Continuous Latent Variable Models: Probabilistic and Bayesian PCA,Kernel PCA

Prof. Nicholas Zabaras
University of Notre Dame
Notre Dame, IN, USA

Email: nzabaras@nd.edu
URL: https://www.zabaras.com/

November 8, 2017

Contents

- Continuous Latent Variable Model, Low-Dimensional Manifold of a Data Set, Generative Point of View, Unidentifiability

- Principal Component Analysis, Maximum variance formulation, Minimum-error formulation, PCA Reconstruction , PCA and SVD, Canonical Correlation Analysis

- Applications of PCA (off-line Digit Images), Whitening of the data with PCA , PCA for Visualization, PCA for High-Dimensional Data

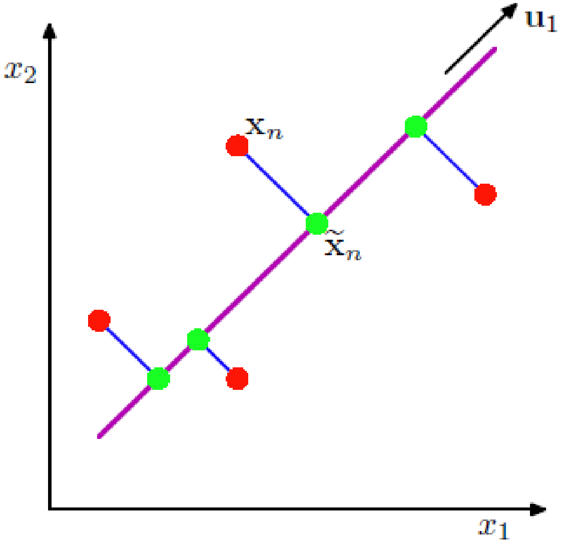
- Probabilistic PCA, Problem setup, Maximum likelihood PCA, EM algorithm for PCA, EM Missing Values, EM for $\sigma^2 \to 0$, Model Selection

- Bayesian PCA, Kernel PCA

Following closely Chris Bishops' PRML book (Chapter 12)

Principal Component Analysis

- 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) minimizing the sum-of-squares of the projection errors (blue lines)



- J. Shlens (2005). *A Tutorial on Principal Component Analysis*

PCA with 1D Principal Subspace

- Consider first the projection onto 1D space ($M = 1$).

- We define the direction of this subspace using vector $\boldsymbol{u}_1$, which we choose to be a unit vector: $\boldsymbol{u}_1^T\boldsymbol{u}_1 = 1$

- Each data point $\boldsymbol{x}_n$ is then projected onto the scalar $\boldsymbol{u}_1^T\boldsymbol{x}_n$

- The mean of the projected data is given as: $\boldsymbol{u}_1^T\overline{\boldsymbol{x}}$

- Similarly, the variance of the projected data is:

$$\frac{1}{N}\sum_{n=1}^{N}\left\{\boldsymbol{u}_1^T\boldsymbol{x}_n-\boldsymbol{u}_1^T\overline{\boldsymbol{x}}\right\}^2=\frac{1}{N}\sum_{n=1}^{N}\left\{\boldsymbol{u}_1^T\boldsymbol{x}_n-\boldsymbol{u}_1^T\overline{\boldsymbol{x}}\right\}\left\{\boldsymbol{u}_1^T\boldsymbol{x}_n-\boldsymbol{u}_1^T\overline{\boldsymbol{x}}\right\}^T=\boldsymbol{u}_1^T\boldsymbol{S}\boldsymbol{u}_1$$

- Key idea of PCA: Maximize the projected variance $\boldsymbol{u}_1^T\boldsymbol{S}\boldsymbol{u}_1$ with respect to $\boldsymbol{u}_1$ under the constraint $\boldsymbol{u}_1^T\boldsymbol{u}_1 = 1$

PCA with 1D Principal Subspace

- Introduce the Lagrange multiplier λ_1 and define the unconstrained maximization of

$$\max_{\boldsymbol{u}_1,\lambda_1}\left\{\boldsymbol{u}_1^T\boldsymbol{S}\boldsymbol{u}_1+\lambda_1\left(1-\boldsymbol{u}_1^T\boldsymbol{u}_1\right)\right\}$$

- We can see immediately that the solution satisfies:

$$\boldsymbol{S}\boldsymbol{u}_1=\lambda_1\boldsymbol{u}_1$$

- $\boldsymbol{u}_1$ must be an eigenvector of $\boldsymbol{S}$ with eigenvalue λ_1

- From the eigen-problem note that the variance of the projected data is $\boldsymbol{u}_1^T\boldsymbol{S}\boldsymbol{u}_1=\lambda_1$ so λ_1 needs to be the largest eigenvalue of $\boldsymbol{S}$.

- $\boldsymbol{u}_1$ is called the first principal component.

PCA with 1D Principal Subspace

- Additional Principal Components: maximize the projected variance amongst all possible directions orthogonal to those already considered.

- Using induction, you can show: For an M-dimensional projection space, the optimal linear projection for which the variance of the projected data is maximized is defined by the M eigenvectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_M$ of the data covariance matrix $\boldsymbol{S}$ corresponding to the largest eigenvalues $\lambda_1, \dots, \lambda_M$.

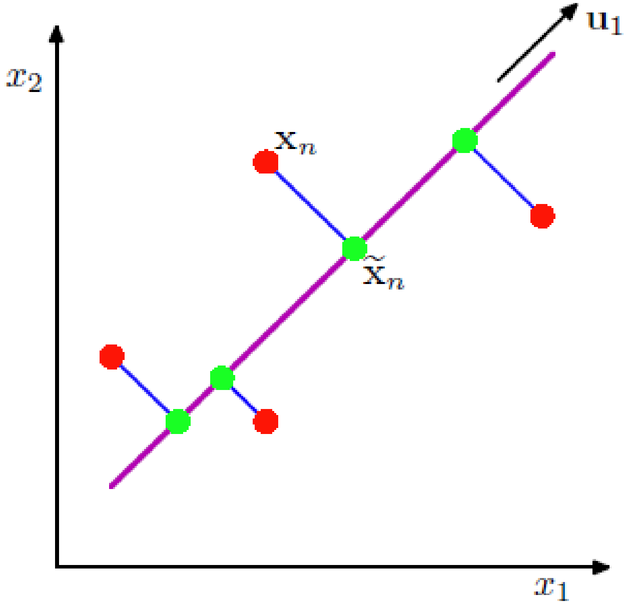
- The computational cost of finding the M principal components (i.e. finding the first M eigenvalues and eigenvectors of $\boldsymbol{S}$) is $\mathcal{O}(MD^2)$.

PCA: Minimum Error Formulation

$$\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n = \sum_{i=M+1}^{D} \left\{ \left(\boldsymbol{x}_n - \overline{\boldsymbol{x}} \right)^T \boldsymbol{u}_i \right\} \boldsymbol{u}_i$$

- We see that the displacement vector $\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n$ lies in the space orthogonal to the principal subspace, i.e. is a linear combination of $\{\boldsymbol{u}_i\}$ for $i = M + 1, \ldots, D$.

- We can also see this geometrically:



  - The projected points $\widetilde{\boldsymbol{x}}_n$ must lie within the principal subspace, but can move them freely within that subspace.

  - The minimum error is then obtained by the orthogonal projection.

PCA: Minimum Error Formulation

- We obtain an expression for the distortion measure J *as* a function purely of the $\{\boldsymbol{u}_i\}$ in the form

$$\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n = \sum_{i=M+1}^{D} \left\{ (\boldsymbol{x}_n - \overline{\boldsymbol{x}})^T \boldsymbol{u}_i \right\} \boldsymbol{u}_i$$

$$J = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=M+1}^{D} \left(\mathbf{x}_n^T \mathbf{u}_i - \overline{\mathbf{x}}^T \mathbf{u}_i \right)^2 = \sum_{i=M+1}^{D} \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i$$

$$J = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n\|^2$$

- The minimum of J is obtained when $\{\boldsymbol{u}_i\}, i = M + 1, \ldots, D$ are the eigenvectors of $\boldsymbol{S}$ associated to the smallest eigenvalues.

- Consider $D = 2$ and $M = 1$. Let $\lambda_1 > \lambda_2$. The principal subspace is aligned with the eigenvector with the larger eigenvalue λ_1, and the min value of $J = \lambda_2$ is obtained by choosing $\boldsymbol{u}_2$ corresponding to λ_2.

Compression of the Original Data Set

- Using the earlier equations, we can derive:

$$\tilde{\boldsymbol{x}}_n = \sum_{i=1}^{M} z_{ni}\boldsymbol{u}_i + \sum_{i=M+1}^{D} b_i\boldsymbol{u}_i$$

$$z_{nj} = \boldsymbol{x}_n^T\boldsymbol{u}_j \;,\; j = 1,\ldots,M$$

$$b_j = \overline{\boldsymbol{x}}^T\boldsymbol{u}_j \;,\; j = M+1,\ldots,D$$

$$\Rightarrow \tilde{\boldsymbol{x}}_n = \sum_{i=1}^{M}(\boldsymbol{x}_n^T\boldsymbol{u}_i)\boldsymbol{u}_i + \sum_{i=M+1}^{D}(\overline{\boldsymbol{x}}^T\boldsymbol{u}_i)\boldsymbol{u}_i \Rightarrow$$

$$Data\,Compression: \tilde{\boldsymbol{x}}_n = \overline{\boldsymbol{x}} + \sum_{i=1}^{M}(\boldsymbol{x}_n^T\boldsymbol{u}_i - \overline{\boldsymbol{x}}^T\boldsymbol{u}_i)\boldsymbol{u}_i$$

- In deriving the last equation, we used the completeness of $\boldsymbol{u}_i$, i.e.

$$\overline{\boldsymbol{x}} = \sum_{i=1}^{D}\left(\overline{\boldsymbol{x}}^T\boldsymbol{u}_i\right)\boldsymbol{u}_i$$

- This clearly shows the compression of the data set: From the D-dimensional $\boldsymbol{x}_n$ to the M-dimensional vector with components $\left(\boldsymbol{x}_n^T\boldsymbol{u}_i - \overline{\boldsymbol{x}}^T\boldsymbol{u}_i\right)$.

PCA Reconstruction

- To reconstruct the data in the original D-dimensional space from a representation in the M-dimensional principal subspace, we simply use:

$$\widetilde{\boldsymbol{x}}_n = \overline{\boldsymbol{x}} + \sum_{i=1}^{M} (\boldsymbol{x}_n^T \boldsymbol{u}_i - \overline{\boldsymbol{x}}^T \boldsymbol{u}_i) \boldsymbol{u}_i$$

- If $M = D$, there is no dimensionality reduction but a rotation to align with the principal components e.g from $\{x_{n_1}, \dots, x_{n_D}\}$ to $\{a_{n_1}, \dots, a_{n_D}\}$ where:

$$\boldsymbol{x}_n = \sum_{i=1}^{M=D} a_{ni} \boldsymbol{u}_i$$

Old Faithful Data Set: Whitening

- With PCA we can normalize the data to give them zero mean and **unit covariance (different variables become decorrelated)**.

- Consider the key eigenvalue problem in PCA in a matrix form:

$$\boldsymbol{SU}=\boldsymbol{UL},\quad \boldsymbol{L}=diag\left(\lambda_1,\ldots,\lambda_D\right),\quad \boldsymbol{U}=\left[\boldsymbol{u}_1\ldots\boldsymbol{u}_D\right](orthogonal)$$

- For each data point $\boldsymbol{x}_n$, define a transformed value as:

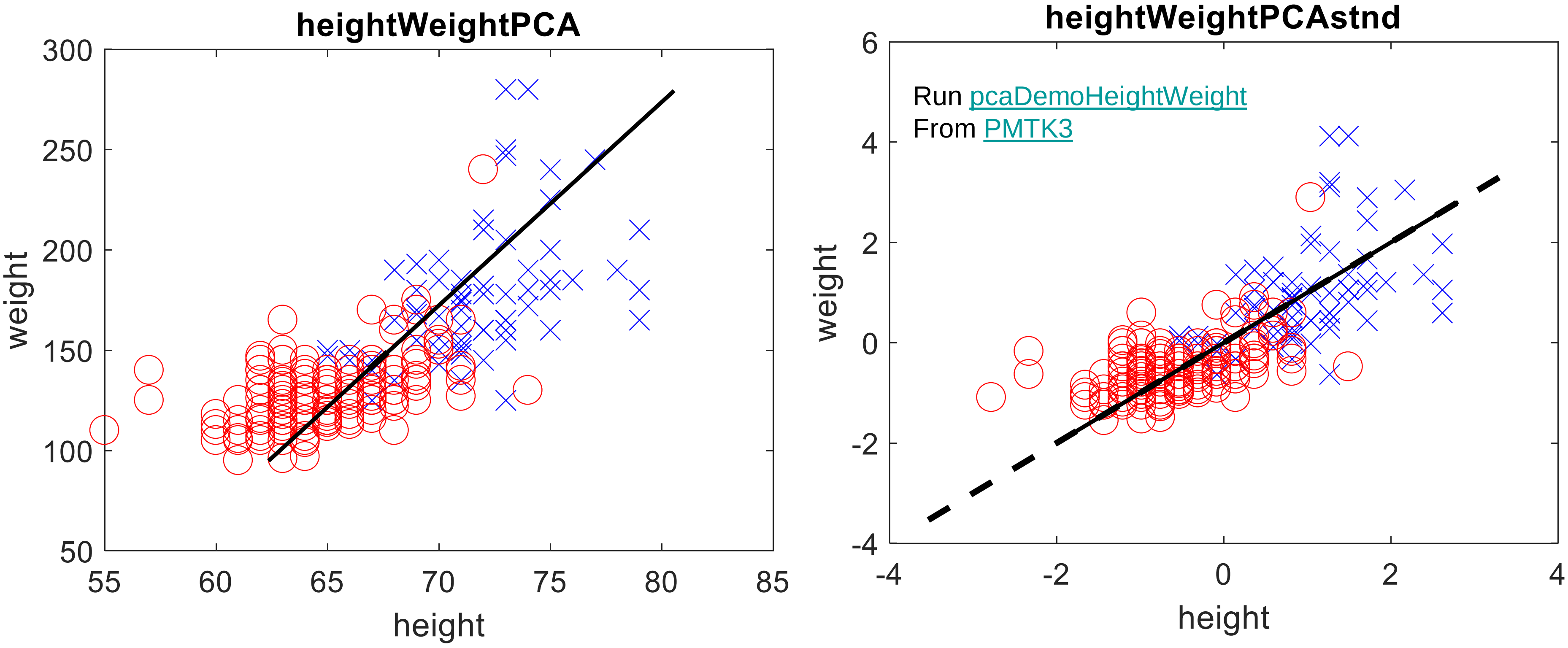
$$\text{Whitening of the data}: \boldsymbol{y}_n=\boldsymbol{L}^{-1/2}\boldsymbol{U}^T\left(\boldsymbol{x}_n-\overline{\boldsymbol{x}}\right)$$

