Preserving Local Structure in Gaussian Process Latent Variable Models

Laurens van der Maaten
TiCC, Tilburg University
P.O. Box 90153, 5000 LE Tilburg, The Netherlands
lvdmaaten@gmail.com

Abstract

The Gaussian Process Latent Variable Model (GPLVM) is a non-linear variant of probabilistic Principal Components Analysis (PCA). The main advantage of the GPLVM over probabilistic PCA is that it can model non-linear transformations from the latent space to the data space. An important disadvantage of the GPLVM is its focus on preserving global data structure in the latent space, whereas preserving local data structure is generally considered to be more important in dimensionality reduction. In this paper, we present an extension of the GPLVM that encourages the preservation of local structure in the latent space. The extension entails the introduction of a prior distribution over the parameters of the GPLVM that measures the divergence between the pairwise distances in the data space and the latent space. We show that the proposed extension leads to strong results.

1 Introduction

Machine learning techniques are often hampered by the high dimensionality of the data on which they are trained. For instance, the number of weights in a neural network typically grows very fast with the dimensionality of the input data. To address this problem, a large number of techniques have been proposed that reduce the dimensionality of the input data. See for a review, e.g., (Lee and Verleysen, 2007). The rationale behind these dimensionality reduction techniques is that the input data is typically not uniformly distributed over the data space. For instance, consider the space of images of size 512×512 pixels. If we draw samples from a uniform distribution defined over this $262,144$-dimensional space, the probability of sampling a natural image is almost zero. Only small parts of the image space are thus filled with natural images. Moreover, the strong correlations between individual image pixels suggest the number of parameters needed to account for all properties of the data is much smaller than $262,144$. One can thus think of images and many other high-dimensional datasets as lying on a non-linear manifold of lower dimensionality that is embedded into the data space. The aim of dimensionality reduction techniques is to transform the input data to a low-dimensional latent space in such a way that the structure of the manifold is retained.

A particularly interesting type of dimensionality reduction techniques are probabilistic latent variable models. Probabilistic latent variable models define a generative model in which a datapoint $\mathbf{x}_n$ is generated from a distribution that is conditioned on a latent variable $\mathbf{z}_n$ and on the parameters of the model. The parameters of the model are typically learned using maximum likelihood learning, although a fully Bayesian treatment may be used as well (Bishop, 1999). An important advantage of latent variable models is that, in contrast to other dimensionality reduction techniques, they provide a probabilistic mapping from the latent space to the data space that is typically *smooth*. Such a probabilistic mapping can be employed to sample new data from the model distribution.

A well-known latent variable model is probabilistic Principal Components Analysis (PCA), which is a linear-Gaussian model for which the maximum-likelihood solution is identical to the solution of 'normal' PCA (Tipping and Bishop, 1999). The main limitations of probabilistic PCA are that (1) it can only learn linear mappings between the latent space and the data space and (2) it does not retain the local structure of the data very well in the latent space. Lawrence (2005) extended the probabilistic PCA model to the Gaussian Process Latent Variable Model (GPLVM) to address the first limitation: the GPLVM is capable of modeling non-linear mappings between the latent space and the data space. However, the GPLVM does not address the second limitation of probabilistic PCA: in both probabilistic PCA and the GPLVM, there is no emphasis on preserving small pairwise distances between datapoints in the latent space. Instead, the GPLVM primarily focuses on retaining global data structure in the latent space. The focus of the GPLVM on retaining global data structure in the latent space conflicts with the popular belief that preserving local data structure is most important in dimensionality reduction (Tenenbaum et al., 2000; Roweis and Saul, 2000; van der Maaten and Hinton, 2008). By preserving the local structure of the data, non-linear low-dimensional data manifolds can be successfully extracted from the data space and embedded in a low-dimensional latent space. Herein, some of the global data structure is lost, however, this global structure is generally not of relevance in typical learning tasks. The importance of preserving local data structure in dimensionality reduction is often illustrated using artificial datasets such as the 'Swiss roll' dataset (Tenenbaum et al., 2000). Preservation of local data structure is the key idea underlying succesful dimensionality reduction techniques such as Isomap (Tenenbaum et al., 2000), LLE (Roweis and Saul, 2000), and t-SNE (van der Maaten and Hinton, 2008).

