Latent Variable Modelling with Gaussian Processes

Neil Lawrence
Machine Learning & Optimisation Group
School of Computer Science
University of Manchester, U.K.

13th September 2007

Outline

Online Resources

All source code and slides are available online

- This talk available from my home page (see talks link on left hand side).
- MATLAB examples in the 'oxford' toolbox (vrs 0.131), `demGplvmTalk`.
    - `http://www.cs.man.ac.uk/~neill/oxford/`.
- And the 'fgplvm' toolbox (vrs 0.15).
    - `http://www.cs.man.ac.uk/~neill/fgplvm/`.
- MATLAB commands used for examples given in `typewriter font`.

Bayes' Rule

Posterior Distribution over Variables

$$p(\mathbf{X}|\mathbf{Y},\mathbf{W}) = \frac{p(\mathbf{Y}|\mathbf{X},\mathbf{W})\, p(\mathbf{X})}{p(\mathbf{Y}|\mathbf{W})}$$

Y— data

X— latent variables

W— parameters

e.g. for EEG signals **X** is true source signal, **Y** is observed signals and **W** is a mixing matrix and noise.

Notation

q— dimension of latent/embedded space
d— dimension of data space
n— number of data points

centred data, $\mathbf{Y} = [\mathbf{y}_{1,:}, \ldots, \mathbf{y}_{n,:}]^{\mathrm{T}} = [\mathbf{y}_{:,1}, \ldots, \mathbf{y}_{:,d}] \in \Re^{n\times d}$
latent variables, $\mathbf{X} = [\mathbf{x}_{1,:}, \ldots, \mathbf{x}_{n,:}]^{\mathrm{T}} = [\mathbf{x}_{:,1}, \ldots, \mathbf{x}_{:,q}] \in \Re^{n\times q}$
mapping matrix, $\mathbf{W} \in \Re^{d\times q}$

$\mathbf{a}_{i,:}$ is a vector from the ith row of a given matrix $\mathbf{A}$
$\mathbf{a}_{:,j}$ is a vector from the jth row of a given matrix $\mathbf{A}$

Reading Notation

$\mathbf{X}$ and $\mathbf{Y}$ are *design matrices*

- Covariance given by $n^{-1}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}$.
- Inner product matrix given by $\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$.

Linear Dimensionality Reduction

Linear Latent Variable Model

- Represent data, $\mathbf{Y}$, with a lower dimensional set of latent variables $\mathbf{X}$.
- Assume a linear relationship of the form

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\eta}_{i,:},$$

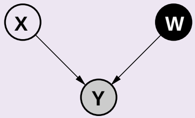
where

$$\eta_{i,:} \sim N\left(\mathbf{0}, \sigma^2\mathbf{I}\right).$$

Linear Latent Variable Model

Probabilistic PCA [Tipping and Bishop, 1999, Roweis, 1998]

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

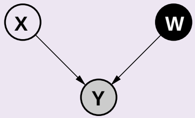


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA [Tipping and Bishop, 1999, Roweis, 1998]

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

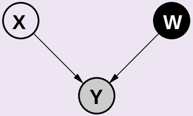


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right) = \prod_{i=1}^{n} N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA [Tipping and Bishop, 1999, Roweis, 1998]

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



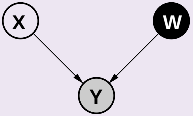
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

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

Linear Latent Variable Model

Probabilistic PCA [Tipping and Bishop, 1999, Roweis, 1998]

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



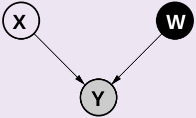
$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

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

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

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln [Tipping and Bishop, 1999]



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

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln [Tipping and Bishop, 1999]

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

$$\log p(\mathbf{Y}|\mathbf{W}) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $n^{-1}\mathbf{Y}^{\top}\mathbf{Y}$ and the corresponding eigenvalues are Λ_q,

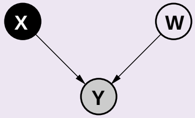
$$\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{V}^{\top}, \quad \mathbf{L} = \left(\Lambda_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{V}$ is an arbitrary rotation matrix.

