Local Distance Preservation in the GP-LVM through Back Constraints

Neil D. Lawrence NEIL@DCS.SHEF.AC.UK
Dept of Computer Science, University of Sheffield, Regent Court, 211 Portobello Street, Sheffield, S1 4DP, U.K.

Joaquin Quiñonero-Candela JOAQUIN@FIRST.FRAUNHOFER.DE
Technical University of Berlin, Electr. Eng. and Computer Science, Franklinstr. 28/29, D-10587 Berlin, Germany
Fraunhofer FIRST.IDA, Kekuléstr. 7, D-12489 Berlin, Germany

Abstract

The Gaussian process latent variable model (GP-LVM) is a generative approach to non-linear low dimensional embedding, that provides a smooth probabilistic mapping from latent to data space. It is also a non-linear generalization of probabilistic PCA (PPCA) (Tipping & Bishop, 1999). While most approaches to non-linear dimensionality methods focus on preserving local distances in data space, the GP-LVM focusses on exactly the opposite. Being a smooth mapping from latent to data space, it focusses on keeping things apart in latent space that are far apart in data space. In this paper we first provide an overview of dimensionality reduction techniques, placing the emphasis on the kind of distance relation preserved. We then show how the GP-LVM can be generalized, through back constraints, to additionally preserve local distances. We give illustrative experiments on common data sets.

1. Introduction

Principal component analysis (PCA) is perhaps the most widely used technique for obtaining a lower dimensional representation of a data set. The PCA algorithm can be motivated in several different ways: seeking orthogonal linear projections of the data with maximum variance, seeking a linear embedding of the data which is optimal under linear reconstruction for a quadratic loss (Jolliffe, 1986); as classical multidimensional scaling (CMDS) where both the latent space and the data space distances are Euclidean (Mardia et al., 1979). More recently, PCA has also been motivated as the maximum likelihood solution to a linear Gaussian latent variable model (Tipping & Bishop, 1999).

The Gaussian process latent variable model (GP-LVM), proposed by Lawrence (2005), is a fully probabilistic, non-linear, latent variable model that generalises principal component analysis. The model was inspired by the observation that a particular probabilistic interpretation of PCA is a product of Gaussian process models each with a *linear* covariance function. Through consideration of non-linear covariance functions a non-linear latent variable model can be constructed.

An important characteristic of the GP-LVM is the ease and accuracy with which probabilistic reconstructions of the data can be made, given a (possibly new) point in the latent space. This characteristic is exploited in several of the successful applications of the GP-LVM: learning style from motion capture data (Grochow et al., 2004) and learning a prior model for tracking (Urtasun et al., 2005). Implicitly the GP-LVM learns a mapping between the latent space and the data space. This mapping will typically be *smooth*.[1] This is a characteristic shared with other probabilistic, non-linear latent variable models. Density networks (MacKay, 1995) and the Generative Topographic Mapping (GTM) (Bishop et al., 1998). Both make use of smooth mappings from the latent space to the data space.

1.1. Local Distance Preservation

A popular perspective for dimensionality reduction approaches is to consider how the low dimensional representation preserves the distances between points in the

[1] Here, by smooth, we mean that points in latent space which are 'close' will be mapped to points in data space which are also 'close'. Many different covariance functions can be employed with the Gaussian process to ensure this.

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

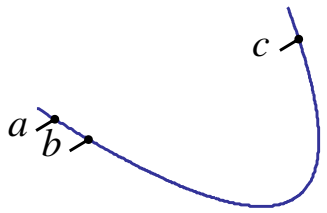


Figure 1. Dimensional reduction of a one dimensional object in two dimensions. The intuition is that we would rather preserve smaller distances than larger distances.

original data. It is often argued that, for dimensionality reduction, we should be more interested in preserving the distances between close together objects than distant objects. Figure 1 illustrates the intuition behind this argument. The figure shows a smooth curve in two dimensions, which we consider to be the high dimensional data. If we wish to recover the low dimensional structure (in this case the one dimensional line) it makes sense to preserve distances of points that are close together (such as a and b). These close together points better reflect the true distance along the underlying object. Points which are further apart (such as a and c) are less representative of the distance along the underlying object.

