Introduction to Gaussian Processes

Raquel Urtasun

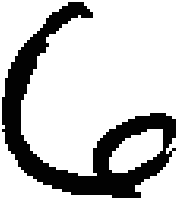
TTI Chicago

August 2, 2013

Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

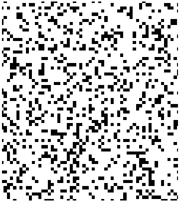
- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.



Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

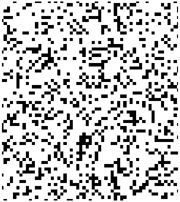
- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

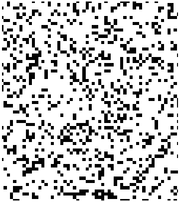
- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



Motivation for Non-Linear Dimensionality Reduction

USPS Data Set Handwritten Digit

- 3648 Dimensions
  - 64 rows by 57 columns
  - Space contains more than just this digit.
  - Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

Rotate a 'Prototype'

Simple Model of Digit

Rotate a 'Prototype'

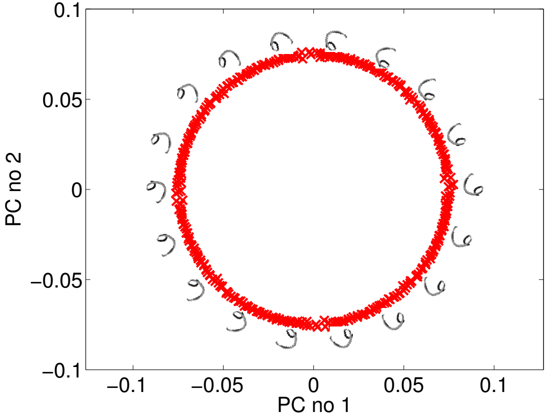
Rotate a 'Prototype'

MATLAB Demo

```
demDigitsManifold([1 2], 'all')
```

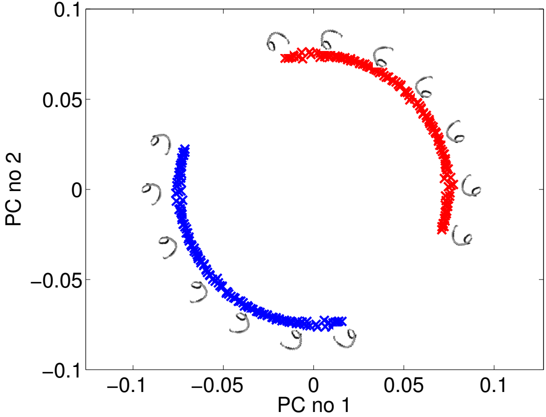
MATLAB Demo

`demDigitsManifold([1 2], 'all')`



MATLAB Demo

```
demDigitsManifold([1 2], 'sixnine')
```



Low Dimensional Manifolds

Pure Rotation is too Simple

- In practice the data may undergo several distortions.
  - *e.g.* digits undergo 'thinning', translation and rotation.
- For data with 'structure':
  - we expect fewer distortions than dimensions;
  - we therefore expect the data to live on a lower dimensional manifold.
- Conclusion: deal with high dimensional data by looking for lower dimensional non-linear embedding.

Feature Selection

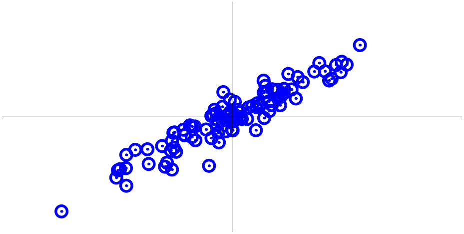


Figure: `demRotationDist`. Feature selection via distance preservation.

Feature Selection

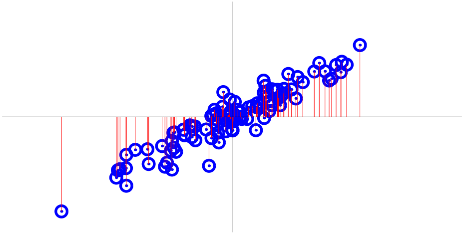


Figure: `demRotationDist`. Feature selection via distance preservation.

Feature Selection

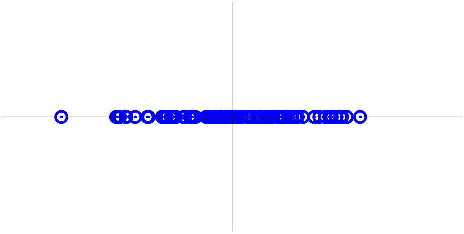


Figure: `demRotationDist`. Feature selection via distance preservation.

Feature Extraction

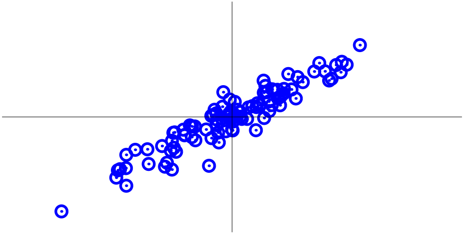


Figure: `demRotationDist`. Rotation preserves interpoint distances.

Feature Extraction

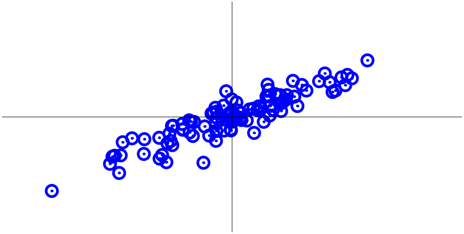


Figure: `demRotationDist`. Rotation preserves interpoint distances.

Feature Extraction

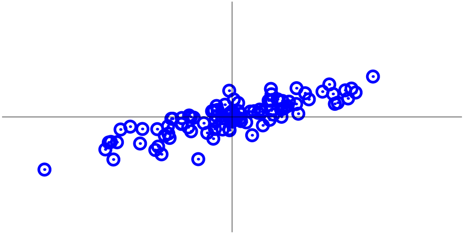


Figure: `demRotationDist`. Rotation preserves interpoint distances.

Feature Extraction

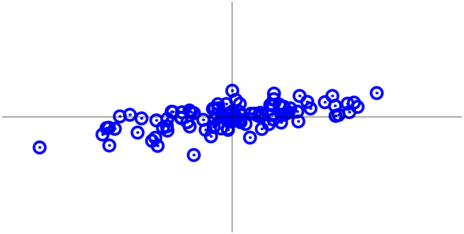


Figure: `demRotationDist`. Rotation preserves interpoint distances.

Feature Extraction

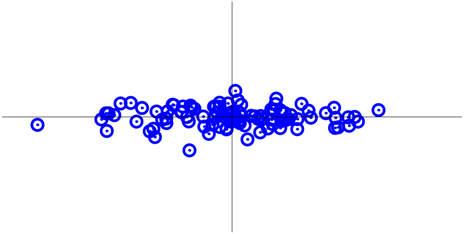


Figure: `demRotationDist`. Rotation preserves interpoint distances.

Feature Extraction

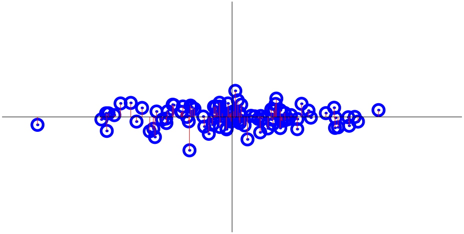


Figure: `demRotationDist`. Rotation preserves interpoint distances. Residuals are much reduced.

