Gaussian Processes and Probabilistic Models for Dimensionality Reduction

Neil D. Lawrence
Departments of Neuro- and Computer Science, University of Sheffield, U.K.

Schloss Dagstuhl

25th August 2011

Outline

Outline

Notation

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

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

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

Reading Notation

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

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

Spectral Dimensionality Reduction in Machine Learning

- Spectral approach to dimensionality reduction.
  1. Convert data to a matrix of dimension $n \times n$.
  2. Visualize data with eigenvectors of matrix.
- Examples:
  - Isomap (Tenenbaum et al., 2000),
  - locally linear embeddings (LLE, Roweis and Saul, 2000),
  - Laplacian eigenmaps (LE, Belkin and Niyogi, 2003) and
  - maximum variance unfolding (MVU, Weinberger et al., 2004).
  - Also kernel PCA (Schölkopf et al., 1998; Ham et al., 2004).

Classical Multidimensional Scaling Perspective

- Classical multidimensional scaling (CMDS)
  1. Compute an $n \times n$ squared distance matrix, $\mathbf{D}$.
  2. Form the centered "similarity matrix" $\mathbf{HKH} = -\frac{1}{2}\mathbf{HDH}$.
  3. Visualize through q principal eigenvectors (as latent matrix $\mathbf{X}$).
- This algorithm matches squared distances computed in $\mathbf{X}$ to those computed in $\mathbf{Y}$ through an L1 error.
- Our Argument:
  - Main innovation in ML work: how to compute the squared distance matrix $\mathbf{D}$.

Isomap

- MDS finds geometric configuration preserving distances.
- MDS applied to Manifold distance.
- Geodesic Distance = Manifold Distance.
- Cannot compute geodesic distance without knowing manifold.
- Idea: compute distance via shortest path between point-pairs Tenenbaum et al. (2000).

Isomap

- Isomap: define neighbors and compute distances between neighbors.
- Geodesic distance approximated by shortest path through adjacency matrix.

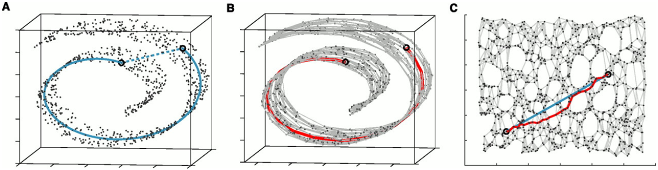


Figure: A: true geodesic distance. B: Approximate distance on graph. C: comparison of true and approximate distances. Image from Tenenbaum et al. (2000).

Isomap Neighborhood

- Compute nearest k neighbors for each point.
- Construct a graph linking data points through neighbors.

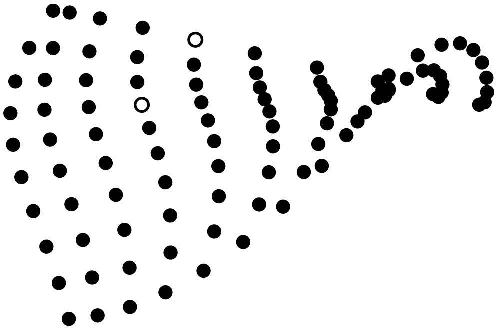


Figure: Distance on graph is a proxy for geodesic distance.

Isomap Neighborhood

- Compute nearest k neighbors for each point.
- Construct a graph linking data points through neighbors.

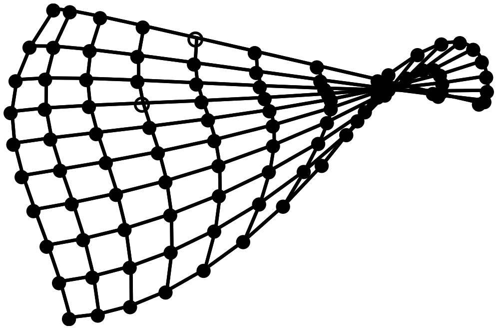


Figure: Distance on graph is a proxy for geodesic distance.

Isomap Neighborhood

- Compute nearest k neighbors for each point.
- Construct a graph linking data points through neighbors.

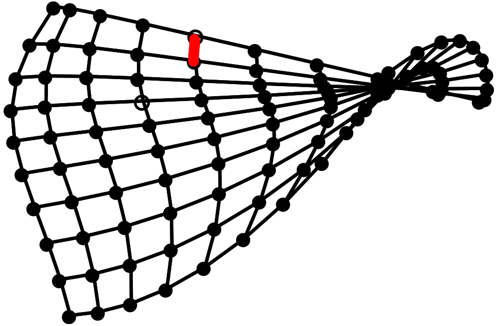


Figure: Distance on graph is a proxy for geodesic distance.

Isomap Neighborhood

- Compute nearest k neighbors for each point.
- Construct a graph linking data points through neighbors.

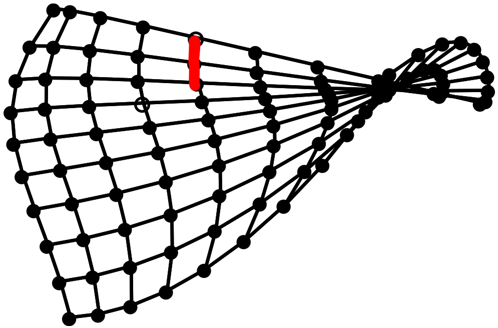


Figure: Distance on graph is a proxy for geodesic distance.

Isomap Neighborhood

- Compute nearest k neighbors for each point.
- Construct a graph linking data points through neighbors.

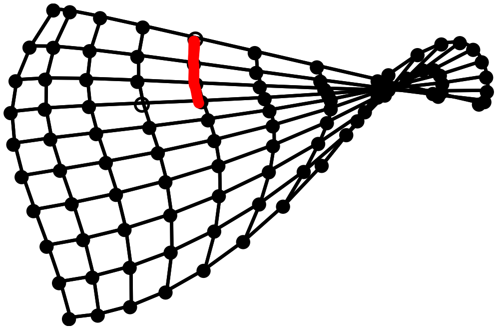


Figure: Distance on graph is a proxy for geodesic distance.

Isomap Neighborhood

- Compute nearest k neighbors for each point.
- Construct a graph linking data points through neighbors.