In this paper, we present an extension of the GPLVM that encourages the preservation of local data structure in the latent space and that does not affect the desirable properties of the GPLVM, such as its non-linear probabilistic mapping from the latent space to the data space. The extension entails the introduction of a data-dependent 'prior' distribution over the parameters of the GPLVM. The prior depends on the divergence between the pairwise distances in the data space and the latent space that is also employed in t-SNE (van der Maaten and Hinton, 2008). The prior takes the form of a Boltzmann distribution, in which the energy function is formed by the objective function of t-SNE. We present experiments in which we compare our extended model to the standard GPLVM. The result of the experiments reveal that the extended model significantly outperforms the standard GPLVM in terms of the nearest-neighbor error of the data representation in the latent space.

The outline of the remainder of this paper is as follows. In Section 2, we review the probabilistic PCA model. Section 3 describes the Gaussian Process Latent Variable Model. In Section 4, we present our approach to the preservation of local structure in the GPLVM. Section 5 presents experiments in which we evaluate the performance of our extended GPLVM. The results of the experiments are discussed in more detail in Section 6. Section 7 concludes the paper and discusses directions for future work.

2 Probabilistic PCA

Probabilistic PCA (Tipping and Bishop, 1999) is a linear-Gaussian generative model that is illustrated in Figure 1(a). The model assumes that a dataset $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ is generated conditioned on a set of latent variables $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ and a set of parameters $\Theta = \{\mathbf{W}, \beta\}$ as follows:

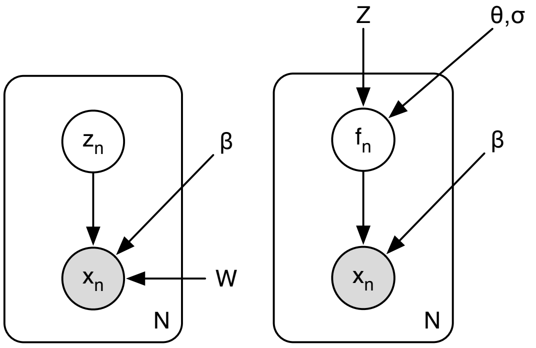
- For n is 1 to N:
  - Sample $\mathbf{z}_n \sim \mathcal{N}(\mathbf{z}_n | 0, \mathbf{I}_d)$.
  - Sample $\mathbf{x}_n \sim \mathcal{N}(\mathbf{x}_n | \mathbf{z}_n \mathbf{W}, \beta^{-1} \mathbf{I}_D)$.

Herein, d represents the dimensionality of the latent space, D the dimensionality of the data space, $\mathbf{I}_j$ is the $j \times j$ identity matrix, $\mathbf{W}$ represents a $d \times D$ linear mapping from the latent space to the data space, and β represents the precision of the Gaussian noise model. The likelihood of the dataset $\mathbf{X}$ under the model can now be obtained by marginalizing out the latent variables $\mathbf{z}_n$ as follows:

$$p(\mathbf{X}|\mathbf{W}, \beta) = \prod_{n=1}^{N} p(\mathbf{x}_n | \mathbf{W}, \beta) = \prod_{n=1}^{N} \int p(\mathbf{x}_n | \mathbf{z}_n, \mathbf{W}, \beta) p(\mathbf{z}_n) d\mathbf{z}_n. \tag{1}$$

It can be shown that the likelihood function can be maximized by setting $\mathbf{W}$ to be the principal eigenvectors of the data, i.e., the maximum-likelihood solution corresponds to the standard PCA solution (Tipping and Bishop, 1999).

Lawrence (2005) showed that the likelihood function of probabilistic PCA is identical to the likelihood function of a similar model which differs from the traditional formulation of probabilistic PCA in three respects: (1) the latent variables $\mathbf{z}_n$ are treated as parameters, (2) a Gaussian prior is defined over the columns $\mathbf{w}_i$ of the linear mapping $\mathbf{W}$, and (3) the columns $\mathbf{w}_i$ are marginalized out



(a) Probabilistic PCA. (b) GPLVM.

