Multi-view Deep Gaussian Process with a Pre-training Acceleration Technique

Han Zhu, Jing Zhao(✉), and Shiliang Sun

School of Computer Science and Technology, East China Normal University, 3663 North Zhongshan Road, Shanghai 200062, People's Republic of China
zhuhanchn@gmail.com, {jzhao,slsun}@cs.ecnu.edu.cn

Abstract. Deep Gaussian process (DGP) is one of the popular probabilistic modeling methods, which is powerful and widely used for function approximation and uncertainty estimation. However, the traditional DGP lacks consideration for multi-view cases in which data may come from different sources or be constructed by different types of features. In this paper, we propose a generalized multi-view DGP (MvDGP) to capture the characteristics of different views and model data in different views discriminately. In order to make the proposed model more efficient in training, we introduce a pre-training network in MvDGP and incorporate stochastic variational inference for fine-tuning. Experimental results on real-world data sets demonstrate that pre-trained MvDGP outperforms the state-of-the-art DGP models and deep neural networks, achieving higher computational efficiency than other DGP models.

Keywords: Deep Gaussian process · Multi-view learning · Variational inference · Stochastic optimization · Pre-training technique

1 Introduction

Gaussian process (GP) owns a significant ability of modeling representation and can estimate the uncertainty of the prediction effectively [5,11,16]. Deep Gaussian process (DGP) is a stack of multi-layer GPs [1,2,13]. Benefitting from the hierarchical structure, DGP not only retains the excellent features of GP, but also overcomes the limitations of GP and obtains stronger mapping capability. However, the difficulty in DGP is mainly located on intractable calculations during the training process. The Bayesian training framework based on variational inference for DGP is a classical method but limited by the scale of data [2]. Doubly stochastic variational inference is a state-of-the-art and widely used inference technique, which adopts stochastic optimization and makes it possible for DGP to be applied to large-scale data [13]. Recently, there are some new works focusing on non-Gaussian posterior in the real-world data to develop DGP comprehensively [3,14].

Traditional GP models only focus on modeling data from a single source. As amounts and sources of data are augmented, data integrations from multiple feature sets are referred to as multi-view data [17,20]. It is improper to treat data

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12085, pp. 299–311, 2020.
https://doi.org/10.1007/978-3-030-47436-2_23

from different views equally, and thus multi-view learning flourishes. GP-based models have been extended to multi-view scenarios, in which the multi-view regularized GP [8] and the sparse multimodal GP [9] are the generalizations of shallow GP model. A DGP-based work is also developed [18], but limited in multi-view unsupervised representation learning, where additional classifiers are needed for classification tasks. Besides, the inference of the unsupervised DGP [18] is based on the Bayesian training framework with strong mean-field and Gaussian assumptions, which underestimates variance and makes the model unable to be applied to large-scale data scenarios. Our goal is to propose a general end-to-end multi-view DGP (MvDGP). We build a scalable model without forcing independence between layers, and apply stochastic variational inference and re-parameterization techniques to improve the ability of modeling on the large-scale data.

In addition, we expect that the MvDGP model possesses significant superiority in training speed. In the multi-view scenario, we tune the model according to the characteristics of each view, which will inevitably introduce more model parameters and lengthen the training time. Pre-training is a widely used technique [4,19], in which a large number of data are taken as training samples to be trained across multiple GPUs. The weights obtained by pre-trained networks are used as the initial weights for new tasks, and then only a few steps of fine-tuning are needed to get prediction results. In order to make the proposed model more competitive in terms of training speed, we introduce a novel pre-training model for MvDGP. Instead of training with the same model using other data sets, we use the same data set to train with other models. Because the neural network with infinite width has been proved equivalent to GP exactly and the training cost of deep neural network (DNN) is much less than DGP [6,7,10], we pre-train the DNN with a similar structure of MvDGP to analogize the initial training process of MvDGP. Through the DNN pre-training, we aim to get a set of appropriate initial parameters for MvDGP. Since the parameter domains of the DNN and the MvDGP are not the same, the initial parameters of each layer in the MvDGP are obtained by auxiliary optimization of single GP. The optimization efficiency of MvDGP is improved significantly by pre-training.

There are three main contributions in our work:

1. **Generalized Multi-view Deep Gaussian Process (MvDGP)**: We propose a generalized and flexible MvDGP, which considers characteristics of different views. Deep structure leads to more powerful abilities of uncertainty estimation and mapping representation compared with shallow models [8,9]. Furthermore, MvDGP is an end-to-end supervised model, which can take advantage of labels to learn models, and provides stronger robustness and generalization performance than unsupervised multi-view DGP [18].
2. **Scalability**: We infer the MvDGP without setting strong mean-field constraints and derive stochastic variational inference. Compared to the model [18] can hardly be applied in large-scale scenarios, our model is capable of it. Meanwhile, our model can be extended to more views easily and can customize the detailed depth of each view according to the view characteristic.

