Local Distance Preservation in the GP-LVM through Back Constraints

Constraining Probabilistic Latent Variable models to reflect Local Distances

Neil D. Lawrence[1] Joaquin Quiñonero-Candela[2]

Tuesday, 27th June 2006

Outline

Online Resources

All source code and slides are available online

- This talk available from my home page (see talks link on side).
- MATLAB examples in the 'fgplvm' (vrs 0.132) and 'oxford' (vrs 0.13) toolbox .
  - `http://www.dcs.shef.ac.uk/~neil/fgplvm/`.
  - `http://www.dcs.shef.ac.uk/~neil/oxford/`.
- MATLAB commands used for examples given in `typewriter font`.

Dimensional Reduction

Dealing with High Dimensional Data

- Many machine learning problems involve high dimensional data.
- Learning in *true* high dimensional requires exponentially many data points.
- Fortunately, in practice, many high dimensional data sets are often intrinsically low dimensional.
- Seek to deal with data by representing a high dimensional data set[a] $\mathbf{Y} \in \Re^{n \times k}$ as a low dimensional matrix $\mathbf{X} \in \Re^{n \times q}$ where $q \ll k$.

[a] Here $\mathbf{Y}$ and $\mathbf{X}$ have the form of design matrices. This means that $\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$ is an inner product matrix and, for centred $\mathbf{Y}$, $\frac{1}{n}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$ is a covariance matrix.

Statistical Approach

Multi-dimensional Scaling (MDS)

- Construct matrix of distances in data space, then:
  - either use spectral techniques.
  - or iteratively minimise a 'stress function' for matching distances, *e.g.*,

$$S = \sum_{j=1}^{n} \sum_{i=1}^{j-1} \left(\delta_{ij} - d_{ij}\right)^2 .$$

$$\delta_{ij} = ||\mathbf{x}_i - \mathbf{x}_j|| \qquad d_{ij} = ||\mathbf{y}_i - \mathbf{y}_j||$$

latent space distance — data space distance

Machine Learning Approaches

Spectral Approaches

- Classical MDS
  - Semi-definite embedding: places constraints on nearby distances [Weinberger et al., 2004].
  - Isomap: constructs an approximation to geodesic distance [Tenenbaum et al., 2000].
- Kernel PCA
  - Genereally doesn't reduce dimension (certainly not with an RBF kernel) [Schölkopf et al., 1998].
- Locally Linear Embeddings [Roweis and Saul, 2000].
- Probabilistic Approaches: GTM [Bishop et al., 1998] and Density Networks [MacKay, 1995].

Preserving Distances

Local Distance Preservation

- Most of the non-probabilistic approaches seek to preserve local distances in the latent space.

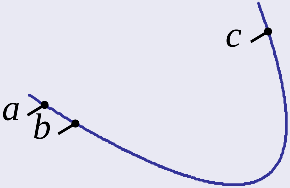


Figure: Local Distance preservation. Preserve distance between (a) and (b) but not between (a) and (c).

Gaussian Processes

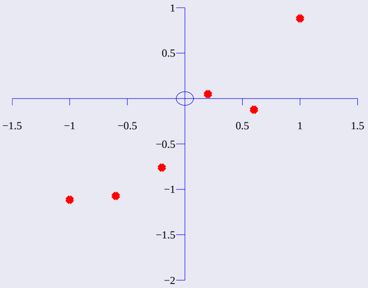
Inference about functions

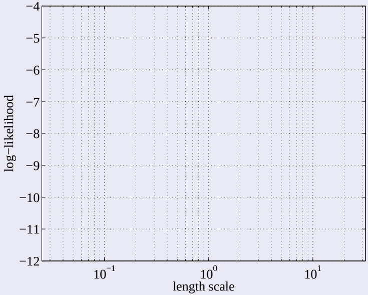
- Gaussian Processes (GPs) are probabilistic models for functions, $p(\mathbf{f}|\mathbf{X})$. [O'Hagan, 1978, 1992, Rasmussen and Williams, 2006]
- GPs allow inference about functions in the presence of uncertainty.
- They are ideal for the domain of regression.
- Probabilistic version of kernel regression: kernel parameters can be determined by data.