Figure 1: Generative models for probabilistic PCA and GPLVM.

instead of the latent points $\mathbf{z}_n$. Mathematically, we define:

$$p(\mathbf{W}) = \prod_{i=1}^{D} \mathcal{N}(\mathbf{w}_i|0, \mathbf{I}_d)$$
$$p(\mathbf{x}_n|\mathbf{W}, \mathbf{z}_n, \beta) = \mathcal{N}(\mathbf{x}_n|\mathbf{z}_n\mathbf{W}, \beta^{-1}\mathbf{I}_D),$$

where $\mathbf{w}_i$ represents a single column of the linear mapping $\mathbf{W}$, and again, β is the precision of the noise model. The likelihood function of the resulting model is given by:

$$\begin{aligned} p(\mathbf{X}|\mathbf{Z}, \beta) &= \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n, \beta) \\ &= \prod_{n=1}^{N} \int p(\mathbf{x}_n|\mathbf{W}, \mathbf{z}_n, \beta) p(\mathbf{W}) d\mathbf{W} \\ &= \prod_{i=1}^{D} \mathcal{N}(\mathbf{X}^{(i)}|0, \mathbf{K}), \end{aligned} \tag{2}$$

where $\mathbf{X}^{(i)}$ represents the i-th column of $\mathbf{X}$, and $\mathbf{K} = \mathbf{Z}\mathbf{Z}^T + \beta^{-1}\mathbf{I}_N$. This likelihood function can be shown to be identical to the likelihood function in Equation 1 (Lawrence, 2005).

3 Gaussian Process Latent Variable Model

The expression in Equation 2 can be recognized as a product of D Gaussian processes for which the covariance function is a linear function. The Gaussian process defines a distribution over functions $f(\mathbf{z})$ in such a way that the set of values $f(\mathbf{z})$ evaluated at any set of points $\mathbf{z}_1, \ldots, \mathbf{z}_N$ is jointly Gaussian distributed (MacKay, 1998). Instead of defining a distribution over the set of linear mappings, as in the second formulation of probabilistic PCA, we can thus use Gaussian processes to define a distribution over all linear and non-linear functions. A key property of the Gaussian process is that the joint distribution $p(f(\mathbf{z}_1), \ldots, f(\mathbf{z}_N))$ is completely specified by its second-order statistics, i.e., by the covariance matrix $\mathbf{K}$. As a result, the probabilistic PCA model can be turned in a non-linear model by simply defining the entries of the covariance matrix $\mathbf{K}$ to be given by a non-linear covariance function $\kappa(\mathbf{z}_i, \mathbf{z}_j)$. In the remainder of the paper, we assume the covariance function is given by a Gaussian kernel function with an additional bias term:

$$\kappa(\mathbf{z}_i, \mathbf{z}_j) = \exp\left(-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{2\sigma^2}\right) + \theta,$$

where σ represents the bandwidth of the kernel and θ represents the bias term. Following Lawrence (2005), we set $\theta = \exp(-1)$ in all our experiments.

The generative model for the GPLVM is depicted in Figure 1(b), where we introduced an additional latent variable $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_N\}$ to make the difference between Gaussian process and the noise model explicit. The likelihood function in the model is given by:

$$p(\mathbf{X}|\mathbf{Z}, \beta, \theta, \sigma) = \int \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{f}_n, \beta) p(\mathbf{F}|\mathbf{Z}, \theta, \sigma) d\mathbf{F}.$$

Herein, the noise model is identical to that of probabilistic PCA, i.e., the noise model is assumed to be an isotropic Gaussian with precision β:

$$p(\mathbf{x}_n|\mathbf{f}_n, \beta) = \prod_{i=1}^{D} \mathcal{N}(x_n^{(i)}|f_n^{(i)}, \beta^{-1}),$$

where $x_n^{(i)}$ is the value of the i-th dimension of $\mathbf{x}_n$. It can be shown (Williams, 1997) that for each new point in the latent space $\mathbf{z}_{N+1}$, its counterpart in the data space is Gaussian distributed:

$$p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}, \mathbf{X}, \mathbf{Z}) =$$
$$\mathcal{N}(\mathbf{x}_{N+1}|\mathbf{X}^T\mathbf{K}^{-1}\mathbf{k}, \kappa(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}),$$

where $\mathbf{k}$ represents a column vector with entries $\kappa(\mathbf{x}_{N+1}, \mathbf{x}_{\cdot})$. Note that this out-of-sample extension is identical to that of Gaussian process regression, see, e.g., (Bishop, 2006).

