Dimensionality Reduction Methods in Predictive Modelling [1] [2]

Stela Makri
S.Makri@warwick.ac.uk
Warwick Centre of Predictive Modelling (WCPM)
The University of Warwick

13 October 2015

WCPM

http://www2.warwick.ac.uk/wcpm/

[1] Based on Stela Makri, "Dimensionality Reduction Methods", MSc Dissertation, Centre for Scientific Computing, University of Warwick

[2] Warwick Centre of Predictive Modelling (WCPM) Seminar Series, University of Warwick

Outline

Motivation

Advancing technology

$\downarrow$

Vast amounts of data being generated each day

$\downarrow$

Need for analysing these data

Background

Data depend on a number of variables D. We assume that each variable takes values in the real space $\mathbb{R}$. We call the union of these sets, the *data space*.

Each variable defines a *dimension* of the data space.

Data generated in the physical world in general depend on a large number of such variables.

Background

Data depend on a number of variables D. We assume that each variable takes values in the real space $\mathbb{R}$. We call the union of these sets, the *data space*.

Each variable defines a *dimension* of the data space.

Data generated in the physical world in general depend on a large number of such variables.

Example: weather forecast measurements
time instance, spatial location, temperature reading, wind speed and direction, humidity rate, atmospheric pressure, UV index, etc

Background

Data depend on a number of variables D. We assume that each variable takes values in the real space $\mathbb{R}$. We call the union of these sets, the *data space*.

Each variable defines a *dimension* of the data space.

Data generated in the physical world in general depend on a large number of such variables.

Example: weather forecast measurements
time instance, spatial location, temperature reading, wind speed and direction, humidity rate, atmospheric pressure, UV index, etc

$\Rightarrow$ data live in a space of ≥ 8 dimensions.

Background

Data depend on a number of variables D. We assume that each variable takes values in the real space $\mathbb{R}$. We call the union of these sets, the *data space*.

Each variable defines a *dimension* of the data space.

Data generated in the physical world in general depend on a large number of such variables.

Example: weather forecast measurements
time instance, spatial location, temperature reading, wind speed and direction, humidity rate, atmospheric pressure, UV index, etc

$\Rightarrow$ data live in a space of ≥ 8 dimensions.

Impossible for the human brain to process raw data and make observations about patterns.

Background

Variables generating the data are strongly dependent on one another.

Background

Variables generating the data are strongly dependent on one another. $\Rightarrow$ the data reside on a subspace of the data space, which has smaller dimensionality.

Background

Variables generating the data are strongly dependent on one another. $\Rightarrow$ the data reside on a subspace of the data space, which has smaller dimensionality.

Example: Handwritten digits
Consider one of the off-line digits, represented by a 64×64 pixel grey-level image (Fig. 1). Embedding this in a larger image of size 100×100 by padding with zero pixels. $\Rightarrow$ each image datapoint lies in a $10,000$ dimensional space.

Background

Variables generating the data are strongly dependent on one another. $\Rightarrow$ the data reside on a subspace of the data space, which has smaller dimensionality.

Example: Handwritten digits
Consider one of the off-line digits, represented by a 64×64 pixel grey-level image (Fig. 1). Embedding this in a larger image of size 100×100 by padding with zero pixels. $\Rightarrow$ each image datapoint lies in a $10,000$ dimensional space.
Create multiple copies of the same digit 3, varying the location and orientation of the digit at random in each copy.

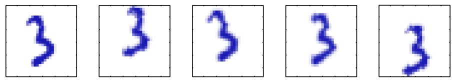


Figure: Samples of the off digit 3 obtained by rotation and translation (obtained from the Mnist Dataset). The *intrinsic dimensionality of the data manifold* is 3.

Background

Variables generating the data are strongly dependent on one another. $\Rightarrow$ the data reside on a subspace of the data space, which has smaller dimensionality.

Example: Handwritten digits
Consider one of the off-line digits, represented by a 64×64 pixel grey-level image (Fig. 1). Embedding this in a larger image of size 100×100 by padding with zero pixels. $\Rightarrow$ each image datapoint lies in a $10,000$ dimensional space.
Create multiple copies of the same digit 3, varying the location and orientation of the digit at random in each copy.

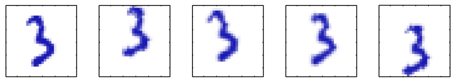


Figure: Samples of the off digit 3 obtained by rotation and translation (obtained from the Mnist Dataset). The *intrinsic dimensionality of the data manifold* is 3.

$\Rightarrow$ there are only three DOF of variability: (a) Vertical displacement, (b) Horizontal displacement and (c) Rotation. (intrinsic dimensionality of the data set is three).

Notation

We will restrict our attention to datasets $\mathcal{X}$ taking values in $\mathbb{R}^D$ and we will represent data points by D dimensional column vectors. Further we will distinguish between linear and non-linear models.

- *Linear models* assume a linear structure for the data. That is, the data reside on some $Q-$dimensional hyperplane where $Q < D$.
- This assumption is relaxed in the case of *non-linear models*.

Linear Dimensionality Reduction Models

Rely on simple intuitive models and therefore provide

- fast algorithms

Linear Dimensionality Reduction Models

Rely on simple intuitive models and therefore provide

- fast algorithms
- clear interpretation of the reduced space

Rely on simple intuitive models and therefore provide

- fast algorithms
- clear interpretation of the reduced space

Further, linear models handle well noise in data

Principal Component Analysis

Consider a data set $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^D$ and a linear subspace $\mathcal{U}$ of $\mathbb{R}^D$ of dimensionality $Q \leq D$.

Assumptions

Fix Q.