There are dangers involved however, in focussing too much attention on preserving local distances. First, if not enough effort is put into preserving large distances, the resulting embedding might 'overlap': objects that were distant in data space end up close in latent space. Second, focussing too much on small distances will lead to large sensitivity to any noise in the high dimensional data. Too much noise might even lead to complete failure in capturing the underlying manifold structure.

1.2. Dissimilarity Preservation

The GP-LVM ensures a smooth mapping from latent to data space. However, this mapping does not guarantee that local distances in data space will be preserved in the latent space. On the contrary, it guarantees that two points which are 'distant' in data space cannot be placed too close together in latent space: this would imply a discontinuity in the mapping. So in some sense the GP-LVM is *dissimilarity preserving*. Note, that the precise distance between far away points is not necessarily preserved in latent space: two such points will simply not be close together: 'far away' is just 'far away'. As we discuss in Section 4, this dissimilarity preservation property is shared by density networks and the GTM. The GP-LVM does not constrain nearby points in data space to be close in latent space. It is possible that such a model will have higher likelihood, but even this will not necessarily be the case as we will see in the motion capture example in Section 6.1.

Unfortunately, when performing dimensionality reduction, it is often not possible to accurately preserve both local distances and dissimilarities. The question naturally arises, of which of the two different approaches is more correct: putting more emphasis on preserving local distances, or on preserving dissimilarities? In practice, the best approach will very much depend on the data set. However, if we believe that local distance preservation is important, but we still want a probabilistic model there is currently no obvious approach. In this paper we introduce the back constrained GP-LVM, where the likelihood is optimised with the constraint of local distance preservation. This constraint is imposed through the form of a mapping from the data space to the latent space. In effect, we therefore have two models in action simultaneously: a dissimilarity preserving, probabilistic GP-LVM mapping from latent to data space, and a local distance preserving mapping from data to latent space.

2. Dimensionality Reduction

Multidimensional scaling (MDS) is the generic statistical denomination for data reduction methods that operate through matching distances, or similarities, in the observed and latent spaces. Typically these approaches define *stress functions* which evaluate the quality of the match between the distances in latent space and the distances in data space (implying rotation and translation invariance). Perhaps the simplest such stress function is the squared difference between distances in latent and distances in data space,

$$S = \sum_{n=1}^{N} \sum_{m=n+1}^{N} \left(\delta_{mn} - d_{mn}\right)^2, \qquad (1)$$

where we have N data points and the distances between points in latent space are given by δ_{mn}, while the distances between points in data space are given by d_{mn}.

If the distances in latent space are Euclidean, $\delta_{mn}^2 = (\mathbf{x}_m - \mathbf{x}_n)^{\mathrm{T}} (\mathbf{x}_m - \mathbf{x}_n)$, where the points in the latent space are given by $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N]^{\mathrm{T}}$, then it is common to seek the zero error point with respect to $\mathbf{X}$ for this cost function through an eigenvalue problem[2] (Mardia et al., 1979). Furthermore if the data is vectorial and its distance function is also Euclidean, $d_{mn}^2 =$

[2]If after retaining q eigenvectors the residual variance is zero, the resulting $\mathbf{X}$ is also a zero error solution to (1).

$(\mathbf{y}_m - \mathbf{y}_n)^{\mathrm{T}} (\mathbf{y}_m - \mathbf{y}_n)$, where the *centred* points from the data space are given by $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_N]^{\mathrm{T}}$, then the eigenvalue problem is of the form

$$\mathbf{K}_Y \Lambda = \Lambda \mathbf{U},$$

where $\mathbf{K}_Y = \mathbf{Y}\mathbf{Y}^{\mathrm{T}}$ is the inner product matrix (which is a form of similarity matrix) and the latent positions are then given by $\mathbf{X} = \mathbf{U}\Lambda^{\frac{1}{2}}$. The form of this eigenvalue problem can be easily shown to be equivalent to that solved in PCA (see, for example, Lawrence, 2005). More generally, replacing $\mathbf{K}_Y$ with any positive definite similarity matrix, implies a feature space within which the Euclidean distances are being measured,

$$d_{mn} = (\phi(\mathbf{y}_m) - \phi(\mathbf{y}_n))^{\mathrm{T}} (\phi(\mathbf{y}_m) - \phi(\mathbf{y}_n)).$$