3. **Efficiency**: We obtain appropriate initial parameters by DNN pre-training for MvDGP, which reduces the oscillation and speeds up the training. Experiments demonstrate that the pre-trained MvDGP guarantees higher performance and runs several times faster than unpre-trained methods.

2 Deep Gaussian Process

Deep Gaussian process (DGP) is a stack of multiple GPs, which possesses a more powerful modeling capability than a GP [2]. For a standard DGP, we review a supervised version as an example. Given a training set, including observed inputs $\mathbf{X} \in \mathcal{R}^{N \times Q}$ and observed outputs $\mathbf{Y} \in \mathcal{R}^{N \times D}$, where N is the number of samples, Q and D are the dimensionality of input and output vector, respectively.

For a DGP with L layers of hidden units, we define $\mathbf{F} = \{F_1, F_2, ..., F_L\}$ as the latent variable set, where F_l is the output for layer l and the input for layer $l+1$, $l = 1, \ldots, L-1$. Furthermore, we add additional sets of inducing inputs $\mathbf{Z} = \{Z_1, Z_2, ..., Z_L\}$ and inducing points $\mathbf{U} = \{U_1, U_2, ..., U_L\}$ to employ variational inference [15]. The assumption of the model prior is as follows,

$$p(\mathbf{U}|\mathbf{Z}) = \mathcal{N}(\mathbf{U}|m(\mathbf{Z}), k(\mathbf{Z}, \mathbf{Z})), \tag{1}$$

where $m(\mathbf{Z})$ is the mean function and $k(\mathbf{Z}, \mathbf{Z})$ is the kernel function. Note that $[k(\mathbf{Z}, \mathbf{Z})]_{ij} = k(\mathbf{Z}_i, \mathbf{Z}_j)$, where $i, j = 1, \ldots, N$. We record $\mathbf{X}$ as F_0, and the conditional distribution, corresponding mean and variance are denoted as follows,

$$p(F_l|F_{l-1}, U_l) = \mathcal{N}(F_l|\mu_l, \Sigma_l), \quad l = 1, \ldots, L \tag{2}$$

$$\mu_l = m(F_{l-1}) + k(F_{l-1}, Z_l)k(Z_l, Z_l)^{-1}(U_l - m(Z_l)), \tag{3}$$

$$\Sigma_l = k(F_{l-1}, F_{l-1}) - k(F_{l-1}, Z_l)k(Z_l, Z_l)^{-1}k(Z_l, F_{l-1}). \tag{4}$$

The likelihood of model is generally set to a Gaussian distribution,

$$p(\mathbf{Y}|F_L) = \mathcal{N}(F_L, \Sigma_L + \Sigma_{\mathbf{Y}}), \tag{5}$$

where $\Sigma_{\mathbf{Y}}$ is the variance of the observation $\mathbf{Y}$. The joint density of the observed output $\mathbf{Y}$, latent variables $\mathbf{F}$ and inducing points $\mathbf{U}$ is written as

$$p(\mathbf{Y}, \mathbf{F}, \mathbf{U}) = p(\mathbf{Y}|F_L) \prod_{l=1}^{L} p(F_l|F_{l-1}, U_l)p(U_l|Z_l). \tag{6}$$

3 Multi-view Deep Gaussian Process

Due to the characteristics of multi-view data, the general DGP cannot utilize the rich information in multiple views reasonably. In this section, we propose a new model named multi-view deep Gaussian process (MvDGP), and introduce stochastic variational inference for optimization.

3.1 Multi-view Deep Gaussian Process

We propose an end-to-end multi-view model and take two views of data and models as an example. For given data $\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{Y}\}$, $\mathbf{X}^{(1)} \in \mathcal{R}^{N\times Q_1}$ and $\mathbf{X}^{(2)} \in \mathcal{R}^{N\times Q_2}$ are observed inputs of the first and the second view respectively and $\mathbf{Y} \in \mathcal{R}^{N\times D}$ is the observed outputs. For data of each view, there is a deep structure to model it. The latent variables of intermediate layers are recorded as $F_l^{(v)}$, $v = \{1, 2\}$, $l = 1, \ldots, H^{(v)}$, where v is the index of view and $H^{(v)}$ is the depth of view v. The depths of the networks in different views can be determined according to the data characteristics of each view for better mapping. The inducing inputs $Z_l^{(v)}$ and the inducing points $U_l^{(v)}$ are introduced for each latent variable $F_l^{(v)}$ as in Sect. 2. In addition to the separated GP layers for each view, there are also common layers that share information for both views, in which variables and model parameters are denoted as $F_l^{(S)}$, $Z_l^{(S)}$, $U_l^{(S)}$, $l = 1, \ldots, H^{(S)}$. The graphical model of MvDGP is illustrated in Fig. 1, and the depth for each view is marked as $H^{(1)} = L, H^{(2)} = R, H^{(S)} = H$.