- The set $\boldsymbol{y}_n$ has zero mean and its covariance is the identity:

$$\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{y}_n\boldsymbol{y}_n^T=\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{L}^{-1/2}\boldsymbol{U}^T\left(\boldsymbol{x}_n-\overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_n-\overline{\boldsymbol{x}}\right)^T\boldsymbol{U}\boldsymbol{L}^{-1/2}=\boldsymbol{L}^{-1/2}\boldsymbol{U}^T\boldsymbol{S}\boldsymbol{U}\boldsymbol{L}^{-1/2}=\boldsymbol{I}$$

Old Faithful Data Set: PCA Example

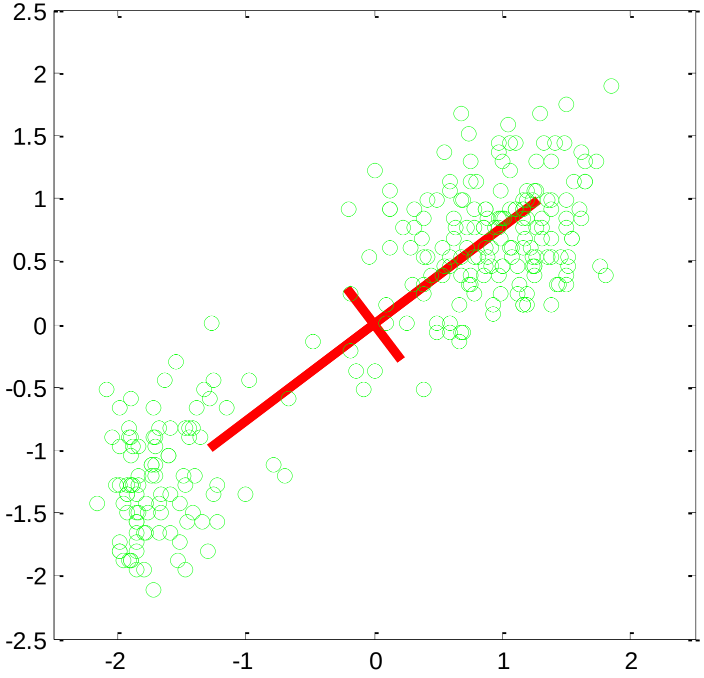
- It should be standard practice to standardize the data first before performing PCA. This is equivalent to working with correlation matrices instead of covariance matrices.
- (Left) PCA for raw data (Right) PCA of standardized data.

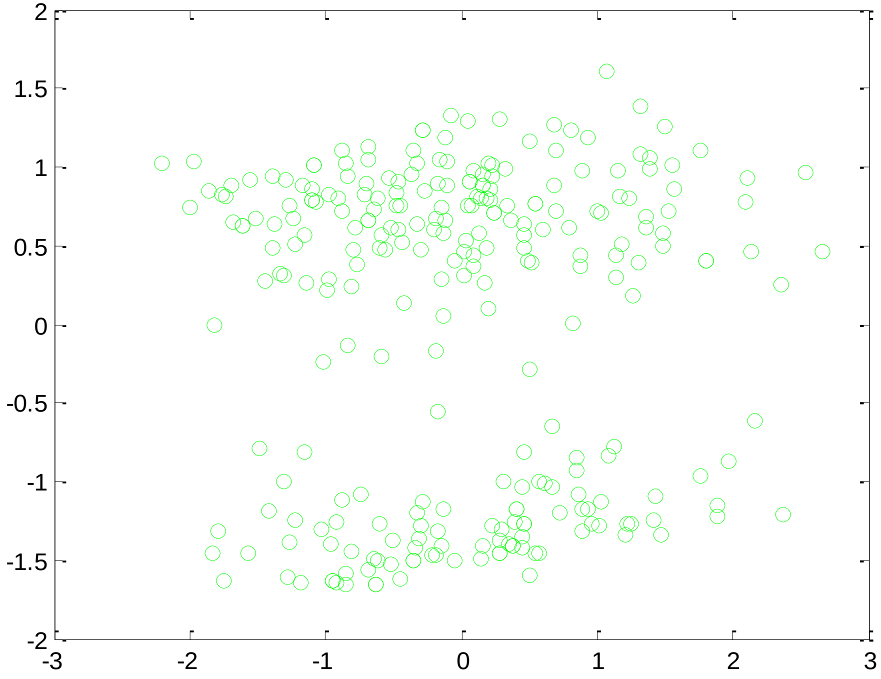


Run pcaDemoHeightWeight
From PMTK3

Old Faithful Data Set: PCA Example

- Left: Standardizing individual variables to zero mean and unit variance. The principal axes of the normalized set are shown for the range $\pm\lambda_i^{1/2}$ (variables still correlated)
- Right: Whitening of the data (zero mean, unit covariance)





MatLab Code

PCA For High Dimensional Data

- In some applications of PCA, the number of data points (N) is smaller than the dimensionality (D) of the data space (e.g. 100 images each with $100{,}000$ pixels).

- N points in a *D*-dimensional space, where $N \ll D$, defines a linear subspace whose dimensionality is at most $N - 1$.
  - There is little point in applying PCA for values of M that are greater than $N - 1$.

- If we perform PCA, we will find that at least $D - N + 1$ of the eigenvalues are zero, corresponding to eigenvectors along whose directions the data set has zero variance.

PCA For High Dimensional Data

- Typical algorithms for finding the eigenvectors of a $D \times D$ matrix have a computational cost that scales like $\mathcal{O}(D^3)$.

- Direct application of PCA in e.g. the image example will be computationally infeasible.

PCA For High Dimensional Data

- To resolve the problem, let $\boldsymbol{X}$ to be the $N \times D$-dimensional centered data matrix, whose nth row is given by $\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right)^T$.

- The covariance $D \times D$ matrix can then be written as:

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right)^T = \frac{1}{N}\boldsymbol{X}^T\boldsymbol{X}$$

- The corresponding eigenvector equation is:

$$\frac{1}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i$$

- Pre-multipling both sides from the left by $\boldsymbol{X}$ gives:

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T\left(\boldsymbol{X}\boldsymbol{u}_i\right) = \lambda_i\left(\boldsymbol{X}\boldsymbol{u}_i\right)$$

PCA For High Dimensional Data

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T\left(\boldsymbol{X}\boldsymbol{u}_i\right)=\lambda_i\left(\boldsymbol{X}\boldsymbol{u}_i\right)$$

- ❑ Let us define:

$$\boldsymbol{v}_i=\boldsymbol{X}\boldsymbol{u}_i$$

- ❑ The eigenvalue problem becomes:

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{v}_i=\lambda_i\boldsymbol{v}_i$$

- ❑ This is an eigenvector equ. for the $N \times N$ matrix $\boldsymbol{X}\boldsymbol{X}^T/N$
- ❑ This has the same $N-1$ eigenvalues as the original covariance matrix (which has an additional $D-N+1$ zero eigenvalues). Note we have $N-1$ (and not N) eigenvalues because the data are centered ($\boldsymbol{X}\boldsymbol{X}^T$ has rank $N-1$).
- ❑ Thus we can solve the eigenvector problem in spaces of lower dimensionality with computational cost $\mathcal{O}(N^3)$.

PCA For High Dimensional Data

- To determine the eigenvectors $\boldsymbol{u}_i$, we multiply by $\boldsymbol{X}^T$: .

$$\frac{1}{N}\left(\boldsymbol{X}^T\boldsymbol{X}\right)\boldsymbol{X}^T\mathbf{v}_i=\lambda_i\left(\boldsymbol{X}^T\mathbf{v}_i\right)$$

- The covariance matrix recall has eigenvectors $\boldsymbol{u}_i$. Thus with proper rescaling (assuming $\boldsymbol{v}_i$ is already normalized):

$$\boldsymbol{u}_i=\frac{1}{\left(N\lambda_i\right)^{1/2}}\boldsymbol{X}^T\mathbf{v}_i$$

- This approach is indeed simple:
  - first evaluate $\boldsymbol{X}\boldsymbol{X}^T$ and then find its eigenvectors and eigenvalues and
  - then compute the eigenvectors in the original data space from the equ. above.

Probabilistic PCA

PCA as the maximum likelihood solution of a probabilistic latent variable model (Tipping & Bishop 1997, 1999, Roweis, 1998)

- ❑ Constrained form of the Gaussian distribution: the number of free parameters is restricted while the model still captures dominant correlations in the data
- ❑ Derive an efficient EM algorithm for PCA
- ❑ Allows dealing with missing values in the dataset
- ❑ Mixture of probabilistic PCA models

- Tipping, M. E. and C. M. Bishop (1997). Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Aston University.
- Tipping, M. E. and C. M. Bishop (1999b). Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, *Series B* **21**(3), 611–622.
- Roweis, S. (1998). EM algorithms for PCA and SPCA. In M. I. Jordan, M. J. Kearns, and S. A. Solla (Eds.), *Advances in Neural Information Processing Systems*, Volume 10, pp. 626–632. MIT Press.

Probabilistic PCA – Model

- Introduce a latent variable $\boldsymbol{z}$ corresponding to the principal-component subspace.

- Define a Gaussian prior distribution $p(\boldsymbol{z})$, together with a Gaussian conditional distribution $p(\boldsymbol{x}|\boldsymbol{z})$ for the observed variable $\boldsymbol{x}$ conditioned on the value of $\boldsymbol{z}$.

- The prior distribution over $\boldsymbol{z}$ is given by

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0},\boldsymbol{I})$$

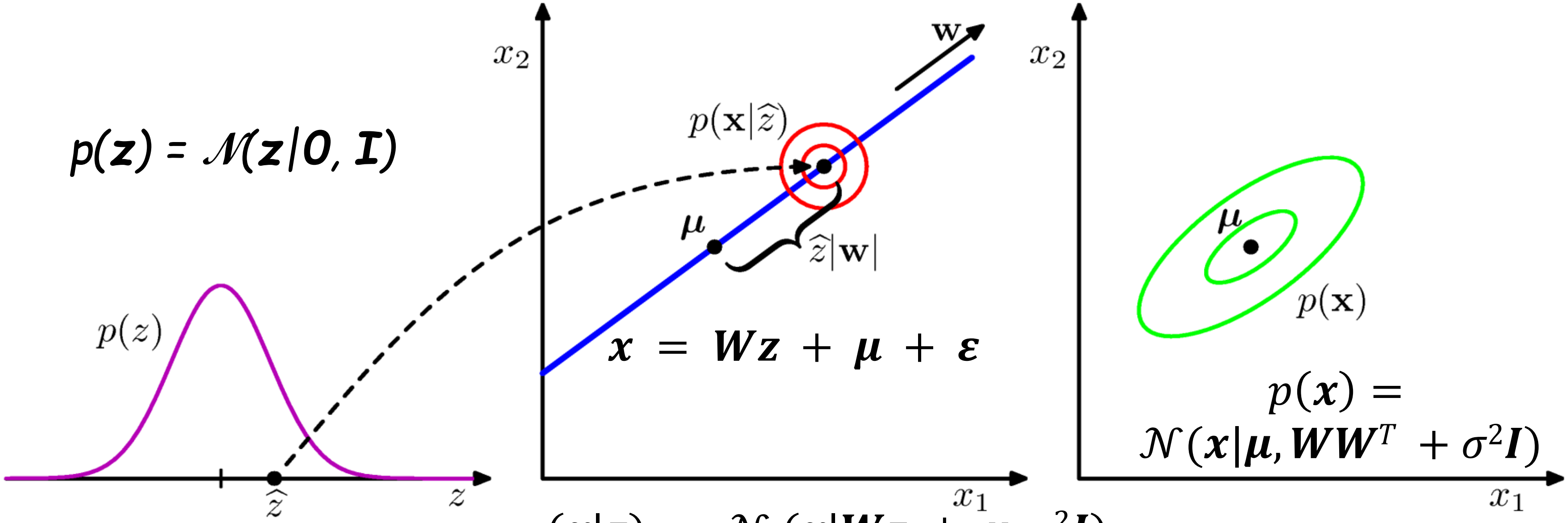
- The conditional distribution of the observed variable $\boldsymbol{x}$, conditioned on the value of $\boldsymbol{z}$, is again Gaussian:

$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu},\sigma^2\boldsymbol{I}\right)$$

  The mean of $\boldsymbol{x}$ is a linear function of $\boldsymbol{z}$ governed by the $D \times M$ matrix $\boldsymbol{W}$ and the *D*-dimensional vector $\boldsymbol{\mu}$.

Probabilistic PCA – Generative View Point

- ❑ Mapping from the latent space to the data space.
- ❑ Assume here 2D data and 1D latent space.
- ❑ An observed $\boldsymbol{x}$ is generated by drawing a value $\hat{z}$ from $p(\boldsymbol{z})$ & then a value for $\boldsymbol{x}$ from an isotropic Gaussian distribution (red circles) having mean $\boldsymbol{w}\hat{z}+\boldsymbol{\mu}$ and covariance $\sigma^2\boldsymbol{I}$. The green ellipses are the density contours of $p(\boldsymbol{x})$.



$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|0, \boldsymbol{I})$

$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\varepsilon}$

$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{W}^T + \sigma^2\boldsymbol{I})$

$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2\boldsymbol{I})$

Probabilistic PCA – Predictive Distribution

$$p(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{C}\right) = \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 \boldsymbol{I}\right)$$

- ❑ To compute the predictive distribution, we need to be able to invert $\boldsymbol{C}$ ($D \times D$ matrix). We use the matrix inversion Lemma:

$$\boldsymbol{C}^{-1} = \left(\boldsymbol{W}\boldsymbol{W}^T + \sigma^2 \boldsymbol{I}\right)^{-1} = \sigma^{-2}\boldsymbol{I} - \sigma^{-2}\boldsymbol{W}\boldsymbol{M}^{-1}\boldsymbol{W}^T$$

where

$$\boldsymbol{M} = \boldsymbol{W}^T\boldsymbol{W} + \sigma^2 \boldsymbol{I} \;(M \times M \; matrix)$$

- ❑ The cost of this inversion is reduced from $\mathcal{O}(D^3)$ to $\mathcal{O}(M^3)$!