We assume that $\exists \mathbf{b} \in \mathbb{R}^D \setminus \mathcal{U}$ such that $\forall n$, we can approximate $\mathbf{x}_n$ by an $\tilde{\mathbf{x}}_n$ of the form

$$\boxed{\tilde{\mathbf{x}}_n = \mathbf{b} + \mathbf{z}_n,} \tag{1}$$

where $\mathbf{z}_n \in \mathcal{U}$

It is convenient to define a basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_Q\}$ be a basis for $\mathcal{U} \subset \mathbb{R}^D$ and extend this to a basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_Q, \mathbf{u}_{Q+1}, \ldots, \mathbf{u}_D\}$ for $\mathbb{R}^D$. Then express

$$\mathbf{b} = \sum_{j=Q+1}^{D} b_j \mathbf{u}_j, \quad \mathbf{z}_n = \sum_{q=1}^{Q} z_{nq} \mathbf{u}_q \quad \text{for } z_{nq}, b_q \text{ reals.}$$

- **Pearson's approach**:
  Minimise the average projection cost

$$J = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2. \tag{2}$$

  with respect to $(\mathbf{u}_q, z_{nq}$ and $b_j)$.
- **Hotelling's approach**:
  Write $z_{nq} = (\mathbf{u}^\intercal \mathbf{x}_n)$ and maximize the variance of the *projected data* $\mathbf{z}_n$.

$$\sigma^2_{\mathbf{C}} = \mathrm{tr}\left(\mathbf{C}\right), \tag{3}$$

  where $\mathbf{C}$ is the covariance matrix of the projected data.

$$\mathbf{C} = \frac{1}{N} \sum_{n} \left(\mathbf{z}_n - \bar{\mathbf{z}}\right) \left(\mathbf{z}_n - \bar{\mathbf{z}}\right)^{\mathrm{T}} = \sum_{p=1}^{Q} \sum_{q=1}^{Q} \left(\mathbf{u}_p^{\mathrm{T}} \mathbf{S} \mathbf{u}_q\right) \mathbf{u}_p \mathbf{u}_q^{\mathrm{T}}. \tag{4}$$

Pearson (1901)
Hotelling (1933)

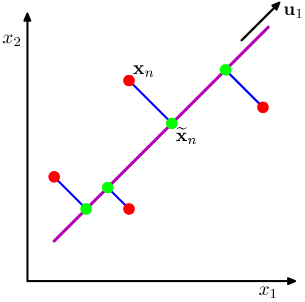


Figure: PCA seeks a space of lower dimensionality (magenta line) such that: (1) the orthogonal projection of the data points (red dots) onto this subspace maximizes the variance of the projected points (green dots). (2) the sum-of-squares of the projection errors (blue lines) is minimised.

Let $\lambda_1, \ldots, \lambda_D$ be the eigenvalues of the data covariance $\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}\left(\mathbf{x}_n - \bar{\mathbf{x}}\right)\left(\mathbf{x}_n - \bar{\mathbf{x}}\right)^{\mathrm{T}}$ ordered in decreasing values. Then the average projection cost J and the data variance σ^2 are extremised if we choose

> $\{\mathbf{u}_q\}_{q=1}^{Q}$ to be the eigenvectors of $\mathbf{S}$ associated to $\lambda_1, \ldots, \lambda_Q$,

In particular, we can approximate each $\mathbf{x}_n$ by

$$\tilde{\mathbf{x}}_n = \sum_{q=1}^{Q}\left\{\left(\mathbf{x}_n - \bar{\mathbf{x}}\right)^T \mathbf{u}_q\right\}\mathbf{u}_q + \bar{\mathbf{x}}.$$

Bishop (2006)

Algorithm 1: Principal Component Analysis (PCA)

begin

1 centralise data by removing the data mean $\bar{\mathbf{x}}$ from each datapoint $\hat{\mathbf{X}} = \mathbf{X} - (1\,1\,\ldots,\,1)^{\mathrm{T}}\,\bar{\mathbf{x}}^{\mathrm{T}}$

2 evaluate the data covariance $\mathbf{S} = \frac{1}{N}\hat{\mathbf{X}}^{\mathrm{T}}\hat{\mathbf{X}}$

3 find all eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_D$ and eigenvalues $\lambda_1, \ldots, \lambda_D$ of $\mathbf{S}$;

4 sort the eigenvalues in decreasing order of magnitude and reorder the eigenvectors accordingly;

5 $\boldsymbol{\lambda} = (\lambda_1, \ldots, , \lambda_Q)^T$; $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_Q)$;

6 compute the reconstruction of the input data points $\mathbf{Z} = \mathbf{U}^{\mathrm{T}}\hat{\mathbf{X}}^{\mathrm{T}}$;

7 $\tilde{\mathbf{X}} = (1\,1\,\ldots,\,1)^{\mathrm{T}}\,\bar{\mathbf{x}} + (\mathbf{U}\mathbf{Z})^{\mathrm{T}}$;

8 Return $\tilde{\mathbf{X}}$;

end

Probabilistic Principal Component Analysis

Introduce the latent variable $\mathbf{z} \in \mathbb{R}^Q$ (principal-component subspace).
Assume a Gaussian prior distribution $p(\mathbf{z})$,

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}). \tag{5}$$

Seek to relate a D-dimensional observation vector $\mathbf{x}$ to the corresponding Q-dimensional Gaussian latent variable $\mathbf{z}$ by a linear transformation $\mathbf{W}$:

Include a Gaussian noise variable $\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{6}$$

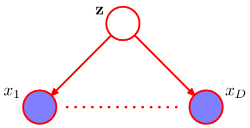


Figure: Probabilistic PCA as a Naive Bayes model - conditioned on $\mathbf{z}$, the components of the observed vector $\mathbf{x} = (x_1, \ldots, x_D)^T$ are assumed to be independent

Tipping, Bishop (1999b)
Bishop (2006)

Induces a Gaussian distribution

$$\boxed{\mathbf{x}|\mathbf{z} \sim \mathcal{N}\left(\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2 \mathbf{I}\right)} \tag{7}$$

To compute the likelihood function, we need an expression for the marginal distribution $p(\mathbf{x})$ of the observed variable.

$$\Rightarrow \boxed{\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I}\right)}, \tag{8}$$

where we've written $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I}$, for the covariance

It is worth noting that there is a whole family of $\mathbf{W}$'s, differing by a rotation of the latent space coordinates, that lead to the same $p(\mathbf{x})$.
For an arbitrary rotation $\mathbf{R}$, set $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$. Then

$$\tilde{\mathbf{W}}\tilde{\mathbf{W}}^{\mathrm{T}} = \mathbf{W}\mathbf{R}\mathbf{R}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}} = \mathbf{W}\mathbf{W}^{\mathrm{T}},$$

and $p(\mathbf{x})$ remains unchanged.

