The Gaussian Process Latent Variable Model

Neil D. Lawrence

27th January 2006

Abstract

The Gaussian process latent variable model (GP-LVM) is a recently proposed probabilistic approach to obtaining a reduced dimension representation of a data set. In this tutorial we motivate and describe the GP-LVM, giving reviews of the model itself and some of the concepts behind it.

1 Introduction

The Gaussian process latent variable model (GP-LVM) is a powerful approach to probabilistic non-linear dimensionality reduction. It was inspired by, and is related to, a class of probabilistic dimensionality reduction techniques known as latent variable models. In this tutorial we will review in detail a *linear* dimensionality reduction technique known as probabilistic PCA [Tipping and Bishop, 1999]. As we shall see, by taking an alternative view point of the latent variable model behind PCA we can develop a novel, alternative, interpretation of probabilistic PCA. One that, as it turns out, will lend itself to an elegant non-linearisation through Gaussian processes. However before discussing the resulting model in detail we will conduct a brief review of Gaussian processes, discussing what it means to have a prior over functions and what Gaussian distributed functions can look like.

Finally we shall round off by discussing the characteristics of the GP-LVM and by mentioning some extensions of the model.

In the notes we will make use of examples that can be recreated through code downloaded from `http://www.dcs.shef.ac.uk/~neil/fgplvm`. An additional package of demonstrations associated with this presentation has been placed at `http://www.dcs.shef.ac.uk/~neil/oxford`.

2 Motivation

Many data sets we deal with are high dimensional. The 'curse of dimensionality' implies that to correctly understand the structure of a high dimensional data set we need many data points, exponentially many in the number of dimensions. However, in practice we find that we often do very well with much smaller data sets than we might expect to need. One possible reason for this is that many data sets of interest, while seemingly high dimensional, have an intrinsic dimensionality which is much lower. Let us consider the example of handwritten digits.

In Figure 1 we show a hand-written 6 taken from the USPS Cedar CD-ROM handwritten digits training set. The data point is 3,648 dimensional as it is printed in a 64 pixel by 57 pixel image. However, if our data is based on a few simple transformations of this digit, it may not span all 3,648 dimensions of the space. To see this consider Figure 2. Here we have created a data set by rotating the original digit 360 times, each time by one degree. The data is then projected onto its second and third principal components. The resulting projection clearly shows a circular shape. There is some noise (presumably associated with the nearest neighbour interpolation used in the rotation of the image) but the structure of the space is clear. Further examination of the principal components (which is possible with the software on line) also reveals that the dataset is inherently one dimensional.

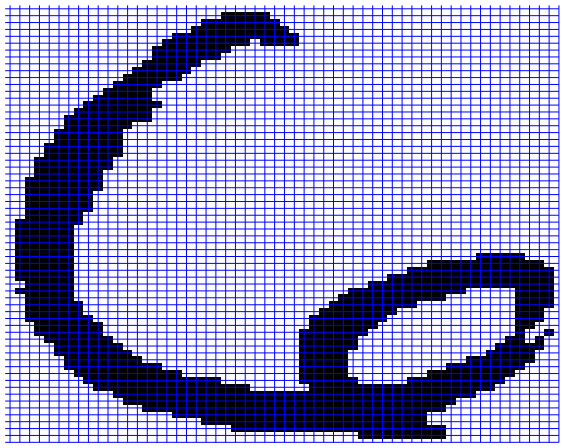
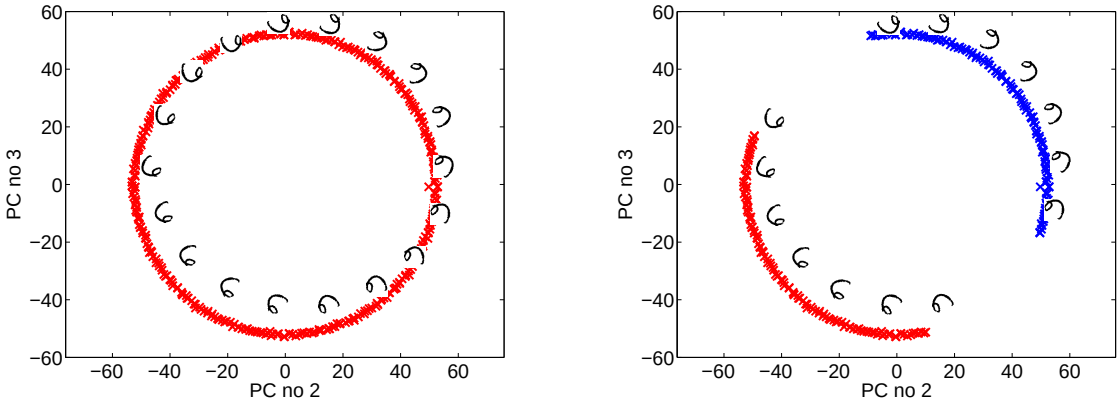


Figure 1: Digit 6 from the USPS Cedar CD-ROM. The digit is 64 pixels by 57 pixels giving it 3,648 dimensions.



(a)

Figure 2: Rotation of handwritten 6. A data set is generated by rotating the original image 360 times (`prepDemManifold`). The data set is then visualised by projecting into the second and third principal component. In (a) the full rotation is visualised (`demManifoldPrint([2 3], 'all')`), in (b) some rotations are assumed to be associated with the digit 6 and others from the digit 9 (`demManifoldPrint([2 3], 'sixnine')`).

In practice of course real data sets will not be generated by a simple rotation of a one dimensional space. However, it seems reasonable to assume that a data set might consist of a fixed number of 'prototypes' which undergo a limited number of transformations and then are, perhaps, corrupted by some noise. If this is the case, then it makes sense to model high dimensional data by seeking a low dimensional representation. In statistics the standard approach to this problem is multi-dimensional scaling (MDS, see *e.g.* Mardia et al. [1979]). More recently in machine learning several spectral approaches have been proposed [Tenenbaum et al., 2000, Roweis and Saul, 2000, Weinberger et al., 2004] some of which may be seen as classical MDS with a particular approach to learning a distance matrix. We wish to focus on probabilistically inspired approaches. Having an algorithm with a probabilistic interpretation allows the algorithm to be extended in a logical manner and eases integration of the approach in a larger system. All currently published extensions and applications of the GP-LVM take advantage of its probabilistic interpretation in one form or another [Grochow et al., 2004, Urtasun et al., 2005, Wang et al., 2006, Shon et al., 2006].

In the next section we discuss, perhaps, the simplest latent variable model that can be used for dimensional reduction, probabilistic PCA. The model is fundamentally linear, but it illustrates the basic concepts behind latent variable models. We will then briefly review Gaussian processes (in Section 4) after which we will introduce the fundamental re-thinking of the latent variable model behind PCA that enables dual probabilistic PCA and leads to the GP-LVM (Section 5). We will then show various results achieved with the GP-LVM and briefly mention some enhancements.

3 Probabilistic PCA

Probabilistic PCA is a simple latent variable model where the latent space, $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^{\mathrm{T}}$ is assumed to be related to the *centred data set*, $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]^{\mathrm{T}}$ through a linear mapping that is corrupted by noise,

$$\mathbf{y}_n = \mathbf{W}\mathbf{x}_n + \boldsymbol{\eta}_n,$$

where the mapping is given by $\mathbf{W} \in \Re^{D\times q}$ with D the dimension of the data space and q the dimension of the latent space and $\boldsymbol{\eta}_n$ is a vector of noise terms. For the particular case of probabilistic PCA, the noise is taken to be Gaussian distributed,

$$p\left(\boldsymbol{\eta}_n|\beta\right) = N\left(\boldsymbol{\eta}_n|\mathbf{0}, \beta^{-1}\mathbf{I}\right),$$

with a mean of zero and a spherical covariance given by $\beta^{-1}\mathbf{I}$. The parameter β is an inverse variance and is therefore referred to as a precision.

The conditional probability of the data given the latent space can be written as

$$p\left(\mathbf{y}_n|\mathbf{x}_n, \mathbf{W}, \beta\right) = N\left(\mathbf{y}_n|\mathbf{W}\mathbf{x}_n, \beta^{-1}\mathbf{I}\right),$$

and assuming independence across data points we have

$$p\left(\mathbf{Y}|\mathbf{X}, \mathbf{W}, \beta\right) = \prod_{n=1}^{N} N\left(\mathbf{y}_n|\mathbf{W}\mathbf{x}_n, \beta^{-1}\mathbf{I}\right). \tag{1}$$

Following the Bayesian nomenclature, in anticipation of a prior distribution over the latent space, this term can be seen as the *likelihood* of the data $\mathbf{Y}$ given $\mathbf{X}$. We note in passing that it is also the likelihood associated with a least-squares multi-variate regression: if we are given $\mathbf{X}$ and maximise the likelihood with respect to $\mathbf{W}$ we recover

$$\hat{\mathbf{W}}\mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{Y}^{\mathrm{T}}\mathbf{X},$$

which may be solved for $\hat{\mathbf{W}}$ to obtain the least squares regression estimate of $\mathbf{W}$. In probabilistic PCA however, the values of $\mathbf{X}$ aren't given, they are *nuisance parameters*. The standard approach when dealing with these parameters is to consider a *prior distribution* over the latent space, $p\left(\mathbf{X}\right)$, and seek to marginalise the values of $\mathbf{X}$.

3.1 Gaussian Prior

The choice of prior distribution will clearly have an effect on the optimum value of $\mathbf{W}$. If the latent distributions are chosen to be independent across q and non-Gaussian, the latent variable model behind independent component analysis [Bell and Sejnowski, 1995, MacKay, 1996] is recovered in the limit as $\beta \to \infty$. It has also long been known that if the latent distribution is chosen to be Gaussian then PCA is recovered in the limit as $\beta \to \infty$ (this observation inspired sensible PCA [Roweis, 1998]). More interestingly Tipping and Bishop [1999] showed that if the latent distribution is taken to be Gaussian then the maximum likelihood solution for $\mathbf{W}$ can span the principal subspace of the data even when β is finite. The form of the Gaussian prior is chosen by convention[1] to be zero mean and unit covariance,

$$p(\mathbf{X}) = \prod_{n=1}^{N} p(\mathbf{x}_n) = \prod_{n=1}^{N} N(\mathbf{x}_n|\mathbf{0}, \mathbf{I}). \tag{2}$$

The marginal likelihood can then be computed as follows

$$\begin{aligned} p(\mathbf{Y}|\mathbf{W}, \beta) &= \prod_{n=1}^{N} \int N\left(\mathbf{y}_n|\mathbf{W}\mathbf{x}_n, \beta^{-1}\mathbf{I}\right) N(\mathbf{x}_n|\mathbf{0}, \mathbf{I})\, d\mathbf{x}_n & (3)\\ &\propto \prod_{n=1}^{N} \int \exp\left(-\frac{1}{2}\left(\beta \mathbf{y}_n^{\mathrm{T}}\mathbf{y}_n - 2\beta \mathbf{y}_n^{\mathrm{T}}\mathbf{W}\mathbf{x}_n + \mathbf{x}_n^{\mathrm{T}}\left(\beta \mathbf{W}^{\mathrm{T}}\mathbf{W} + \mathbf{I}\right)\mathbf{x}_n\right)\right) d\mathbf{x}_n & (4)\\ &\propto \prod_{n=1}^{N} \exp\left(-\frac{1}{2}\left(\mathbf{y}_n^{\mathrm{T}}\left(\beta\mathbf{I} - \beta^2\mathbf{W}\left(\beta\mathbf{W}^{\mathrm{T}}\mathbf{W} + \mathbf{I}\right)^{-1}\mathbf{W}^{\mathrm{T}}\right)\mathbf{y}_n\right)\right) & (5) \end{aligned}$$

where the first line (3) is obtained through multiplying (1) and (2) to obtain the joint likelihood, and introducing the intergrad to marginalise $\mathbf{X}$. The second line (4) is obtained by expanding the squares, (5) is then obtained by standard integrals on Gaussians (see Appendix B on Gaussian integrals in Bishop [1995] for a proof). We can obtain the final solution through inspection of (5): the matrix associated with the quadratic term has the form of the matrix inversion lemma[2] and there are no linear terms in $\mathbf{y}_n$, implying that the solution is a product of zero mean Gaussians,

$$p(\mathbf{Y}|\mathbf{W}, \beta) = \prod_{n=1}^{N} N(\mathbf{y}_n|\mathbf{0}, \mathbf{C}), \tag{6}$$

where the covariance is given by $\mathbf{C} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \beta^{-1}\mathbf{I}$. This is immediately recognised as a reduced rank representation of the covariance. Since $\mathbf{W} \in \Re^{D\times q}$ the matrix $\mathbf{W}\mathbf{W}^{\mathrm{T}} \in \Re^{D\times D}$ will have rank of at most q. For finite β the term $\beta^{-1}\mathbf{I}$ then acts as a 'regulariser' to ensure that the resulting covariance has full rank and the distribution is thereby properly defined.