Probabilistic PCA – Posterior Distribution

- The posterior distribution can be derived directly from earlier results on linear Gaussian models.
  - We know $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \boldsymbol{I})$
  - The conditional: $p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2\boldsymbol{I})$
  - From these we conclude:

$$p(\mathbf{z}\,|\,\mathbf{x}) = \mathcal{N}\left(\mathbf{z}\,|\,\boldsymbol{M}^{-1}\boldsymbol{W}^T(\mathbf{x}-\boldsymbol{\mu}), \sigma^2\boldsymbol{M}^{-1}\right), \;\; \boldsymbol{M} = \boldsymbol{W}^T\boldsymbol{W} + \sigma^2\boldsymbol{I}$$

Here we used results from an earlier lecture:

$$\begin{cases} p\left(\boldsymbol{x}\right) = \mathcal{N}\left(\boldsymbol{x}\,|\,\boldsymbol{\mu}, \Lambda^{-1}\right) \\ p\left(\boldsymbol{y}\,|\,\boldsymbol{x}\right) = \mathcal{N}\left(\boldsymbol{y}\,|\,\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{L}^{-1}\right) \end{cases} \Rightarrow$$

$$p\left(\boldsymbol{x}\,|\,\boldsymbol{y}\right) = \mathcal{N}\left(\boldsymbol{x}\,|\,\Sigma\left(\Lambda\boldsymbol{\mu} + \boldsymbol{A}^T\boldsymbol{L}(\boldsymbol{y}-\boldsymbol{b})\right), \Sigma\right),$$

$$\Sigma = \left(\Lambda + \boldsymbol{A}^T\boldsymbol{L}\boldsymbol{A}\right)^{-1}$$

Maximum Likelihood PCA

- Consider determining the model parameters using maximum likelihood. Using

$$p(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, \underbrace{\boldsymbol{W}\boldsymbol{W}^T + \sigma^2 \boldsymbol{I}}_{\boldsymbol{C}}\right)$$

we derive:

$$\ln p\left(\boldsymbol{X} \mid \boldsymbol{\mu}, \boldsymbol{W}, \sigma^2\right) = \sum_{n=1}^{N} \ln p\left(\boldsymbol{x}_n \mid \boldsymbol{\mu}, \boldsymbol{W}, \sigma^2\right)$$

$$= -\frac{ND}{2}\ln\left(2\pi\right) - \frac{N}{2}\ln\left|\boldsymbol{C}\right| - \frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})^T \boldsymbol{C}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu})$$

- Setting the derivative wrt $\boldsymbol{\mu}$ equal to zero gives:

$$\boxed{\boldsymbol{\mu} = \sum_{n=1}^{N} \boldsymbol{x}_n \Big/ N \equiv \overline{\boldsymbol{x}}}$$

Maximum Likelihood PCA

- The log-likelihood is then simplified as:

$$\ln p\left(\boldsymbol{X}\,|\,\boldsymbol{\mu},\boldsymbol{W},\sigma^2\right)=-\frac{ND}{2}\ln\left(2\pi\right)-\frac{N}{2}\ln\left|\boldsymbol{C}\right|-\frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{x}_n-\overline{\boldsymbol{x}})^T\boldsymbol{C}^{-1}(\boldsymbol{x}_n-\overline{\boldsymbol{x}})$$

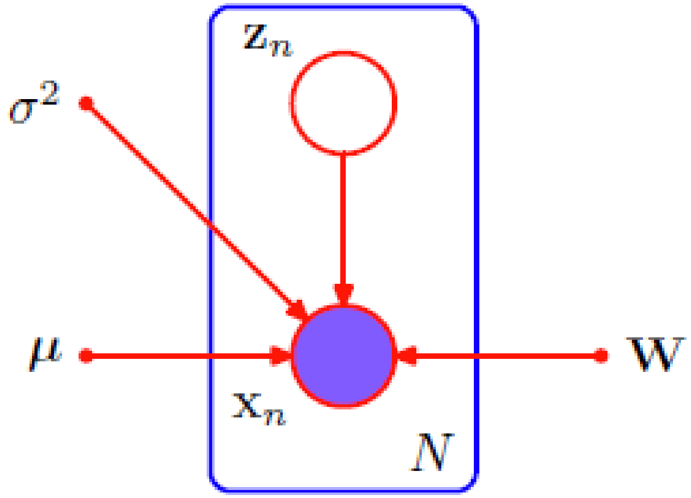
or

$$\ln p\left(\boldsymbol{X}\,|\,\boldsymbol{\mu},\boldsymbol{W},\sigma^2\right)=-\frac{N}{2}\left\{D\ln\left(2\pi\right)+\ln\left|\boldsymbol{C}\right|+Tr\left(\boldsymbol{C}^{-1}\boldsymbol{S}\right)\right\},$$

$$\boldsymbol{S}=\frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n-\overline{\boldsymbol{x}})(\boldsymbol{x}_n-\overline{\boldsymbol{x}})^T$$

- Maximization wrt $\boldsymbol{W}$ and σ^2 can also be done analytically:

$$\boldsymbol{W}_{ML}=\boldsymbol{U}_M\left(\boldsymbol{L}_M-\sigma^2\boldsymbol{I}\right)^{1/2}\boldsymbol{R}$$



- Tipping, M. E. and C. M. Bishop (1999b). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B* **21**(3), 611–622.
- Roweis, S. (1998). EM algorithms for PCA and SPCA. In M. I. Jordan, M. J. Kearns, and S. A. Solla (Eds.), *Advances in Neural Information Processing Systems*, Volume 10, pp. 626–632. MIT Press.

Maximum Likelihood PCA

$$\boldsymbol{W}_{ML} = \boldsymbol{U}_M \left(\boldsymbol{L}_M - \sigma^2 \boldsymbol{I} \right)^{1/2} \boldsymbol{R}$$

- ❑ $\boldsymbol{U}_M$ is a $D \times M$ matrix whose columns are given by any subset (of size M) of the eigenvectors of the data covariance matrix $\boldsymbol{S}$.
- ❑ $\boldsymbol{L}_M$ is the $M \times M$ diagonal matrix with elements given by the corresponding eigenvalues λ_i of $\boldsymbol{S}$.
- ❑ $\boldsymbol{R}$ is an arbitrary $M \times M$ orthogonal matrix.
- ❑ The max of the likelihood function is obtained when the M eigenvectors are chosen to be those whose eigenvalues are the M largest (all other solutions being saddle points).
- ❑ For eigenvectors arranged in order of decreasing λ_i values, the *M* principal eigenvectors are $\boldsymbol{u}_1, \dots, \boldsymbol{u}_M$. The columns of ***W*** then define the principal subspace as in standard PCA.

Maximum Likelihood PCA

- The corresponding MLE solution for σ^2 is given as:

$$\sigma_{ML}^2 = \frac{1}{D-M}\sum_{i=M+1}^{N}\lambda_i$$

i.e. the average of the discarded eigenvalues.

- $\boldsymbol{R}$ in $\boldsymbol{W}_{ML} = \boldsymbol{U}_M\left(\boldsymbol{L}_M - \sigma^2\boldsymbol{I}\right)^{1/2}\boldsymbol{R}$ is a rotation matrix in the M dimensional latent space.

- Substituting this into the predictive variance $\boldsymbol{C} = \boldsymbol{W}_{ML}\boldsymbol{W}_{ML}^T + \sigma_{ML}^2\boldsymbol{I}$ and using $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$, we see that $\boldsymbol{C}$ is independent of $\boldsymbol{R}$.

- The predictive density is unchanged by rotations in the latent space (statistical non-identifiability).

Maximum Likelihood PCA

- Consider the variance of the predictive distribution $\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 \boldsymbol{I}$ along some direction defined by the unit vector $\boldsymbol{v}$ given by $\boldsymbol{v}^T\boldsymbol{C}\boldsymbol{v}$.

- Let $\boldsymbol{v}$ be orthogonal to the principal subspace $\boldsymbol{U}$, i.e. $\boldsymbol{v}$ is given by some linear combination of the discarded eigenvectors.

- Then $\boldsymbol{v}^T\boldsymbol{U} = 0$ and hence

$$\boldsymbol{v}^T\boldsymbol{C}\boldsymbol{v} = \boldsymbol{v}^T\left(\boldsymbol{W}\boldsymbol{W}^T + \sigma^2\boldsymbol{I}\right)\boldsymbol{v} = \sigma^2$$

- Thus the model predicts a noise variance orthogonal to the principal subspace which it was shown to be the average of the discarded eigenvalues.

$$\sigma_{ML}^2 = \frac{1}{D-M}\sum_{i=M+1}^{N}\lambda_i$$

Maximum Likelihood PCA

- Consider the variance of the predictive distribution $\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 \boldsymbol{I}$ along some direction defined by the unit vector $\boldsymbol{v}$ given by $\boldsymbol{v}^T\boldsymbol{C}\boldsymbol{v}$.

- Let now $\boldsymbol{v} = \boldsymbol{u}_i$ where $\boldsymbol{u}_i$ is one of the retained eigenvectors defining the principal subspace.

- Then using $\boldsymbol{W}_{ML} = \boldsymbol{U}_M \left(\boldsymbol{L}_M - \sigma^2 \boldsymbol{I}\right)^{1/2} \boldsymbol{R}$, and $\boldsymbol{u}_i^T \boldsymbol{u}_j = 0, j = 1, \ldots, M$ we see that:

$$\mathbf{v}^T \boldsymbol{C}\mathbf{v} = \boldsymbol{u}_i^T \left(\boldsymbol{W}\boldsymbol{W}^T + \sigma^2 \boldsymbol{I}\right)\boldsymbol{u}_i = \left(\lambda_i - \sigma^2\right) + \sigma^2 = \lambda_i$$

  i.e. the model correctly captures the variance of the data along the principal axes.

- As shown earlier, the variance in all remaining directions is approximated with a single value of σ^2. Variance is `lost' in the projections.

Maximum Likelihood PCA

$$\boldsymbol{W}_{ML} = \boldsymbol{U}_M \left(\boldsymbol{L}_M - \sigma^2 \boldsymbol{I} \right)^{1/2} \boldsymbol{R}$$

$$\sigma^2_{ML} = \frac{1}{D-M} \sum_{i=M+1}^{N} \lambda_i$$

- We construct the MLE model by solving the underlying eigenvalue problem for the data covariance matrix and evaluate $\boldsymbol{W}$ and σ^2 from the Eqs. above. We choose $\boldsymbol{R} = \boldsymbol{I}$.

- Note: if the MLE model is found using optimization methods or via the EM algorithm (see following slides), $\boldsymbol{R}$ is arbitrary and the columns of $\boldsymbol{W}$ not orthogonal.

  In this case, orthogonality can be enforced

  - as postprocessing or

  - by modifying the EM algorithm.

- Ahn, J. H. and J. H. Oh (2003). A constrained EM algorithm for principal component analysis. *Neural Computation* **15**(1), 57–65.

Maximum Likelihood PCA vs Standard PCA

- In the limit σ²→0, the posterior mean becomes:

$$\mathbb{E}\left[\boldsymbol{z}\,|\,\boldsymbol{x}\right]=\boldsymbol{M}^{-1}\boldsymbol{W}_{ML}^{T}(\boldsymbol{x}-\overline{\boldsymbol{x}}),\ \boldsymbol{M}=\boldsymbol{W}_{ML}^{T}\boldsymbol{W}_{ML}+\sigma^{2}\boldsymbol{I}\rightarrow\boldsymbol{W}_{ML}^{T}\boldsymbol{W}_{ML}\Rightarrow$$

$$\mathbb{E}\left[\boldsymbol{z}\,|\,\boldsymbol{x}\right]=\left(\boldsymbol{W}_{ML}^{T}\boldsymbol{W}_{ML}\right)^{-1}\boldsymbol{W}_{ML}^{T}(\boldsymbol{x}-\overline{\boldsymbol{x}})$$

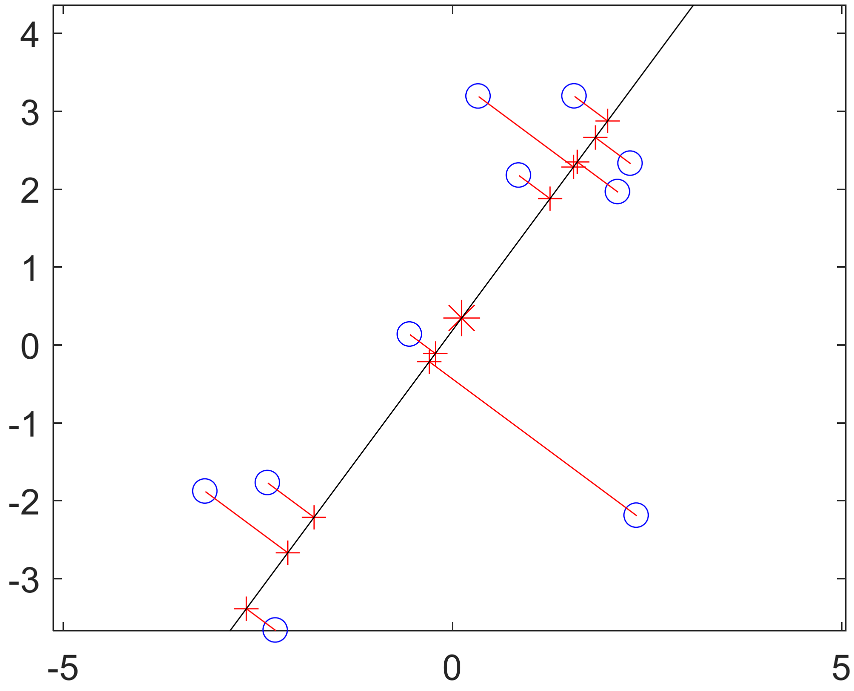
- Now substitute $\boldsymbol{W}_{ML}=\boldsymbol{U}_{M}\left(\boldsymbol{L}_{M}-\sigma^{2}\boldsymbol{I}\right)^{1/2}\boldsymbol{R}=\boldsymbol{U}_{M}\boldsymbol{L}_{M}^{1/2}$ for σ²→0 with $\boldsymbol{R}=\boldsymbol{I}$ (for consistency with PCA). Then

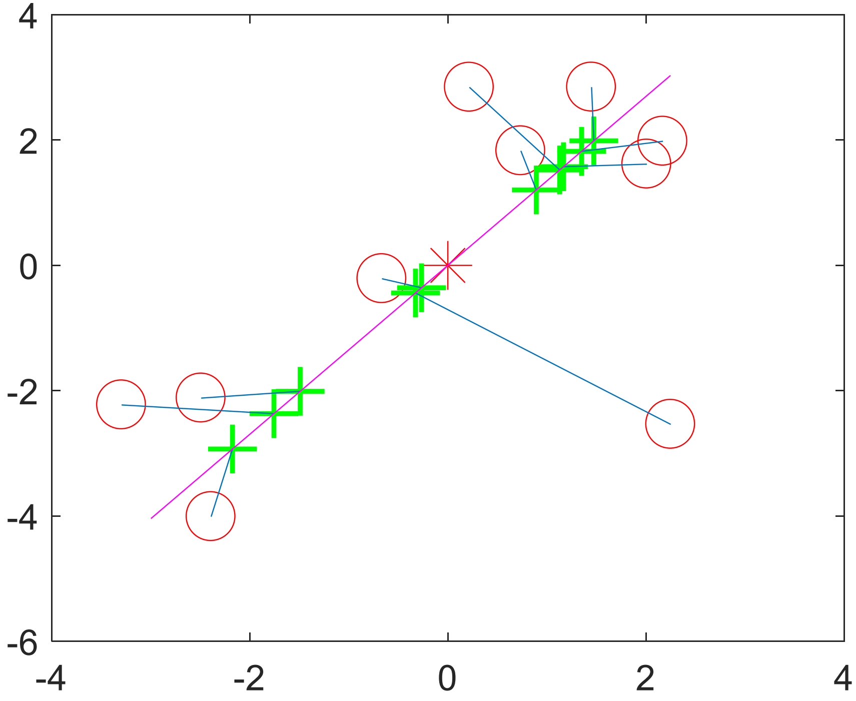
$$\mathbb{E}\left[\boldsymbol{z}\,|\,\boldsymbol{x}\right]=\left(\boldsymbol{W}_{ML}^{T}\boldsymbol{W}_{ML}\right)^{-1}\boldsymbol{W}_{ML}^{T}(\boldsymbol{x}-\overline{\boldsymbol{x}})=\boldsymbol{L}_{M}^{-1/2}\boldsymbol{U}_{M}^{T}(\boldsymbol{x}-\overline{\boldsymbol{x}})$$

- This is an orthogonal projection of the data point onto the latent space, i.e. for the limit σ²→0, we recover the standard PCA model. The posterior covariance $\mathrm{var}\left[\boldsymbol{z}\,|\,\boldsymbol{x}\right]=\sigma^{2}\boldsymbol{M}^{-1}\rightarrow 0$ and the density becomes singular.
- For $\sigma^2 > 0$, the latent projection is shifted towards the origin, relative to the orthogonal projection.

