Feature-correlation-aware Gaussian Process Latent Variable Model

Ping Li PING.LI.NJ@NUAA.EDU.CN

Songcan Chen* S.CHEN@NUAA.EDU.CN

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

Editors: Jun Zhu and Ichiro Takeuchi

Abstract

Gaussian Process Latent Variable Model (GPLVM) is a powerful nonlinear dimension reduction model and has been widely used in many machine learning scenarios. However, the original GPLVM and its variants do not explicitly model the correlations among the original features, leading to the underutilization of underlying information involved in the data. To compensate for this shortcoming, we propose a feature-correlation-aware GPLVM (fcaGPLVM) to simultaneously learn the latent variables and the feature correlations. The main contributions of this paper are 1) introducing a set of extra latent variables into the original GPLVM and proposing a feature-correlation-aware kernel function to explicitly model the feature-description information and infer the feature correlations; 2) defining a joint objective function and developing a stochastic optimization algorithm based on the stochastic variational inference (SVI) to learn all the latent variables. To the best of our knowledge, this is the first work that explicitly considers the feature correlations in the GPLVM and makes many existing GPLVMs become its special cases. Furthermore, it can be applied to both unsupervised and supervised learnings to improve the performance of dimension reduction. Experimental results show that in these two learning scenarios the proposed models outperform their state-of-the-art counterparts.

Keywords: Gaussian Process, GPLVM, feature correlations, stochastic optimization

1. Introduction

Gaussian Process Latent Variable Model (GPLVM) (Lawrence, 2005) is a powerful nonlinear dimension reduction model and can effectively discover the low dimensional manifold embedded in the high dimensional space. Furthermore, it naturally inherits the characteristics of probabilistic non-parametric Gaussian Process (GP) such as uncertainty quantification (Iwata and Ghahramani, 2017), flexible non-parametric modeling (Huang et al., 2015a), etc., and has been widely used in many machine learning scenarios (Lu and Tang, 2015; Xiong and Tao, 2017).

To date, many extensions to the conventional GPLVM have been proposed, including local distance preservation GPLVM (Lawrence and Quiñonero-Candela, 2006), supervised

* This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61672281 and 61472186, the Key Program of NSFC under Grant No. 61732006 and the founding of Jiangsu Innovation Program for Graduate Education under Grant No. KYLX15_0323. (Corresponding author: Songcan Chen.)

GPLVMs (Urtasun and Darrell, 2007; Gao et al., 2011; Jiang et al., 2012), semi-supervised GPLVM (Wang et al., 2010), shared GPLVMs (Eleftheriadis et al., 2013, 2015; Song et al., 2017), transfer GPLVM (Gao et al., 2010) and so on. All these GPLVMs just inherit the basic structure of the original GPLVM and assume that all the features in the original space are independent conditioned on the latent variables. Obviously, these models just implicitly model the feature-correlation information into the low-dimensional latent variables and mainly focus on learning representative latent variables for tasks such as classification, regression etc., leading to the underutilization of underlying information involved in the data and limiting the final performance. In fact, feature correlation is as important as the feature value itself. It often involves explicit or implicit valuable information that is beneficial to the promotion of dimension reduction tasks. For example, when dealing with structured data such as spatial and temporal data (whose feature correlations are known), we can significantly improve the performance of related learning algorithms by utilizing such correlation information (Cai et al., 2007; Huang et al., 2015b). For more general data whose feature correlations are unknown, a learning-from-data method (Hsieh et al., 2011) can also firstly be used to infer the feature correlations. It is these above-mentioned observations that motivate us to propose a ***F***eature-***C***orrelation-***A***ware GPLVM (fcaGPLVM) to simultaneously learn the low dimensional latent variables and the feature correlations. Compared with the existing GPLVMs, our fcaGPLVM has three advantages:

- it can explicitly model and learn the correlations among the original input features.
- it scales well with large-scale data by using stochastic variational inference (SVI) method for the optimization.
- it can be applied to both supervised and unsupervised learning scenarios to improve the performance of dimension reduction.

To the best of our knowledge, this is the first work that explicitly models the feature correlations in the GPLVM and makes many existing GPLVMs become its special cases. Experimental results on four real-world data sets demonstrate that, in both the unsupervised and supervised scenarios, the proposed models outperform their state-of-the-art counterparts.

The rest of this paper is organized as follows. In Section 2, we review the original GPLVM and other related GP-based models. In Section 3, we firstly propose our fcaGPLVM for latent variables and feature correlations learnings. Secondly, we develop a stochastic fcaGPLVM to speed up the model optimization. Finally, we propose a supervised fcaGPLVM to utilize the label information of data. Section 4 gives three special cases of fcaGPLVM and shows that, in these three cases, fcaGPLVM becomes the existing GPLVMs. We present the experimental results in Section 5 and give conclusion in Section 6.

2. Related Work

2.1. Gaussian Process Latent Variable Model

In original GPLVM (Lawrence, 2005), given N training samples $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N]^T \in \mathbb{R}^{N \times D}$, our goal is to learn the corresponding low dimensional latent variables $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]^T$

where $\boldsymbol{x}_n \in \mathbb{R}^Q$ denotes the latent variable of the observed data $\boldsymbol{y}_n$ $(n = 1, \cdots, N)$. In general, $Q \ll D$, thus we realize the dimension reduction. We assume that the d^{th} feature of $\boldsymbol{y}_n$ is generated by function f_d with input $\boldsymbol{x}_n$:

$$y_{nd} = f_d(\boldsymbol{x}_n) + \epsilon_{nd} \tag{1}$$

where ϵ_{nd} denotes the noise term which follows a Gaussian distribution: $\epsilon_{nd} \sim \mathcal{N}(0, \sigma^2)$. For all the functions $\{f_d\}_{d=1}^D$, we assume that they have the same GP prior and thus the vector $\boldsymbol{f}_d \in \mathbb{R}^N$ (composed by the function values of all the N training samples) follows a multivariate Gaussian distribution:

$$p(\boldsymbol{f}_d) = \mathcal{N}(\boldsymbol{f}_d|0, \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}) \tag{2}$$

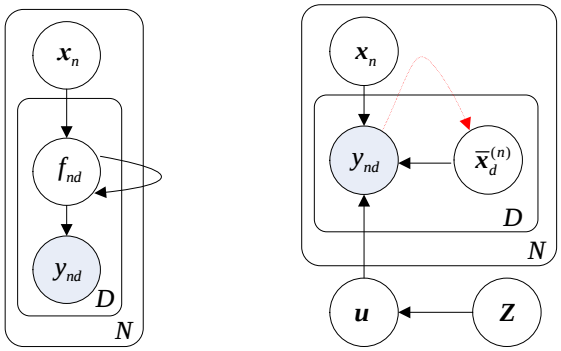
where $\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} \in \mathbb{R}^{N\times N}$ denotes the kernel matrix computed by using the kernel function (e.g., RBF kernel) $k(\cdot,\cdot)$ on $\boldsymbol{X}$. By integrating out all the D variables $\{\boldsymbol{f}_d\}_{d=1}^D$, we write the marginal likelihood as

$$p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}) = \prod_{d=1}^{D} \frac{\exp\left(-\frac{1}{2}\boldsymbol{y}_{:,d}^T(\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}_{:,d}\right)}{(2\pi)^{N/2}|\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} + \sigma^2\boldsymbol{I}|^{1/2}} \tag{3}$$

where $\boldsymbol{y}_{:,d} \in \mathbb{R}^N$ denotes the d^{th} column of matrix $\boldsymbol{Y}$; $\boldsymbol{\theta}$ denotes all the hyper-parameters (the hyper-parameters in the kernel function and noise distribution) involved in GPLVM. As a result, we can seek the maximum likelihood solution for the hyper-parameters in $\boldsymbol{\theta}$ and the latent variables by maximizing (3). The graphical model representation of GPLVM is shown in Figure 1(a). Obviously, GPLVM just makes a conditional independence assumption on the features of $\boldsymbol{Y}$, thus losing the feature-correlation information.