If this matrix is constructed through a *Mercer kernel* then the approach is recognised as kernel PCA (KPCA) (Schölkopf et al., 1998), which implies a (typically smooth) mapping from the data to the feature space,

$$f(\mathbf{y}_n) = \sum_{m=1}^{M} \alpha_m \phi(\mathbf{y}_m)^{\mathrm{T}} \phi(\mathbf{y}_n).$$

Note that KPCA preserves local similarities, since the smooth mapping is from data to latent space.

Many CMDS practitioners also use similarity matrices which are non-positive definite. These are equivalent to non-Euclidean distances and, indeed are often constructed from distances which are non-Euclidean. A classic example is the distance matrix generated by the Isomap algorithm (Tenenbaum et al., 2000).

3. Local Distance Preservation

The objective function (1) seeks a configuration for which all distances are considered, regardless of their relative magnitude. However, it can be modified to force an algorithm to focus more on local distances. In the Sammon mapping[3] (Sammon, 1969) an alternative stress is used, one which can be written as a weighted sum of squares,

$$S = \sum_{n=1}^{N} \sum_{m=n+1}^{N} w_{mn} (\delta_{mn} - d_{mn})^2, \qquad (2)$$

where the weights are proportional to the inverse distance in data space,

$$w_{mn} \propto d_{mn}^{-1}.$$

[3] 'Sammon mapping' is perhaps a misnomer as, in its standard form, there is no explicit mapping associated with the algorithm.

The weights reduce the contribution of entries in the error matrix with large d_{mn} forcing the algorithm to focus on matching more local distances. Unfortunately such a modification of the stress function comes with a cost: the zero error point solution can no longer be found through an eigenvalue problem. Instead, an iterative optimisation of this non-convex cost function must be attempted.

3.1. SDE, LLE, SNE and Isomap

Most of the recent work in machine learning has been on preserving local distances in the latent space. For example the locally linear embedding (LLE) (Roweis & Saul, 2000) seeks an embedded space for which locally linear relationships are preserved. The Isomap algorithm (Tenenbaum et al., 2000) computes an approximation to geodesic distance through constructing neighbourhood graph and the semidefinite embedding (Weinberger et al., 2004) maximises the variance in the latent space with a constraint that local distances should be preserved. These methods share a common thread with the Sammon mapping in that they are interested in preserving local distances. However, they achieve their aim while maintaining the unimodality of solutions associated with CMDS. In the case of SDE and Isomap this is achieved through explicitly using the CMDS solution for obtaining the visualisation. The innovation is in how the distance matrix is developed. The distance matrix is designed to reflect local distances: in the case of Isomap through construction of a graph based on local distances and for the SDE by adapting the distance matrix such that non-local distances are explicitly maximised while forcing the preservation of local distances. Stochastic neighbour embedding (SNE) was proposed by Hinton and Roweis (2003) as "an improvement over methods like LLE or SOM in which widely separated data-points can be 'collapsed' as near neighbors in the low-dimensional space". SNE does not provide a probabilistic mapping, but rather an information theoretic objective function which is easily interpretable and expandable (Zien & Quiñonero Candela, 2005).

4. Probabilistic Dimensionality Reduction

A difficulty with relying on local distances when constructing a low dimensional embedding is sensitivity to noise. If the data of interest lies precisely on a low dimensional manifold the local distances will typically be reliable, however if the low dimensional manifold is corrupted by high dimensional noise, such local distances can become unreliable. In this context it *may*

be more sensible attach less importance to local distances. An alternative class of algorithms model the data probabilistically as a lower dimensional manifold. One feature of these algorithms is that they are, perhaps surprisingly, more constrained by the non-local distances than by the local ones.