PPCA Versus PCA

- Consider PCA and PPCA where $D = 2$ and $M = 1$. The red star is the data mean. In PCA the points are orthogonally projected onto the line. In PPCA the projection is no longer orthogonal and the reconstructions are shrunk towards the data mean (red star).





Run pcaDemo2d
From PMTK3

Run ppcaDemo2d
From PMTK3

Probabilistic PCA: Number of DOF

- ❑ In PPCA, the M most significant correlations are captured while the total number of parameters grows linearly with D.

- ❑ We can see this by evaluating the DOF in PPCA:

  - The covariance $\boldsymbol{C}$ depends on $\boldsymbol{W}$ ($D \times M$), and σ^2: $\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 \boldsymbol{I}$ total parameters $DM + 1$.

  - We need to subtract the redundancy associated with rotations of the coordinate system in the latent space.

  - $\boldsymbol{R}$ is $M \times M$. In the 1st column there are $M - 1$ independent parameters (must be normalized). In the 2nd column there are $M - 2$ independent parameters (normalized & orthogonal to the 1st column), etc. $\boldsymbol{R}$ has a total of $M(M-1)/2$ independent parameters.

- ❑ The number of degrees of freedom in $\boldsymbol{C}$ grows linearly with D

$$D \times M + 1 - M \times (M-1)/2$$

EM Algorithms for PCA

- We first take the expectation of the complete-data log likelihood with respect to the posterior distribution of the latent distribution evaluated using ‘old’ parameter values.

- Maximization of this expected complete data log likelihood then yields the ‘new’ parameter values.

$$\ln p\left(\boldsymbol{X},\boldsymbol{Z}\,|\,\boldsymbol{\mu},\boldsymbol{W},\sigma^2\right)=\sum_{n=1}^{N}\left\{\ln p\left(\boldsymbol{x}_n\,|\,\boldsymbol{z}_n\right)+\ln p\left(\boldsymbol{z}_n\right)\right\}$$

- The nth row of $\boldsymbol{Z}$ is given by $\boldsymbol{z}_n$.

- Recall that $p(\boldsymbol{z})\ =\ \mathcal{N}(\boldsymbol{z}|\boldsymbol{0},\boldsymbol{I})\,,p(\boldsymbol{x}|\boldsymbol{z})\ =\ \mathcal{N}\ (\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z}\ +\ \boldsymbol{\mu},\sigma^2\boldsymbol{I})$

- We can now write the expectation with respect to the posterior distribution over the latent variables.

EM Algorithms for PCA

$$\mathbb{E}\left[\ln p\left(\boldsymbol{X},\boldsymbol{Z}\,|\,\boldsymbol{\mu},\boldsymbol{W},\sigma^2\right)\right]=-\sum_{n=1}^{N}\left\{\begin{array}{l}\dfrac{D}{2}\ln\left(2\pi\sigma^2\right)+\dfrac{1}{2}Tr\left(\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\right)\\ +\dfrac{1}{2\sigma^2}\left\|\boldsymbol{x}_n-\boldsymbol{\mu}\right\|^2-\dfrac{1}{\sigma^2}\mathbb{E}\left[\boldsymbol{z}_n\right]^T\boldsymbol{W}^T\left(\boldsymbol{x}_n-\boldsymbol{\mu}\right)\\ +\dfrac{1}{2\sigma^2}Tr\left(\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\boldsymbol{W}^T\boldsymbol{W}\right)+\dfrac{M}{2\ln\left(2\pi\right)}\end{array}\right\}$$

- E-Step: We use the old parameters to evaluate:

$$\mathbb{E}\left[\boldsymbol{z}_n\right]=\boldsymbol{M}^{-1}\boldsymbol{W}^T\left(\boldsymbol{x}_n-\overline{\boldsymbol{x}}\right) \qquad \mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]=\sigma^2\boldsymbol{M}^{-1}+\mathbb{E}\left[\boldsymbol{z}_n\right]\mathbb{E}\left[\boldsymbol{z}_n\right]^T$$

- This follows directly from

$$p(\boldsymbol{z}\,|\,\boldsymbol{x})=\mathcal{N}\left(\boldsymbol{z}\,|\,\boldsymbol{M}^{-1}\boldsymbol{W}^T(\boldsymbol{x}-\boldsymbol{\mu}),\sigma^2\boldsymbol{M}^{-1}\right),\;\; \boldsymbol{M}=\boldsymbol{W}^T\boldsymbol{W}+\sigma^2\boldsymbol{I}$$

together with the standard result

$$\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]=\mathrm{cov}\left[\boldsymbol{z}_n\right]+\mathbb{E}\left[\boldsymbol{z}_n\right]\mathbb{E}\left[\boldsymbol{z}_n\right]^T$$

EM Algorithms for PCA

$$\mathbb{E}\left[\ln p\left(\boldsymbol{X},\boldsymbol{Z}\,|\,\boldsymbol{\mu},\boldsymbol{W},\sigma^2\right)\right] = -\sum_{n=1}^{N}\left\{\begin{array}{l}\dfrac{D}{2}\ln\left(2\pi\sigma^2\right)+\dfrac{1}{2}Tr\left(\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\right)\\ +\dfrac{1}{2\sigma^2}\left\|\boldsymbol{x}_n-\boldsymbol{\mu}\right\|^2-\dfrac{1}{\sigma^2}\mathbb{E}\left[\boldsymbol{z}_n\right]^T\boldsymbol{W}^T\left(\boldsymbol{x}_n-\boldsymbol{\mu}\right)\\ +\dfrac{1}{2\sigma^2}Tr\left(\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\boldsymbol{W}^T\boldsymbol{W}\right)+\dfrac{M}{2\ln\left(2\pi\right)}\end{array}\right\}$$

- ❑ M-Step: We maximize with respect to $\boldsymbol{W}$ and σ^2 keeping the posterior statistics fixed. We obtain:

$$\boldsymbol{W}_{new} = \left[\sum_{n=1}^{N}\left(\boldsymbol{x}_n-\overline{\boldsymbol{x}}\right)\mathbb{E}\left[\boldsymbol{z}_n\right]^T\right]\left[\sum_{n=1}^{N}\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\right]^{-1}$$

$$\sigma^2_{new} = \frac{1}{ND}\sum_{n=1}^{N}\left\{\begin{array}{l}\left\|\boldsymbol{x}_n-\overline{\boldsymbol{x}}\right\|^2-2\mathbb{E}\left[\boldsymbol{z}_n\right]^T\boldsymbol{W}_{new}^T\left(\boldsymbol{x}_n-\overline{\boldsymbol{x}}\right)\\ +Tr\left(\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\boldsymbol{W}_{new}^T\boldsymbol{W}_{new}\right)\end{array}\right\}$$

Proof of the M-Step Equations

$$\mathbb{E}\left[\ln p\left(\boldsymbol{X},\boldsymbol{Z}\,|\,\boldsymbol{\mu},\boldsymbol{W},\sigma^2\right)\right]=-\sum_{n=1}^{N}\left\{\begin{aligned}&\frac{D}{2}\ln\left(2\pi\sigma^2\right)+\frac{1}{2}Tr\left(\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\right)\\&+\frac{1}{2\sigma^2}\left\|\boldsymbol{x}_n-\boldsymbol{\mu}\right\|^2-\frac{1}{\sigma^2}\mathbb{E}\left[\boldsymbol{z}_n\right]^T\boldsymbol{W}^T\left(\boldsymbol{x}_n-\boldsymbol{\mu}\right)\\&+\frac{1}{2\sigma^2}Tr\left(\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\boldsymbol{W}^T\boldsymbol{W}\right)+\frac{M}{2\ln\left(2\pi\right)}\end{aligned}\right\}$$

- ❑ The two M-equations are derived by setting the derivatives wrt ***W*** and σ² equal to zero:

$$\frac{\partial\mathbb{E}\left[\ln p\left(\boldsymbol{X},\boldsymbol{Z}\,|\,\boldsymbol{\mu},\boldsymbol{W},\sigma^2\right)\right]}{\partial\boldsymbol{W}}=\sum_{n=1}^{N}\left\{\frac{1}{\sigma^2}\left(\boldsymbol{x}_n-\boldsymbol{\mu}\right)\mathbb{E}\left[\boldsymbol{z}_n\right]^T-\frac{1}{\sigma^2}\boldsymbol{W}\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\right\}=0$$

$$\frac{\partial\mathbb{E}\left[\ln p\left(\boldsymbol{X},\boldsymbol{Z}\,|\,\boldsymbol{\mu},\boldsymbol{W},\sigma^2\right)\right]}{\partial\sigma^2}=\sum_{n=1}^{N}\left\{\begin{aligned}&-\frac{D}{2\sigma^2}-\frac{1}{\sigma^4}\mathbb{E}\left[\boldsymbol{z}_n\right]^T\boldsymbol{W}^T\left(\boldsymbol{x}_n-\boldsymbol{\mu}\right)+\frac{1}{2\sigma^4}\left\|\boldsymbol{x}_n-\boldsymbol{\mu}\right\|^2\\&+\frac{1}{2\sigma^4}Tr\left(\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\boldsymbol{W}^T\boldsymbol{W}\right)\end{aligned}\right\}=0$$

- ❑ Here we used $\frac{\partial}{\partial\boldsymbol{A}}Tr(\boldsymbol{A}\boldsymbol{B}\boldsymbol{A}^T)=\boldsymbol{A}(\boldsymbol{B}+\boldsymbol{B}^T),\ \frac{\partial}{\partial\boldsymbol{A}}Tr(\boldsymbol{A}\boldsymbol{B})=\boldsymbol{B}^T$

EM Algorithms for PCA

- ❑ Initialize the parameters

- ❑ Compute the sufficient statistics of the latent space posterior distribution in the E-Step

$$\mathbb{E}\left[\boldsymbol{z}_n\right] = \boldsymbol{M}^{-1}\boldsymbol{W}^T\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right) \qquad \mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right] = \sigma^2\boldsymbol{M}^{-1} + \mathbb{E}\left[\boldsymbol{z}_n\right]\mathbb{E}\left[\boldsymbol{z}_n^T\right]$$

- ❑ Revise the parameter values in the M-Step.

$$\boldsymbol{W}_{new} = \left[\sum_{n=1}^{N}\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right)\mathbb{E}\left[\boldsymbol{z}_n\right]^T\right]\left[\sum_{n=1}^{N}\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\right]^{-1}$$

$$\sigma_{new}^2 = \frac{1}{ND}\sum_{n=1}^{N}\left\{\begin{array}{l}\left\|\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right\|^2 - 2\left[\boldsymbol{z}_n\right]^T\boldsymbol{W}_{new}^T\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right) \\ +Tr\left(\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\boldsymbol{W}_{new}^T\boldsymbol{W}_{new}\right)\end{array}\right\}$$

EM Algorithms for PCA: Computational Cost

- Each cycle of the EM algorithm can be computationally more efficient than conventional PCA in high dimensions.

- The eigen-decomposition of the covariance matrix requires $\mathcal{O}(D^3)$ computation. If interested only in the first *M* eigenvectors, we can use algorithms that are $\mathcal{O}(MD^2)$.

- However, the evaluation of the covariance matrix itself

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right)^T$$

  takes $\mathcal{O}(ND^2)$ computations.

- Algorithms such as the snapshot method (Sirovich,1987), assume that the eigenvectors are linear combinations of the data vectors and avoid direct evaluation of the covariance matrix but are $\mathcal{O}(N^3)$ and hence unsuited to large data sets.

  - Sirovich, L. (1987). Turbulence and the dynamics of coherent structures. *Quarterly Applied Mathematics* **45**(3), 561–590.

EM Algorithms for PCA: Missing Values

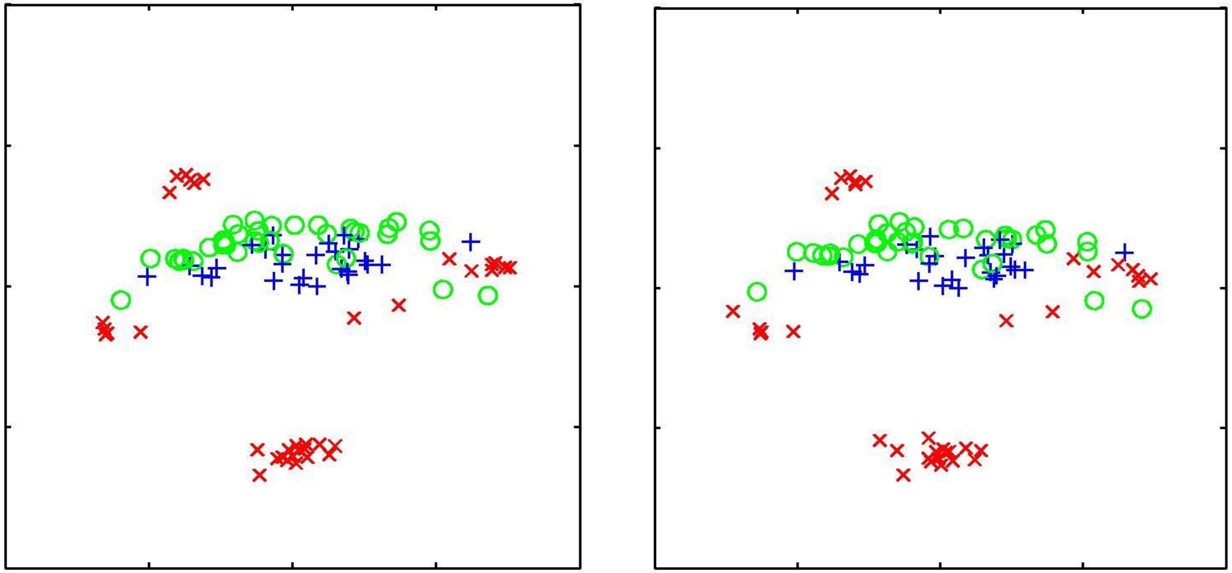
- ❑ Because we now have a fully probabilistic model for PCA, we can deal with missing data, provided that it is missing at random, by marginalizing over the distribution of the unobserved variables.