This model was suggested simultaneously by Roweis [1998], Tipping and Bishop [1999], but Tipping and Bishop [1999] also provided the proof that the maximum likelihood solution for $\mathbf{W}$ spans the principal sub-space of the data. The proof for the dual probabilistic PCA we introduce in Section 5 closely tracks the proof of Tipping and Bishop [1999] so we omit the details here, merely giving the result. The optimum value for $\mathbf{W}$ is given by

$$\hat{\mathbf{W}} = \mathbf{U}_q'\mathbf{L}\mathbf{V}^{\mathrm{T}}$$

where $\mathbf{U}_q'$ are the q eigenvectors of the covariance matrix $N^{-1}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$ associated with the q largest eigenvalues, $\{\lambda_i\}_{i=1}^{q}$ which may be obtained by solving

$$N^{-1}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\mathbf{U}' = \mathbf{U}'\Lambda. \tag{7}$$

[1] Using non-zero mean and non-unit covariance merely leads to a redundant parameterisation of the model. However this redundant parameterisation can be exploited in certain circumstances to give faster converging algorithms [Sanguinetti et al., 2005].

[2] In its most general form the matrix inversion lemma is $(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}\mathbf{D}\mathbf{A}^{-1}$.

The matrix $\mathbf{L}$ is diagonal and its ith diagonal element is given by $l_i = \left(\lambda_i - \beta^{-1}\right)^{\frac{1}{2}}$.

The principal components of a data set are the eigenvectors of the covariance matrix, and the principal sub-space is the space spanned by those eigenvectors. We therefore see that the solution for probabilistic PCA spans the q-dimensional principal sub-space of the data.

We will revisit probabilistic principal component analysis in Section 5 when we discuss the Gaussian process latent variable model. First we will briefly review Gaussian processes.

4 Gaussian Processes

Gaussian processes [O'Hagan, 1978, 1992, Williams and Rasmussen, 1996, Williams, 1998, MacKay, 1998, Rasmussen and Williams, 2006] are probability distributions over functions. We can combine a Gaussian process prior with a likelihood (or noise model) to obtain a posterior over functions. If the likelihood is also Gaussian the form of the posterior will also be a Gaussian process. In practice the likelihood is often non-Gaussian but even in this case we typically approximate the posterior process with a Gaussian process.

4.1 A Prior Over Functions

A distribution over functions is seemingly non-sensical as functions are infinite dimensional objects. However, let us proceed by considering a finite Gaussian distribution over some values instantiated from a function $\mathbf{f} = \{f_n\}_{n=1}^N \in \Re^{N\times 1}$. If we assume that these values are drawn from a Gaussian distribution with mean zero and covariance $\mathbf{K}$, then we can write

$$\begin{aligned} p\left(\mathbf{f}|\mathbf{K}\right) &= N\left(\mathbf{f}|\mathbf{0},\mathbf{K}\right) \\ &= \frac{1}{\left(2\pi\right)^{\frac{N}{2}}\left|\mathbf{K}\right|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}\mathbf{f}\mathbf{K}^{-1}\mathbf{f}\right). \end{aligned}$$

To illustrate the form of this function we now consider a particular covariance matrix. We will take *one sample* from a Gaussian with this covariance matrix. Within this single sample there will be $N = 25$ instantiations.

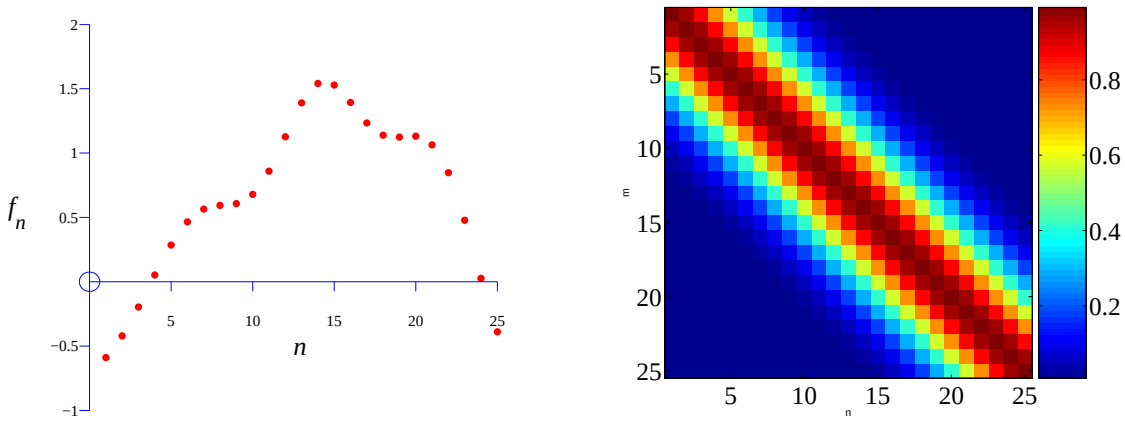
The covariance matrix we used is shown as a greyscale image in Figure 3(b). Note that the covariance function shows correlation between points f_m and f_n if n is near to m. There is less correlation if n is distant from m. The sample from the Gaussian is plotted in Figure 3(a). Note that points that have nearby indices have similar f_n . If the plot is seen as a function of n the function appears smooth. This smoothness comes from the fact that nearby points are correlated in the covariance.

In practice the covariance will not be a function of the indices, but of an input space $\mathbf{X}$. However, each point in that input space, $\mathbf{x}_n$, will also be indexed by n so for the moment it is convenient to ignore this relationship.

To see how it is possible to make predictions given the covariance matrix, let us first consider the covariance of two points. Marginalising the remaining points leads to a two dimensional Gaussian whose covariance is made up of the rows and columns from the original covariance associated with those points. This allows us to plot a contour and visualise the joint probability over these points. First we take the points indexed as f_1 and f_2. A contour of the joint probability distribution over this space is shown in Figure 4(a). Also, in Figure 4(c) we visualise the conditional distribution for $p\left(f_2|f_1,\mathbf{K}\right)$. This can be viewed as the predictive distribution for f_2 having observed f_1. The strong correlation induced by the covariance, $\mathbf{K}$, means that the conditional distribution for f_2 has a mean that is close to f_1.

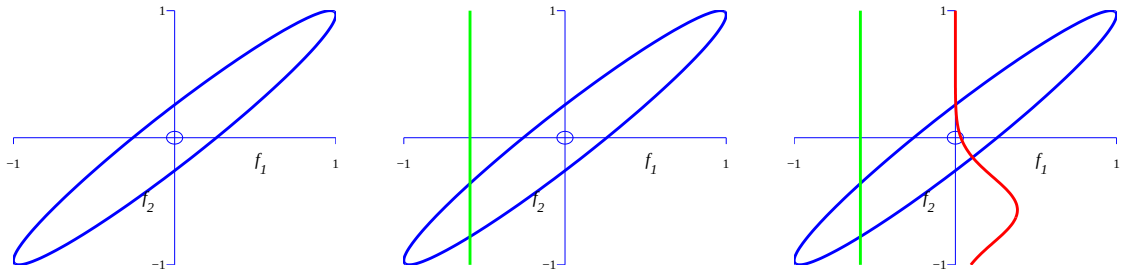
A similar plot is shown in Figure 5 but this time for the joint distribution between f_1 and f_5. The correlation induced by the covariance function is now much weaker, the conditional distribution for f_5 has a mean much closer to zero than that for f_1 had.

The obvious question is, where does this covariance matrix come from? In this case the covariance matrix is built using the inputs to the function $\mathbf{x}_n$. The covariance shown in Figure (b)



(a) (b)

Figure 3: In (a) we show 25 instantiations of a function, f_n, as sampled from a zero mean Gaussian with the covariance matrix given in (b). In (b) we show the covariance matrix as a greyscale plot. Each element square in the plot gives the covariance between two points of the function f_n and f_m. The plots can be recreated with the command `demGPSample`.



(a) (b) (c)

Figure 4: Joint distribution between the values of f_1 and f_2: (a) shows the a single contour (one standard deviation from the mean) of the Gaussian distribution; (b) shows the instantiated value of f_1 as a line dashed in the plot and (c) shows the conditional distribution of $p(f_2|f_1)$ as a dotted line rotated to be a function of the f_2-axis of the plot. These plots can be recreated through the script `demGPCov2D([1 2])`. The portion of the covariance function as computed between these two points is given by $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

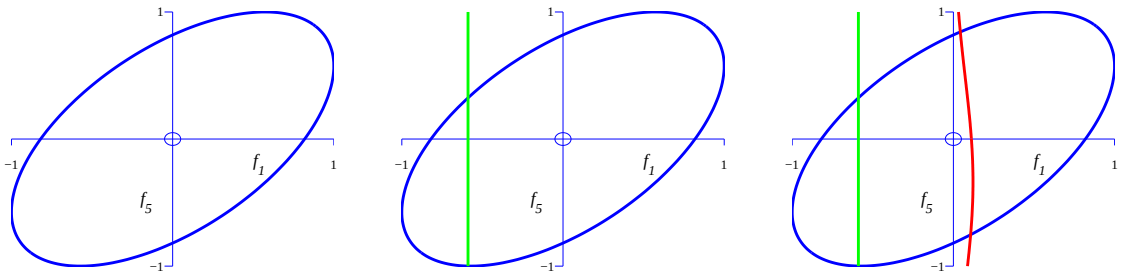


Figure 5: Joint distribution between the values of f_1 and f_5: (a) shows the a single contour (one standard deviation from the mean) of the Gaussian distribution; (b) shows the instantiated value of f_1 as a line dashed in the plot and (c) shows the conditional distribution of $p(f_5|f_1)$ as a dotted line rotated to be a function of the f_5-axis of the plot. These plots can be recreated through the script `demGPCov2D([1 5])`. The portion of the covariance function as computed by these two points is given by $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

is based on Euclidean distance between the points. The input points used were one dimensional and equally spaced along a line between -1 and 1. The covariance between points m and n was given by

$$k(\mathbf{x}_m, \mathbf{x}_n) = \exp\left(-\frac{\gamma}{2}(\mathbf{x}_m - \mathbf{x}_n)^{\mathrm{T}}(\mathbf{x}_m - \mathbf{x}_n)\right), \tag{8}$$

where the inverse width parameter γ was taken to be 10. Note that if $m = n$ then the variance of the point is 1. This is why the furthest extent of the contour at one standard deviation in each of Figures 4 and 5 is also one. This covariance function is known as the radial basis function (RBF), squared exponential, or Gaussian covariance function. We note that it shares the same form as the RBF kernel used in support vector machines [Schölkopf and Smola, 2001]. In fact the class of valid covariance functions is the same as the class of Mercer kernels. We will therefore use the terms covariance function and kernel interchangeably in what follows.

The covariance function provides the joint distribution over the instantiations of the functions. The conditional distribution provides predictions for as yet unseen locations given points at known locations. This is analogous to a training set/test set situation in machine learning. The predictions are locations are on the left hand side of the conditional, the training data is on the right hand side of the conditional, if we denote instantiations from the training set as $\mathbf{f}$ and positions in the test set as $\mathbf{f}_*$ we can denote this conditional as $p(\mathbf{f}_*|\mathbf{f})$. Since the joint distribution is Gaussian, we known this conditional distribution must also be Gaussian. To find the conditional distribution we make use of a partitioned version of the kernel matrix,

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{\mathbf{f},*} \\ \mathbf{K}_{*,\mathbf{f}} & \mathbf{K}_{*,*} \end{bmatrix}$$

where $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ is the covariance matrix for the training data points, $\mathbf{f}$, the sub-matrix $\mathbf{K}_{*,*}$ is the covariance matrix for the test data points, $\mathbf{f}_*$, and the sub-matrix $\mathbf{K}_{*,\mathbf{f}} = \mathbf{K}_{\mathbf{f},*}^{\mathrm{T}}$ is the cross correlations between training and test data. We are now in a position to write down the joint distribution of the data via the partition inverse,

$$\mathbf{K}^{-1} = \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} + \mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\Sigma^{-1}\mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} & -\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\Sigma^{-1} \\ -\Sigma^{-1}\mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} & \boldsymbol{\Sigma^{-1}} \end{bmatrix}$$

where

$$\Sigma = \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}.$$

Through the partitioned inverse we can re-express the joint distribution, for convenience we write it below as the logarithm of the joint distribution,

$$\begin{aligned}\log p(\mathbf{f}, \mathbf{f}_*) &= -\frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f} - \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\Sigma^{-1}\mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f} \\ &\quad + \mathbf{f}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\Sigma^{-1}\mathbf{f}_* - \frac{1}{2}\mathbf{f}_*^{\mathrm{T}}\Sigma^{-1}\mathbf{f}_* + \text{const}_1\end{aligned}$$

where the constant term contains portions that are not dependent on $\mathbf{f}$ or $\mathbf{f}_*$. Strictly speaking, the joint distribution is also conditioned on the parameters of the covariance function, the training input locations, $\mathbf{X}$, and the test input locations, $\mathbf{X}_*$. This dependence occurs through the kernel functions. However we are dropping this dependence in what follows to avoid cluttering the notation.