Feature Extraction

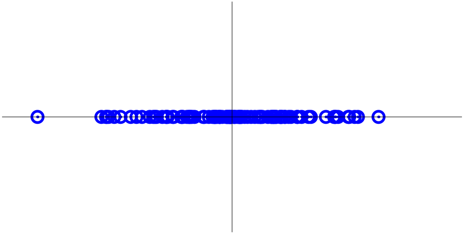


Figure: `demRotationDist`. Rotation preserves interpoint distances. Residuals are much reduced.

Which Rotation?

- We need the rotation that will minimise residual error.
- Retain features/directions with *maximum variance*.

Which Rotation?

- We need the rotation that will minimise residual error.
- Retain features/directions with *maximum variance*.
- Error is then given by the sum of residual variances.

$$E(\mathbf{X}) = \frac{2}{p} \sum_{k=q+1}^{p} \sigma_k^2.$$

Which Rotation?

- We need the rotation that will minimise residual error.
- Retain features/directions with *maximum variance*.
- Error is then given by the sum of residual variances.

$$E(\mathbf{X}) = \frac{2}{p} \sum_{k=q+1}^{p} \sigma_k^2.$$

- Rotations of data matrix *do not* effect this analysis.
- Rotate data so that largest variance directions are retained.

Which Rotation?

- We need the rotation that will minimise residual error.
- Retain features/directions with *maximum variance*.
- Error is then given by the sum of residual variances.

$$E(\mathbf{X}) = \frac{2}{p} \sum_{k=q+1}^{p} \sigma_k^2.$$

- Rotations of data matrix *do not* effect this analysis.
- Rotate data so that largest variance directions are retained.

Reminder: Principal Component Analysis

- How do we find these directions?
- Find directions in data with maximal variance.
  - That's what PCA does!
- **PCA**: rotate data to extract these directions.
- **PCA**: work on the sample covariance matrix $\mathbf{S} = n^{-1}\hat{\mathbf{Y}}^\top \hat{\mathbf{Y}}$.

Principal Coordinates Analysis

- The rotation which finds directions of maximum variance is the eigenvectors of the covariance matrix.
- The variance in each direction is given by the eigenvalues.
- **Problem:** working directly with the sample covariance, $\mathbf{S}$, may be impossible.
- Why?

Equivalent Eigenvalue Problems

- Principal Coordinate Analysis operates on $\hat{\mathbf{Y}}^T\hat{\mathbf{Y}} \in \Re^{p\times p}$.
- Can we compute $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^\top$ instead?
- When $p < n$ it is easier to solve for the rotation, $\mathbf{R}_q$. But when $p > n$ we solve for the embedding (principal coordinate analysis).
- Two eigenvalue problems are equivalent: One solves for the rotation, the other solves for the location of the rotated points.

The Covariance Interpretation

- $n^{-1}\hat{\mathbf{Y}}^\top\hat{\mathbf{Y}}$ is the data covariance.
- $\hat{\mathbf{Y}}\hat{\mathbf{Y}}^\top$ is a centred inner product matrix.
    - Also has an interpretation as a covariance matrix (Gaussian processes).
    - It expresses correlation and anti correlation between *data points*.
    - Standard covariance expresses correlation and anti correlation between *data dimensions*.

Mapping of Points

- Mapping points to higher dimensions is easy.

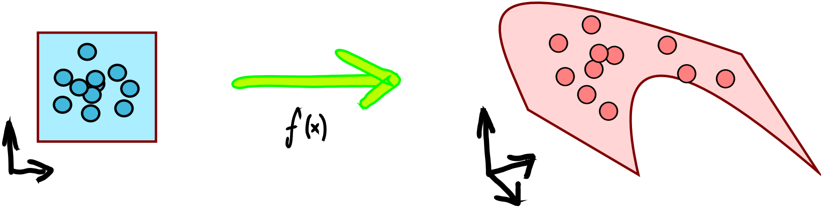


Figure: Two dimensional Gaussian mapped to three dimensions.

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

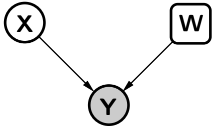
$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:},$$

where

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

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:

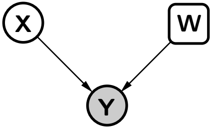


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

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*, $\mathbf{X}$.

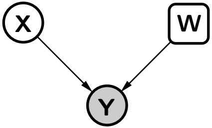


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

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*.

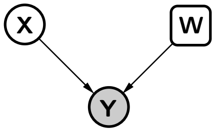


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

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

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*.

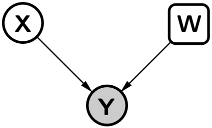


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

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

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

Probabilistic PCA Max. Likelihood Soln (Tipping 99)



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

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping 99)

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

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

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

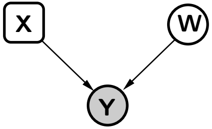
$$\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{R}^\top, \quad \mathbf{L} = \left(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

where $\mathbf{R}$ 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:

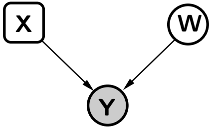


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}\mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

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*, $\mathbf{W}$.

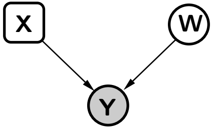


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

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*.

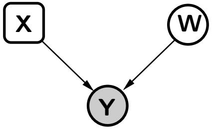


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

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

Dual Probabilistic PCA

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



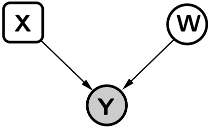
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

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

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

Linear Latent Variable Model IV

Dual Probabilistic PCA Max. Likelihood Soln (Lawrence 03, Lawrence 05)



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

Linear Latent Variable Model IV

Dual Probabilistic PCA Max. Likelihood Soln (Lawrence 03, Lawrence 05)

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

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

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

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

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

Linear Latent Variable Model IV

Probabilistic PCA Max. Likelihood Soln (Tipping 99)

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

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

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

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

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

The Eigenvalue Problems are equivalent

- Solution for Probabilistic PCA (solves for the mapping)

$$\mathbf{Y}^\top \mathbf{Y} \mathbf{U}_q = \mathbf{U}_q \boldsymbol{\Lambda}_q \qquad \mathbf{W} = \mathbf{U}_q \mathbf{L} \mathbf{R}^\top$$

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

$$\mathbf{Y} \mathbf{Y}^\top \mathbf{U}'_q = \mathbf{U}'_q \boldsymbol{\Lambda}_q \qquad \mathbf{X} = \mathbf{U}'_q \mathbf{L} \mathbf{R}^\top$$

- Equivalence is from

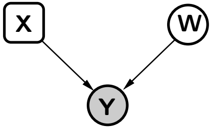
$$\mathbf{U}_q = \mathbf{Y}^\top \mathbf{U}'_q \boldsymbol{\Lambda}_q^{-\frac{1}{2}}$$

- You have probably used this trick to compute PCA efficiently when number of dimensions is much higher than number of points.

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*.



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

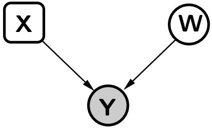
$$p(\mathbf{W}) = \prod_{i=1}^{p} \mathcal{N}\left(\mathbf{w}_{i,:}|\mathbf{0},\mathbf{I}\right)$$

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

Non-Linear Latent Variable Model

Dual Probabilistic PCA

- Inspection of the marginal likelihood shows ...
  - The covariance matrix is a covariance function.