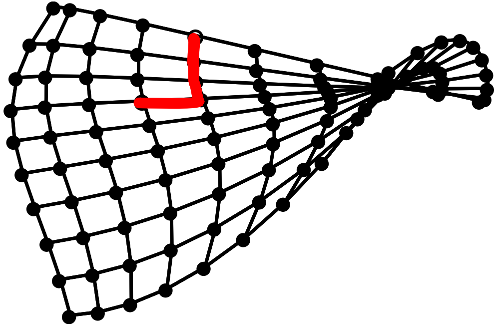


Figure: Distance on graph is a proxy for geodesic distance.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

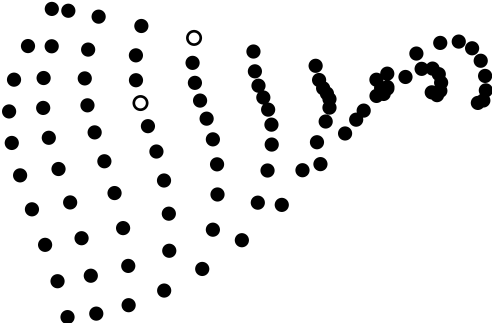


Figure: Quality of approximation depends on quality of graph.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

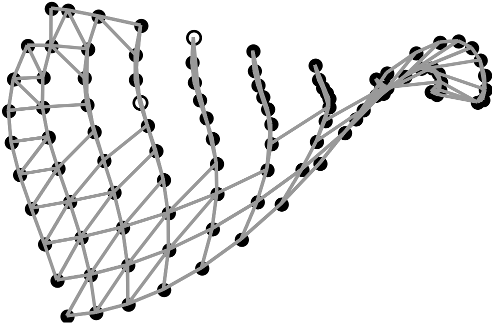


Figure: Quality of approximation depends on quality of graph.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

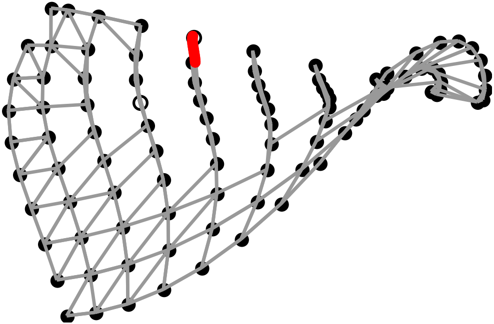


Figure: Quality of approximation depends on quality of graph.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

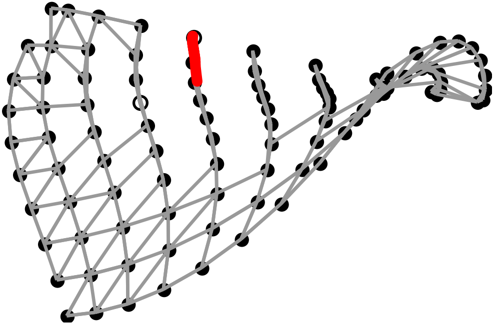


Figure: Quality of approximation depends on quality of graph.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

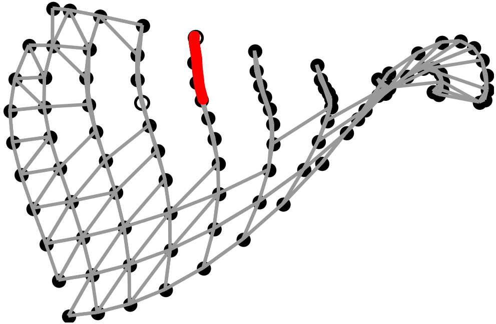


Figure: Quality of approximation depends on quality of graph.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

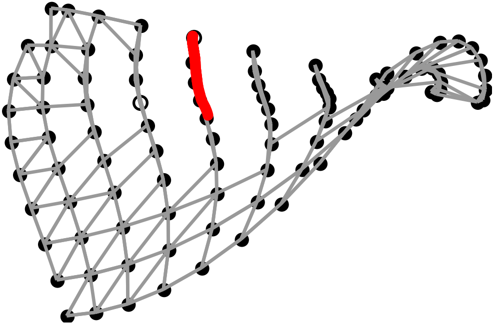


Figure: Quality of approximation depends on quality of graph.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

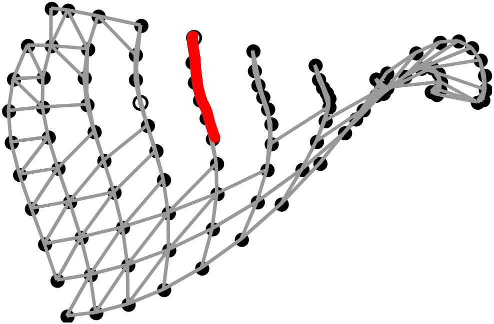


Figure: Quality of approximation depends on quality of graph.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

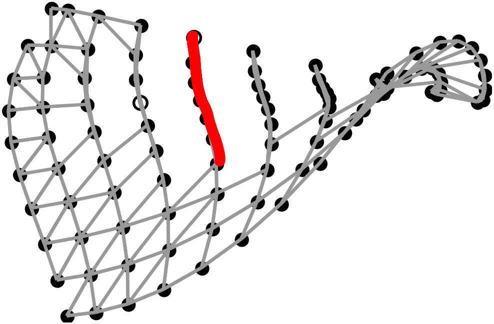


Figure: Quality of approximation depends on quality of graph.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

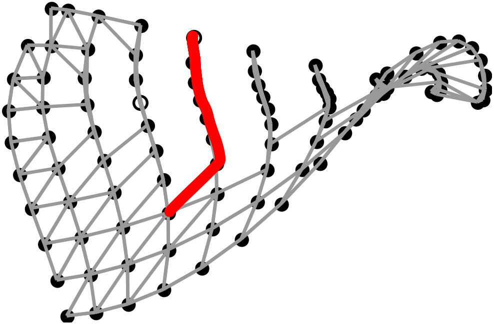


Figure: Quality of approximation depends on quality of graph.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

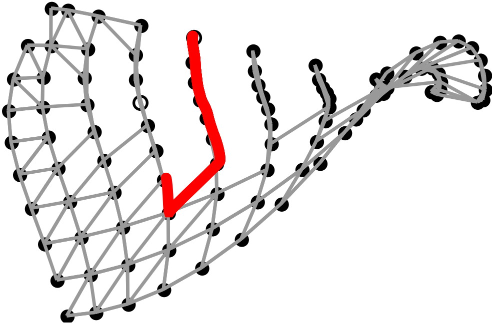


Figure: Quality of approximation depends on quality of graph.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

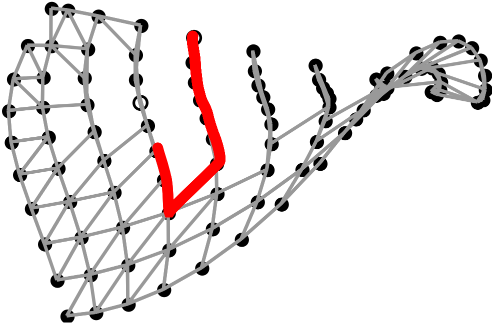


Figure: Quality of approximation depends on quality of graph.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

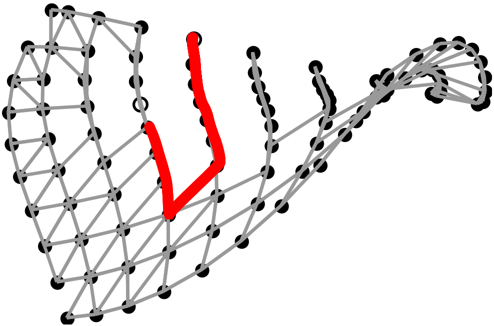


Figure: Quality of approximation depends on quality of graph.