- ❑ Again these missing values can be treated using the EM algorithm.

- ❑ We give an example of the use of this approach for data visualization in the figure shown next.

EM Algorithms for PCA: Missing Values

- ❑ PPCA visualization for the first 100 data points of the oil flow data set.
- ❑ Left: the posterior mean projections of the data points on the principal subspace.
- ❑ Right: randomly omitting 30% of the variable values and using EM to handle the missing values. Even though each data point has at least one missing measurement, the plot is similar to the one obtained without missing values.



Matlab Implementation

EM Algorithm for PCA: Limit $\sigma^2 \to 0$

- When $\sigma^2 \to 0$, EM corresponds to standard PCA (Roweis, 1998)

- Defining $\widetilde{\boldsymbol{X}}$ a matrix of size $N \times D$ whose nth row is given by $\boldsymbol{x}_n - \overline{\boldsymbol{x}}$.

- Defining a matrix Ω of size $M \times N$ whose nth column is given by the vector $\mathbb{E}[\boldsymbol{z}_n]$.

- The E-Step for $\sigma^2 \to 0$ becomes

$$\mathbb{E}\left[\boldsymbol{z}_n\right] = \boldsymbol{M}^{-1}\boldsymbol{W}^T\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right) \Rightarrow \quad \boldsymbol{\Omega} = \left(\boldsymbol{W}_{old}^T \boldsymbol{W}_{old}\right)^{-1} \boldsymbol{W}_{old}^T \widetilde{\boldsymbol{X}}^T$$

This is simply the orthogonal projection of the data points on the current estimate for the principal subspace.

- Roweis, S. (1998). EM algorithms for PCA and SPCA. In M. I. Jordan, M. J. Kearns, and S. A. Solla (Eds.), *Advances in Neural Information Processing Systems*, Volume 10, pp. 626–632. MIT Press.

EM Algorithm for PCA: Limit $\sigma^2 \to 0$

- When $\sigma^2 \to 0$, EM corresponds to standard PCA

- $\widetilde{\boldsymbol{X}}$ an $N \times D$ matrix whose nth row is given by $\boldsymbol{x}_n - \overline{\boldsymbol{x}}$.

- Ω a matrix $M \times N$ whose nth column is given by $\mathbb{E}[\boldsymbol{z}_n]$.

- Noting that

$$\sigma^2 \to 0 \;\Rightarrow\; \mathbb{E}\left[\boldsymbol{z}_n \boldsymbol{z}_n^T\right] = \sigma^2 \boldsymbol{M}^{-1} + \mathbb{E}\left[\boldsymbol{z}_n\right]\mathbb{E}\left[\boldsymbol{z}_n\right]^T \to \mathbb{E}\left[\boldsymbol{z}_n\right]\mathbb{E}\left[\boldsymbol{z}_n\right]^T$$

the M-Step takes the form:

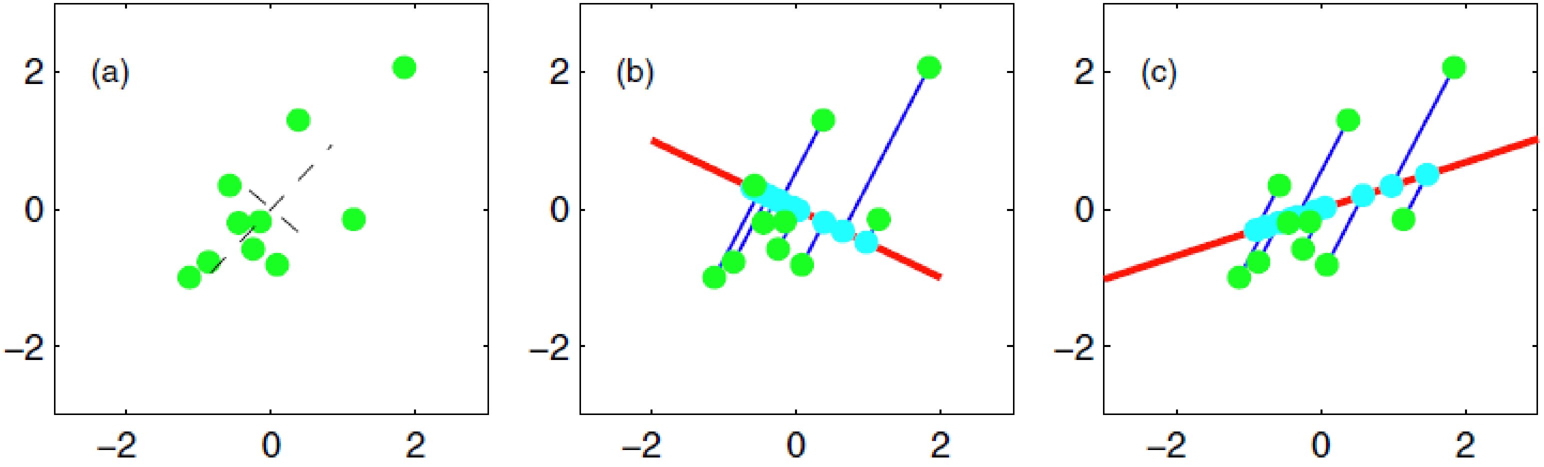
$$\boldsymbol{W}_{new} = \left[\sum_{n=1}^{N}\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right)\mathbb{E}\left[\boldsymbol{z}_n\right]^T\right]\left[\sum_{n=1}^{N}\mathbb{E}\left[\boldsymbol{z}_n \boldsymbol{z}_n^T\right]\right]^{-1} \Rightarrow \quad \boldsymbol{W}_{new} = \widetilde{\boldsymbol{X}}^T \boldsymbol{\Omega}^T (\boldsymbol{\Omega}\boldsymbol{\Omega}^T)^{-1}$$

Re-estimation of the principal subspace minimizing the squared reconstruction errors in which the projections are fixed (see interpretation next and also here)

EM Algorithm for PCA: Example, D=2, M=1

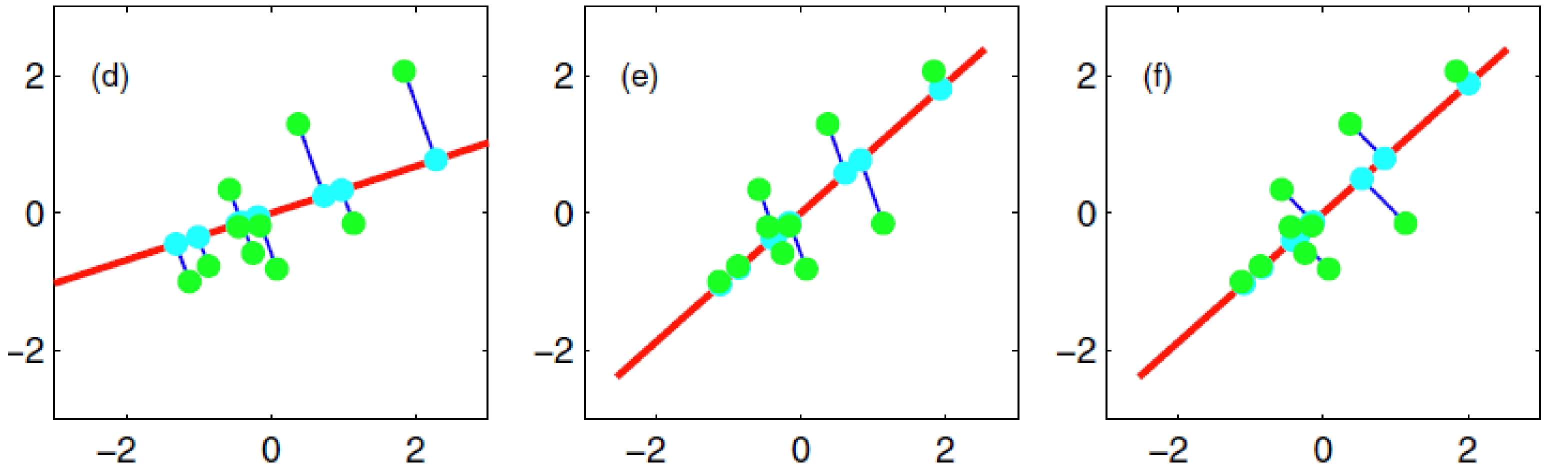
- Synthetic data illustrating the EM algorithm for PCA
  - (a) A data set $\boldsymbol{X}$ with the data points (green), together with the true principal components (eigenvectors scaled by the square roots of the eigenvalues).
  - (b) Initial configuration of the principal subspace defined by $\boldsymbol{W}$ (red) together with the projections of the latent points $\boldsymbol{Z}$ into the data space, given by $\boldsymbol{Z}\boldsymbol{W}^T$ (cyan)
  - (c) After one M-Step, the latent space has been updated with $\boldsymbol{Z}$ held fixed.

Matlab Implementation



EM Algorithm for PCA: Example, D=2, M=1

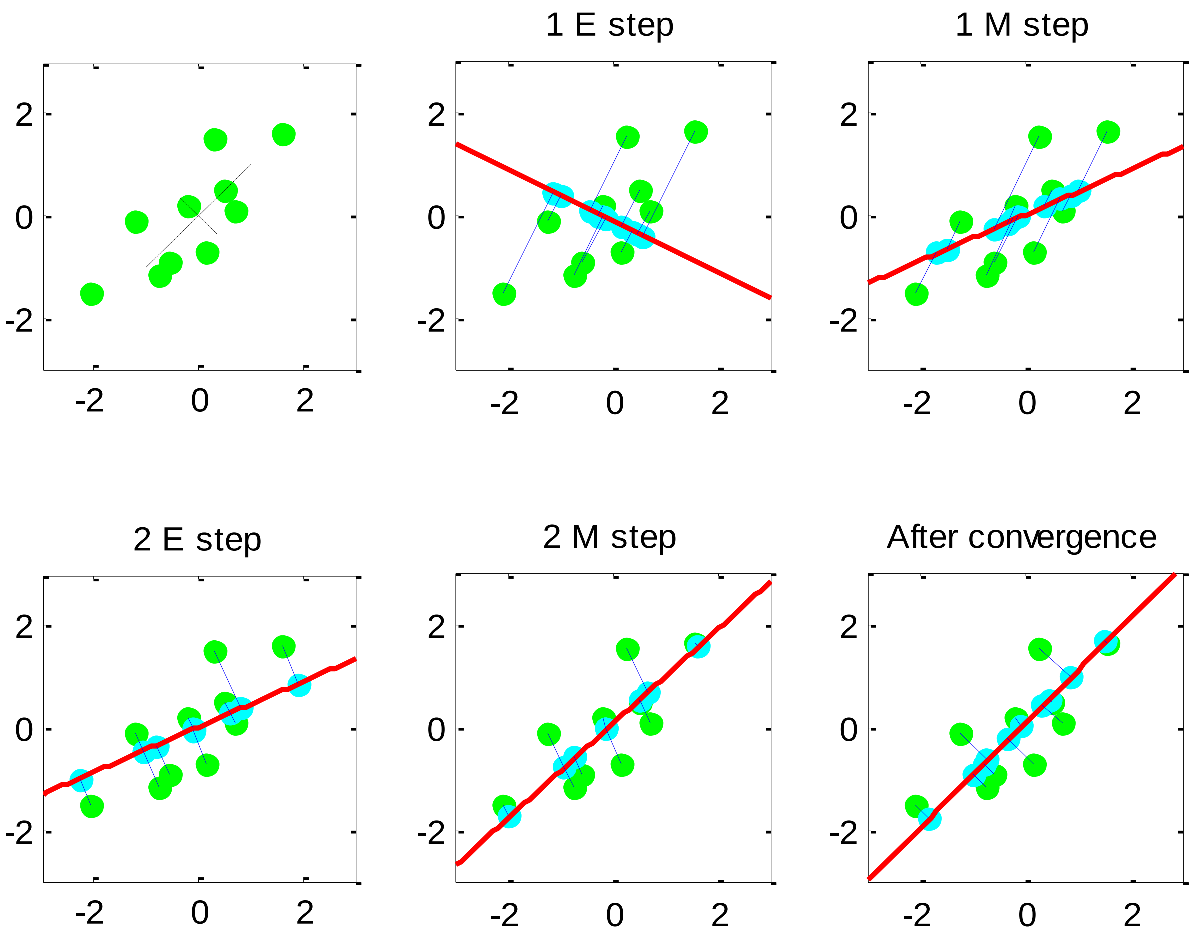
- Synthetic data illustrating the EM algorithm for PCA
  - (d) After the successive E-Step, the values of $\boldsymbol{Z}$ have been updated, giving orthogonal projections, with $\boldsymbol{W}$ held fixed.
  - (e) After the second M-Step.
  - (f) The converged solution.



$$\boldsymbol{\Omega} = \left(\boldsymbol{W}_{old}^T \boldsymbol{W}_{old}\right)^{-1} \boldsymbol{W}_{old}^T \widetilde{\boldsymbol{X}}^T$$

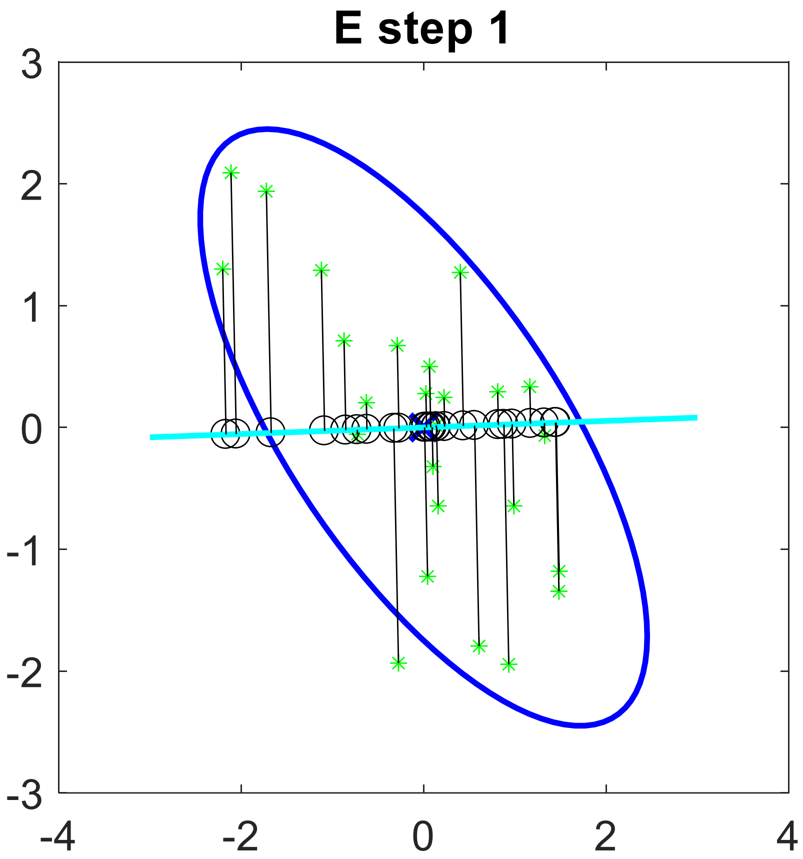
$$\boldsymbol{W}_{new} = \widetilde{\boldsymbol{X}}^T \boldsymbol{\Omega}^T \left(\boldsymbol{\Omega}\boldsymbol{\Omega}^T\right)^{-1}$$