Linear Latent Variable Model III

Dual Probabilistic PCA

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

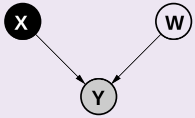


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

Linear Latent Variable Model III

Dual Probabilistic PCA

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

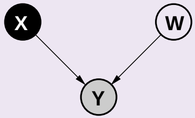


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right) = \prod_{i=1}^{n} N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model III

Dual Probabilistic PCA

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



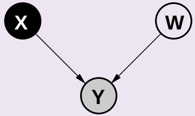
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

$$p(\mathbf{W}) = \prod_{i=1}^{d} N\left(\mathbf{w}_{i,:}|\mathbf{0}, \mathbf{I}\right)$$

Linear Latent Variable Model III

Dual Probabilistic PCA

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



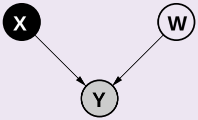
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

$$p(\mathbf{W}) = \prod_{i=1}^{d} N\left(\mathbf{w}_{i,:}|\mathbf{0},\mathbf{I}\right)$$

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

Linear Latent Variable Model IV

Dual Probabilistic PCA Max. Likelihood Soln [Lawrence, 2004, 2005]



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

Linear Latent Variable Model IV

Dual Probabilistic PCA Max. Likelihood Soln [Lawrence, 2004, 2005]

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

$$\log p(\mathbf{Y}|\mathbf{X}) = -\frac{d}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\top}\right) + \text{const.}$$

If $\mathbf{U}'_q$ are first q principal eigenvectors of $d^{-1}\mathbf{Y}\mathbf{Y}^{\top}$ and the corresponding eigenvalues are Λ_q,

$$\mathbf{X} = \mathbf{U'}_q\mathbf{L}\mathbf{V}^{\top}, \quad \mathbf{L} = \left(\Lambda_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{V}$ is an arbitrary rotation matrix.

Linear Latent Variable Model IV

Probabilistic PCA Max. Likelihood Soln [Tipping and Bishop, 1999]

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

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log\left|\mathbf{C}\right| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\top}\mathbf{Y}\right) + \text{const.}$$

If $\mathbf{U}_q$ are first q principal eigenvectors of $n^{-1}\mathbf{Y}^{\top}\mathbf{Y}$ and the corresponding eigenvalues are Λ_q,

$$\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{V}^{\top}, \quad \mathbf{L} = \left(\Lambda_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{V}$ is an arbitrary rotation matrix.

Equivalence of Formulations

The Eigenvalue Problems are equivalent

- Solution for Probabilistic PCA (solves for the mapping)

$$\mathbf{Y}^{\top}\mathbf{Y}\mathbf{U}_q = \mathbf{U}_q\Lambda_q \qquad \mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{V}^{\top}$$

- Solution for Dual Probabilistic PCA (solves for the latent positions)

$$\mathbf{Y}\mathbf{Y}^{\top}\mathbf{U}'_q = \mathbf{U}'_q\Lambda_q \qquad \mathbf{X} = \mathbf{U}'_q\mathbf{L}\mathbf{V}^{\top}$$

- Equivalence is from

$$\mathbf{U}_q = \mathbf{Y}^{\top}\mathbf{U}'_q\Lambda_q^{-\frac{1}{2}}$$

Gaussian Process (GP)

Prior for Functions

- Probability Distribution over Functions
  - Functions are infinite dimensional.
  - Prior distribution over *instantiations* of the function: finite dimensional objects.
- Can prove by induction that GP is 'consistent'.
- Mean and Covariance Functions
  - Instead of mean and covariance matrix, GP is defined by mean function and covariance function.
  - Mean function often taken to be zero or constant.
  - Covariance function must be *positive definite*.
  - Class of valid covariance functions is the same as the class of *Mercer kernels*.