The conditional distribution is found by dividing joint distribution by the prior distribution on $\mathbf{f}$, $p(\mathbf{f}) = N(\mathbf{f}|\mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}})$. In log space this is equivalent to subtraction of

$$\log p(\mathbf{f}) = -\frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f} + \text{const}_2$$

giving

$$\begin{aligned}\log p(\mathbf{f}_*|\mathbf{f}) &= \log p(\mathbf{f}_*, \mathbf{f}) - \log p(\mathbf{f}) \\ &= -\frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\Sigma^{-1}\mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f} + \mathbf{f}^{\mathrm{T}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\Sigma\mathbf{f}_* \\ &\quad -\frac{1}{2}\mathbf{f}_*^{\mathrm{T}}\Sigma^{-1}\mathbf{f}_* + \text{const}_1 - \text{const}_2 \qquad (9) \\ &= -\frac{1}{2}\left(\mathbf{f}_* - \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f}\right)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\left(\mathbf{f}_* - \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f}\right) \\ &\quad +\text{const}_3 \qquad (10) \\ &= \log N\left(\mathbf{f}_*|\bar{\mathbf{f}}_*, \Sigma\right). \qquad (11)\end{aligned}$$

where $\bar{\mathbf{f}} = \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f}$, $\text{const}_3 = \text{const}_1 - \text{const}_2$ and (10) is derived from (9) by completing the square.

So we can see that if we observe points from the function, $\mathbf{f}$, directly for a given set of training data $\mathbf{X}$ then we can predict the locations of functions at as yet unseen locations whose inputs are given by $\mathbf{X}_*$. The resulting distribution is also a Gaussian process, but with a mean given by $\bar{\mathbf{f}}$ and a covariance given by Σ. In general though, we will not make direct observations of the function, our observations are more likely to be corrupted by noise. We therefore also define a noise model $p(\mathbf{y}|\mathbf{f})$ which relates our actual observations, $\mathbf{y}$, to the function $\mathbf{f}$ (see Figure **??**). A standard noise model for regression is independent Gaussian random noise. In this case we can write the noise model as

$$p(\mathbf{y}|\mathbf{f}) = \prod_{n=1}^{N} p(y_n|f_n) = \prod_{n=1}^{N} N\left(y_n|f_n, \beta^{-1}\right), \qquad (12)$$

i.e. we are assuming that the function becomes corrupted by the addition of independent Gaussian noise with a precision of β^{-1} at each observation. Given the Gaussian noise model in (12) computation of the marginal likelihood,

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})\, p(\mathbf{f})\, d\mathbf{f},$$

is straightforward,

$$p(\mathbf{y}) \propto \int \exp\left(-\frac{\beta}{2}(\mathbf{y} - \mathbf{f})^{\mathrm{T}}(\mathbf{y} - \mathbf{f}) - \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f}\right) d\mathbf{f}$$

$$\propto \int \exp\left(-\frac{\beta}{2}\mathbf{y}^{\mathrm{T}}\mathbf{y} - \frac{1}{2}\mathbf{f}^{\mathrm{T}}\left(\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} + \beta\mathbf{I}\right)\mathbf{f} + \beta\mathbf{y}^{\mathrm{T}}\mathbf{f}\right) d\mathbf{f} \tag{13}$$

$$\propto \exp\left(-\frac{1}{2}\mathbf{y}^{\mathrm{T}}\left(\beta\mathbf{I} - \beta^2\left(\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} + \beta\mathbf{I}\right)^{-1}\right)\mathbf{y}\right) \tag{14}$$

$$\propto \exp\left(-\frac{1}{2}\mathbf{y}^{\mathrm{T}}\left(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \beta^{-1}\mathbf{I}\right)^{-1}\mathbf{y}\right) \tag{15}$$

$$= N\left(\mathbf{y}|\mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}} + \beta^{-1}\mathbf{I}\right), \tag{16}$$

where the integral in (13) can again be undertaken through standard Gaussian results [Bishop, 1995, Appendix B] and we move from (14) to (15) through inspection by recognising the form of the matrix inversion lemma in (14). The resulting marginal likelihood is then a Gaussian process on $\mathbf{y}$ with a modified covariance function of the form $\hat{\mathbf{K}}_{\mathbf{y},\mathbf{y}} = \mathbf{K}_{\mathbf{f},\mathbf{f}} + \beta^{-1}\mathbf{I}$.

4.2 Summing Covariance Functions

As an aside we note that the form of $p(\mathbf{y}|\mathbf{f})$ can also be seen as a Gaussian process over $\mathbf{y}$ with a given mean $\mathbf{f}$ and a covariance function $\mathbf{K}_{\mathbf{y},\mathbf{y}} = \beta^{-1}\mathbf{I}$. The particular form of this covariance function is that all points are uncorrelated, *i.e.* the process is just white noise. However regardless of the form of the covariance function the result of the marginalisation above would remain the same,

$$N\left(\mathbf{y}|\mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}} + \mathbf{K}_{\mathbf{y},\mathbf{y}}\right) = \int N\left(\mathbf{y}|\mathbf{f}, \mathbf{K}_{\mathbf{y},\mathbf{y}}\right) N\left(\mathbf{f}|\mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}}\right) d\mathbf{f},$$

so we see that a new covariance function can be generated by adding two different covariance functions together. This has the interpretation of a hierarchical Gaussian process, where the mean of each process is itself treated as a Gaussian process.

4.3 Parameters of the Covariance Function

The covariance function we described in (8) has a parameter: the inverse width. We also saw from the contour plots of the correlation between the points, that the maximum standard deviation was unity. If we wish to have a covariance function that existed on a non unit scale we need to introduce a further parameter, α,

$$k\left(\mathbf{x}_m, \mathbf{x}_n\right) = \alpha \exp\left(-\frac{\gamma}{2}\left(\mathbf{x}_m - \mathbf{x}_n\right)^{\mathrm{T}}\left(\mathbf{x}_m - \mathbf{x}_n\right)\right), \tag{17}$$

which controls the variance of the function. Note that this parameter α is analogous to β^{-1} (which controls the variance of the white noise process). Here α is controlling the variance of the function generated by the RBF kernel. In the context of the marginal distribution over $\mathbf{y}$,

$$p\left(\mathbf{y}|\alpha, \beta, \gamma\right) = N\left(\mathbf{y}|\mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}} + \beta^{-1}\mathbf{I}\right), \tag{18}$$

where we have made explicit the dependence of the marginal likelihood on α, β and γ. This dependence occurs through $\mathbf{K}_{\mathbf{f},\mathbf{f}}$, the elements of which are given by (17), we can view $\sqrt{\alpha\beta}$ as a signal to noise ratio. The standard deviation of the signal is $\sqrt{\alpha}$ and the standard deviation of the noise is $\sqrt{\beta^{-1}}$. In many kernel methods, these parameters must be selected through cross validation. An advantage of the Gaussian process point of view is that they can be optimised by maximisation of the marginal likelihood $p\left(\mathbf{y}|\alpha, \beta, \gamma\right)$. This is known as empirical Bayes or type II maximum likelihood. Priors can also be placed over these parameters and sampling used to estimate their posteriors (see *e.g.*Williams and Rasmussen 1996).

4.4 Different Covariance Functions

By changing the characteristics of the covariance function we can sample different functions from the prior. For example, setting each element of the kernel matrix to an inner product between the

points,

$$k(\mathbf{x}_m, \mathbf{x}_n) = \alpha \mathbf{x}_m^{\mathrm{T}} \mathbf{x}_n,$$

produces functions that are linear. Note that this kernel function can also be written as

$$\mathbf{K}_{\mathbf{f},\mathbf{f}} = \mathbf{X}\mathbf{X}^{\mathrm{T}}.$$

Williams [1997] showed that a multi-layer perceptron with infinite hidden nodes has a covariance function of the form

$$k(\mathbf{x}_m, \mathbf{x}_n) = \alpha \sin^{-1}\left(\frac{w\mathbf{x}_m^{\mathrm{T}}\mathbf{x}_n + b}{\sqrt{w\mathbf{x}_m^{\mathrm{T}}\mathbf{x}_m + b + 1}\sqrt{w\mathbf{x}_n^{\mathrm{T}}\mathbf{x}_n + b + 1}}\right),$$

where a Gaussian prior over the weights from the input to hidden units is used with a variance w and a prior over the locations of the activation functions with variance b.

Finally a constant offset in the function can be accounted for by adding a kernel function which is constant in value.

$$k(\mathbf{x}_m, \mathbf{x}_n) = \alpha,$$

we will refer to this as the bias kernel. We show some examples of samples associated with tese covariance functions in Figure 6

4.5 Consistency

Gaussian processes are consistent in that the posterior predictions at each point remain the same regardless of the number and location of the test points. To see this we first consider an additional set of test points $\mathbf{f}_+$ which is disjoint from $\mathbf{f}_*$. The conditional probability of our original test points can be expressed as

$$p(\mathbf{f}_*|\mathbf{f}) = \int p(\mathbf{f}_*, \mathbf{f}_+|\mathbf{f})\, d\mathbf{f}_+,$$

for the system to be consistent this marginal likelihood must be the same regardless of $\mathbf{f}_+$. In other words, if we replaced $\mathbf{f}_+$ with $\hat{\mathbf{f}}_+$ we would require

$$p(\mathbf{f}_*|\mathbf{f}) = \int p(\mathbf{f}_*, \mathbf{f}_+|\mathbf{f})\, d\mathbf{f}_+ = \int p\left(\mathbf{f}_*, \hat{\mathbf{f}}_+|\mathbf{f}\right) d\hat{\mathbf{f}}_+$$

where $\hat{\mathbf{f}}_+ \neq \mathbf{f}_+$.

4.6 Summary

We have reviewed some of the salient points of Gaussian processes, in particular we have shown how a Gaussian process arises from the specification of a covariance function. Given a sub-set of observations of a function, and an associated covariance, we can make predictions about the the likely location of the function in regions where we hadn't previously observed data.

The parameters of the covariance function can be found through maximisation of the marginal likelihood (18).