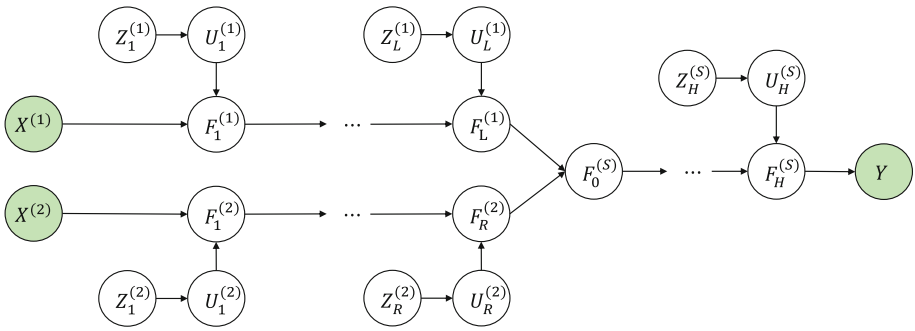

Fig. 1. The graphical model for multi-view deep Gaussian process.

We record $F_0^{(S)}$ as the transition layer from the separated views $\mathbf{F}^{(1)}, \mathbf{F}^{(2)}$ to merged view $\mathbf{F}^{(S)}$, and the joint density of MvDGP is written as

$$\begin{aligned} p(\mathbf{Y}, \mathbf{F}, \mathbf{U}) =& p(\mathbf{Y}|F_H^{(S)}) p(\mathbf{F}^{(S)}, \mathbf{U}^{(S)}) p(F_0^{(S)}|F_L^{(1)}, F_R^{(2)}) \\ & p(\mathbf{F}^{(1)}, \mathbf{U}^{(1)}) p(\mathbf{F}^{(2)}, \mathbf{U}^{(2)}), \end{aligned} \tag{7}$$

where $p(F_0^{(S)}|F_L^{(1)}, F_R^{(2)}) = \mathcal{N}(F_0^{(S)}|[F_L^{(1)}, F_R^{(2)}], \Sigma_0^{(S)})$, $[F_L^{(1)}, F_R^{(2)}]$ is the concatenation of the last layers of two views, and $\Sigma_0^{(S)}$ represents corresponding unit variance. The joint distribution of latent variables in view v is specifically as

$$p(\mathbf{F}^{(v)}, \mathbf{U}^{(v)}) = \prod_{l=1}^{H^{(v)}} p(F_l^{(v)}|F_{l-1}^{(v)}, U_l^{(v)}) p(U_l^{(v)}|Z_l^{(v)}). \tag{8}$$

The depth for each view is $H^{(1)} = L, H^{(2)} = R, H^{(S)} = H$ and the symbols of $F_0^{(1)}$, $F_0^{(2)}$ denote the observed inputs $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, respectively.

3.2 Variational Inference

Directly inferring MvDGP is intractable and complex computationally, we take stochastic variational inference for optimization. The main idea of variational inference is to find an approximate posterior distribution $q(\mathbf{F}, \mathbf{U})$ that is as close as possible to the true posterior $p(\mathbf{F}, \mathbf{U})$.

We adopt a factorized form for joint posterior distribution as

$$\begin{aligned} q(\mathbf{F}, \mathbf{U}) &= q(\{F_l^{(1)}, U_l^{(1)}\}_{l=1}^{L}, \{F_l^{(2)}, U_l^{(2)}\}_{l=1}^{R}, \{F_l^{(S)}, U_l^{(S)}\}_{l=1}^{H}) \\ &= p(F_0^{(S)} | F_L^{(1)}, F_R^{(2)}) q(\mathbf{F}^{(S)}, \mathbf{U}^{(S)}) q(\mathbf{F}^{(1)}, \mathbf{U}^{(1)}) q(\mathbf{F}^{(2)}, \mathbf{U}^{(2)}), \end{aligned} \tag{9}$$

where $q(\mathbf{F}^{(v)}, \mathbf{U}^{(v)})$ is the variational distribution of view v, $v = 1, 2, S$, and $q(\mathbf{F}^{(v)}, \mathbf{U}^{(v)}) = \prod_{l=1}^{H^{(v)}} p(F_l^{(v)} | F_{l-1}^{(v)}, U_l^{(v)}) q(U_l^{(v)})$. The depth $H^{(v)}$ and $F_0^{(v)}$ for each view are denoted as Sect. 3.1. We take Gaussian forms for variational distribution of $\mathbf{U}$ as $q(U_l^{(v)}) = \mathcal{N}(U_l^{(v)} | m_l^{(v)}, S_l^{(v)})$, where layer $l = 1, \ldots, H^{(v)}$, view $v = 1, 2, S$, and $m_l^{(v)}$, $S_l^{(v)}$ are mean and variance of $q(U_l^{(v)})$, respectively. Under this setting, the variational posterior can be obtained analytically as

$$q(F_l^{(v)} | F_{l-1}^{(v)}, U_l^{(v)}) = \int p(F_l^{(v)} | F_{l-1}^{(v)}, U_l^{(v)}) q(U_l^{(v)}) dU_l^{(v)} = \mathcal{N}(F_l^{(v)} | \tilde{\mu_l}^{(v)}, \tilde{\Sigma_l}^{(v)}),$$