Data Neighborhood

- Need to determine correct number of neighbors.
- Manifold distortions mean neighbors in latent space may not be neighbors in data space.

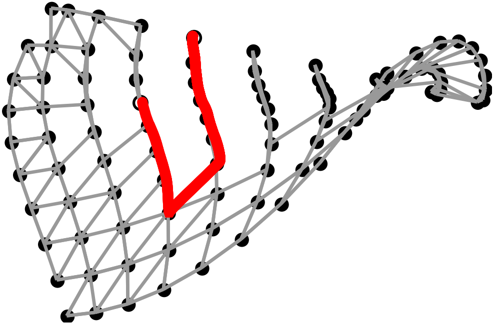


Figure: Quality of approximation depends on quality of graph.

Outline

Spectral Approaches

- Standard classical MDS gives a *linear* embedding in the Euclidean space implied by $\mathbf{D}$.

Spectral Approaches

- Standard classical MDS gives a *linear* embedding in the Euclidean space implied by $\mathbf{D}$.
- This implies a linear transformation between $\mathbf{X}$ and $\mathbf{Y}$ (if squared distances are computed directly in $\mathbf{Y}$).

Spectral Approaches

- Standard classical MDS gives a *linear* embedding in the Euclidean space implied by $\mathbf{D}$.
- This implies a linear transformation between $\mathbf{X}$ and $\mathbf{Y}$ (if squared distances are computed directly in $\mathbf{Y}$).
- Spectral approaches in machine learning give a *nonlinear* relationship between the data and the distances.

Spectral Approaches

- Standard classical MDS gives a *linear* embedding in the Euclidean space implied by $\mathbf{D}$.
- This implies a linear transformation between $\mathbf{X}$ and $\mathbf{Y}$ (if squared distances are computed directly in $\mathbf{Y}$).
- Spectral approaches in machine learning give a *nonlinear* relationship between the data and the distances.
- This is done by not computing $\mathbf{D}$ directly in the space of $\mathbf{Y}$.

Spectral Approaches

- Standard classical MDS gives a *linear* embedding in the Euclidean space implied by $\mathbf{D}$.
- This implies a linear transformation between $\mathbf{X}$ and $\mathbf{Y}$ (if squared distances are computed directly in $\mathbf{Y}$).
- Spectral approaches in machine learning give a *nonlinear* relationship between the data and the distances.
- This is done by not computing $\mathbf{D}$ directly in the space of $\mathbf{Y}$.
- This is very clear for kernel PCA, where $\mathbf{D}$ is computed in a feature space derived from $\mathbf{Y}$.

Kernel PCA

- Kernel PCA squared distance is defined through a kernel:

$$d_{i,j} = k(\mathbf{y}_{i,:}, \mathbf{y}_{i,:}) - 2k(\mathbf{y}_{i,:}, \mathbf{y}_{j,:}) - k(\mathbf{y}_{j,:}, \mathbf{y}_{j,:}) \qquad (1)$$

Kernel PCA

- Kernel PCA squared distance is defined through a kernel:

$$d_{i,j} = k(\mathbf{y}_{i,:}, \mathbf{y}_{i,:}) - 2k(\mathbf{y}_{i,:}, \mathbf{y}_{j,:}) - k(\mathbf{y}_{j,:}, \mathbf{y}_{j,:}) \qquad (1)$$

- $k(\cdot, \cdot)$ is a Mercer kernel (Ham et al., 2004).

Kernel PCA

- Kernel PCA squared distance is defined through a kernel:

$$d_{i,j} = k(\mathbf{y}_{i,:}, \mathbf{y}_{i,:}) - 2k(\mathbf{y}_{i,:}, \mathbf{y}_{j,:}) - k(\mathbf{y}_{j,:}, \mathbf{y}_{j,:}) \tag{1}$$

- $k(\cdot, \cdot)$ is a Mercer kernel (Ham et al., 2004).
- Kernel PCA (KPCA) recovers an $\mathbf{x}_{i,:}$ and a mapping from $\mathbf{Y}$ to $\mathbf{X}$ space.

Kernel PCA

- Kernel PCA squared distance is defined through a kernel:

$$d_{i,j} = k(\mathbf{y}_{i,:}, \mathbf{y}_{i,:}) - 2k(\mathbf{y}_{i,:}, \mathbf{y}_{j,:}) - k(\mathbf{y}_{j,:}, \mathbf{y}_{j,:}) \qquad (1)$$

- $k(\cdot, \cdot)$ is a Mercer kernel (Ham et al., 2004).
- Kernel PCA (KPCA) recovers an $\mathbf{x}_{i,:}$ and a mapping from $\mathbf{Y}$ to $\mathbf{X}$ space.
- The mapping is induced through the choice of the *Mercer kernel*.

Classical MDS and KPCA

- CMDS procedure performs eigenvalue problem on

$$\mathbf{B} = \mathbf{HKH}.$$

[1]Kernel PCA also has an interpretation as a particular form of *metric* multidimensional scaling, see Williams (2001) for details.

Classical MDS and KPCA

- CMDS procedure performs eigenvalue problem on

$$\mathbf{B} = \mathbf{HKH}.$$

- This matches the KPCA algorithm (Schölkopf et al., 1998)[1].

[1]Kernel PCA also has an interpretation as a particular form of *metric* multidimensional scaling, see Williams (2001) for details.

Classical MDS and KPCA

- CMDS procedure performs eigenvalue problem on

$$\mathbf{B} = \mathbf{H}\mathbf{K}\mathbf{H}.$$

- This matches the KPCA algorithm (Schölkopf et al., 1998)[1].
- **However**, for the commonly used exponentiated quadratic kernel,

$$k(y_{i,:}, y_{j,:}) = \exp(-\gamma \left\|\mathbf{y}_{i,:} - \mathbf{y}_{j,:}\right\|_2^2),$$

  KPCA actually *expands* the feature space (Weinberger et al., 2004).

---

[1]Kernel PCA also has an interpretation as a particular form of *metric* multidimensional scaling, see Williams (2001) for details.

Maximum Variance Unfolding

Learn a "Kernel" for Dimensionality Reduction

- In maximum variance unfolding (MVU) (Weinberger et al., 2004): learn a "kernel matrix" that will allow for dimensionality reduction.

Maximum Variance Unfolding

Learn a "Kernel" for Dimensionality Reduction

- In maximum variance unfolding (MVU) (Weinberger et al., 2004): learn a "kernel matrix" that will allow for dimensionality reduction.
- Preserve only *local* proximity relationships in the data.

