4.1.2. Discriminative GPLVM

In supervised learning scenarios, the goal is to learn models from labeled data and predict the labels of new samples directly [6–8]. As described in Section 3.1, we can also introduce label constraint information into GPLVM for supervised learning. *Discriminative Gaussian Process Latent Variable Model* (Discriminative GPLVM) [6] imposes a discriminative prior to the latent variables which can significantly improve the discriminant property of GPLVM. Specifically, this discriminative prior is constructed by borrowing the idea of *Linear Discriminant Analysis* (LDA) [21], as shown in the following equation,

$$J(\boldsymbol{X}) = tr\left(\boldsymbol{S}_w^{-1}\boldsymbol{S}_b\right), \tag{16}$$

where $\boldsymbol{S}_w$ and $\boldsymbol{S}_b$ denote within-class and between-class divergence matrices, respectively. The definition of $\boldsymbol{S}_w$ and $\boldsymbol{S}_b$ is as follows,

$$\boldsymbol{S}_w = \sum_{i=1}^{L} \frac{N_i}{N} (\boldsymbol{M}_i - \boldsymbol{M}_0)(\boldsymbol{M}_i - \boldsymbol{M}_0)^T \tag{17}$$

$$\boldsymbol{S}_b = \sum_{i=1}^{L} \frac{N_i}{N} \left[\frac{1}{N_i} \sum_{k=1}^{N_i} \left(\boldsymbol{x}_k^{(i)} - \boldsymbol{M}_i \right) \left(\boldsymbol{x}_k^{(i)} - \boldsymbol{M}_i \right)^T \right], \tag{18}$$

where $\boldsymbol{M}_i$ is the mean of the elements of class i, $\boldsymbol{M}_0$ is the mean of all the training points of all classes, $\boldsymbol{X}^{(i)} = [\boldsymbol{x}_1^{(i)}, \ldots, \boldsymbol{x}_{N_i}^{(i)}]$ are the N_i training points of class i. By maximizing the object function of LDA, we can find a transformation that maximizes between-class separability and minimizes within-class variability. Inspired by this motivation, Discriminative GPLVM constructs a prior over latent variables that forces the latent points of the same class to be close together and far from those of other classes, as shown in Eq. (19)

$$p(\boldsymbol{X}) = \frac{1}{Z_d} \exp\left\{ -\frac{1}{\sigma_d^2} J^{-1} \right\} \tag{19}$$

By using Bayes theorem described in Eq. (9), we can get the posterior distribution of latent variable $\boldsymbol{X}$. Then, we can minimize the negative log posterior in Eq. (20) to learn these latent variables,

$$\mathcal{L}_S = \mathcal{L}_r + \sum_i \ln \theta_i + \frac{1}{\sigma_d^2} tr\left(\boldsymbol{S}_b^{-1} \boldsymbol{S}_w \right), \tag{20}$$

where $\mathcal{L}_r$ represents the negative log likelihood of GPLVM, $\sum \ln \theta_i$ denotes the prior of hyper-parameters, $\frac{1}{\sigma_d^2}$ can be considered as the coefficient that balances the discriminant capability and the fitness to data $\boldsymbol{Y}$. Furthermore, since the kernel matrix learned in the discriminative GPLVM is more discriminative and flexible, it can directly be used in *Gaussian Process Classification* (GPC) [22] for supervised learning tasks.

4.1.3. Supervised GPLVM

Gao et al. [7] proposes a *Supervised Gaussian Processes Latent Variables Model* (Supervised GPLVM) by using latent variables to connect observations and their corresponding labels. Specifically, it assumes that $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, $\boldsymbol{Y} \in \mathbb{R}^{N \times L}$, $\boldsymbol{Z} \in \mathbb{R}^{N \times Q}$ denote the matrices of inputs, labels and the corresponding latent variables, respectively. The graphical representation is shown in Fig. 3. As we can see, Supervised GPLVM assumes that both $\boldsymbol{X}$ and $\boldsymbol{Y}$ are generated from the same latent variables $\boldsymbol{Z}$ by GPs. The latent variable $\boldsymbol{Z}$ can serve as a bridge between two observed matrices. This approach has already been extensively studied in many machine learning models such as joint manifold model [52] and supervised probabilistic PCA (SPPCA) [53]. As shown in Fig. 3, each dimension of $\boldsymbol{X}$ and $\boldsymbol{Y}$ is independent conditioned on $\boldsymbol{Z}$. Thus, the log marginal likelihood of the model can be obtained the following equation,

$$\begin{aligned} \ln p(\boldsymbol{X}, \boldsymbol{Y} | \boldsymbol{Z}) &= \ln p(\boldsymbol{X} | \boldsymbol{Z}) + \ln p(\boldsymbol{Y} | \boldsymbol{Z}) \\ &= \sum_{d=1}^{D} \ln p(\boldsymbol{x}_{:,d} | \boldsymbol{Z}) + \sum_{l=1}^{L} \ln p\left(\boldsymbol{y}_{:,l} \middle| \boldsymbol{Z} \right) \end{aligned} \tag{21}$$