Maximum likelihood learning in the GPLVM can be performed by maximizing the logarithm of

Equation 2 with respect to the parameters $\mathbf{Z}$. The log-likelihood function L is given by:

$$L = -\frac{DN}{2}\log(2\pi) - \frac{D}{2}\log|\mathbf{K}| - \frac{1}{2}\mathbf{X}\mathbf{K}^{-1}\mathbf{X}^T, \tag{3}$$

where we assumed that the elements k_{ij} of the matrix $\mathbf{K}$ are given by $k_{ij} = \kappa(\mathbf{z}_i, \mathbf{z}_j) + \beta^{-1}\delta_{ij}$, in which δ_{ij} represents the Dirac delta function. The log-likelihood function can be optimized with respect to $\mathbf{Z}$ using standard optimization techniques such as conjugate gradients. The log-likelihood could also be optimized with respect to the precision of the noise model β, but for simplicity, we opt to treat β as a hyperparameter.

The main advantage of the GPLVM over probabilistic PCA is that it allows for the use of non-linear covariance functions, i.e., that it can represent non-linear functions from the latent space to the data space. The probabilistic nature of the GPLVM also gives it advantages over Kernel PCA (the non-linear variant of 'normal' PCA), e.g., it provides a principled way to deal with missing data values. Please note that the GPLVM is *not* a probabilistic version of Kernel PCA: in Kernel PCA, the kernel function is defined over the data space, whereas in the GPLVM, the covariance function is defined over the latent space.

The main disadvantage of the GPLVM is that (as in probabilistic PCA) there is no guarantee that the local structure of the data is retained in the latent space (Lawrence and Candela, 2006). Instead, the GPLVM focuses on constructing a smooth mapping from the latent to the data space. In order to facilitate the successful construction of such a smooth mapping, the GPLVM only has to make sure that dissimilar datapoints are far apart in the latent space: if the global data structure would not be modelled correctly, this would lead to discontinuities in the mapping. Hence, the GPLVM mainly aims to model the global structure of the data correctly. The focus of the GPLVM on preserving global data structure conflicts with the popular belief that retaining local data structure is much more important in dimensionality reduction (Tenenbaum et al., 2000; Roweis and Saul, 2000; van der Maaten and Hinton, 2008). In general, data can be thought of as lying on one or more low-dimensional manifolds that are embedded in the high-dimensional space. Using artificial manifolds such as the 'Swiss roll' dataset (Tenenbaum et al., 2000), it can easily be demonstrated that large pairwise distances are of small relevance to typical learning tasks.

4 Preserving Local Structure

Above, we introduced the GPLVM and discussed some of its merits. Also, we discussed the main weakness of the GPLVM: it mainly focuses on preserving global data structure. In this section, we present an extension of the GPLVM that aims to preserve more of the local data structure in the latent space.

It is possible to add additional terms to the likelihood function of the GPLVM by designing a suitable prior distribution $p(\mathbf{Z})$ over the parameters of the GPLVM. This prior distribution can be used to provide additional (soft) constraints on the data representation in the latent space $\mathbf{Z}$. For instance, Urtasun *et al.* (2008) use a prior that is based on the LLE cost function (Roweis and Saul, 2000) to constrain the topology of the data representation in the latent space. The main drawback of this approach is that the LLE cost function is hampered by the presence of a trivial solution (viz. $\mathbf{Z} = 0$). The trivial solution is not selected because of a constraint on the covariance of the solution, but the optimization can easily cheat on this constraint (van der Maaten and Hinton, 2008).

We propose to use a prior that is based on the recently proposed t-Distributed Stochastic Neighbor Embedding (t-SNE). In t-SNE, the pairwise affinities p_{nm} between all pairs of points $(\mathbf{x}_n, \mathbf{x}_m)$ in the data space are measured using a Gaussian kernel, which is renormalized in order to obtain probabilities that reflect the similarity between the datapoints. Subsequently, a similar kernel is defined to measure the pairwise affinities q_{nm} between all pairs of points $(\mathbf{z}_n, \mathbf{z}_m)$ in the latent space, but now densities under a Student-t distribution are measured (and renormalized) to obtain the probabilities (van der Maaten and Hinton, 2008). Mathematically, the pairwise affinities are given by:

$$p_{nm} = \frac{\exp(-\|\mathbf{x}_n - \mathbf{x}_m\|/2s^2)}{\sum_{n' \neq m'} \exp(-\|\mathbf{x}_{n'} - \mathbf{x}_{m'}\|/2s^2)},$$

$$q_{nm} = \frac{(1 + \|\mathbf{z}_n - \mathbf{z}_m\|)^2}{\sum_{n' \neq m'} (1 + \|\mathbf{z}_{n'} - \mathbf{z}_{m'}\|)^2},$$

where p_{nm} and q_{nm} represent the probability picking the pair of points $(\mathbf{x}_n, \mathbf{x}_m)$ and $(\mathbf{z}_n, \mathbf{z}_m)$, respectively, from the set of all pairs of points. The parameter s is set automatically according to an information-theoretic heuristic (see (van der Maaten and Hinton, 2008) for details).

The key idea behind t-SNE is to use a different distribution to measure pairwise affinities in the latent space than in the data space. The use of the Student-t distribution in the latent space corrects for the difference in the volume of the high-dimensional data space and the low-dimensional latent space (note that the volume of a space grows exponentially with its dimensionality). In t-SNE, the locations of the points in the latent space $\mathbf{Z}$ are obtained by arranging them in such a way as to minimize the Kullback-Leibler divergence between the probabilities in the data space and the probabilities in the latent space. Mathematically, t-SNE minimizes the cost function:

$$C = \sum_{n \neq m} p_{nm} \log \frac{p_{nm}}{q_{nm}}.$$

The asymmetric nature of the Kullback-Leibler divergence causes the cost function to focus on appropriately modeling the large values of p_{nm}, i.e., on appropriately modeling the local structure of the data.

In our extended GPLVM, we incorporate the t-SNE cost function into the model by defining a data-dependent 'prior' distribution $p(\mathbf{Z})$ that takes the form of a Boltzmann distribution in which the t-SNE cost function is used as the energy function. Mathematically, we define:

$$p(\mathbf{Z}) \propto \exp\left(-\frac{1}{\gamma} \sum_{n \neq m} p_{nm} \log \frac{p_{nm}}{q_{nm}}\right),$$

where we omitted the normalization constant, and where γ represents a scaling parameter that was set to 10^{-7} in all our experiments. Note that the distribution $p(\mathbf{Z})$ depends on the data $\mathbf{X}$, as a result of which it is not technically a prior. The main aim of the prior distribution is to encourage solutions $\mathbf{Z}$ in which the divergence between the similarities in the data and the latent space is small, i.e., in which the local structure of the data is appropriately modeled in the latent space.

The introduction of the prior distribution gives rise to an additional term in the log-likelihood function

$$L = -\frac{D}{2} \log |\mathbf{K}| - \frac{1}{2} \mathbf{X}\mathbf{K}^{-1}\mathbf{X}^T - \\ \frac{1}{\gamma} \sum_{n \neq m} p_{nm} \log \frac{p_{nm}}{q_{nm}} + \text{const},$$

where const comprises terms that are not affected by changes in the model parameters $\mathbf{Z}$.

5 Experiments

In order to evaluate the performance of the extended GPLVM, we performed experiments on three datasets. The three datasets are briefly described in 5.1. The setup of our experiments is described in 5.2. The results of the experiments are described in 5.3.

5.1 Datasets

We performed experiments on three datasets: (1) the MNIST dataset, (2) the 20 newsgroups dataset, and (3) the Olivetti dataset. The MNIST dataset contains $70,000$ images of handwritten digit images of size 28×28 pixels. We used $2,500$ randomly selected images in our experiments. The 20 newsgroups dataset consists of 100-dimensional binary word-occurence features of $16,242$ documents, of which we randomly selected $2,500$ documents for our experiments. Each document is labeled according to the newsgroup it was extracted from. The Olivetti dataset contains 400 images of 40 individuals (10 images per individual). The face images have size 92×112 pixels.

5.2 Experimental setup

We compare the performance of the standard GPLVM and the extended GPLVM by measuring the log-likelihood of the training data under the trained models. Ideally, we would like to measure the likelihood (or reconstruction error) of test points under the model, but the likelihood of new datapoints under a GPLVM cannot be computed without resorting to approximations (we discuss this issue in more detail in Section 6). Next to the evaluation of log-likelihoods, we also evaluate the nearest-neighbor errors of the data representations in the latent space. In other words, we measure the percentage of points that have a point with a different class label as nearest neighbor in the latent space. In addition, we present two-dimensional scatter plots that visualize the data representations learned by the model.