Maximum Variance Unfolding

Learn a “Kernel” for Dimensionality Reduction

- In maximum variance unfolding (MVU) (Weinberger et al., 2004): learn a “kernel matrix” that will allow for dimensionality reduction.
- Preserve only *local* proximity relationships in the data.
  - Take a set of neighbors.

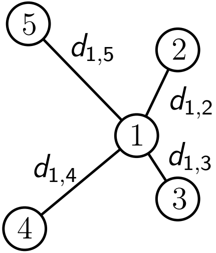
Maximum Variance Unfolding

Learn a "Kernel" for Dimensionality Reduction

- In maximum variance unfolding (MVU) (Weinberger et al., 2004): learn a "kernel matrix" that will allow for dimensionality reduction.
- Preserve only *local* proximity relationships in the data.
  - Take a set of neighbors.
  - Construct a kernel matrix where only distances between neighbors match data distances.

Maximum Variance Unfolding

- Optimize elements of K by maximizing[2] $\mathrm{tr}(\mathsf{K})$.



- Subject to squared distance constraints between neighbors

$$d_{i,j} = k_{i,i} - 2k_{i,j} + k_{j,j}$$

[2] The trace is the *total variance* of the data in feature space

Maximum Entropy Unfolding

Our Contribution

- Maximize *entropy* instead of variance (Jaynes, 1986): MEU (Lawrence, 2011, 2010).

Maximum Entropy Unfolding

Our Contribution

- Maximize *entropy* instead of variance (Jaynes, 1986): MEU (Lawrence, 2011, 2010).
- Entropy and variance are closely related.

Maximum Entropy Unfolding

Our Contribution

- Maximize *entropy* instead of variance (Jaynes, 1986): MEU (Lawrence, 2011, 2010).
- Entropy and variance are closely related.
- Maximum entropy leads to a *probabilistic model*.

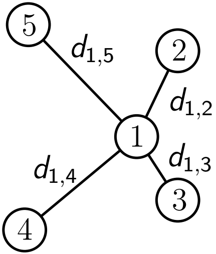
Maximum Entropy Unfolding

Our Contribution

- Maximize *entropy* instead of variance (Jaynes, 1986): MEU (Lawrence, 2011, 2010).
- Entropy and variance are closely related.
- Maximum entropy leads to a *probabilistic model*.
- Each spectral approach approximates MEU in some way.

Maximum Entropy Unfolding

- Find distribution with maximum entropy subject to constraints on *moments*.

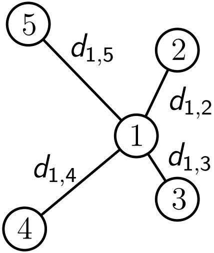


- MEU constraints are on expected distances between neighbors.

$$d_{i,j} = \left\langle \mathbf{y}_{i,:}^{\top}\mathbf{y}_{i,:} \right\rangle - 2 \left\langle \mathbf{y}_{i,:}^{\top}\mathbf{y}_{j,:} \right\rangle + \left\langle \mathbf{y}_{j,:}^{\top}\mathbf{y}_{j,:} \right\rangle$$

Maximum Entropy Unfolding

- Find distribution with maximum entropy subject to constraints on *moments*.



- MEU constraints are on expected distances between neighbors.

$$d_{i,j} = k_{i,i} - 2k_{i,j} + k_{j,j}$$

which can be written in terms of the covariance.

Gaussian Random Field

- The maximum entropy probability distribution is a *Gaussian random field*

$$p(\mathbf{Y}) = \prod_{j=1}^{p} \frac{1}{|\mathbf{K}|^{\frac{1}{2}} (2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}\mathbf{y}_{:,j}^{\top}\mathbf{K}^{-1}\mathbf{y}_{:,j}\right),$$

- Covariance matrix is

$$\mathbf{K} = (\mathbf{L} + \gamma\mathbf{I})^{-1}$$

.

- Where $\mathbf{L}$ is the *Laplacian* matrix associated with the neighborhood graph.
- Off diagonal elements of the Laplacian are Lagrange multipliers from moment constraints.
- On diagonal elements given by negative sum of off-diagonal ($\mathbf{L}\mathbf{1} = \mathbf{0}$).

Data Feature Independence

- The GRF specifying independence across data *features*.
- Most applications of Gaussian models are applied independently across data *points*.
  - Notable exceptions include Zhu et al. (2003); Lawrence (2004, 2005); Kemp and Tenenbaum (2008).
- Maximum likelihood in this model is equivalent maximizing entropy under distance constraints.

Blessing of Dimensionality

$$p(\mathbf{Y}) = \prod_{j=1}^{p} \frac{1}{|\mathbf{K}|^{\frac{1}{2}} (2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}\mathbf{y}_{:,j}^{\top}\mathbf{K}^{-1}\mathbf{y}_{:,j}\right),$$

Blessing of Dimensionality

$$p(\mathbf{Y}) = \prod_{j=1}^{p} \frac{1}{|\mathbf{K}|^{\frac{1}{2}} (2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}\mathbf{y}_{:,j}^{\top}\mathbf{K}^{-1}\mathbf{y}_{:,j}\right),$$

- Maximum likelihood is consistent: (see e.g. Wasserman, 2003, pg 126)

Blessing of Dimensionality

$$p(\mathbf{Y}) = \prod_{j=1}^{p} \frac{1}{|\mathbf{K}|^{\frac{1}{2}} (2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}\mathbf{y}_{:,j}^{\top}\mathbf{K}^{-1}\mathbf{y}_{:,j}\right),$$

- Maximum likelihood is consistent: (see e.g. Wasserman, 2003, pg 126)
  - As we increase data points parameters become better determined.

Blessing of Dimensionality

$$p(\mathbf{Y}) = \prod_{j=1}^{p} \frac{1}{\left|\mathbf{K}\right|^{\frac{1}{2}} (2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}\mathbf{y}_{:,j}^{\top}\mathbf{K}^{-1}\mathbf{y}_{:,j}\right),$$

- Maximum likelihood is consistent: (see e.g. Wasserman, 2003, pg 126)
  - As we increase data points parameters become better determined.
  - **Not** in this model.

Blessing of Dimensionality

$$p(\mathbf{Y}) = \prod_{j=1}^{p} \frac{1}{|\mathbf{K}|^{\frac{1}{2}} (2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}\mathbf{y}_{:,j}^{\top}\mathbf{K}^{-1}\mathbf{y}_{:,j}\right),$$

- Maximum likelihood is consistent: (see e.g. Wasserman, 2003, pg 126)
  - As we increase data points parameters become better determined.
  - **Not** in this model.
  - As we increase data features parameters become better determined.