$$\tilde{\mu_l}^{(v)} = m(F_{l-1}^{(v)}) + k(F_{l-1}^{(v)}, Z_l^{(v)}) k(Z_l^{(v)}, Z_l^{(v)})^{-1} (m_l^{(v)} - m(Z_l^{(v)})), \tag{10}$$

$$\begin{aligned} \tilde{\Sigma_l}^{(v)} =& k(F_{l-1}^{(v)}, F_{l-1}^{(v)}) - k(F_{l-1}^{(v)}, Z_l^{(v)}) k(Z_l^{(v)}, Z_l^{(v)})^{-1} \\ & (k(Z_l^{(v)}, Z_l^{(v)}) - S_l^{(v)}) k(Z_l^{(v)}, Z_l^{(v)})^{-1} k(Z_l^{(v)}, F_{l-1}^{(v)}). \end{aligned} \tag{11}$$

In order to maintain gradients and update layer-wise parameters in the process of optimization, we introduce the re-parameterization trick and choose Monte Carlo method to estimate variational posterior $q(\mathbf{F})$ [12]. Firstly, draw a noise term $\epsilon_l^{(v)}$ from a standard Gaussian distribution, for view $v = 1, 2, S$ and layer $l = 1, \ldots, H^{(v)} - 1$. Then, iteratively sample latent variable $\hat{F}_l^{(v)} \sim q(F_l^{(v)} | \hat{F}_{l-1}^{(v)}, U_l^{(v)})$, in which $\hat{F}_l^{(v)}$ can be clearly written as

$$\hat{F}_l^{(v)} = \mu_l(\hat{F}_{l-1}^{(v)}) + \epsilon_l^{(v)} \sqrt{\Sigma_l(\hat{F}_{l-1}^{(v)}, \hat{F}_{l-1}^{(v)})}, \tag{12}$$

where μ_l and Σ_l are mean and covariance functions denoted in (10), (11).

3.3 Stochastic Optimization and Predictions

To minimize the KL divergence of q and p, we maximize the lower bound $\mathcal{L}$ of the logarithm marginal likelihood $\log p(\mathbf{Y})$, which is formulated as

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{F}, \mathbf{U})} \left[\log \frac{p(\mathbf{Y}, \mathbf{F}, \mathbf{U})}{q(\mathbf{F}, \mathbf{U})} \right]. \tag{13}$$

By substituting the joint density (7) and posterior distribution (9) to lower bound expression (13), the term $\prod_{l=1}^{H^{(v)}} p(F_l^{(v)} | F_{l-1}^{(v)}, U_l^{(v)})$ in the numerator and

denominator can be offset. The variational lower bound of model evidence in MvDGP can be rearranged to

$$\begin{aligned}\mathcal{L} =&\mathbb{E}_{q(\mathbf{F},\mathbf{U})}[\log p(\mathbf{Y}|F_H^{(S)})p(F_0^{(S)}|F_L^{(1)},F_R^{(2)}) \prod_{v}^{v\in\{1,2,S\}}\prod_{l=1}^{H^{(v)}}\frac{p(U_l^{(v)}|Z_l^{(v)})}{q(U_l^{(v)})}]\\ =&\mathbb{E}_{q(F_H^{(S)})}[\log p(\mathbf{Y}|F_H^{(S)})p(F_0^{(S)}|F_L^{(1)},F_R^{(2)})] - \sum_{v}^{v\in\{1,2,S\}}\mathrm{KL}^{(v)}, \end{aligned} \tag{14}$$

where $\mathrm{KL}^{(v)}$ represents $\sum_{l=1}^{H^{(v)}}\mathrm{KL}[q(U_l^{(v)})\|p(U_l^{(v)}|Z_l^{(v)})]$, $v=1,2,S$.

The expection about $q(F_H^{(S)})$ in the variational lower bound can be written in the form of additions for samples as follows,

$$\mathcal{L}=\sum_{i=1}^{N}\mathbb{E}_{q(F_{Hi}^{(S)})}[\log p(\mathbf{y}_i|F_{Hi}^{(S)})p(F_{0i}^{(S)}|F_{Li}^{(1)},F_{Ri}^{(2)})] - \sum_{v}^{v\in\{1,2,S\}}\mathrm{KL}^{(v)}, \tag{15}$$

where $\mathbf{y}_i$ is observed outputs, and $F_{Hi}^{(S)}, F_{0i}^{(S)}, F_{Li}^{(1)}, F_{Ri}^{(2)}$ are corresponding latent variables for sample i, $i=1,\ldots,N$. The addition expression of lower bound allows stochastic optimization to be employed in inference. The samples of minibatch can be regarded as an unbiased estimation of all samples.

