Human Motion Modelling with Gaussian Processes

Neil D. Lawrence

7th February 2008

Outline

Outline

The Model

A Probabilistic Non-linear PCA

- PCA has a probabilistic interpretation [Tipping and Bishop, 1999].
- It is difficult to 'non-linearise'.

Dual Probabilistic PCA

- We present a new probabilistic interpretation of PCA [Lawrence, 2005].
- This interpretation can be made non-linear.
- The result is non-linear probabilistic PCA.

Notation

q— dimension of latent/embedded space

d— dimension of data space

n— number of data points

centred data, $\mathbf{Y} = [\mathbf{y}_{1,:}, \dots, \mathbf{y}_{n,:}]^{\mathrm{T}} = [\mathbf{y}_{:,1}, \dots, \mathbf{y}_{:,d}] \in \Re^{n \times d}$

latent variables, $\mathbf{X} = [\mathbf{x}_{1,:}, \dots, \mathbf{x}_{n,:}]^{\mathrm{T}} = [\mathbf{x}_{:,1}, \dots, \mathbf{x}_{:,q}] \in \Re^{n \times q}$

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 $\mathbf{A}$

X and 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

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

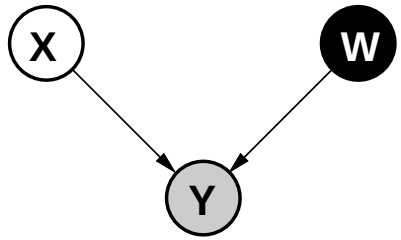
where

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

Linear Latent Variable Model

Probabilistic PCA

- 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*.

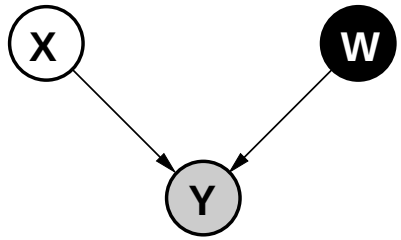


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- 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*.

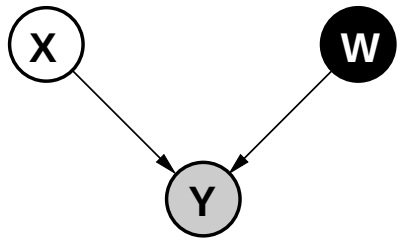


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

- 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*.



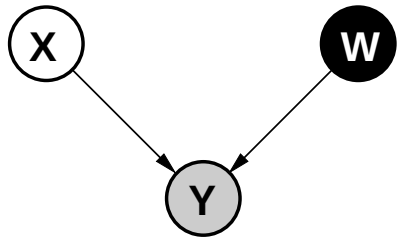
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

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

Linear Latent Variable Model

Probabilistic PCA

- 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*.

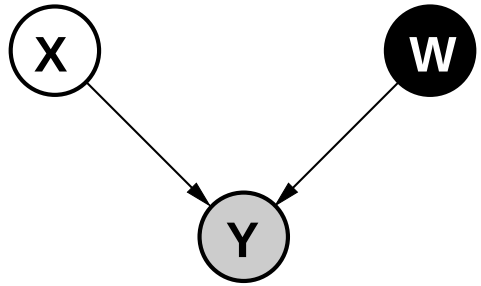


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

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

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

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



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

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

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

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

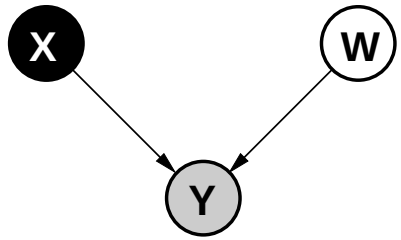
$$\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{V}^{\mathsf{T}}, \quad \mathbf{L} = \left(\Lambda_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

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*.

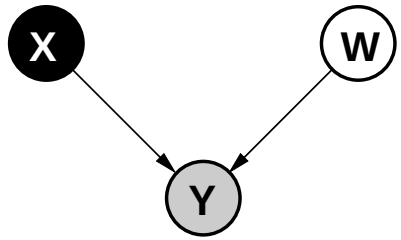


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

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*.

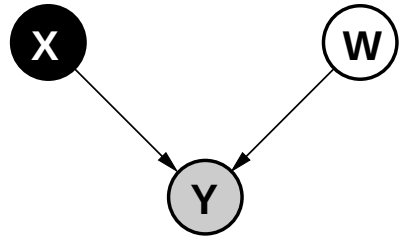


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

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*.



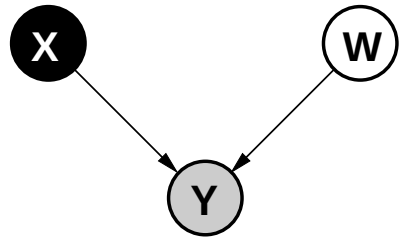
$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

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

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*.

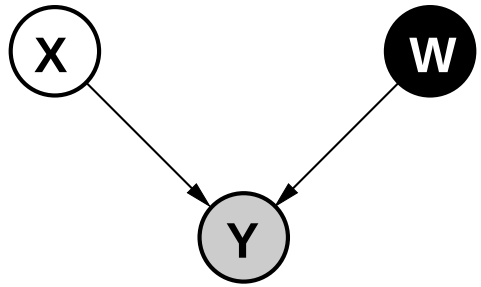


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} N\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

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

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

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



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

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

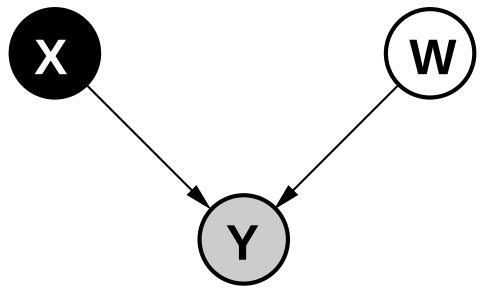
$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = -\frac{n}{2}\log|\mathbf{C}| - \frac{1}{2}\mathrm{tr}\left(\mathbf{C}^{-1}\mathbf{Y}^{\mathsf{T}}\mathbf{Y}\right) + \text{const.}$$

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

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

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

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



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

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

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

If $\mathbf{U}'_q$ are first q principal eigenvectors of $d^{-1}\mathbf{Y}\mathbf{Y}^{\mathsf{T}}$ and the corresponding eigenvalues are Λ_q,

$$\mathbf{X} = \mathbf{U}'_q\mathbf{L}\mathbf{V}^{\mathsf{T}}, \quad \mathbf{L} = \left(\Lambda_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

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

Equivalence of Formulations

The Eigenvalue Problems are equivalent

- Solution for Probabilistic PCA (solves for the mapping)

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

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

$$\mathbf{Y}\mathbf{Y}^{\mathsf{T}}\mathbf{U}'_q = \mathbf{U}'_q\Lambda_q \qquad \mathbf{X} = \mathbf{U}'_q\mathbf{L}\mathbf{V}^{\mathsf{T}}$$

- Equivalence is from

$$\mathbf{U}_q = \mathbf{Y}^{\mathsf{T}}\mathbf{U}'_q\Lambda_q^{-\frac{1}{2}}$$

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

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

  where $\mathbf{K}$ is the covariance function or *kernel*.
- The *linear kernel* with noise has the form

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

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

Covariance Samples

`demCovFuncSample`

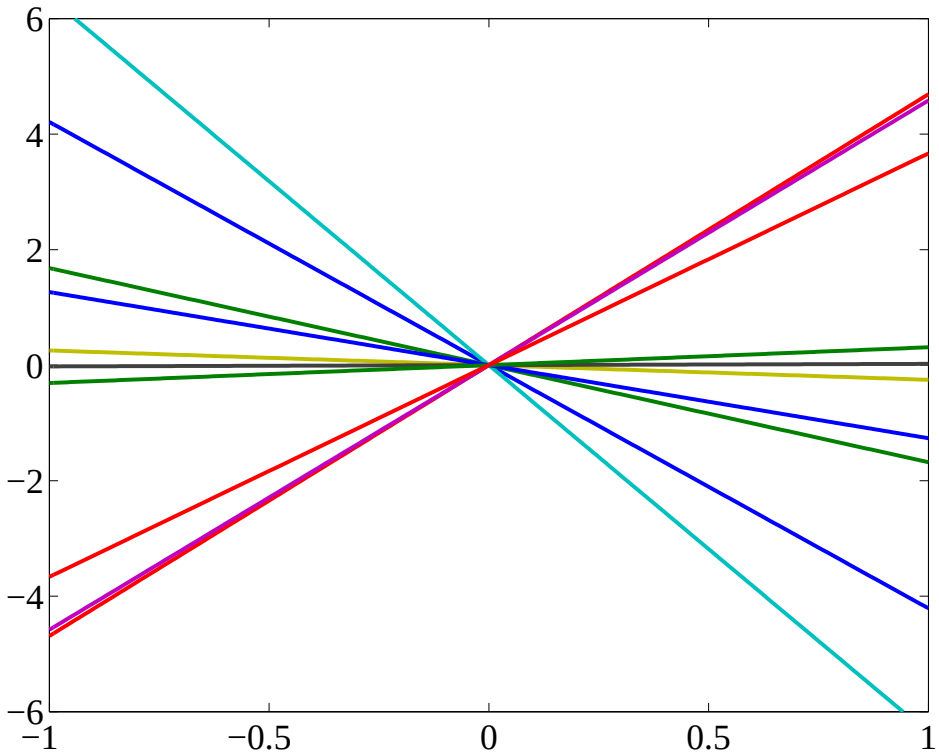


Figure: linear kernel, $\mathbf{K} = \mathbf{X}\mathbf{X}^{\mathsf{T}}$

Covariance Samples

`demCovFuncSample`

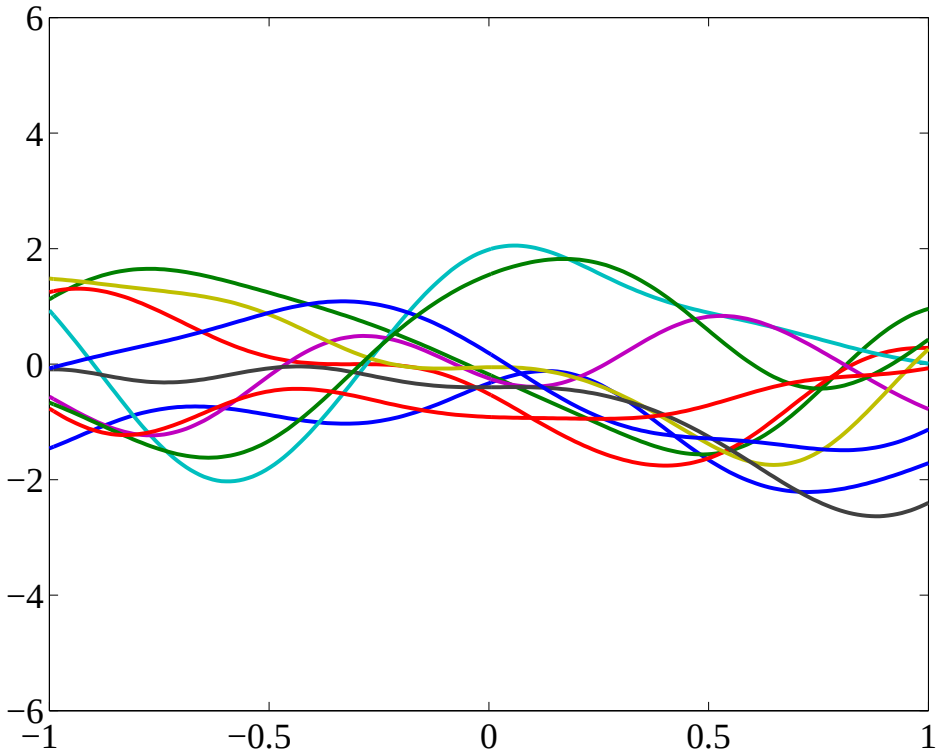


Figure: RBF kernel, $k_{i,j} = \alpha \exp\left(-\frac{1}{2l}\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)$, with $l = 0.32$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

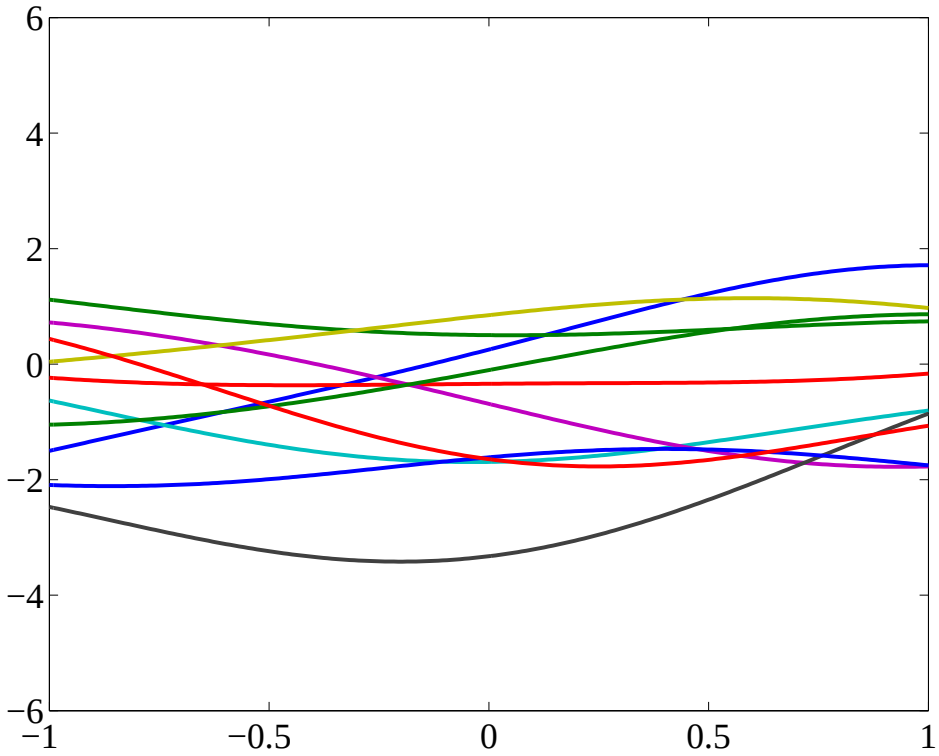


Figure: RBF kernel, $k_{i,j} = \alpha \exp\left(-\frac{1}{2l}\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)$, with $l = 1$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