Gaussian Processes II

Zero mean Gaussian Process

- A (zero mean) Gaussian process likelihood is of the form

$$p(\mathbf{y}|\mathbf{X}) = N(\mathbf{y}|\mathbf{0}, \mathbf{K}),$$

where $\mathbf{K}$ is the covariance function or *kernel*.

- The *linear kernel* with noise has the form

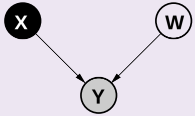
$$\mathbf{K} = \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}$$

- Priors over non-linear functions are also possible.
    - To see what functions look like, we can sample from the prior process.

Non-Linear Latent Variable Model

Dual Probabilistic PCA

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



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

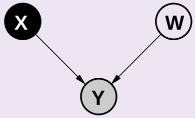
$$p\left(\mathbf{W}\right) = \prod_{i=1}^{d} N\left(\mathbf{w}_{i,:}|\mathbf{0},\mathbf{I}\right)$$

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

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.
  - We recognise it as the 'linear kernel'.
  - We call this the Gaussian Process Latent Variable model (GP-LVM).

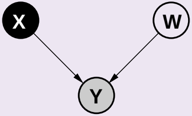


$$p\left(\mathbf{Y}|\mathbf{X}\right) = \prod_{j=1}^{d} N\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}\right)$$

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.
  - We recognise it as the 'linear kernel'.
  - We call this the Gaussian Process Latent Variable model (GP-LVM).



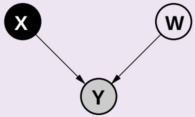
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{d} N\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$$\mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.
  - We recognise it as the 'linear kernel'.
  - We call this the Gaussian Process Latent Variable model (GP-LVM).



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

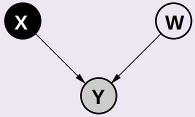
$$\mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

This is a product of Gaussian processes with linear kernels.

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.
  - We recognise it as the 'linear kernel'.
  - We call this the Gaussian Process Latent Variable model (GP-LVM).



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

$$\mathbf{K} = ?$$

Replace linear kernel with non-linear kernel for non-linear model.

Non-linear Latent Variable Models

RBF Kernel

- The RBF kernel has the form $k_{ij} = k\left(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}\right)$, where

$$k\left(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}\right) = \alpha \exp\left(-\frac{\left(\mathbf{x}_{i,:} - \mathbf{x}_{j,:}\right)^{\top}\left(\mathbf{x}_{i,:} - \mathbf{x}_{j,:}\right)}{2l^2}\right).$$

- No longer possible to optimise wrt $\mathbf{X}$ via an eigenvalue problem.
- Instead find gradients with respect to $\mathbf{X}, \alpha, l$ and σ^2 and optimise using conjugate gradients.

Applications

Style Based Inverse Kinematics

Facilitating animation through modelling human motion with the GP-LVM [Grochow et al., 2004]

Tracking

Tracking using models of human motion learnt with the GP-LVM [Urtasun et al., 2005, 2006]

.

Stick Man

Generalization with less Data than Dimensions

- Powerful uncertainly handling of GPs leads to suprising properties.
- Non-linear models can be used where there are fewer data points than dimensions *without overfitting*.
- Example: Modelling a stick man in 102 dimensions with 55 data points!

Stick Man II

demStick1

Figure: The latent space for the stick man motion capture data.

Stick Man II

demStick1

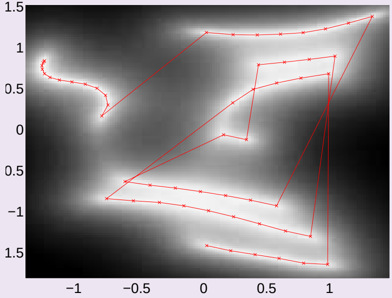


Figure: The latent space for the stick man motion capture data.