2.2. Modeling feature correlations

In the research of GP, we often consider GPLVM as a multiple-output prediction model where only the output data are given. Thus we should respectively review two kinds of GP-based models: prediction and latent variable models. On the one hand, in the context of the former kind of model, the most relevant work to ours is the Kronecker GP (Bonilla et al., 2008) which is actually a special design for multi-task learning. However, through analyzing the formulation of this model, we find that it can also be extended to GPLVM for feature correlation learning as a special case of our model under certain settings (see Section 4 for more detail). On the other hand, in the context of GP-based latent variable models, the most relevant works to ours are Dai et al. (2017) and Bodin et al. (2017), both of which introduce a set of additional latent variables to expand the input domain for the constructions of multi-output GP (Jawanpuria et al., 2015) and non-stationary GP (Lázaro-Gredilla and Titsias, 2011) respectively. In this paper, inspired by both, we also introduce a set of extra latent variables for the inference of feature correlations. However, both the structure and the solution of our model highlight its significant differences from those above-mentioned two models. More specifically, instead of using only a set of shared latent variables to model the output information in Dai et al. (2017) or using only one latent variable for one sample to model the non-stationary behaviour of function in Bodin



(a) GPLVM (b) Stochastic fcaGPLVM

Figure 1: The graphical model representations of the original GPLVM and stochastic fcaGPLVM. The red dashed arrows in 1(b) denotes the projection functions $\{h_{(d)}\}_{d=1}^{D}$.

et al. (2017), we impose a set of latent variables on each sample to model its feature-description information, which results in a specially designed model structure. In order to incorporate the feature correlation learning mechanism into this model, we firstly propose a feature-correlation kernel function and then define a joint objective function with respect to all the latent variables. At last, we develop a stochastic optimization algorithm by using the stochastic variational inference (SVI) (Hoffman et al., 2013; Hensman et al., 2013) to maximize the objective.

3. The Proposed Model

The original GPLVM just assumes that all the dimensions of the original data are independent conditioned on the latent variables, thus leaving the room of performance promotion. In this section, we first propose our fcaGPLVM and then develop a stochastic fcaGPLVM for efficiently model learning. At last we also explore the supervised learning of fcaGPLVM by utilizing the label information to improve the performance of dimension reduction.

3.1. Feature-correlation-aware GPLVM

In order to model the feature correlations into the GPLVM, apart from the latent $\boldsymbol{x}_n$, we also assume that there are D feature-description variables $\bar{\boldsymbol{X}}^{(n)} = [\bar{\boldsymbol{x}}_1^{(n)}, ..., \bar{\boldsymbol{x}}_D^{(n)}]^T \in \mathbb{R}^{D\times P}$ for the n^{th} sample $\boldsymbol{y}_n$. Thus $\bar{\boldsymbol{X}}^{(n)}$ contains the description information of D features in sample $\boldsymbol{y}_n$ and we can synchronously learn the variable $\boldsymbol{x}_n$ and $\bar{\boldsymbol{X}}^{(n)}$ (for $n = 1, \cdots, N$) to realize the dimension reduction and infer the feature correlations. Specifically, we define that the n^{th} row and d^{th} column element of matrix $\boldsymbol{Y}$ is obtained as follows:

$$y_{nd} = f(\boldsymbol{x}_n, \bar{\boldsymbol{x}}_d^{(n)}) + \epsilon_{nd} \tag{4}$$

where ϵ_{nd} denotes the noise term that follows a Gaussian distribution: $\epsilon_{nd} \sim \mathcal{N}(0, \sigma^2)$. We denote $\boldsymbol{F}$ as the noise-free part of $\boldsymbol{Y}$ where the n^{th} row and d^{th} column element of matrix

$\boldsymbol{F}$ is $f_{nd} = f(\boldsymbol{x}_n, \bar{\boldsymbol{x}}_d^{(n)})$. Assuming that function f follows a GP prior, we have

$$vec(\boldsymbol{F}) \sim \mathcal{N}(\boldsymbol{0}, cov(\boldsymbol{F}, \boldsymbol{F})) \tag{5}$$

where $vec(\cdot)$ denotes the *vectorization* operator of matrix; $cov(\boldsymbol{F}, \boldsymbol{F}) \in \mathbb{R}^{ND \times ND}$ is the covariance matrix that encodes the correlation between arbitrary two elements of matrix $\boldsymbol{F}$. We therefore should design a suitable kernel function that can utilize the information involved in both the low dimensional latent variables and feature-description variables. In this paper, we propose the following feature-correlation-aware kernel function for the computation of the covariance matrix $cov(\boldsymbol{F}, \boldsymbol{F})$:

$$\hat{k}_{(d-1)N+n,(t-1)N+j} = cov(f_{nd}, f_{jt}) = \bar{k}(\bar{\boldsymbol{x}}_d^{(n)}, \bar{\boldsymbol{x}}_t^{(j)}) k(\boldsymbol{x}_n, \boldsymbol{x}_j) \tag{6}$$

where $\bar{k}(\cdot,\cdot)$ and $k(\cdot,\cdot)$ are two different kernels (e.g., two different RBF kernels) for feature-description variables and low dimensional latent variables. Then, we use the kernel matrix $\hat{\boldsymbol{K}} \in \mathbb{R}^{ND \times ND}$ (induced by the kernel function $\hat{k}$) to replace the covariance matrix $cov(\boldsymbol{F}, \boldsymbol{F})$ where $\hat{\boldsymbol{K}}$ has the following form:

$$\hat{\boldsymbol{K}} = \bar{\boldsymbol{K}}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} \circ (\boldsymbol{E}_D \otimes \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}) \tag{7}$$

$\bar{\boldsymbol{X}} = [\bar{\boldsymbol{x}}_1^{(1)}, \cdots, \bar{\boldsymbol{x}}_1^{(N)}, \cdots, \bar{\boldsymbol{x}}_D^{(1)}, \cdots, \bar{\boldsymbol{x}}_D^{(N)}]^T \in \mathbb{R}^{ND \times P}$; $\bar{\boldsymbol{K}}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}$ denotes the kernel matrix computed by using kernel function $\bar{k}(\cdot,\cdot)$ on $\bar{\boldsymbol{X}}$; $\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}$ is the same as in Equation (2); $\boldsymbol{E}_D \in \mathbb{R}^{D \times D}$ is a matrix with all elements being one; $\otimes$ denotes the *Kronecker* product; $\circ$ denotes the *Hadamard* product; $\hat{k}_{(d-1)N+n,(t-1)N+j}$ denotes the $((d-1)N+n)^{th}$ row and $((t-1)N+j)^{th}$ column element of $\hat{\boldsymbol{K}}$.

Making a deep analysis on the GP that uses the above formulated kernel function, we can find that it models the covariance of the two elements f_{nd} and f_{jt} in matrix $\boldsymbol{F}$ by four latent variables $\bar{\boldsymbol{x}}_d^{(n)}$, $\bar{\boldsymbol{x}}_t^{(j)}$, $\boldsymbol{x}_n$ and $\boldsymbol{x}_j$. Thus the kernel functions $\bar{k}(\bar{\boldsymbol{x}}_d^{(n)}, \bar{\boldsymbol{x}}_t^{(j)})$ and $k(\boldsymbol{x}_n, \boldsymbol{x}_j)$ are used to encode the correlation between features and samples respectively. Different from the original GPLVM which models the covariance between f_{nd} and f_{jt} as $k(\boldsymbol{x}_n, \boldsymbol{x}_j)$ if $d = t$, and 0 otherwise, our fcaGPLVM models the covariance as the product of two terms ($\bar{k}(\bar{\boldsymbol{x}}_d^{(n)}, \bar{\boldsymbol{x}}_t^{(j)})$ and $k(\boldsymbol{x}_n, \boldsymbol{x}_j)$) by which we can explicitly obtain the feature-description variables and thus the feature correlations.