Model parameters are optimized with the Adam optimizer during training, which include inducing inputs $\{Z_l^{(v)}\}_{l=1}^{H^{(v)}}$, variational parameters $\{m_l^{(v)}, S_l^{(v)}\}_{l=1}^{H^{(v)}}$ of inducing points $\{U_l^{(v)}\}_{l=1}^{H^{(v)}}$, and kernel parameters $\{\theta_l^{(v)}\}_{l=1}^{H^{(v)}}$, $v=1,2,S$. Stochastic optimization and unbiased minibatch samples ensure the scalability of MvDGP. Our model can be easily generalized to large-scale data.

For predictions, we take the mean of multiple samples of $F_H^{(S)*}$ as the predict outputs $\mathbf{Y}^*$ for test inputs $\mathbf{X}^*=\{\mathbf{X}^{(1)*},\mathbf{X}^{(2)*}\}$, and $q(F_H^{(S)*})$ is distributed as $q(F_H^{(S)*})\approx\frac{1}{K}\sum_{k=1}^{K}q(F_H^{(S)*}|\hat{F}_{H-1}^{(S)*},U_H)$, where K is the number of samples, and the value of $\hat{F}_0^{(v)}$ is set as $\mathbf{X}^{(v)*}, v=1,2$. The samples can be obtained according to the re-parameterization Monte Carlo sample steps (12) iteratively.

If there are more than two views in data, the MvDGP is easily to be generalized to multiple views by adding separated multi-layer GPs structure for new views.

4 Pre-training Technique for MvDGP

In order to better model the function approximation of each view, MvDGP introduces more latent variables and model parameters than single-view DGP. The training time of the model with a large number of parameters is not optimistic even with doubly stochastic optimization. Due to the initial parameters of the model have a significant impact on the training efficiency, the training speed of the model with proper initial parameters is faster than the random one. We consider introducing a novel technique of pre-training to MvDGP by training a

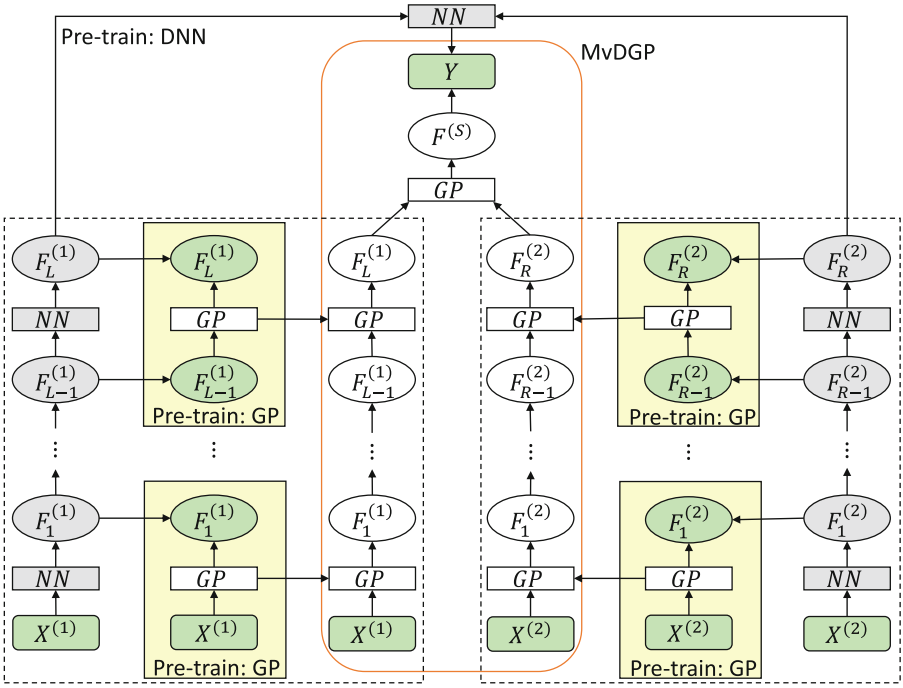

Fig. 2. The schematic diagram of pre-trained MvDGP.

computational cost-dominant model and getting a suitable set of initial parameters for MvDGP.

Deep neural network (DNN) is a type of powerful model for representation learning and model mapping. Inspired by the similar characteristics of DNN and DGP [7,10], we adopt the DNN with a similar structure to MvDGP to simulate the initial training process of MvDGP. We model the DNN separately for two views and build common network layers whose inputs are the concatenation of the outputs of the separated networks. The number of parameters is related to the number and dimension of hidden units. The number of model parameters in the DNN we used is much smaller than MvDGP, which leads to faster training speed.

Since it is not possible to directly use the parameters such as the network weights of the DNN in MvDGP, we use some single-layer GPs as auxiliary pre-training models. We take the values of the adjacent two layers in the DNN as the input and output of the single GP to obtain a set of initial parameters suitable for corresponding layers in MvDGP. Since the training difficulties of DNN and single GP are much lower than that of MvDGP, the pre-training step can be quickly calculated and is reasonable for roughly selecting the initial parameters of MvDGP. Then, taking advantage of powerful uncertainty estimation and robust characteristics of MvDGP, we can perform more precise probability learning in multi-view data. In the processes of training DNN, single GP, as well as MvDGP, stochastic optimization is all adopted to facilitate the generalization of massive data.