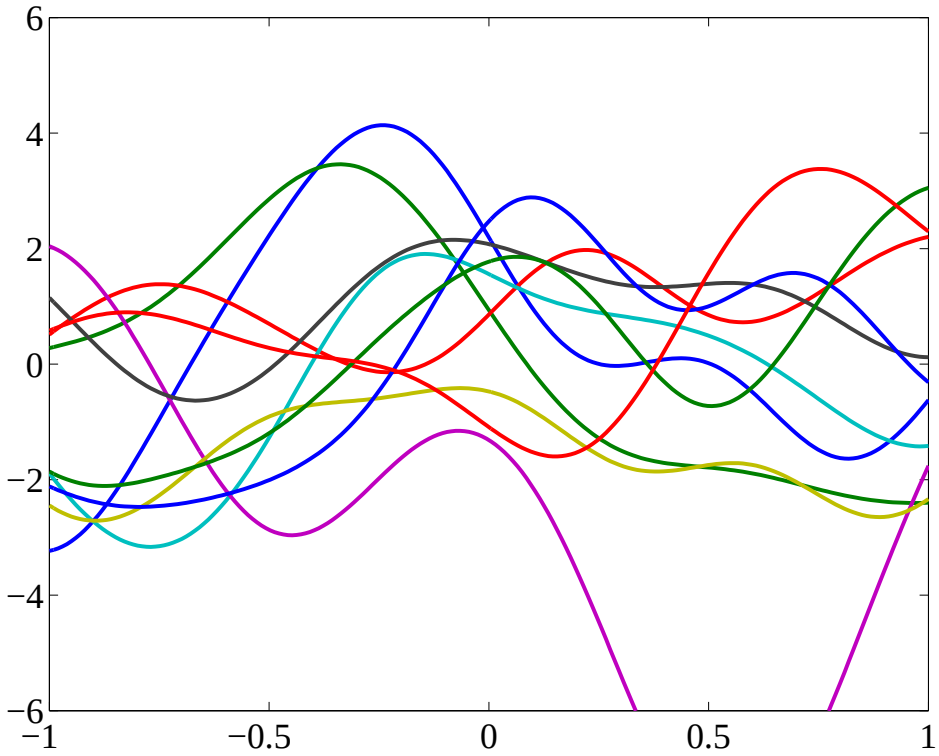


Figure: RBF kernel, $k_{i,j} = \alpha \exp\left(-\frac{1}{2l}\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)$, with $l = 0.3$, $\alpha = 4$

Covariance Samples

`demCovFuncSample`

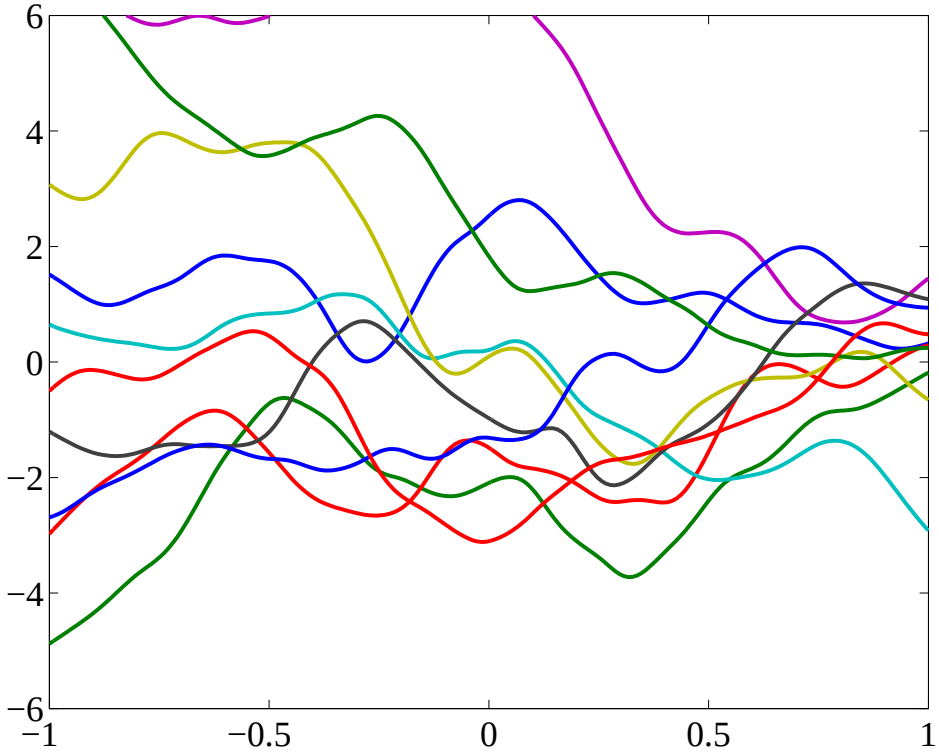


Figure: MLP kernel, $k_{i,j} = \alpha \sin^{-1}\left(\frac{w\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j + b}{\sqrt{\left(w\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_i + b + 1\right)\left(w\mathbf{x}_j^{\mathrm{T}}\mathbf{x}_j + b + 1\right)}}\right)$, with $\alpha = 8$, $w = 100$ and $b = 100$

Covariance Samples

`demCovFuncSample`

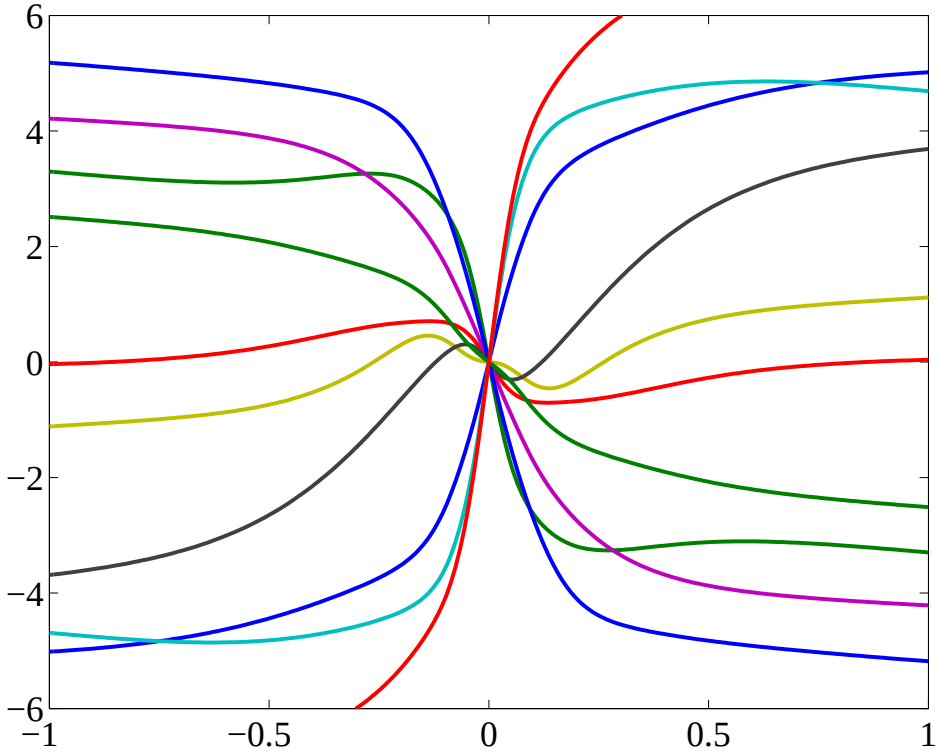


Figure: MLP kernel, $k_{i,j} = \alpha \sin^{-1}\left(\frac{w\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j + b}{\sqrt{\left(w\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_i + b + 1\right)\left(w\mathbf{x}_j^{\mathrm{T}}\mathbf{x}_j + b + 1\right)}}\right)$, with $\alpha = 8$, $b = 0$ and $w = 100$

Covariance Samples

`demCovFuncSample`

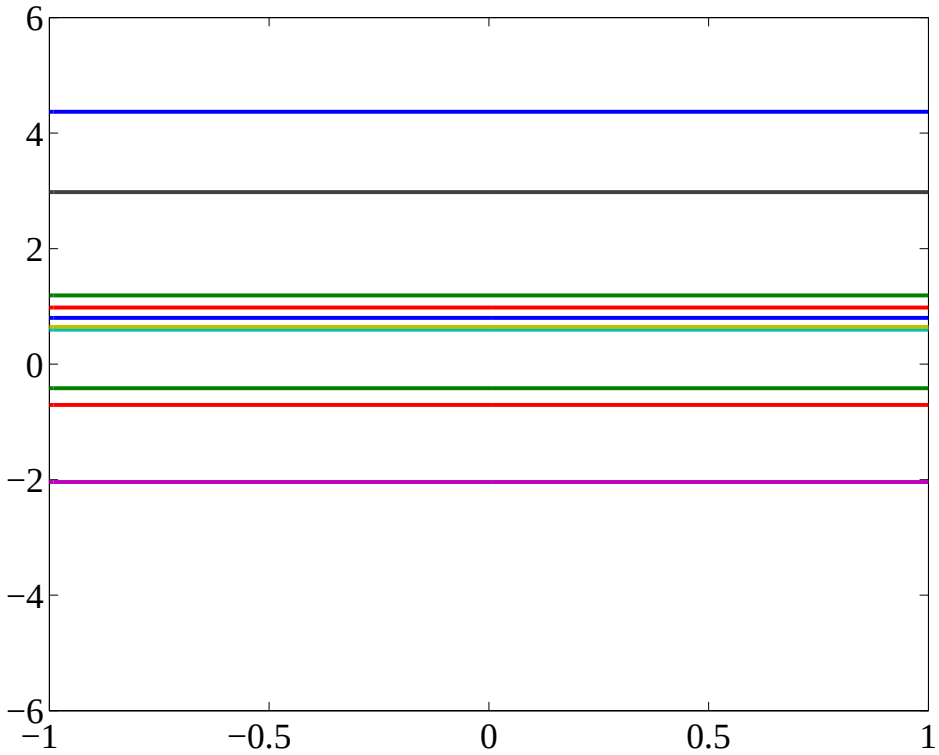


Figure: bias 'kernel', $k_{i,j} = \alpha$, with $\alpha = 1$ and

Covariance Samples

`demCovFuncSample`

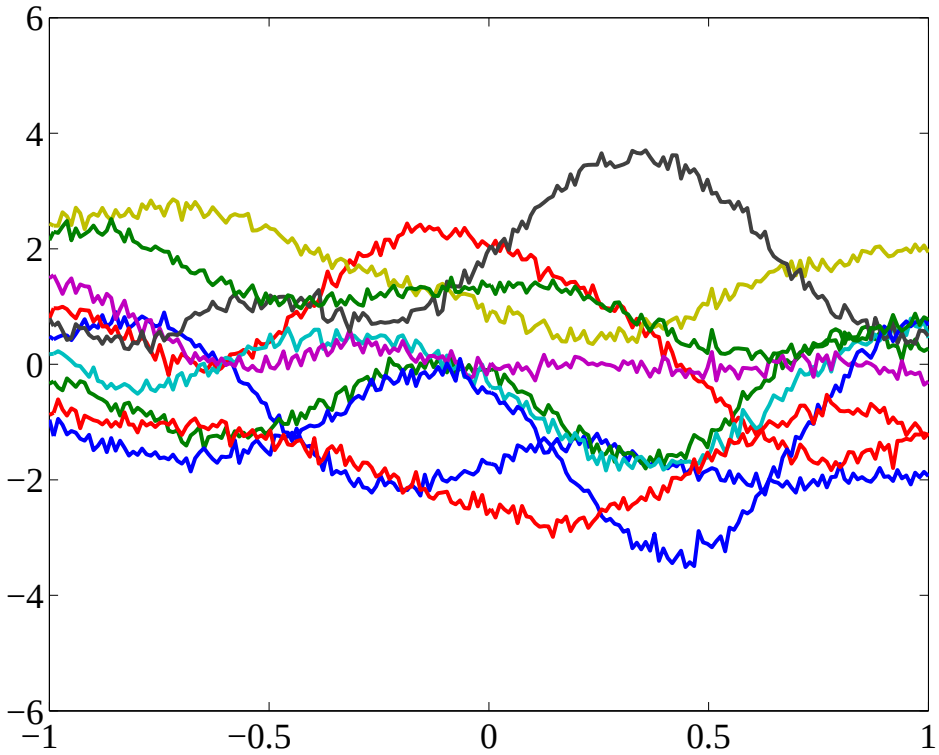


Figure: summed combination of: RBF kernel, $\alpha = 1$, $l = 0.3$; bias kernel, $\alpha = 1$; and white noise kernel, $\beta = 100$

Gaussian Process Regression

Posterior Distribution over Functions

- Gaussian processes are often used for regression.
- We are given a known inputs $\mathbf{X}$ and targets $\mathbf{Y}$.
- We assume a prior distribution over functions by selecting a kernel.
- Combine the prior with data to get a *posterior* distribution over functions.