Blessing of Dimensionality

$$p(\mathbf{Y}) = \prod_{j=1}^{p} \frac{1}{|\mathbf{K}|^{\frac{1}{2}} (2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}\mathbf{y}_{:,j}^{\top}\mathbf{K}^{-1}\mathbf{y}_{:,j}\right),$$

- Maximum likelihood is consistent: (see e.g. Wasserman, 2003, pg 126)
  - As we increase data points parameters become better determined.
  - **Not** in this model.
  - As we increase data features parameters become better determined.
- This turns the large p small n problem on its head.

Blessing of Dimensionality

$$p(\mathbf{Y}) = \prod_{j=1}^{p} \frac{1}{|\mathbf{K}|^{\frac{1}{2}} (2\pi)^{\frac{n}{2}}} \exp\left(-\frac{1}{2}\mathbf{y}_{:,j}^{\top}\mathbf{K}^{-1}\mathbf{y}_{:,j}\right),$$

- Maximum likelihood is consistent: (see e.g. Wasserman, 2003, pg 126)
  - As we increase data points parameters become better determined.
  - **Not** in this model.
  - As we increase data features parameters become better determined.
- This turns the large p small n problem on its head.
- There is a "Blessing of Dimensionality" in this model.

Locally Linear Embedding

- The Laplacian should be constrained positive definite.

Locally Linear Embedding

- The Laplacian should be constrained positive definite.
- This constraint can be imposed by factorizing it as

$$\mathbf{L} = \mathbf{M}\mathbf{M}^\top$$

Locally Linear Embedding

- The Laplacian should be constrained positive definite.
- This constraint can be imposed by factorizing it as

$$\mathbf{L} = \mathbf{M}\mathbf{M}^\top$$

- To ensure it is a Laplacian, we need to constrain $\mathbf{M}^\top\mathbf{1} = \mathbf{0}$ giving $\mathbf{L}\mathbf{1} = \mathbf{0}$.

Locally Linear Embedding

- The Laplacian should be constrained positive definite.
- This constraint can be imposed by factorizing it as

$$\mathbf{L} = \mathbf{M}\mathbf{M}^\top$$

- To ensure it is a Laplacian, we need to constrain $\mathbf{M}^\top\mathbf{1} = \mathbf{0}$ giving $\mathbf{L}\mathbf{1} = \mathbf{0}$.
  - i.e. $m_{i,i} = -\sum_{j\in\mathcal{N}(i)} m_{j,i}$

Locally Linear Embedding

- The Laplacian should be constrained positive definite.
- This constraint can be imposed by factorizing it as

$$\mathbf{L} = \mathbf{M}\mathbf{M}^\top$$

- To ensure it is a Laplacian, we need to constrain $\mathbf{M}^\top \mathbf{1} = \mathbf{0}$ giving $\mathbf{L}\mathbf{1} = \mathbf{0}$.
  - i.e. $m_{i,i} = -\sum_{j \in \mathcal{N}(i)} m_{j,i}$
  - Set $m_{j,i} = 0$ if $j \notin \mathcal{N}(i)$.

Locally Linear Embedding

- Locally linear embeddings (Roweis and Saul, 2000) are then a specific case of MEU where

Locally Linear Embedding

- Locally linear embeddings (Roweis and Saul, 2000) are then a specific case of MEU where
  1. The diagonal sums, $m_{i,i}$, are further constrained to unity.

Locally Linear Embedding

- Locally linear embeddings (Roweis and Saul, 2000) are then a specific case of MEU where
  1. The diagonal sums, $m_{i,i}$, are further constrained to unity.
  2. Model parameters found by maximizing *pseudolikelihood* of the data.

Point One

- For unit diagonals we have $\mathbf{M} = \mathbf{I} - \mathbf{W}$.

Point One

- For unit diagonals we have $\mathbf{M} = \mathbf{I} - \mathbf{W}$.
- Here the off diagonal sparsity pattern of $\mathbf{W}$ matches $\mathbf{M}$.

Point One

- For unit diagonals we have $\mathbf{M} = \mathbf{I} - \mathbf{W}$.
- Here the off diagonal sparsity pattern of $\mathbf{W}$ matches $\mathbf{M}$.
- Thus
$$(\mathbf{I} - \mathbf{W})^\top \mathbf{1} = \mathbf{0}.$$

Point One

- For unit diagonals we have $\mathbf{M} = \mathbf{I} - \mathbf{W}$.
- Here the off diagonal sparsity pattern of $\mathbf{W}$ matches $\mathbf{M}$.
- Thus

$$(\mathbf{I} - \mathbf{W})^\top \mathbf{1} = \mathbf{0}.$$

- LLE proscribes that the smallest eigenvectors of

$$(\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W})^\top = \mathbf{M}\mathbf{M}^\top = \mathbf{L}$$

(like Laplacian Eigenmaps).

Point One

- For unit diagonals we have $\mathbf{M} = \mathbf{I} - \mathbf{W}$.
- Here the off diagonal sparsity pattern of $\mathbf{W}$ matches $\mathbf{M}$.
- Thus
$$(\mathbf{I} - \mathbf{W})^\top \mathbf{1} = \mathbf{0}.$$
- LLE proscribes that the smallest eigenvectors of
$$(\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W})^\top = \mathbf{M}\mathbf{M}^\top = \mathbf{L}$$
(like Laplacian Eigenmaps).
- Equivalent to CMDS on the GRF described by $\mathbf{L}$.

Second Point

- Pseudolikelihood approximation (see e.g. Koller and Friedman, 2009, pg 970): product of the conditional densities:

$$p(\mathbf{Y}) \approx \prod_{i=1}^{n} p(\mathbf{y}_{i,:}|\mathbf{Y}_{\backslash i}),$$

$\mathbf{Y}_{\backslash i}$ represents data other than the ith point.

Second Point

- Pseudolikelihood approximation (see e.g. Koller and Friedman, 2009, pg 970): product of the conditional densities:

$$p(\mathbf{Y}) \approx \prod_{i=1}^{n} p(\mathbf{y}_{i,:}|\mathbf{Y}_{\backslash i}),$$

  $\mathbf{Y}_{\backslash i}$ represents data other than the ith point.
- True likelihood is proportional to this but requires renormalization.

Second Point

- Pseudolikelihood approximation (see e.g. Koller and Friedman, 2009, pg 970): product of the conditional densities:

$$p(\mathbf{Y}) \approx \prod_{i=1}^{n} p(\mathbf{y}_{i,:}|\mathbf{Y}_{\backslash i}),$$

  $\mathbf{Y}_{\backslash i}$ represents data other than the ith point.
- True likelihood is proportional to this but requires renormalization.
- In pseudolikelihood normalization is ignored.