The schematic diagram of pre-trained MvDGP (PreMvDGP) is depicted in Fig. 2. The basic MvDGP model is framed in orange lines. The gray node in the outermost circle represents the DNN with a similar structure to MvDGP as the first stage of pre-training. The middle layers of the DNN, $F_{l-1}^{(v)}$, $F_l^{(v)}$, are used as the observed inputs and observed outputs to train the parameters of each single GP, where $v = 1, 2$, $l = 1, \ldots, H^{(v)}$, and $F_0^{(1)} = \mathbf{X}^{(1)}$, $F_0^{(2)} = \mathbf{X}^{(2)}$. The yellow blocks in the second column of the left and the second column of the right are both single GPs as the second stage of pre-training. The training results of each GP are taken as the initial parameters of the corresponding layer in MvDGP. At last, a precise mapping learning is performed through MvDGP.

5 Experiments

In this section, we evaluate the performance of the proposed model in four real-world data sets. Our concerns about model performance include accuracy and training speed. We analyze experimental results compared with the state-of-the-art DGP models and deep neural network.

5.1 Data Sets

1. **WebKB University Data Set (WebKB).** The WebKB data set[1] is composed of four universities, Cornell, Texas, Washington, and Wisconsin, in which data are captured from two views, words in web pages and hyperlinks. The web page can be divided into five categories, where we denote the category of the largest number of samples as positive class and the rest as negative class.
2. **Multiple Feature Data Set (MFeat).** There are 200 samples as well as six features for each handwritten number ('0'–'9') in MFeat data set[2]. We adopt these features as six views. The data is divided into ten partitions denoted as M-0∼M-9, in which partition M-i represents the samples labeled 'i' as positive class and others as negative class samples.
3. **Internet Advertisements Data Set (Ads).** The Ads data set[3] is composed of the features extracted from five aspects. We consider five features as five views of data. There is a unique label to mark if the sample is an ad.
4. **Forest CoverType Data Set (CoverType).** The data[4] are composed of quantitative real variables and binary one-hot variables, for which we adopt two views to model. We use samples labeled Spruce-Fir or Lodgepole as positive samples to form two data sets, respectively (marked as partition C-1 and C-2).

[1] WebKB data set is available at http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/.
[2] Multiple feature data set is available at https://archive.ics.uci.edu/ml/datasets.php.
[3] Ads data set is available at http://archive.ics.uci.edu/ml/datasets.php.
[4] CoverType data set is available at http://archive.ics.uci.edu/ml/datasets.php.

Table 1. Detailed data set information.

Data Set	Sample	Dimension of views						Sample number	
WebKB	number	V1: web page			V2: url			Positive	Negative
Cornell	195	1703			195			83	112
Texas	187	1703			187			103	84
Washington	230	1703			230			107	123
Wisconsin	265	1703			265			122	143
MFeat		V1:fou	V2:fac	V3:kar	V4:pix	V5:zer	V6:mor		
M-0 ∼ M-9	2000	76	216	64	240	27	6	200	1800
Internet		V1:url	V2:origurl	V3:ancurl	V4:alt	V5:caption			
Ads	3279	457	495	472	111	19		458	2821
CoverType		V1: real			V2: binary				
C-1	581012	10			44			211840	369172
C-2	581012	10			44			283301	297711

The total number of samples, dimension of each view, and the sample number of each class for four data sets and partitions are presented detailed in Table 1.

5.2 Experimental Settings

We conduct a series of experiments on four data sets to verify the performance of our PreMvDGP model. For each experiment, we take 5-fold cross-validation to obtain 80% samples for the train set and 20% samples for the test set. We perform ten repeated experiments to each sample partition and take the average as the final experimental results. We adopt 20 samples as a minibatch and 128 inducing points for every layer in the experiments. The number of hidden layers of different views and the shared layers can be customized by the characteristics of each view data. To illustrate the general characteristics of PreMvDGP, we show the experimental results with $L = 1, R = 1, H = 1$.

To demonstrate the superior performance of our model, we compare with two state-of-the-art DGP methods, including doubly stochastic variational inference DGP (DSVI-DGP) [13] and stochastic gradient Hamilton Monte Carlo DGP (SGHMC-DGP) [3], and the deep neural network (DNN) which is designed to adapt to multi-view data in this experiments. Since the single-view DGP methods cannot utilize multi-view data directly, we consider separately taking the data of view 1 (V1), view 2 (V2), and the concatenation of two view data (Con) as three types of inputs for WebKB data set to verify the necessity of multi-view modeling.

Experiments using multiple single-source data are redundant and incomplete, so we concatenate the data from all views as the inputs of the other three data sets to make the most of the data. For single-view DGP methods, we abbreviate the methods as DSVI-DGP-Con, SGHMC-DGP-Con. To ensure adequate training and convergence, we use 500 epochs to train DSVI-DGP and SGHMC-DGP, respectively. In the pre-training phase of PreMvDGP, we set the number of hidden units as 64 and the dimension of hidden units as 10 to get a rough

Table 2. The average classification accuracies (%), standard deviations, and computational time(s) of comparison methods and PreMvDGP on the WebKB data sets.