5 The GP-LVM

The standard probabilistic interpretation of PCA we reviewed in Section 3 combines a Gaussian likelihood,

$$p(\mathbf{Y}|\mathbf{W}, \mathbf{X}, \beta) = \prod_{n=1}^{N} N\left(\mathbf{y}_n|\mathbf{W}\mathbf{x}_n, \beta^{-1}\mathbf{I}\right)$$

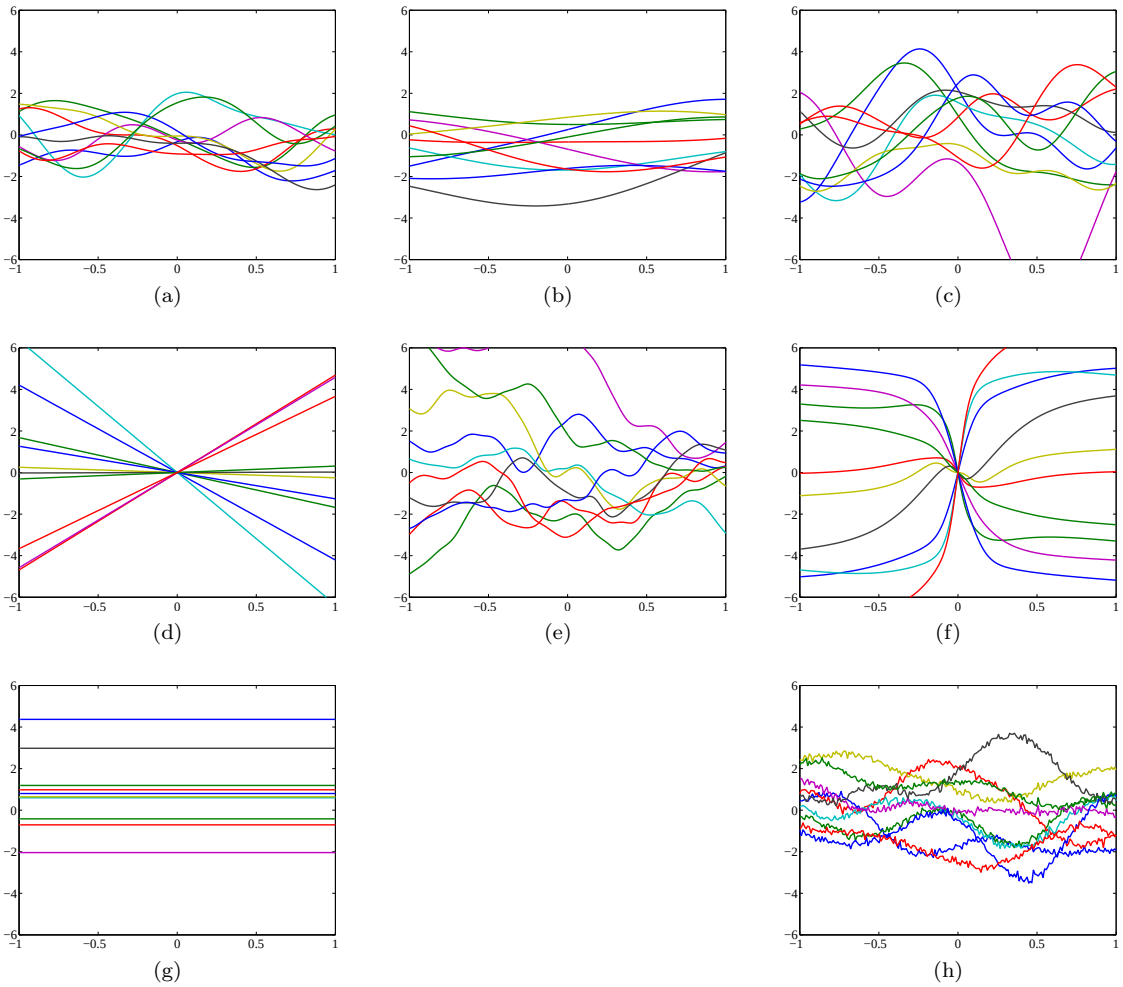


Figure 6: Samples from different covariance functions. (a) RBF kernel with $\gamma = 10$, $\alpha = 1$, (b) RBF kernel with $\gamma = 1$, $\alpha = 1$ (c) RBF kernel with $\gamma = 10$, $\alpha = 4$, (d) linear kernel with $\alpha = 16$, (e) MLP kernel with $\alpha = 8$, $w = 100$ and $b = 100$, (f) MLP kernel with $\alpha = 8$, $b = 0$ and $w = 100$, (g) bias kernel with $\alpha = 1$ and (h) Summed combination of: RBF kernel, $\alpha = 1$, $\gamma = 10$; bias kernel, α =1; and white noise kernel, $\beta = 100$. Samples can be recreated with the script `demCovFuncSample`.

Figure 7: Graphical representation of (a) the standard probabilistic PCA model and (b) its dual representation which also leads to a probabilistic interpretation of PCA. The nodes are shaded to represent different treatments. *Black* shaded nodes are optimised, *white* shaded nodes are marginalised and *grey* shaded nodes are observed variables.

with a Gaussian prior on the latent variables, $\mathbf{X}$. The GP-LVM takes a different perspective on the model. Rather than marginalising the latent variables, we seek to marginalise the mapping. Graphically, we can depict the two different approaches as shown in Figure 7.

As we shall see this approach will lead to a dual representation of probabilistic PCA. The required marginalisation now takes the form

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

By specifying a Gaussian prior distribution over the parameters of the mapping,

$$p\left(\mathbf{W}\right)=\prod_{i}N\left(\mathbf{w}_i|\mathbf{0},\mathbf{I}\right)$$

where $\mathbf{w}_i$ is the ith row of the matrix $\mathbf{W}$, and then integrating over $\mathbf{W}$ we obtain a marginalised likelihood for $\mathbf{Y}$,

$$p\left(\mathbf{Y}|\mathbf{X},\beta\right)=\frac{1}{\left(2\pi\right)^{\frac{DN}{2}}\left|\mathbf{K}\right|^{\frac{D}{2}}}\exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\right)\right), \tag{19}$$

where $\mathbf{K}=\mathbf{X}\mathbf{X}^{\mathrm{T}}+\beta^{-1}\mathbf{I}$ and $\mathbf{X}=\left[\mathbf{x}_1^{\mathrm{T}}\dots\mathbf{x}_N^{\mathrm{T}}\right]^{\mathrm{T}}$. The structure of this model is shown in 7(b). Note that with our earlier definition of $\mathbf{C}=\mathbf{W}\mathbf{W}^{\mathrm{T}}+\beta^{-1}\mathbf{I}$ we can write the marginal likelihood for standard PPCA (6) as

$$p\left(\mathbf{Y}|\mathbf{W},\beta\right)=\frac{1}{\left(2\pi\right)^{\frac{DN}{2}}\left|\mathbf{C}\right|^{\frac{N}{2}}}\exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\right)\right),$$

which highlights to a greater extent the duality between (19) and (6). Optimisation of (19) is clearly highly related to optimisation of (6). Tipping and Bishop [1999] showed how to optimise (6), in the next section we review this optimisation for DPPCA, but generalise it slightly so that it applies for any positive definite matrix $\mathbf{S}$, rather than only the inner product matrix $\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$. First though we make the connection to Gaussian processes by highlighting the fact that (19) can be written as

$$p\left(\mathbf{Y}|\mathbf{X},\beta\right)=\prod_{i=1}^{D}\frac{1}{\left(2\pi\right)^{\frac{N}{2}}\left|\mathbf{K}\right|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}\mathbf{y}_{:,i}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y}_{:,i}\right), \tag{20}$$

here $\mathbf{y}_{:,i}$ is the ith column of $\mathbf{Y}$. This likelihood is thus recognised as a product of D independent Gaussian processes, each process being associated with a different dimension of the data set. However, here we are suggesting maximising over $\mathbf{X}$ as well as the kernel parameters. If $q>D$ this maximisation would not be well determined, but as long as $q<D$ we are obtaining a reduced dimensional representation of our data. We will now show how, for the case of a linear covariance matrix, this model is equivalent to PCA.

5.1 Maximisation of the Marginal Likelihood

The proof of the maximum likelihood solution for dual probabilistic PCA closely mirrors that given in Tipping and Bishop [1999], we include it here for completeness. For a more general proof see Lawrence and Sanguinetti [2004]. Maximising (19) is equivalent to minimising its negative logarithm,

$$L=\frac{N}{2}\ln 2\pi+\frac{1}{2}\ln\left|\mathbf{K}\right|+\frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{S}\right), \tag{21}$$

where $\mathbf{S}=D^{-1}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$. The gradient of the negative log likelihood with respect to $\mathbf{X}$ can be found as

$$\frac{\partial L}{\partial\mathbf{X}}=-\mathbf{K}^{-1}\mathbf{S}\mathbf{K}^{-1}\mathbf{X}+\mathbf{K}^{-1}\mathbf{X},$$

setting the equation to zero and pre-multiplying by $\mathbf{K}$ gives

$$\mathbf{S}\left[\beta^{-1}\mathbf{I}+\mathbf{X}\mathbf{X}^{\mathrm{T}}\right]^{-1}\mathbf{X}=\mathbf{X}.$$

We substitute $\mathbf{X}$ with its singular value decomposition, $\mathbf{X}=\mathbf{U}\mathbf{L}\mathbf{V}^{\mathrm{T}}$, giving

$$\mathbf{S}\mathbf{U}\left[\mathbf{L}+\beta^{-1}\mathbf{L}^{-1}\right]^{-1}\mathbf{V}^{\mathrm{T}}=\mathbf{U}\mathbf{L}\mathbf{V}^{\mathrm{T}}$$

Right multiplying both sides by $\mathbf{V}$ (note that the solution is invariant to $\mathbf{V}$) we have, after some rearrangement,

$$\mathbf{S}\mathbf{U}=\mathbf{U}\left(\beta^{-1}\mathbf{I}+\mathbf{L}^{2}\right),$$

which, since $\left(\beta^{-1}\mathbf{I}+\mathbf{L}^{2}\right)$ is diagonal can be solved by an eigenvalue problem where $\mathbf{U}$ are eigenvectors of $\mathbf{S}$ and $\Lambda=\left(\beta^{-1}\mathbf{I}+\mathbf{L}^{2}\right)$ are the eigenvalues. This implies that the elements from the diagonal of $\mathbf{L}$ are given by

$$l_i=\left(\lambda_i-\beta^{-1}\right)^{\frac{1}{2}}. \tag{22}$$

5.2 The Retained Eigenvalues

If $q<D$ we must select which eigenvectors to retain, all eigenvectors are associated with stationary points, so how do we choose which to retain? For convenience let us ignore our previously defined ordering of the eigenvalues in terms of their magnitude and assume that we keep the first q eigenvalues.

First note that

$$\mathbf{K}=\mathbf{U}\left[\mathbf{L}^{2}+\beta^{-1}\mathbf{I}\right]\mathbf{U}^{\mathrm{T}}$$

where $\mathbf{U}$ is all the eigenvectors of $\mathbf{S}$. The Kullback Leibler (KL) divergence between zero mean Gaussians with covariances given by $\mathbf{K}$ and $\mathbf{S}$ given by (21) minus the log determinant of $\mathbf{S}$, which is constant in $\mathbf{X}$. Minimising this KL divergence is thus equivalent to minimising (21).

$$\begin{aligned}
\mathrm{KL}\left(\mathbf{S}||\mathbf{K}\right) &= \frac{1}{2}\ln|\mathbf{K}|-\frac{1}{2}\ln|\mathbf{S}|+\frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{S}\right)-\frac{N}{2}\\
&= \frac{1}{2}\sum_{i=1}^{q}\ln\lambda_i-\frac{N-q}{2}\ln\beta-\frac{1}{2}\sum_{i=1}^{N}\ln\lambda_i+\frac{1}{2}\mathrm{tr}\left(\left[\mathbf{L}^{2}+\beta^{-1}\mathbf{I}\right]^{-1}\Lambda\right)\\
&= -\frac{1}{2}\sum_{i=q+1}^{N}\ln\lambda_i-\frac{N-q}{2}\ln\beta-\frac{N-q}{2}+\frac{\beta}{2}\sum_{i=q+1}^{N}\lambda_i
\end{aligned}$$

where we have used the fact that $\mathbf{S}=\mathbf{U}\Lambda\mathbf{U}^{\mathrm{T}}$. Differentiating with respect to β and setting the result to zero to obtain a fixed point equation then gives

$$\beta=\frac{N-q}{\sum_{i=q+1}^{N}\lambda_i}$$

which when substituted back leads to

$$\mathrm{KL}\left(\mathbf{S}||\mathbf{K}\right)=\frac{N-q}{2}\left(\ln\frac{\sum_{i=q+1}^{N}\lambda_i}{N-q}-\frac{1}{N-q}\sum_{i=q+1}^{N}\ln\lambda_i\right), \tag{23}$$

which is recognised as the difference between the log ratio of the arithmetic and geometric means of the discarded eigenvalues. This difference will be zero if and only if the discarded eigenvalues are constant (when the arithmetic and geometric means become equal) otherwise it is positive. The difference is minimised by ensuring that the eigenvalues we discard are adjacent to each other in terms of magnitude.

Which eigenvalues should we then discard? From (22) we note that the retained eigenvalues must be larger than β, otherwise l_i will be complex. The only way this can be true is if we discard the smallest $N-q$ eigenvalues.

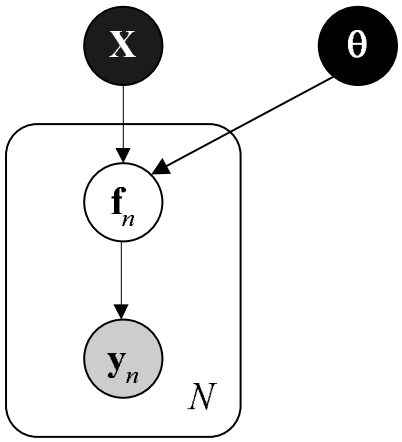


Figure 8: The Gaussian process as a latent variable model, now both kernel parameters, $\boldsymbol{\theta}$ and latent positions are optimised.