Tipping, Bishop (1999b)

Probabilistic Principal Component Analysis

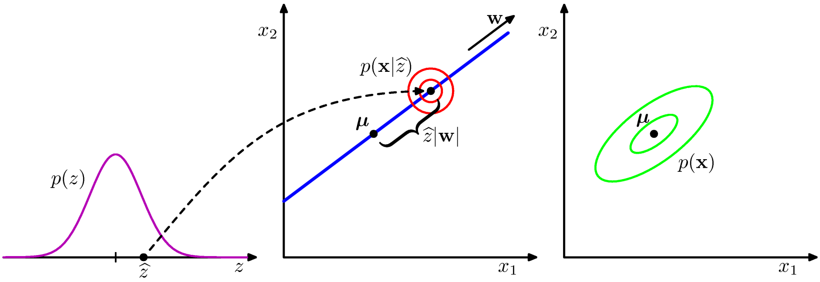


Figure: Mapping from the latent space to the data space. We assume here 2D data and 1D latent space. An observed $\mathbf{x}$ is generated by drawing a value $\hat{z}$ from $p(z) = \mathcal{N}(z|0,1)$ and then a value for $\mathbf{x}$ from an isotropic Gaussian distribution (red circles) having mean $\mathbf{w}\hat{z} + \boldsymbol{\mu}$ and covariance $\sigma^2\mathbf{I}_D$ (i.e. from $p(\mathbf{x}|z) = \mathcal{N}(\mathbf{x}|\mathbf{w}z + \boldsymbol{\mu}, \sigma^2)$. The green ellipses are the density contours of $p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}, \mathbf{w}\mathbf{w}^{\mathrm{T}} + \sigma^2\right)$.

Bishop (2006)

The log-likelihood of $\mathbf{x}$ is:

$$\begin{aligned}\mathcal{L}\left(\mathbf{x};\boldsymbol{\mu},\mathbf{W},\sigma^2\right) &= \sum_{n=1}^{N}\left\{\log p\left(\mathbf{x}_n;\boldsymbol{\mu},\mathbf{W},\sigma^2\right)\right\} \\ &= -\frac{DN}{2}\log 2\pi - \frac{N}{2}\log|\mathbf{C}| - \frac{1}{2}\sum_{n=1}^{N}\left(\mathbf{x}_n-\boldsymbol{\mu}\right)^{\mathrm{T}}\mathbf{C}^{-1}\left(\mathbf{x}_n-\boldsymbol{\mu}\right).\end{aligned} \tag{9}$$

Infer the values of the model parameters $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2 by maximum likelihood estimation to get:

$$\boxed{\boldsymbol{\mu}_{mle} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_n \equiv \bar{\mathbf{x}}, \quad \sigma^2_{mle} = \frac{1}{D-Q}\sum_{q=Q+1}^{D}\lambda_q, \quad \mathbf{W}_{mle} = \mathbf{U}\left(\boldsymbol{\Lambda} - \sigma^2_{mle}\mathbf{I}_Q\right)^{1/2}\mathbf{R}.} \tag{10}$$

The log-likelihood of $\mathbf{x}$ is:

$$\begin{aligned}\mathcal{L}\left(\mathbf{x};\boldsymbol{\mu},\mathbf{W},\sigma^2\right) &= \sum_{n=1}^{N}\left\{\log p\left(\mathbf{x}_n;\boldsymbol{\mu},\mathbf{W},\sigma^2\right)\right\} \\ &= -\frac{DN}{2}\log 2\pi - \frac{N}{2}\log|\mathbf{C}| - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}_n-\boldsymbol{\mu}).\end{aligned} \tag{9}$$

Infer the values of the model parameters $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2 by maximum likelihood estimation to get:

$$\boxed{\boldsymbol{\mu}_{mle} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_n \equiv \bar{\mathbf{x}}, \quad \sigma^2_{mle} = \frac{1}{D-Q}\sum_{q=Q+1}^{D}\lambda_q, \quad \mathbf{W}_{mle} = \mathbf{U}\left(\boldsymbol{\Lambda} - \sigma^2_{mle}\mathbf{I}_Q\right)^{1/2}\mathbf{R}.} \tag{10}$$

σ^2_{mle} is given as the average of the discarded eigenvalues

The log-likelihood of $\mathbf{x}$ is:

$$\begin{aligned}\mathcal{L}\left(\mathbf{x};\boldsymbol{\mu},\mathbf{W},\sigma^2\right) &= \sum_{n=1}^{N}\left\{\log p\left(\mathbf{x}_n;\boldsymbol{\mu},\mathbf{W},\sigma^2\right)\right\}\\ &= -\frac{DN}{2}\log 2\pi - \frac{N}{2}\log|\mathbf{C}| - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}_n-\boldsymbol{\mu}).\end{aligned} \tag{9}$$

Infer the values of the model parameters $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2 by maximum likelihood estimation to get:

$$\boxed{\boldsymbol{\mu}_{mle} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_n \equiv \bar{\mathbf{x}}, \quad \sigma^2_{mle} = \frac{1}{D-Q}\sum_{q=Q+1}^{D}\lambda_q, \quad \mathbf{W}_{mle} = \mathbf{U}\left(\boldsymbol{\Lambda} - \sigma^2_{mle}\mathbf{I}_Q\right)^{1/2}\mathbf{R}.} \tag{10}$$

the columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{S}$ corresponding to the Q maximal eigenvalues λ_q