Back Constraints I

Local Distance Preservation [Lawrence and Quiñonero Candela, 2006]

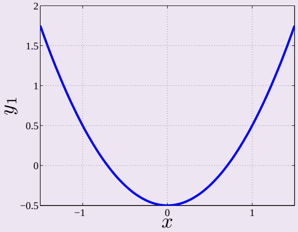
- Most dimensional reduction techniques preserve local distances.
- The GP-LVM does not.
- GP-LVM maps smoothly from latent to data space.
  - Points close in latent space are close in data space.
  - This does not imply points close in data space are close in latent space.
- Kernel PCA maps smoothly from data to latent space.
  - Points close in data space are close in latent space.
  - This does not imply points close in latent space are close in data space.

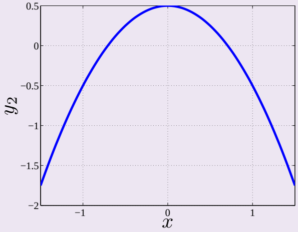
Back Constraints II

Forward Mapping (demBackMapping in oxford toolbox)

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

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



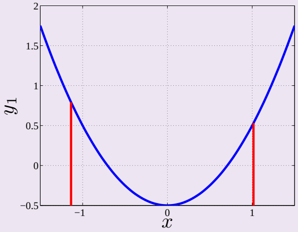


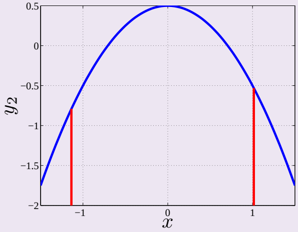
Back Constraints II

Forward Mapping (`demBackMapping` in oxford toolbox)

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

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



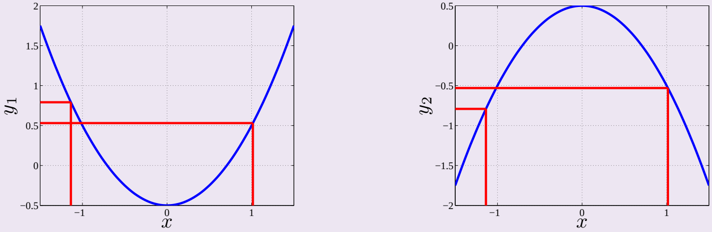


Back Constraints II

Forward Mapping (demBackMapping in oxford toolbox)

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

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

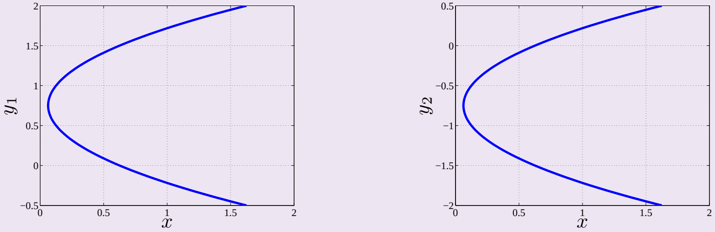


Back Constraints II

Backward Mapping (demBackMapping in oxford toolbox)

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

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

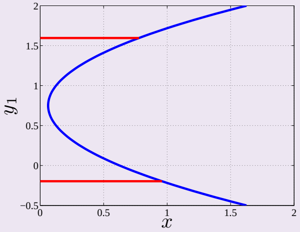


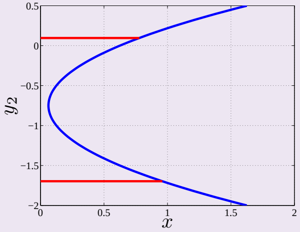
Back Constraints II

Backward Mapping (demBackMapping in oxford toolbox)

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

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



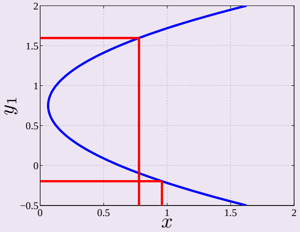


Back Constraints II

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.

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

- 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.[Lawrence and Quiñonero Candela, 2006]
  - 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

$$x_{ij} = f_j(\mathbf{y}_{i,:}; \mathbf{B})$$

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

Motion Capture Results

demStick1 and demStick3

Figure: The latent space for the motion capture data with (*right*) and without (*left*) dynamics. The dynamics us a Gaussian process with an RBF kernel.

.

