Gaussian Process Latent Random Field

Guoqiang Zhong[†], **Wu-Jun Li**[‡], **Dit-Yan Yeung**[‡], **Xinwen Hou**[†], **Cheng-Lin Liu**[†]
[†] National Laboratory of Pattern Recognition (NLPR),
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[‡] Department of Computer Science and Engineering, The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong, China
gqzhong@nlpr.ia.ac.cn, {liwujun, dyyeung}@cse.ust.hk, {xwhou, liucl}@nlpr.ia.ac.cn

Abstract

The Gaussian process latent variable model (GPLVM) is an *unsupervised* probabilistic model for nonlinear dimensionality reduction. A *supervised* extension, called discriminative GPLVM (DGPLVM), incorporates supervisory information into GPLVM to enhance the classification performance. However, its limitation of the latent space dimensionality to at most $C-1$ (C is the number of classes) leads to unsatisfactorily performance when the intrinsic dimensionality of the application is higher than $C-1$. In this paper, we propose a novel *supervised* extension of GPLVM, called Gaussian process latent random field (GPLRF), by enforcing the latent variables to be a Gaussian Markov random field with respect to a graph constructed from the supervisory information. In GPLRF, the dimensionality of the latent space is no longer restricted to at most $C-1$. This makes GPLRF much more flexible than DGPLVM in applications. Experiments conducted on both synthetic and real-world data sets demonstrate that GPLRF performs comparably with DGPLVM and other state-of-the-art methods on data sets with intrinsic dimensionality at most $C-1$, and dramatically outperforms DGPLVM on data sets when the intrinsic dimensionality exceeds $C-1$.

Introduction

In many artificial intelligence applications, one often has to deal with high-dimensional data. Such data require dimensionality reduction to reveal the low-dimensional latent structure of the data so that the underlying tasks, such as visualization, classification and clustering, can benefit from it. Many dimensionality reduction methods have been proposed over the past few decades. Nevertheless, classical linear dimensionality reduction methods such as principal component analysis (PCA) (Joliffe 1986) and multidimensional scaling (MDS) (Cox and Cox 2001) remain to be popular choices due to their simplicity and efficiency. However, they fail to discover nonlinear latent structure of the data in more complex data sets. Starting from about a decade ago, a number of nonlinear manifold learning methods such as isometric feature mapping (Isomap) (Tenenbaum, Silva, and Langford 2000) and locally linear embedding (LLE) (Roweis and Saul 2000) have been proposed. They can discover the low-dimensional manifold structure of the data embedded in a high-dimensional space. However, under situations with relatively sparse or noisy data sets, it was found that these methods do not perform well (Geiger, Urtasun, and Darrell 2009).

The Gaussian process latent variable model (GPLVM) (Lawrence 2005) is a fully probabilistic, nonlinear latent variable model based on Gaussian processes (Rasmussen and Williams 2006). GPLVM can learn a nonlinear mapping from the latent space to the observation space. It has achieved very promising performance in many real-world applications, especially for situations with only a small number of training examples, i.e., sparse data sets. As pointed out by Lawrence and Quiñonero-Candela (2006), GPLVM can preserve the *dissimilarity* between points, which means that the points will be far apart in the latent space if they are far apart in the observation space. However, GPLVM cannot preserve the *similarity* between points, i.e., the points that are close in the observation space are not necessarily close in the latent space. Typically, the points that are close in the observation space are expected to belong to the same class. Consequently, in GPLVM, there is no guarantee that data points from the same class are close in the latent space, which makes the learned latent representation not necessarily good for discriminative applications.

In many applications, we often have some supervisory information such as class labels though the information may be rather limited. However, GPLVM is *unsupervised* in nature. If we can explicitly incorporate the supervisory information into the learning procedure of GPLVM to make the points from the same class close in the latent space, we can obtain a more discriminative latent representation. To the best of our knowledge, only one work, called discriminative GPLVM (DGPLVM) (Urtasun and Darrell 2007), has integrated supervisory information into the GPLVM framework. However, since DGPLVM is based on the linear discriminant analysis (LDA) (Fukunnaga 1991) or generalized discriminant analysis (GDA) (Baudat and Anouar 2000) criterion, the dimensionality of the learned latent space in DGPLVM is restricted to at most $C-1$, where C is the number of classes. For applications with intrinsic dimensionality equal to or higher than C, DGPLVM might not be able to deliver satisfactory performance. This will be verified by the experiments reported later in this paper.