In all experiments, we set $\theta = \beta^{-1} = \exp(-1)$, $\gamma = 10^{-7}$, and $d = 2$. In preliminary experiments, we determined a range of appropriate values for the bandwidth σ of the covariance function. We only present results for the best setting of σ within this range. Note that the appropriate value of σ depends, among others, on the *scale* of the latent space. As the value of the t-SNE cost function depends on the scale of the latent space, the

Dataset	Norm. GPLVM	Ext. GPLVM
MNIST	$-1.8582 \cdot 10^6$	$-1.8499 \cdot 10^6$
Newsgroups	$-2.3174 \cdot 10^5$	$-2.3088 \cdot 10^5$
Olivetti	$-3.6496 \cdot 10^6$	$-3.6598 \cdot 10^6$

Table 1: Best GPLVM log-likelihood of the normal and extended GPLVMs on the three datasets.

Dataset	Norm. GPLVM	Ext. GPLVM
MNIST	62.40%	5.92%
Newsgroups	39.40%	36.12%
Olivetti	61.00%	1.50%

Table 2: Nearest neighbor error of the normal and extended GPLVMs on the three datasets.

appropriate values of σ for the normal GPLVM and the extended GPLVM are typically not equal.

The initialization of the normal GPLVM is performed using PCA, as proposed by Lawrence (2005). The extended GPLVM is initialized using t-SNE. The optimization is performed using conjugate gradients[1].

5.3 Results

In Table 1, we present the best GPLVM log-likelihood (computed using Equation 3) for each experiment across the entire range of values for σ. The results presented in the table reveal that the extended GPLVM and the normal GPLVM perform on par in terms of the GPLVM log-likelihood of the training data.

In Table 2, the nearest neighbor errors of the best latent space representations constructed by the GPLVMs are shown for the three datasets. The nearest neighbor error is defined as the fraction of points that have a point with a different class label as their nearest neighbor. The results reveal the strong performance of the extended GPLVM model in terms of nearest-neighbor error in the latent space: for all datasets, significant improvements in terms of nearest neighbor errors are obtained.

Figure 2 shows two scatter plots of the latent space that correspond to the best solutions of both models on the MNIST dataset. The scatter plots reveal that the extended GPLVM is much better capable of revealing the structure of the dataset, because of its focus on local structure preservation.

6 Discussion

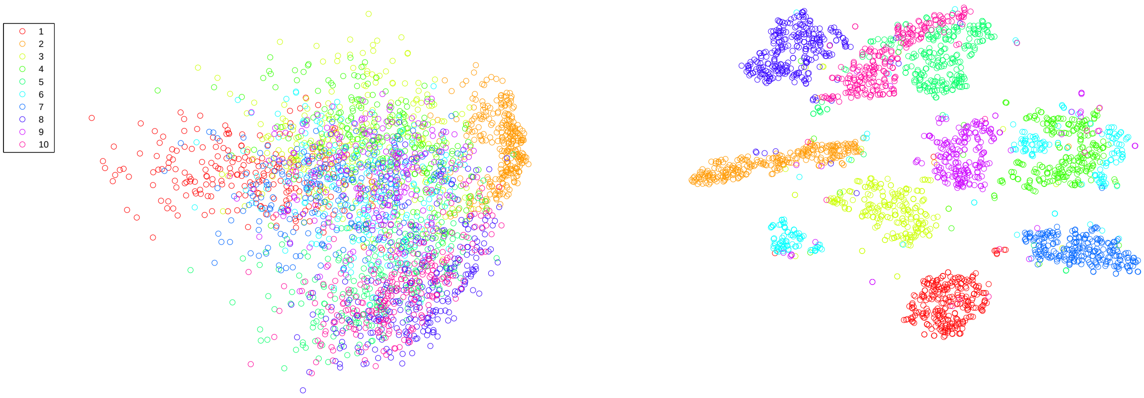
In the previous sections, we presented our extended GPLVM that aims at preservation of local data structure in the latent space. We presented experimental results revealing the strong performance of the new model. In this section, we discuss some issues regarding the normal and the extended GPLVM.