The log-likelihood of $\mathbf{x}$ is:

$$\begin{aligned}\mathcal{L}\left(\mathbf{x};\boldsymbol{\mu},\mathbf{W},\sigma^2\right) &= \sum_{n=1}^{N}\left\{\log p\left(\mathbf{x}_n;\boldsymbol{\mu},\mathbf{W},\sigma^2\right)\right\} \\ &= -\frac{DN}{2}\log 2\pi - \frac{N}{2}\log|\mathbf{C}| - \frac{1}{2}\sum_{n=1}^{N}\left(\mathbf{x}_n-\boldsymbol{\mu}\right)^{\mathrm{T}}\mathbf{C}^{-1}\left(\mathbf{x}_n-\boldsymbol{\mu}\right).\end{aligned} \tag{9}$$

Infer the values of the model parameters $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2 by maximum likelihood estimation to get:

$$\boxed{\boldsymbol{\mu}_{mle} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_n \equiv \bar{\mathbf{x}}, \quad \sigma_{mle}^2 = \frac{1}{D-Q}\sum_{q=Q+1}^{D}\lambda_q, \quad \mathbf{W}_{mle} = \mathbf{U}\left(\boldsymbol{\Lambda} - \sigma_{mle}^2\mathbf{I}_Q\right)^{1/2}\mathbf{R}.} \tag{10}$$

$$\boldsymbol{\Lambda} = \operatorname{diag}\left(\lambda_1,\ldots,\lambda_Q\right)$$

Probabilistic Principal Component Analysis

The log-likelihood of $\mathbf{x}$ is:

$$\begin{aligned}\mathcal{L}\left(\mathbf{x};\boldsymbol{\mu},\mathbf{W},\sigma^2\right) &= \sum_{n=1}^{N}\left\{\log p\left(\mathbf{x}_n;\boldsymbol{\mu},\mathbf{W},\sigma^2\right)\right\} \\ &= -\frac{DN}{2}\log 2\pi - \frac{N}{2}\log|\mathbf{C}| - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}_n-\boldsymbol{\mu}).\end{aligned} \tag{9}$$

Infer the values of the model parameters $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2 by maximum likelihood estimation to get:

$$\boxed{\boldsymbol{\mu}_{mle} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_n \equiv \bar{\mathbf{x}}, \quad \sigma^2_{mle} = \frac{1}{D-Q}\sum_{q=Q+1}^{D}\lambda_q, \quad \mathbf{W}_{mle} = \mathbf{U}\left(\boldsymbol{\Lambda} - \sigma^2_{mle}\mathbf{I}_Q\right)^{1/2}\mathbf{R}.} \tag{10}$$

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

We need to find the expected value of $\mathbf{W}\mathbf{z}_n + \boldsymbol{\mu} + \boldsymbol{\epsilon}$ conditioned on a data instance $\mathbf{x}_n$. i.e. we need to evaluate

$$\boxed{\mathbf{W}\mathbb{E}\left[\mathbf{z}_n|\mathbf{x}_n\right] + \boldsymbol{\mu}}. \tag{11}$$

The posterior predictive distribution $p(\mathbf{z}|\mathbf{x})$ can be derived easily from standard results for Gaussian distributions and using Eqs. (5) and (7). It is given by

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}), \sigma^2\mathbf{M}^{-1}), \tag{12}$$

where $\mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}_Q$. So $\mathbb{E}\left[\mathbf{z}_n|\mathbf{x}_n\right] = \mathbf{M}^{-1}\mathbf{W}^T\left(\mathbf{x}_n - \boldsymbol{\mu}\right)$.

Compare the above result with the analogous from the PCA model

$$\tilde{\mathbf{x}}_n = \mathbf{U}\mathbf{z}_n + \bar{\mathbf{x}}.$$

Expressing PPCA as an EM-algorithm

PPCA is a latent variable model $\Rightarrow$ can infer the model parameters $\mathbf{W}$ and σ^2 through an EM algorithm.

EM is computationally more efficient:
though iterative, EM does not require the evaluation of the $D \times D$ covariance matrix ($\sim \mathcal{O}\left(ND^2\right)$ operations), nor the eigen-decomposition of $\mathbf{S}$ ($\sim \mathcal{O}\left(D^3\right)$ operations),
$\Rightarrow$ computationally faster for D large.

Substitute $\bar{\mathbf{x}}$ for $\boldsymbol{\mu}$. The complete data log likelihood is

$$\begin{aligned}\hat{\mathcal{L}}\left(\mathbf{x}, \mathbf{z}; \boldsymbol{\mu}, \mathbf{W}, \sigma^2\right) &= \sum_{n=1}^{N} \left\{\log p\left(\mathbf{x}_n; \boldsymbol{\mu}, \mathbf{W}, \sigma^2\right) + \log p\left(\mathbf{z}_n | \mathbf{x}_n; \boldsymbol{\mu}, \mathbf{W}, \sigma^2\right)\right\} \\ &= -\frac{DN}{2}\log\left(2\pi\sigma^2\right) - \frac{QN}{2}\log\left(2\pi\right) \\ &\quad - \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left\|\mathbf{x}_n - \mathbf{W}\mathbf{z}_n - \boldsymbol{\mu}\right\|^2 - \frac{1}{2}\sum_{n=1}^{N}\left\|\mathbf{z}_n\right\|^2.\end{aligned} \tag{13}$$

Tipping, Bishop (1999b)

Taking the expectation of the log-likelihood w.r.t the data $\mathcal{X}$ and maximising w.r.t the model parameters $\mathbf{W}$, σ^2 gives

- E-step equations

$$\begin{aligned}\mathbb{E}\left[\mathbf{z}_n\right] &= \mathbf{M}^{-1}\mathbf{W}^T\left(\mathbf{x}_n - \boldsymbol{\mu}\right), \\ \mathbb{E}\left[\mathbf{z}_n\mathbf{z}_n^T\right] &= \sigma^2\mathbf{M}^{-1} + \mathbb{E}\left[\mathbf{z}_n\right]\mathbb{E}\left[\mathbf{z}_n\right]^T.\end{aligned} \tag{14}$$