In this paper, we propose a novel supervised GPLVM method, called *Gaussian process latent random field* (GPLRF), by enforcing the latent variables to be a Gaussian Markov random field (GMRF) (Rue and Held 2005) with respect to (w.r.t.) a graph constructed from the supervisory information. Some promising properties of GPLRF are highlighted below:

- GPLRF is nonlinear and hence can deal with complex data sets which linear methods, such as PCA and MDS, cannot handle.
- Compared with GPLVM, GPLRF can learn a more discriminative latent representation in which data points of the same class are clustered together and those of different classes form different clusters.
- The dimensionality of the latent space is no longer restricted to at most $C - 1$, making GPLRF more flexible than DGPLVM in applications.
- GPLRF achieves performance at least comparable with DGPLVM and other state-of-the-art methods on data sets with intrinsic dimensionality at most $C - 1$, and dramatically outperforms DGPLVM on data sets when the intrinsic dimensionality exceeds $C - 1$.

Gaussian Process Latent Variable Model

Given a set of N training examples represented as a matrix $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]^T$, where $\mathbf{y}_i \in \mathbb{R}^d$. Let the matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T$, where $\mathbf{x}_i \in \mathbb{R}^q$ with $q < d$, denote their corresponding positions in the latent space. In the context of latent variable models, we often call $\mathbf{Y}$ the *observed data* and $\mathbf{X}$ the *latent representation* (or latent variables) of $\mathbf{Y}$. GPLVM relates each high-dimensional observation $\mathbf{y}_i$ with its corresponding latent position $\mathbf{x}_i$ using a Gaussian process mapping from the latent space to the observation space. Given a covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$ for the Gaussian process, the likelihood of the observed data given the latent positions is

$$p(\mathbf{Y} \,|\, \mathbf{X}) = \frac{1}{\sqrt{(2\pi)^{Nd}|\mathbf{K}|^d}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T)\right), \quad (1)$$

where $\mathbf{K}$ is the kernel matrix with elements $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. We use a kernel defined as follows:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \exp(-\frac{\theta_2}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2) + \theta_3 + \frac{\delta_{ij}}{\theta_4}, \quad (2)$$

where δ_{ij} is the Kronecker delta function and $\theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}$ are the kernel parameters. Maximizing the likelihood is equivalent to minimizing

$$\mathcal{L}_r = \frac{d}{2}\ln|\mathbf{K}| + \frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) + \frac{Nd}{2}\ln(2\pi). \quad (3)$$

The gradients of (3) w.r.t. the latent variables can be computed through combining

$$\frac{\partial \mathcal{L}_r}{\partial \mathbf{K}} = \frac{d}{2}\mathbf{K}^{-1} - \frac{1}{2}\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1} \quad (4)$$

with $\frac{\partial \mathbf{K}}{\partial x_{ij}}$ via the chain rule, where x_{ij} is the jth dimension of $\mathbf{x}_i$. Based on (3) and (4), a nonlinear optimizer such as scaled conjugate gradient (SCG) (Møler 1993) can be used to learn the latent variables. The gradients of (3) with respect to the parameters of the kernel function in (2) can also be computed and used to jointly optimize $\mathbf{X}$ and the kernel parameters θ.

Our Model

In this section, we introduce in detail our proposed model, GPLRF, including its formulation, learning algorithm and out-of-sample prediction. GPLRF is a supervised extension of GPLVM based on a GMRF (Rue and Held 2005).

Gaussian Process Latent Random Field

The basic idea of GPLRF is to enforce the latent variables $\mathbf{X}$ to be a GMRF (Rue and Held 2005) w.r.t. a graph constructed from the supervisory information. Based on the constructed GMRF, we get a prior for $\mathbf{X}$ and then apply *maximum a posteriori* (MAP) estimation to learn $\mathbf{X}$.