However, as we can see, many more latent variables need to be learned in fcaGPLVM ($\bar{\boldsymbol{X}}$ and $\boldsymbol{X}$), which may lead to over-fitting and lower convergence rate. To address these problems, we further assume that $\bar{\boldsymbol{x}}_d^{(n)}$ (for $d = 1, ..., D$) is obtained by the following projection function:

$$\bar{\boldsymbol{x}}_d^{(n)} = h^{(d)}(\boldsymbol{y}_n) \tag{8}$$

where $h^{(d)}$ is a vector-valued function $\mathbb{R}^D \to \mathbb{R}^P$. Our goal therefore is to learn the latent variable $\boldsymbol{x}_n$ for $n = 1, ..., N$ and projection function $h^{(d)}$ for $d = 1, ..., D$. By integrating out $\boldsymbol{F}$, we get the following marginal likelihood:

$$p\left(\boldsymbol{Y}|\boldsymbol{X}, \{h^{(d)}\}_{d=1}^D\right) = \frac{\exp\left(-\frac{1}{2} vec(\boldsymbol{Y})^T (\hat{\boldsymbol{K}} + \sigma^2 \boldsymbol{I})^{-1} vec(\boldsymbol{Y})\right)}{(2\pi)^{ND/2} |\hat{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}|^{1/2}} \tag{9}$$

It is worth to note that, for notational convenience, we have dropped the dependency on the hyper-parameters in $\boldsymbol{\theta}$ during the derivation process. We can consider the log likelihood $\mathcal{L} = \ln p\left(vec(\boldsymbol{Y})|\boldsymbol{X}, \{h^{(d)}\}_{d=1}^{D}\right)$ as the joint objective function and maximize it to learn the optimal latent variables, hyper-parameters and projection functions.

3.2. Stochastic Feature-correlation-aware GPLVM

As we can see, during the optimization process, we need to compute the inversion of a $ND \times ND$ matrix $(\hat{\boldsymbol{K}} + \sigma^2\boldsymbol{I})$ with a computation complexity $\mathcal{O}(N^3D^3)$ and a storage complexity $\mathcal{O}(N^2D^2)$ which are so high that ordinary computers are unaffordable. To reduce the complexities of GP, many sparse GP method have been proposed (Titsias, 2009; Hensman et al., 2013; Wang et al., 2014; Xu et al., 2014). In this subsection, we develop a stochastic fcaGPLVM via using the stochastic variational inference (Hoffman et al., 2013; Hensman et al., 2013). Specifically, we introduce M auxiliary points $\boldsymbol{Z} = [\boldsymbol{z}_1, ..., \boldsymbol{z}_M]^T \in \mathbb{R}^{M\times Q}$ which are in the same latent space as the latent variables in $\boldsymbol{X}$. Correspondingly, we also define the outputs of fcaGPLVM (with the inputs in $\boldsymbol{Z}$) as $\boldsymbol{u} = [u_1, ..., u_M]^T \in \mathbb{R}^M$ where we just assume that the output u_m of $\boldsymbol{z}_m$ is a scalar. In this case, the feature-description variables of the auxiliary points must be defined as $\bar{\boldsymbol{Z}} = [\bar{\boldsymbol{z}}_1, ..., \bar{\boldsymbol{z}}_M]^T \in \mathbb{R}^{M\times P}$. Thus we can use all these auxiliary variables to improve the scalability of fcaGPLVM. The generation process of $\boldsymbol{Y}$ is defined as

$$p(\boldsymbol{u}) = \mathcal{N}\left(\boldsymbol{u}|\boldsymbol{0}, \hat{\boldsymbol{K}}_{\boldsymbol{ZZ}}\right) \tag{10}$$

$$p(\boldsymbol{F}|\boldsymbol{u}, \boldsymbol{Z}, \bar{\boldsymbol{Z}}, \boldsymbol{X}) = \mathcal{N}\left(vec(\boldsymbol{F})|\hat{\boldsymbol{K}}_{\boldsymbol{XZ}}\hat{\boldsymbol{K}}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{u}, \tilde{\boldsymbol{K}}\right) \tag{11}$$

$$p(\boldsymbol{Y}|\boldsymbol{F}) = \mathcal{N}\left(vec(\boldsymbol{Y})|vec(\boldsymbol{F}), \sigma^2\boldsymbol{I}\right) \tag{12}$$

where $\hat{\boldsymbol{K}}_{\boldsymbol{ZZ}} = \bar{\boldsymbol{K}}_{\bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}} \circ \boldsymbol{K}_{\boldsymbol{ZZ}} \in \mathbb{R}^{M\times M}$; $\hat{\boldsymbol{K}}_{\boldsymbol{XZ}} = \bar{\boldsymbol{K}}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{Z}}} \circ (\boldsymbol{E}_{D\times 1} \otimes \boldsymbol{K}_{\boldsymbol{XZ}}) \in \mathbb{R}^{ND\times M}$; $\boldsymbol{E}_{D\times 1} \in \mathbb{R}^{D\times 1}$ is a matrix with all elements being one; $\tilde{\boldsymbol{K}} = \hat{\boldsymbol{K}} - \hat{\boldsymbol{K}}_{\boldsymbol{XZ}}\hat{\boldsymbol{K}}_{\boldsymbol{ZZ}}^{-1}\hat{\boldsymbol{K}}_{\boldsymbol{XZ}}^T$. In the derivations, we should integrate out $\boldsymbol{F}$ and $\boldsymbol{u}$ to obtain the margin likelihood. We therefore use the stochastic variational inference method to derive out a lower bound of the log likelihood. Assuming the variational posterior distribution of $\boldsymbol{u}$ to be $q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}|\boldsymbol{m}, \boldsymbol{S})$ ($\boldsymbol{m} \in \mathbb{R}^M$ and $\boldsymbol{S} \in \mathbb{R}^{M\times M}$ are the variational parameters) and following the derivations in Hensman et al. (2013), the log marginal likelihood can be expressed as:

$$\begin{aligned}\mathcal{L} &= \ln \int p(\boldsymbol{Y}|\boldsymbol{u}, \boldsymbol{Z}, \bar{\boldsymbol{Z}}, \boldsymbol{X})p(\boldsymbol{u})d\boldsymbol{u} \\ &= \int q(\boldsymbol{u}) \ln\left(\frac{p(\boldsymbol{Y}|\boldsymbol{u}, \boldsymbol{Z}, \bar{\boldsymbol{Z}}, \boldsymbol{X})p(\boldsymbol{u})}{q(\boldsymbol{u})}\right) d\boldsymbol{u} - \int q(\boldsymbol{u}) \ln\left(\frac{p(\boldsymbol{u}|\boldsymbol{Y}, \boldsymbol{Z}, \bar{\boldsymbol{Z}}, \boldsymbol{X})}{q(\boldsymbol{u})}\right) d\boldsymbol{u}\end{aligned} \tag{13}$$