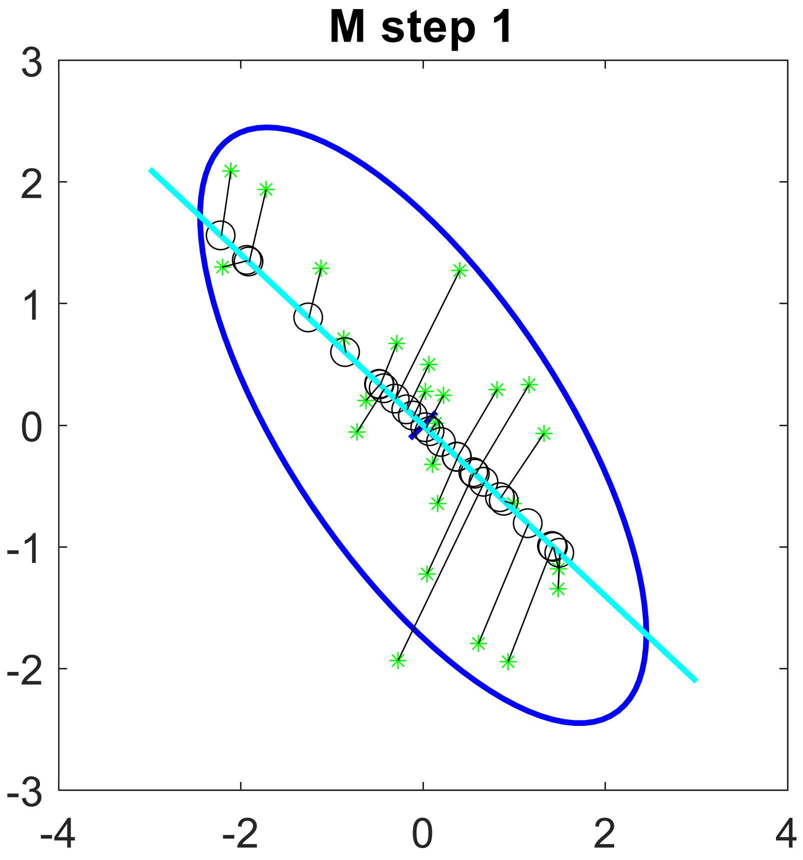
EM Algorithm for PCA: Example, D=2, M=1



Matlab Implementation

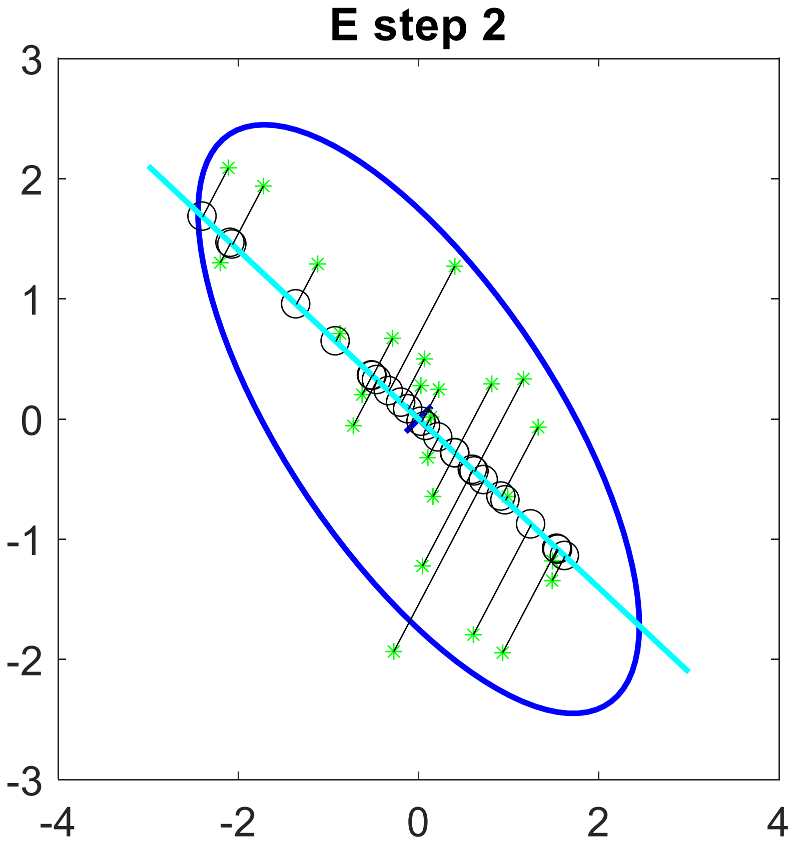
EM Algorithm for PCA: Example, D=2, M=1

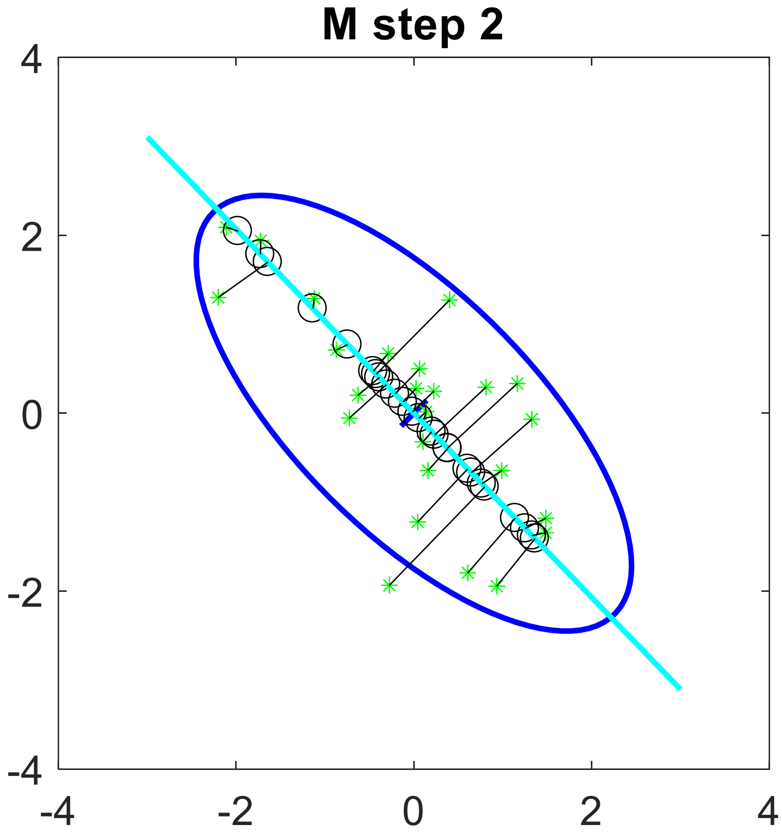




Run pcaEmStepByStep
From PMTK3

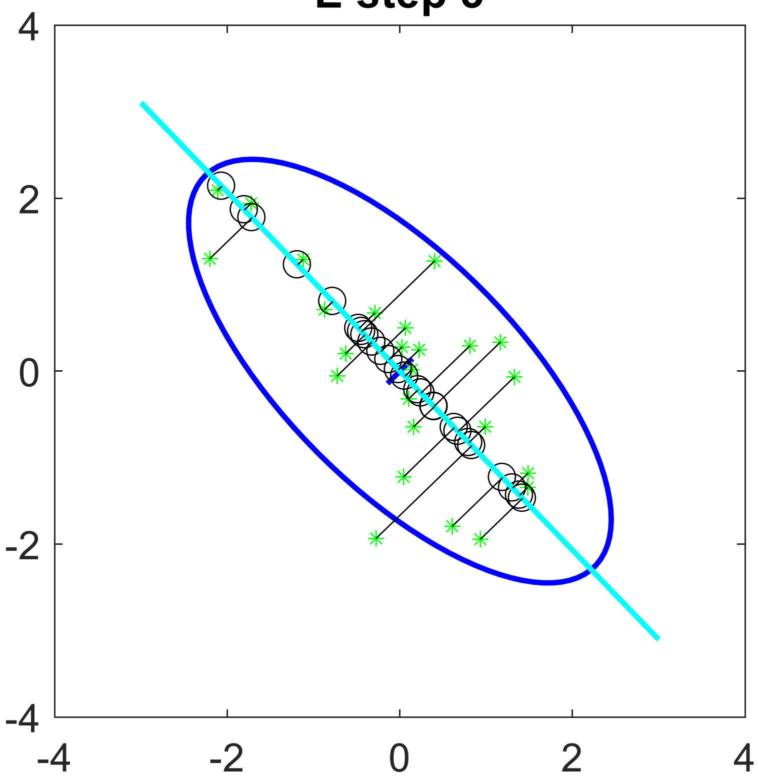
EM Algorithm for PCA: Example, D=2, M=1



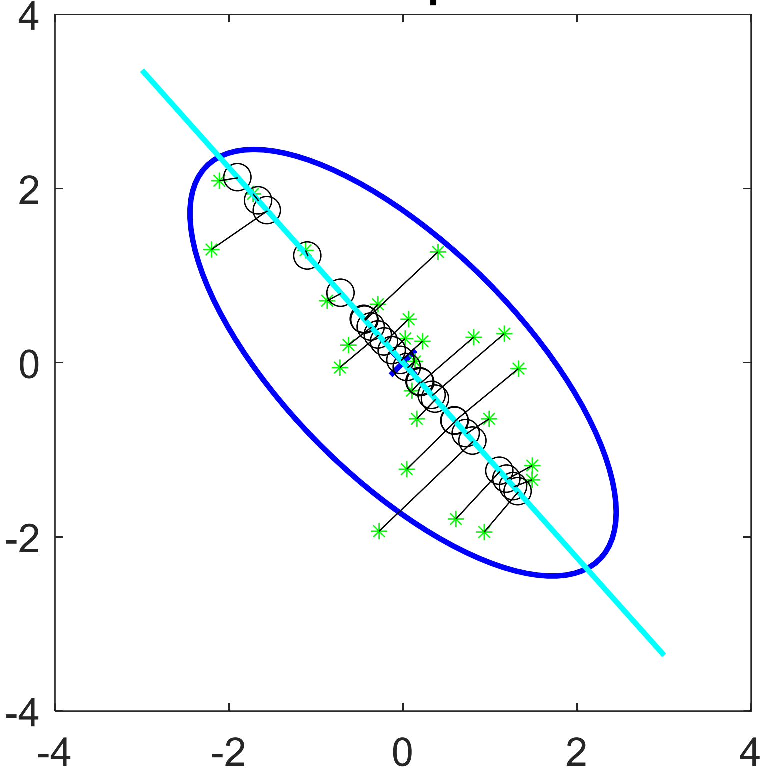


Run pcaEmStepByStep
From PMTK3

EM Algorithm for PCA: Example, D=2, M=1



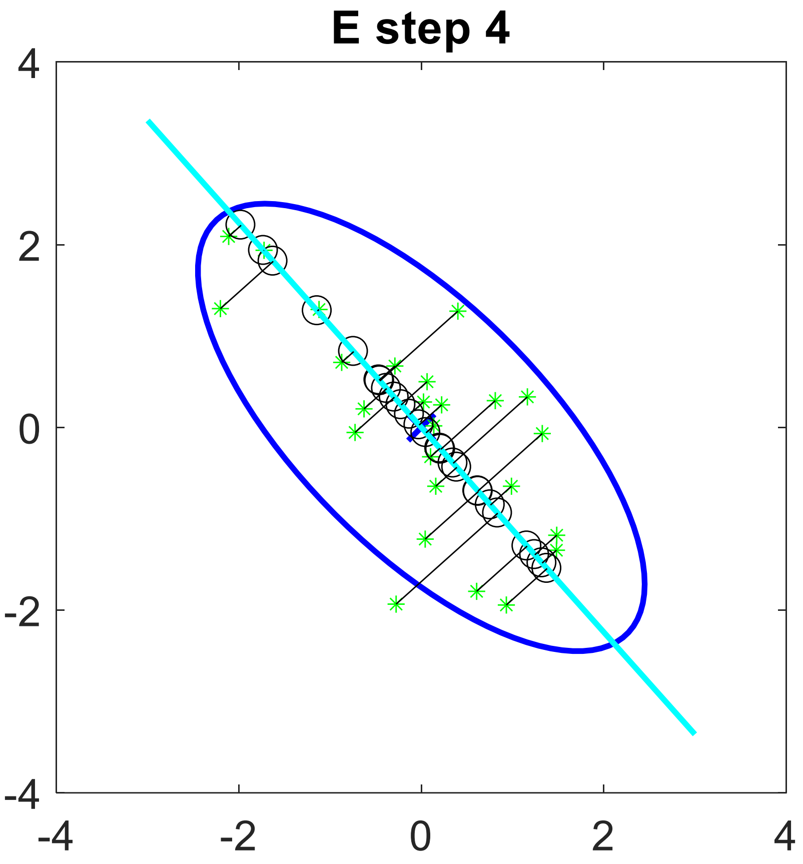
E step 3

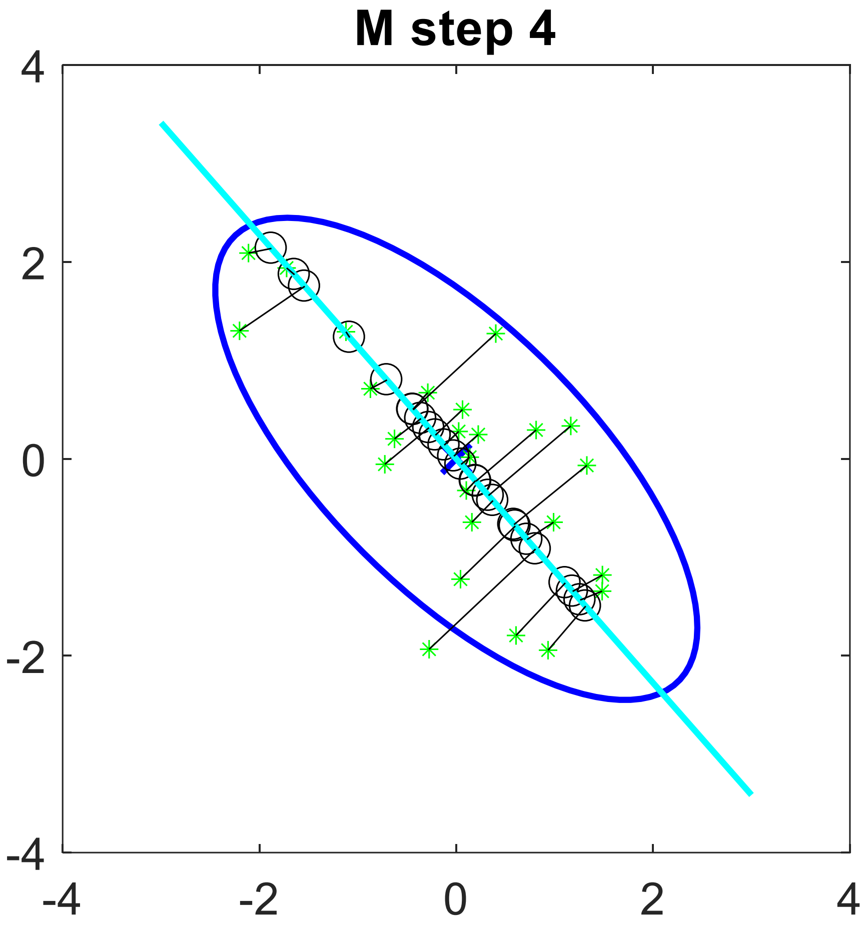


M step 3

Run pcaEmStepByStep
From PMTK3

EM Algorithm for PCA: Example, D=2, M=1





Run pcaEmStepByStep
From PMTK3

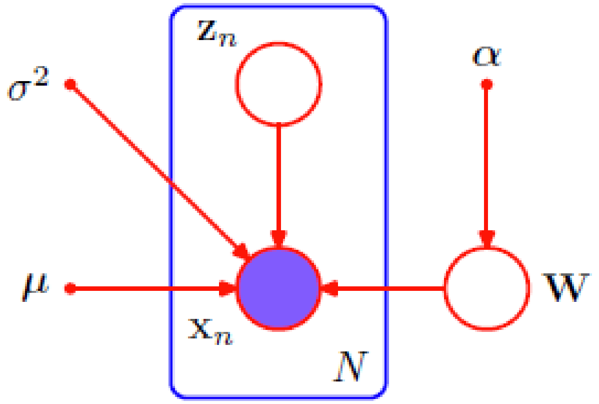
Bayesian PCA

- ❑ It involves a specific choice of prior over $\boldsymbol{W}$ that allows surplus dimensions in the principal subspace to be pruned out of the model (automatic relevance determination, or ARD)

- ❑ Specifically, we define an independent Gaussian prior over each column of $\boldsymbol{W}$, which represent the vectors defining the principal subspace.

- ❑ Each such Gaussian has an independent variance governed by a precision hyperparameter a_i where $\boldsymbol{w}_i$ is the ith column of $\boldsymbol{W}$.

$$p(\boldsymbol{W} \mid a) = \prod_{i=1}^{M} \left(\frac{\alpha_i}{2\pi}\right)^{D/2} \exp\left\{-\frac{1}{2}\alpha_i \boldsymbol{w}_i^T \boldsymbol{w}_i\right\}$$



Bayesian PCA: Effective Dimensionality

- □ Find a_i iteratively by maximizing the marginal likelihood function in which $\boldsymbol{W}$ has been integrated out.

$$p(\boldsymbol{X} \mid \boldsymbol{\alpha}, \mu, \sigma^2) = \int p(\boldsymbol{X} \mid \boldsymbol{W}, \mu, \sigma^2)\, p(\boldsymbol{W} \mid \boldsymbol{\alpha}) d\boldsymbol{W}$$

- □ In the optimization, some of the a_i → ∞, with the corresponding $\boldsymbol{w}_i$ →0 (the posterior distribution becomes a delta function at the origin) giving a sparse solution.

- □ The effective dimensionality of the principal subspace is then the number of finite a_i ’s.The corresponding $\boldsymbol{w}_i$ can be thought of as relevant for modeling the data distribution.

- □ The Bayesian approach is making a trade-off between improving the fit to the data by using a larger number of $\boldsymbol{w}_i$‘s with their corresponding eigenvalues λ_i each tuned to the data, and reducing the complexity of the model by suppressing some of the $\boldsymbol{w}_i$'s.

Maximizing the Marginal Likelihood

- The values of a_i are re-estimated during training by maximizing the marginal likelihood

$$p(\boldsymbol{X}\,|\,\boldsymbol{\alpha},\mu,\sigma^2)=\int p(\boldsymbol{X}\,|\,\boldsymbol{W},\mu,\sigma^2)\,p(\boldsymbol{W}\,|\,\boldsymbol{\alpha})d\boldsymbol{W}$$

where

$$\ln p\left(\boldsymbol{X}\,|\,\boldsymbol{\mu},\boldsymbol{W},\sigma^2\right)=-\frac{ND}{2}\ln\left(2\pi\right)-\frac{N}{2}\ln\left|\boldsymbol{C}\right|-\frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{x}_n-\boldsymbol{\mu})^T\boldsymbol{C}^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu})$$

with

$$\boldsymbol{C}=\boldsymbol{W}\boldsymbol{W}^T+\sigma^2\boldsymbol{I}$$

For simplicity μ and σ^2 are treated as parameters to be estimated, rather than defining priors over these parameters.

Bayesian PCA: Re-estimation Equations

$$p(\boldsymbol{X} \mid \boldsymbol{\alpha}, \mu, \sigma^2) = \int p(\boldsymbol{X} \mid \boldsymbol{W}, \mu, \sigma^2)\, p(\boldsymbol{W} \mid \boldsymbol{\alpha})\, d\boldsymbol{W}$$

- Because this integration is intractable, we make use of the Laplace approximation.

- Assume that the posterior distribution is sharply peaked, as will occur for sufficiently large data sets, then the re-estimation equations obtained by maximizing the marginal likelihood (see earlier lecture notes on model evidence) with respect to α_i take the simple form