Gaussian Process Regression

`demRegression`

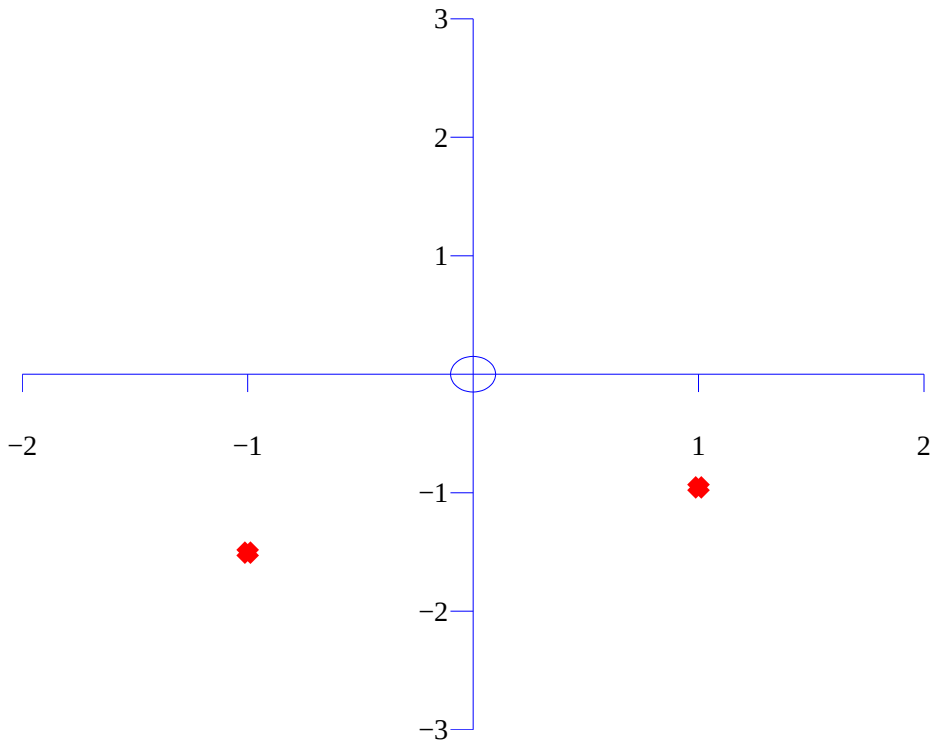


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

`demRegression`

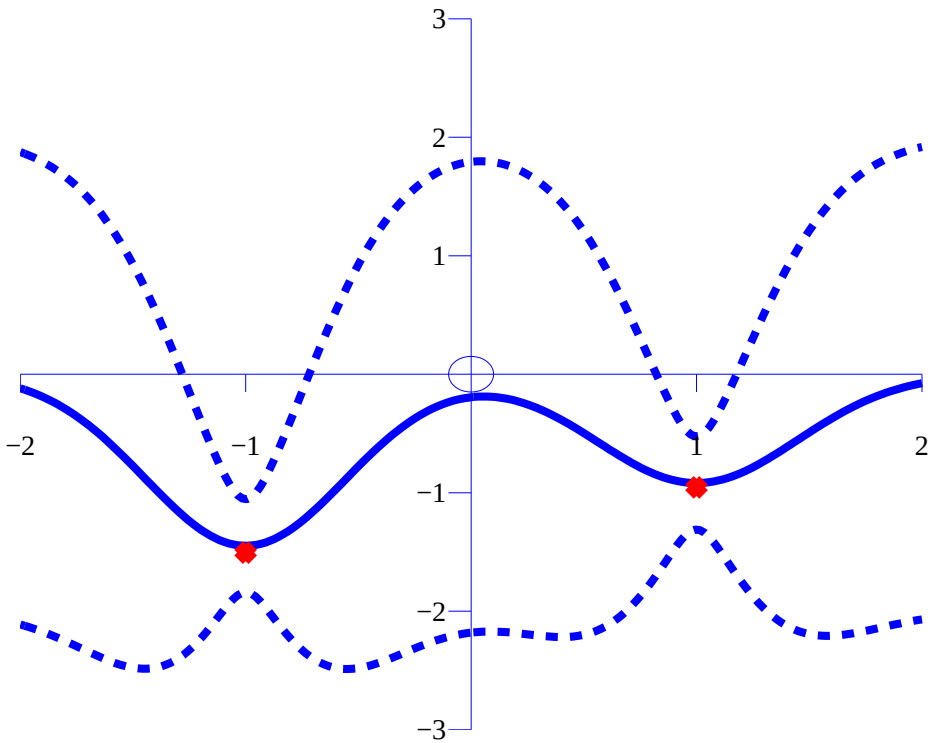


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

`demRegression`

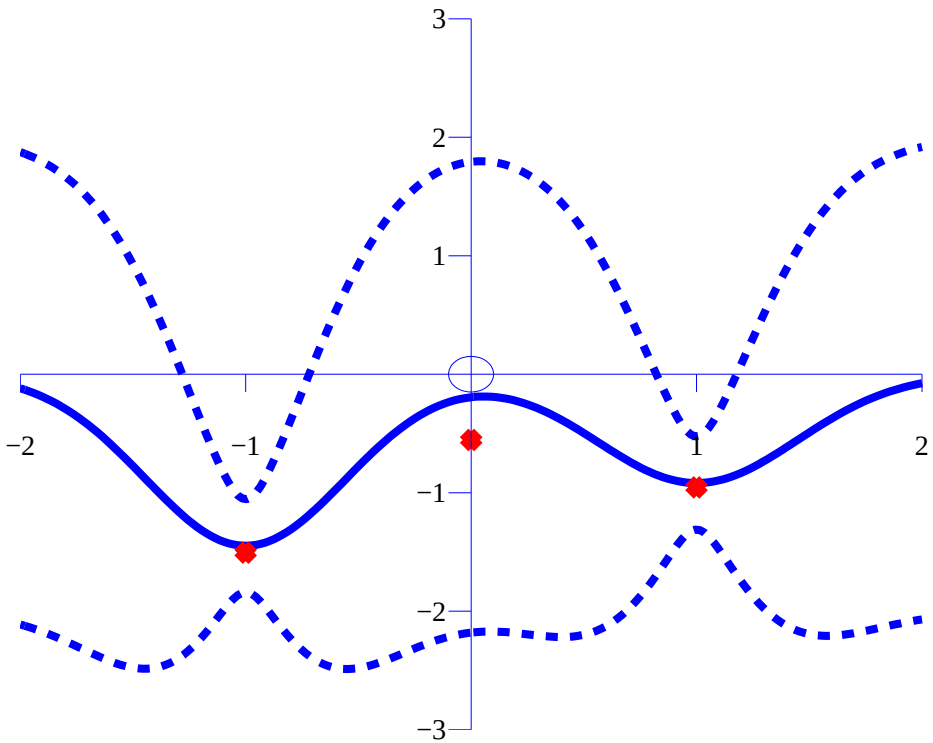


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

`demRegression`

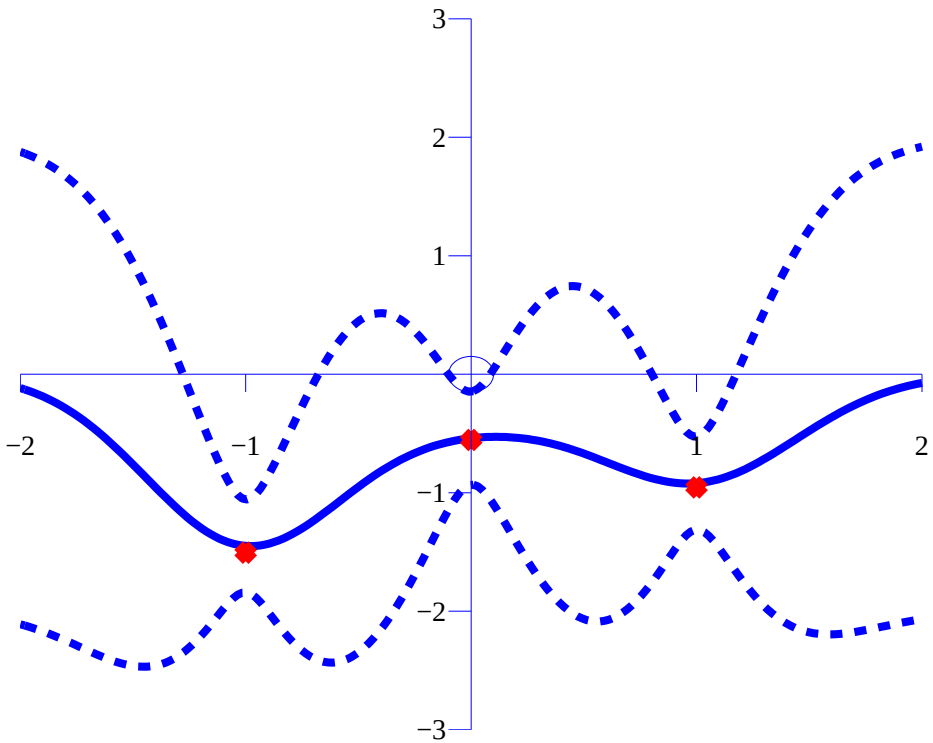


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

`demRegression`

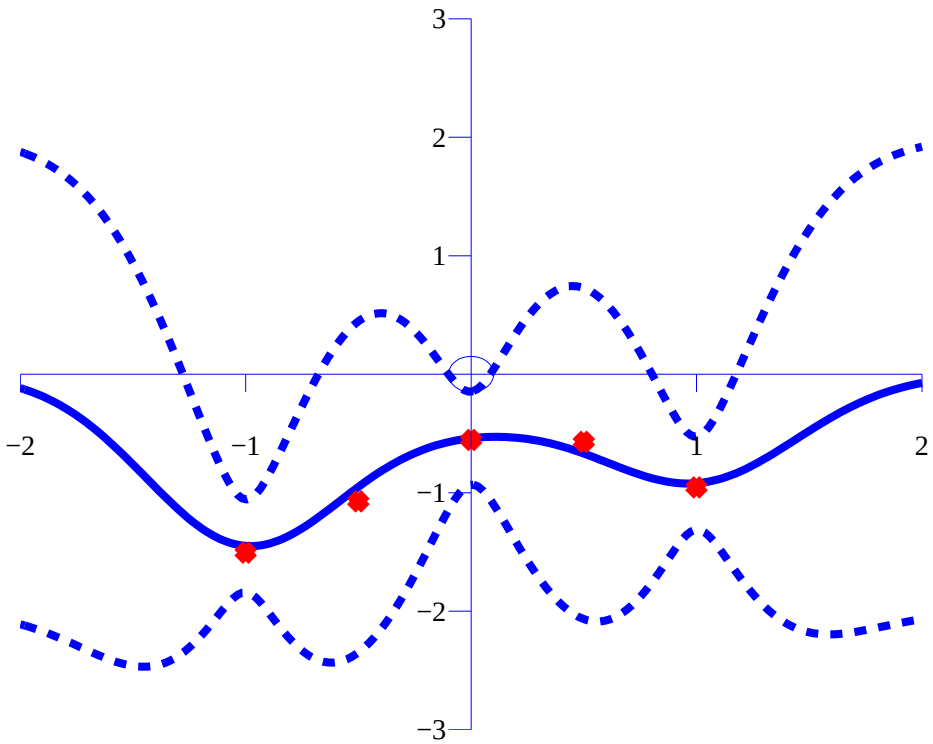


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

`demRegression`

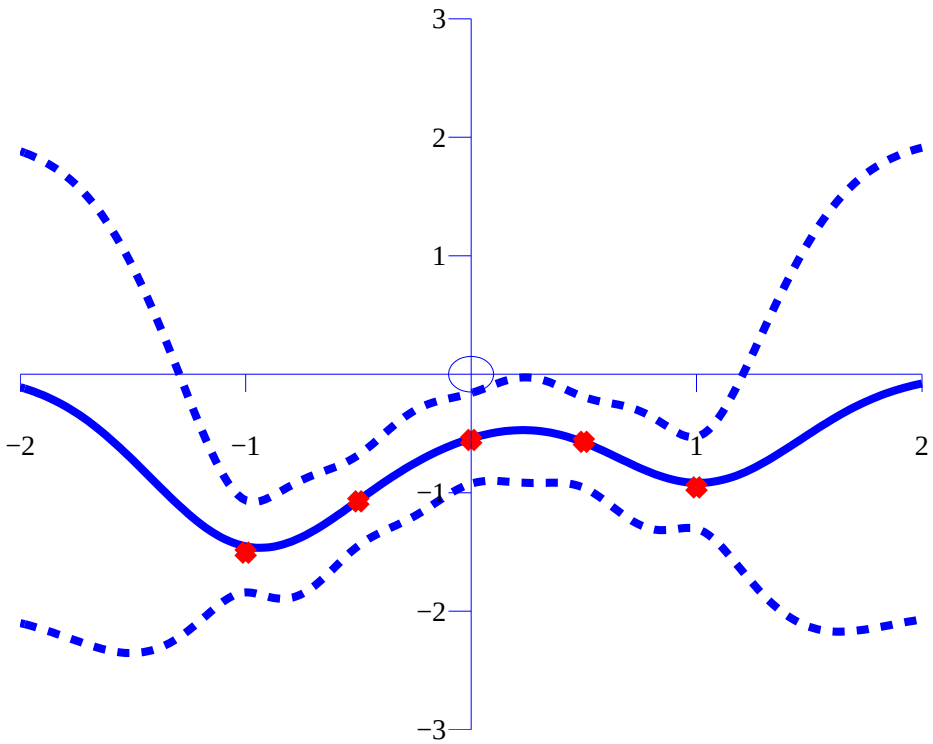


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

`demRegression`

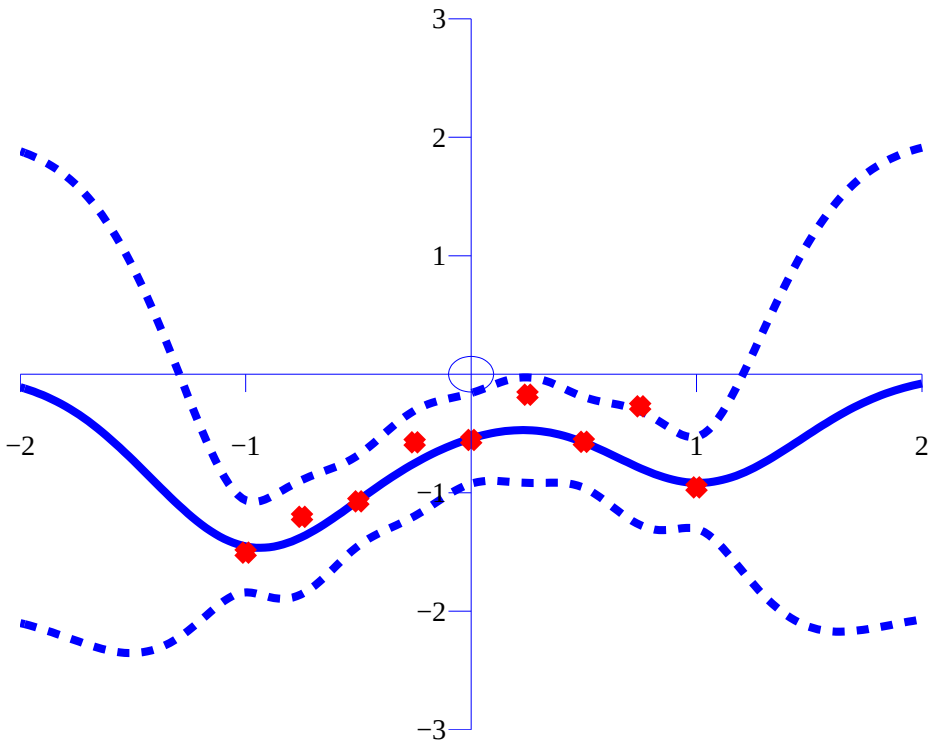


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

`demRegression`

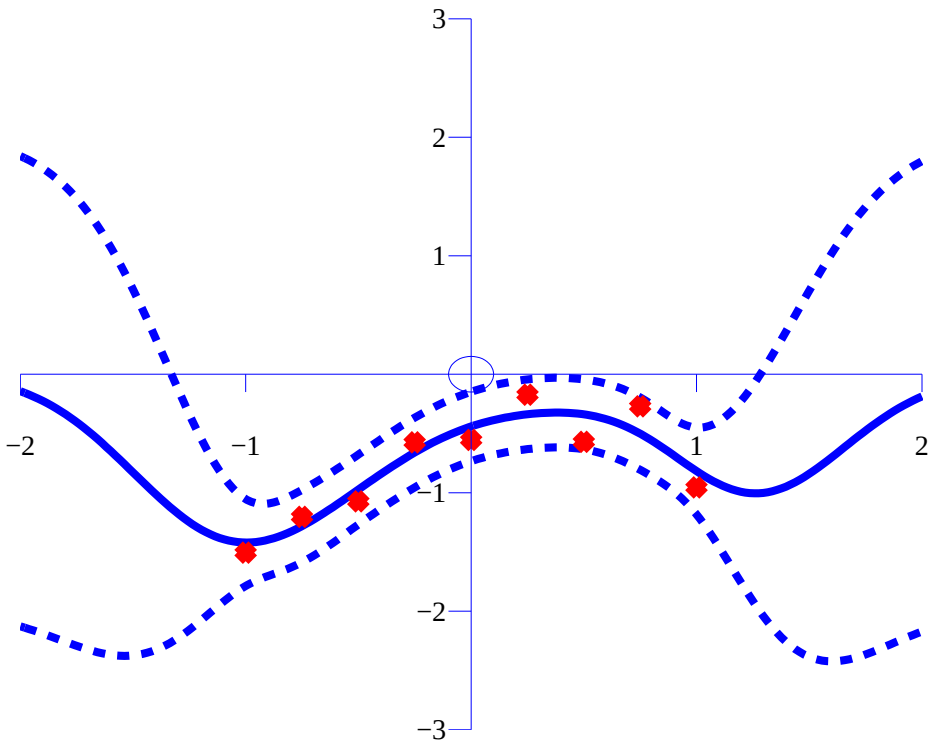
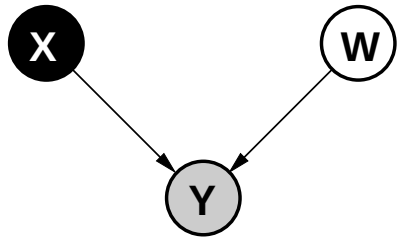


Figure: Examples include WiFi localization, C14 callibration curve.