Since the KL divergence $KL(q(\boldsymbol{u})||p(\boldsymbol{u}|\boldsymbol{Y}, \boldsymbol{Z}, \bar{\boldsymbol{Z}}, \boldsymbol{X})) = -\int q(\boldsymbol{u}) \ln\left(\frac{p(\boldsymbol{u}|\boldsymbol{Y}, \boldsymbol{Z}, \bar{\boldsymbol{Z}}, \boldsymbol{X})}{q(\boldsymbol{u})}\right) d\boldsymbol{u} \geq 0$, the remaining part $\mathcal{L}_0 = \int q(\boldsymbol{u}) \ln\left(\frac{p(\boldsymbol{Y}|\boldsymbol{u}, \boldsymbol{Z}, \bar{\boldsymbol{Z}}, \boldsymbol{X})p(\boldsymbol{u})}{q(\boldsymbol{u})}\right) d\boldsymbol{u}$ can be considered as a lower bound of $\mathcal{L}$. This lower bound is maximized when the KL divergence $KL(q(\boldsymbol{u})||p(\boldsymbol{u}|\boldsymbol{Y}, \boldsymbol{Z}, \bar{\boldsymbol{Z}}, \boldsymbol{X}))$ vanishes, which occurs when $q(\boldsymbol{u})$ equals the posterior distribution $p(\boldsymbol{u}|\boldsymbol{Y}, \boldsymbol{Z}, \bar{\boldsymbol{Z}}, \boldsymbol{X})$. We

further apply Jensen's inequality to $\ln p(\boldsymbol{Y}|\boldsymbol{u})$ and obtain

$$\ln p(\boldsymbol{Y}|\boldsymbol{u},\boldsymbol{Z},\bar{\boldsymbol{Z}},\boldsymbol{X}) = \ln \int p(\boldsymbol{Y}|\boldsymbol{F})p(\boldsymbol{F}|\boldsymbol{u},\boldsymbol{Z},\bar{\boldsymbol{Z}},\boldsymbol{X})d\boldsymbol{F} \quad \geq \int p(\boldsymbol{F}|\boldsymbol{u},\boldsymbol{Z},\bar{\boldsymbol{Z}},\boldsymbol{X}) \ln p(\boldsymbol{Y}|\boldsymbol{F})d\boldsymbol{F} \tag{14}$$

By using (13) and (14), a lower bound of $\mathcal{L}$ is immediately obtained:

$$\begin{aligned}\mathcal{L} \geq \mathcal{LB}(\boldsymbol{Y}) &= \int q(\boldsymbol{u}) \ln \left[\frac{\exp\left[\int p(\boldsymbol{F}|\boldsymbol{u},\boldsymbol{Z},\bar{\boldsymbol{Z}},\boldsymbol{X}) \ln p(\boldsymbol{Y}|\boldsymbol{F})d\boldsymbol{F}\right] p(\boldsymbol{u})}{q(\boldsymbol{u})} \right] d\boldsymbol{u} \\ &= \sum_{n=1}^{N}\sum_{d=1}^{D} \left[\ln \mathcal{N}\left(y_{nd}|\hat{\boldsymbol{k}}_i \hat{\boldsymbol{K}}_{\boldsymbol{ZZ}}^{-1}\boldsymbol{m}, \sigma^2\right) - \frac{1}{2\sigma^2}\tilde{k}_{i,i} - \frac{1}{2}tr\left(\boldsymbol{S}\boldsymbol{\Lambda}_i\right) \right] - KL(q(\boldsymbol{u})||p(\boldsymbol{u}))\end{aligned} \tag{15}$$

where $i = (d-1)N+n$; $\hat{\boldsymbol{k}}_i \in \mathbb{R}^{1\times M}$ denotes the i^{th} row of matrix $\hat{\boldsymbol{K}}_{\boldsymbol{XZ}}$; $\boldsymbol{\Lambda}_i = \frac{1}{\sigma^2}\hat{\boldsymbol{K}}_{\boldsymbol{ZZ}}^{-1}\hat{\boldsymbol{k}}_i^T\hat{\boldsymbol{k}}_i\hat{\boldsymbol{K}}_{\boldsymbol{ZZ}}^{-1}$; $\tilde{k}_{i,i}$ is the i^{th} diagonal element of matrix $\tilde{\boldsymbol{K}}$.

For the projection functions $\{h^{(1)},\cdots,h^{(D)}\}$, we just assume that they are all linear functions[1]:

$$\bar{\boldsymbol{x}}_d^{(n)} = h^{(d)}(\boldsymbol{y}_n) = {\boldsymbol{W}_{(d)}}^T\boldsymbol{y}_n + \boldsymbol{b}_{(d)}, \ \ for\ d = 1,\cdots,D. \tag{16}$$

where $\boldsymbol{W}_{(d)} \in \mathbb{R}^{D\times P}$ and $\boldsymbol{b}_{(d)} \in \mathbb{R}^P$ denote the weight and the bias parameters respectively. To avoid the over-fitting problem, we also impose a l_1-norm regularization term on each $\boldsymbol{W}_{(d)}$ and therefore maximize the following joint objective function to learn the model:

$$\mathcal{LB}_r(\boldsymbol{Y}) = \mathcal{LB}(\boldsymbol{Y}) - \lambda_1 \sum_{d=1}^{D} \|\boldsymbol{W}_{(d)}\|_1 \tag{17}$$

where $\lambda_1 > 0$ denotes the regularization hyper-parameter. Obviously, $\mathcal{LB}_r$ can be factorized into N terms with respect to each training sample. Thus, a stochastic gradient method can be used to train fcaGPLVM. Specifically, we use the mini-batch mode (with batch size B) to improve the precision of the gradient computation. As shown in Figure 1(*b*), the variables in stochastic fcaGPLVM can be divided into two categories: (1) local variable $\boldsymbol{X}$; (2) global variables including $\boldsymbol{m}$, $\boldsymbol{S}$, $\{\boldsymbol{W}_{(d)},\boldsymbol{b}_{(d)}\}_{d=1}^D$, $\boldsymbol{Z}$, $\bar{\boldsymbol{Z}}$ and the hyper-parameters in $\boldsymbol{\theta}$. After initializing all the variables, in each iteration, we firstly randomly draw B samples (denoted by $\boldsymbol{Y}^{(B)}$) from $\boldsymbol{Y}$ and then maximize $\mathcal{LB}_r(\boldsymbol{Y}^{(B)})$ (with fixed global variables) to learn all the local variables for $\boldsymbol{Y}^{(B)}$. Finally, we update the global variables by using the learned local variables (we use different step-sizes α_1, α_2 and α_3 for the updating of different global variables). The detailed implementation of the stochastic fcaGPLVM is summarised in Algorithm 1.

It is worth to note that the computation of the lower bound seems also to be decomposable in terms of the D dimensions. However, as we have mentioned in this section, the projection function $h^{(d)}(\boldsymbol{y}_n)$ in (16) depends on the complete input vector $\boldsymbol{y}_n$. As a result, the lower bound still depends on the complete input vector $\boldsymbol{y}_n$ and cannot be factorized in terms of the D dimensions. In fact, the introduced function $h^{(d)}(\boldsymbol{y}_n)$ is one of important contribution of this paper.

1. It is worth to note that many other projection functions (i.e., Multi-layer Perceptron, Kernel Ridge Regression and so on) can also be used without adding any complexity in formulations.