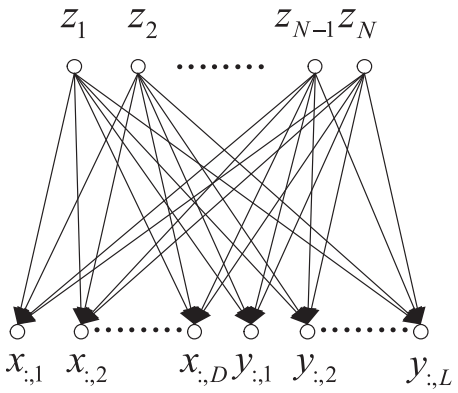

Fig. 3. The graphical representation of Supervised GPLVM. The latent variable $\boldsymbol{Z}$ can serve as a bridge between two observed matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$.

4.1.4. Transfer Latent Variable Model

When using GPLVM to predict new samples, we make a strong assumption that both training and testing samples are drawn from an *independent identity distribution* (iid). However, when these two data sets are drawn from different distributions, the GPLVM trained by using training data set will have a poor performance on the new test samples. One useful tool for this problem is the transfer learning strategy [51]. Transfer learning is widely studied by researchers in machine learning, which focuses on storing knowledge which gained when solving one problem and applying it to a different but related problem.

Gao et al. [54] proposes a transfer learning framework for GPLVM (*Transfer Latent Variable Model*, TLVM) based on the distance between training and testing data sets. Specifically, it assumes that the training set $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N]^T$ and testing set $\boldsymbol{Y}_t = [\boldsymbol{y}_{t1}, \ldots, \boldsymbol{y}_{tM}]^T$ are drawn from two different distribution and their corresponding latent variables matrix are $\boldsymbol{X}$ and $\boldsymbol{X}_t$, respectively. TLVM uses KL-divergence to measure the distance between the training and testing data set,

$$\begin{aligned} D(p(\boldsymbol{Y}) || p(\boldsymbol{Y}_t)) &= \sum_{d=1}^{D} KL\left(p\left(\boldsymbol{Y}_{(:,d)} \right) \middle|\middle| p\left(\boldsymbol{Y}_{t(:,d)} \right) \right) \\ &= \frac{D}{2} \ln \left| \boldsymbol{K}_t \boldsymbol{K}^{-1} \right| + \frac{D}{2} tr\left(\boldsymbol{K}_t^{-1} (\boldsymbol{K} - \boldsymbol{K}_t) \right), \end{aligned} \tag{22}$$

where $\boldsymbol{K}$ and $\boldsymbol{K}_t$ are the kernel matrices of $\boldsymbol{X}$ and $\boldsymbol{X}_t$, respectively. The object function $F(\boldsymbol{X}, \theta)$ can be written as

$$F(\boldsymbol{X}, \boldsymbol{\theta}) = L(\boldsymbol{X}, \boldsymbol{\theta}) + D(p(\boldsymbol{Y}) || p(\boldsymbol{Y}_t)). \tag{23}$$

Then, we can minimize Eq. (23) to find the optimal values of latent variable $\boldsymbol{X}$ and hyper-parameter $\boldsymbol{\theta}$.

4.2. GPLVMs based on generation process reconstruction

In general, there are mainly two strategies to reconstruct the generation process, data-driven approaches and task-driven approaches. In data-driven approaches, we can reconstruct

the generation process of GPLVM with respect to the characteristics of the data, such as the works in [30,9] for multi-view learning. In task-driven approaches, the generation process is reconstructed to meet the demands of the tasks [8]. In this section, we will demonstrate the concrete examples of these two approaches.

4.2.1. Shared GPLVM

In computer science, many tasks are associated with data coming from multiple streams or views of the same underlying phenomenon [9,29,30,42,55]. In video processing, there may be many cameras each of which focus on objects from different viewpoints. In user-centric social networks, information from different sources (text, image, video, audio and social information) can be obtained [56]. Our goal is to utilize these complementary information to fulfill machine learning tasks such as person re-Identification [57], human pose estimation [30], facial expression recognition [9] and so on.