Conditionals

- Factors in the GRF are the conditionals,

$$p(\mathbf{y}_{i,:}|\mathbf{Y}_{\backslash i}) = \left(\frac{m_{i,i}^2}{2\pi}\right)^{\frac{p}{2}} \exp\left(-\frac{m_{i,i}^2}{2}\left\|\mathbf{y}_{i,:} - \sum_{j\in\mathcal{N}(i)} \frac{w_{j,i}}{m_{i,i}}\mathbf{y}_{j,:}\right\|_2^2\right).$$

Conditionals

- Factors in the GRF are the conditionals,

$$p(\mathbf{y}_{i,:}|\mathbf{Y}_{\backslash i}) = \left(\frac{m_{i,i}^2}{2\pi}\right)^{\frac{p}{2}} \exp\left(-\frac{m_{i,i}^2}{2}\left\|\mathbf{y}_{i,:} - \sum_{j\in\mathcal{N}(i)}\frac{w_{j,i}}{m_{i,i}}\mathbf{y}_{j,:}\right\|_2^2\right).$$

- Maximizing each conditional is equivalent to optimizing LLE objective.

Conditionals

- Factors in the GRF are the conditionals,

$$p(\mathbf{y}_{i,:}|\mathbf{Y}_{\backslash i}) = \left(\frac{m_{i,i}^2}{2\pi}\right)^{\frac{p}{2}} \exp\left(-\frac{m_{i,i}^2}{2}\left\|\mathbf{y}_{i,:} - \sum_{j\in\mathcal{N}(i)} \frac{w_{j,i}}{m_{i,i}}\mathbf{y}_{j,:}\right\|_2^2\right).$$

- Maximizing each conditional is equivalent to optimizing LLE objective.
- Constraint that LLE weights sum to one arises naturally because $w_{j,i}/m_{i,i}$ and $m_{i,i} = \sum_{j\in\mathcal{N}(i)} w_{j,i}$.

Conditionals

- Factors in the GRF are the conditionals,

$$p(\mathbf{y}_{i,:}|\mathbf{Y}_{\backslash i}) = \left(\frac{m_{i,i}^2}{2\pi}\right)^{\frac{p}{2}} \exp\left(-\frac{m_{i,i}^2}{2}\left\|\mathbf{y}_{i,:} - \sum_{j\in\mathcal{N}(i)} \frac{w_{j,i}}{m_{i,i}}\mathbf{y}_{j,:}\right\|_2^2\right).$$

- Maximizing each conditional is equivalent to optimizing LLE objective.
- Constraint that LLE weights sum to one arises naturally because $w_{j,i}/m_{i,i}$ and $m_{i,i} = \sum_{j\in\mathcal{N}(i)} w_{j,i}$.
- In LLE a *further* constraint is imposed $m_{i,i} = 1$.

LLE Approximates MEU

- LLE is an approximation to maximum likelihood.

LLE Approximates MEU

- LLE is an approximation to maximum likelihood.
- Laplacian has factorized form.

LLE Approximates MEU

- LLE is an approximation to maximum likelihood.
- Laplacian has factorized form.
- Pseudolikelihood also allows for relatively quick parameter estimation.

LLE Approximates MEU

- LLE is an approximation to maximum likelihood.
- Laplacian has factorized form.
- Pseudolikelihood also allows for relatively quick parameter estimation.
  - ignoring the partition function removes the need to invert to recover the covariance matrix.

LLE Approximates MEU

- LLE is an approximation to maximum likelihood.
- Laplacian has factorized form.
- Pseudolikelihood also allows for relatively quick parameter estimation.
  - ignoring the partition function removes the need to invert to recover the covariance matrix.
  - LLE can be applied to larger data sets than MEU or MVU.

Note: The sparsity pattern in the Laplacian for LLE will not match that used in the Laplacian for the other algorithms due to the factorized representation.

LLE and PCA

- LLE is motivated by considering local linear embeddings of the data.

LLE and PCA

- LLE is motivated by considering local linear embeddings of the data.
- Interestingly, as we increase the neighborhood size to $K = n - 1$ we do not recover PCA.

LLE and PCA

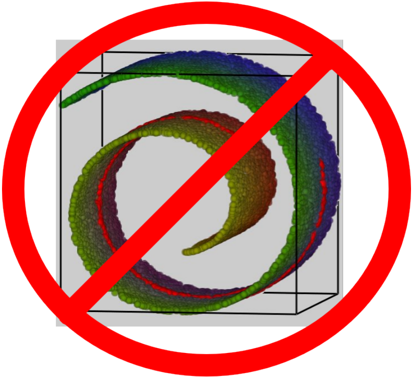
- LLE is motivated by considering local linear embeddings of the data.
- Interestingly, as we increase the neighborhood size to $K = n - 1$ we do not recover PCA.
- But PCA is the "optimal" linear embedding!!

LLE and PCA

- LLE is motivated by considering local linear embeddings of the data.
- Interestingly, as we increase the neighborhood size to $K = n - 1$ we do not recover PCA.
- But PCA is the "optimal" linear embedding!!
- LLE is optimizing a pseudolikelihood: in contrast the MEU algorithm, which LLE approximates, does recover PCA when $K = n - 1$.

Campaign for Real Data

Say NO to the Swiss Roll



Simple Experiments

- Simple motion capture data example.
- Changing incline of run of human captured with 34 markers (102 dimensions).
- 55 frames in the data.
- Follow the suggestion of Harmeling. (Harmeling, 2007) and use the GPLVM likelihood (Lawrence, 2005) for embedding quality.

Simple Experiments

- Simple motion capture data example.
- Changing incline of run of human captured with 34 markers (102 dimensions).
- 55 frames in the data.
- Follow the suggestion of Harmeling. (Harmeling, 2007) and use the GPLVM likelihood (Lawrence, 2005) for embedding quality.
- The higher the likelihood the better the embedding.

Motion Capture Data

- Data consists of a 3-dimensional point cloud of the location of 34 points from a subject performing a run.

Motion Capture Data

- Data consists of a 3-dimensional point cloud of the location of 34 points from a subject performing a run.
- 102 dimensional data set containing 55 frames of motion capture.

Motion Capture Data

- Data consists of a 3-dimensional point cloud of the location of 34 points from a subject performing a run.
- 102 dimensional data set containing 55 frames of motion capture.
- Subject begins the motion from stationary and takes approximately three strides of run.

Motion Capture Data

- Data consists of a 3-dimensional point cloud of the location of 34 points from a subject performing a run.
- 102 dimensional data set containing 55 frames of motion capture.
- Subject begins the motion from stationary and takes approximately three strides of run.
- Should see this structure in the visualization: a starting position followed by a series of loops.

Motion Capture Data

- Data consists of a 3-dimensional point cloud of the location of 34 points from a subject performing a run.
- 102 dimensional data set containing 55 frames of motion capture.
- Subject begins the motion from stationary and takes approximately three strides of run.
- Should see this structure in the visualization: a starting position followed by a series of loops.
- Data was made available by Ohio State University.

Motion Capture Data

- Data consists of a 3-dimensional point cloud of the location of 34 points from a subject performing a run.
- 102 dimensional data set containing 55 frames of motion capture.
- Subject begins the motion from stationary and takes approximately three strides of run.
- Should see this structure in the visualization: a starting position followed by a series of loops.
- Data was made available by Ohio State University.
- The two dominant eigenvectors are visualized in following figures.