5.3 Equivalence of Eigenvalue Problems

In Section 3 we reviewed probabilistic PCA, here we have introduced a new dual version of probabilistic PCA which leads to a different eigenvalue problem. However, these eigenvalue problems are equivalent as we shall now show. For DPPCA the eigenvalue problem is of the form

$$\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\mathbf{U} = \mathbf{U}\Lambda.$$

Premultiplying by $\mathbf{Y}^{\mathrm{T}}$ then gives

$$\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\mathbf{U} = \mathbf{Y}^{\mathrm{T}}\mathbf{U}\Lambda \tag{24}$$

Since $\mathbf{U}$ is the eigenvectors of $\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$ (see the previous section) the matrix $\mathbf{U}^{\mathrm{T}}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\mathbf{U} = \Lambda$, therefore matrix $\mathbf{U}' = \mathbf{Y}^{\mathrm{T}}\mathbf{U}\Lambda^{-\frac{1}{2}}$ is orthonormal. Post multiplying both sides of (24) by $\Lambda^{-\frac{1}{2}}$ gives

$$\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\mathbf{U}' = \mathbf{U}'\Lambda$$

which is recognised as the form of the eigenvalue problem associated with PPCA as given in (7), where the eigenvectors of $\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$ are given by $\mathbf{U}' = \mathbf{Y}^{\mathrm{T}}\mathbf{U}\Lambda^{-\frac{1}{2}}$ and the eigenvalues are given by Λ (as they were for DPPCA).

6 Non-linear GP-LVM

We saw in the previous section how PCA can be interpreted as a product of Gaussian processes that maps latent-space points to points in data-space. The positions of the points in the latent-space can be determined by maximising the process likelihood with respect to $\mathbf{X}$. It is natural, therefore, to consider alternative GP-LVMs by introducing covariance functions which allow for non-linear processes. The resulting models will not, in general, be optimisable through an eigenvalue problem.

6.1 Optimisation of the Non-linear Model

In Section 5 we saw for the linear kernel that a closed form solution for dual PPCA could be obtained up to an arbitrary rotation matrix. For non-linear kernels, such as the RBF kernel and MLP kernel discussed in Section 4.4 there will be no such closed form solution and there are likely to be multiple local optima. To use a particular kernel in the GP-LVM we first note that gradients

of (18) with respect to the latent points can be found through first taking the gradient with respect to the kernel,

$$\frac{\partial L}{\partial \mathbf{K}} = \mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\mathbf{K}^{-1} - D\mathbf{K}^{-1}, \tag{25}$$

and then combining it with $\frac{\partial \mathbf{K}}{\partial x_{n,j}}$ through the chain rule. As computation of (25) is straightforward and independent of the kernel choice we only require that the gradient of the kernel with respect to the latent points can be computed. These gradients may then be used in combination with (18) in a non-linear optimiser to obtain a latent variable representation of the data. Furthermore, gradients with respect to the parameters of the kernel matrix may be computed and used to jointly optimise $\mathbf{X}$ and the kernel's parameters.

The log-likelihood is a highly non-linear function of the embeddings and the parameters. We are therefore forced to turn to gradient based optimisation of the objective function. In all our experiments we made use of conjugate gradients or the scaled conjugate gradient [Møller, 1993] algorithm.

6.2 Illustration of GP-LVM via SCG

To illustrate the Gaussian process latent variable model we now make use of the 'multi-phase oil flow' data [Bishop and James, 1993]. This is a twelve dimensional data set containing data of three known classes corresponding to the phase of flow in an oil pipeline: stratified, annular and homogeneous. In Bishop et al. [1998] this data was used to demonstrate the GTM algorithm. Here we use a sub-sampled version of the data (containing 100 data points) to demonstrate the fitting of a GP-LVM with a simple radial basis function (RBF) kernel.

As we saw in Section 5, seeking a lower dimensional embedding with PCA is equivalent to a GP-LVM model with a linear kernel,

$$k\left(\mathbf{x}_n, \mathbf{x}_m\right) = \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m + \beta^{-1}\delta_{nm},$$

where δ_{ij} is the Kronecker delta function.

For comparison we visualised the data set using several of the approaches mentioned in the introduction. In Figure 9(a) we show the first two principal components of the data. Figure 9(b) then shows the visualisation obtained using the GP-LVM with the RBF kernel,

$$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \alpha_{\mathrm{rbf}} \exp\left(-\frac{\gamma}{2}\left(\mathbf{x}_i - \mathbf{x}_j\right)^{\mathrm{T}}\left(\mathbf{x}_i - \mathbf{x}_j\right)\right) + \alpha_{\mathrm{bias}} + \beta^{-1}\delta_{ij}.$$

To obtain this visualisation the log likelihood was optimised jointly with respect to the latent positions $\mathbf{X}$ and the kernel parameters α_{bias}, α_{rbf}, β and γ. The kernel was initialised using PCA to set $\mathbf{X}$, the kernel parameters were initialised as $\alpha_{\mathrm{rbf}} = \gamma = 1$ and $\beta^{-1} = \alpha_{\mathrm{bias}} = \exp\left(-1\right)$.

Note that there is a redundancy in the representation between the overall scale of the matrix $\mathbf{X}$ and the value of γ. This redundancy was removed by penalising the log likelihood with half the sum of the squares of each element of $\mathbf{X}$: this implies we were actually seeking a MAP solution[3] with a Gaussian prior for $\mathbf{X}$,

$$p\left(\mathbf{X}\right) = \prod_{n=1}^{N} N\left(\mathbf{x}_n|\mathbf{0}, \mathbf{I}\right).$$

The likelihood for the RBF kernel was optimised using scaled conjugate gradient (see `http://www.dcs.shef.ac.uk/~neil/gplvmcpp/` for the C++ code used).

In Figure 9(c) we show the result of non-metric MDS using the stress criterion of Kruskal [1964]. Figure 9(d) shows the result from the 'Sammon mapping' [Sammon, 1969]. To objectively evaluate the quality of the visualisations we classified each data point according to the class of its nearest neighbour in the two dimensional latent-space supplied by each method. The errors made

[3]Multiplying the likelihood by this prior leads to a joint distribution over data points and latent points. As a function of $\mathbf{X}$ this joint distribution is proportional to the posterior distribution $p\left(\mathbf{X}|\mathbf{Y}\right)$, therefore maximising the joint distribution is equivalent to seeking a MAP solution.

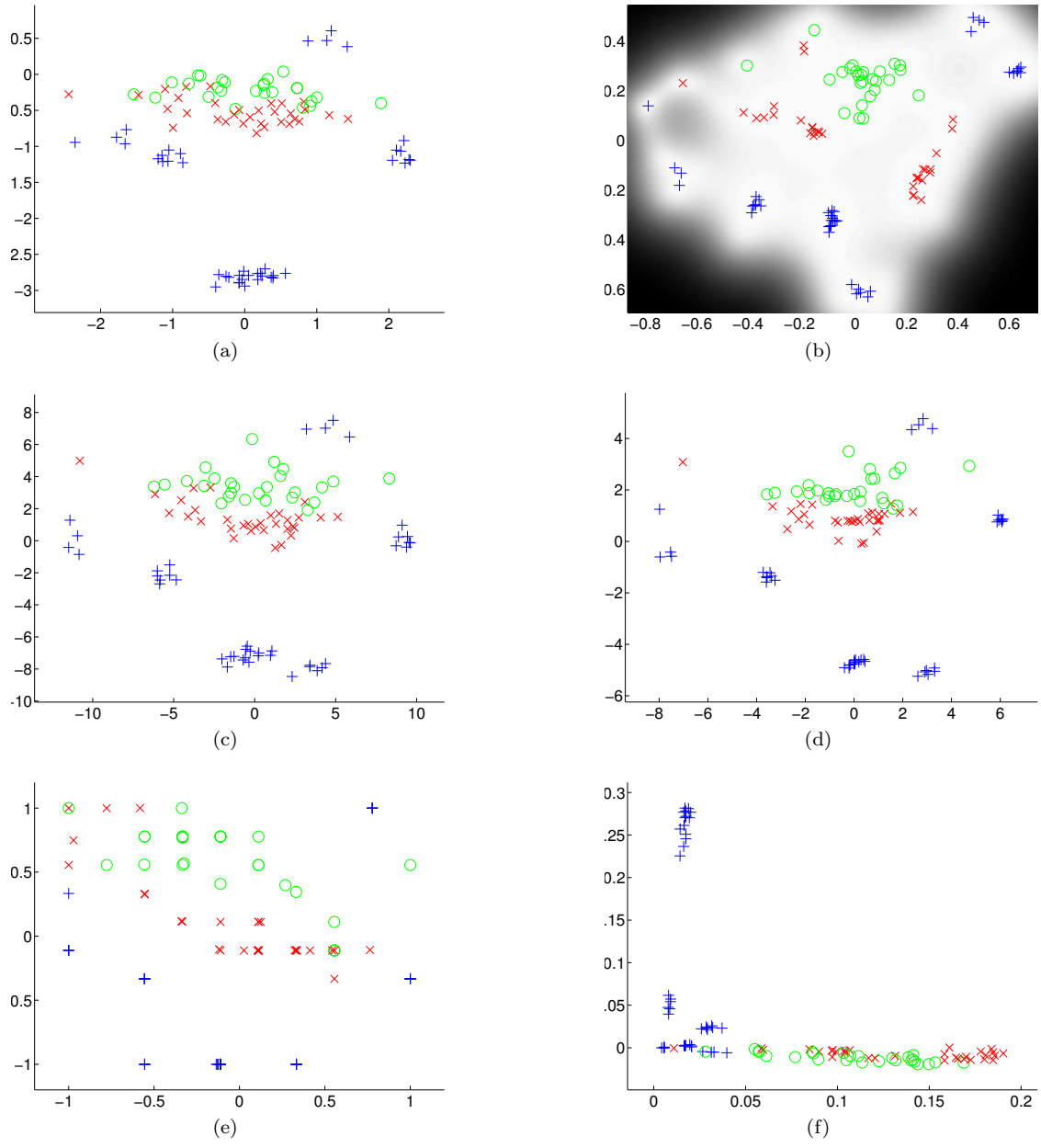


Figure 9: Visualisation of the Oil data with (a) PCA (a linear GP-LVM) and (b) A GP-LVM which uses an RBF kernel, (c) Non-metric MDS using Kruskal's stress, (d) M 'Sammon Mapping', (e) GTM and (f) kernel PCA. Red crosses, green circles and blue plus signs represent stratified, annular and homogeneous flows respectively. The greyscales in plot (b) indicate the precision with which the manifold is expressed in data-space for that latent point.

Method	PCA	GP-LVM	Non-metric MDS	Metric MDS	GTM*	kernel PCA*
Errors	20	4	13	6	7	13

Table 1: Errors made by the different methods when using the latent-space for nearest neighbour classification in the latent space. Both the GTM and kernel PCA are given asterisks as the result shown is the best obtained for each method from a range of different parameterisations.

by such a classification are given in Table 1. For the GTM and kernel PCA some selection of parameters is required. For GTM we varied the size of the latent grid between 3×3 and 15×15, and the number of hidden nodes in the RBF network was varied between 4 and 36. The best result was obtained for a 10×10 latent grid with 25 nodes in the RBF network, it is shown in Figure 9(e). Note the characteristic gridding effect in the GTM's visualisation which arises from the layout of the latent points. For kernel PCA we used the RBF kernel and varied the kernel width between 0.01 and 100. The best result was obtained for a kernel width of 0.75, the associated visualisation is shown in Figure 9(f).

The gradient based optimisation of the RBF based GP-LVM's latent-space shows results which are clearly superior (in terms of separation between the different flow phases) to those achieved by the linear PCA model. The GP-LVM approach leads to a number of errors that is the smallest of all the approaches used. Additionally the use of a Gaussian process to perform our 'mapping' means that we can express uncertainty about the positions of the points in the *data* space. For our formulation of the GP-LVM the level of uncertainty is shared across all D dimensions and thus may be visualised in the latent-space.

6.2.1 Visualising the Uncertainty

Recall that the likelihood (20) is a product of D separate Gaussian processes. In all that has followed we have retained the implicit assumption in PCA that *a priori* each dimension is identically distributed by assuming that the processes shared the same covariance/kernel function $\mathbf{K}$. Sharing of the covariance function also leads to an *a posteriori* shared level of uncertainty in each process. While it is possible to use different covariance functions for each dimension and may be necessary when each of the data's attributes have different characteristics[4]; the more constrained model implemented here allows us to visualise the uncertainty in the latent space and will be preferred for our empirical studies[5]. In Figure 9(b) (and subsequently) the uncertainty is visualised by varying the intensity of the background pixels. The lighter the pixel the higher the precision of the mapping.