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*.



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

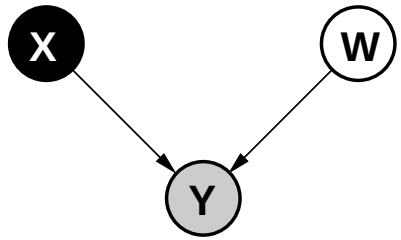
$$p\left(\mathbf{W}\right)=\prod_{i=1}^{d}N\left(\mathbf{w}_{i,:}|\mathbf{0},\mathbf{I}\right)$$

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

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).

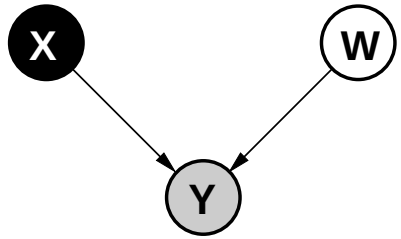


$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{d} N\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^{\mathsf{T}} + \sigma^2\mathbf{I}\right)$$

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).



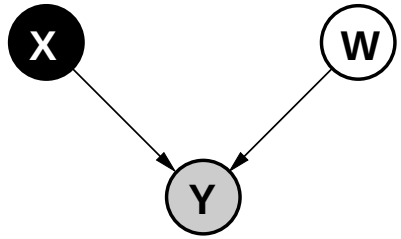
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{d} N(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K})$$

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

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).



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

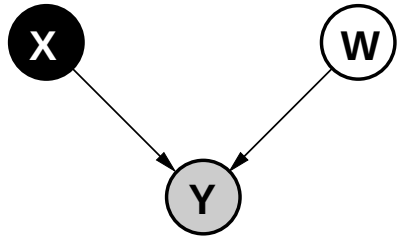
$$\mathbf{K} = \mathbf{X}\mathbf{X}^{\mathsf{T}} + \sigma^2\mathbf{I}$$

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).



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

$$\mathbf{K} = ?$$

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

RBF Kernel

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

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

- 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.

Test Data

Traditional Model

- For a parameteric model maximise the marginal.

$$\log p\left(\mathbf{Y}|\mathbf{W}\right) = \log \int p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right) p\left(\mathbf{X}\right) \mathrm{d}\mathbf{X} = \prod_{i=1}^{N} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{0},\mathbf{W}\mathbf{W}^{\mathrm{T}} + \mathbf{I}\sigma^2\right)$$

  leading to $\hat{\mathbf{W}}$.

- Evaluate the log likelihood of a new, test point $\mathbf{y}_*$,

$$\log p\left(\mathbf{y}_*|\hat{\mathbf{W}}\right) = \log \mathcal{N}\left(\mathbf{y}_*|\mathbf{0},\hat{\mathbf{W}}\hat{\mathbf{W}}^{\mathrm{T}} + \mathbf{I}\sigma^2\right).$$

Test Data

New Model

- For non-parameteric model maximise the marginal

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

  leading to $\hat{\mathbf{X}}$.
- Evaluate the log likelihood of a new, test point $\mathbf{y}_*$??

$$\log p\left(\mathbf{y}_*, \mathbf{Y}|\mathbf{x}_*, \hat{\mathbf{X}}\right) = \log p\left(\mathbf{y}_*|\mathbf{Y}, \mathbf{x}_*, \hat{\mathbf{X}}\right) + \log p\left(\mathbf{Y}|\hat{\mathbf{X}}\right).$$

Maximise New Latent Variable

- Maximise log likelihood with respect to $\mathbf{x}_*$,

$$\log p\left(\mathbf{y}_*|\mathbf{Y}, \mathbf{x}_*, \hat{\mathbf{X}}\right) = \sum_{j=1}^{d} \log \mathcal{N}\left(y_{j*}|\mathbf{k}_*^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y}_{:,j}, k_{*,*} - \mathbf{k}_*^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{k}_*\right)$$

where the new covariance (of the joint process) is partitioned as

$$\mathbf{K}' = \begin{bmatrix} \mathbf{K} & \mathbf{k}_* \\ \mathbf{k}_* & k_{*,*} \end{bmatrix}$$

Multiple Test Data

- What if we are given two data points though? $\mathbf{y}_*$ and $\mathbf{y}_{**}$

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

$$\log p\left(\mathbf{y}_*, \mathbf{y}_{**}|\mathbf{Y}, \mathbf{x}_*, \mathbf{x}_{**}, \hat{\mathbf{X}}\right) = \sum_{j=1}^{d} \log \mathcal{N}\left([y_{j*}\ y_{j**}]^{\mathrm{T}}|\mathbf{K}_*^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y}_{:,j}, \mathbf{K}_{*,*} - \mathbf{K}_*^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{K}_*\right)$$

- But this is a joint covariance!!! It is not the same as maximising the individual data twice:

$$\log p\left(\mathbf{y}_*|\mathbf{Y}, \mathbf{x}_*, \hat{\mathbf{X}}\right) p\left(\mathbf{y}_{**}|\mathbf{Y}, \mathbf{x}_{**}, \hat{\mathbf{X}}\right) = \log \sum_{j=1}^{d} \mathcal{N}\left(y_{j*}|\mathbf{k}_*^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y}_{:,j}, k_{*,*} - \mathbf{k}_*^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{k}_*\right)$$

$$+ \log \sum_{j=1}^{d} \mathcal{N}\left(y_{j**}|\mathbf{k}_{**}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{y}_{:,j}, k_{**,**} - \mathbf{k}_{**}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{k}_{**}\right)$$

- What does it mean?

Possible Explanation?

Comment from John yesterday:

- For Gaussians maximising $\log p\left(x, y|\Sigma\right)$ wrt Σ is equivalent to $\log p\left(x|y, \Sigma\right) + \log p\left(y|x, \Sigma\right)$ wrt Σ.
- So perhaps maximising

$$\log p\left(\mathbf{y}_*, \mathbf{y}_{**}|\mathbf{Y}, \mathbf{x}_*, \mathbf{x}_{**}, \hat{\mathbf{X}}\right) \equiv \log p\left(\mathbf{y}_*, \mathbf{y}_{**}|\mathbf{Y}, \mathbf{K}''\right)$$

  is equivalent to maximising

$$\log p\left(\mathbf{y}_*|\mathbf{y}_{**}, \mathbf{Y}, \mathbf{K}''\right) + \log p\left(\mathbf{y}_{**}|\mathbf{y}_*, \mathbf{Y}, \mathbf{K}''\right)$$

- *cf*

$$\log p\left(\mathbf{y}_*|\mathbf{Y}, \mathbf{K}''\right) + \log p\left(\mathbf{y}_{**}|\mathbf{Y}, \mathbf{K}''\right)$$

Outline

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`

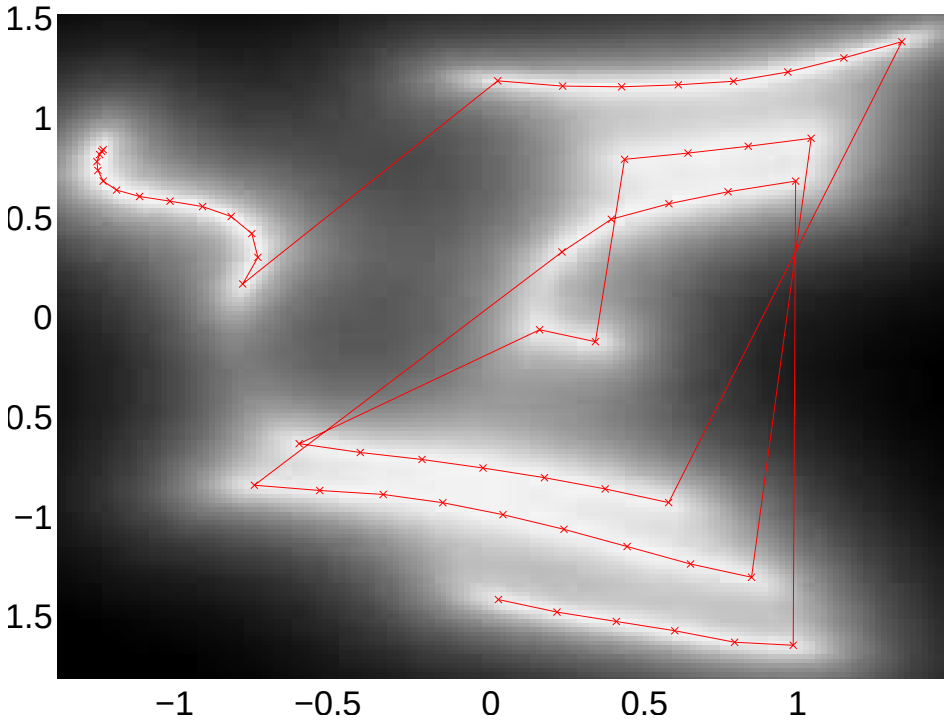


Figure: The latent space for the stick man motion capture data.

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]

.

Outline

Back Constraints I

Local Distance Preservation [Lawrence and Quiñonero Candela, 2006]

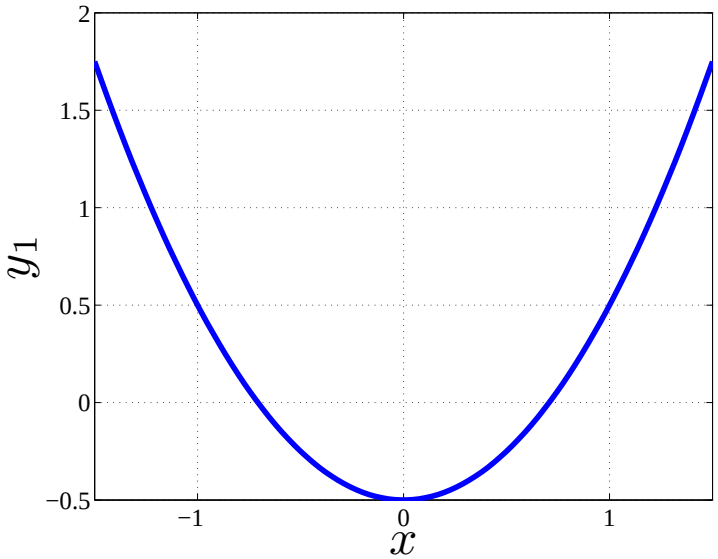
- Most dimensional reduction techniques preserve local distances.
- The GP-LVM does not.
- GP-LVM maps smoothly from latent to data space.
- Points close in latent space are close in data space.
  - This does not imply points close in data space are close in latent space.
- Many methods map smoothly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

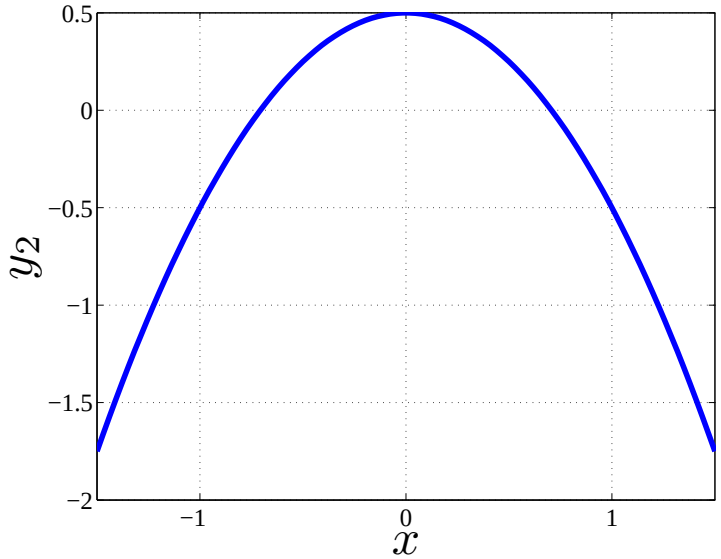
Back Constraints II

Forward Mapping (`demBackMapping` in oxford toolbox)

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

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



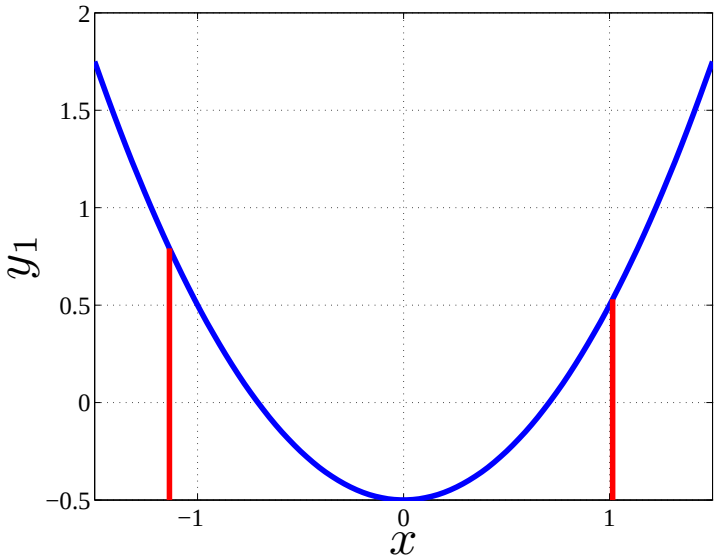


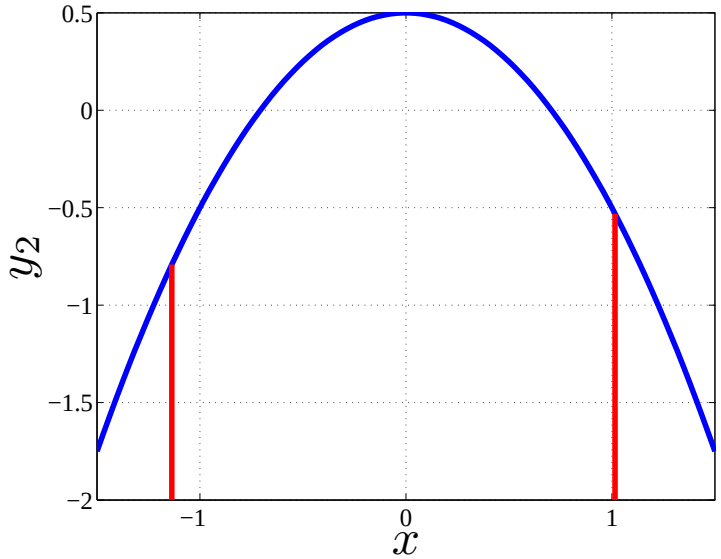
Back Constraints II

Forward Mapping (`demBackMapping` in oxford toolbox)