- M-step equations:

$$\begin{aligned}\mathbf{W}_{new} &= \sum_{n=1}^{N}\left\{\left(\mathbf{x}_n - \boldsymbol{\mu}\right)\mathbb{E}\left[\mathbf{z}_n\right]^T\right\}\left(\sum_{n=1}^{N}\mathbb{E}\left[\mathbf{z}_n\mathbf{z}_n^{\mathrm{T}}\right]\right)^{-1}, \\ \sigma^2_{new} &= \frac{1}{ND}\sum_{n=1}^{N}\left\{\|\mathbf{x}_n - \boldsymbol{\mu}\|^2 - 2\mathbb{E}\left[\mathbf{z}_n\right]^{\mathrm{T}}\mathbf{W}_{new}^{\mathrm{T}}\left(\mathbf{x}_n - \boldsymbol{\mu}\right)\right. \\ &\quad \left. + \mathrm{tr}\left(\mathbb{E}\left[\mathbf{z}_n\mathbf{z}_n^{\mathrm{T}}\right]\mathbf{W}_{new}^{\mathrm{T}}\mathbf{W}_{new}\right)\right\}.\end{aligned} \tag{15}$$

Tipping, Bishop (1999b)

Expectation-Maximisation algorithm for PPCA

Algorithm 2: EM-PPCA

begin

$\boldsymbol{\mu}$ =data mean;

initialise model parameters $\mathbf{W}$ and σ^2;

while *(until convergence of* $\mathbf{W}$*)* **do**

```
/* E step                                                          */
```

$\mathbf{M} = \left(\mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma^2\mathbf{I}_Q\right); \mathbf{M}^{-1} = \mathrm{inv}\left(\mathbf{M}\right);$

$\langle \mathbf{z}_n \rangle = \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}\left(\mathbf{x}_n - \boldsymbol{\mu}\right) \quad \forall n$

$\left\langle \mathbf{z}_n\mathbf{z}_n^{\mathrm{T}} \right\rangle = \sigma^2\mathbf{M}^{-1} + \langle \mathbf{z}_n \rangle \langle \mathbf{z}_n \rangle^{\mathrm{T}} \quad \forall n$

```
/* M step                                                          */
```

$\mathbf{W} = \sum_{n=1}^{N}\left(\mathbf{x}_n - \boldsymbol{\mu}\right)\langle \mathbf{z}_n \rangle^{\mathrm{T}}\left[\sum_{n=1}^{N}\left\langle \mathbf{z}_n\mathbf{z}_n^{\mathrm{T}} \right\rangle\right]^{-1}$

$\sigma_j^2 = \frac{1}{ND}\left\{\sum_{n=1}^{N}\|\mathbf{x}_n\|^2 - 2\sum_{n=1}^{N}\langle \mathbf{z}_n \rangle^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\left(\mathbf{x}_n - \boldsymbol{\mu}\right)\right.$
$\left. + \sum_{n=1}^{N}\mathrm{tr}\left[\left\langle \mathbf{z}_n\mathbf{z}_n^{\mathrm{T}} \right\rangle \mathbf{W}^{\mathrm{T}}\mathbf{W}\right]\right\}$

end

```
/* update the value of M^-1                                        */
```

$\mathbf{M} = \left(\mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma^2\mathbf{I}_Q\right); \mathbf{M}^{-1} = \mathrm{inv}\left(\mathbf{M}\right);$

Return $\mathbf{W}$, $\boldsymbol{\mu}$, $\mathbf{M}^{-1}$;

end

EM algorithm for PPCA: schematic

Idea: linearise locally the neighbourhood of the datapoints.

We use a fixed number J of PPCA models.
We assume that each data point $\mathbf{x}_n$ is generated by one of the PPCA models:
assign to $\mathbf{x}_n$, a boolean vector $\mathbf{r}_n$ such that $r_{nj} = 1 \Leftrightarrow$ data point $\mathbf{x}_n$ is taken from the j^{th} PPCA model and $r_{nj} = 0$ otherwise.

For each model $p\left(\mathbf{x}_n | r_{nj} = 1\right)$, we assign a proportion $\pi_j = \mathbb{P}\left(r_{nj} = 1\right)$ such that $\sum_i \pi_j = 1$.
For simplicity we denote the event "$r_{nj} = 1$" simply by "j".
The revised likelihood will now be

$$\mathcal{L}\left(\mathbf{X}; \boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{W}, \sigma^2\right) = \sum_{n=1}^{N} \log\left[p\left(\mathbf{x}_n\right)\right] = \sum_{n=1}^{N} \log\left\{\sum_{j=1}^{J} \pi_j p\left(\mathbf{x}_n | j\right)\right\}. \quad (16)$$

Tipping, Bishop (1999a)

Once we have observed the corresponding $\mathbf{x}_n$, we obtain the posterior probability

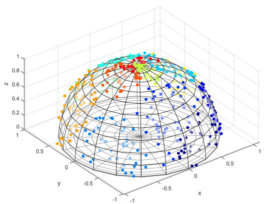
$$R_{nj} = p\left(j|\mathbf{x}_n\right) = \pi_j p\left(\mathbf{x}_n|j\right)/p\left(\mathbf{x}_n\right). \tag{17}$$

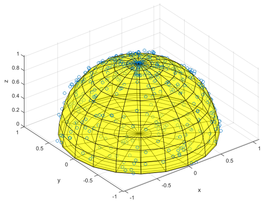
This can be seen as the *responsibility* for generating data point $\mathbf{x}_n$ from mixture j. Taking the expectation of the complete data log-likelihood w.r.t the data $\mathcal{X}$ we get