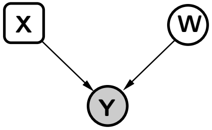


$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}\right)$$

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'.



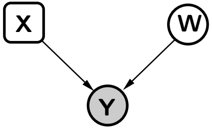
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

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

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).



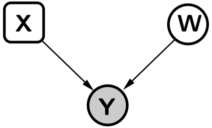
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

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

This is a product of Gaussian processes with linear kernels.

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).



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

$$\mathbf{K} = ?$$

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

Non-linear Latent Variable Models

Exponentiated Quadratic (EQ) Covariance

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

$$k\left(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}\right) = \alpha \exp\left(-\frac{\left\|\mathbf{x}_{i,:} - \mathbf{x}_{j,:}\right\|_2^2}{2\ell^2}\right).$$

- No longer possible to optimise wrt $\mathbf{X}$ via an eigenvalue problem.

Exponentiated Quadratic (EQ) Covariance

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

$$k\left(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}\right) = \alpha \exp\left(-\frac{\left\|\mathbf{x}_{i,:} - \mathbf{x}_{j,:}\right\|_2^2}{2\ell^2}\right).$$

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

$$\operatorname*{argmin}_{\mathbf{X},\alpha,\ell,\sigma} \frac{p}{2} \log\left|K(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}\right| + \frac{p}{2} tr\left(\left(K(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}\right)^{-1}\mathbf{Y}\mathbf{Y}^T\right)$$

Exponentiated Quadratic (EQ) Covariance

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

$$k\left(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}\right) = \alpha \exp\left(-\frac{\left\|\mathbf{x}_{i,:} - \mathbf{x}_{j,:}\right\|_2^2}{2\ell^2}\right).$$

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

$$\underset{\mathbf{X},\alpha,\ell,\sigma}{\operatorname{argmin}} \frac{p}{2} \log \left|K(\mathbf{X},\mathbf{X}) + \sigma^2 \mathbf{I}\right| + \frac{p}{2} tr\left(\left(K(\mathbf{X},\mathbf{X}) + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{Y}\mathbf{Y}^T\right)$$

Let's look at some applications

[K. Grochow, S. Martin, A. Hertzmann and Z. Popovic, Siggraph 2004]

- Learn a GPLVM from a small mocap sequence
- Smooth the latent space by adding noise in order to reduce the number of local minima.
- Let's replay the same motion

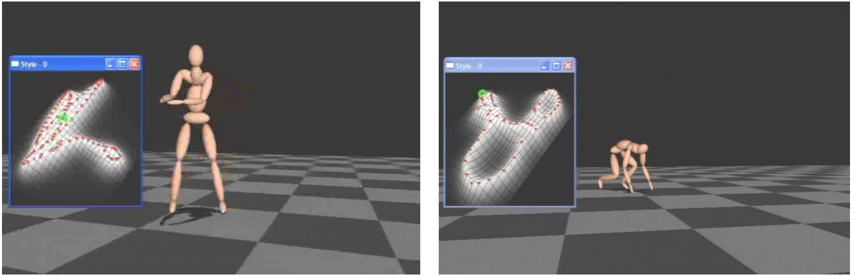


Figure: Syle-IK

1) GPLVM for Character Animation

[K. Grochow, S. Martin, A. Hertzmann and Z. Popovic, Siggraph 2004]

- Pose synthesis by solving an optimization problem

$$\arg\min_{\mathbf{x},\mathbf{y}} -\log p(\mathbf{y}|\mathbf{x})$$
$$\text{such that} \quad C(\mathbf{y}) = 0$$

- Constraints from a user in an interactive session or from a mocap system

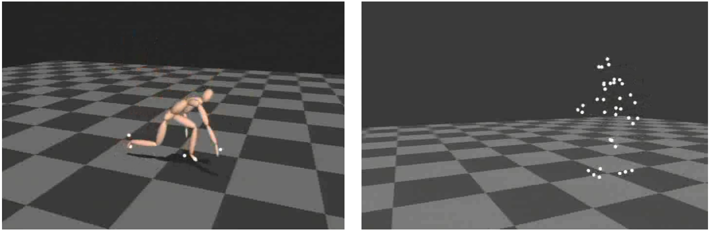
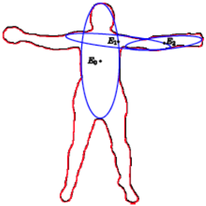


Figure: Syle-IK

2) Shape Priors in Level Set Segmentation

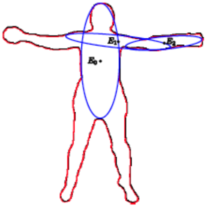
- Represent contours with elliptic Fourier descriptors



- Learn a GPLVM on the parameters of those descriptors

2) Shape Priors in Level Set Segmentation

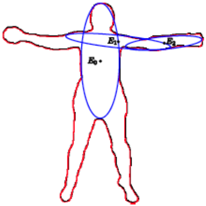
- Represent contours with elliptic Fourier descriptors



- Learn a GPLVM on the parameters of those descriptors
- We can now generate closed contours from the latent space

2) Shape Priors in Level Set Segmentation

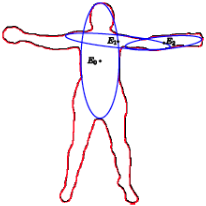
- Represent contours with elliptic Fourier descriptors



- Learn a GPLVM on the parameters of those descriptors
- We can now generate closed contours from the latent space
- Segmentation is done by non-linear minimization of an image-driven energy which is a function of the latent space

2) Shape Priors in Level Set Segmentation

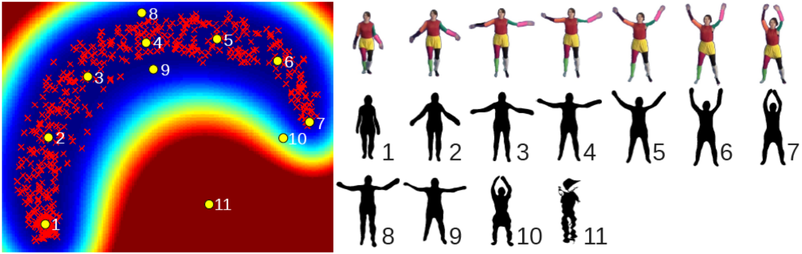
- Represent contours with elliptic Fourier descriptors

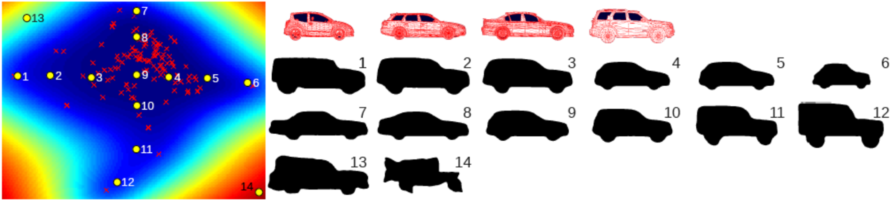


- Learn a GPLVM on the parameters of those descriptors
- We can now generate closed contours from the latent space
- Segmentation is done by non-linear minimization of an image-driven energy which is a function of the latent space