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

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



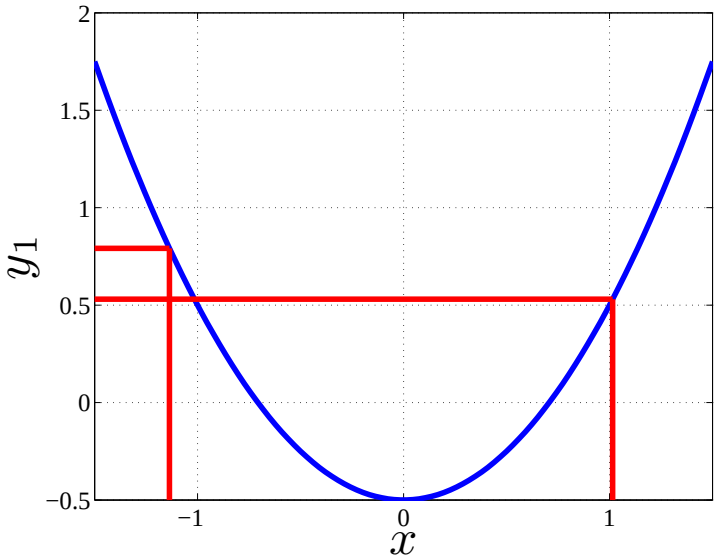


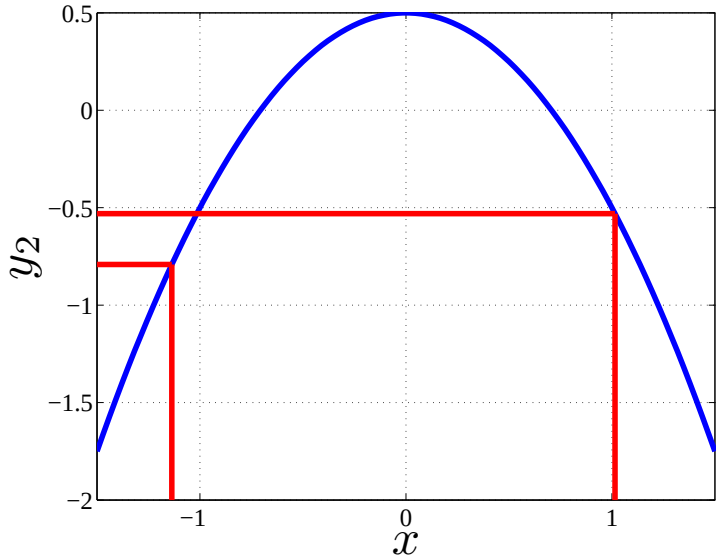
Back Constraints II

Forward Mapping (`demBackMapping` in oxford toolbox)

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

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



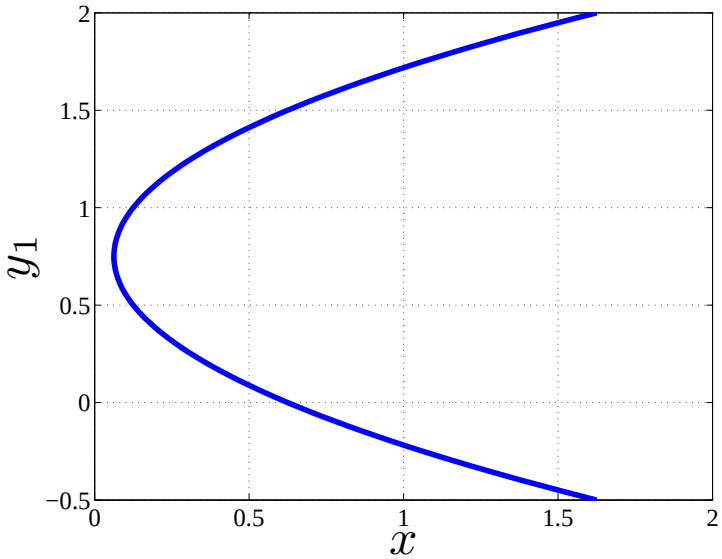


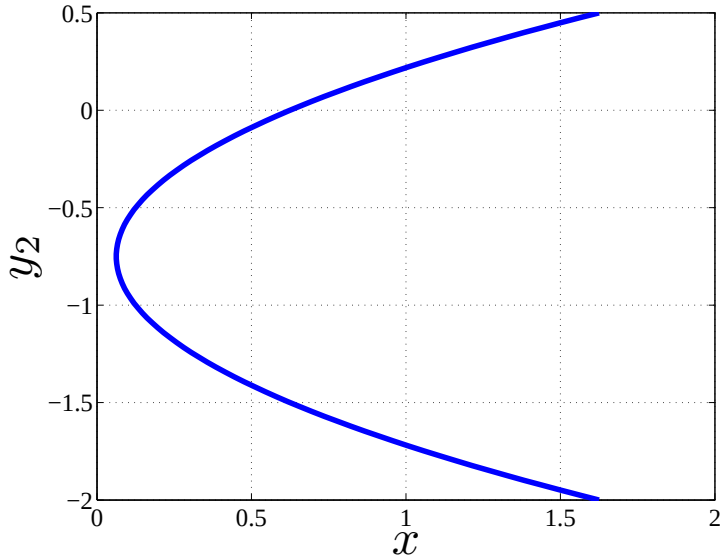
Back Constraints II

Backward Mapping (`demBackMapping` in oxford toolbox)

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

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



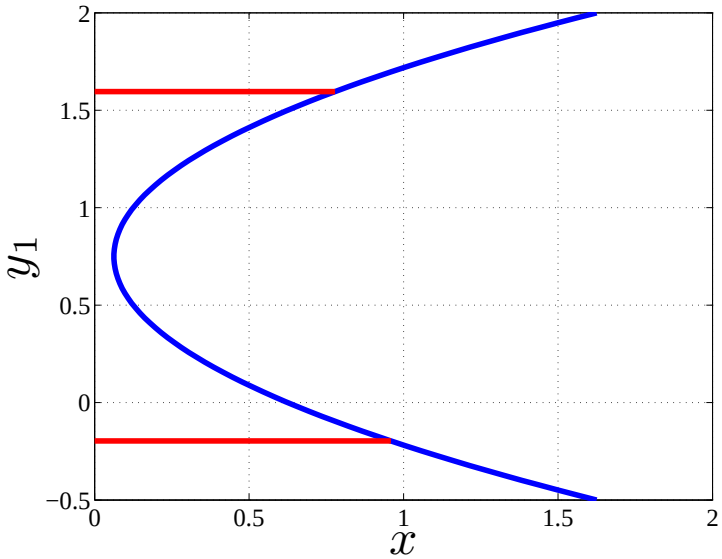


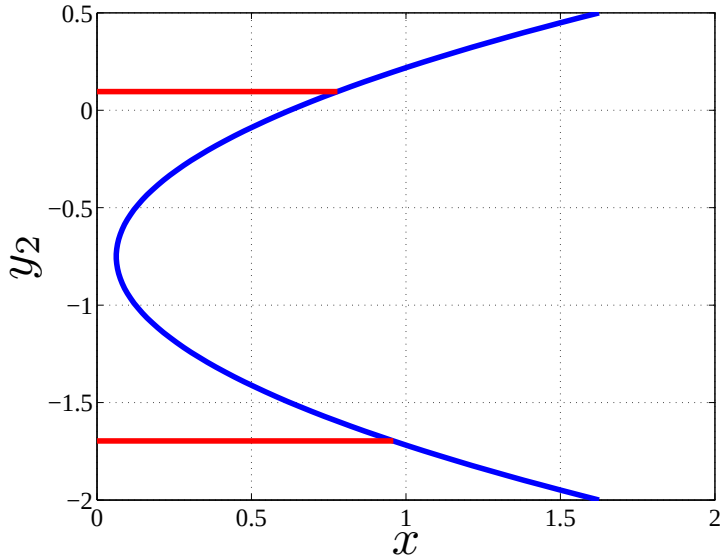
Back Constraints II

Backward Mapping (`demBackMapping` in oxford toolbox)

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

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



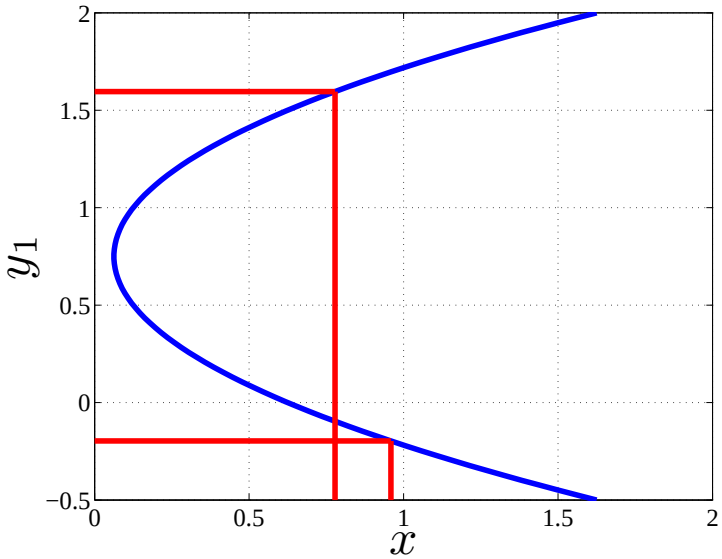


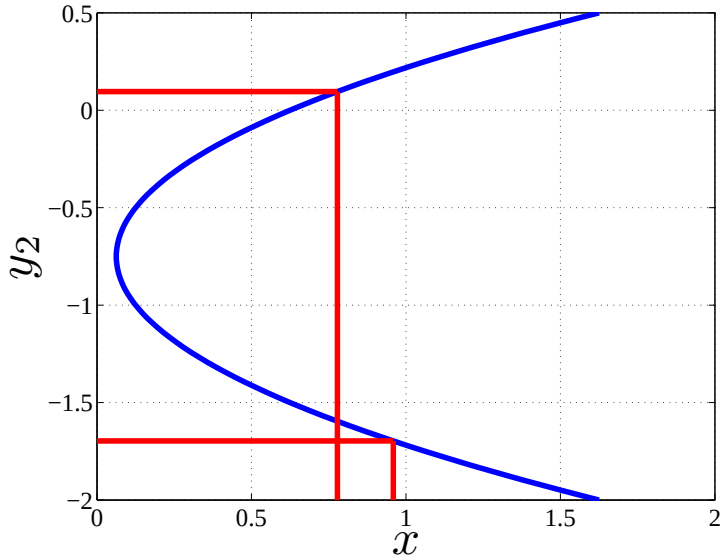
Back Constraints II

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.

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

- 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.

.

Motion Capture Results

`demStick1` and `demStick3`

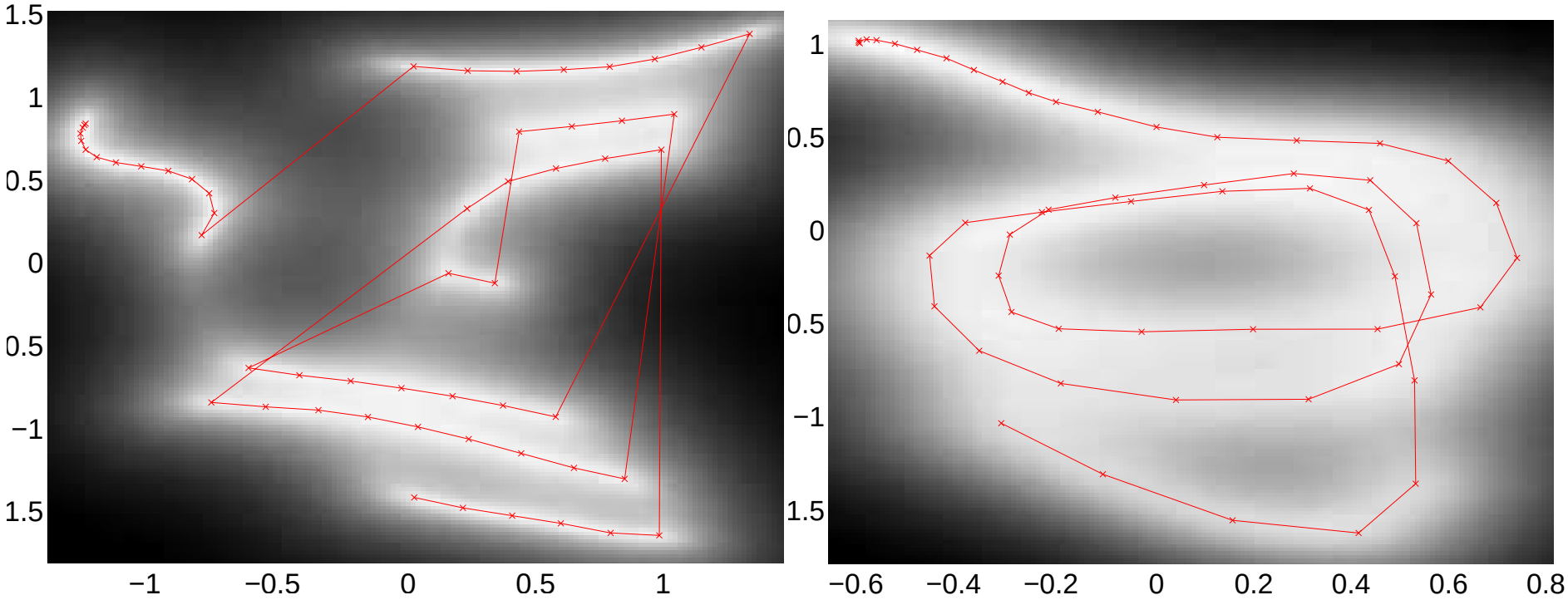
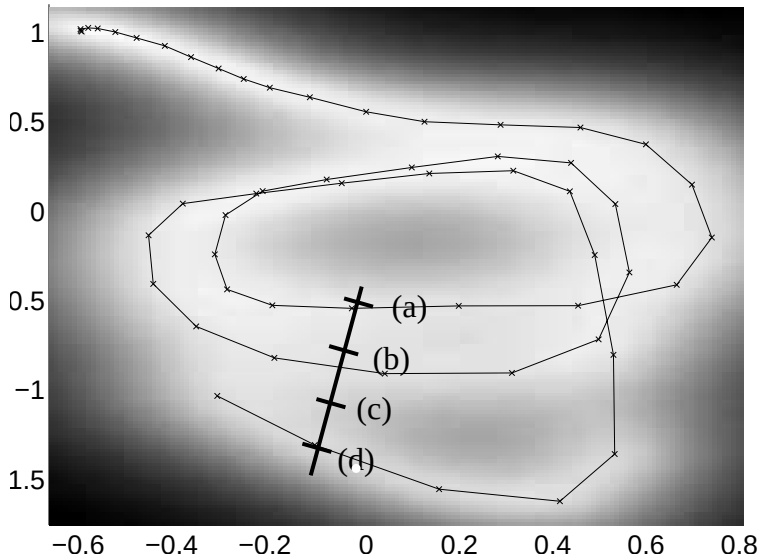


Figure: The latent space for the motion capture data with (*right*) and without (*left*) dynamics. The dynamics us a Gaussian process with an RBF kernel.