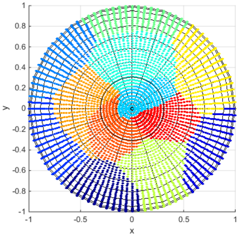
$$\begin{aligned}\left\langle \hat{\mathcal{L}}\left(\mathbf{X},\mathbf{Z};\theta\right)\right\rangle = \sum_{n=1}^{N}\sum_{j=1}^{J}\Bigg[R_{nj}\Bigg\{&\log\pi_j - \frac{Q}{2}p\log 2\pi - \frac{D}{2}\log 2\pi\sigma^2 - \frac{1}{2}\left\langle \mathbf{z}_{nj}\mathbf{z}_{nj}^T\right\rangle \\ &-\frac{1}{2\sigma^2}\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 + \frac{1}{\sigma^2}\left(\mathbf{x}_n - \boldsymbol{\mu}_j\right)^T\mathbf{W}_j\left\langle \mathbf{z}_{nj}\right\rangle \\ &-\frac{1}{2\sigma^2}\mathrm{tr}\left(\mathbf{W}^T\mathbf{W}\left\langle \mathbf{z}_{nj}\mathbf{z}_{nj}^T\right\rangle\right)\Bigg\}\Bigg].\end{aligned} \tag{18}$$

Mixtures of Probabilistic Principal Component Analysers

Consider a synthetic dataset comprised of points lying on the surface of a unit hemisphere, that have undergone a random translation sampled from a Gaussian distribution. We fit a mixture of 12 PPCA models to the data. Reconstruction is performed a) by voting, b) by averaging over all PPCA models.







Suppose, the data lie on some compact $Q-$dimensional smooth submanifold $\mathcal{M}$ of $\mathbb{R}^D$.

Instead of their Euclidean distances, take into account the geodesic distances between points $\mathbf{x}, \mathbf{z} \in \mathcal{M}$:

$$d_M(\mathbf{x}, \mathbf{z}) = \inf_{\gamma} \{\text{length}(\gamma)\},$$

where, γ is any piecewise smooth path from $\mathbf{x}$ to $\mathbf{z}$.

This will allow us to construct an embedding of the data in a Q-dimensional Euclidean space $\mathcal{M}$ that best describes the manifolds intrinsic geometry.

Since the manifold is not known beforehand, we need to find a way of approximating the geodesic distances.

We can approximate the geodesic distance from an arbitrary point $\mathbf{x}$ to $\mathbf{z}$ by

- their Euclidean distance if $\mathbf{z}$ is near $\mathbf{x}$
- summing over Euclidean distances between intermediate points, if $\mathbf{z}$ is far from $\mathbf{x}$

Assumptions:
Assume that $\mathcal{X}$ lies on a $Q-$dimensional Riemannian manifold $\mathcal{M} \subset \mathcal{R}^D$, where $Q \ll D$. Denote the geodesic distance of points on $\mathcal{M}$ by d_M.
Assume there exists an isometric mapping $f : \mathcal{M} \mapsto \mathbb{R}^Q$ from the manifold $\mathcal{M}$ to the Euclidean space of dimensionality Q, so that

$$\|f(\mathbf{x}) - f(\mathbf{z})\| = d_M(\mathbf{x}, \mathbf{z}) \ \forall \mathbf{x}, \mathbf{z} \in \mathcal{M}.$$

Seek to find the image of $\mathcal{X}$ under $f(\mathcal{X}) = \mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\} \in \mathbb{R}^Q$.
$\mathcal{Y}$ describes the points in $\mathcal{X}$ completely, in a space of lower dimensionality Q.

Tenenbaum, Silva, Langford (2000)

Given a definition for a neighbourhood $\mathcal{N}(\mathbf{x})$ of $\mathbf{x} \in \mathcal{X}$, we construct a weighted graph $G = [\mathcal{X}, E]$ such that edge $(\mathbf{x}_i - \mathbf{x}_j) \in E \iff \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)$. The weights of edges in G are given by $d_G(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|$. If $(\mathbf{x}_i - \mathbf{x}_j) \notin E$, we say that $d_G(\mathbf{x}_i, \mathbf{x}_j) = \infty$.

Let $\Gamma(\mathbf{a}, \mathbf{b})$ be the set of all piecewise linear paths from $\mathbf{a}$ to $\mathbf{b}$ of the form $\gamma = \left(\mathbf{x}_{\pi(0)}, \ldots, \mathbf{x}_{\pi(P-1)}\right)$ where π is some permutation of the $(1, \ldots, N)$ such that $\mathbf{x}_{\pi(0)} = \mathbf{a}$ and $\mathbf{x}_{\pi(P-1)} = \mathbf{b}$ and $P \leq N$ some integer.
Define the path distance along γ by $d_\gamma = \sum_{p=1}^{P-1} d_G\left(\mathbf{x}_{\pi(p-1)}, \mathbf{x}_{\pi(p)}\right)$, and the graph metric:

$$d_\Gamma(\mathbf{a}, \mathbf{b}) = \inf_{\gamma \in \Gamma(\mathbf{a}, \mathbf{b})} d_\gamma.$$

It can be shown that provided $\mathcal{M}$ is geodesically convex (no holes), $d_\Gamma \approx d_M$.

Tenenbaum, Silva, Langford (2000)
Bernstein, de Silva, Langford, Tenenbaum (2000)

Isomap

Typically a neighbourhood is defined as either the open ball of radius ϵ centred at $\mathbf{x}$ or the set of $k-$nearest neighbours.

- $\mathcal{N}(\mathbf{x}) = \{\mathbf{z} \in \mathcal{X} : \|\mathbf{x} - \mathbf{z}\| < \epsilon\}$
- $\mathcal{N}(\mathbf{x}) =$ the k datapoints $\mathbf{z} \in \mathcal{X} \setminus \{\mathbf{x}\}$ whose Euclidean distance from $\mathbf{x}$ is the smallest.

To find the shortest paths between points on the graph $\mathbf{G}$ we use Dijkstra's algorithm, that is computationally efficient on sparse graphs.