Algorithm 1: Stochastic fcaGPLVM

Input: $\boldsymbol{Y}$, Q, P, M, B, λ_1, α_1, α_2, α_3

Output: $\boldsymbol{X}$, $\{\boldsymbol{W}_d, \boldsymbol{b}_{(d)}\}_{d=1}^{D}$, $\boldsymbol{m}$, $\boldsymbol{S}$, $\boldsymbol{Z}$, $\bar{\boldsymbol{Z}}$ and $\boldsymbol{\theta}$

Train a conventional sparse GPLVM on a small subset of $\boldsymbol{Y}$ and use the learned GPLVM initialize $\boldsymbol{Z}$, $\boldsymbol{m}$, $\boldsymbol{S}$, $\boldsymbol{\theta}$;

Initialize $\{\boldsymbol{W}_d, \boldsymbol{b}_{(d)}\}_{d=1}^{D}$, $\bar{\boldsymbol{Z}}$ randomly;

repeat

  Draw B samples ($\boldsymbol{Y}^{(B)}$) from $\boldsymbol{Y}$ randomly;

  Maximize $\mathcal{LB}_r(\boldsymbol{Y}^{(B)})$ to learn the corresponding local variables with fixed global variables;

  Compute the gradients of $\mathcal{LB}_r(\boldsymbol{Y}^{(B)})$ with respect to all the global variables;

  for $d = 1, \cdots, D$ **do**

    $\left(\boldsymbol{W}_{(d)}, \boldsymbol{b}_{(d)}\right) += \alpha_1 \left(\frac{\partial \mathcal{LB}_r(\boldsymbol{Y}^{(B)})}{\partial \boldsymbol{W}_{(d)}}, \frac{\partial \mathcal{LB}_r(\boldsymbol{Y}^{(B)})}{\partial \boldsymbol{b}_{(d)}}\right)$

  Update the variational parameters and auxiliary points:

  $\boldsymbol{m} += \alpha_2 \frac{\partial \mathcal{LB}_r(\boldsymbol{Y}^{(B)})}{\partial \boldsymbol{m}}$, $\boldsymbol{S} += \alpha_2 \frac{\partial \mathcal{LB}_r(\boldsymbol{Y}^{(B)})}{\partial \boldsymbol{S}}$, $\boldsymbol{Z} += \alpha_2 \frac{\partial \mathcal{LB}_r(\boldsymbol{Y}^{(B)})}{\partial \boldsymbol{Z}}$, $\bar{\boldsymbol{Z}} += \alpha_2 \frac{\partial \mathcal{LB}_r(\boldsymbol{Y}^{(B)})}{\partial \bar{\boldsymbol{Z}}}$

  Update the the hyper-parameter $\boldsymbol{\theta}$:

  $\boldsymbol{\theta} += \alpha_3 \frac{\partial \mathcal{LB}_r(\boldsymbol{Y}^{(B)})}{\partial \boldsymbol{\theta}}$

until *the lower bound in* (15) *converges*;

3.3. Supervised Feature-correlation-aware GPLVM

Both fcaGPLVM and stochastic fcaGPLVM are unsupervised, i.e., they ignore the label (class) information of data (Urtasun and Darrell, 2007; Gao et al., 2011; Jiang et al., 2012). As a consequence, it is essential to improve the performance of fcaGPLVM by developing a supervised fcaGPLVM (S-fcaGPLVM). In this case, we assume that the classes of N training samples are denoted by $\boldsymbol{l} = [l_1, ..., l_N]^T \in \mathbb{R}^N$, where $l_n \in \{1, \cdots, L\}$ indicates the class of $\boldsymbol{y}_n$ and L denotes the number of classes. We also define the class centers of L classes as $\{\boldsymbol{c}_1, \cdots, \boldsymbol{c}_L\}$ and then propose the following joint objective function for supervised dimension reduction:

$$\mathcal{LB}_s(\boldsymbol{Y}) = \mathcal{LB}_r(\boldsymbol{Y}) - \lambda_2 \sum_{n=1}^{N}\sum_{l=1}^{L} \mathbb{1}(l_n = l)\|\boldsymbol{x}_n - \boldsymbol{c}_l\|_2^2 + \lambda_3 \sum_{a=1}^{L-1}\sum_{b=a+1}^{L} \|\boldsymbol{c}_a - \boldsymbol{c}_b\|_2^2 \tag{18}$$

where $\mathbb{1}(l_n = l) = 1$ if $l_n = l$ and 0 otherwise. Obviously, in this S-fcaGPLVM, we try to learn the fcaGPLVM and maximize the between-class scatter (as well as minimize within-class scatter) synchronously. λ_2 and λ_3 denote the positive regularization hyper-parameters that control the tradeoff between the ability of discrimination and generalization as in Urtasun and Darrell (2007). The optimization process of S-fcaGPLVM is the same as in Algorithm 1, except that we should also update the class centers in each iteration. In fact, we can also use the stochastic gradient method to update the l^{th} class center by utilizing the gradient of $\mathcal{LB}_s(\boldsymbol{Y}^{(B)})$ with respect to the class center:

$$\frac{\partial \mathcal{LB}_s(\boldsymbol{Y}^{(B)})}{\partial \boldsymbol{c}_l} = 2\lambda_2 \sum_{n=1}^{B} \mathbb{1}(l_n = l)(\boldsymbol{c}_l - \boldsymbol{x}_n) + 2\sum_{a=1}^{L} \mathbb{1}(a \neq l)(\boldsymbol{c}_l - \boldsymbol{c}_a)$$

where $\mathbb{1}(l_n \neq l) = 1$ if $l_n \neq l$ and 0 otherwise.

3.4. Prediction of new samples

In the prediction step, given a new sample $\boldsymbol{y}^*$, our goal is to predict the corresponding feature-description variables $\{\bar{\boldsymbol{x}}_d^*\}_{d=1}^D$ and the low dimensional latent variable $\boldsymbol{x}^*$. Since we have learned the projection functions $\{h^{(d)}\}_{d=1}^D$ in the learning step, the feature-description variable $\bar{\boldsymbol{x}}_d^*$ can be computed directly as

$$\bar{\boldsymbol{x}}_d^* = h^{(d)}(\boldsymbol{y}^*), \ \ for \ d = 1, \cdots, D \tag{19}$$

At last, we can maximize the following objective function to learn the corresponding low dimensional latent variable $\boldsymbol{x}^*$:

$$\mathcal{LB}(\boldsymbol{y}^*) = \sum_{d=1}^{D} \left(\ln \mathcal{N}\left(y_{nd} | \hat{\boldsymbol{k}}_d^* \hat{\boldsymbol{K}}_{\boldsymbol{ZZ}}^{-1} \boldsymbol{m}, \sigma^2\right) - \frac{1}{2\sigma^2} \tilde{k}_{d,d}^* - \frac{1}{2} tr\left(\boldsymbol{S}\boldsymbol{\Lambda}_d^*\right) \right) \tag{20}$$

where $\hat{\boldsymbol{k}}_d^* \in \mathbb{R}^{1\times M}$ and its m^{th} element $(\hat{k}_{dm}^*)$ denotes the kernel $\bar{k}(\bar{\boldsymbol{x}}_d^*, \bar{\boldsymbol{z}}_m) k(\boldsymbol{x}^*, \boldsymbol{z}_m)$; $\boldsymbol{\Lambda}_d^* = \frac{1}{\sigma^2} \hat{\boldsymbol{K}}_{\boldsymbol{ZZ}}^{-1} \hat{\boldsymbol{k}}_d^{*T} \hat{\boldsymbol{k}}_d^* \hat{\boldsymbol{K}}_{\boldsymbol{ZZ}}^{-1}$; $\tilde{k}_{d,d}^* = \bar{k}(\bar{\boldsymbol{x}}_d^*, \bar{\boldsymbol{x}}_d^*) k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \hat{\boldsymbol{k}}_d^* \hat{\boldsymbol{K}}_{\boldsymbol{ZZ}}^{-1} \hat{\boldsymbol{k}}_d^{*T}$.

4. Special Cases of FcaGPLVM

In this section, we introduce three special cases of fcaGPLVM and show the connections between fcaGPLVM and the previous GPLVMs. Firstly, since the fcaGPLVM is an extension of the original GPLVM, the following proposition shows that fcaGPLVM becomes the original GPLVM by ignoring the feature correlations.