MANCHESTER 1824

The University of Manchester

Motion Capture Results

demStick1 and demStick3

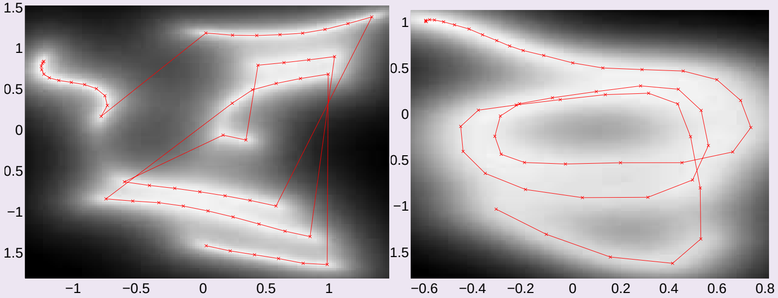
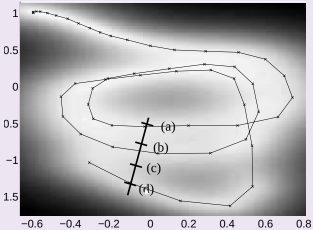


Figure: The latent space for the motion capture data with (*right*) and without (*left*) dynamics. The dynamics us a Gaussian process with an RBF kernel.

Stick Man Results

demStickResults



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

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

Adding Dynamics

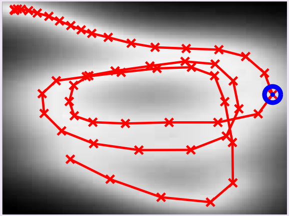
MAP Solutions for Dynamics Models

- Data often has a temporal ordering.
- Markov-based dynamics are often used.
- For the GP-LVM
  - Marginalising such dynamics is intractable.
  - But: MAP solutions are trivial to implement.
- Many choices: Kalman filter, Markov chains *etc.*.
- Wang et al. [2006] suggest using a Gaussian Process.

Gaussian Process Dynamics

GP-LVM with Dynamics

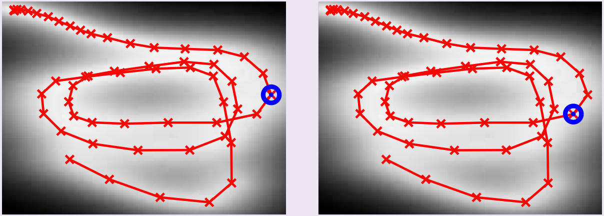
- Autoregressive Gaussian process mapping in latent space between time points.



Gaussian Process Dynamics

GP-LVM with Dynamics

- Autoregressive Gaussian process mapping in latent space between time points.

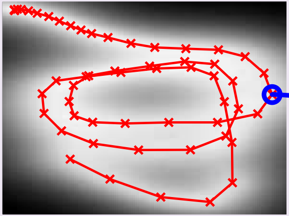


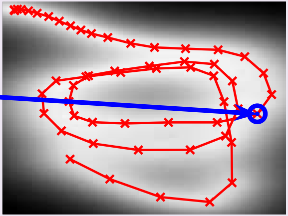
t $t+1$

Gaussian Process Dynamics

GP-LVM with Dynamics

- Autoregressive Gaussian process mapping in latent space between time points.





t $\quad$ $t+1$

Motion Capture Results

demStick1 and demStick2

Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*right*) based on an RBF kernel.

Motion Capture Results

demStick1 and demStick2

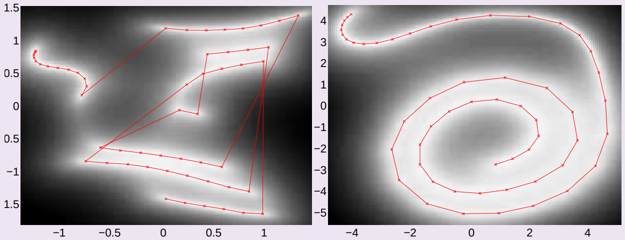
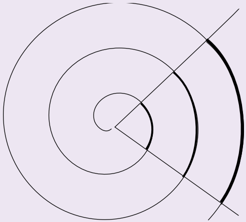


Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*right*) based on an RBF kernel.