Isomap

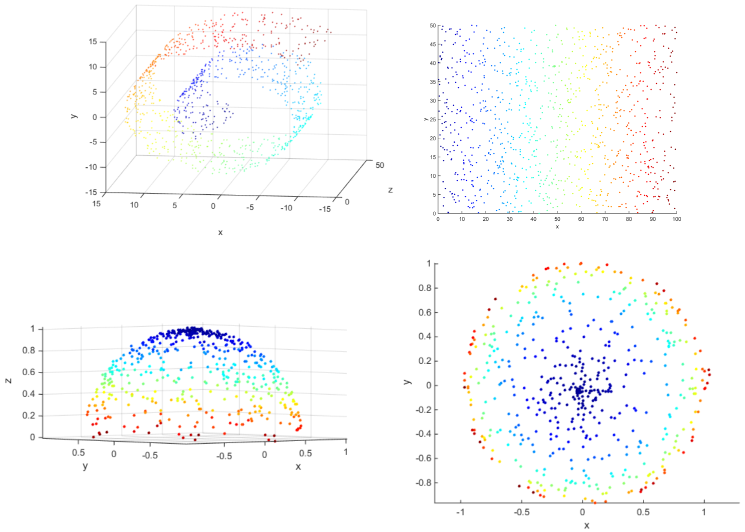
Having computed the shortest path distances $d_\Gamma(\mathbf{x}_i, \mathbf{x}_j)$ for each pair of $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$, and using $d_\Gamma \approx d_M$, and that $\|\mathbf{y}_i - \mathbf{y}_j\| = d_M(\mathbf{x}_i, \mathbf{x}_j)$, we can construct a matrix $\mathbf{S}$ s.t. $S_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|^2$.
We can find the dot products $\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j$ for each pair $\mathbf{y}_i, \mathbf{y}_j \in \mathcal{Y}$ from $\|\mathbf{y}_i - \mathbf{y}_j\|^2 = \|\mathbf{y}_i\|^2 - 2\mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j + \|\mathbf{y}_j\|^2$. We summarise this by

$$\mathbf{S}_c = -\tfrac{1}{2}\mathbf{HSH},$$

where $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_N)^\intercal$, $\mathbf{S}_c = \mathbf{YY}^{\mathrm{T}}$ and $H_{ij} = \delta_{ij} - \frac{1}{N}$.

Need to find the decomposition of $\mathbf{S}_c$ into $\mathbf{YY}^{\mathrm{T}}$.
Easy to do using an eigen-decomposition of the symmetric $\mathbf{S}_c$, into $\mathbf{S}_c = \mathbf{U\Lambda U}^{\mathrm{T}}$ (where $\mathbf{U}$ has columns the eigenvectors of $\mathbf{S}_c$ and $\mathbf{\Lambda}$ is the diagonal matrix of the eigenvalues).
Then compute $\mathbf{Y} = \mathbf{U}\sqrt{\mathbf{\Lambda}}$.

Isomap



GTM typically assumes $Q = 2$ (used for visualisation purposes).

The motivation for this algorithm originates from biology and in particular from the model of self-organisation exhibited by the sensory cortex of the brain. The idea is that similar stimuli are responsible for the activation of neighbouring neurons.

Bishop,Svensén, Williams (1998a)
Bishop,Svensén, Williams (1998b)
Svensén (1998)

Generative Topographic Model

The GTM model assumes the existence of a set of latent variables $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_K\} \subset \mathbb{R}^Q$, arranged in latent space in a $Q-$dimensional regular grid of nodes and a function $\mathbf{y} : \mathbb{R}^Q \to \mathbb{R}^D$ mapping

$$\mathbf{z} \mapsto \mathbf{y}\left(\mathbf{z}; \mathbf{W}\right) \in \mathbb{R}^D,$$

where $\mathbf{W}$ is the matrix of governing model parameters.

The data $\mathbf{x}$ however only approximately live on a Q-dimensional space$\Rightarrow$ include an additive noise variable $\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \beta^{-1}\mathbf{I}_D\right)$ such that

$$\mathbf{x} = \mathbf{y}\left(\mathbf{z}; \mathbf{W}\right) + \boldsymbol{\epsilon},$$

and

$$p\left(\mathbf{x}|\mathbf{z}; \mathbf{W}, \beta\right) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\left\|\mathbf{y}\left(\mathbf{z}; \mathbf{W}\right) - \mathbf{x}\right\|^2\right\}. \tag{19}$$

A probability density $p(\mathbf{z})$ over the latent space, is introduced, inducing a probability distribution in data space

$$p(\mathbf{x};\theta) = \int p(\mathbf{x}|\mathbf{z};\mathbf{W},\beta)\, p(\mathbf{z})\, \mathrm{d}\mathbf{x}, \tag{20}$$

We define a prior over latent space of the form

$$p(\mathbf{z}) = \frac{1}{K}\sum_{i=1}^{K} \delta(\mathbf{z}-\mathbf{z}_i). \tag{21}$$

$\Rightarrow$ Each node $\mathbf{z}_i$ will be mapped to a point $\mathbf{y}(\mathbf{z}_i;\mathbf{W})$ in dataspace which will be the centre of a Gaussian distribution $\mathcal{N}\left(\mathbf{y}(\mathbf{z}_i;\mathbf{W}), \beta^{-1}\mathbf{I}\right)$.

$$p(\mathbf{x};\mathbf{W},\beta) = \frac{1}{K}\sum_{i=1}^{K}\left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\|\mathbf{x}-\mathbf{y}(\mathbf{z}_i;\mathbf{W})\|^2\right\}. \tag{22}$$

constrained Gaussian mixture model [**?**]: the position of the centres $\mathbf{y}(\mathbf{z}_i; \mathbf{W})$ is governed by the mapping $\mathbf{y}$.

Smoothness of the mapping $\mathbf{y}$ suffices to ensure that the centres $\mathbf{y}(\mathbf{z}_i; \mathbf{W})$ have the desired “topographic ordering” (i.e. that points in latent space are mapped to points close in data space).