Learning Kernel Parameters

Adapting the Covariance function to Data

demOptimiseKern (in oxford toolbox vrs 0.13)

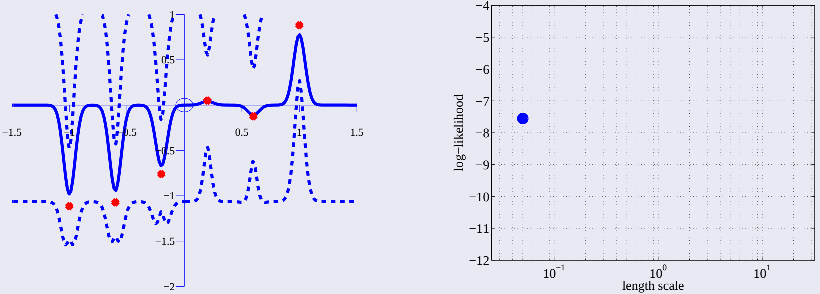




Learning Kernel Parameters

Adapting the Covariance function to Data

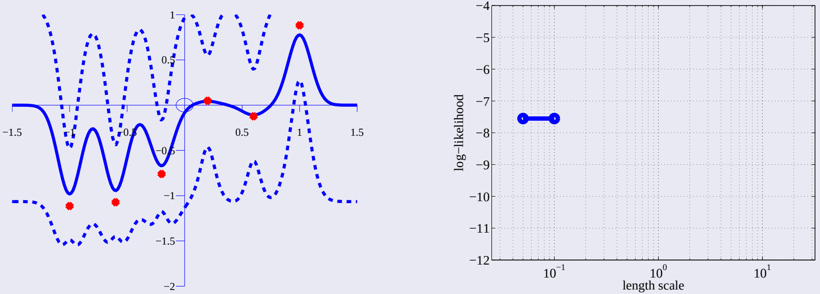
`demOptimiseKern` (in oxford toolbox vrs 0.13)



Learning Kernel Parameters

Adapting the Covariance function to Data

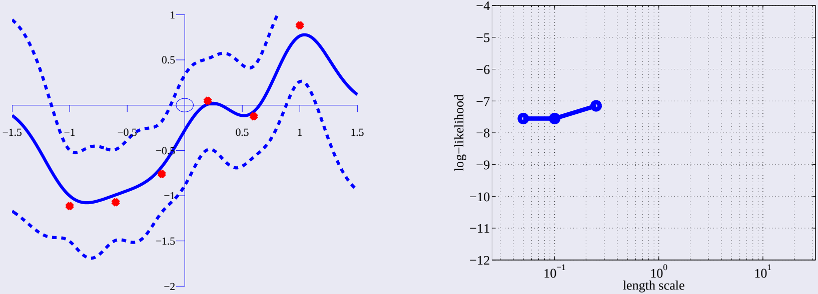
`demOptimiseKern` (in oxford toolbox vrs 0.13)



Learning Kernel Parameters

Adapting the Covariance function to Data

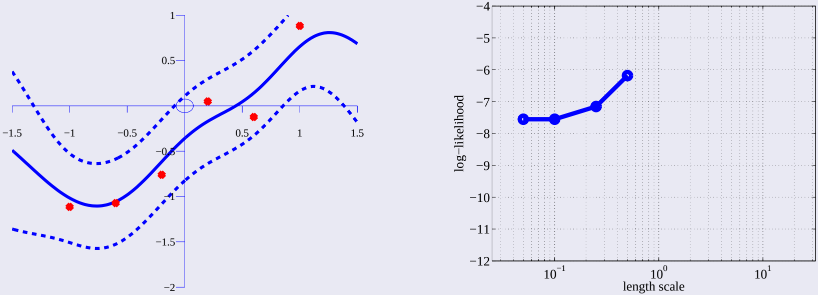
`demOptimiseKern` (in oxford toolbox vrs 0.13)



Learning Kernel Parameters

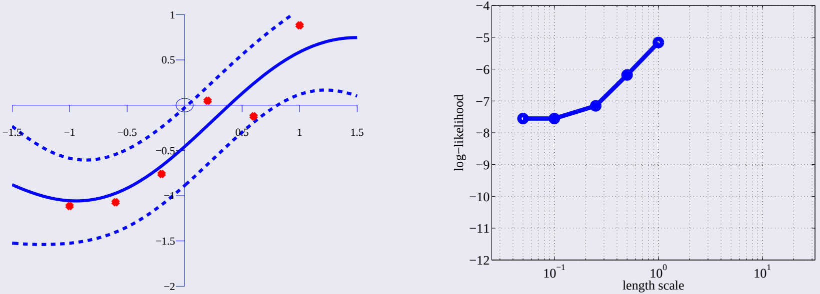
Adapting the Covariance function to Data