4.1. Probabilistic Approaches

The probabilistic approach to dimensionality reduction is to formulate a latent variable model, where the latent dimension, q, is lower than the data dimension, d. The latent space is then governed by a prior distribution $p(\mathbf{X})$. The latent variable is related to the observation space through a probabilistic mapping,

$$y_{ni} = f_i(\mathbf{x}_n) + \epsilon_n,$$

where y_{ni} is the ith feature of the nth data point and ϵ_n is a noise term that is typically taken to be Gaussian,[4] $p(\epsilon_n) = N\left(\epsilon_n|0, \beta^{-1}\right)$. If the prior is taken to be independent across data points the marginal likelihood of the data can be written as

$$p(\mathbf{Y}) = \int \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{x}_n)\, p(\mathbf{x}_n)\, \mathrm{d}\,\mathbf{X},$$

where $p(\mathbf{y}_n|\mathbf{x}_n) = \prod_{i=1}^{d} N\left(y_{in}|f_{in}(\mathbf{x}_n), \beta^{-1}\right)$. If the mapping is chosen to be linear, $f_i(\mathbf{x}_n) = \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_n$, and the prior over the latent variables is taken to be Gaussian, then the maximum likelihood solution of the model spans the principal subspace of the data (Tipping & Bishop, 1999). However if the mapping is non-linear it is unclear, in general, how to propagate the prior distributions uncertainty through the non-linearity. One suggested approach is to make use of point based representations of the latent space either through sampling, such as in density networks (MacKay, 1995), or through an explicitly point based representation of the latent space, such as the generative topographic mapping (Bishop et al., 1998). Both approaches are strongly related to each other.

4.1.1. The GP-LVM

An alternative suggestion given in (Lawrence, 2004; Lawrence, 2005) is to place the prior distribution over the mappings rather than the latent variables. The mappings may then be marginalised and the marginal likelihood optimised with respect to the latent variables,

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{i=1}^{d}\prod_{n=1}^{N} p(y_{in}|f_{in})\, p(\mathbf{f}|\mathbf{X}). \qquad (3)$$

[4] We denote a Gaussian distribution over $\mathbf{z}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ by $N(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

One motivation for the approach was provided by the fact that if the prior is taken to be a Gaussian process which is independent across data dimensions and has a *linear* covariance function (thus restricting the mappings to linearity) the maximum likelihood solution with respect to the embeddings is given by principal component analysis. Thus the algorithm provides an alternative probabilistic model for PCA to that given by (Tipping & Bishop, 1999). However, if the covariance function is one which allows non-linear functions (*e.g.* the RBF kernel) then the model provides a probabilistic non-linear latent variable model. This approach is known as the Gaussian process latent variable model.

4.1.2. Probabilistic Models and Locality

Each of the probabilistic models mentioned above imposes some constraints on the mapping function from the latent space to the data space. In the case of density networks the mapping is given by a multi-layer perceptron and for the GTM an RBF network is used. For the GP-LVM a prior over functions constrains the type of map. For the purposes of this discussion we will assume that these mappings are smooth. In general terms, by smooth, we mean that if two points, $\mathbf{x}_n$ and $\mathbf{x}_m$, are separated by a 'small distance' (relative to distances between the other points in the space) in latent space then the positions they map to, $\mathbf{y}_n$ and $\mathbf{y}_m$, will also be separated by a 'small distance'. This smoothness is an inherent constraint in all these models. It implies that no two points which are 'far apart' in data space can be embedded as 'close together' in latent space. However, it is not in line with the locality preserving constraints we discussed earlier. There is nothing in the constraint to prevent two points which are close in data space being far apart in latent space. (There is at most a mild encouragement from the likelihood function.) While this is not in line with the intuition illustrated by Figure 1, it is likely to be a more robust approach when the manifold is corrupted by high dimensional noise. As mentioned in Sect. 1.1, neither constraint can necessarily be considered more correct: the better approach is dependent on the data to hand. However, the very fact that the latter models are probabilistic can be viewed as an advantage in itself. Indeed the successful applications and extensions of the GP-LVM (Grochow et al., 2004; Urtasun et al., 2005; Shon et al., 2006; Wang et al., 2006) each take explicit advantage of the fact that the model is probabilistic. In the next section we will show how the GP-LVM can be modified to preserve locality leading to a fully probabilistic, locality-preserving, non-linear dimensional reduction technique.