Proposition 1 *Assume that matrix $\bar{\boldsymbol{K}}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}$ has the following form:*

$$\bar{\boldsymbol{K}}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} = \boldsymbol{I}_D \otimes \boldsymbol{E}_N \in \mathbb{R}^{ND\times ND} \tag{21}$$

where $\boldsymbol{E}_N \in \mathbb{R}^{N\times N}$ denotes the matrix with all elements being one, then $\hat{\boldsymbol{K}}$ can be written as

$$\hat{\boldsymbol{K}} = \boldsymbol{I}_D \otimes \boldsymbol{K}_{\boldsymbol{XX}} \in \mathbb{R}^{ND\times ND} \tag{22}$$

and fcaGPLVM becomes the original GPLVM.

The work in Lawrence (2005) gives a detailed discussion of the GPLVM with the kernel matrix in (22) and shows the equivalence between it and the original GPLVM.

Secondly, the fcaGPLVM also has a close connection to the Kronecker GP as shown in the following proposition:

Proposition 2 *Assume that matrix $\bar{\boldsymbol{K}}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}$ has the following form:*

$$\bar{\boldsymbol{K}}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} = \boldsymbol{\Sigma}_D \otimes \boldsymbol{E}_N \in \mathbb{R}^{ND\times ND} \tag{23}$$

where $\boldsymbol{\Sigma}_D \in \mathbb{R}^{D\times D}$ is a positive semi-definite matrix, then $\hat{\boldsymbol{K}}$ can be written as

$$\hat{\boldsymbol{K}} = \boldsymbol{\Sigma}_D \otimes \boldsymbol{K}_{\boldsymbol{XX}} \in \mathbb{R}^{ND\times ND} \tag{24}$$

and fcaGPLVM becomes the Kronecker GP (see Bonilla et al. (2008) for more detail).

Table 1: Detailed information of the data sets.

Data sets	# of samples	# of features	# of classes
FC	7466	11	\
USPS	2,000	256	10
Statlog	6,435	36	7
SDD	58,509	48	11
GFE	27,936	300	9

Obviously, in this case, we use the matrix $\boldsymbol{\Sigma}_D$ to encode the correlations among D features. A joint learning can be conducted to synchronously estimate $\boldsymbol{\Sigma}_D$ and the low dimensional latent variables.

Thirdly, by imposing some constraints on both $\boldsymbol{K}_{\boldsymbol{XX}}$ and the projection functions $\{h^{(d)}\}_{d=1}^{D}$, Proposition 3 shows that the GPLVM with back-constraints becomes a special case of fcaGPLVM.

Proposition 3 *Assume that matrix $\boldsymbol{K}_{\boldsymbol{XX}} = \boldsymbol{E}_N$, the projection functions $h^{(1)} = h^{(2)} = \cdots = h^{(D)} = h$ and $\bar{\boldsymbol{K}} = \boldsymbol{I}_D \otimes \bar{\boldsymbol{K}}_{\dot{\boldsymbol{X}}\dot{\boldsymbol{X}}}$ where $\bar{\boldsymbol{K}}_{\dot{\boldsymbol{X}}\dot{\boldsymbol{X}}}$ denotes the kernel matrix computed by using the kernel function $\bar{k}(\cdot,\cdot)$ on $\dot{\boldsymbol{X}} = [h(\boldsymbol{x}_1), \cdots, h(\boldsymbol{x}_N)]^T$, then $\hat{\boldsymbol{K}}$ can be written as*

$$\hat{\boldsymbol{K}} = \bar{\boldsymbol{K}} = \boldsymbol{I}_D \otimes \bar{\boldsymbol{K}}_{\dot{\boldsymbol{X}}\dot{\boldsymbol{X}}} \in \mathbb{R}^{ND \times ND} \tag{25}$$

Thus fcaGPLVM becomes the GPLVM with back-constraints (see Lawrence and Quiñonero-Candela (2006) for more detail).

5. Experiments and Analysis

In this section, we evaluate our fcaGPLVM and S-fcaGPLVM (by using stochastic gradient ascent algorithm) on multiple real-world data sets. The compared methods include unsupervised GPLVMs, i.e., original GPLVM (Lawrence, 2005), GPLVM with back-constraints (BC-GPLVM) (Lawrence and Quiñonero-Candela, 2006) in which we use a Multi-layer Perceptron (MLP) back-constraint with one hidden-layer, Kronecker GPLVM (the extension of Kronecker GP in Bonilla et al. (2008)); supervised GPLVMs, i.e., Discriminative GPLVM (D-GPLVM) (Urtasun and Darrell, 2007), Supervised GPLVM (S-GPLVM) (Gao et al., 2011). We also compare the proposed S-fcaGPLVM with the GP for classification (GPC) model (Hensman et al., 2015) to demonstrate its performance.

5.1. Data Sets and Settings

In the experiments, we use the flow cytometry (FC), United States Post Services (USPS) and another three UCI data sets, i.e., Landsat Satellite (Statlog), Sensorless Drive Diagnosis (SDD), grammatical facial expression (GFE), to verify the performance of the proposed models. Table 1 shows the detailed information of these data sets. All the experiments are run on computer with Intel(R) Core(TM) i5-3470 @ 3.20GHz CPU and 16.0 GB RAM.

In the experiments, we randomly draw 10% samples as test set and use the remaining samples for 5-fold cross-validation to obtain the best hyper-parameters. To validate the performance of the compared methods, we use the nearest neighborhood (NN) classifier (with nearest neighborhood number $K = 3$) on the learned latent variables to obtain the

classification accuracy. We run the experiments on randomly split data set 10 times and obtain the average classification accuracy. It is worth to note that, for the fcaGPLVM and S-fcaGPLVM, we only use the low dimensional latent variables (excluding the feature-description variables) for classification, thus providing a fair comparison. Since the GPC is a classification method, we directly use it on the split data sets and then report its averaged accuracies. The fcaGPLVM, S-fcaGPLVM and GPC are all trained in the mini-batch mode with $B = 20$. For all the models, we use the RBF kernel function and the number of auxiliary points is set to 100. The dimension of auxiliary points is the same as that of low dimensional latent variables. The dimension of feature-description variables in both fcaGPLVM and S-fcaGPLVM is set to $P = 10$.

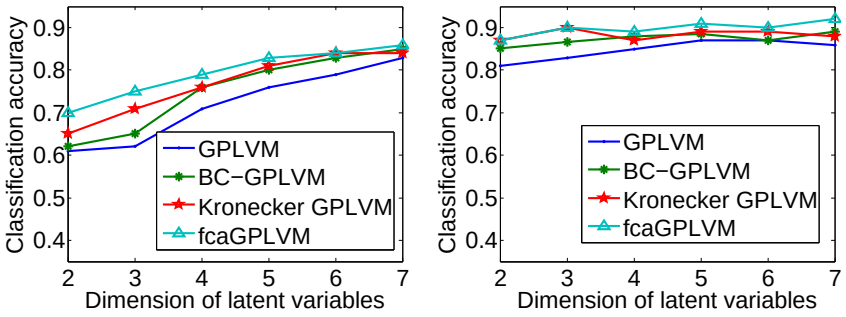
5.2. Unsupervised Dimension Reduction

In this section, we compare fcaGPLVM with other models in unsupervised dimension reduction. The classification accuracies with different dimensions (from 2 to 7) of latent variables are shown in Figures 2(a)-2(d). As we can see, BC-GPLVM outperforms GPLVM by preserving the local distance among samples. Both Kronecker GPLVM and fcaGPLVM also outperform GPLVM, demonstrating that we can improve the performance of GPLVM by considering the feature correlations. In the experiments, to the best of our knowledge, it is the first attempt to use the Kronecker GP in GPLVM. Furthermore, in Section 4, we have demonstrated that it becomes the special case of our fcaGPLVM with certain settings. In most cases, fcaGPLVM has the highest classification accuracy, indicating its significant improvements in performance. For the FC data, we report the feature correlation matrices learned by fcaGPLVM. Specifically, after obtaining feature-description variables for each sample, we compute the feature kernel matrix of each sample using the kernel function $\bar{k}(\cdot, \cdot)$. At last, we show the inversions[2] of these matrices for which we only show those elements with large absolute values (≥ 0.15 in Figure 3(b) and ≥ 0.2 in Figure 3(c)). Figure 3(a) shows the directed acyclic graph (DAG) learned by Sachs et al. (2005). As we can see, Figures 3(b) and 3(c) almost coincide with the DAG in Figure 3(a), indicating that fcaGPLVM takes good effect on the learning of feature correlations.