Model	Dataset							
	Cornell		Texas		Washington		Wisconsin	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
DSVI-DGP-V1	69.2 ± 6.0	306	93.1 ± 2.1	297	81.7 ± 4.3	345	90.3 ± 1.6	414
DSVI-DGP-V2	56.7 ± 4.2	46	62.1 ± 2.4	40	65.2 ± 1.6	55	73.2 ± 1.2	68
DSVI-DGP-Con	70.0 ± 5.8	455	94.0 ± 1.7	434	82.0 ± 4.3	523	91.5 ± 1.7	563
SGHMC-DGP-V1	62.5 ± 3.4	1597	83.1 ± 2.9	1572	77.3 ± 3.7	1655	91.3 ± 1.7	1686
SGHMC-DGP-V2	57.9 ± 2.8	206	59.7 ± 2.8	197	63.0 ± 0.9	228	62.2 ± 3.0	273
SGHMC-DGP-Con	62.8 ± 3.2	1846	77.8 ± 4.1	1821	68.0 ± 3.2	1940	89.4 ± 1.7	1948
DNN	67.1 ± 6.9	61	91.5 ± 3.7	47	72.1 ± 4.9	67	85.6 ± 3.2	81
PreMvDGP	$\mathbf{84.4} \pm 4.4$	141	$\mathbf{95.7} \pm 1.3$	131	$\mathbf{88.6} \pm 5.6$	172	$\mathbf{92.8} \pm 1.6$	200

set of parameters as quickly as possible. Meanwhile, we set 300 epochs for DNN pre-training, 100 epochs for training single GPs, and 100 epochs for training MvDGP. In practice, the number of iterations set in this way can ensure that each step is completely trained. All parameter settings in our experiments remain fixed in each dataset and comparison method.

5.3 Results and Analysis

The experimental results on the four WebKB data sets, including average classification accuracies, standard deviations, and computational costs, are presented in Table 2. Experimental results show that the representation with only view 1 is significantly better than the representation with only view 2 in this data set. Concatenating data from two views (Con) has no significant effect on improving accuracy compared to results with view 1 (V1). In Table 2, the results of (Con) achieves better than (V1) for DSVI-DGP, while the results of (V1) take a bit advantage than (Con) for SGHMC-DGP. Concatenating data from different views causes an increase in the dimensions of the inputs, making the training process more expensive. The experiments prove that PreMvDGP achieves better classification performance than comparison methods, indicating that single-view methods cannot model the data characteristics of different views properly.

Since DNN is used as the initializer in our model, we also list the average time required for 300 iterations of DNN and the average classification accuracy only using the DNN optimizer. It can be found that the computational time of the pre-trainer takes a small part of the total time, and the training results of the DNN are suitable for the initialization of the MvDGP. PreMvDGP with appropriate initial parameters speeds up the training and learns function approximation more subtly than only using the DNN, resulting in more competitive results.

We model the data from six and five views separately for MFeat and Ads data sets, which means that our approach can be easily generalized to more views instead of using combinations of any two views. The experimental results

Table 3. The average classification accuracies (%), standard deviations, and computational time (s) on multiple data sets and partitions, i.e., MFeat (M-0∼M-9), Ads, CoverType (C-1, C-2).

Data set	DSVI-DGP-Con		SGHMC-DGP-Con		DNN		PreMvDGP	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
M-0	99.43 ± 0.17	3541	90.23 ± 3.33	4091	99.36 ± 0.19	513	**99.55** ± 0.25	1273
M-1	99.05 ± 0.39	3527	83.43 ± 9.10	4120	98.81 ± 0.23	489	**99.63** ± 0.14	1164
M-2	98.75 ± 1.92	3607	89.01 ± 4.05	4230	99.52 ± 0.08	584	**99.58** ± 0.11	1261
M-3	98.05 ± 1.71	3606	86.05 ± 2.45	4326	**99.42** ± 0.16	552	99.20 ± 0.35	1240
M-4	99.37 ± 0.14	3476	90.82 ± 5.70	4560	99.81 ± 0.08	584	**99.98** ± 0.12	1238
M-5	98.25 ± 1.75	3627	87.45 ± 1.78	4765	98.62 ± 0.21	586	**98.92** ± 0.33	1256
M-6	99.20 ± 0.25	3511	86.68 ± 2.27	4764	99.47 ± 0.11	590	**99.60** ± 0.22	1328
M-7	99.97 ± 0.03	3499	85.80 ± 4.94	4771	99.60 ± 0.23	584	**99.98** ± 0.08	1238
M-8	99.55 ± 0.29	3570	88.00 ± 1.64	4292	99.40 ± 0.28	538	**99.60** ± 0.14	1272
M-9	99.37 ± 0.24	3481	87.20 ± 3.09	4385	99.35 ± 0.15	529	**99.50** ± 0.21	1264
Ads	95.15 ± 0.33	3352	94.75 ± 0.26	3154	95.87 ± 0.23	458	**97.13** ± 0.36	1290
C-1	63.95 ± 1.01	21474	63.93 ± 1.26	9764	78.57 ± 0.41	3071	**80.61** ± 1.29	7360
C-2	59.88 ± 3.49	21672	57.51 ± 7.79	9771	76.61 ± 0.97	3164	**78.84** ± 2.06	7411