GPLVM on Contours

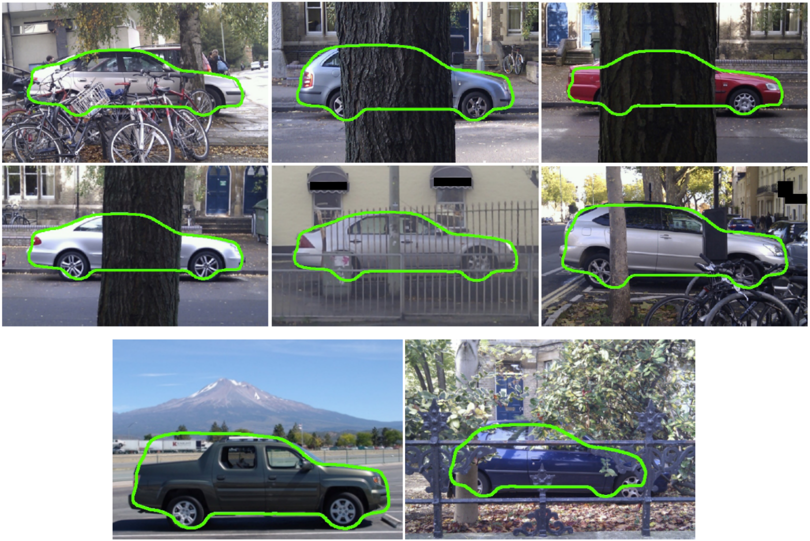
[V. Prisacariu and I. Reid, ICCV 2011]



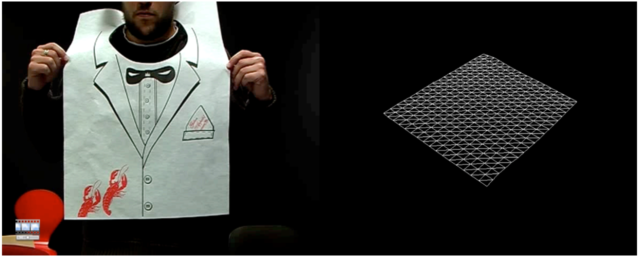


Segmentation Results

[V. Prisacariu and I. Reid, ICCV 2011]



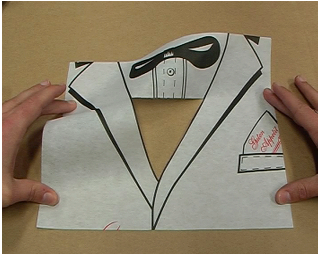
3) Non-rigid shape deformation



Monocular 3D shape recovery is severely under-constrained:

- Complex deformations and low-texture objects.
- Deformation models are required to disambiguate.
- Building realistic physics-based models is very complex.
- Learning the models is a popular alternative.

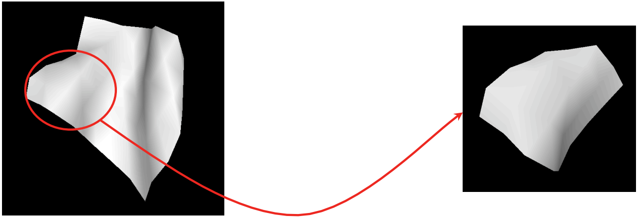
Global deformation models



State-of-the-art techniques learn global models that

- require large amounts of training data,
- must be learned for each new object.

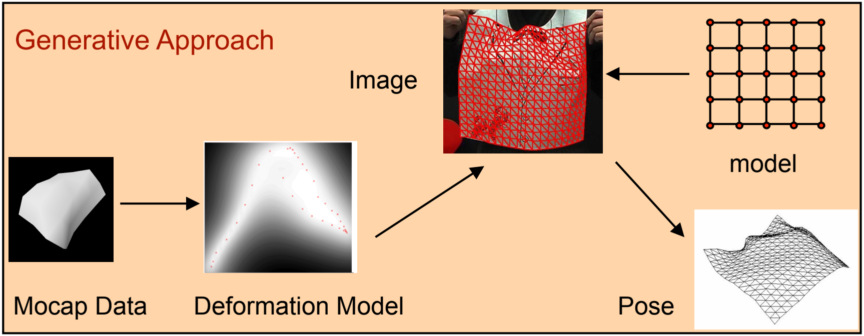
Key observations



1. Locally, all parts of a physically homogeneous surface obey the same deformation rules.
2. Deformations of small patches are much simpler than those of a global surface, and thus can be learned from fewer examples.

⇒ Learn Local Deformation Models and combine them into a global one representing the particular shape of the object of interest.

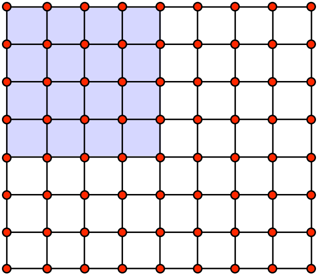
Overview of the method



Combining the deformations

Use a Product of Experts (POE) paradigm (Hinton 99):

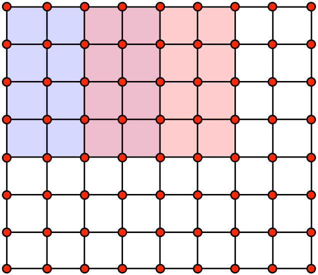
- High dimensional data subject to low dimensional constraints.
- A global deformation should be composed of highly probable local ones.
- For homogeneous materials, all local patches follow the same deformation rules.
- Learn a single local model, and replicate it to cover the whole object.



Combining the deformations

Use a Product of Experts (POE) paradigm (Hinton 99):

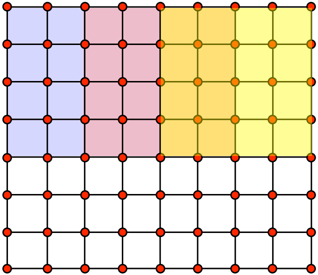
- High dimensional data subject to low dimensional constraints.
- A global deformation should be composed of highly probable local ones.
- For homogeneous materials, all local patches follow the same deformation rules.
- Learn a single local model, and replicate it to cover the whole object.



Combining the deformations

Use a Product of Experts (POE) paradigm (Hinton 99):

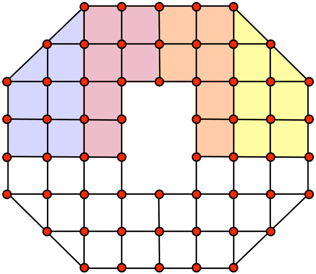
- High dimensional data subject to low dimensional constraints.
- A global deformation should be composed of highly probable local ones.
- For homogeneous materials, all local patches follow the same deformation rules.
- Learn a single local model, and replicate it to cover the whole object.



Combining the deformations

Use a Product of Experts (POE) paradigm (Hinton 99):

- High dimensional data subject to low dimensional constraints.
- A global deformation should be composed of highly probable local ones.
- For homogeneous materials, all local patches follow the same deformation rules.
- Learn a single local model, and replicate it to cover the whole object.



$\Rightarrow$ Same deformation model represents arbitrary shapes and topologies.