GMRF Construction We define an *undirected* graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the node set $\mathcal{V} = \{V_1, V_2, \cdots, V_N\}$ and V_i corresponds to a training example $\mathbf{y}_i$ (or $\mathbf{x}_i$), and $\mathcal{E} = \{(V_i, V_j) |\ i \neq j, \mathbf{y}_i \text{ and } \mathbf{y}_j \text{ belong to the same class}\}$ is the edge set. If we associate each edge with a weight 1, we can get a weight matrix $\mathbf{W}$ with its entries defined as follows:

$$\mathbf{W}_{ij} = \begin{cases} 1 & \text{if } \mathbf{y}_i \text{ and } \mathbf{y}_j,\ i \neq j, \text{ belong to the same class} \\ 0 & \text{otherwise.} \end{cases}$$

Hence, we can find that $\mathbf{W}_{ij} = 1$ if and only if there exists an edge between nodes V_i and V_j for any $i \neq j$.

We can see that the graph $\mathcal{G}$ is constructed from the label (supervisory) information. If we associate each node with a random variable, we can get a Markov random field (MRF) (Rue and Held 2005) w.r.t. the graph $\mathcal{G}$. Here, we associate this constructed graph $\mathcal{G}$ with the random vector $\mathbf{X}_{*k} = (\mathbf{X}_{1k}, \mathbf{X}_{2k}, \cdots, \mathbf{X}_{Nk})^T$ (for any $k = 1, 2, \ldots, q$).

Based on the weight matrix $\mathbf{W}$, we can compute the graph Laplacian matrix (Chung 1997) $\mathbf{L}$ as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. With $\mathbf{L}$, we define a prior distribution on the latent variables $\mathbf{X}$ as:

$$p(\mathbf{X}) = \prod_{k=1}^{q} p(\mathbf{X}_{*k}),$$

and

$$p(\mathbf{X}_{*k}) = \frac{1}{Z}\exp\Big[-\frac{\alpha}{2}(\mathbf{X}_{*k}^T\mathbf{L}\mathbf{X}_{*k})\Big], \quad (5)$$

where Z is a normalization constant and $\alpha > 0$ is a scaling parameter. Then we have

Theorem 1 $\mathbf{X}_{*k}$ *is a Gaussian Markov random field (GMRF) w.r.t. the graph* $\mathcal{G}$.

Proof: According to the property of GMRF (Rue and Held 2005), it suffices to prove that the missing edges in $\mathcal{G}$ correspond to the zero entries in the precision matrix (i.e., inverse covariance matrix). Because in (5) the precision matrix is $\mathbf{L}$, it is easy to check that $\mathbf{L}_{ij} = 0$ if and only if $\mathbf{W}_{ij} = 0$ for all $i \neq j$. ■

Hence, we can say that $p(\mathbf{X}_{*k})$ is a meaningful prior for $\mathbf{X}_{*k}$ because its underlying graph $\mathcal{G}$ effectively reflects the *conditional independence* relationship among the random variables. More specifically, on one perspective, from the definition of $\mathbf{W}$, $\mathbf{W}_{ij} = 0$ means that point i and point j are from different classes. On the other perspective, from the semantics implied by the MRF, $\mathbf{W}_{ij} = 0$ means that $\mathbf{X}_{ik}$ is independent of $\mathbf{X}_{jk}$ given the other variables. Because it is reasonable to make the latent representations of two points from different classes independent, these two perspectives are coherent.

Furthermore, if we compute

$$p(\mathbf{X}) = \prod_{k=1}^{q} p(\mathbf{X}_{*k}) = \frac{1}{Z^q} \exp\Big[- \frac{\alpha}{2} \mathrm{tr}(\mathbf{X}^T \mathbf{L} \mathbf{X}) \Big], \tag{6}$$

we can also understand the effectiveness of the prior in (6). To see this, let us rewrite the term $\mathrm{tr}(\mathbf{X}^T\mathbf{L}\mathbf{X})$ as follows:

$$\begin{aligned} \mathrm{tr}(\mathbf{X}^T\mathbf{L}\mathbf{X}) &= \frac{1}{2}\sum_{k=1}^{q}\Big[\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{W}_{ij}(\mathbf{X}_{ik} - \mathbf{X}_{jk})^2\Big] \\ &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\Big[\mathbf{W}_{ij}\sum_{k=1}^{q}(\mathbf{X}_{ik} - \mathbf{X}_{jk})^2\Big] \\ &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{W}_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|^2, \end{aligned} \tag{7}$$

where $\|\mathbf{x}_i - \mathbf{x}_j\|$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. We can see that $\mathrm{tr}(\mathbf{X}^T\mathbf{L}\mathbf{X})$ actually reflects the sum of the distances between points from the same class. The closer the points from the same class are, the smaller is the term $\mathrm{tr}(\mathbf{X}^T\mathbf{L}\mathbf{X})$, and consequently the higher is $p(\mathbf{X})$. Hence, the latent representation which makes the data points from the same class closer will be given higher probability. This is exactly what we need to learn – a discriminative latent space.