6.2.2 Computational Complexity

While the quality of the results seem good, a quick analysis of the algorithmic complexity shows that each gradient step requires an inverse of the kernel matrix (see (25)), an $O\left(N^3\right)$ operation, rendering the algorithm impractical for many data sets of interest.

6.3 Large Data Sets

The sparse approximation suggested in Lawrence [2004, 2005] is a sub-set of data approach [Lawrence et al., 2003, Rasmussen and Williams, 2006, pg. 177]. Whilst this approach leads to somewhat simple algorithms for optimisation of the GP-LVM, it suffers from the lack of a convergence criterion and discards information in the data set. A more promising approach to sparsification is suggested for Gaussian process regression by Snelson and Ghahramani [2006]

[4] A simple example of this is given by Grochow et al. [2004] with the 'scaled GP-LVM', where a scale parameter is associated with each dimension of the data.

[5] The two approaches, constraining each data direction to the same kernel and allowing each data dimension to have its own kernel are somewhat analogous to the difference between probabilistic PCA, where each output data shares a variance, and factor analysis, where each data dimension maintains its own variance.

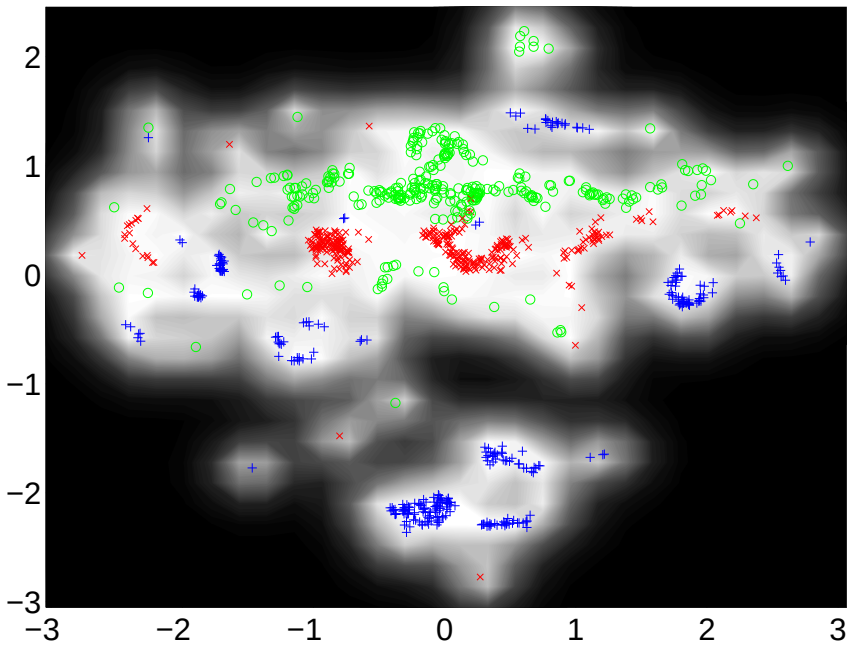


Figure 10: The full oil flow data set visualised with an RBF based kernel using sub-set of data approximations.

Model	PCA	Sparse GP-LVM (IVM)	GP-LVM (RBF)	GTM	Y
Errors	162	24	1	11	2

Table 2: Number of errors for nearest neighbour classification in the latent-space for the full oil data set (1000 points). Far right column contains result for nearest neighbour in the data space, also presented is a result for the GTM algorithm.

and has recently be placed in a more general framework by Quiñonero Candela and Rasmussen [2005]. The application of this approach in the GP-LVM is available on-line and is the subject of a forthcoming paper [Lawrence, 2006, in preparation].

In Figure 10 we present visualisations of the oil data using a sub-set of data based sparse GP-LVM algorithm with the RBF kernel. In Figure 11 we show the data visualised with the non-sparse GP-LVM algorithm. Again we considered a nearest neighbour classifier in the latent-space to quantify the quality of the visualisations. We note that there appears to be a degradation in the quality of the GP-LVM model associated with the sparsification, in comparision to the full GP-LVM algorithm and the sub-set of data based sparse GP-LVM performs worse.

6.4 Back Constraints

An interesting characteristic of the GP-LVM is that it provides a smooth mapping from latent space to the data space. This implies that points which are close in latent space will be close in data space. However, it does not imply that points which are close in data space will be necessarily mapped as close together in latent space. In recent work [Lawrence and Quiñonero Candela, 2006, in preparation] the use of back constraints is suggested. Back constraints constrain each latent points to be a smooth function of its corresponding data point. This forces points which are close in data space to be close in latent space.

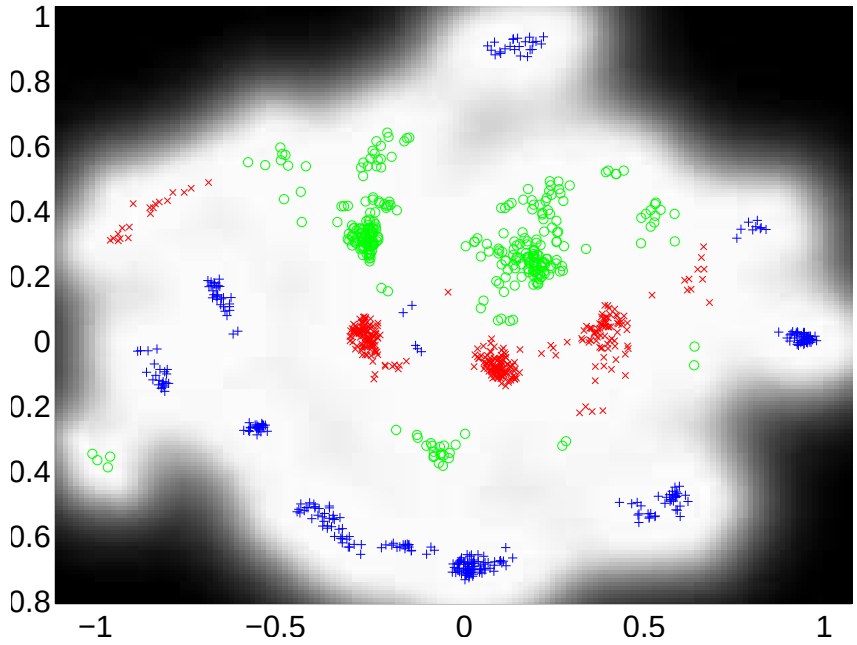


Figure 11: The full GP-LVM algorithm with RBF kernel on the oil flow data (uses the GPLVMCPP toolbox).

6.4.1 Motion Capture Data

A neat illustration of the issues that arise when the GP-LVM is used without back constraints is given by a simple motion capture data set. The data consists of a subject breaking into a run from standing[6]. There are approximately three full strides in the sequence. The mean of the data is removed from each frame so in effect the subject is running 'in place'. The data is therefore somewhat periodic in nature, however the subject changes the angle of the run throughout the sequence becoming more upright as it proceeds. Our experimental set up was as follows. For both models a GP-LVM with an RBF kernel for a covariance function was used. The back constraint was implemented through an RBF based kernel mapping for which we set $\gamma = 1 \times 10^{-3}$. Both models were initialised using PCA. For the RBF model this is straightforward, but for the kernel model this was achieved by setting the kernel parameters, $\mathbf{A}$, to minimise the squared distance between the latent positions given by the mapping and those given by PCA. The latent positions/mapping parameters and the GP covariance function parameters were then jointly optimised using conjugate gradients. Scripts for re-implementing these experiments are available on line in the FGPLVM toolbox.

The results from visualisation using the GP-LVM both in unconstrained and back constrained forms are shown in Figure 12. The data is temporal in nature (although the GP-LVM is not taking advantage of this fact) and we have connected points in the plots that are neighbours in time. In Figure 12(a) the sequence does not clearly show the periodic nature of the data. The likelihood of this model is higher, as we should expect given that the other model is constrained, however the sequence is split across several sub-sequences[7]. To reflect the periodic nature of the sequence it is necessary to use a circular structure. Such a structure will be of the form of a squashed spiral which will either have less representational power in the inner rings (analogous to inner groove distortion in gramophone records) or will cross over itself in a manner which is not consistent with

[6]Data made available by the Ohio State University Advanced Computing Centre for the Arts and Design, available from `http://accad.osu.edu/research/mocap/mocap_data.htm`, sequence 'Figure Run 1' in unprocessed `.txt` format.

[7]Note this is *not* due to overfitting: the model provides a smooth representation of the data which generalises well across the latent space.

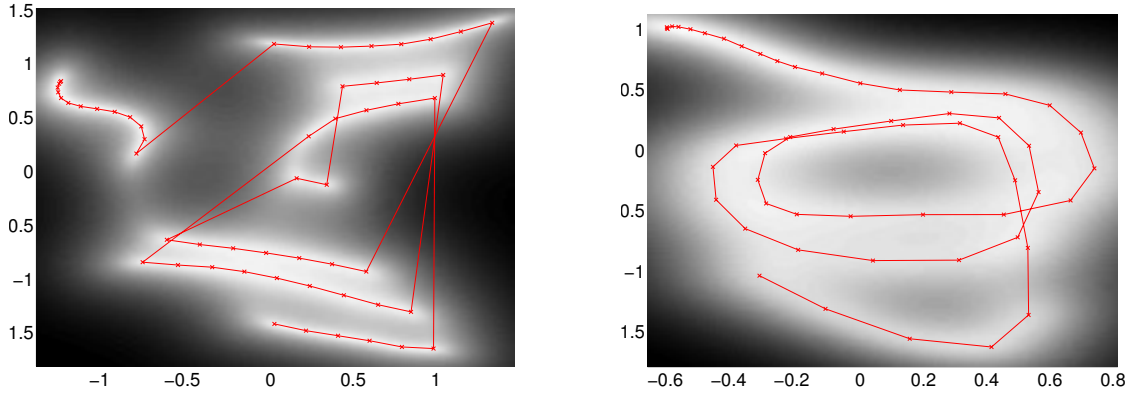


Figure 12: Visualisation of the motion capture data. (a) The regular GP-LVM, log likelihood 1,543 (`demStick1` in the FGPLVM toolbox) and (b) the GP-LVM with back constraints (`demStick3`), log likelihood 1,000. The paths of the sequences through latent space are shown as solid lines.The back constraint used was an RBF kernel mapping with $\gamma = 1 \times 10^{-3}$. In both cases the start of the sequence is towards the top left and the end is towards the bottom centre-left. The grey scale background image indicates the precision with which the mapping is expressed.

the data. The higher likelihood solution turns out to be placing points far apart which are actually close together. Note that the problem arises because the latent space is too constrained. Using a three dimensional latent space alleviates the problem[8] and we expect a two dimensional latent space which is topologically cylindrical would also resolve the issue. The back constrained model shows a squashed spiral structure which reflects the periodic nature of the data and maintains a representation of the angle of the run. The changing angle of the run as the sequence proceeds is depicted in Figure 13.

6.4.2 Vowel Data

As a further example we considered a single speaker vowel data set. The data consists of the cepstral coefficients and deltas of ten different vowel phonemes and is acquired as part of a vocal joystick system Bilmes et al. [2006]. A particular characteristic of this data set is that PCA, which is used as the initalisation when the back constraints aren't used, fails to separate the data at all. As a result the non-back constrained model tends to fragment the different vowels. The results with the back constrianed model tend to keep like vowels closer together (Figure 14).

6.5 GP-LVM with Dynamics

Recently Wang et al. [2006] described an approach to applying dynamics to the GP-LVM. To see how this is done, we assume the data is presented in temporal order (*i.e.* $\mathbf{y}_1$ is the first data point in the series and $\mathbf{y}_N$ is the last). The obvious route to augmenting the model with dynamics is to place a Markov chain distribution over the latent space by defining $p(\mathbf{x}_n|\mathbf{x}_{n-1})$, which gives a prior distribution $p(\mathbf{X}) = p(\mathbf{x}_1)\prod_{n=2}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1})$. Of course, combining this prior with $p(\mathbf{Y}|\mathbf{X})$ to obtain the marginal likelihood $p(\mathbf{Y})$ is in general not tractable. However, it is straightforward to obtain maximum *a posteriori* (MAP) estimates of the solution. Instead of a simple Markov chain, Wang et al. [2006] suggest a Gaussian process to relate $\mathbf{x}_n$ to $\mathbf{x}_{n-1}$. If this GP predicts, at each time step, the change in position for the next time step, the joint likelihood over the latent

[8] A script to run the experiment is available on line (`demStick4` in the FGPLVM toolbox).