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

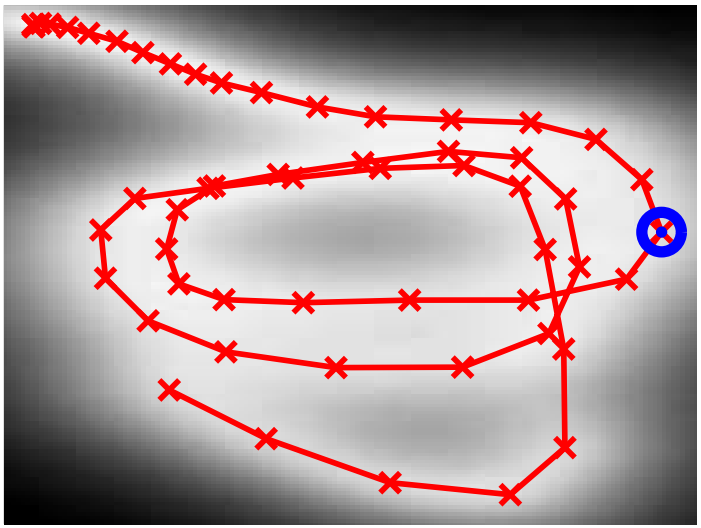
MAP Solutions for Dynamics Models

- 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.

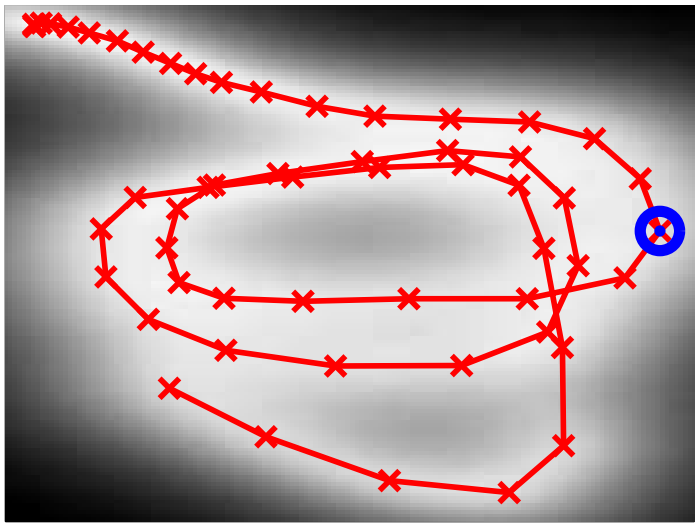


t

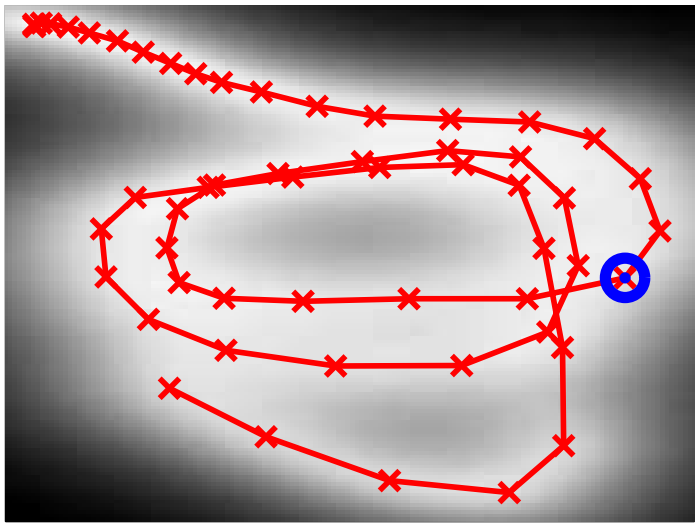
Gaussian Process Dynamics

GP-LVM with Dynamics

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



t

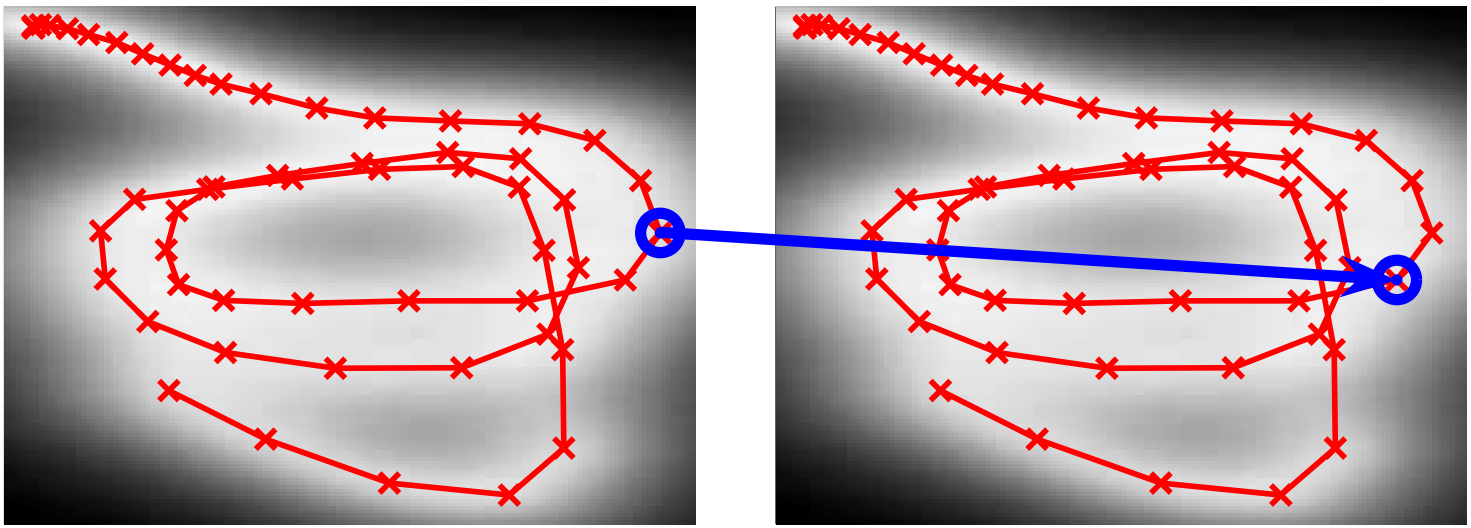


$t+1$

Gaussian Process Dynamics

GP-LVM with Dynamics

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



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`

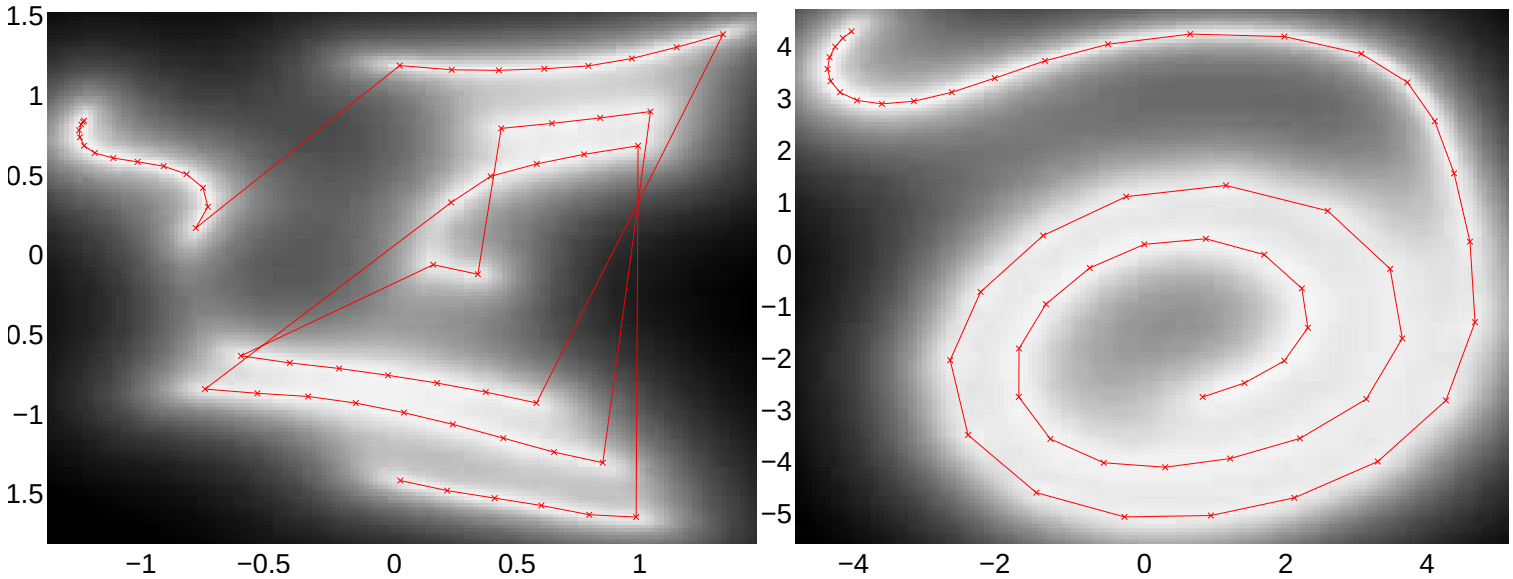
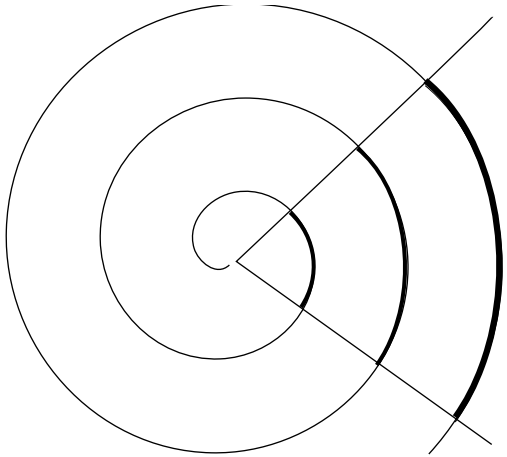


Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*right*) based on an RBF kernel.

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`

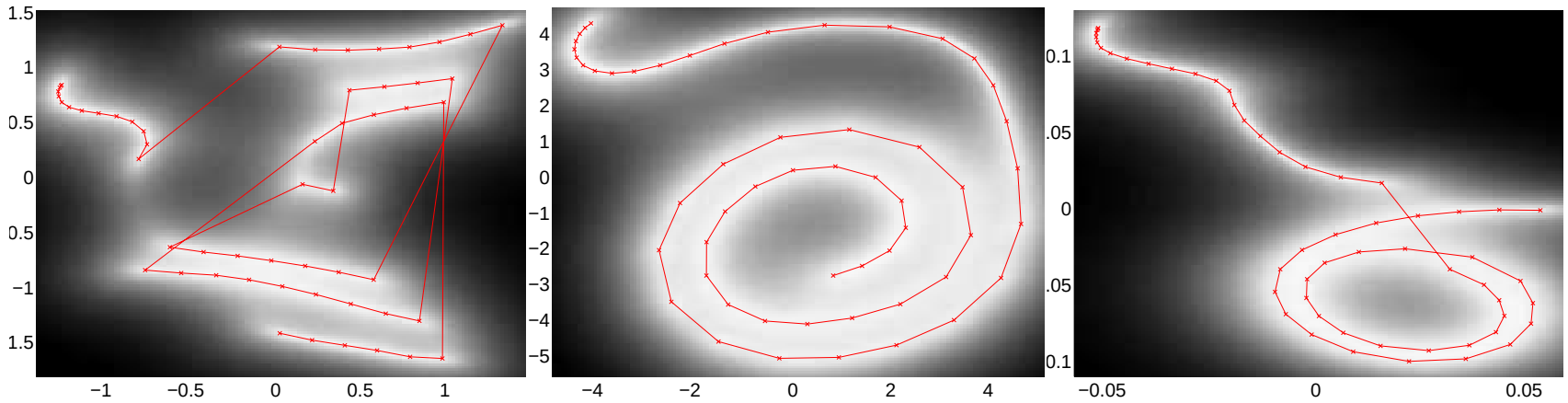


Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*middle*) and with regressive dynamics (*right*) based on an RBF kernel.

Extensions

MAP Solutions for Dynamics Models

- Autoregressive Gaussian Processes. Wang et al. [2006]

Force the Model to Respect Local Distances

- Back constrained GP-LVM.

Developments Made Under Pump Priming Grant

- Sparse Approximations for Large Data Sets
- Hierarchical Models for Subject Decomposition
- Three Dimensional Pose Reconstruction from Images

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`

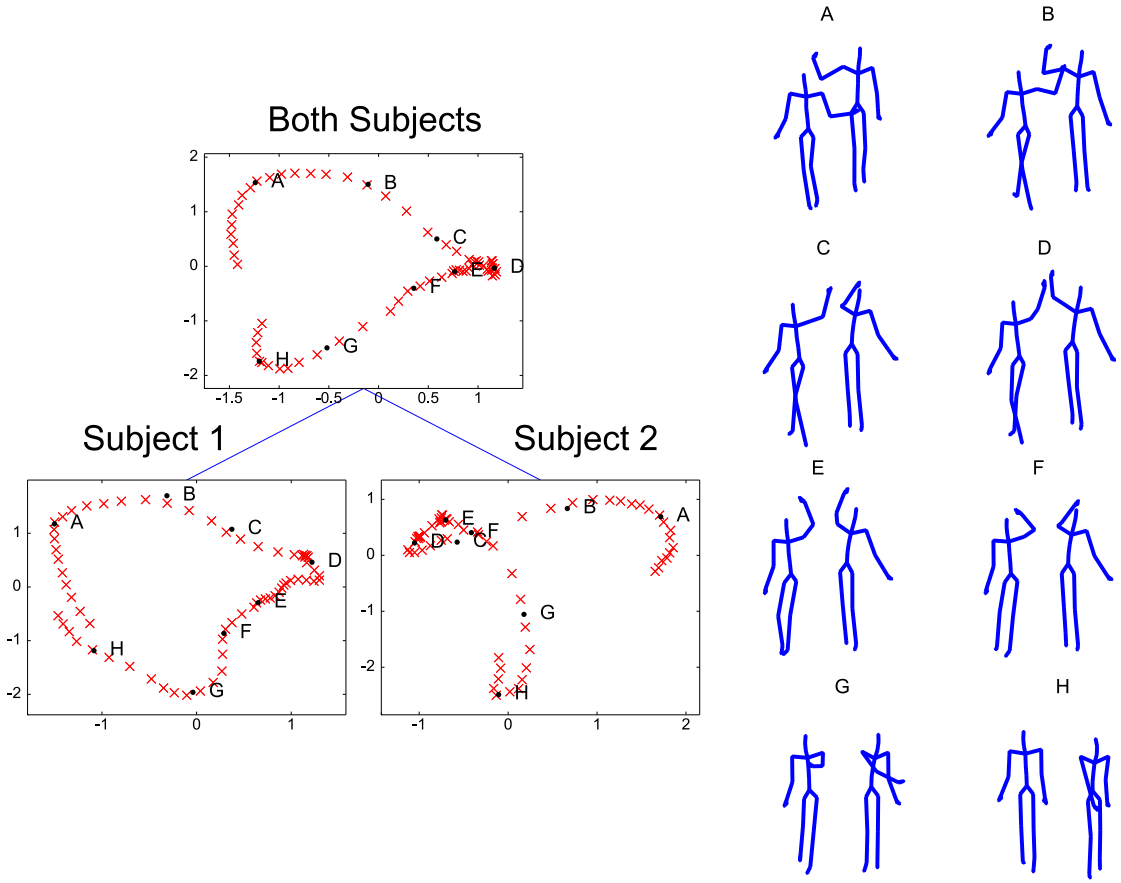


Figure: Hierarchical model of a 'high five'.