Model Formulation of GPLRF With the prior in (6) and the likelihood in (1), we can obtain the posterior distribution

$$p(\mathbf{X} \,|\, \mathbf{Y}) \propto p(\mathbf{Y} \,|\, \mathbf{X}) p(\mathbf{X}). \tag{8}$$

If we use the MAP strategy to estimate $\mathbf{X}$, the supervisory information in $p(\mathbf{X})$ will be seamlessly integrated into the model. We call this model *Gaussian process latent random field* (GPLRF) because $p(\mathbf{Y} \,|\, \mathbf{X})$ corresponds to *Gaussian processes* and the prior $p(\mathbf{X})$ for the *latent* variables is a Gaussian Markov *random field* w.r.t. a graph constructed from the supervisory information.

Then the MAP estimate of $\mathbf{X}$ can be obtained by minimizing the following objective function:

$$\mathcal{L}_s = \frac{d}{2}\ln|\mathbf{K}| + \frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) + \frac{\alpha}{2}\mathrm{tr}(\mathbf{X}^T\mathbf{L}\mathbf{X}). \tag{9}$$

To minimize $\mathcal{L}_s$, we can first compute the gradient of (9)

$$\frac{\partial \mathcal{L}_s}{\partial \mathbf{X}} = \frac{\partial \mathcal{L}_r}{\partial \mathbf{X}} + \alpha \mathbf{L}\mathbf{X},$$

and then apply the SCG method (Møler 1993). This procedure is similar to that for GPLVM.

Discussions The objective function in (9) can be interpreted as a regularized version of GPLVM, where the regularizer is a nonlinear variant of the supervised locality preserving projection (SLPP) model (Zheng et al. 2007). When $\alpha \to +\infty$, (9) has a closed-form solution and is equivalent to a variant of supervised kernel locality preserving projection (SKLPP) model (Zheng et al. 2007) or a variant of Laplacian eigenmap (Belkin and Niyogi 2001).

Although the objective function in (9) is only a simple combination of two components, this combination is very effective because the two components are intrinsically complementary to each other.

Although the Laplacian matrix $\mathbf{L}$ in this paper is constructed from the supervisory information, it may also be constructed based on the similarity between the observed data $\mathbf{Y}$ (often called a similarity graph), which is similar to the Laplacian matrix for Laplacian eigenmap. Another possible extension is that we may combine both supervisory information and a similarity graph to get a semi-supervised version of GPLRF. It is worth noting that all these extensions can be integrated into the GPLRF framework easily. Due to space limitation, these extensions will be left to our future pursuit.

Out-of-Sample Prediction

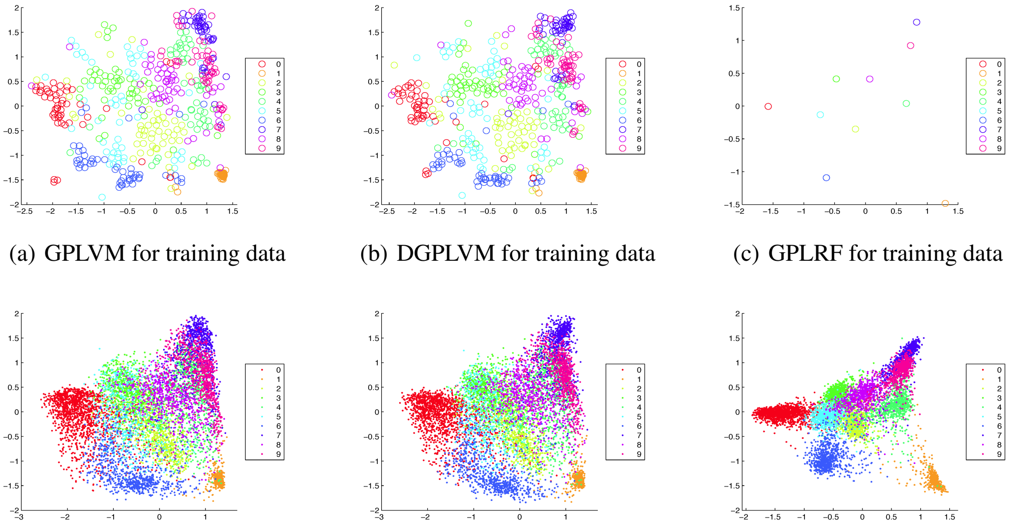
For a new test point $\mathbf{y}_t$, we exploit the back-constrained strategy in (Lawrence and Quiñonero-Candela 2006) and (Urtasun and Darrell 2007) to estimate its low-dimensional representation $\mathbf{x}_t$. More specifically, we minimize (9) with the constraints