Regressive Dynamics

Inner Groove Distortion

- Autoregressive unimodal dynamics, $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)$.
- Forces spiral visualisation.
- Poorer model due to inner groove distortion.



Regressive Dynamics

Direct use of Time Variable

- Instead of auto-regressive dynamics, consider regressive dynamics.
- Take $\mathbf{t}$ as an input, use a prior $p(\mathbf{X}|\mathbf{t})$.
- User a Gaussian process prior for $p(\mathbf{X}|\mathbf{t})$.
- Also allows us to consider variable sample rate data.

Motion Capture Results

demStick1, demStick2 and demStick5

Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*middle*) and with regressive dynamics (*right*) based on an RBF kernel.

Motion Capture Results

demStick1, demStick2 and demStick5

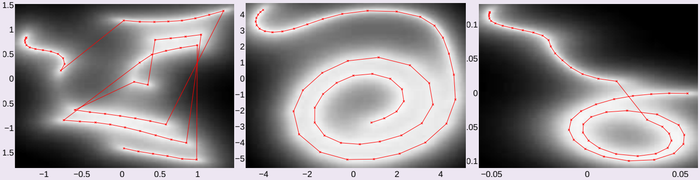


Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*middle*) and with regressive dynamics (*right*) based on an RBF kernel.

Hierarchical GP-LVM

Stacking Gaussian Processes

- Regressive dynamics provides a simple hierarchy.
  - The input space of the GP is governed by another GP.
- By stacking GPs we can consider more complex hierarchies.
- Ideally we should marginalise latent spaces
  - In practice we seek MAP solutions.

Two Correlated Subjects

demHighFive1

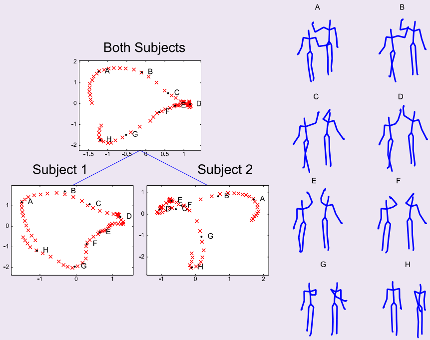


Figure: Hierarchical model of a 'high five'.

Within Subject Hierarchy

Decomposition of Body

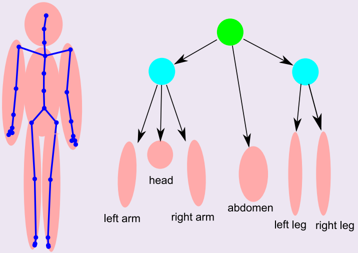


Figure: Decomposition of a subject.

Single Subject Run/Walk

demRunWalk1

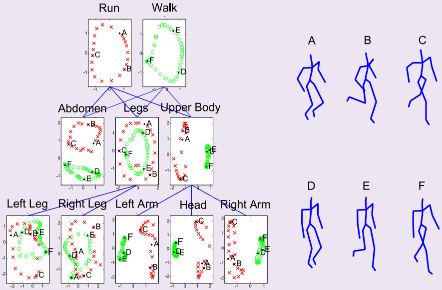


Figure: Hierarchical model of a walk and a run.

Summary

- We maximised latent variables integrated out parameters.
- This allowed us to:
  - Optimise latent variables by constrained maximum likelihood.
  - Apply complex dynamics models to the latent space and seek MAP solutions.
  - Build hierarchies of the MAP models for decomposition of data structure.

MANCHESTER 1824

The University of Manchester

References

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.

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, Nov 2005.

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.

D. Lowe and M. E. Tipping. Neuroscale: Novel topographic feature extraction with radial basis function networks. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 543–549, Cambridge, MA, 1997. MIT Press.

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.

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

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.

R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 238–245, New York, U.S.A., 17–22 Jun. 2006. IEEE Computer Society Press.

J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.