5. Back Constraints

The smooth mapping in the Gaussian process latent variable model ensures that distant points in data space remain distant in latent space. However there are several algorithms that lead to a smooth mapping in the opposite direction, from data space to latent space, *e.g.* kernel PCA (Schölkopf et al., 1998) and the neuroscale algorithm (Lowe & Tipping, 1996). In kernel PCA the mapping is implicit and arises through the use of the kernel function. A key advantage of kernel PCA over other eigenvalue based techniques is that this mapping arises as a side effect. It is interesting to note that kernel PCA can also be obtained by inverting the rôles of inputs and outputs in the GP-LVM: maximizing the marginal likelihood (3) with respect to the latent variables (outputs of the GP-LVM) yields kernel PCA. In neuroscale an alternative approach is taken: the mapping is explicitly included by constraining the latent points to be a function of the input points,[5]

$$x_{nj} = g_j\left(\mathbf{y}_n; \mathbf{w}\right). \tag{4}$$

A stress function such as (1) or (2) can then be minimised with respect to the parameters of the mapping rather than the latent points themselves. Constraining the latent points to be a smooth mapping from data space forces small distances in data space to be small in latent space, in line with the locality constraints imposed by the algorithms mentioned above.

5.1. Constrained Maximum Likelihood

Applying the approach taken in the neuroscale algorithm to the GP-LVM is straightforward. Rather than maximising the likelihood (3) with respect to $\mathbf{X}$ directly, we replace each element of $\mathbf{X}$ with a mapping of the form given in (4). Two points in latent space will then be constrained to always be close if their data space counterparts are close. Instead of direct likelihood maximisation, we now maximize a constrained likelihood, the constraints preserving nearby 'localities'. How close is 'nearby' is determined by the smoothness of the mapping. For example if the mapping is kernel based using an RBF kernel,

$$g_j\left(\mathbf{y}_n\right) = \sum_{m=1}^{N} \alpha_{jm} k\left(\mathbf{y}_n, \mathbf{y}_m\right),$$

where $\mathbf{A} = \{\{\alpha_{jn}\}_{n=1}^{N}\}_{j=1}^{q}$ are the parameters, and the kernel matrix is,

$$k\left(\mathbf{y}_n, \mathbf{y}_m\right) = \exp\left(-\frac{\gamma}{2}\left(\mathbf{y}_n - \mathbf{y}_m\right)^{\mathrm{T}}\left(\mathbf{y}_n - \mathbf{y}_m\right)\right), \tag{5}$$

closeness is determined by the setting of the inverse width parameter γ. On the other hand, if the mapping is given by a multi-layer perceptron,

$$g_j\left(\mathbf{y}\right) = v_{ij} \sum_{i=1}^{h} \sigma\left(\mathbf{u}_i^{\mathrm{T}}\mathbf{y}\right),$$

where

$$\sigma\left(z\right) = \frac{1}{1+\exp\left(-z\right)}$$

is the sigmoid function and the parameters are given by $\{\{v_{ij}\}_{i=1}^{h}\}_{j=1}^{q}$ then the closeness is controlled by the number of hidden units used, h. One elegant feature of the approach is that any mapping can be used: the only requirement is that the derivatives of the outputs with respect to the parameters can be computed so that the likelihood can be maximised by gradient based methods. We refer to this constrained version of the GP-LVM as the GP-LVM with back constraints.