$$x_{tj} = g_j(\mathbf{y}_t),$$

where x_{tj} is the jth dimension of $\mathbf{x}_t$ and g_j is a function of the input points. In particular, to get a smooth inverse mapping, we use an RBF kernel for the back-constraint in each dimension $g_j(\mathbf{y}_t) = \sum_{m=1}^{N} \beta_{jm} k(\mathbf{y}_t, \mathbf{y}_m)$, where $\{\beta_{jm}\}(m = 1, 2, \ldots, N; j = 1, 2, \ldots, q)$ are the parameters to learn and the kernel function is $k(\mathbf{y}_n, \mathbf{y}_m) = \exp\big(-\frac{\gamma}{2}\|\mathbf{y}_n - \mathbf{y}_m\|^2\big)$, with γ being the inverse width parameter. The latent position of a test point can be computed directly by evaluating the inverse mapping learned by the back-constraint at the test point $x_{tj} = g_j(\mathbf{y}_t)$.

Experiments

To demonstrate the effectiveness of GPLRF, we compare it with some related methods on both synthetic and real-world data sets. These baseline methods include: 1-nearest neighbor (1NN) classifier in the original space, PCA, LDA, LPP (He and Niyogi 2003), SLPP (Zheng et al. 2007) and their kernel counterparts, GPLVM, DGPLVM (Urtasun and Darrell 2007), LL-GPLVM (Urtasun et al. 2008), rankGPLVM (Geiger, Urtasun, and Darrell 2009), and tSNEGPLVM (van der Maaten 2009). Among them, PCA and LPP are unsupervised, LDA and SLPP are supervised, LPP and SLPP account for preserving similarity between points, PCA, LDA, LPP and SLPP are linear dimensionality reduction methods, and their kernel counterparts and all GPLVM based methods are nonlinear methods.



(a) GPLVM for training data

(b) DGPLVM for training data

(c) GPLRF for training data

(d) GPLVM for test data

(e) DGPLVM for test data

(f) GPLRF for test data

Figure 1: 2D latent spaces learned by GPLVM, DGPLVM and GPLRF on the USPS data set.

We use both the visualization and classification settings to evaluate all these methods. For classification, we first use these methods to perform dimensionality reduction to get the lower-dimensional representation (or latent representation in the case of latent variable models), then a 1NN classifier is employed to predict the labels of the test data in the latent space. For all methods based on GPLVM, testing is repeated with different parameter values: $\alpha \in \{10^{-5}, 10^{-4}, \ldots, 10^{4}, 10^{5}\}$ and $\gamma \in \{0.001, 0.01, 0.1\}$. The settings which result in minimum mean errors over 20 random partitions are used. For the other methods compared, we also choose the best parameter settings over 20 random partitions and report the best classification results.

Three data sets are used in our experiments: the USPS handwritten digit data set[1], the Oil data set[2], and the CMU motion capture (CMU mocap) data set[3]. The USPS data set contains 9298 handwritten digits from 10 classes. The size of each digit image is 16 × 16 pixels (so the dimensionality of the data space is 256). In our experiment, we use normalized digit images with pixel values in $[0, 1]$. For all the dimensionality reduction methods, we first project the data onto a linear subspace by performing PCA on the training data with 99% of the variance retained and then use each method for dimensionality reduction. Other than PCA, no other preprocessing is applied. The Oil data set is a synthetic data set. It contains 1000 examples grouped into three classes, with dimensionality 12. Nevertheless, its intrinsic dimensionality is only two[4]. The CMU mocap data set includes three categories, namely, jumping, running and walking. We choose 49 video sequences from four subjects. For each sequence, the features are generated using Lawrence's method[5], with dimensionality 93.