`demOptimiseKern` (in oxford toolbox vrs 0.13)



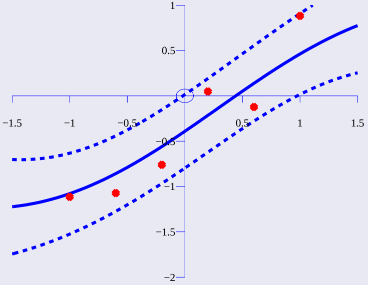
Learning Kernel Parameters
Adapting the Covariance function to Data

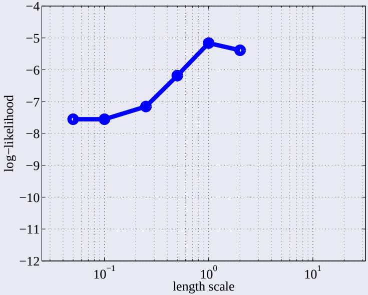
demOptimiseKern (in oxford toolbox vrs 0.13)



Learning Kernel Parameters
Adapting the Covariance function to Data

`demOptimiseKern` (in oxford toolbox vrs 0.13)

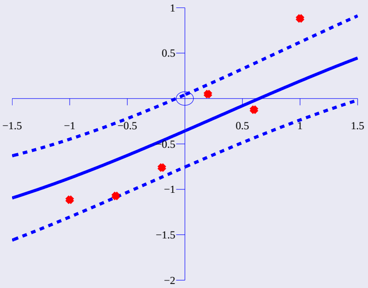


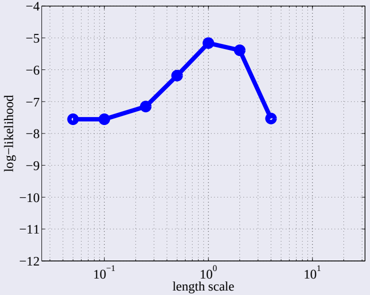


Learning Kernel Parameters

Adapting the Covariance function to Data

demOptimiseKern (in oxford toolbox vrs 0.13)

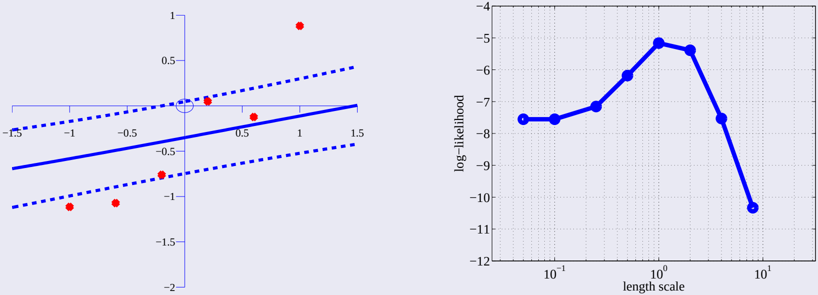




Learning Kernel Parameters

Adapting the Covariance function to Data

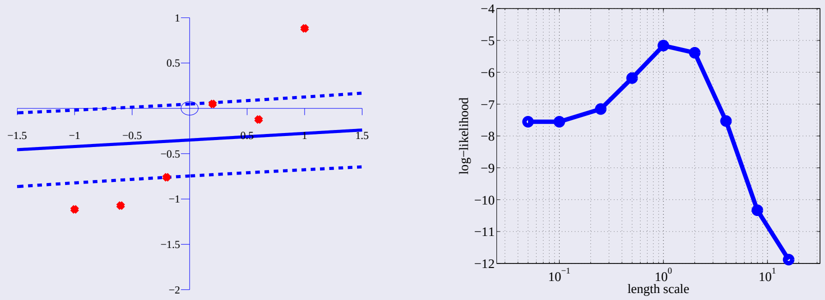
`demOptimiseKern` (in oxford toolbox vrs 0.13)



Learning Kernel Parameters

Adapting the Covariance function to Data

`demOptimiseKern` (in oxford toolbox vrs 0.13)

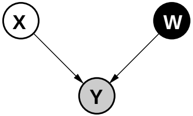


A Latent Variable Model

How can a model designed primarily for regression be used as a technique for dimensional reduction?