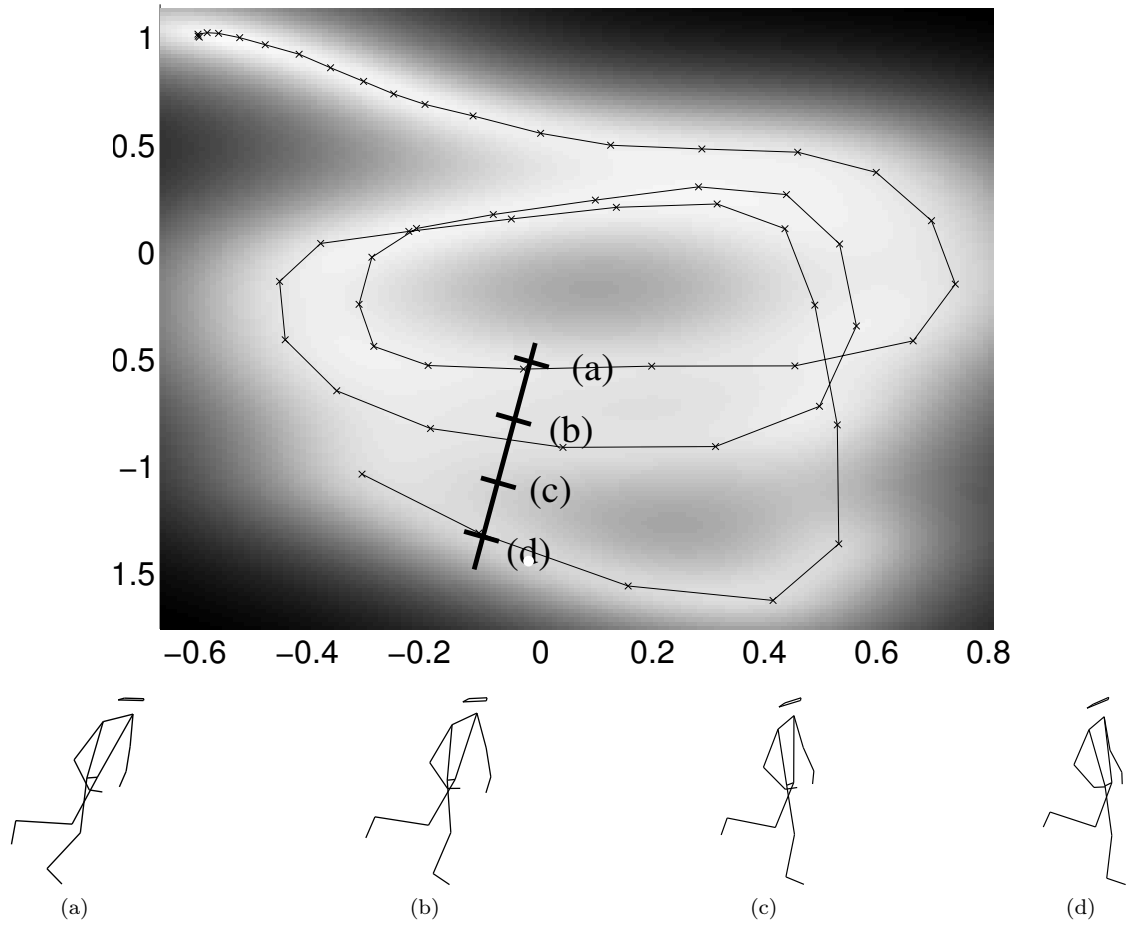
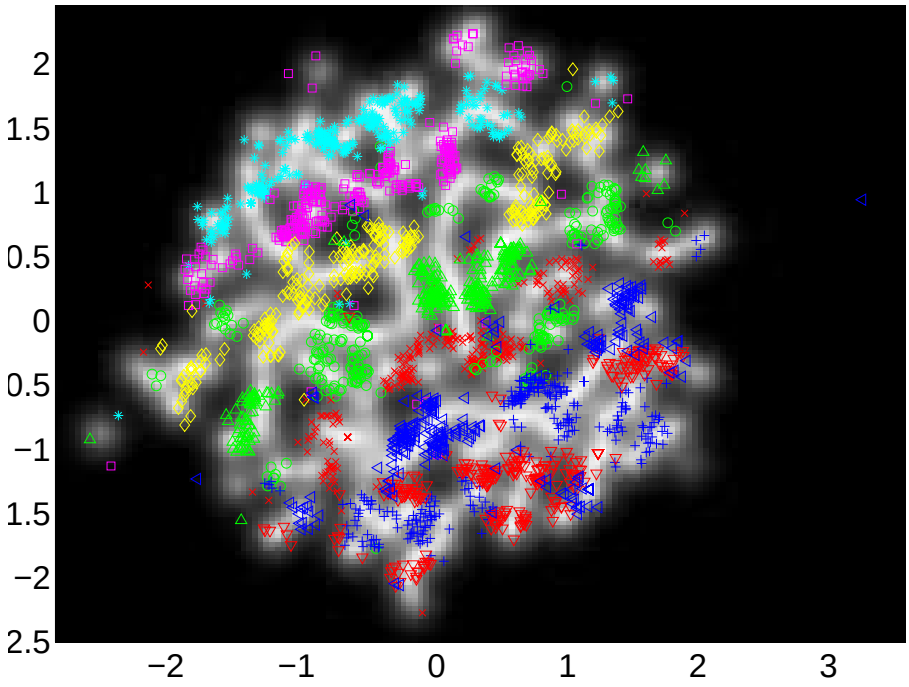
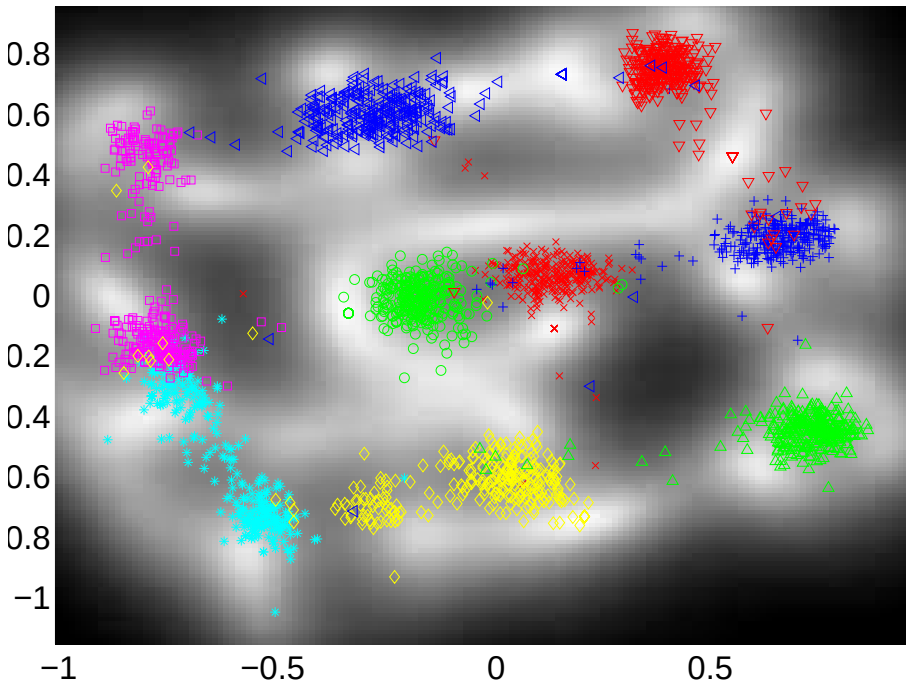


Figure 13: Projection into data space from four points in the latent space. Note how the position in the cycle is the same but the inclination of the runner differs becoming more upright as the sequence proceeds.



(a)



(b)

Figure 14: Visualisation of the vowel data (a) without back constraints and (b) with back constraints. The different vowels are shown as follows: */a/* red cross */ae/* green circle */ao/* blue plus */e/* cyan asterix */i/* magenta square */ibar/* yellow diamond */o/* red down triangle */schwa/* green up triangle and */u/* blue left triangle (`demVowels2` and `demVowels3` in the FGPLVM toolbox).

variables and $\mathbf{Y}$ is given by

$$\begin{aligned} p(\mathbf{Y}, \mathbf{X}) &= -\frac{DN}{2}\log 2\pi - \frac{D}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\right) \\ &\quad -\frac{qN}{2}\log 2\pi - \frac{q}{2}\log|\mathbf{K}_x| - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_x^{-1}\left(\hat{\mathbf{X}} - \tilde{\mathbf{X}}\right)\left(\hat{\mathbf{X}} - \tilde{\mathbf{X}}\right)^{\mathrm{T}}\right), \end{aligned} \tag{26}$$

where $\hat{\mathbf{X}} = [\mathbf{x}_2 \dots \mathbf{x}_N]^{\mathrm{T}}$ and $\tilde{\mathbf{X}} = [\mathbf{x}_1 \dots \mathbf{x}_{N-1}]^{\mathrm{T}}$ the kernel $\mathbf{K}_x$ is that associated with the dynamics Gaussian process and is constructed on the matrix $\tilde{\mathbf{X}}$.

6.5.1 Sampling from Dynamics

Consider a dynamics Gaussian process based on an RBF kernel and a white noise term,

$$k(\mathbf{x}_n, \mathbf{x}_m) = \alpha'_{\mathrm{rbf}} \exp\left(-\frac{\gamma'}{2}(\mathbf{x}_n - \mathbf{x}_m)^{\mathrm{T}}(\mathbf{x}_n - \mathbf{x}_m)\right) + \beta'^{-1}\delta_{nm},$$

where δ_{nm} is the Kronecker delta function. Rather than learning the parameters of the dynamics model we suggest an alternative approach of selecting the dynamics model parameters by hand. Such an approach may seem unwieldy, but there are only three parameters in the covariance function, each of which has a clear interpretation. The signal variance is given by α'_{rbf} and the noise variance by β'^{-1}, thus the signal to noise ratio is given by $\sqrt{\alpha'_{\mathrm{rbf}}\beta'}$. The remaining parameter controls the smoothness of the function, taking its square root and inverting, $l = \frac{1}{\sqrt{\gamma}}$, gives a parameter known as the *characteristic length scale.* In each dimension the mean level of zero up-crossings in a unit interval is given by $(2\pi l)^{-1} = \frac{\sqrt{\gamma}}{2\pi}$ [Rasmussen and Williams, 2006], this is related to the number of times the dynamics switches direction. For the example given below we used $\gamma = 0.2$, $\alpha_{\mathrm{rbf}} = 0.01$ and $\beta^{-1} = 1 \times 10^{-6}$ which is equivalent to a signal to noise ratio of 100. In Figure 15 we show some examples of two dimensional dynamics fields sampled using parameters in the neighbourhood of those given above.

6.5.2 Motion Capture Data

By selecting a sensible dynamics prior in the latent space the motion capture data again reflects the period nature of the paces (Figure 16).

6.6 Loop Closure in Robotics

In on-going work with Dieter Fox and Brian Ferris at the University of Washington we are interested in loop closure for robotic navigation, included as a final example is a data set of a robot completing a loop while reading signal strengths from 30 different wireless access points. To produce a neat track and close the loop it turns out it is necessary to use dynamics and back constraints as seen in Figure 17. When the GP-LVM is used without dynamics (Figure 17(a) and (b)) the path in the latent space is noisy. Dynamics forces a tighter path in latent space (Figure 17(c)) but there is no loop closure. Finally by combining back constraints with dynamics we can obtain loop closure (Figure 17(d)).

7 Conclusion

This tutorial has aimed to give an overview of the Gaussian process latent variable model, starting from the perspective of a simple linear latent variable model, and through the introduction of Gaussian processes, finishing with a fully probabilistic approach to non-linear dimensionality reduction. In the results section we showed some simple visualisations achieved with the algorithm and gave an overview of some of the extensions to the algorithm. Code for recreating all the results we presented is available on-line: (`http://www.dcs.shef.ac.uk/~neil/gpsoftware.html`) and in many cases we have referred to the specific scripts in captions of figures.