Visualization

We compare our model with GPLVM and DGPLVM on the visualization of the USPS data set in a 2-dimensional (2D) latent space. The training and test sets are randomly selected from the whole data set, with 50 examples per class for training and the rest for testing. For all three methods, PCA is used for initialization. Figure 1 shows the learned 2D latent spaces by using GPLVM, DGPLVM and GPLRF. As we can see, for the training data, GPLRF learns a 2D latent space in which all 50 data points of each class are almost mapped to the same point, but the latent representations learned by GPLVM and DGPLVM are more scattered with significant overlapping between classes. For the test data, the regions for different digit classes in the latent space can be distinguished more easily using GPLRF but there is more significant overlapping when GPLVM or DGPLVM is used. Hence, we can say that the learned space of GPLRF is more discriminative than those of GPLVM and DGPLVM, which conforms to the theoretical analysis of GPLRF.

Effect of Dimensionality

In this experiment, we assess the performance of GPLRF in latent spaces of different dimensionality. Since DGPLVM is derived from the LDA or GDA criterion, the dimensionality of the DGPLVM latent space is at most $C - 1$, where C is the number of classes. However, GPLRF has no such restriction and can learn latent spaces of dimensionality higher than $C - 1$. Figure 2(a) and Figure 2(b) show the classification errors on the Oil data set and the CMU mocap data set, respectively. On the Oil data set, the best performance of DGPLVM is achieved when the dimensionality of the latent space is equal to $C - 1$. Although GPLRF can learn a

[1] http://www-stat-class.stanford.edu/∼tibs/ElemStatLearn/data.html

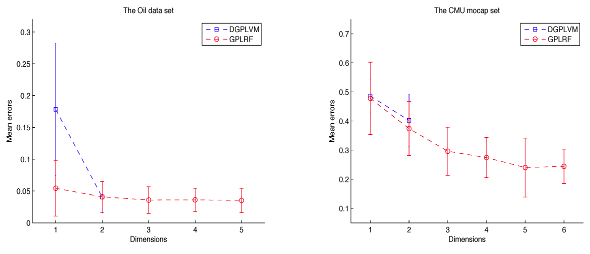
[2] http://is6.cs.man.ac.uk/∼neill/datasets/

[3] http://mocap.cs.cmu.edu/

[4] http://www.ncrg.aston.ac.uk/GTM/3PhaseData.html

[5] http://is6.cs.man.ac.uk/∼neill/mocap/

latent space of higher dimensionality, the performance cannot be further improved by more than 1%. Besides, we can see that DGPLVM and GPLRF achieve comparable performance when the dimensionality of the latent space is set to $C - 1\,(= 2)$, which is the true dimensionality of the data. However, for the CMU mocap data set, the best performance of GPLRF is achieved when the dimensionality is higher than $C - 1$, which means that the intrinsic dimensionality of this data set might be higher than $C - 1$. For this case, DGPLVM cannot achieve satisfactory results due to the dimensionality restriction. Hence, by comparing the results in Figure 2(a) and Figure 2(b), we can conclude that GPLRF can model more complex data sets than DGPLVM.



(a) (b)

Figure 2: Mean errors obtained by DGPLVM and GPLRF in latent spaces of different dimensionality on the Oil data set and the CMU mocap data set.

Classification on the USPS Data Set

In this experiment, we compare our model with 12 other related methods on the USPS data set. For all methods based on GPLVM, we use PCA for initialization. For comparison, the dimensionality of the latent space is set to 9. Mean classification errors and their standard deviations are shown in Table 1, where the best results are shown in bold. We can find that GPLRF achieves the best performance consistently. Although GDA performs well and can obtain mean errors close to GPLRF, paired t-tests still support that, in most cases, GPLRF is significantly better than GDA.

Classification on the Oil Data Set

In this experiment, all the GPLVM based methods are initialized with SLPP. The dimensionality of the latent space is set to 2. Mean classification errors and their standard deviations are shown in Figure 3. Paired t-tests show that, in most cases (with p-value less than 0.1), GPLRF is significantly better than the other methods. Even in cases that GPLRF is not the best, there is no significant difference between GPLRF and the best one. From Figure 2(a), we can find that this data set might be relatively simple. Hence, DGPLVM can achieve performance comparable with GPLRF under some settings on this data set.