An important problem of the GPLVM is that it does not allow for the evaluation of the likelihood $p(\mathbf{x}_{N+1}|\mathbf{X}, \mathbf{Z}) \propto \int p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}, \mathbf{X}, \mathbf{Z})p(\mathbf{z}_{N+1})d\mathbf{z}_{N+1}$ of an unseen test point $\mathbf{x}_{N+1}$ under the model. The GPLVM allows for the computation of $p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}, \mathbf{X}, \mathbf{Z})$, but it is not possible to integrate out the latent variable $\mathbf{z}_{N+1}$ without resorting to approximation methods.

The inability to compute the (log)likelihood of unseen test data is problematic, because it prohibits evaluation of the generalization performance of the GPLVM. The strong performance in terms of log-likelihood of the training data may thus be due to overfitting. A simple approach to determine whether a GPLVM overfits on the training data is by drawing samples from the model. We performed an experiment in which we trained a one-dimensional extended GPLVM on a dataset of $1,965$ images of the face of a single individual. All images have size 20×28 pixels (Roweis et al., 2001). We sampled 64 uniformly spaced points from the latent space, and computed the distribution over the data space for each of these points. In Figure 6, we visualize the mean of each of these distributions. The figure shows that sampling from the extended GPLVM leads to natural 'fantasy' faces, as a result of which it is unlikely that the model severely overfits the training data.

An important issue in the training of GPLVMs that we did not discuss until now is the question how to initialize the parameters of the model. This issue is addressed by Geiger *et al.* (2008) who compared various initialization approaches. Geiger *et al.* (2008) propose an approach that initializes the parameters of the model as the original datapoints (i.e., the latent space has the same dimensionality as the data space). Subsequently, a prior is placed over the rank of the solution that promotes the identification of low-dimensional latent data representations. The advantage of this approach is that it provides a natural way to esti-

[1] Specifically, we used C. Rasmussen's `minimize.m` function.



(a) Normal GPLVM.

(b) Extended GPLVM.

Figure 2: Latent space representation of the MNIST dataset for the normal and the extended GPLVM.

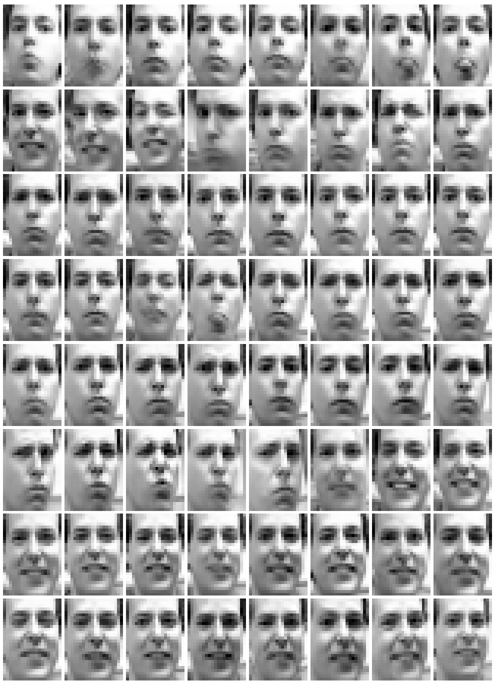


Figure 3: Fantasy faces sampled from a one-dimensional extended GPLVM.

mate the intrinsic dimensionality of the dataset. The disadvantage of this approach is that it leads to significant additional computational costs due to the high dimensionality of the latent space and the computation of the rank prior (and its gradient), which may prohibit practical applications of the approach. The extended GPLVM provides a much simpler approach to the initialization of the model parameters: minimize the energy function of the prior $p(\mathbf{Z})$ and use the result as initial model parameters. In contrast to the approach presented by Geiger *et al.* (2008), this does not lead to significant additional computational costs in the GPLVM optimization.

An alternative way to view the extended GPLVM is as an extension of t-SNE that provides it with a probabilistic mapping from the latent space to the data space (not to be confused with variants of t-SNE that learn a mapping from the data space to the latent space as presented by, e.g., van der Maaten (2009)). The extended GPLVM could be trained on, e.g., motion-capture data, and the resulting model could be used to generate realistic human movement animations by drawing samples from the model in the same way that the fantasy faces in Figure 6 are drawn. Applications of GPLVM models on motion capture data are discussed in more detail by, e.g., Urtasun *et al.* (2008).