Shared GPLVM [30] can efficiently capture the correlations among different sets of corresponding features and is applied in machine learning tasks. Specifically for two data views, $\boldsymbol{Y}=[\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N]^T\in\mathbb{R}^{N\times D}$ and $\boldsymbol{Z}=[\boldsymbol{z}_1,\ldots,\boldsymbol{z}_N]^T\in\mathbb{R}^{N\times L}$ (where $\boldsymbol{y}_n$ and $\boldsymbol{z}_n$ denote the representations of the n^{th} sample in the two views, respectively), the goal of Shared GPLVM is to learn a common low-dimensional representation $\boldsymbol{X}$ for $\boldsymbol{Y}$ and $\boldsymbol{Z}$. Its graphical representation is shown in Fig. 4. As we can see, it assumes that both $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are generated by GP from a shared latent variable $\boldsymbol{X}$ and its corresponding generation processes can respectively be written as

$$\begin{aligned}\boldsymbol{y}_i&=f^Y(\boldsymbol{x}_i)+\varepsilon_Y,\ \varepsilon_Y\sim\mathcal{N}\left(0,\sigma_Y^2\boldsymbol{I}\right),\\ \boldsymbol{z}_i&=f^Z(\boldsymbol{x}_i)+\varepsilon_Z,\ \varepsilon_Z\sim\mathcal{N}\left(0,\sigma_Z^2\boldsymbol{I}\right),\end{aligned}\tag{24}$$

where f^Y and f^Z denote the functions with GP prior, ε_Y and ε_Z denote the noises, respectively. According to the assumption that $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are independent conditioned on $\boldsymbol{X}$, the likelihood of Shared GPLVM is formulated as,

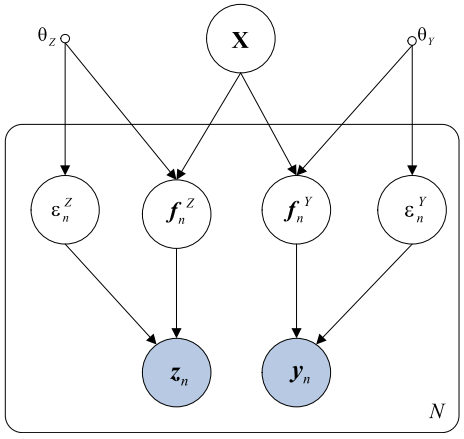

Fig. 4. The graphical of Shared GPLVM. The generation process is redefined by assuming that both $\boldsymbol{Z}$ and $\boldsymbol{Y}$ are generated by GP from $\boldsymbol{X}$. $\boldsymbol{Z}$ and $\boldsymbol{Y}$ are independent conditioned on $\boldsymbol{X}$.

$$p(\boldsymbol{Y},\boldsymbol{Z}|f^Y,f^Z,\boldsymbol{X},\boldsymbol{\theta}_Y,\boldsymbol{\theta}_Z)=\prod_{i=1}^{N}p(\boldsymbol{y}_i|f^Y,\boldsymbol{x}_i,\boldsymbol{\theta}_Y)p(\boldsymbol{z}_i|f^Z,\boldsymbol{x}_i,\boldsymbol{\theta}_Z)\tag{25}$$

where $\boldsymbol{\theta}_Y$ and $\boldsymbol{\theta}_Z$ denote the hyper-parameters of GPs, respectively. The latent variables are obtained by integrating out the latent variables f^Y and f^Z, then maximizing the following joint marginal likelihood:

$$p(\boldsymbol{Y},\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta}_Y,\boldsymbol{\theta}_Z)=\prod_{i=1}^{N}p(\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{\theta}_Y)p(\boldsymbol{z}_i|\boldsymbol{x}_i,\boldsymbol{\theta}_Z)\tag{26}$$

4.2.2. Discriminative shared GPLVM

By redefining the generation process, Shared GPLVM that described in [30] can efficiently capture the correlations among different sets of corresponding features and is applied in many machine learning tasks [9,58] [9]. Proposed *Discriminative Shared Gaussian Processes Latent Variable Model* (DS-GPLVM) by considering the situation in which user has observed the label information of each view. It uses such label information to construct a discriminant prior based on the Laplacian matrix [59]. Specifically, the joint Laplacian matrix has the following form:

$$\tilde{\boldsymbol{L}}=\boldsymbol{L}_N^{(1)}+\boldsymbol{L}_N^{(2)}+\ldots+\boldsymbol{L}_N^{(V)}+\xi\boldsymbol{I}=\sum_v\boldsymbol{L}_N^{(v)}+\xi\boldsymbol{I}\tag{27}$$

where $\boldsymbol{L}_N^{(v)}$ corresponds the Laplacian matrix of the *vth* view. Thus, based on the joint Laplacian matrix, we can define the discriminant shared prior as

$$p(\boldsymbol{X})=\prod_{v=1}^{V}p\left(\boldsymbol{X}|\boldsymbol{Y}^{(v)}\right)^{\frac{1}{V}}=\frac{1}{V\bullet Z_q}\exp\left[-\frac{\beta}{2}tr\left(\boldsymbol{X}^T\tilde{\boldsymbol{L}}\boldsymbol{X}\right)\right]\tag{28}$$

where Z_q is the normalization constant. After getting the prior, we learn the latent variable $\boldsymbol{X}$ by maximizing

$$L_s(\boldsymbol{X})=\sum_v L^{(v)}+\frac{\beta}{2}tr\left(\boldsymbol{X}^T\tilde{\boldsymbol{L}}\boldsymbol{X}\right)\tag{29}$$

where $L^{(v)}$ denotes the marginal likelihood of the vth view and β is a regularizer. By combining the Shared GPLVM with Discriminative GPLVM, DS-GPLVM gets competitive performance in multi-view and view-invariant facial expression recognition tasks.