Decomposition of Body

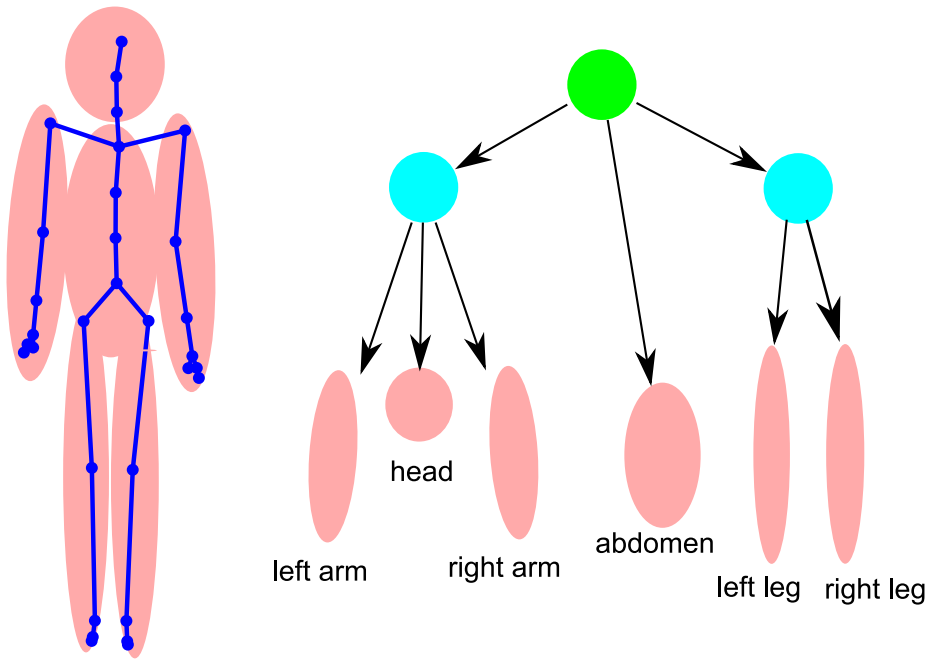


Figure: Decomposition of a subject.

Single Subject Run/Walk

`demRunWalk1`

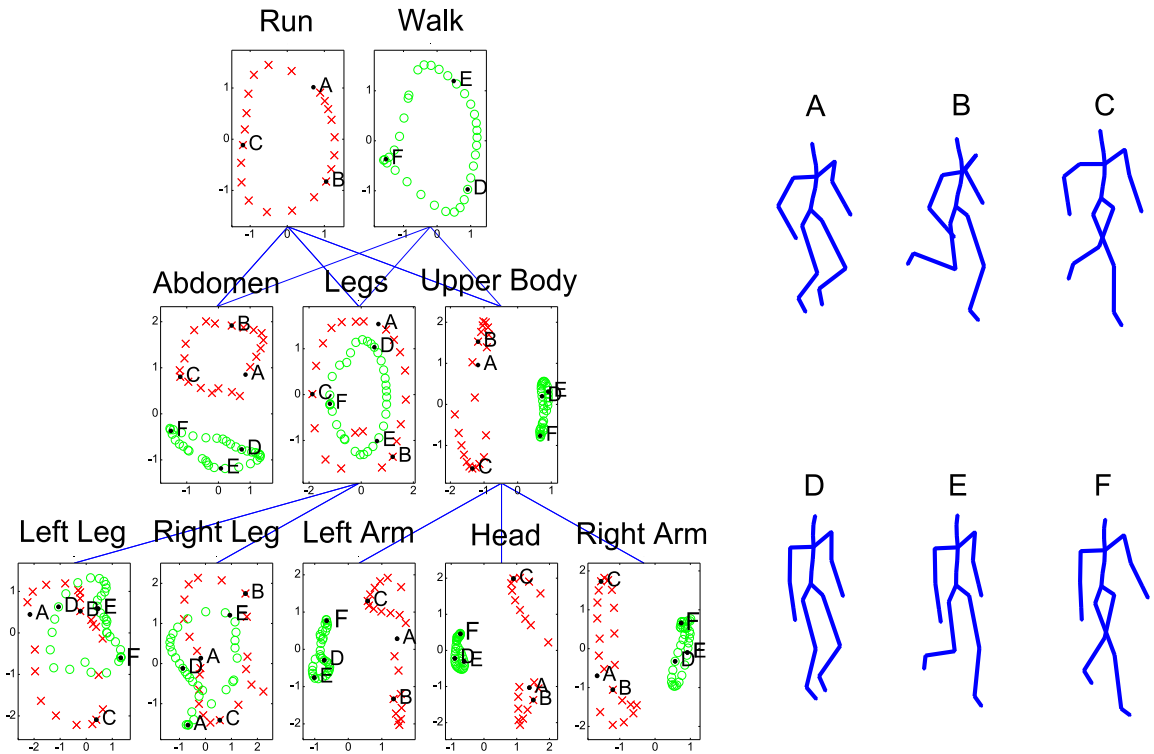


Figure: Hierarchical model of a walk and a run.

Complexity Issues

- Gaussian processes inherently
  - $O\left(N^3\right)$ complexity,
  - $O\left(N^2\right)$ storage.
- Sparse Gaussian processes normally give
  - $O\left(k^2 N\right)$ complexity,
  - $O\left(kN\right)$ storage
- FITC Approximation [Snelson and Ghahramani, 2006, Quiñonero Candela and Rasmussen, 2005, Presented/Developed at PASCAL workshop!].

Sparse GP-LVM

- Recreate results of Taylor et al. [2007] on human motion capture data set.
- Data was walking and running motions from subject 35 in the CMU Mocap data base.
- Used dynamical refinement of the GP-LVM proposed by Wang et al. [2006]
- Taylor et al. [2007] applied their binary latent variable model to two missing data problems
  - right leg was removed from the test seqence
  - upper body was removed.
- Reconstruction obtained compared with nearest neighbour.

FITC Approximation

- Used the FITC approximation with 100 inducing points.
- The models were back constrained [Lawrence and Quiñonero Candela, 2006] .
- The data set size was 2613 frames.

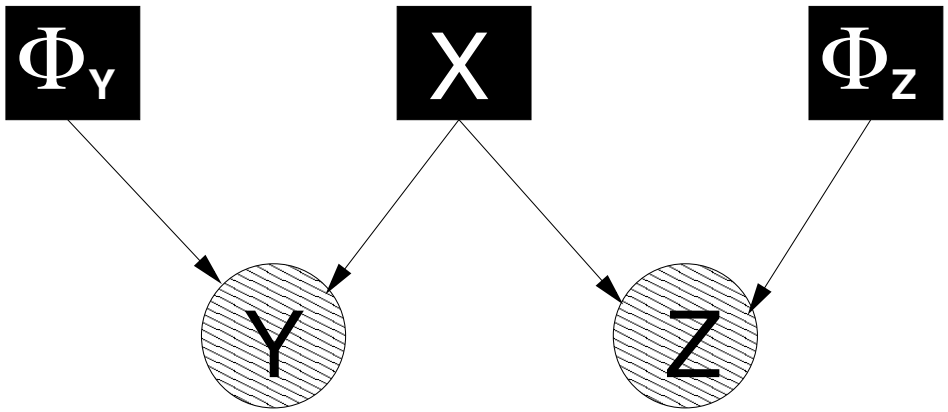
Results

Root mean squared angle error results on test data.

Data	Leg	Body
GP-LVM ($q = 3$)	3.40	**2.49**
GP-LVM ($q = 4$)	**3.38**	2.72
GP-LVM ($q = 5$)	4.25	2.78
NN (s)	4.44	2.62
NN	4.11	3.20

Table: NN: nearest neighbour, NN (s): nearest neighbour in scaled space, GP-LVM (latent dimension): the GP-LVM with different latent dimensions, q.

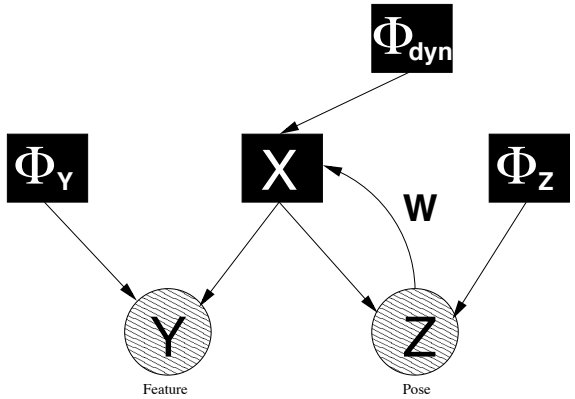
Shared GP-LVM



- Learn two separate kernels from a single shared latent representation $\mathbf{X}$ [Shon et al., 2006]
- Objective

$$p(\mathbf{Y}, \mathbf{Z}|\mathbf{X}, \Phi_Y, \Phi_Z) = p(\mathbf{Y}|\mathbf{X}, \Phi_Y)p(\mathbf{Z}|\mathbf{X}, \Phi_Z)$$

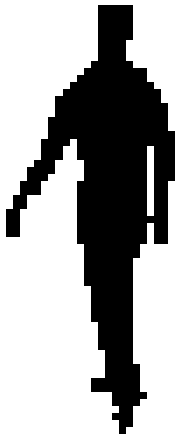
Shared GP-LVM Experiments[1]

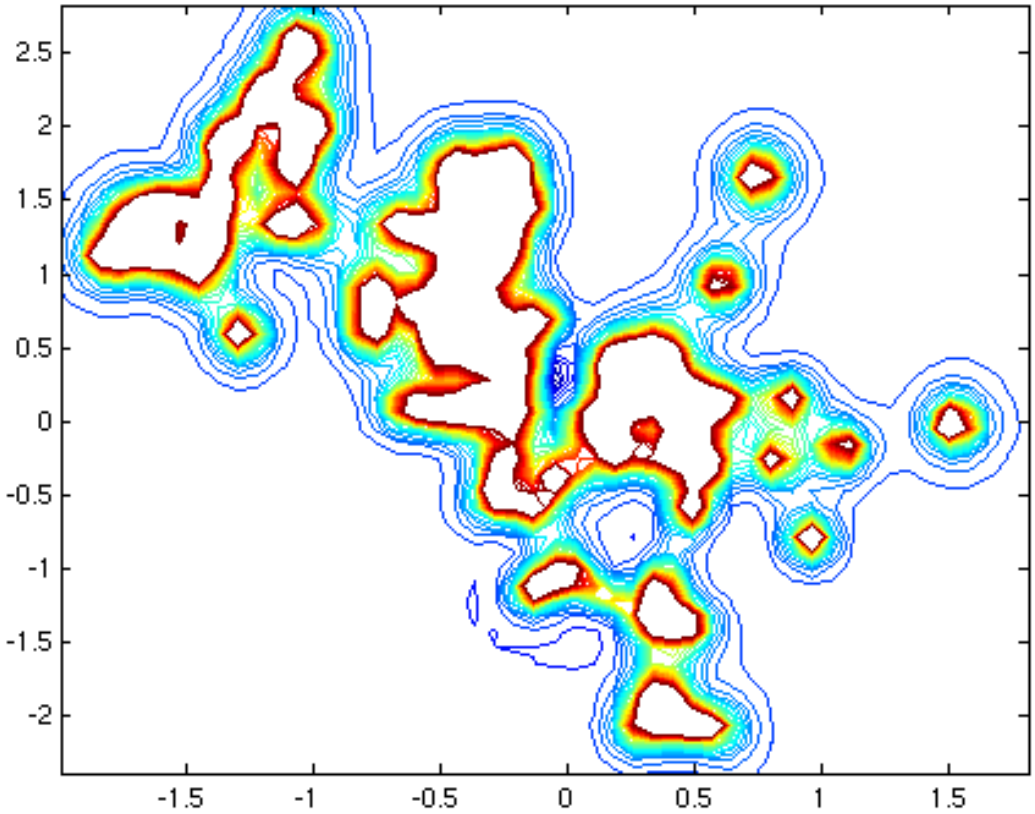


- Silhouette Features: $\mathbf{y}_i \in \Re^{100}$, Pose Parameters: $\mathbf{z}_i \in \Re^{54}$
- **Back constraints:** force bijective mapping between latent space and pose [Lawrence and Quiñonero Candela, 2006].
- **Dynamics:** add GP auto regressive dynamics to latent space [Wang et al., 2006].
- **Artificially generated training data:** from Agarwal and Triggs [2006].