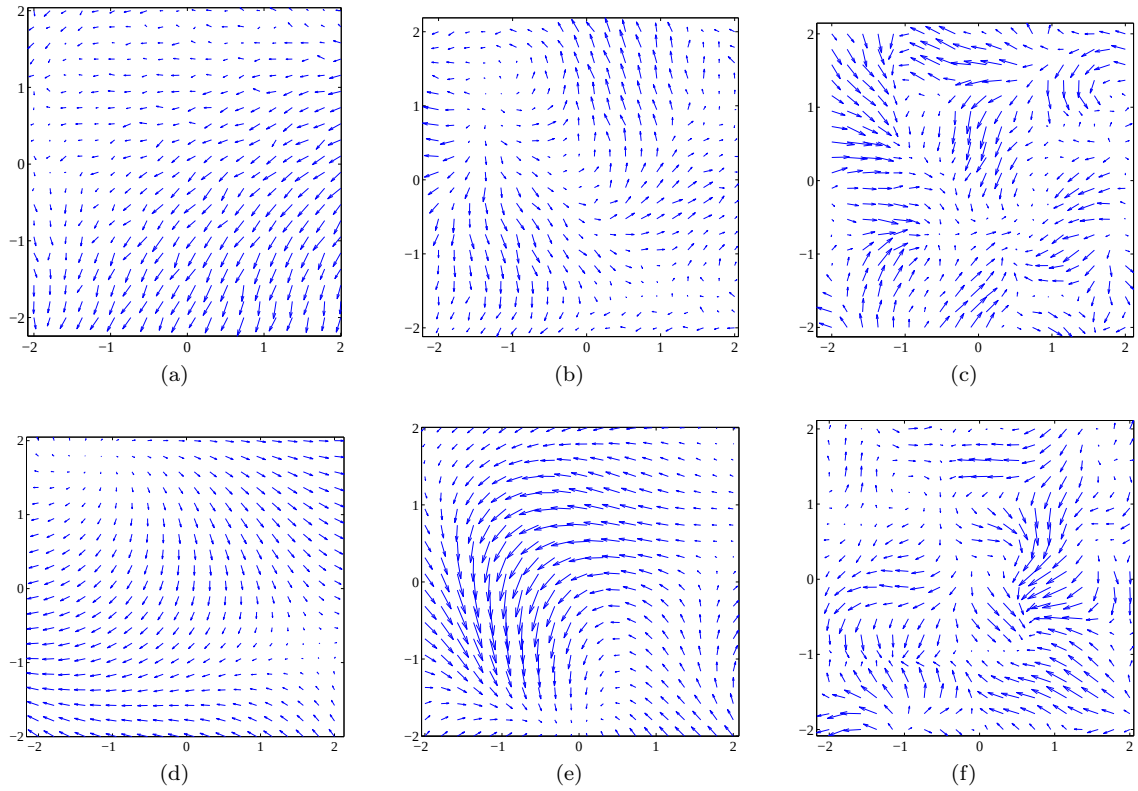


Figure 15: Samples from the prior over the latent space dynamics. One sample from each parameterisation is shown. The top row of plots has a signal to noise ratio of 5 and the bottom row is 100. Length scales decrease from left to right, left most column $l = 2.24$; middle column $l = 1$; and rightmost column $l = 0.447$. More specifically the parameters used in each plot are (a) $\gamma' = 0.2$, $\beta'^{-1} = 4 \times 10^{-4}$, (b) $\gamma' = 1$, $\beta' = 4 \times 10^{-4}$, (c) $\gamma' = 5$, $\beta'^{-1} = 4 \times 10^{-4}$, (d) $\gamma' = 0.2$, $\beta'^{-1} = 1 \times 10^{-6}$, (e) $\gamma' = 1$, $\beta'^{-1} = 1 \times 10^{-6}$ and (f) $\gamma' = 5$, $\beta'^{-1} = 4 \times 10^{-6}$ with $\alpha'_{\text{rbf}} = 0.1$ for all plots. The overall scale was set to unity. The parameter settings used to produce Figure 16 are associated with (d).

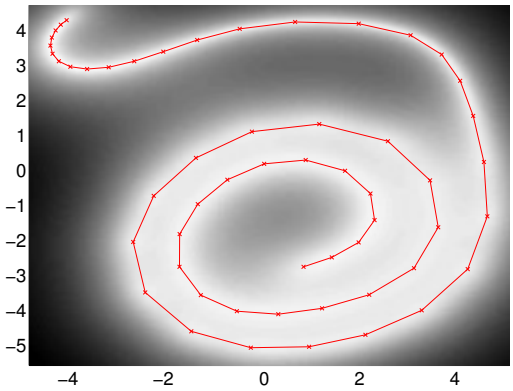


Figure 16: Visualisation of the motion capture data using the GP-LVM with dynamics (`demStick2` in the FGPLVM toolbox)

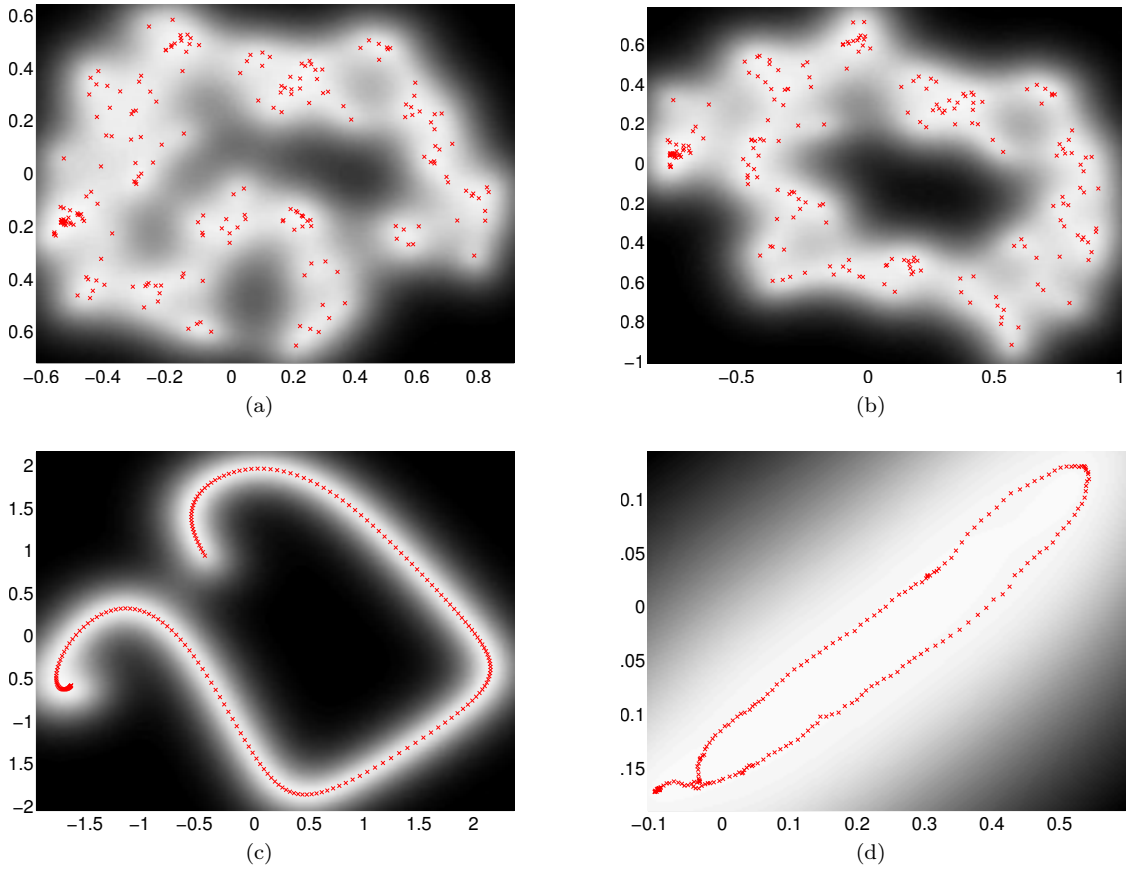


Figure 17: Use of back constraints and dynamics to obtain loop closure in a robot navigation example. (a) GP-LVM without back constraints or dynamics, (b) GP-LVM with back constraints, no dynamics, (c) GP-LVM with dynamics, no back constraints, (d) GP-LVM with back constraints and dynamics. These results can be recreated with scripts `demRobotWireless1` through `demRobotWireless4`.

References

A. J. Bell and T. J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

J. Bilmes, J. Malkin, X. Li, S. Harada, K. Kilanski, K. Kirchhoff, R. Wright, A. Subramanya, J. Landay, P. Dowden, and H. Chizeck. The vocal joystick. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*. IEEE, May 2006. To appear.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995. ISBN 0198538642.

C. M. Bishop and G. D. James. Analysis of multiphase flows using dual-energy gamma densitometry and neural networks. *Nuclear Instruments and Methods in Physics Research*, A327:580–593, 1993. doi: 10.1016/0168-9002(93)90728-Z.

C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: the Generative Topographic Mapping. *Neural Computation*, 10(1):215–234, 1998. doi: 10.1162/089976698300017953.

K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. In *ACM Transactions on Graphics (SIGGRAPH 2004)*, pages 522–531, 2004. doi: 10.1145/1186562.1015755.

J. B. Kruskal. Multidimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–28, 1964. doi: 10.1007/BF02289565.

N. D. Lawrence. Gaussian process models for visualisation of high dimensional data. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 329–336, Cambridge, MA, 2004. MIT Press.

N. D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 11 2005.

N. D. Lawrence. Large scale learning with the Gaussian process latent variable model. Technical Report CS-06-05, University of Sheffield, 2006. Document updated on December 16, 2008. Original from February 17th, 2006.

N. D. Lawrence and J. Quiñonero Candela. Local distance preservation in the GP-LVM through back constraints. In W. Cohen and A. Moore, editors, *Proceedings of the International Conference in Machine Learning*, volume 23, pages 513–520. Omnipress, 2006. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143909.

N. D. Lawrence and G. Sanguinetti. Matching kernels through Kullback-Leibler divergence minimisation. Technical Report CS-04-12, The University of Sheffield, Department of Computer Science, 2004.

N. D. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 625–632, Cambridge, MA, 2003. MIT Press.

D. J. C. MacKay. Maximum likelihood and covariant algorithms for independent component analysis. Unpublished manuscript, available from `http://wol.ra.phy.cam.ac.uk/mackay/homepage.html`, 1996.

D. J. C. MacKay. Introduction to Gaussian Processes. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, volume 168 of *Series F: Computer and Systems Sciences*, pages 133–166. Springer-Verlag, Berlin, 1998.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis*. Academic Press, London, 1979. ISBN 0-12-471252-5.

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

A. O'Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society, B*, 40:1–42, 1978.

A. O'Hagan. Some Bayesian numerical analysis. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 4*, pages 345–363, Valencia, 1992. Oxford University Press.

J. Quiñonero Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. ISBN 0-262-18253-X.

S. T. Roweis. EM algorithms for PCA and SPCA. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 626–632, Cambridge, MA, 1998. MIT Press.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. doi: 10.1126/science.290.5500.2323.

J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, 1969. doi: 10.1109/T-C.1969.222678.

G. Sanguinetti, M. Milo, M. Rattray, and N. D. Lawrence. Accounting for probe-level noise in principal component analysis of microarray data. *Bionformatics*, 21(19):3748–3754, 2005. doi: 10.1093/bioinformatics/bti617.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2001.

A. P. Shon, K. Grochow, A. Hertzmann, and R. P. N. Rao. Learning shared latent structure for image synthesis and robotic imitation. In Weiss et al. [2006].

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Weiss et al. [2006].

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. doi: 10.1126/science.290.5500.2319.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6(3):611–622, 1999. doi: doi:10.1111/1467-9868.00196.

R. Urtasun, D. J. Fleet, A. Hertzmann, and P. Fua. Priors for people tracking from small training sets. In *IEEE International Conference on Computer Vision (ICCV)*, pages 403–410, Bejing, China, 17–21 Oct. 2005. IEEE Computer Society Press. doi: 10.1109/ICCV.2005.193.

J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. In Weiss et al. [2006].

K. Q. Weinberger, F. Sha, and L. K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In R. Greiner and D. Schuurmans, editors, *Proceedings of the International Conference in Machine Learning*, volume 21, pages 839–846. Omnipress, 2004.

Y. Weiss, B. Schölkopf, and J. C. Platt, editors. *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.

C. K. I. Williams. Computing with infinite networks. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, Cambridge, MA, 1997. MIT Press.

C. K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan, editor, *Learning in Graphical Models*, volume 89 of *Series D: Behavioural and Social Sciences*. Kluwer, Dordrecht, The Netherlands, 1998.

C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In D. Touretzky, M. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 514–520, Cambridge, MA, 1996. MIT Press.