4.2.3. Supervised Latent Linear GPLVM (SLLGPLVM)

The work in [8] implements supervised learning of GPLVM by redefining the generation process. Specifically, as shown in Fig. 5, the model assumes that the latent variable $\boldsymbol{Z}$ is generated by a linear transformation of $\boldsymbol{X}$. Then, the target variable $\boldsymbol{Y}$ is generated by a GP with latent variables as inputs. The whole generation process is as follows:

$$\boldsymbol{y}=g(\boldsymbol{z})+\varepsilon=g(\boldsymbol{W}\boldsymbol{x})+\varepsilon,\tag{30}$$

where $\boldsymbol{W}$ represents the projection (parameter) matrix from input space to latent space. The function $g(\bullet)$ denotes the

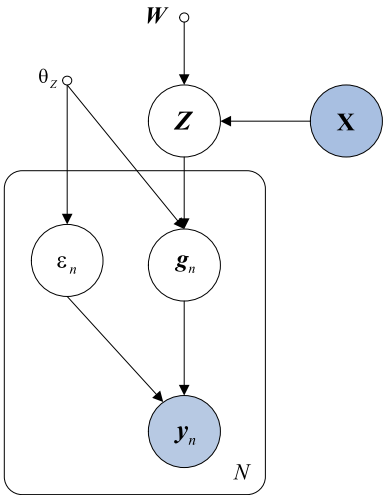

Fig. 5. The graphical representation of Supervised Latent Linear GPLVM. As we can see, both $\boldsymbol{X}$ and $\boldsymbol{Y}$ are observed. Our goal is to learn the project matrix $\boldsymbol{W}$ for prediction.

mapping that transforms the latent variables to the output variables and $g(\cdot) \sim \mathcal{GP}(\mathbf{0}, k(\cdot,\cdot))$. Thus, the prior distribution of $\boldsymbol{g}$ can be written as

$$p(\boldsymbol{g}) = \mathcal{N}(\boldsymbol{g}|0, \boldsymbol{K}_{Z,Z}), \tag{31}$$

where $\boldsymbol{K}_{Z,Z}$ denotes the kernel matrix of latent variables $\boldsymbol{Z} = \boldsymbol{XW}$. Based on this prior, we can construct GPLVM and maximize the log marginal likelihood to learn the matrix $\boldsymbol{W}$ as in Section 2.2.

4.3. GPLVMs based on specific kernels

As a kernel method, GPLVM can utilize various kernels for specific tasks. Firstly, many existing kernels can be introduced into GPLVM to address different problems [28,60]. Secondly, we can also construct kernels by combining multiple kernels [44] or learning the kernel matrices directly [45–49].

4.3.1. Structure consolidation latent variable model

As described in Section 3.3, many kernels can be introduced into GPLVM to meet the demands of different tasks. The SCLVM [44] uses a compositional kernel to address the problem of label imbalance. Specifically, it separates the latent space into a shared space with the dimensionality Q_s and a private space with the dimensionality Q_p. Thus, the latent representation can be denoted as $\boldsymbol{x} = [\boldsymbol{x}_s^T, \boldsymbol{x}_p^T]^T$, where $\boldsymbol{x}_s \in \mathbb{R}^{Q_s}$ and $\boldsymbol{x}_p \in \mathbb{R}^{Q_p}$ are the latent representations in shared and private space respectively. Then, the compositional kernel can be defined as follows:

$$k((\boldsymbol{x}, c_{\boldsymbol{x}}), (\boldsymbol{x}', c_{\boldsymbol{x}'})) = k_s(\boldsymbol{x}_s, \boldsymbol{x}_s') + k_p\Big((\boldsymbol{x}_p, c_{\boldsymbol{x}}), \big(\boldsymbol{x}_p', c_{\boldsymbol{x}'}\big)\Big) \tag{32}$$

where k_s is the kernel function for the shared space and k_p is the kernel function for the private space. The private kernel is defined as follows:

$$k_p\Big((\boldsymbol{x}_p, c_{\boldsymbol{x}}), \big(\boldsymbol{x}_p', c_{\boldsymbol{x}'}\big)\Big) = \begin{cases} k'\big(\boldsymbol{x}_p, \boldsymbol{x}_p'\big), & c_{\boldsymbol{x}} = c_{\boldsymbol{x}'} \\ 0, & c_{\boldsymbol{x}} \neq c_{\boldsymbol{x}'} \end{cases} \tag{33}$$

where k' is a common kernel function and $c_{\boldsymbol{x}}$ is the label of data point $\boldsymbol{x}$. By such a definition the shared space can capture the common regularities among categories and the private spaces can model the variance specific to individual categories. Thus the data in each category can be modeled appropriately to solve the problem of label imbalance.