5.3. Supervised Dimension Reduction

In this subsection, we compare S-fcaGPLVM with supervised learning models to verify its performance in classification. The dimension of latent variables is also set from 2 to 7. The results are shown in Figures 4(a)-4(d). As we can see, with the growth of the number of latent variable dimensions, the performance of all the GPLVM-based models including S-fcaGPLVM, S-GPLVM and D-GPLVM increases and becomes stable at last. In most cases, S-fcaGPLVM outperforms all other models, demonstrating that S-fcaGPLVM can indeed improve its performance by modeling feature-correlations. Furthermore, all the latent variable models outperform the GPC model, indicating that these latent variable models can learn more representative features to improve the performance of supervised tasks. We also calculate the p-values of the non-parametric Wilcoxon rank-sum test to

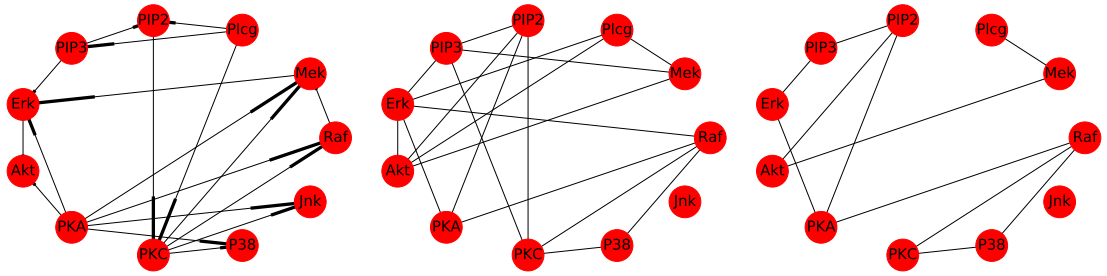
2. In fact, it is the inversion matrices (of feature-description kernel matrices) that imply the feature correlations among the observed variables as in Hsieh et al. (2011). We therefore show these inversion matrices to illustrate the learned feature-correlations.



(a) USPS (b) Statlog

(c) SDD (d) GFE

Figure 2: Performance of classification in unsupervised dimension reduction.



(a) DAG learned by Sachs et al. (2005) (b) ≥ 0.15 (c) ≥ 0.2

Figure 3: The feature correlations learned by Sachs et al. (2005) and our fcaGPLVM.

check the statistical significance. Although the p-values on different data sets are not provided in this paper, in most cases they are all less than 0.05, indicating the statistical significance between fcaGPLVM with other compared methods. We can also see that the performance of fcaGPLVM on USPS and SDD data sets is not significantly improved with the increase of dimension of latent variables, indicating that fcaGPLVM can also obtain better classification accuracy with lower latent variable dimension.

5.4. Analyses of hyper-parameters

In this subsection, we analyze the effect of both the dimension of feature-description variables and the number of auxiliary points on the performance of fcaGPLVM. Specifically,

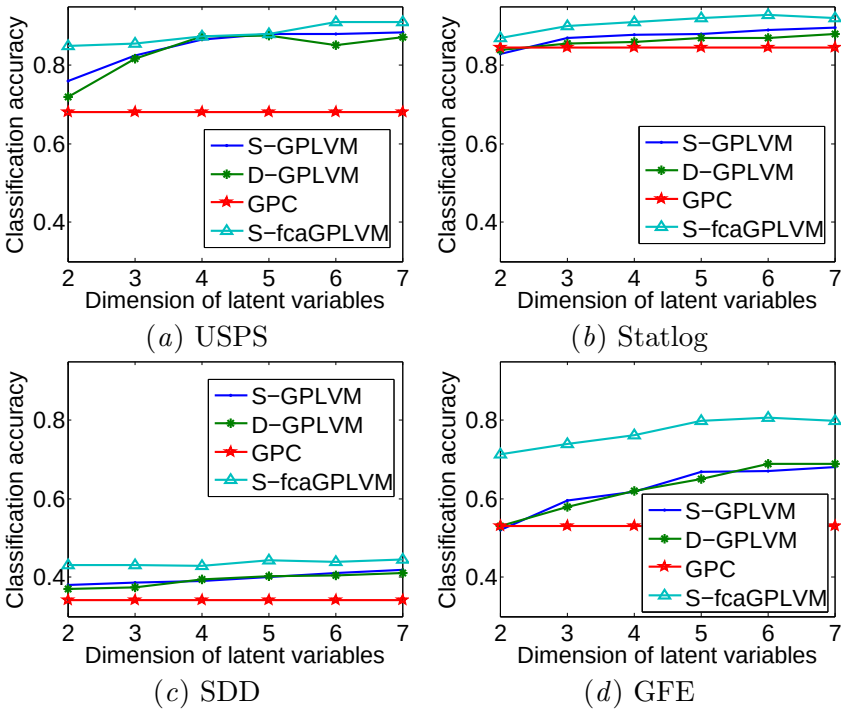


Figure 4: Performance of classification in supervised dimension reduction.

we select the dimension P of feature-description variables and the number M of auxiliary points from $\{0, 5, 10, 15\}$ and $\{10, 20, ..., 90, 100\}$ respectively. It is worth noting that when $P = 0$, fcaGPLVM becomes the original GPLVM. We fix the dimension of latent variable, $Q = 7$, and conduct experiments on USPS and Statlog data sets by using 5-fold cross-validation. Their classification accuracies are shown in Figure 5. As we can see, since the number of auxiliary points controls the approximate ability of the stochastic fcaGPLVM towards the full fcaGPLVM, more auxiliary points lead to higher classification accuracy. The dimension of feature-description variables controls the complexity of fcaGPLVM, thus a cross-validation should be conducted to choose the appropriate value of P for specific data, e.g., for the USPS and Statlog, their reasonable values of P are 10.

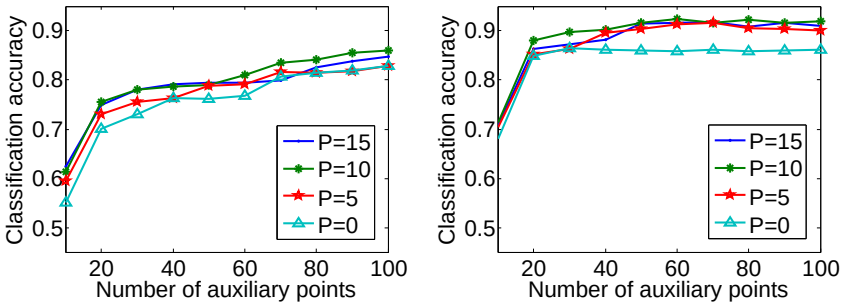


Figure 5: Classification accuracy of fcaGPLVM with different P and M.

Table 2: Comparison of the computation and storage complexities in each iteration.