Tracking

- For each image I_t we have to estimate the state $\phi_t = (\mathbf{y}_t, \mathbf{x}_t)$.
- Bayesian formulation of the tracking

$$p(\phi_t|\mathbf{I}_t, \mathbf{X}, \mathbf{Y}) \propto p(\mathbf{I}_t|\phi_t)p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{X}, \mathbf{Y})p(\mathbf{x}_t)$$

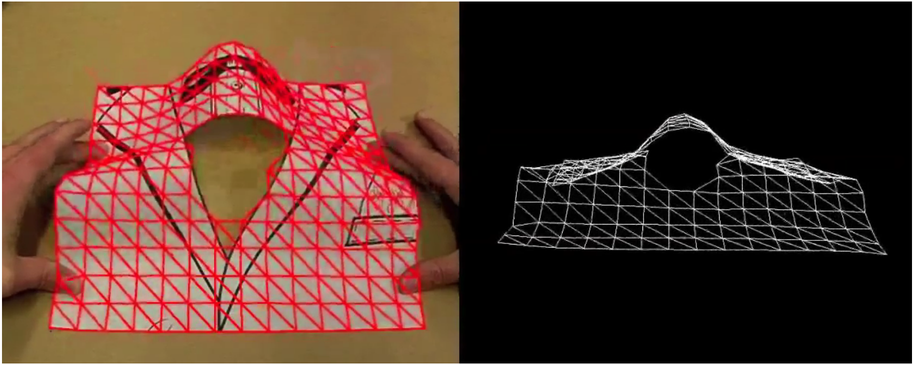
- The image likelihood is composed of texture (template matching) and edge information

$$p(\mathbf{I}_t|\phi_t) = p(\mathbf{T}_t|\phi_t)p(\mathbf{E}_t|\phi_t)$$

- Tracking by minimizing the posterior

Shape deformation estimation

[M. Salzmann, **R. Urtasun** and P. Fua, CVPR 2008]



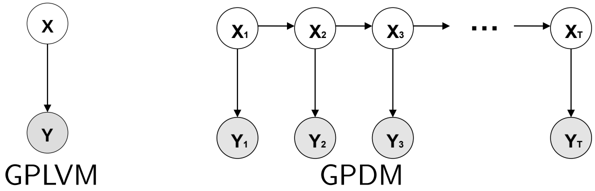
Incorporating dynamics

- The mapping from latent space to high dimensional space as

$$\mathbf{y}_{i,:} = \mathbf{W}\psi(\mathbf{x}_{i,:}) + \boldsymbol{\eta}_{i,:}, \quad \text{where} \quad \eta_{i,:} \sim N\left(\mathbf{0}, \sigma^2 \mathbf{I}\right).$$

- We can augment the model with ARMA dynamics. This is called Gaussian process dynamical models (GPDM) (Wang et al., 05).

$$\mathbf{x}_{t+1,:} = \mathbf{P}\phi(\mathbf{x}_{t:t-\tau,:}) + \boldsymbol{\gamma}_{i,:}, \quad \text{where} \quad \gamma_{i,:} \sim N\left(\mathbf{0}, \sigma_d^2 \mathbf{I}\right).$$



Model Learned for tracking

Model learned from 6 walking subjects,1 gait cycle each, on treadmill at same speed with a 20 DOF joint parameterization (no global pose)

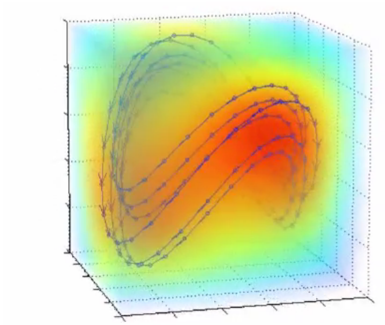


Figure: Density

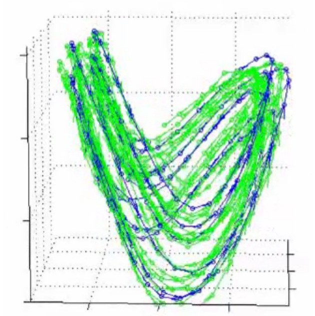
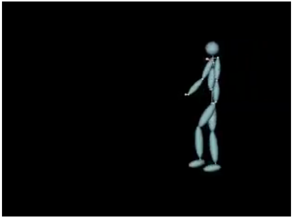
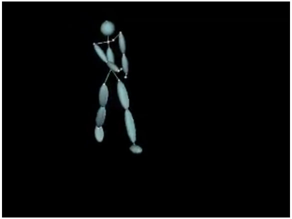


Figure: Randomly generated trajectories

Tracking results

[R. Urtasun, D. Fleet and P. Fua, CVPR 2006]





Estimated latent trajectories

[R. Urtasun, D. Fleet and P. Fua, CVPR 2006]

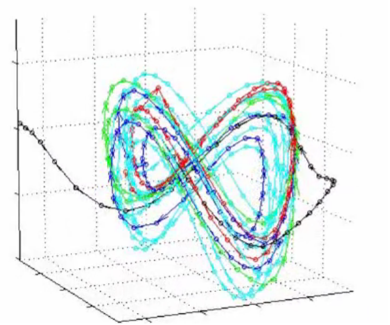
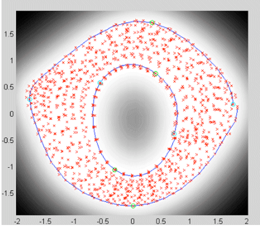
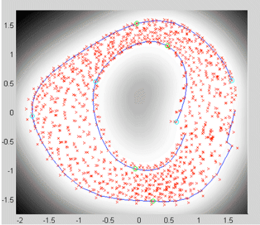


Figure: Estimated latent trajectories. (cian) - training data, (black) - exaggerated walk, (blue) - occlusion.

Visualization of Knee Pathology

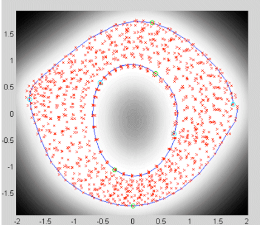
Two subjects, four walk gait cycles at each of 9 speeds (3-7 km/hr)

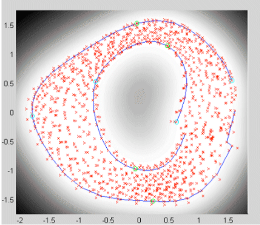




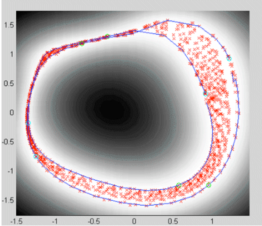
Visualization of Knee Pathology

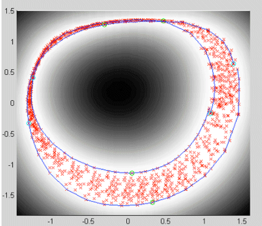
Two subjects, four walk gait cycles at each of 9 speeds (3-7 km/hr)





Two subjects with a knee pathology.





Does it work all the time?

Is training with so little data a bug or a feature?

Problems with the GPLVM

- It relies on the optimization of a non-convex function

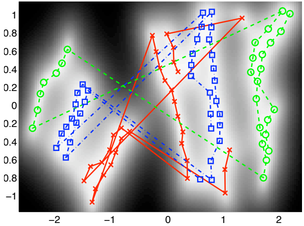
$$\mathcal{L} = \frac{p}{2} \ln |\mathbf{K}| + \frac{p}{2} tr(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) \ .$$

Problems with the GPLVM

- It relies on the optimization of a non-convex function

$$\mathcal{L} = \frac{p}{2} \ln |\mathbf{K}| + \frac{p}{2} tr(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) \ .$$

- Even with the right dimensionality, they can result in poor representations if initialized far from the optimum.