$$\alpha_i^{new} = \frac{D}{w_i^T w_i}$$

noting that the dimensionality of $\boldsymbol{w}_i$ is D.

Bayesian PCA: EM Implementation

- These re estimations are interleaved with the EM algorithm updates for determining $\boldsymbol{W}$ and σ^2.

- The E-Step equations are again given as

$$\mathbb{E}\left[\boldsymbol{z}_n\right] = \boldsymbol{M}^{-1}\boldsymbol{W}^T\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right) \qquad \mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right] = \sigma^2\boldsymbol{M}^{-1} + \mathbb{E}\left[\boldsymbol{z}_n\right]\mathbb{E}\left[\boldsymbol{z}_n\right]^T$$

- Similarly, the M-Step equation for σ^2 is again given by

$$\sigma_{new}^2 = \frac{1}{ND}\sum_{n=1}^{N}\left\{\left\|\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right\|^2 - 2\mathbb{E}\left[\boldsymbol{z}_n\right]^T\boldsymbol{W}_{new}^T\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right) + Tr\left(\mathbb{E}\left[\boldsymbol{z}_n\boldsymbol{z}_n^T\right]\boldsymbol{W}_{new}^T\boldsymbol{W}_{new}\right)\right\}$$

- The only change is to the M-Step equation for $\boldsymbol{W}$:

$$W_{new} = \left[\sum_{n=1}^{N}\left(x_n - \overline{x}\right)\mathbb{E}\left[z_n\right]^T\right]\left[\sum_{n=1}^{N}\mathbb{E}\left[z_n z_n^T\right] + \sigma^2\boldsymbol{A}\right]^{-1}, \ \boldsymbol{A} = diag(\alpha_i)$$

- The value of $\boldsymbol{\mu}$ is given by the sample mean, as before.

Bayesian PCA

- ❑ For $M = D - 1$, if all a_i's are finite, the model represents a full-covariance Gaussian.

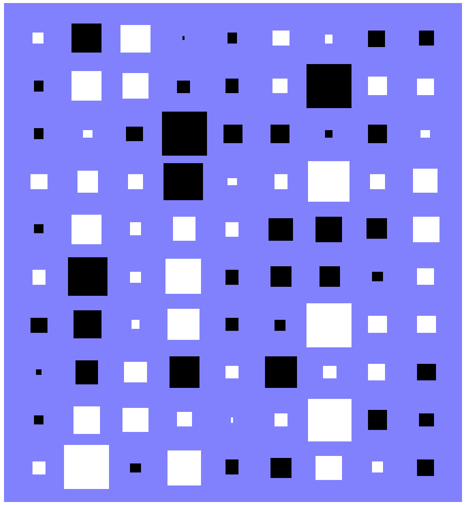
- ❑ For $M = D - 1$, if all a_i →∞ the model is equivalent to an isotropic Gaussian, and so the model can encompass all permissible values for the effective dimensionality of the principal subspace.

- ❑ Using smaller values of M saves on computational cost but limits the maximum dimensionality of the subspace.

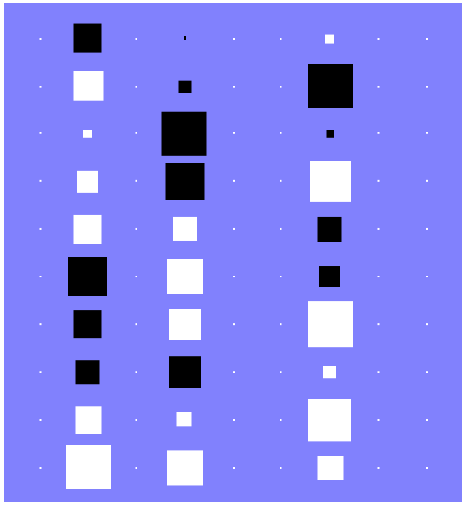
- ❑ A comparison of the Bayesian PCA algorithm with standard probabilistic PCA is shown next.

Bayesian PCA: Hinton Diagrams of W

- 'Hinton' diagrams of the matrix $\boldsymbol{W}$. Each element of the matrix is depicted as a square (white for positive & black for negative values) whose area is proportional to the magnitude of that element.



W:10x9



W:10x3

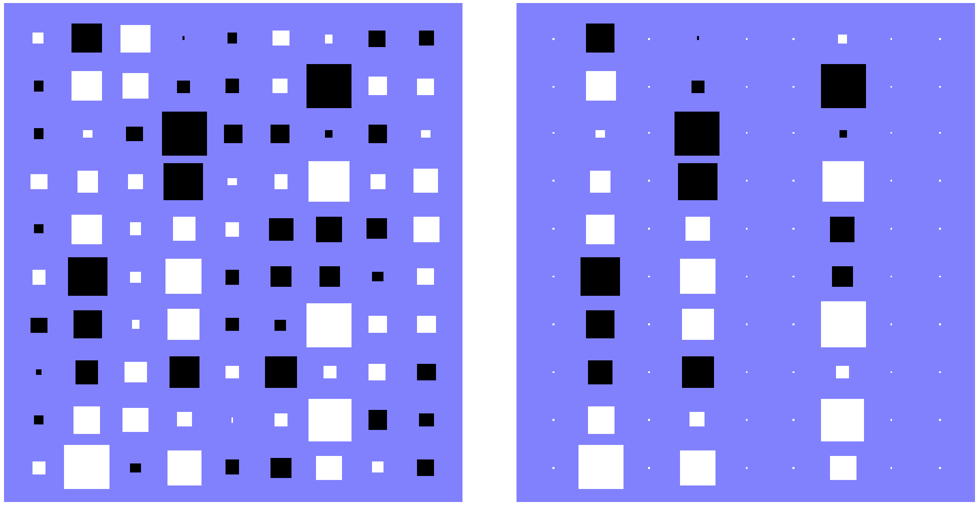
Matlab Implementation

- The synthetic data set comprises 300 data points in $D = 10$ dimensions sampled from a Gaussian distribution having std 1.0 in 3 directions and std 0.5 in the remaining 7 directions.

Bayesian PCA: Hinton Diagrams for W

- Left-hand plot: result from maximum likelihood probabilistic PCA.
- Right-hand plot: corresponding result from Bayesian PCA.
- The Bayesian model discovers the appropriate dimensionality by suppressing the 6 surplus degrees of freedom.

Matlab Implementation



Kernel PCA

- Kernel trick: take an algorithm expressed in terms of scalar products of the form $\boldsymbol{x}^T\boldsymbol{x}$ and generalize it by replacing the scalar products with a nonlinear kernel.

- Apply this technique of kernel substitution to PCA, obtaining a nonlinear algorithm called kernel PCA (Scholkopf et al. 1998)

- Consider a data set $\{\boldsymbol{x}_n\}, n = 1, \ldots, N$, of dimension D. Assume that we have already subtracted the sample mean from each $\boldsymbol{x}_n$, so that $\sum_n \boldsymbol{x}_n = 0$.

- Start by expressing PCA in such a form that the data $\{\boldsymbol{x}_n\}$ appear only in the form of the scalar products $\boldsymbol{x}_n^T \boldsymbol{x}_m$.

- Scholkopf, B., A. Smola, and K.-R. Muller (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* **10**(5), 1299–1319.