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - *Optimise over parameters* integrate out latent variables.
- Define Gaussian prior over *latent space*, $\mathbf{X}$.



$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} N\left(\mathbf{y}_i|\mathbf{W}\mathbf{x}_i, \sigma^2\mathbf{I}\right)$$

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

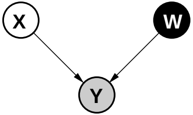
$$p(\mathbf{Y}|\mathbf{W}) = \prod_{i=1}^{n} N\left(\mathbf{y}_i|\mathbf{0},\mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}\right)$$

A Latent Variable Model

How can a model designed primarily for regression be used as a technique for dimensional reduction?

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:
    - *Optimise over parameters* integrate out latent variables.
- Define Gaussian prior over *latent space*, $\mathbf{X}$.



$$p(\mathbf{Y}|\mathbf{W}) = \prod_{i=1}^{n} N\left(\mathbf{y}_i|\mathbf{0}, \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}\right)$$

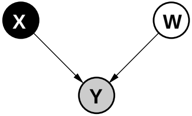
Maximum wrt $\mathbf{W}$ found from eigendecomposition of $\frac{1}{n}\mathbf{Y}^{\top}\mathbf{Y}$

A Latent Variable Model

How can a model designed primarily for regression be used as a technique for dimensional reduction?

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Novel** Latent variable approach
  - *Optimise over latent variables* integrate out parameters
- Define Gaussian prior over *parameteters*, $\mathbf{W}$.



$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} N\left(\mathbf{y}_i|\mathbf{W}\mathbf{x}_i, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{W}) = \prod_{j=1}^{k} N\left(\mathbf{w}_j|\mathbf{0},\mathbf{I}\right)$$

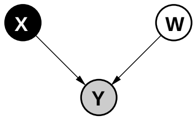
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{k} N\left(\mathbf{y}_{:,j}|\mathbf{0},\mathbf{X}\mathbf{X}^{\mathrm{T}} + \sigma^2\mathbf{I}\right)$$

A Latent Variable Model

How can a model designed primarily for regression be used as a technique for dimensional reduction?

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Novel** Latent variable approach
  - *Optimise over latent variables* integrate out parameters
- Define Gaussian prior over *parameteters*, **W**.



$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{k} N\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^{\top} + \sigma^{2}\mathbf{I}\right)$$

Maximum wrt **X** found from eigendecomposition of $\frac{1}{k}\mathbf{Y}\mathbf{Y}^{\top}$

A Latent Variable Model

How can a model designed primarily for regression be used as a technique for dimensional reduction?

Dual Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Novel** Latent variable approach
  - *Optimise over latent variables* integrate out parameters
- Define Gaussian prior over *parameteters*, **W**.

This likelihood is recognised as a product of Gaussian Processes,

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{k} N\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right),$$

with a linear kernel

$$\mathbf{K} = \mathbf{X}\mathbf{X}^{\mathsf{T}} + \sigma^2\mathbf{I}.$$

GP-LVM

Low Dimensional Manifolds for High Dimensional Data

- By replacing the linear model with a Gaussian process we obtain non-linear probabilistic PCA [Lawrence, 2005].
  - The Gaussian process gives a mapping from the low dimensional *latent* space to high dimensional data space.
- Several important applications including tracking [Urtasun et al., 2005] and graphics [Grochow et al., 2004].

Motion Capture Example

Generalization with less Data than Dimensions

- Subject runs for three paces.
- Data consists of x, y, z locations of markers.
  - 55 frames of motion capture.
  - 34 markers giving $k = 34 \times 3 = 102$.
- Data from Ohio State University
  `http://accad.osu.edu/research/mocap/mocap_data.htm`

Motion Capture Results

demStick1

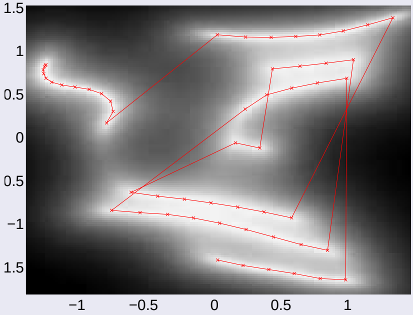


Figure: The latent space for the motion capture data. Lines connect points that are neighbours in time, temporal nature of the data not used by the algorithm (see *e.g.* Wang et al. [2006]). Note the jumps in the sequence.