6. Experiments

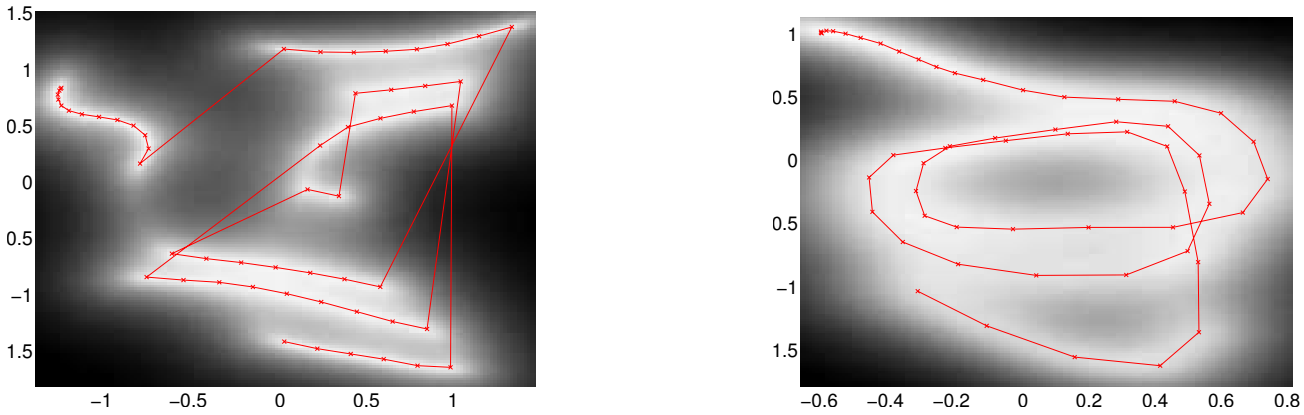
We analyze the GP-LVM with back constraints, and compare it to unconstrained GP-LVM, and to Isomap.

6.1. 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] The dimension of the data is 102, from the three coordinates of each of the 34 markers. 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. Our experimental set up was as follows. We trained both an unconstrained and a back constrained GP-LVM with an RBF covariance function. The back constraint was implemented through an RBF based kernel mapping (5), with $\gamma = 1 \times 10^{-3}$. 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 mapping's 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

[5] For the neuroscale algorithm a radial basis function network or multi-layer perceptron are suggested but other mappings such as a kernel based could equally be applied.

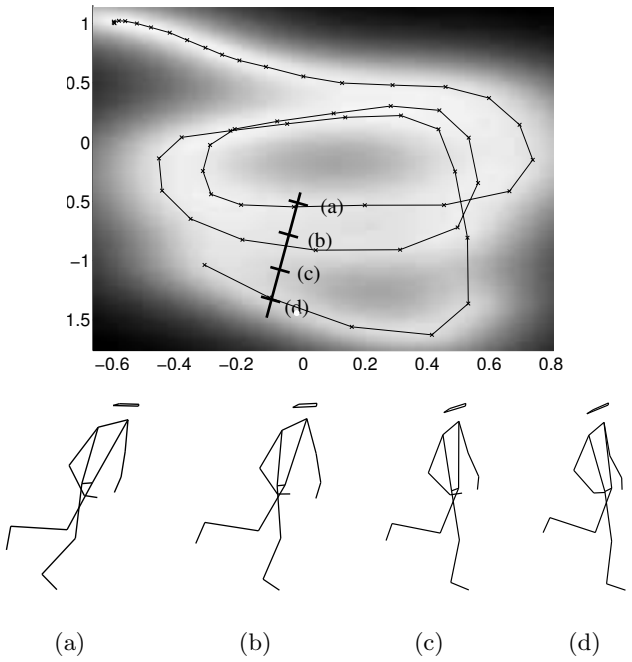
[6] Data made available by the Ohio State University Advanced Computing Center for the Arts and Design, available from `http://accad.osu.edu/research/mocap/mocap_data.htm`, sequence 'Figure Run 1' in `.txt` format.



(a) (b)

Figure 2. Visualisation of the motion capture data. (a) The regular GP-LVM, log likelihood 1,543 and (b) the GP-LVM with back constraints, 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 indicates the precision with which the mapping is expressed.

conjugate gradients. Scripts for running these experiments are available on line, see Appendix.



(a) (b) (c) (d)

Figure 3. 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.

The visualisation results are shown in Figure 2. The data is temporal in nature (although the models do not take advantage of this fact) and we have connected points in the plots that are neighbours in time. In Figure 2(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. (This is *not* due to overfitting, since the model provides a smooth representation of the data which generalises well across the latent space.) 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 the data. The higher likelihood solution turns out to be placing points far apart which are actually close together. The problem arises because the latent space is too constrained. Using a three dimensional latent space alleviates the problem, (not shown here due to space considerations, but a script to run the experiment is available on line) 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 reflecting 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 3.