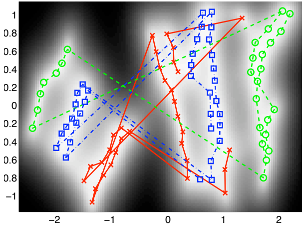


Problems with the GPLVM

- It relies on the optimization of a non-convex function

$$\mathcal{L} = \frac{p}{2} \ln |\mathbf{K}| + \frac{p}{2} tr(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) \ .$$

- Even with the right dimensionality, they can result in poor representations if initialized far from the optimum.



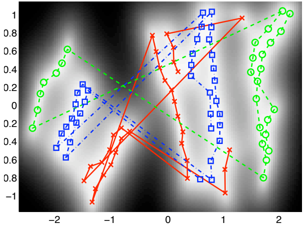
- This is even worst if the dimensionality of the latent space is small.

Problems with the GPLVM

- It relies on the optimization of a non-convex function

$$\mathcal{L} = \frac{p}{2} \ln |\mathbf{K}| + \frac{p}{2} tr(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) \ .$$

- Even with the right dimensionality, they can result in poor representations if initialized far from the optimum.



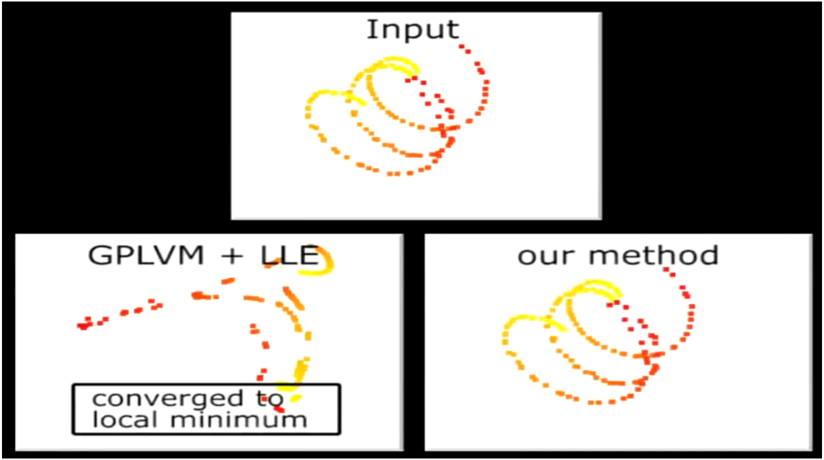
- This is even worst if the dimensionality of the latent space is small.
- As a consequence these models have only been applied to small databases of a single activity.

Solutions

- **Back-constraints:** Constrain the inverse mapping to be smooth [Lawrence et al. 06]
- **Topologies:** Add smoothness and topological priors, e.g., style content separation [Urtasun et al. 08]
- **Dynamics:** to smooth the latent space [Wang et al. 06]
- **Continuous dimensionality reduction:** Add rank priors and reduce the dimensionality as you do the optimization [Geiger et al. 09]
- **Stochastic:** learning algorithms [Lawrence et al. 09]
- etc

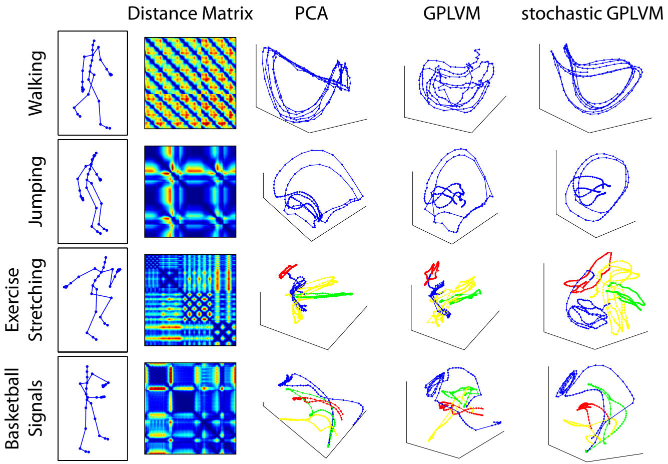
Continuous dimensionality reduction

[A. Geiger, R. Urtasun and T. Darrell, CVPR 2009]



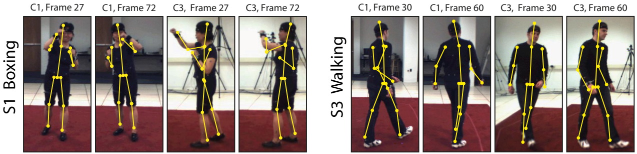
Stochastic Algorithms

[A. Yao, J. Gall, L. Van Gool and R. Urtasun, NIPS 2011]



Humaneva Results

[A. Yao, J. Gall, L. Van Gool and R. Urtasun, NIPS 2011]



Train	Test	[Xu07]	[Li10]	GPLVM	CRBM	imCRBM	Ours
S1	S1	-	-	57.6 ± 11.6	48.8 ± 3.7	58.6 ± 3.9	**44.0 ± 1.8**
S1,2,3	S1	140.3	-	64.3 ± 19.2	55.4 ± 0.8	54.3 ± 0.5	**41.6 ± 0.8**
S2	S2	-	68.7 ± 24.7	98.2 ± 15.8	**47.4 ± 2.9**	67.0 ± 0.7	54.4 ± 1.8
S1,2,3	S2	149.4	-	155.9 ± 48.8	99.1 ± 23.0	69.3 ± 3.3	**64.0 ± 2.9**
S3	S3	-	69.6 ± 22.2	71.6 ± 10.0	49.8 ± 2.2	51.4 ± 0.9	**45.4 ± 1.1**
S1,2,3	S3	156.3	-	123.8. ± 16.7	70.9 ± 2.1	**43.4 ± 4.1**	46.5 ± 1.4

Model	Tracking Error
[Pavlovic00] as reported in [Li07]	569.90 ± 209.18
[Lin06] as reported in [Li07]	380.02 ± 74.97
GPLVM	121.44 ± 30.7
[Li07]	117.0 ± 5.5
Best CRBM [Taylor10]	75.4 ± 9.7
Ours	**74.1 ± 3.3**

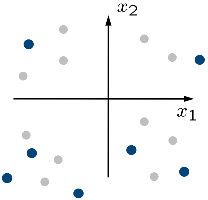
Other extensions

1) Priors for supervised learning

- We introduce a prior that is based on the Fisher criteria

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{\sigma_d^2} tr\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\right\} ,$$

with $\mathbf{S}_b$ the between class matrix and $\mathbf{S}_w$ the within class matrix

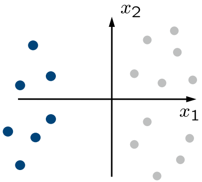


1) Priors for supervised learning

- We introduce a prior that is based on the Fisher criteria

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{\sigma_d^2} tr\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\right\} ,$$

with $\mathbf{S}_b$ the between class matrix and $\mathbf{S}_w$ the within class matrix



$$\mathbf{S}_b = \sum_{i=1}^{L} \frac{n_i}{N}(\mathbf{M}_i - \mathbf{M}_0)(\mathbf{M}_i - \mathbf{M}_0)^T$$

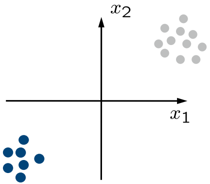
where $\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \cdots, \mathbf{x}_{n_i}^{(i)}]$ are the n_i training points of class i, $\mathbf{M}_i$ is the mean of the elements of class i, and $\mathbf{M}_0$ is the mean of all the training points of all classes.