Stick Man Data

- Visualize data.

PCA on Stick Man

- First two principal components of stick man data.

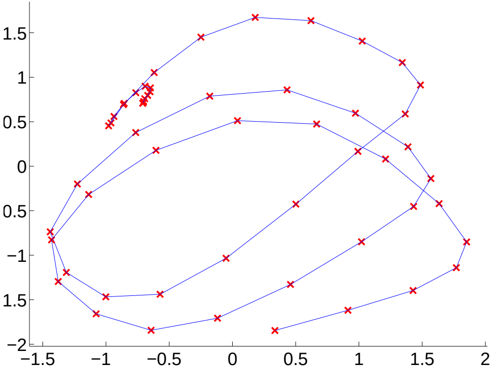


Figure: Stick man data projected onto their first two principal components. `demStickPpca1`.

Laplacian Eigenmaps and LLE

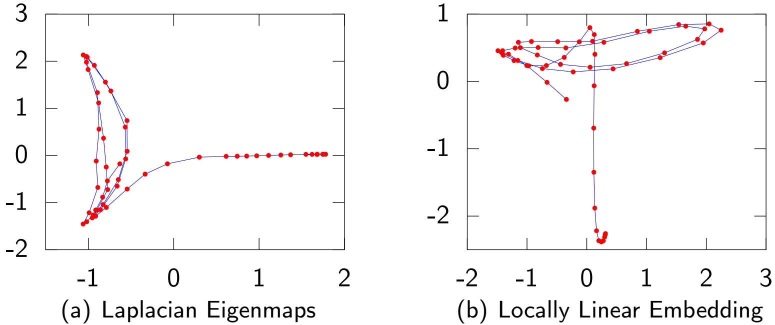
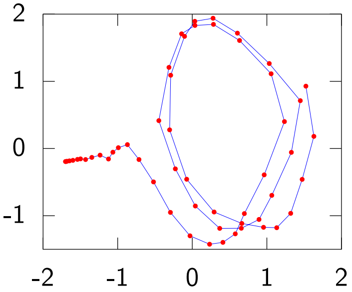
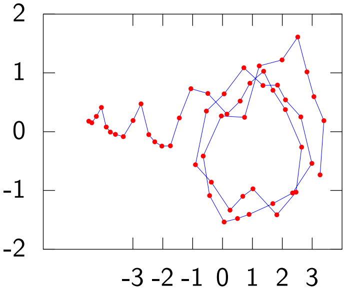


Figure: Models capture either the cyclic structure or the structure associated with the start of the run or both parts.

Isomap and MVU



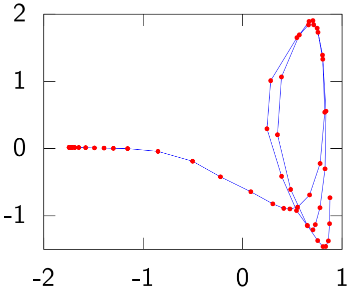
(a) Isomap



(b) MVU

Figure: Models capture either the cyclic structure or the structure associated with the start of the run or both parts.

MEU



(a) MEU

Figure: Models capture either the cyclic structure or the structure associated with the start of the run or both parts.

Motion Capture: Model Scores

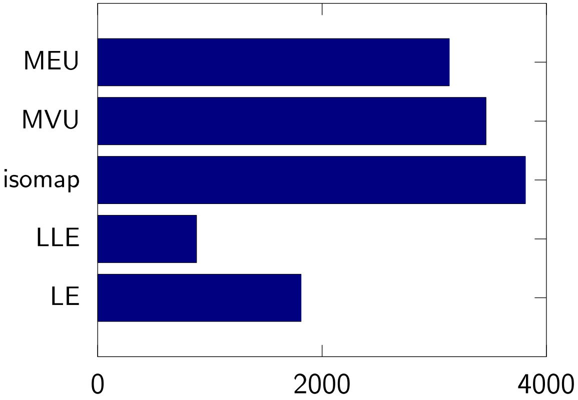


Figure: Model score for the different spectral approaches.

Outline

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

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

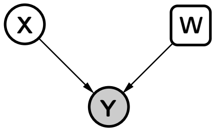
where

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

Linear Latent Variable Model

Probabilistic PCA

- Define *linear-Gaussian relationship* between latent variables and data.

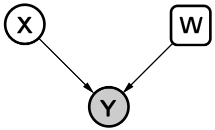


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}\mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

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

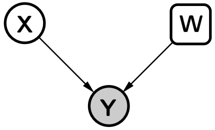


$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model

Probabilistic PCA

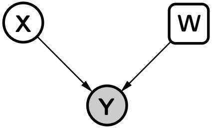
- Define *linear-Gaussian relationship* between latent variables and data.
- **Standard** Latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.



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

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

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*.

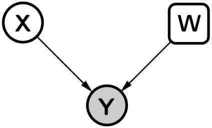
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

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

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

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)



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

Linear Latent Variable Model II

Probabilistic PCA Max. Likelihood Soln (Tipping and Bishop, 1999)

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

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

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

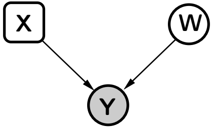
$$\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{R}^\top, \quad \mathbf{L} = \left(\mathbf{\Lambda}_q - \sigma^2\mathbf{I}\right)^{\frac{1}{2}}$$

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.

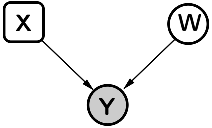


$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

Linear Latent Variable Model III

Dual Probabilistic PCA

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

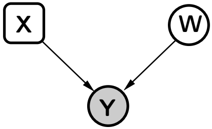


$$p\left(\mathbf{Y}|\mathbf{X},\mathbf{W}\right)=\prod_{i=1}^{n}\mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:},\sigma^{2}\mathbf{I}\right)$$

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*, $\mathbf{W}$.



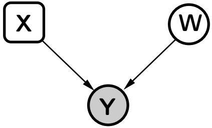
$$p(\mathbf{Y}|\mathbf{X},\mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

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

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*.



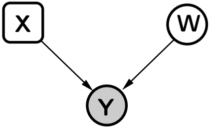
$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_{i,:}|\mathbf{W}\mathbf{x}_{i,:}, \sigma^2\mathbf{I}\right)$$

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

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

Linear Latent Variable Model IV

Dual Probabilistic PCA Max. Likelihood Soln (Lawrence, 2004)



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

Linear Latent Variable Model IV

Dual Probabilistic PCA Max. Likelihood Soln (Lawrence, 2004)

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

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

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 and Bishop, 1999)

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

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

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.

Equivalence of Formulations

The Eigenvalue Problems are equivalent

- Solution for Probabilistic PCA (solves for the mapping)

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

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

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

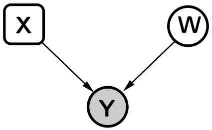
- Equivalence is from

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

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*.