Classification on the CMU Mocap Data Set

To evaluate the performance of GPLRF when the input dimensionality is higher than the number of training examples, we carry out an experiment on the CMU mocap data set. We

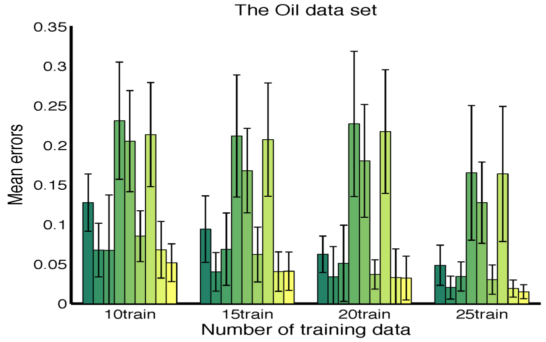


Figure 3: Mean classification errors and standard deviations for nine methods on the Oil data set. The bars in each group represent the classification results in the original space, the learned latent space using SLPP, SKLPP, GPLVM, LL-GPLVM, rankGPLVM, tSNEGPLVM, DGPLVM and GPLRF, respectively.

use PCA to initialize the latent variables of both DGPLVM and GPLRF. Based on Figure 2(b), we set the dimensionality of DGPLVM to 2 and that of GPLRF to 5. The mean classification errors and standard deviations are shown in Figure 4. We can see that GPLRF consistently outperforms DGPLVM and 1NN in the original space. Paired t-tests also support this conclusion.

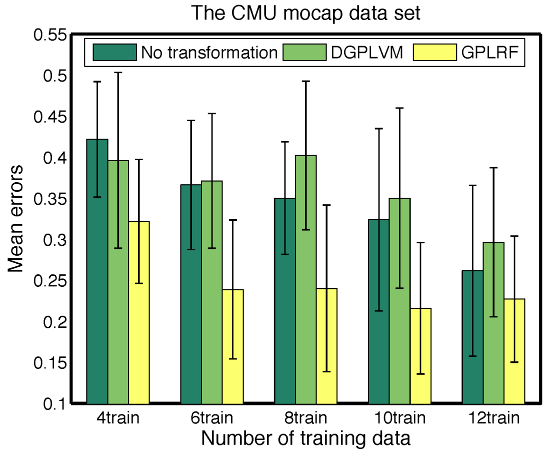


Figure 4: Mean classification errors and standard deviations for three methods on the CMU mocap data set.

Conclusion

In this paper, we have proposed a novel supervised extension of GPLVM, called GPLRF, for nonlinear dimensionality reduction. GPLRF can learn a discriminative latent representation which is beneficial for classification. Extensive experiments on synthetic data and diverse applications demonstrate that GPLRF can achieve performance comparable to other state-of-the-art methods, either linear or nonlinear, for simple data sets, and can dramatically outperform other methods for complex data sets.

Table 1: Mean classification errors and standard deviations obtained by different methods on the USPS data set. The first row, ‘No transformation’, presents the 1NN classification results in the original space. The first four methods below ‘No transformation’ are linear methods and the rest are nonlinear methods. The blank entry for ’rankGPLVM’ is due to out of memory (2GB).