7 Conclusions

We proposed an extension of the GPLVM that encourages the model to preserve local data structure in the latent space. The extension entails the introduction of a prior distribution on the parameters of the GPLVM that incorporates the t-SNE objective function. Our experimental evaluations reveal that the extension leads to significantly better models of the data. Code to reproduce the experimental results presented in this paper is available from `http://ticc.uvt.nl/~lvdrmaaten/tsne`.

Future work focuses on extending the model with a parametric mapping between the data space and the latent space in order to obtain a bijective mapping between the data space and the latent space. This amounts to maximizing the log-likelihood with respect to the parameters of the mapping (Lawrence and Candela, 2006). The parametrization may be formed by, e.g., a deep network. It is likely that the best results can be obtained by pretraining the network using, e.g., a stack of RBMs (Hinton and Salakhutdinov, 2006) or denoising autoencoders (Erhan et al., 2009). An additional advantage of a GPLVM with a parametric mapping between the data space and the latent space is that it allows the computation of the likelihood of test data under the model.

We also plan on investigating (semi)supervised (data-dependent) priors over **Z**, for instance, by employing a linear combination of the t-SNE cost function and the NCA cost function (Goldberger et al., 2005) as an energy function in the prior. Such an approach may improve the results on datasets in which partial label information is available (Urtasun and Darrell, 2007). Moreover, we aim to investigate learning the covariance function of the Gaussian process as proposed by, e.g., Salakhutdinov and Hinton (2008).

Acknowledgements

Laurens van der Maaten is supported by NWO-CATCH, project RICH, under grant 640.002.401. The author thanks Geoffrey Hinton and Neil Lawrence for helpful discussions, and Stijn Vanderlooy and Eric Postma for their comments on early versions of this paper.

References

C.M. Bishop. 1999. Bayesian PCA. In *Advances in Neural Information Processing Systems*, volume 11, pages 382–388.

C.M. Bishop. 2006. *Pattern recognition and machine learning*. Springer, New York, NY.

D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent. 2009. The difficulty of training deep architectures and the effect of unsupervised pre-training. In *Proceedings of AI-STATS, JMLR: W&CP 5*, pages 153–160.

A. Geiger, R. Urtasun, T. Darrell, and R. Stiefelhagen. 2008. Rank priors for continuous non-linear dimensionality reduction. Technical Report MIT-CSAIL-TR-2008-056, CSAIL, MIT, Cambridge, MA.

J. Goldberger, S. Roweis, G.E. Hinton, and R.R. Salakhutdinov. 2005. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*, volume 17, pages 513–520.

G.E. Hinton and R.R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

N.D. Lawrence and J. Quiñonero Candela. 2006. Local distance preservation in the GP-LVM through back constraints. In *Proceedings of the International Conference on Machine Learning*, pages 513–520.

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

J.A. Lee and M. Verleysen. 2007. *Nonlinear dimensionality reduction*. Springer, New York, NY.

D.J.C. MacKay. 1998. Introduction to Gaussian processes. In C.M. Bishop, editor, *Neural Networks and Machine Learning*, pages 133–165. Springer.

S.T. Roweis and L.K. Saul. 2000. Nonlinear dimensionality reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326.

S.T. Roweis, L. Saul, and G. Hinton. 2001. Global coordination of local linear models. In *Advances in Neural Information Processing Systems*, volume 14, pages 889–896.

R.R. Salakhutdinov and G.E. Hinton. 2008. Using deep belief nets to learn covariance kernels for gaussian processes. In *Advances in Neural Information Processing Systems*, volume 20, pages 1249–1256.

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

M.E. Tipping and C.M. Bishop. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6(3):611–622.

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

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

L.J.P. van der Maaten and G.E. Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2431–2456.

L.J.P. van der Maaten. 2009. Learning a parametric mapping by retaining local structure. In *Proceedings of AI-STATS, JMLR: W&CP 5*, pages 384–391.

C.K.I. Williams. 1997. Computing with infinite networks. In *Advances in Neural Information Processing Systems*, volume 9, pages 295–301.