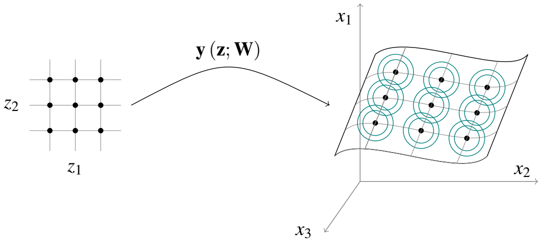


Figure: Schematic view of the GTM model: Latent variable points on a regular grid in latent space (left) are mapped under $\mathbf{y}(\mathbf{z}; \mathbf{W})$ onto the dataspace (right). Each latent variable induces a Gaussian distribution in dataspace, centred at $\mathbf{y}(\mathbf{z}; \mathbf{W})$.

Typically, we take $\mathbf{y}$ to be a *generalised linear regression model* [?]: Consider the components $y_d(\mathbf{z};\mathbf{W})$, $d = 1,\ldots,D$. Each will be of the form

$$y_d(\mathbf{z};\mathbf{W}) = \sum_{m=1}^{M} \phi_m(\mathbf{z}) W_{md} : \tag{23}$$

The basis functions ϕ_m need only be a non-linear and smooth functions over $\mathbf{z}$. Typically we use a Radial Basis Network [?]:

$$\phi_m(\mathbf{z}) = \begin{cases} \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{z}-\boldsymbol{\mu}_m\|^2\right\} & \text{if } m \leq M-Q-1 \\ z_q & \text{if } m = M-Q+1+q \quad \forall q = 1,\ldots,Q \\ 1 & \text{if } m = M \end{cases} . \tag{24}$$

For simplicity, we write Eq. (23) in matrix form as $\mathbf{Y} = \boldsymbol{\Phi}\mathbf{W}$.

Generative Topographic Mapping:an EM algorithm

Gaussian Mixture model: suggestive to use an EM-algorithm to infer the values of $\mathbf{W}$ and β.

Data point $\mathbf{x}_n$ is generated by node $\mathbf{z}_k$ with responsibility:

$$R_{kn} = p\left(\mathbf{z}_k | \mathbf{x}_n; \mathbf{W}, \beta\right) = \frac{p\left(\mathbf{x}_n | \mathbf{z}_k; \mathbf{W}, \beta\right) p\left(\mathbf{z}_k\right)}{\sum_{\kappa=1}^{K} p\left(\mathbf{x}_n | \mathbf{z}_\kappa; \mathbf{W}, \beta\right) p\left(\mathbf{z}_\kappa\right)} = \frac{p\left(\mathbf{x}_n | \mathbf{z}_k; \mathbf{W}, \beta\right)}{\sum_{\kappa=1}^{K} p\left(\mathbf{x}_n | \mathbf{z}_\kappa; \mathbf{W}, \beta\right)}. \tag{25}$$

The expected value of the complete data log-likelihood is

$$\begin{aligned} \left\langle \hat{\mathcal{L}}\left(\mathcal{X}, \mathcal{Z}; \mathbf{W}, \beta\right)\right\rangle &= \sum_{n=1}^{N}\sum_{k=1}^{K}\left[R_{kn}\left\{\log p\left(\mathbf{x}_n | \mathbf{z}_k; \mathbf{W}, \beta\right) + \log p\left(\langle \mathbf{z}_k\rangle\right)\right\}\right] \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\left[R_{kn}\left\{\frac{D}{2}\log\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2}\|\mathbf{x}_n - \mathbf{W}\phi\left(\langle \mathbf{z}_k\rangle\right)\|^2\right\} + \log p\left(\mathbf{z}_n\right)\right]. \end{aligned} \tag{26}$$

Maximising w.r.t to the model parameters $\mathbf{W}$, β gives

- The update of $\mathbf{W}$ is given as the solution of

$$\boldsymbol{\Phi}^\intercal \mathbf{G} \boldsymbol{\Phi} \mathbf{W}^{mle} = \boldsymbol{\Phi}^\intercal \mathbf{R} \mathbf{X}, \tag{27}$$

  where $\mathbf{G}$ is diagonal matrix with elements $G_{kk} = \sum_{n=1}^{N} R_{kn}$.

- The update for β is given by

$$\frac{1}{\beta^{mle}} = \frac{1}{ND} \sum_{n=1}^{N} \sum_{k=1}^{K} R_{kn} \|\mathbf{W}^{mle} \phi\left(\mathbf{z}_k\right) - \mathbf{x}_n\|^2. \tag{28}$$

To control overfitting, we turn to Bayesian framework, and treat $\mathbf{W}$ as a random variable. We introduce a prior distribution $p(\mathbf{W})$ expressing an initial belief about the value of the weights $\mathbf{W}$:

$$p(\mathbf{W}) = \left(\frac{\alpha}{2\pi}\right)^{MD/2} \exp\left\{-\frac{\alpha}{2}\|\mathbf{W}\|_F^2\right\}. \tag{29}$$

This leads to the following updating equation for $\mathbf{W}^{mle}$

$$\left(\mathbf{\Phi}^\intercal \mathbf{G} \mathbf{\Phi} + \lambda \mathbf{I}_M\right) \mathbf{W}^{mle} = \mathbf{\Phi}^\intercal \mathbf{R} \mathbf{X}, \tag{30}$$

where $\lambda = \alpha/\beta$.

To visualise the results, we use either:

- *Posterior-mode projection*: the mode of the posterior distribution of the latent variables

$$\mathbf{z}_n^{mode} = \arg\max_{z_k} p\left(\mathbf{z}_k | \mathbf{x}_n\right) = \arg\max_{z_k} R_{kn} \tag{31}$$

- *Posterior-mean projection*: the mean of the posterior distribution of the latent variables

$$\mathbf{z}_n^{mean} = \sum_{k=1}^{K} \mathbf{z}_k p\left(\mathbf{z}_k | \mathbf{x}_n\right) = \sum_{k=1}^{K} R_{kn} \mathbf{z}_k \tag{32}$$

Svensén (1998)

EM algorithm for GTM: schematic

MSc Dissertation

Acknowledgments

- Professor N. Zabaras (Thesis Advisor)
- EPSRC Strategic Package Project EP/L027682/1 for research at WCPM