1) Priors for supervised learning

- We introduce a prior that is based on the Fisher criteria

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{\sigma_d^2} tr\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\right\} \;,$$

with $\mathbf{S}_b$ the between class matrix and $\mathbf{S}_w$ the within class matrix



$$\mathbf{S}_b = \sum_{i=1}^{L} \frac{n_i}{N}(\mathbf{M}_i - \mathbf{M}_0)(\mathbf{M}_i - \mathbf{M}_0)^T$$

$$\mathbf{S}_w = \sum_{i=1}^{L} \frac{n_i}{n}\left[\frac{1}{n_i}\sum_{k=1}^{N_i}(\mathbf{x}_k^{(i)} - \mathbf{M}_i)(\mathbf{x}_k^{(i)} - \mathbf{M}_i)^T\right]$$

where $\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \cdots, \mathbf{x}_{n_i}^{(i)}]$ are the n_i training points of class i, $\mathbf{M}_i$ is the mean of the elements of class i, and $\mathbf{M}_0$ is the mean of all the training points of all classes.

- As before the model is learned by maximizing $p(\mathbf{Y}|\mathbf{X})p(\mathbf{X})$.

1) Priors for supervised learning

- We introduce a prior that is based on the Fisher criteria

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{\sigma_d^2} tr\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right)\right\} ,$$

with $\mathbf{S}_b$ the between class matrix and $\mathbf{S}_w$ the within class matrix

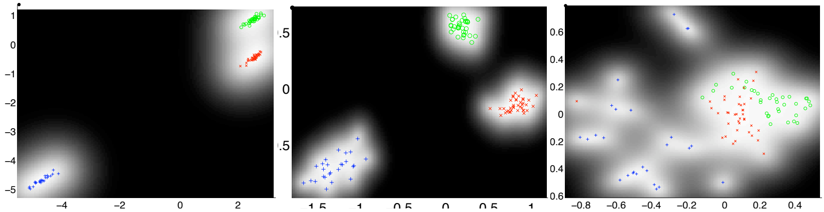
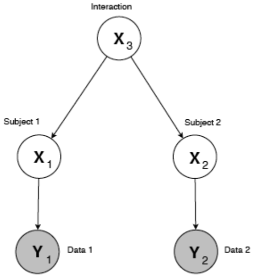


Figure: 2D latent spaces learned by D-GPLVM on the oil dataset for different values of σ_d [Urtasun et al. 07].

2) 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.

Within Subject Hierarchy

Decomposition of Body

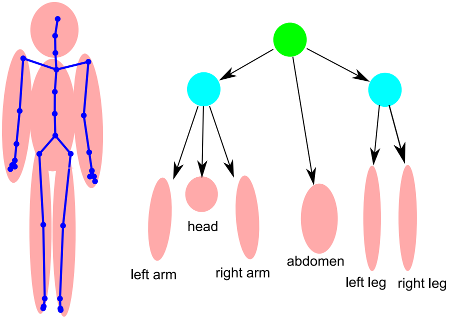


Figure: Decomposition of a subject.

Single Subject Run/Walk

[N. Lawrence and A. Moore, ICML 2007]

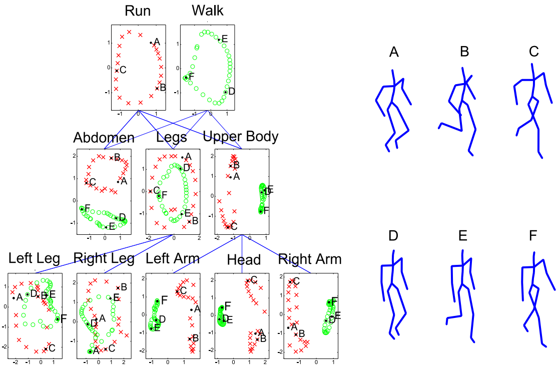
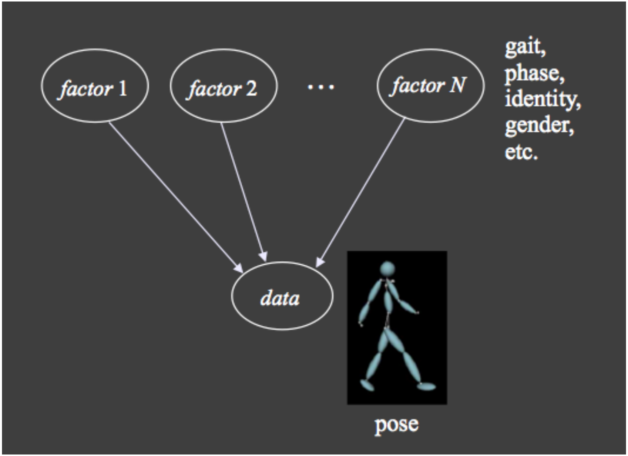


Figure: Hierarchical model of a walk and a run.

3) Style Content Separation and Multi-linear models

Multiple aspects that affect the input signal, interesting to factorize them



Multilinear models

- Style-Content Separation (Tenenbaum & Freeman 00)

$$\mathbf{y} = \sum_{ij} w_{ij} a_i b_j + \epsilon$$

- Multi-linear analysis (Vasilescu & Terzopoulous 02)

$$\mathbf{y} = \sum_{ijk\cdots} w_{ijk\ldots} a_i b_j c_k \cdots + \epsilon$$

- Non-linear basis functions (Elgammal & Lee, 2004)

$$\mathbf{y} = \sum_{ij} w_{ij} a_i \phi_j(b) + \epsilon$$

Multi (non)-linear models with GPs

- In the GPLVM

$$\mathbf{y} = \sum_j w_j \phi_j(\mathbf{x}) + \epsilon = \mathbf{w}^T \Phi(\mathbf{x}) + \epsilon$$

with

$$\mathbb{E}[\mathbf{y}, \mathbf{y}'] = \Phi(\mathbf{x})^T \Phi(\mathbf{y}) + \beta^{-1}\delta = k(\mathbf{x}, \mathbf{x}') + \beta^{-1}\delta$$

- Multifactor Gaussian process

$$\mathbf{y} = \sum_{i,j,k,\cdots} w_{ijk\cdots} \phi_i^{(1)} \phi_j^{(1)} \phi_k^{(1)} \cdots + \epsilon$$

with

$$\mathbb{E}[\mathbf{y}, \mathbf{y}'] = \prod_i {\Phi^{(i)}}^T \Phi^{(i)} + \beta^{-1}\delta = \prod_i k_i(\mathbf{x}^{(i)}, {\mathbf{x}^{(i)}}') + \beta^{-1}\delta$$

Multi (non)-linear models with GPs

- In the GPLVM

$$\mathbf{y} = \sum_j w_j \phi_j(\mathbf{x}) + \epsilon = \mathbf{w}^T \Phi(\mathbf{x}) + \epsilon$$

with

$$\mathbb{E}[\mathbf{y}, \mathbf{y}'] = \Phi(\mathbf{x})^T \Phi(\mathbf{y}) + \beta^{-1}\delta = k(\mathbf{x}, \mathbf{x}') + \beta^{-1}\delta$$

- Multifactor Gaussian process

$$\mathbf{y} = \sum_{i,j,k,\cdots} w_{ijk\cdots} \phi_i^{(1)} \phi_j^{(1)} \phi_k^{(1)} \cdots + \epsilon$$

with

$$\mathbb{E}[\mathbf{y}, \mathbf{y}'] = \prod_i {\Phi^{(i)}}^T \Phi^{(i)} + \beta^{-1}\delta = \prod_i k_i(\mathbf{x}^{(i)}, {\mathbf{x}^{(i)}}') + \beta^{-1}\delta$$

- Learning in this model is the same, just the kernel changes.