including accuracies and computation time in the other three data sets are shown in Table 3. Our method almost achieves the best accuracy and is dominant in running time in all data sets and partitions, which means that discriminately modeling data of different views is necessary and the pre-training technique plays an important role in optimizing the initial parameters. Significantly, PreMvDGP also works well in the large forest CoverType data set. Stochastic optimization and inducing points help save the computational overhead of our model. Experiments prove that our method is appropriate for multi-view scenarios of large-scale data.

6 Conclusions

In this paper, we propose an end-to-end multi-view deep Gaussian process (MvDGP) model, which is suitable for modeling multi-view data. The inference is based on doubly stochastic optimization and can be applied in large-scale data scenarios. To speed up the training, we introduce a pre-training deep neural network in MvDGP. The initial parameters obtained by the pre-training are proper for MvDGP, and more precise learning is performed by MvDGP. Experimental results demonstrate that pre-trained MvDGP (PreMvDGP) outperforms the state-of-the-art DGP methods in multi-view data modeling, and achieves better performance in training speed. Our work is a generalization of DGP in multi-view scenarios, which helps to develop the MvDGP under the trend of large-scale data with its superior computational performance.

Acknowledgments. The corresponding author Jing Zhao would like to thank supports from the National Natural Science Foundation of China under Projects 61673179, Shanghai Knowledge Service Platform Project (No. ZF1213) and Shanghai Sailing Program 17YF1404600.

References

1. Dai, Z., Damianou, A., González, J., Lawrence, N.: Variational auto-encoded deep Gaussian processes. arXiv preprint arXiv:1511.06455 (2015)
2. Damianou, A., Lawrence, N.: Deep Gaussian processes. In: Artificial Intelligence and Statistics, pp. 207–215 (2013)
3. Havasi, M., Hernández-Lobato, J.M., Murillo-Fuentes, J.J.: Inference in deep Gaussian processes using stochastic gradient hamiltonian monte carlo. In: Advances in Neural Information Processing Systems, pp. 7506–7516 (2018)
4. Hinton, G.E., Salakhutdinov, R.R.: A better way to pretrain deep boltzmann machines. In: Advances in Neural Information Processing Systems, pp. 2447–2455 (2012)
5. Ko, J., Fox, D.: GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. Auton. Robots **27**, 75–90 (2009)
6. Koriyama, T., Kobayashi, T.: A training method using DNN-guided layerwise pretraining for deep Gaussian processes. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2787–2791 (2019)
7. Lee, J., Bahri, Y., Novak, R., Schoenholz, S.S., Pennington, J., Sohl-Dickstein, J.: Deep neural networks as Gaussian processes. arXiv preprint arXiv:1711.00165 (2017)
8. Liu, Q., Sun, S.: Multi-view regularized Gaussian processes. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 655–667 (2017)
9. Liu, Q., Sun, S.: Sparse multimodal Gaussian processes. In: International Conference on Intelligent Science and Big Data Engineering, pp. 28–40 (2017)
10. Matthews, A.G.d.G., Rowland, M., Hron, J., Turner, R.E., Ghahramani, Z.: Gaussian process behaviour in wide deep neural networks. arXiv preprint arXiv:1804.11271 (2018)
11. Rasmussen, C.E., Nickisch, H.: Gaussian processes for machine learning toolbox. J. Mach. Learn. Res. **11**, 3011–3015 (2010)
12. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082 (2014)
13. Salimbeni, H., Deisenroth, M.: Doubly stochastic variational inference for deep Gaussian processes. In: Advances in Neural Information Processing Systems, pp. 4588–4599 (2017)
14. Salimbeni, H., Dutordoir, V., Hensman, J., Deisenroth, M.P.: Deep Gaussian processes with importance-weighted variational inference. arXiv preprint arXiv:1905.05435 (2019)
15. Snelson, E., Ghahramani, Z.: Sparse Gaussian processes using pseudo-inputs. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2006)
16. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. In: Advances in Neural Information Processing Systems, pp. 2951–2959 (2012)
17. Sun, S.: A survey of multi-view machine learning. Neural Comput. Appl. **23**, 2031–2038 (2013)

18. Sun, S., Liu, Q.: Multi-view deep Gaussian processes. In: International Conference on Neural Information Processing, pp. 130–139 (2018)
19. Yu, D., Deng, L., Seide, F.T.B., Li, G.: Discriminative pretraining of deep neural networks (2016)
20. Zhao, J., Xie, X., Xu, X., Sun, S.: Multi-view learning overview: recent progress and new challenges. Inf. Fusion **38**, 43–54 (2017)