Method	10train	20train	30train	40train	50train
No transformation	0.1976 ± 0.0161	0.1463 ± 0.0069	0.1260 ± 0.0064	0.1107 ± 0.0056	0.0986 ± 0.0047
PCA	0.2497 ± 0.0154	0.2090 ± 0.0133	0.1849 ± 0.0122	0.1757 ± 0.0136	0.1631 ± 0.0054
LDA	0.2636 ± 0.0275	0.1937 ± 0.0103	0.1657 ± 0.0068	0.1520 ± 0.0093	0.1392 ± 0.0055
LPP	0.3433 ± 0.0356	0.2761 ± 0.0284	0.2470 ± 0.0154	0.2150 ± 0.0131	0.1929 ± 0.0125
SLPP	0.2448 ± 0.0204	0.2070 ± 0.0095	0.1792 ± 0.0067	0.1639 ± 0.0097	0.1498 ± 0.0063
KPCA	0.3515 ± 0.0260	0.2764 ± 0.0294	0.2490 ± 0.0179	0.2422 ± 0.0154	0.2204 ± 0.0115
GDA	0.1827 ± 0.0240	0.1455 ± 0.0205	0.0971 ± 0.0044	0.0921 ± 0.0070	0.0729 ± 0.0053
KLPP	0.5087 ± 0.0490	0.5024 ± 0.0414	0.4751 ± 0.0316	0.4645 ± 0.0257	0.4517 ± 0.0206
SKLPP	0.2155 ± 0.0402	0.1702 ± 0.0453	0.1450 ± 0.0440	0.1345 ± 0.0493	0.1059 ± 0.0374
GPLVM	0.2554 ± 0.0191	0.1940 ± 0.0090	0.1695 ± 0.0155	0.1422 ± 0.0070	0.1311 ± 0.0079
LL-GPLVM	0.2533 ± 0.0200	0.1908 ± 0.0158	0.1637 ± 0.0152	0.1393 ± 0.0094	0.1278 ± 0.0088
rankGPLVM	0.1930 ± 0.0146	0.1427 ± 0.0075	0.1186 ± 0.0067	0.1084 ± 0.0529	–
tSNEGPLVM	0.2399 ± 0.0138	0.1932 ± 0.0096	0.1673 ± 0.0157	0.1418 ± 0.0072	0.1303 ± 0.0084
DGPLVM	0.2491 ± 0.0230	0.1922 ± 0.0102	0.1692 ± 0.0155	0.1442 ± 0.0122	0.1303 ± 0.0081
GPLRF	**0.1769 ± 0.0165**	**0.1129 ± 0.0113**	**0.0946 ± 0.0100**	**0.0809 ± 0.0085**	**0.0690 ± 0.0042**

Acknowledgments

This research is partially supported by the NSFC Outstanding Youth Fund (No. 60825301) and by General Research Fund 621407 from the Research Grants Council of the Hong Kong Special Administrative Region, China.

References

Baudat, G., and Anouar, F. 2000. Generalized discriminant analysis using a kernel approach. *Neural Computation* 12:2385–2404.

Belkin, M., and Niyogi, P. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, 585–591.

Chung, F. 1997. *Spectral Graph Theory*. Number 92 in Regional Conference Series in Mathematics. American Mathematical Society.

Cox, T., and Cox, M. 2001. *Multidimensional Scaling*. Chapman and Hall, Boca Raton.

Fukunnaga, K. 1991. *Introduction to Statistical Pattern Recognition, second edition*. Academic Press.

Geiger, A.; Urtasun, R.; and Darrell, T. 2009. Rank priors for continuous nonlinear dimensionality reduction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 880–887.

He, X., and Niyogi, P. 2003. Locality preserving projections. In *Advances in Neural Information Processing Systems*.

Joliffe, I. 1986. *Principal Component Analysis*. Springer-Verlag.

Lawrence, N., and Quiñonero-Candela, J. 2006. Local distance preservation in the GP-LVM through back constraints. In *Proceedings of the International Conference on Machine Learning*, 96–103.

Lawrence, N. 2005. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research* 6:1783–1816.

Møler, F. 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 6(4):525–533.

Rasmussen, C. E., and Williams, C. 2006. *Gaussian Processes for Machine Learning*. Monographs on Statistics and Applied Probability. MIT Press.

Roweis, S., and Saul, L. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326.

Rue, H., and Held, L. 2005. *Gaussian Markov Random Fields: Theory and Applications*, volume 104 of *Monographs on Statistics and Applied Probability*. London: Chapman & Hall.

Tenenbaum, J.; Silva, V.; and Langford, J. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323.

Urtasun, R., and Darrell, T. 2007. Discriminative Gaussian process latent variable models for classification. In *Proceedings of the International Conference on Machine Learning*, 927–934.

Urtasun, R.; Fleet, D. J.; Geiger, A.; Popovic, J.; Darrell, T.; and Lawrence, N. 2008. Topologically-constrained latent variable models. In *Proceedings of the International Conference on Machine Learning*, 1080–1087.

van der Maaten, L. 2009. Preserving local structure in Gaussian process latent variable models. In *Proceedings of the 18th Annual Belgian-Dutch Conference on Machine Learning*, 81–88.

Zheng, Z.; Yang, F.; Tan, W.; Jia, J.; and Yang, J. 2007. Gabor feature-based face recognition using supervised locality preserving projection. *Signal Processing* 87(10):2473–2483.