Table 1. Nearest neighbour errors in latent space for the vowels data (in data space 24 errors).

Method	Isomap	GP-LVM	BC-GP-LVM
Errors	458	226	155

6.2. 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, used as the initialisation, fails to separate the data at all. As a result the non-back constrained model tends to fragment the different vowels.

We present results using the Isomap (Fig. 4(a)), the GP-LVM (Fig. 4(b)) and the back constrained GP-LVM (Fig. 4(c)). The GP-LVM obtains good separation between the vowels, but does not maintain the neighbourhood relations. Isomap preserves neighbourhood relations, but with severe overlap, particularly between */u/*, */o/*, */ae/* and */ao/*. The back constrained GP-LVM obtains good separation between the different vowels, while keeping neighbourhood structure. Table 1 offers a quantitative comparison.

7. Discussion

We have reviewed dimensionality reduction from the perspective of distance preservation. Emphasizing local distances preservation is a very common paradigm. This paradigm is not followed by probabilistic approaches to dimensionality reduction, which instead preserve dissimilarities. Inspired by the neuroscale algorithm, we have shown how to introduce locality preservation in the Gaussian process latent variable model. We constrain the latent variables to be generated by a parametric "backwards" mapping, from data space to latent space, the parameters of which are learnt by maximizing the marginal likelihood of the GP-LVM. We end up with two mappings, one top-down (GP-LVM), and another bottom-up (back constraint), which is somewhat reminiscent of the wake-sleep algorithm (Hinton et al., 1995). We illustrated the advantages of adding locality preservation to the GP-LVM on a small motion capture data set as well as on a larger vowel data set.

Acknowledgements

The ideas in this paper were formulated during a visit by JQC to the University of Sheffield funded by the