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

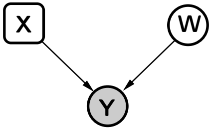
$$p(\mathbf{W}) = \prod_{i=1}^{p} \mathcal{N}\left(\mathbf{w}_{i,:}|\mathbf{0},\mathbf{I}\right)$$

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

Non-Linear Latent Variable Model

Dual Probabilistic PCA

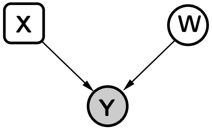
- Inspection of the marginal likelihood shows ...



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

Dual Probabilistic PCA

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



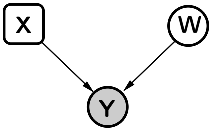
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

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

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'.



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

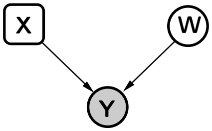
$$\mathbf{K} = \mathbf{X}\mathbf{X}^{\top} + \sigma^2\mathbf{I}$$

This is a product of Gaussian processes with linear kernels.

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).



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

$$\mathbf{K} = ?$$

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

RBF Kernel

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

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

- 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.

Applications

Style Based Inverse Kinematics

- Facilitating animation through modeling human motion with the GP-LVM (Grochow et al., 2004)

Tracking

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

Stick Man

Generalization with less Data than Dimensions

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

Stick Man II

`demStick1`

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

Stick Man II

`demStick1`

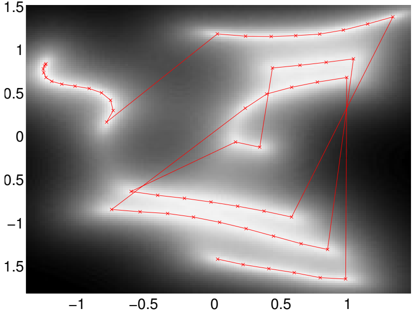


Figure: The latent space for the stick man motion capture data.

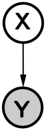
Selecting Data Dimensionality

- GP-LVM Provides probabilistic non-linear dimensionality reduction.
- How to select the dimensionality?
- Bayesian approach to model selection (Titsias and Lawrence, 2010).

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.

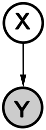


$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.
- Apply standard latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.

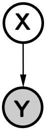


$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0},\mathbf{K}\right)$$

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.
- Apply standard latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.



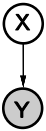
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$$p(\mathbf{X}) = \prod_{j=1}^{q} \mathcal{N}\left(\mathbf{x}_{:,j}|\mathbf{0}, \alpha_i^{-2}\mathbf{I}\right)$$

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.
- Apply standard latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.
  - Unfortunately integration is intractable. Use variational approximations (Titsias and Lawrence, 2010).



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

$$p(\mathbf{X}) = \prod_{j=1}^{q} \mathcal{N}\left(\mathbf{x}_{:,j}|\mathbf{0}, \alpha_i^{-2}\mathbf{I}\right)$$

$$p(\mathbf{Y}|\alpha) = ??$$

Learning Dimensionality: Automatic Relevance Determination

- Precision parameters, $\{\alpha_i\}_{i=1}^q$, softly switch off latent dimensions.

$$p(\mathbf{X}) = \prod_{j=1}^{q} \mathcal{N}\left(\mathbf{x}_{:,j}|\mathbf{0}, \alpha_i^{-2}\mathbf{I}\right)$$

- Equivalently, scale columns of $\mathbf{X}$ in the covariance function

$$k(\mathbf{x}_{i,:}, \mathbf{x}_{j,:}) = \exp\left(-\frac{1}{2}(\mathbf{x}_{:,i} - \mathbf{x}_{:,j})^\top \mathbf{A}^{-1}(\mathbf{x}_{:,i} - \mathbf{x}_{:,j})\right)$$

$\mathbf{A}$ is diagonal with elements α_i^2. Now keep prior spherical

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

- Covariance functions of this type are known as ARD (see e.g. Neal, 1996; MacKay, 2003; Rasmussen and Williams, 2006).

Summary

- Spectral approaches to dimensionality reduction have an underlying interpretation as a Gaussian random field.
- The probabilistic model is consistent as $p \to \infty$, not $n \to \infty$.
- Spectral approaches have the neighborhood pre-specified.
- The GP-LVM is also a Gaussian model of data with a generative interpretation.
- In the GP-LVM the "neighborhood" is learnt.
- The Bayesian GP-LVM allows the number of latent dimensions to be determined.

References I

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. [DOI].

R. Greiner and D. Schuurmans, editors. *Proceedings of the International Conference in Machine Learning*, volume 21, 2004. Omnipress.

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.

J. Ham, D. D. Lee, S. Mika, and B. Schölkopf. A kernel view of dimensionality reduction of manifolds. In Greiner and Schuurmans (2004). [PDF].

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

E. T. Jaynes. Bayesian methods: General background. In J. H. Justice, editor, *Maximum Entropy and Bayesian Methods in Applied Statistics*, pages 1–25. Cambridge University Press, 1986.

C. Kemp and J. B. Tenenbaum. The discovery of structural form. *Proc. Natl. Acad. Sci. USA*, 105(31), 2008.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009. [Google Books] .

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. A unifying probabilistic perspective for spectral dimensionality reduction. Technical report, University of Sheffield, [PDF].

N. D. Lawrence. Spectral dimensionality reduction via maximum entropy. In G. Gordon and D. Dunson, editors, *Proceedings of the Fourteenth International Workshop on Artificial Intelligence and Statistics*, volume 15, Fort Lauderdale, FL, USA, 11-13 April 2011. JMLR W&CP 15. [PDF]. Notable Paper.

References II

D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge, U.K., 2003. [Google Books] .

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, 1996. Lecture Notes in Statistics 118.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. [Google Books] .

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. [DOI].

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. [DOI].

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. [DOI].

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

M. K. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. In Y. W. Teh and D. M. Titterington, editors, *Proceedings of the Thirteenth International Workshop on Artificial Intelligence and Statistics*, volume 9, pages 844–851, Chia Laguna Resort, Sardinia, Italy, 13-16 May 2010. JMLR W&CP 9. [PDF].

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

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.

L. A. Wasserman. *All of Statistics*. Springer-Verlag, New York, 2003. [Google Books] .

K. Q. Weinberger, F. Sha, and L. K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In Greiner and Schuurmans (2004), pages 839–846.

C. K. I. Williams. On a connection between kernel PCA and metric multidimensional scaling. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 675–681, Cambridge, MA, 2001. MIT Press.

X. Zhu, J. Lafferty, and Z. Ghahramani. Semi-supervised learning: From Gaussian fields to Gaussian processes. Technical Report CMU-CS-03-175, Carnegie Mellon University, [PDF].