4.3.2. Matrix-valued Gaussian process

Recently, transposable data (such as proteins interaction networks in biology and movies rating data in recommendation system) which describes the relationships between pairs of entities, has been analyzed. Such data can often be organized as a matrix, with one set of entities as the rows, the other set of entities as the columns [60]. Proposes a *Matrix-valued Gaussian Process* (MV-GP) to model such data. Specifically, it assumes that $\boldsymbol{Y} \in \mathbb{R}^{N \times D}$ denotes the transposable data matrix with N rows and D columns. It models $\boldsymbol{Y}$ as a matrix-variate normal distribution with iid Gaussian observation noise.

$$p\big(\boldsymbol{Y}\big|\boldsymbol{Z}, \sigma^2\big) = \mathcal{N}\big(vec(\boldsymbol{Y})\big|vec(\boldsymbol{Z}), \sigma^2 \boldsymbol{I}_{ND}\big) \tag{34}$$

where latent variables $\boldsymbol{Z}$ can be thought of as the noise-free observations and $vec(\boldsymbol{Y})$ denotes the vector obtained by concatenating the columns of $\boldsymbol{Y}$.

A further assumption is that $\boldsymbol{Z}$ is drawn from a MV-GP,

$$p(\boldsymbol{Z}|\boldsymbol{C}, \boldsymbol{R}) = \frac{\exp\big\{-\frac{1}{2} Tr[\boldsymbol{C}^{-1}\boldsymbol{Z}^T\boldsymbol{R}^{-1}\boldsymbol{Z}]\big\}}{(2\pi)^{ND/2}\big|\boldsymbol{R}\big|^{N/2}\big|\boldsymbol{C}\big|^{D/2}} \tag{35}$$

where matrix $\boldsymbol{C}$ is a $D \times D$ column covariance matrix and $\boldsymbol{R}$ is an $N \times N$ row covariance matrix. Obviously, MV-GP can be considered as a multivariate normal distribution by writing the MV-GP as

$$p(vec(\boldsymbol{Z})|\boldsymbol{C}, \boldsymbol{R}) = \mathcal{N}(vec(\boldsymbol{Z})|0_{ND}, \boldsymbol{C} \otimes \boldsymbol{R}) \tag{36}$$

where $\otimes$ denotes the *Kronecker product*. Furthermore, it can also be considered as a GPLVM in which the common kernel function is replaced with the product of two kernel functions. Based on this formulation, the MV-GP has been applied in many machine learning tasks [61–64] which try to learn the relation between pairwise entities.

4.4. Other GPLVMs for special applications

Apart from the various GPLVMs that described above, there are also many other GPLVMs that try to address the special problems in machine learning. These GPLVMs can be considered as a set of self-contained models that are difficult to be accommodated to one of the three strategies described in Section 3. In this section we just give a simple description for these models.

4.4.1. Bayesian Gaussian Process Latent Variable Model

GPLVM can also be used in the situation with uncertain inputs. In this case, we can take a Bayesian estimation of GPLVM whose latent variables are integrated out instead of optimized [28]. However, as described in [28], the main difficulty is that, to apply Bayes inference to GP-LVM, we need to approximately integrate out the latent variables nonlinearly in the inverse kernel matrix of the GP model [28]. Proposes a variational Bayesian GPLVM model by using variational inference [65] in an expanded probability model to tackle the above problem. This model and its extension have been widely used in many machine learning tasks, such as gaussian process regression with uncertain inputs [66], hybrid discriminative-generative approach [67] and so on.

4.4.2. Gaussian Process Dynamical Systems

In robot control, computer vision, computational biology, users are often faced with high dimensional time series data [68]. By assuming $\boldsymbol{x}$ as a multivariate gaussian process indexed by time t, GPLVM can be extended to a dynamical model which are called Gaussian Process Dynamical System (GPDS) [43,68–70] to adapt such dynamic environment. In fact, this model can be considered as Hierarchical GPLVM and obtain a satisfactory performance in analysis of data with time series information. Models based on GPDS have been used in many tasks such as human pose recognition [69], modeling raw high dimensional video sequences [68], video repair [71] and some other related applications.

4.4.3. Deep GPLVM

Although Gaussian Process Latent Variables Model provides a flexible, non-parametric, non-linear dimension reduction strategy, their representation ability is still restricted by the kernel functions [72]. In general, GPLVM, as a shallow model, can be stacked to a deep architecture [73]. This structure has been widely used in many deep models [74,75]. And some deep models based on GPLVM have been proposed such as auto-encoded deep gaussian processes [72], deep gaussian processes [76], deep gaussian processes for regression [77].