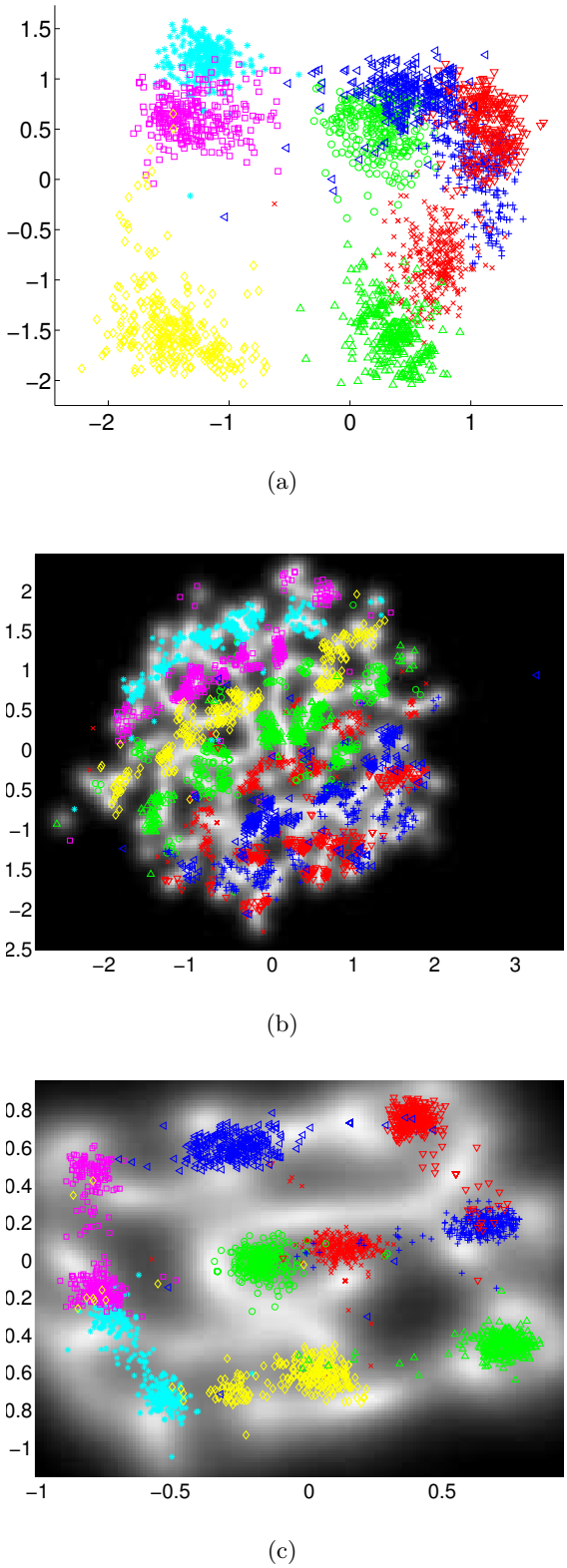


Figure 4. Visualisation of the vowel data using (a) Isomap with 7 neighbours, (b) the GP-LVM and (c) the back constrained GP-LVM. The different vowels are shown as follows: */a/* cross */ae/* circle */ao/* plus */e/* asterix */i/* square */ibar/* diamond */o/* down triangle */schwa/* up triangle and */u/* left triangle.

EU FP6 PASCAL Network of Excellence in June 2005. Thanks to Jon Malkin for providing the vowel data used. This work was partly done while JQC was with the Max Planck Institute for Biological Cybernetics.

A. Recreating the Experiments

The source code for re-running all the experiments detailed here is available from `http://www.dcs.shef.ac.uk/~neil/fgplvm/`, release 0.132. The motion capture results can be recreated by `demStick1.m` and `demStick3.m`. The vowels results can be recreated with `demVowelsIsomap.m`, `demVowels2.m` and `demVowels3.m`.

References

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

Bishop, C. M., Svensén, M., & Williams, C. K. I. (1998). GTM: the Generative Topographic Mapping. *Neural Computation*, *10*, 215–234.

Grochow, K., Martin, S. L., Hertzmann, A., & Popovic, Z. (2004). Style-based inverse kinematics. *ACM Trans. on Graphics (SIGGRAPH 2004)*.

Hinton, G., & Roweis, S. (2003). Stochastic neighbor embedding. *Advances in Neural Information Processing Systems 15*. Cambridge, MA: MIT Press.

Hinton, G. E., Dayan, P., Frey, B. J., & Neal, R. M. (1995). The wake-sleep algorithm for unsupervised neural networks. *Science*, *268*, 1158–1161.

Jolliffe, I. T. (1986). *Principal component analysis*. New York: Springer-Verlag.

Lawrence, N. D. (2004). Gaussian process models for visualisation of high dimensional data. *Advances in Neural Information Processing Systems* (pp. 329–336). Cambridge, MA: MIT Press.

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

Lowe, D., & Tipping, M. E. (1996). Feed-forward neural networks and topographic mappings for exploratory data analysis. *Neural Computing and Applications*, *4*.

MacKay, D. J. C. (1995). Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research, A*, *354*, 73–80.

Mardia, K. V., Kent, J. T., & Bibby, J. M. (1979). *Multivariate analysis*. London: Academic Press.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, *C-18*, 401–409.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299–1319.

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

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

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

Urtasun, R., Fleet, D. J., Hertzmann, A., & Fua, P. (2005). Priors for people tracking from small training sets. *IEEE International Conference on Computer Vision (ICCV)* (pp. 403–410). Bejing, China: IEEE Computer Society Press.

Wang, J. M., Fleet, D. J., & Hertzmann, A. (2006). Gaussian process dynamical models. In (Weiss et al., 2006).

Weinberger, K. Q., Sha, F., & Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. *Proceedings of the International Conference in Machine Learning* (pp. 839–846). San Francisco, CA: Morgan Kauffman.

Weiss, Y., Schölkopf, B., & Platt, J. C. (Eds.). (2006). *Advances in neural information processing systems*, vol. 18. Cambridge, MA: MIT Press.

Zien, A., & Quiñonero Candela, J. (2005). Large margin non-linear embedding. *Proceedings of the 22nd International Conference on Machine Learning* (pp. 1065–1072). San Francisco, CA: Morgan Kaufmann.