[1] Ek et al. [2007]

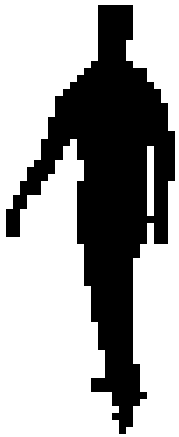
Shared GP-LVM Experiments

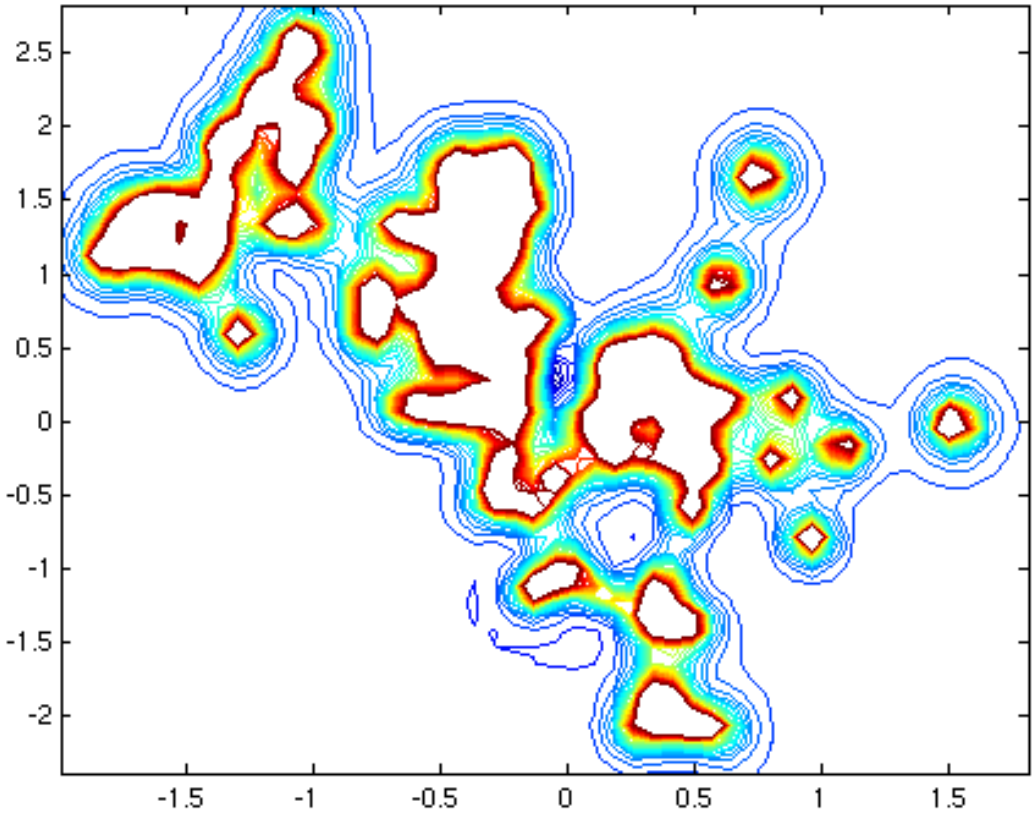




- Highly multimodal latent space given silhouette.

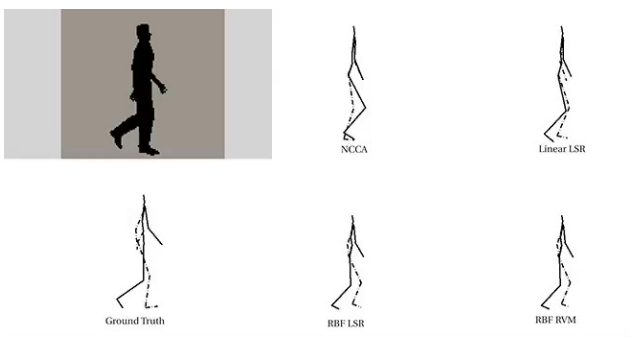
Shared GP-LVM Experiments



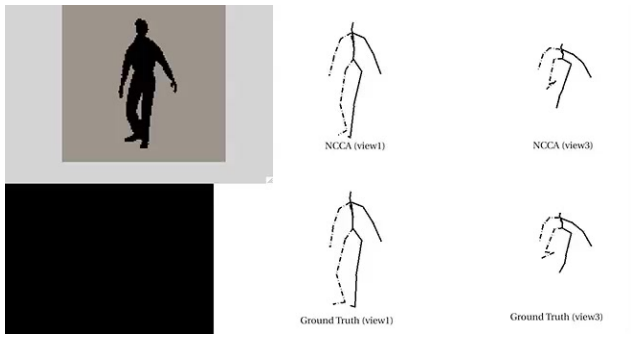


- Highly multimodal latent space given silhouette.

Video

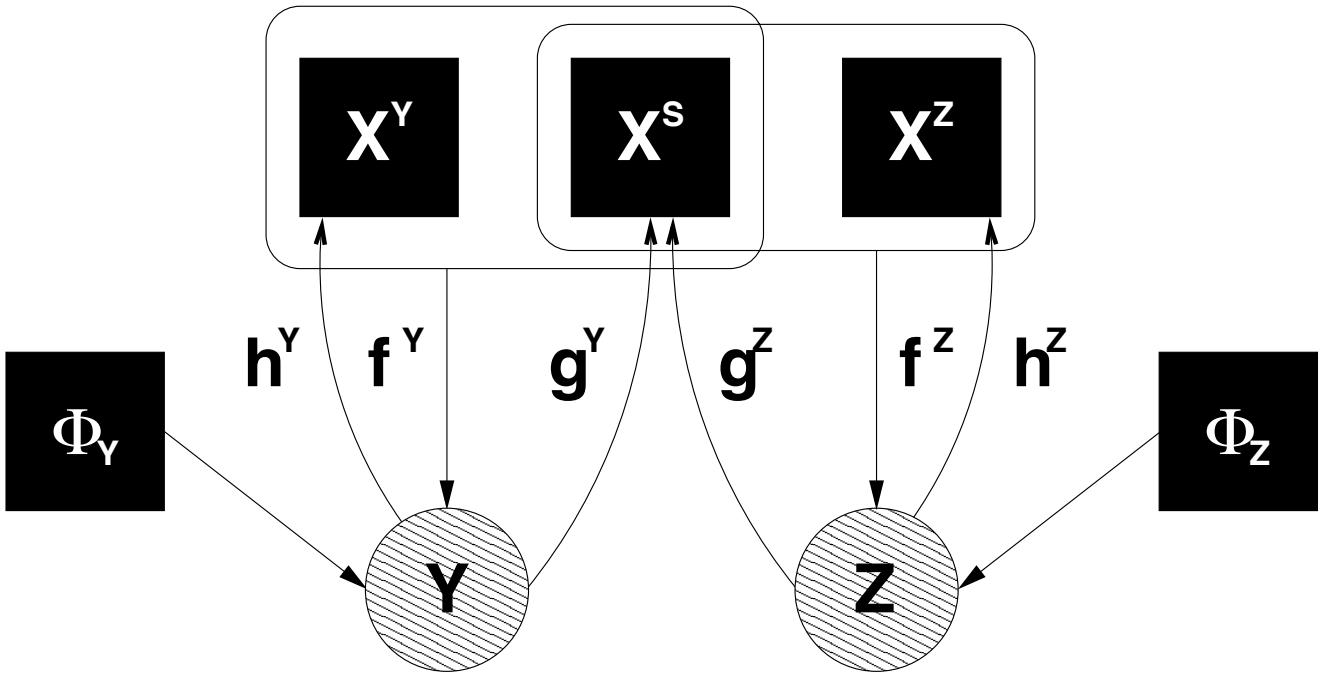


`runcca_all.sh`



`runcca_only.sh`

Modified Model



Initialisation

Shared Latent space by kernel CCA:

- Find directions $\{\mathbf{W_Y}, \mathbf{W_Z}\}$ in each feature space maximizing the correlation
- Canonical variate $\begin{cases} \mathbf{a_Y} & = & \mathbf{Y}\mathbf{W_Y} \\ \mathbf{a_Z} & = & \mathbf{Z}\mathbf{W_Z} \end{cases}$
- Solution through Eigenvalue problem.

Non Shared Latent Space

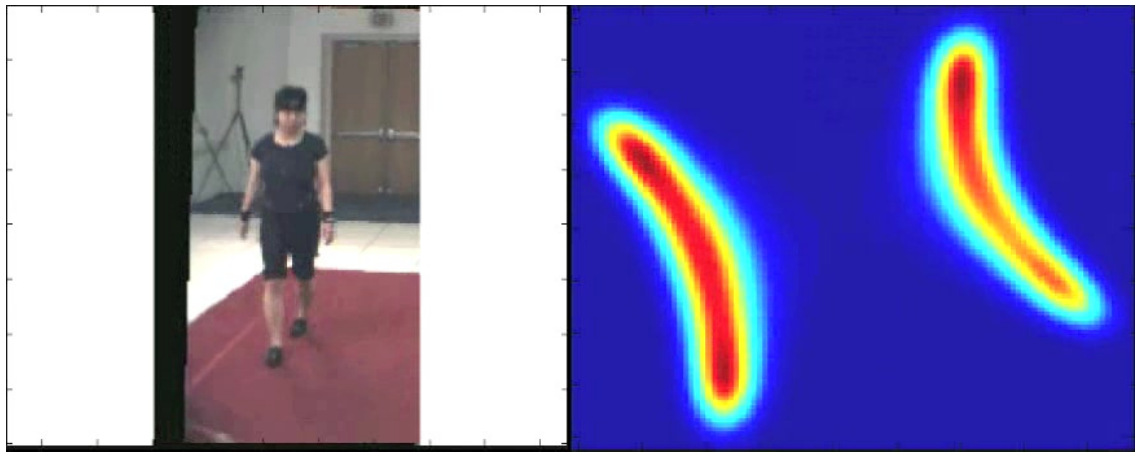
- Find further directions *orthogonal* to CCA directions of maximum variance.
- We named these non-consolidating components.
- Solution through eigenvalue problem.

Kernels

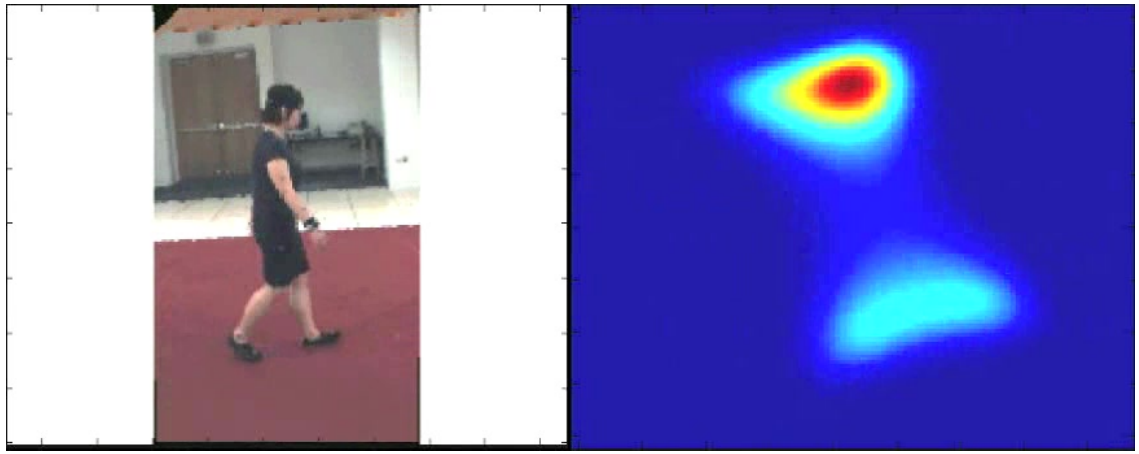
Feature Spaces:

- Many possible choices of feature space
  1. Linear Kernel
  2. RBF
  3. Maximum Variance Unfolding, Isomap
- Choose between them using GP-LVM likelihood [Harmeling, 2007].

Results of Initialisation

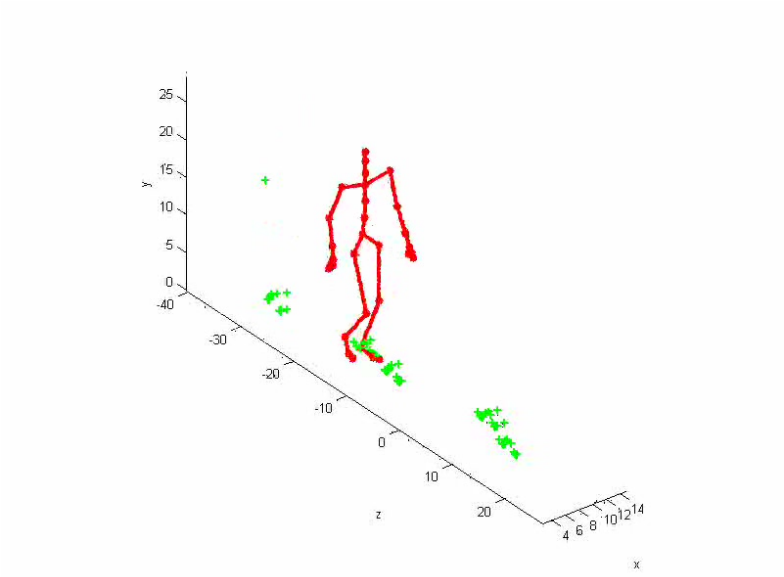


`runspectral_test.sh`

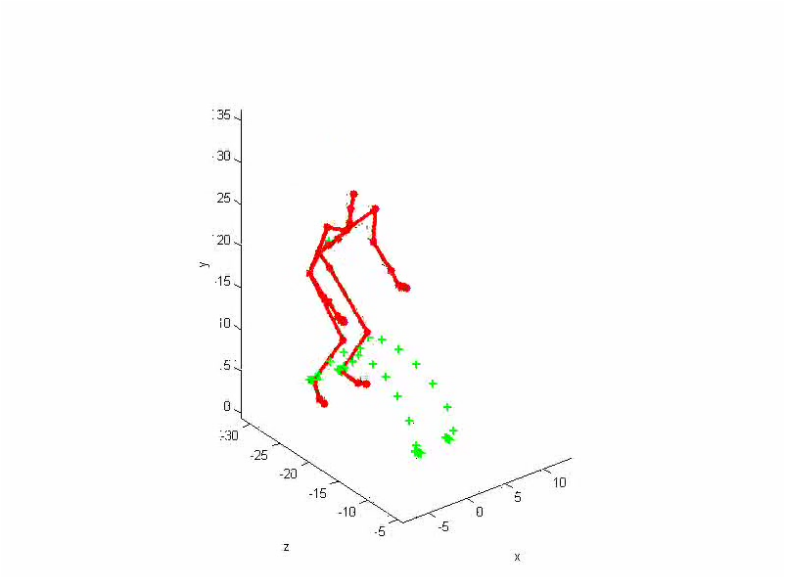


`runspectral_test.sh`

Topological Constraints



`runtopology_walk.sh`



`runtopology_jump.sh`

Outline

Summary

- GP-LVM is a Probabilistic Non-Linear Generalisation of PCA.
- Works Effectively as a Probabilistic Model in High Dimensional Spaces.
- Back constraints can be introduced to force local distance preservation.
- Dynamics can be introduced for modelling data with a temporal structure.
- Hierarchical models can encode conditional independencies.
- Topologically constraints can be imposed.
  -

References

A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1), 2006. doi: 10.1109/TPAMI.2006.21.

C. H. Ek, P. H. Torr, and N. D. Lawrence. Gaussian process latent variable models for human pose estimation. In *MLMI 2007*, 2007. To appear.

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.

S. Harmeling. Exploring model selection techniques for nonlinear dimensionality reduction. Technical Report EDI-INF-RR-0960, University of Edinburgh, 2007.

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 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.

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.

A. P. Shon, K. Grochow, A. Hertzmann, and R. P. N. Rao. Learning shared latent structure for image synthesis and robotic imitation. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.

G. W. Taylor, G. E. Hinton, and S. Roweis. Modeling human motion using binary latent variables. In B. Schölkopf, J. C. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems*, volume 19, Cambridge, MA, 2007. 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