5. Conclusion and discussion

In this paper, we first give a detail formulation of GPLVM and its relation to PKPCA. Then, we summarize the main strategies to improve GPLVM and a taxonomy is constructed in terms of the various strategies used. We also review the main GPLVMs that extensively developed based on the methods described in Section 3. In this section, we will draw some promising lines for future researches of GPLVM.

5.1. GPLVM for discrete variables

The conventional GPLVM and its extensions are suitable for analysis of continuous data. However, in many machine learning task such as natural language processing and medical diagnosis, users often get discrete variables. Although [78], has proposed a GPLVM for the estimation of multivariate categorical data, its inference is based on the variational approximation and sampling approaches which have a high computational complexity. To overcome the difficulty, we mainly should consider two factors: the construction of likelihood and the inference method of latent variables, which deserve more attentions.

5.2. Scalable inference in GPLVM

During the inference of GPLVM, we should evaluate the distribution $p(\boldsymbol{Y}|\boldsymbol{X})$ which has a time complexity of $\mathcal{O}(N^3)$ by computing the inversion of the $N \times N$ kernel matrix. Although, there have been many methods for the sparse estimation of GP and GPLVM [45–49], they can not improve the scalability of GPLVM without the risk of accuracy loss. For this reason, the scalable inference of GP and GPLVM has become a popular research content recent years and will be paid more attention in the future.

5.3. Similarity Gaussian Process Latent Variable Model

Recent years, metric learning methods are widely studied. Its goal is to learn a suitable metric by using the distance constraints of pairwise samples [42]. To our best knowledge, there is only one model [32] that uses GPLVM to construct a similarity learning model (m-SimGP) of multi-modal data. This model can be applied to various tasks to discover the nonlinear correlations and obtain the comparable low-dimensional representation for heterogeneous modalities. This kind of GPLVM-based metric learning model has a more flexible structure than the conventional metric models and is likely to receive increasing interest in the near future.

References

[1] G. Darnell, S. Georgiev, S. Mukherjee, B. E. Engelhardt, Adaptive randomized dimension reduction on massive data, arXiv preprint arXiv: 1504.03183.
[2] A. Sarveniazi, Am. J. Comput. Math. 04 (2) (2014) 55–72.
[3] D.M. Blei, A.Y. Ng, M.I. Jordan, J. Mach. Learn. Res. 3 (2003) 993–1022.
[4] M.A. Carreira-Perpin, M.A. Carreira-Perpin, Perpinan (1997) 1–69.
[5] S.T. Roweis, L.K. Saul, Science 290 (5500) (2000) 2323–2326.
[6] R. Urtasun, T. Darrell, Discriminative gaussian process latent variable model for classification, in: Proceedings of the 24th International Conference on Machine Learning, ICML'07, 2007.
[7] X. Gao, X. Wang, D. Tao, X. Li, IEEE Trans. Syst. Man Cybern. Part B 41 (2) (2011) 425–434.
[8] X. Jiang, J. Gao, T. Wang, L. Zheng, IEEE Trans. Syst. Man Cybern. Part B 42 (6) (2012) 1620–1632.
[9] S. Eleftheriadis, O. Rudovic, M. Pantic, IEEE Trans. Image Process. 24 (1) (2015) 189–204.
[10] A.K. Jain, M.N. Murty, P.J. Flynn, ACM Comput. Surv. 31 (3) (1999) 264–323.
[11] J.A. Hartigan, M.A. Wong, Appl. Stat. 28 (1) (2013) 100–108.
[12] M.M. Adankon, M. Cheriet, Support Vector Machine, Springer, US, 2015.
[13] S.C. Kothari, H. Oh, Neural Networks for Pattern Recognition, MIT Press, 1993.
[14] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York, Inc., 2006.
[15] N. Lawrence, J. Mach. Learn. Res. 6 (3) (2005) 1783–1816.
[16] J. Philbin, J. Sivic, A. Zisserman, Int. J. Comput. Vis. 95 (2) (2011) 138–153.