	GPLVM	BC-GPLVM	Kronecker GPLVM	S-GPLVM	D-GPLVM	GPC	fcaGPLVM	S-fcaGPLVM
CC	$\mathcal{O}(M^2N)$	$\mathcal{O}(M^2N)$	$\mathcal{O}(M^2(DB+M))$	$\mathcal{O}(M^2N)$	$\mathcal{O}(M^2N)$	$\mathcal{O}(M^2(B+M))$	$\mathcal{O}(M^2(DB+M))$	$\mathcal{O}(M^2(DB+M))$
SC	$\mathcal{O}(MN)$	$\mathcal{O}(MN)$	$\mathcal{O}(M(DB+M))$	$\mathcal{O}(MN)$	$\mathcal{O}(MN)$	$\mathcal{O}(M(B+M))$	$\mathcal{O}(M(DB+M))$	$\mathcal{O}(M(DB+M))$

Table 3: Precise comparison of the times consumed by the eight models.

Data sets	GPLVM	BC-GPLVM	Kronecker GPLVM	S-GPLVM	D-GPLVM	GPC	fcaGPLVM	S-fcaGPLVM
USPS	331.2s	923.8s	17,521.4s	1,169.6s	42,279.5s	2,700.2s	19,388.1s	22,963.4s
Statlog	1,633.9s	7,156.1s	16,668.4s	2,676.5s	96,347.3s	10,158.9s	20,391.9s	25,922.3s
SDD	14,865.4s	73,439.3s	173,172.2s	103,076.2s	523,471.3s	74,173.2s	216,016.7s	284,771.5s
GFE	8,981.7s	42,463.7s	96.399.8s	33,704.6s	382,321.7s	51,454.3s	135,152.6s	171,912.1s

5.5. Complexity Analysis

We theoretically compare the complexities of all the models and report the computation and storage complexities in each iteration in Table 2 (CC: Computation Complexity; SC: Storage Complexity). As we can see, both the computation and storage complexities of Kronecker GPLVM, GPC and fcaGPLVM are much lower than other models (it is worth to note that we use variational inference based sparse approximation (Titsias, 2009) for the optimization of GPLVM, BC-GPLVM, S-GPLVM and D-GPLVM. Thus both their computation and storage complexities are much lower than their original settings). However, this comparison just makes a rough estimation since the number of iterations is ignored. We therefore also give an accurate time comparison of these models on the four data sets (with $Q = 2$) in Table 3. Although the running times of fcaGPLVM and S-fcaGPLVM are not the lowest, they have lower storage complexities and achieve much better classification accuracies. In fact, our model can deal with much bigger datasets than the batch learning models such as BC-GPLVM, D-GPLVM and S-GPLVM, since its dominant computation and storage complexities (in each iteration) only depend on the dimension of observed variables and the size of mini-batch. However, the batch learning models such as GPLVM, BC-GPLVM, D-GPLVM and S-GPLVM usually run out of memory when dealing with much bigger datasets, since they need to compute and store the kernel matrix between every training samples and every auxiliary points.

6. Conclusion

In this paper, we have presented the feature-correlation-aware GPLVM and used the stochastic variational inference method to speed up the learning of latent variables and hyper-parameters. Furthermore, we have also proposed the Supervised fcaGPLVM that utilizes the label information to improve the performance of dimension reduction. The performance of fcaGPLVM and S-fcaGPLVM has been shown from the experiments on multiple real-world data sets. Although it is not discussed in detail, fcaGPLVM can also be used in many other learning scenarios such as semi-supervised and multi-view learnings which are the main contents of our future work.

References

Erik Bodin, Neill DF Campbell, and Carl Henrik Ek. Latent gaussian process regression. *arXiv preprint arXiv:1707.05534*, 2017.

Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *NIPS*, pages 153–160, 2008.

Deng Cai, Xiaofei He, Yuxiao Hu, Jiawei Han, and Thomas Huang. Learning a spatially smooth subspace for face recognition. In *CVPR*, pages 1–7, 2007.

Zhenwen Dai, Mauricio A Álvarez, and Neil D Lawrence. Efficient modeling of latent information in supervised learning using gaussian processes. *arXiv preprint arXiv:1705.09862*, 2017.

Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. Shared gaussian process latent variable model for multi-view facial expression recognition. In *ISVC*, pages 527–538, 2013.

Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. Discriminative shared gaussian processes for multiview and view-invariant facial expression recognition. *IEEE transactions on image processing*, 24(1):189–204, 2015.

Xinbo Gao, Xiumei Wang, Xuelong Li, and Dacheng Tao. Transfer latent variable model based on divergence analysis. *Pattern Recognition*, 44(10):2358–2366, 2010.

Xinbo Gao, Xiumei Wang, Dacheng Tao, and Xuelong Li. Supervised gaussian process latent variable model for dimensionality reduction. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(2):425–434, 2011.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. In *UAI*, pages 282–290, 2013.

James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification. In *AISTATS*, pages 351–360, 2015.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Cho-Jui Hsieh, Inderjit S Dhillon, Pradeep K Ravikumar, and Mátyás A Sustik. Sparse inverse covariance matrix estimation using quadratic approximation. In *NIPS*, pages 2330–2338, 2011.

Wen-bing Huang, Deli Zhao, Fuchun Sun, Huaping Liu, and Edward Y Chang. Scalable gaussian process regression using deep neural networks. In *IJCAI*, pages 3576–3582, 2015a.

Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Projection metric learning on grassmann manifold with application to video based face recognition. In *CVPR*, pages 140–149, 2015b.

Tomoharu Iwata and Zoubin Ghahramani. Improving output uncertainty estimation and generalization in deep learning via neural network gaussian processes. *arXiv preprint arXiv:1707.05922*, 2017.

Pratik Jawanpuria, Maksim Lapin, Matthias Hein, and Bernt Schiele. Efficient output kernel learning for multiple tasks. In *NIPS*, pages 1189–1197, 2015.

Xinwei Jiang, Junbin Gao, Tianjiang Wang, and Lihong Zheng. Supervised latent linear gaussian process latent variable model for dimensionality reduction. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(6):1620–1632, 2012.

Neil Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of machine learning research*, 6(Nov):1783–1816, 2005.

Neil D Lawrence and Joaquin Quiñonero-Candela. Local distance preservation in the gp-lvm through back constraints. In *ICML*, pages 513–520, 2006.

Miguel Lázaro-Gredilla and Michalis K Titsias. Variational heteroscedastic gaussian process regression. In *ICML*, pages 841–848, 2011.

Chaochao Lu and Xiaoou Tang. Surpassing human-level face verification performance on lfw with gaussianface. In *AAAI*, pages 3811–3819, 2015.

K Sachs, O Perez, D Pe'Er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Guoli Song, Shuhui Wang, Qingming Huang, and Qi Tian. Multimodal gaussian process latent variable models with harmonization. In *CVPR*, pages 5029–5037, 2017.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *AISTATS*, pages 567–574, 2009.

Raquel Urtasun and Trevor Darrell. Discriminative gaussian process latent variable model for classification. In *ICML*, pages 927–934, 2007.

Xiumei Wang, Xinbo Gao, Yuan Yuan, Dacheng Tao, and Jie Li. Semi-supervised gaussian process latent variable model with pairwise constraints. *Neurocomputing*, 73(10C12): 2186–2195, 2010.

Yali Wang, Marcus A. Brubaker, Brahim Chaib-draa, and Raquel Urtasun. Bayesian filtering with online gaussian process latent variable models. In *UAI*, pages 849–857, 2014.

Hao Xiong and Dacheng Tao. A diversified generative latent variable model for wifi-slam. In *AAAI*, pages 3841–3847, 2017.

Nuo Xu, Kian Hsiang Low, Jie Chen, and Keng Kiat Lim. Gp-localize: persistent mobile robot localization using online sparse gaussian process observation model. In *AAAI*, pages 2585–2592, 2014.