Why are there Jumps?

Discontinuities in the Latent Space

1. GP-LVM gives a smooth mapping from latent to data space.
   - Points that are close in latent space will be close in data space.
   - Points close in the data space *may not* be close in latent space.
2. Kernel PCA gives a smooth mapping from data to latent space.
   - Points that are close in data space will be close in latent space.
   - Points close in the latent space *may not* be close in data space.

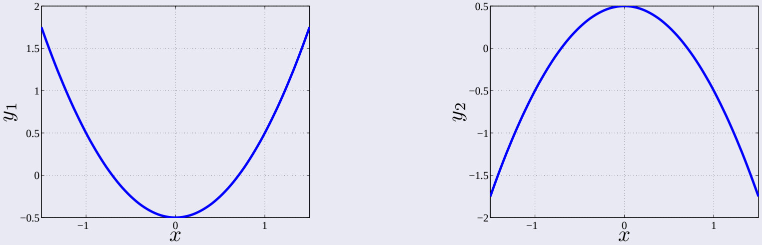
However, we can constrain the GP-LVM to force it to fulfill the second property.

Mapping in Different Directions

Forward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 1-D latent space to 2-D data space.

$$y_1 = x^2 - 0.5, \quad y_2 = -x^2 + 0.5$$

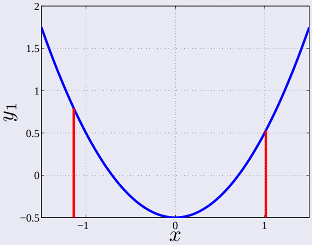


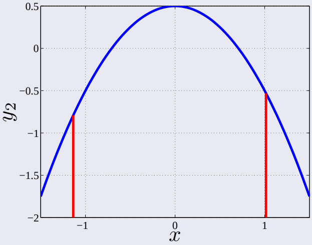
Mapping in Different Directions

Forward Mapping (demBackMapping in oxford toolbox)

- Mapping from 1-D latent space to 2-D data space.

$$y_1 = x^2 - 0.5, \quad y_2 = -x^2 + 0.5$$



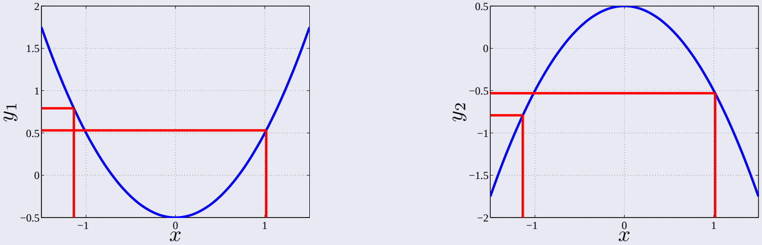


Mapping in Different Directions

Forward Mapping (demBackMapping in oxford toolbox)

- Mapping from 1-D latent space to 2-D data space.

$$y_1 = x^2 - 0.5, \quad y_2 = -x^2 + 0.5$$

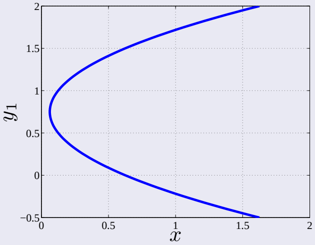


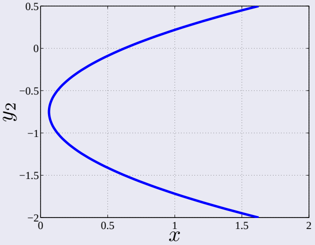
Mapping in Different Directions

Backward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 2-D data space to 1-D latent.