[17] B. Brosseau-Villeneuve, J.Y. Nie, N. Kando, Inf. Retr. 17 (1) (2014) 21−51.
[18] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, IEEE Signal Process. Mag. 29 (6) (2012) 82−97.
[19] S. Maneeroj, A. Takasu, Hybrid recommender system using latent features, in: International Conference on Advanced Information NETWORKING and Applications Workshops, 2009, pp. 661−666.
[20] X.Y. Liu, Z.W. Liao, Z.S. Wang, W.F. Chen, Int. Conf. Mach. Learn. Cybern. (2006) 4155−4159.
[21] A.M. Martínez, A.C. Kak, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2) (2001) 228−233.
[22] D. Petelin, Int. J. Neural Syst. 14 (6) (2006) 3011−3015.
[23] R. Calandra, J. Peters, C. E. Rasmussen, M. P. Deisenroth, Manifold gaussian processes for regression, arXiv preprint arXiv:1402.5876.
[24] Z. Qiang, J. Ma, Automatic Model Selection of the Mixtures of Gaussian Processes for Regression, Springer International Publishing, 2015.
[25] E.V. Bonilla, K.M.A. Chai, C.K.I. Williams, Multi-task gaussian process prediction, in: Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December, 2007.
[26] H.C. Kim, J. Lee, Neural Comput. 19 (11) (2007) 3088−3107.
[27] B. Abolhasanzadeh, Gaussian process latent variable model for dimensionality reduction in intrusion detection, in: Electrical Engineering, 2015.
[28] M.K. Titsias, N.D. Lawrence, Bayesian gaussian process latent variable model, in: Proceedings of the Thirteenth International Workshop on Artificial Intelligence & Statistics Jmlr W & Cp, vol. 9, 2010, pp. 844−851 (9).
[29] S. Eleftheriadis, O. Rudovic, M. Pantic, Shared gaussian process latent variable model for multi-view facial expression recognition, in: International Symposium on Visual Computing, 2013, pp. 527−538.
[30] C.H. Ek, P.H.S. Torr, N.D. Lawrence, Gaussian process latent variable models for human pose estimation, in: Machine Learning for Multimodal Interaction, International Workshop, Mlmi 2007, Czech Republic, Brno, 2007, pp. 132−143. June 28−30, 2007, Revised Selected Papers.
[31] L. Cai, L. Huang, C. Liu, Multimedia Tools Appl. (2015) 1−18.
[32] G. Song, S. Wang, Q. Huang, Q. Tian, Similarity gaussian process latent variable model for multi-modal data analysis, in: IEEE International Conference on Computer Vision, 2015, pp. 4050−4058.
[33] B. Schlkopf, A. Smola, K. Mller, Neural Comput. 10 (5) (1998) 1299−1319.
[34] T. Cover, P. Hart, IEEE Trans. Inf. Theory 13 (1) (1967) 21−27.
[35] M.L. Zhang, Z.H. Zhou, Pattern Recognit. 40 (7) (2007) 2038−2048.
[36] P. Hall, T.C. Hu, J.S. Marron, Ann. Stat. 23 (1) (1995) 1−10.
[37] L. Devroye, A. Krzyak, Stat. Prob. Lett. 44 (3) (1999) 299−308.
[38] F. Bohnert, I. Zukerman, D.F. Schmidt, Using gaussian spatial processes to model and predict interests in museum exhibits, in: The Workshop on Intelligent Techniques for Web Personalization & Recommender Systems, 2009.
[39] W. Herlands, A. Wilson, H. Nickisch, S. Flaxman, D. Neill, W. V. Panhuis, E. Xing, Scalable gaussian processes for characterizing multidimensional change surfaces, arXiv preprint arXiv:1511.04408.
[40] A. Datta, S. Banerjee, A.O. Finley, A.E. Gelfand, J. Am. Stat. Assoc. (2015) (just-accepted).
[41] X. Wang, X. Gao, Y. Yuan, D. Tao, J. Li, Neurocomputing 73 (10−12) (2010) 2186−2195.
[42] C. Kang, S. Liao, Y. He, J. Wang, W. Niu, S. Xiang, C. Pan, Cross-modal similarity learning: a low rank bilinear formulation, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM'15, 2015.
[43] N.D. Lawrence, A.J. Moore, Hierarchical gaussian process latent variable models, in: Machine Learning, Proceedings of the Twenty-Fourth International Conference, 2007, pp. 481−488.
[44] F. Yousefi, Z. Dai, C. H. Ek, N. Lawrence, Unsupervised learning with imbalanced data via structure consolidation latent variable model, arXiv preprint arXiv:1607.00067.
[45] E. Snelson, Z. Ghahramani, Adv. Neural Inf. Process. Syst. 18 (1) (2006) 1257−1264.
[46] E. Snelson, Z. Ghahramani, Local and global sparse gaussian process approximations, in: Proceedings of Artificial Intelligence and Statistics (AISTATS 2), 2007, pp. 524−531.
[47] T.V. Nguyen, E.V. Bonilla, Fast allocation of gaussian process experts, in: In International Conference on Machine Learning, 2014.
[48] Y. Gal, M. V. D. Wilk, Variational inference in sparse gaussian process regression and latent variable models − a gentle tutorial, arXiv preprint arXiv:1402.1412.