Multi (non)-linear models with GPs

- In the GPLVM

$$\mathbf{y} = \sum_j w_j \phi_j(\mathbf{x}) + \epsilon = \mathbf{w}^T \Phi(\mathbf{x}) + \epsilon$$

with

$$\mathbb{E}[\mathbf{y}, \mathbf{y}'] = \Phi(\mathbf{x})^T \Phi(\mathbf{y}) + \beta^{-1}\delta = k(\mathbf{x}, \mathbf{x}') + \beta^{-1}\delta$$

- Multifactor Gaussian process

$$\mathbf{y} = \sum_{i,j,k,\cdots} w_{ijk\cdots} \phi_i^{(1)} \phi_j^{(1)} \phi_k^{(1)} \cdots + \epsilon$$

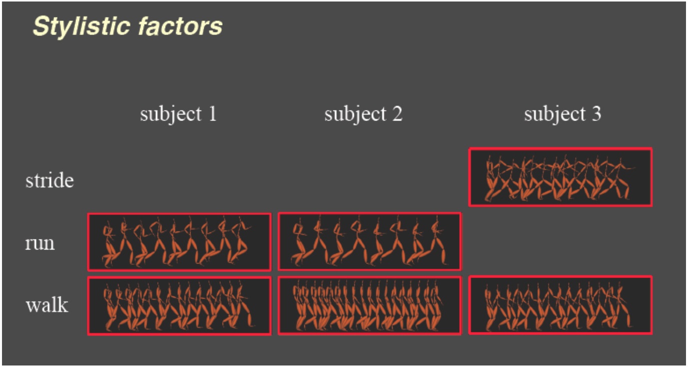
with

$$\mathbb{E}[\mathbf{y}, \mathbf{y}'] = \prod_i {\Phi^{(i)}}^T \Phi^{(i)} + \beta^{-1}\delta = \prod_i k_i(\mathbf{x}^{(i)}, {\mathbf{x}^{(i)}}') + \beta^{-1}\delta$$

- Learning in this model is the same, just the kernel changes.

Training Data

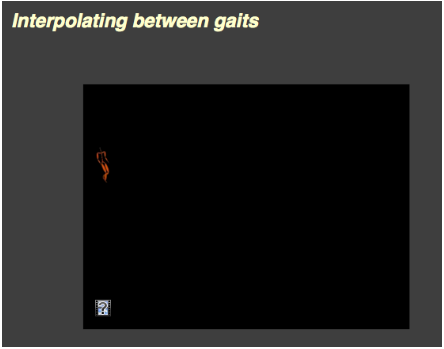
Each training motion is a collection of poses, sharing the same combination of subject (s) and gait (g).

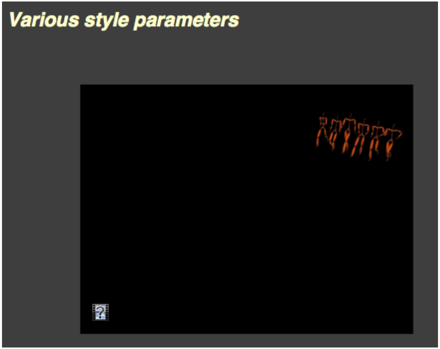


Training data, 6 sequences, 314 frames in total

Results

[J. Wang, D. Fleet and A. Hertzmann, ICML 2007]





4) Continuous Character Control

- When employing GPLVM, different motions get too far apart
- Difficult to generate animations where we transition between motions
- Back-constraints or topologies are not enough
- New prior that enforces connectivity in the graph

$$\ln p(\mathbf{X}) = w_c \sum_{i,j} \ln K_{ij}^d$$

with the **graph diffusion kernel $\mathbf{K}^d$** obtain from

$$K_{ij}^d = \exp(\beta \mathbf{H}) \qquad \text{with} \qquad \mathbf{H} = -\mathbf{T}^{-1/2}\mathbf{L}\mathbf{T}^{-1/2}$$

the graph Laplacian, and $\mathbf{T}$ is a diagonal matrix with $T_{ii} = \sum_j w(\mathbf{x}_i, \mathbf{x}_j)$,

$$L_{ij} = \begin{cases} \sum_k w(\mathbf{x}_i, \mathbf{x}_k) & \text{if } i = j \\ -w(\mathbf{x}_i, \mathbf{x}_j) & \text{otherwise.} \end{cases}$$

and $w(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||^{-p}$ measures similarity.

Embeddings: Walking

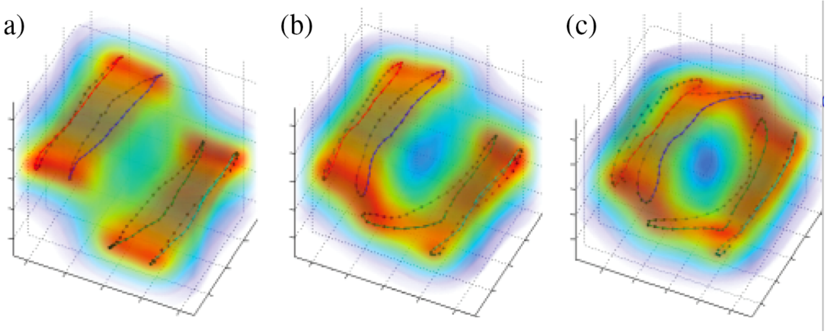


Figure: Walking embeddings learned (a) without the connectivity term, (b) with $w_c = 0{:}1$, and (c) with $w_c = 1{:}0$.

Embeddings: Punching

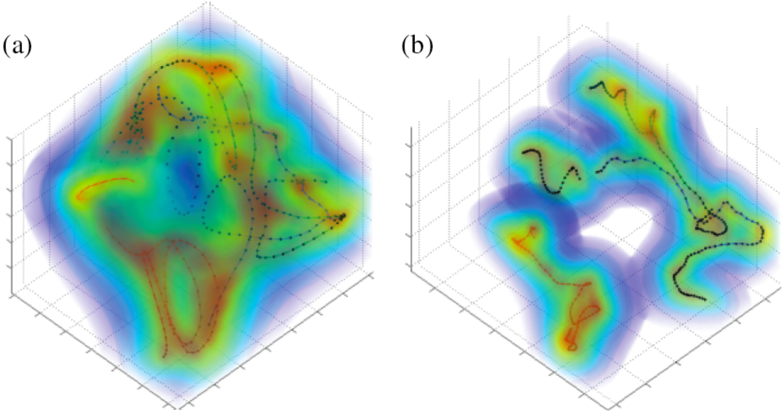


Figure: Embeddings for the punching task (a) with and (b) without the connectivity term.

Video Results

[S. Levine, J. Wang, A. Haraux, Z. Popovic and V. Koltun, Siggraph 2012]