$$x = 0.5\left(y_1^2 + y_2^2 + 1\right)$$



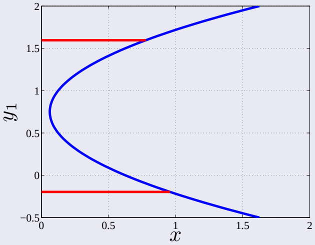


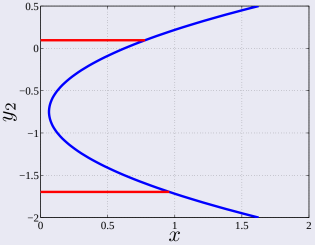
Mapping in Different Directions

Backward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 2-D data space to 1-D latent.

$$x = 0.5\left(y_1^2 + y_2^2 + 1\right)$$



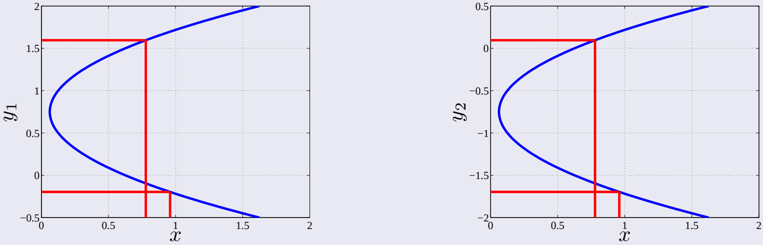


Mapping in Different Directions

Backward Mapping (`demBackMapping` in oxford toolbox)

- Mapping from 2-D data space to 1-D latent.

$$x = 0.5\left(y_1^2 + y_2^2 + 1\right)$$



NeuroScale

Multi-Dimensional Scaling with a Mapping

- Lowe and Tipping [1997] made latent positions a function of the data.

$$\delta_{ij} = \left|\left|\mathbf{x}_i - \mathbf{x}_j\right|\right|, \quad x_{ij} = f_j\left(\mathbf{y}_i; \mathbf{w}\right)$$

- Function was either multi-layer perceptron or a radial basis function network.
- Their motivation was different from ours:
  - They wanted to add the advantages of a true mapping to multi-dimensional scaling.

Back Constraints in the GP-LVM

Back Constraints

- We can use the same idea to force the GP-LVM to respect local distances.
  - By constraining each $\mathbf{x}_i$ to be a 'smooth' mapping from $\mathbf{y}_i$ local distances can be respected.
- This works because in the GP-LVM we maximise wrt latent variables, we don't integrate out.
- Can use any 'smooth' function:
  1. Neural network.
  2. RBF Network.
  3. Kernel based mapping.

Optimising BC-GPLVM

Computing Gradients

- GP-LVM normally proceeds by optimising

$$L(\mathbf{X}) = \log p(\mathbf{Y}|\mathbf{X})$$

  with respect to $\mathbf{X}$ using $\frac{dL}{d\mathbf{X}}$.
- The back constraints are of the form

$$x_{ij} = f_j(\mathbf{y}_i; \mathbf{w})$$

  where $\mathbf{w}$ are parameters.
- We can compute $\frac{dL}{d\mathbf{w}}$ via chain rule and optimise parameters of mapping.

Motion Capture Results

demStick3

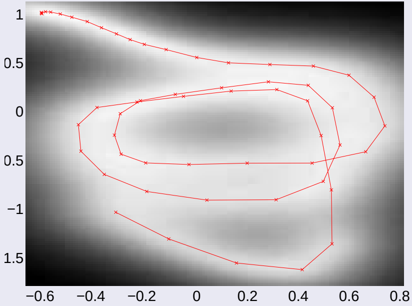
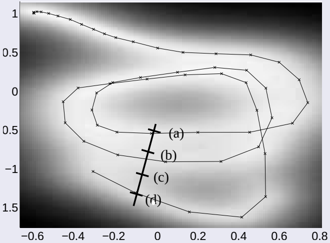
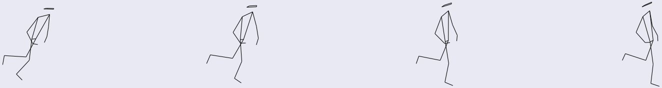


Figure: The latent space for the motion capture data with back constraints based on an RBF kernel.

Stick Man Results

demStick2





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

Projection into data space from four points in the latent space. The inclination of the runner changes becoming more upright.