[49] N.D. Lawrence, J. Mach. Learn. Res. 2 (2007) 243−250.
[50] N.D. Lawrence, J. Quiñonero Candela, Local distance preservation in the gp-lvm through back constraints, in: Proceedings of the 23rd International Conference on Machine Learning, ICML'06, 2006.
[51] S.J. Pan, Q. Yang, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345−1359.
[52] S. Roweis, Adv. Neural Inf. Process. Syst. 10 (1999) 626−632.
[53] S. Yu, K. Yu, V. Tresp, H.P. Kriegel, M. Wu, Supervised probabilistic principal component analysis, in: Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August, 2006, pp. 464−473.
[54] X. Gao, X. Wang, X. Li, D. Tao, Pattern Recognit. 44 (10−11) (2011) 2358−2366.
[55] Y. Fu, L. Wang, Y. Guo, Comput. Sci. 12 (7) (2014) 717−729.
[56] P. Xie, E.P. Xing, Multi-modal distance metric learning, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI'13, 2013.
[57] T. Wang, S. Gong, X. Zhu, S. Wang, IEEE Trans. Pattern Anal. Mach. Intell. (2016), 1−1.
[58] A.P. Shon, K. Grochow, A. Hertzmann, R.P.N. Rao, Adv. Neural Inf. Process. Syst. (2005) 1233−1240.
[59] R. K. C. Fan, Spectral graph theory, Published for the Conference Board of the Mathematical Sciences by the American Mathematical Society, 1997.
[60] O. Stegle, C. Lippert, J.M. Mooij, N.D. Larence, K. Borgwardt, Adv. Neural Inf. Process. Syst. (2011) 2011.
[61] F. Yan, Z. Xu, Y. A. Qi, Sparse matrix-variate gaussian process blockmodels for network modeling, arXiv preprint arXiv:1202.3769.
[62] O. Koyejo, C. Lee, J. Ghosh, Mach. Learn. 97 (1−2) (2014) 103−127.
[63] O. Koyejo, L. Cheng, J. Ghosh, The trace norm constrained matrix-variate gaussian process for multitask bipartite ranking, arXiv preprint arXiv:1302.2576.
[64] N. Houlsby, J.M. Hernndez-Lobato, F. Huszr, Z. Ghahramani, Adv. Neural Inf. Process. Syst. 3 (2012) 2096−2104.
[65] M.J. Wainwright, M.I. Jordan, Found. Trends Mach. Learn. 1 (12) (2010) 1−305.
[66] A. C. Damianou, M. K. Titsias, N. D. Lawrence, Variational inference for uncertainty on the inputs of gaussian process models, arXiv preprint arXiv:1409.2287.
[67] R. Andrade Pacheco, J. Hensman, M. Zwiessele, N. Lawrence, Hybrid discriminative-generative approach with gaussian processes, in: Proceedings of the Thirteenth International Workshop on Artificial Intelligence & Statistics Jmlr W & Cp, 2014, pp. 47−56.
[68] J.M. Wang, D.J. Fleet, A. Hertzmann, Gaussian process dynamical models, in: In NIPS, MIT Press, 2006, pp. 1441−1448.
[69] J.M. Wang, D.J. Fleet, A. Hertzmann, IEEE Trans. Pattern Anal Mach. Intell. 30 (2) (2008) 283−298.
[70] A.C. Damianou, M.K. Titsias, N.D. Lawrence, Variational gaussian process dynamical systems, in: Advances in Neural Information Processing System, IEEE Conf. Publications, 2011, pp. 2510−2518.
[71] H. Xiong, T. Liu, D. Tao, H. Shen, IEEE Trans. Image Process. 25 (8) (2016), 1−1.
[72] Z. Dai, A. Damianou, J. Gonzlez, N. Lawrence, Comput. Sci. 14 (9) (2015) 3942−3951.
[73] Y. Bengio, Found. Trends Mach. Learn. 2 (1) (2009) 1−127.
[74] G.E. Hinton, Scholarpedia 4 (6) (2009) 786−804.
[75] J. Masci, U. Meier, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: In International Conference on Artificial Neural Networks, 2011.
[76] A.C. Damianou, N.D. Lawrence, Comput. Sci. (2012) 207−215.
[77] T. D. Bui, D. Hernndezlobato, Y. Li, J. M. Hernndezlobato, R. E. Turner, Deep gaussian processes for regression using approximate expectation propagation, arXiv preprint arXiv:1602.04133.
[78] Y. Gal, Y. Chen, Z. Ghahramani, Statistics (2015) 645−654.

Ping Li received his B.S. and M.S. degree in Management Science & Engineering from Anhui University of Technology in 2011 and 2014. He is currently pursuing the Ph.D. degree with the College of Computer Science & Technology, Nanjing University of Aeronautics and Astronautics. Her research interests include pattern recognition and machine learning.

Songcan Chen received his B.S. degree in mathematics from Hangzhou University (now merged into Zhejiang University) in 1983. In 1985, he completed his M.S. degree in computer applications at Shanghai Jiaotong University and then worked at NUAA in January 1986. There he received a Ph.D. degree in communication and information systems in 1997. Since 1998, as a full-time professor, he has been with the College of Computer Science & Technology at NUAA. His research interests include pattern recognition, machine learning and neural computing.