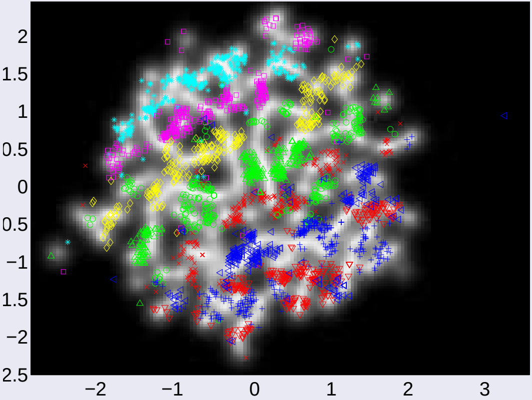
Vowel Data

Vocal Joystick Data

- Vowel sounds from a vocal joystick system [Bilmes et al., 2006].
- Vowels are from a single speaker and represented as:
  - cepstral coefficients (12 dimensions) and
  - 'deltas' (further 12 dimensions).
- 2700 data points in total (300 for each vowel).

GP-LVM Results

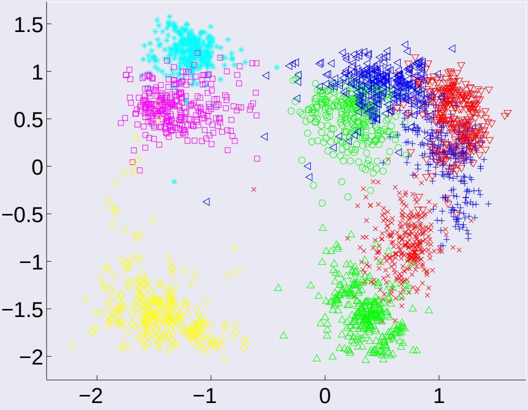
demVowels2



The different vowels are shown as follows: */a/* red cross */ae/* green circle */ao/* blue plus */e/* cyan asterix */i/* pink square */ibar/* yellow diamond */o/* red down triangle */schwa/* green up triangle and */u/* blue left triangle.

Isomap Results

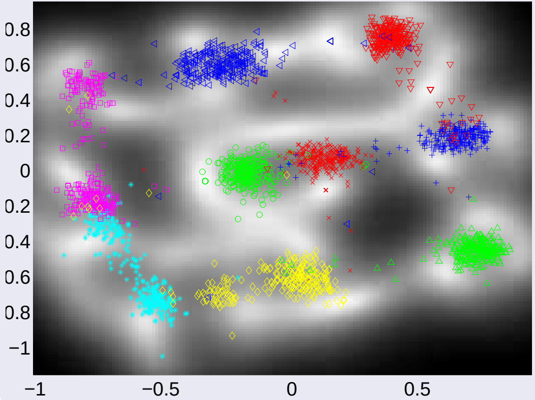
demVowelsIsomap



The different vowels are shown as follows: */a/* red cross */ae/* green circle */ao/* blue plus */e/* cyan asterix */i/* pink square */ibar/* yellow diamond */o/* red down triangle */schwa/* green up triangle and */u/* blue left triangle.

BC-GPLVM Results

demVowels3



The different vowels are shown as follows: */a/* red cross */ae/* green circle */ao/* blue plus */e/* cyan asterix */i/* pink square */ibar/* yellow diamond */o/* red down triangle */schwa/* green up triangle and */u/* blue left triangle.

1-Nearest Neighbour in $\mathbf{X}$

Comparison of the Approaches

- Nearest neighbour classification in latent space.

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

cf 24 errors in data space.

Conclusions

Summary

- Most Dimension Reduction techniques preserve local distances in the latent space.
- The GP-LVM preserves 'dissimilarities'.
- Constrained maximum likelihood forces the GP-LVM to respect local distances.

Funding

- Joaquin's visit to Sheffield was funded by the PASCAL FP6 Network of excellence.

References

Jeff Bilmes, Jonathan Malkin, Xiao Li, Susumu Harada, Kelley Kilanski, Katrin Kirchhoff, Richard Wright, Amarnag Subramanya, James Landay, Patricia Dowden, and Howard Chizeck. The vocal joystick. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*. IEEE, May 2006. To appear.

Christopher M. Bishop, Marcus Svensén, and Christopher K. I. Williams. GTM: the Generative Topographic Mapping. *Neural Computation*, 10(1):215–234, 1998.

Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popovic. Style-based inverse kinematics. In *ACM Transactions on Graphics (SIGGRAPH 2004)*, 2004.

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