Kernel PCA

- Recall that the principal components are defined by the normalized eigenvectors $\boldsymbol{u}_i$ of the $N \times N$ covariance matrix

$$\boldsymbol{S}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i, \ \text{where}: \boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^T$$

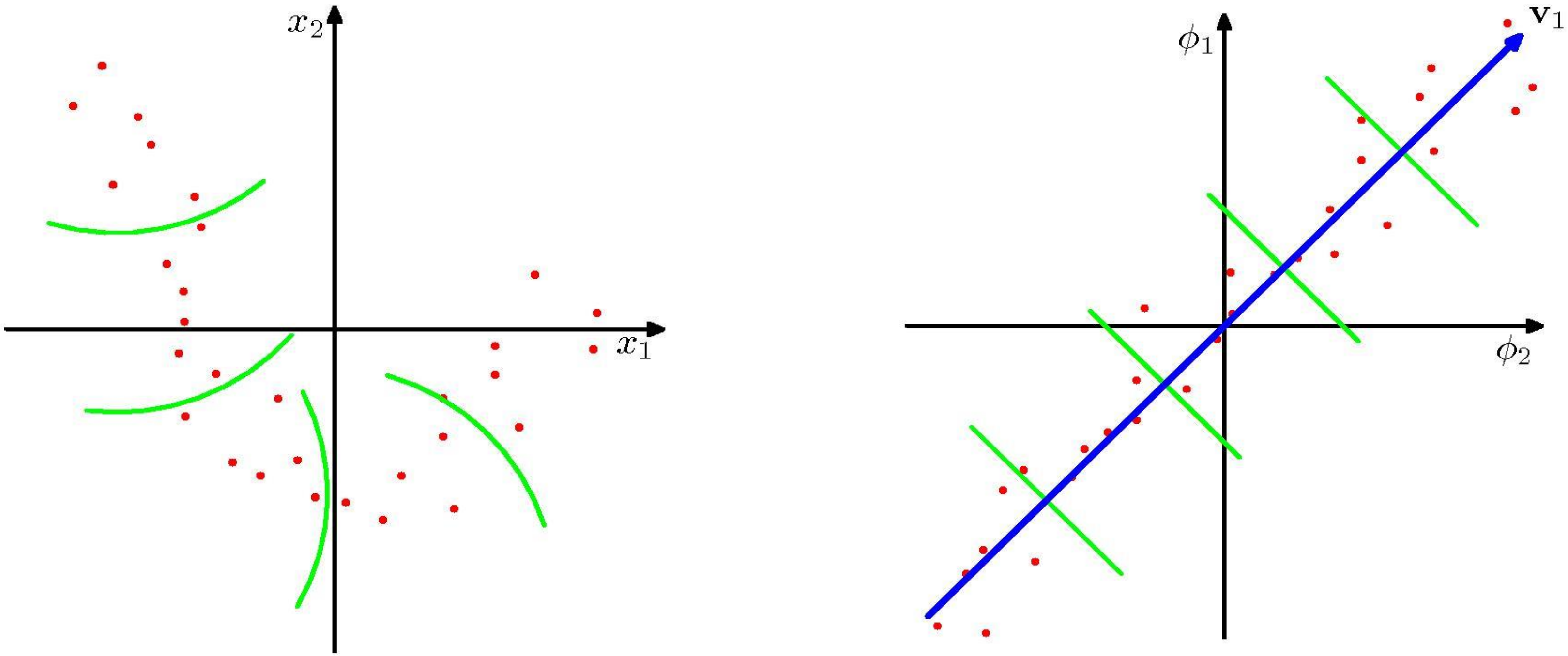
- Consider a nonlinear transformation $\boldsymbol{\phi}(\boldsymbol{x})$ into an M-dimensional feature space, so that each data point $\boldsymbol{x}_n$ is thereby projected onto a point $\boldsymbol{\phi}(\boldsymbol{x}_n)$.

- Now perform standard PCA in the feature space.
    - This implicitly defines a nonlinear principal component model in the original data space.

Kernel PCA

- A data set in the original data space (left) is projected by $\boldsymbol{\phi}(\boldsymbol{x})$ into a feature space (right).

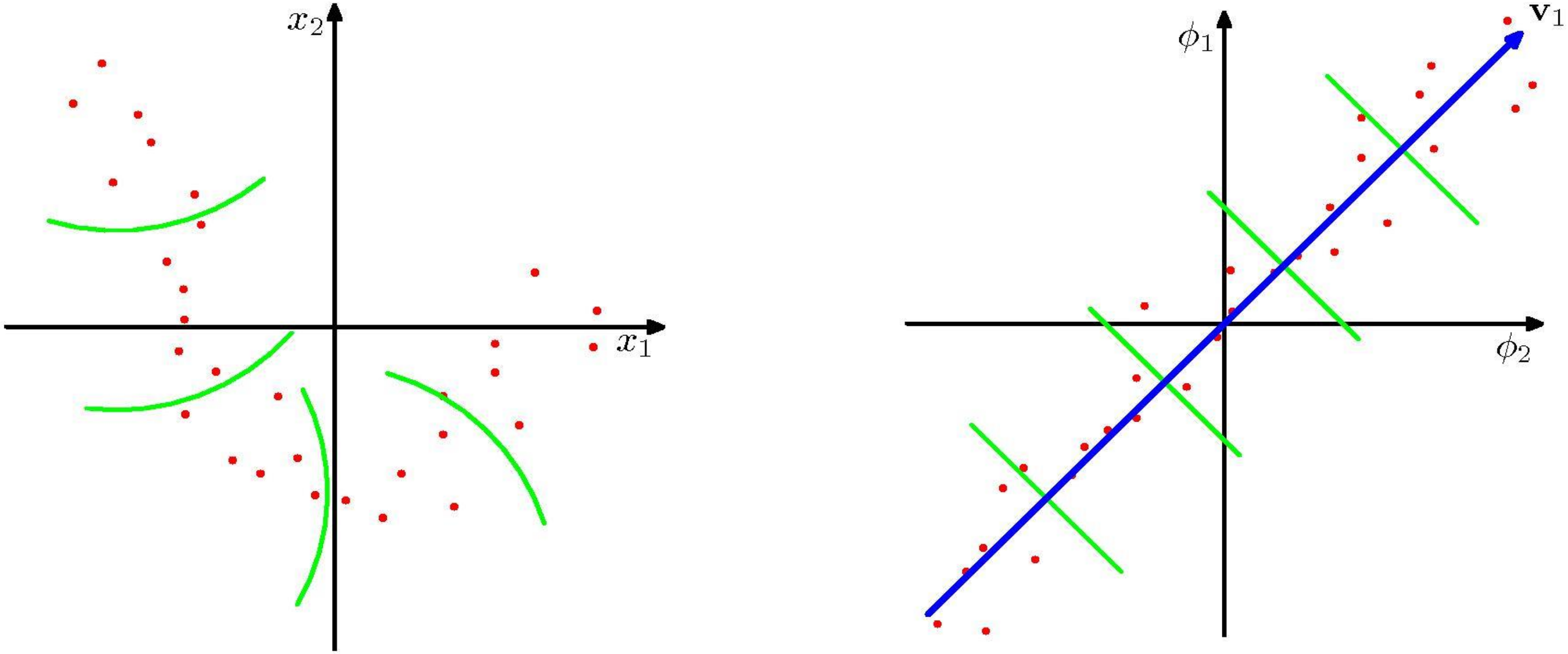
- By performing PCA in the feature space, we obtain the principal components ($\boldsymbol{v}_1$ being the 1st one).



Kernel PCA

- ❑ The green lines in feature space indicate the linear projections onto $\boldsymbol{v}_1$, which correspond to nonlinear projections in the original data space.

- ❑ In general, it is not possible to represent the nonlinear principal component by a vector in $\boldsymbol{x}$ space.



Kernel PCA

- For the moment, let us assume that the projected data set also has zero mean, so that $\sum_{n=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_n)=0$. We shall return to this point shortly.

- The $M \times M$ sample covariance matrix in feature space is given by

$$C=\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}(\boldsymbol{x}_n)^T$$

- The eigenvector expansion is defined as:

$$C\boldsymbol{v}_i=\lambda_i\boldsymbol{v}_i,\ i=1,\ldots,M$$

- Can we solve this eigenvalue problem without having to work explicitly in the feature space?

Kernel PCA

$$C = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}(\boldsymbol{x}_n)^T \qquad C\mathbf{v}_i = \lambda_i\mathbf{v}_i,\, i = 1,\ldots,M$$

- ❑ From the definition of $\boldsymbol{C}$, the eigenvector equations tells us that $\boldsymbol{v}_i$ satisfies

$$\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_n)\left\{\boldsymbol{\phi}(\boldsymbol{x}_n)^T\mathbf{v}_i\right\} = \lambda_i\mathbf{v}_i,\, i = 1,\ldots,M$$

- ❑ We see that (provided $\lambda_i > 0$), the vector $\boldsymbol{v}_i$ is given by a linear combination of the $\boldsymbol{\phi}(\boldsymbol{x}_n)$ and so can be written in the form

$$\mathbf{v}_i = \sum_{n=1}^{N} a_{in}\boldsymbol{\phi}(\boldsymbol{x}_n),\, i = 1,\ldots,M$$

Kernel PCA

$$\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_n)\left\{\boldsymbol{\phi}(\boldsymbol{x}_n)^T\boldsymbol{v}_i\right\}=\lambda_i\boldsymbol{v}_i,\ i=1,\ldots,M$$

- Substituting

$$\boldsymbol{v}_i=\sum_{n=1}^{N}a_{in}\boldsymbol{\phi}(\boldsymbol{x}_n),\ i=1,\ldots,M$$

back into the eigenvector equation, we obtain

$$\left[\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}(\boldsymbol{x}_n)^T\right]\sum_{m=1}^{N}a_{im}\boldsymbol{\phi}(\boldsymbol{x}_m)=\lambda_i\sum_{n=1}^{N}a_{in}\boldsymbol{\phi}(\boldsymbol{x}_n),\ i=1,\ldots,M$$

- The key step is now to express this in terms of the kernel function

$$K_{nm}\equiv k\left(\boldsymbol{x}_n,\boldsymbol{x}_m\right)=\boldsymbol{\phi}(\boldsymbol{x}_n)^T\boldsymbol{\phi}(\boldsymbol{x}_m)$$

which we do by multiplying both sides by $\boldsymbol{\phi}(\boldsymbol{x}_l)^T$.

Kernel PCA

$$\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}(\boldsymbol{x}_n)^T\sum_{m=1}^{N}a_{im}\boldsymbol{\phi}(\boldsymbol{x}_m)=\lambda_i\sum_{n=1}^{N}a_{in}\boldsymbol{\phi}(\boldsymbol{x}_n),\ i=1,\ldots,M$$

- Multiplying both sides by $\boldsymbol{\phi}(\boldsymbol{x}_l)^T$ to give

$$\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_l)^T\boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}(\boldsymbol{x}_n)^T\sum_{m=1}^{N}a_{im}\boldsymbol{\phi}(\boldsymbol{x}_m)=\lambda_i\sum_{n=1}^{N}a_{in}\boldsymbol{\phi}(\boldsymbol{x}_l)^T\boldsymbol{\phi}(\boldsymbol{x}_n),\ i=1,\ldots,M\Rightarrow$$

$$\frac{1}{N}\sum_{n=1}^{N}k\left(\boldsymbol{x}_l,\boldsymbol{x}_n\right)\sum_{m=1}^{N}a_{im}k\left(\boldsymbol{x}_n,\boldsymbol{x}_m\right)=\lambda_i\sum_{n=1}^{N}a_{in}k\left(\boldsymbol{x}_l,\boldsymbol{x}_n\right),\ i=1,\ldots,M\Rightarrow$$

- This can be written in a matrix form as:

$$\boldsymbol{K}^2\boldsymbol{a}_i=\lambda_i N\boldsymbol{K}\boldsymbol{a}_i$$

$\boldsymbol{a}_i$ *N*-dimensional column vector with elements $a_{in}, n = 1,..,N$.

- Can simplify $\boldsymbol{K}\boldsymbol{a}_i=\lambda_i N\boldsymbol{a}_i$. These Eqs. differ by eigen-vectors of $\boldsymbol{K}$ with 0 eigenvalues that don't affect the principal components projection.

Kernel PCA

$$\boldsymbol{K}\boldsymbol{a}_i = \lambda_i N \boldsymbol{a}_i$$

- We can find solutions for $\boldsymbol{a}_i$ by solving the above eigenvalue problem.
- The normalization condition for the coefficients $\boldsymbol{a}_i$ is obtained by requiring that the eigenvectors in feature space be normalized.
- Using

and

$$\boldsymbol{v}_i = \sum_{n=1}^{N} a_{in} \boldsymbol{\phi}(\boldsymbol{x}_n), \; i = 1, \ldots, M$$

we have:

$$\boldsymbol{K}\boldsymbol{a}_i = \lambda_i N \boldsymbol{a}_i$$

$$1 = \boldsymbol{v}_i^T \boldsymbol{v}_i = \frac{1}{N} \sum_{n=1}^{N} \sum_{m=1}^{N} a_{in} a_{im} \boldsymbol{\phi}(\boldsymbol{x}_n)^T \boldsymbol{\phi}(\boldsymbol{x}_m) = \boldsymbol{a}_i^T \boldsymbol{K}\boldsymbol{a}_i = \lambda_i N \boldsymbol{a}_i^T \boldsymbol{a}_i \Rightarrow \quad 1 = \lambda_i N \boldsymbol{a}_i^T \boldsymbol{a}_i$$

Kernel PCA

- The principal component projections can now be cast in terms of the kernel function.

- Using

$$\boldsymbol{v}_i = \sum_{n=1}^{N} a_{in} \boldsymbol{\phi}(\boldsymbol{x}_n),\ i = 1, \ldots, M$$

the projection of a point $\boldsymbol{x}$ onto eigenvector i is given by

$$y_i(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{v}_i \ = \sum_{n=1}^{N} a_{in} \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x}_n) = \sum_{n=1}^{N} a_{in} k(\boldsymbol{x}, \boldsymbol{x}_n)$$

Kernel PCA

- In the original D-dimensional $\boldsymbol{x}$ space: D orthogonal eigenvectors and thus at most D linear principal components.
- The dimensionality M of the feature space, however, can be much larger than D (even infinite), and thus we can find a number of nonlinear principal components that can exceed D.
- Note, however, that the number of nonzero eigenvalues cannot exceed the number N of data points:
  - The covariance matrix in feature space has rank at most equal to N.
  - Recall that kernel PCA involves the eigenvector expansion of the $N \times N$ matrix $\boldsymbol{K}$.

Kernel PCA

- ❑ So far we have assumed that the projected data set given by $\boldsymbol{\phi}(\boldsymbol{x}_n)$ has zero mean, which in general will not be the case.

- ❑ We cannot simply compute and then subtract off the mean, since we want to formulate the algorithm purely in terms of the kernel function.

- ❑ The projected data points after centralizing are given by

$$\boldsymbol{\phi}(\boldsymbol{x}_n) = \boldsymbol{\phi}(\boldsymbol{x}_n) - \frac{1}{N}\sum_{l=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_l)$$

- ❑ The corresponding elements of the Gram matrix $\widetilde{\boldsymbol{K}}_{nm} = \widetilde{\boldsymbol{\phi}}(x_n)^T\widetilde{\boldsymbol{\phi}}(x_m)$ can be computed as shown next.

Kernel PCA

$$\widetilde{\boldsymbol{K}}_{nm} = \widetilde{\boldsymbol{\phi}}(x_n)^T\widetilde{\boldsymbol{\phi}}(x_m) \;= \left(\boldsymbol{\phi}(\boldsymbol{x}_n) - \frac{1}{N}\sum_{l=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_l)\right)^T \left(\boldsymbol{\phi}(\boldsymbol{x}_m) - \frac{1}{N}\sum_{l=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_l)\right)$$

$$= \boldsymbol{\phi}(\boldsymbol{x}_n)^T\boldsymbol{\phi}(\boldsymbol{x}_m) - \frac{1}{N}\sum_{l=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_n)^T\boldsymbol{\phi}(\boldsymbol{x}_l)$$

$$-\frac{1}{N}\sum_{l=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_l)^T\boldsymbol{\phi}(\boldsymbol{x}_m) + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{l=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_j)^T\boldsymbol{\phi}(\boldsymbol{x}_l)$$

$$= k(\boldsymbol{x}_n,\boldsymbol{x}_m) - \frac{1}{N}\sum_{l=1}^{N}k(\boldsymbol{x}_n,\boldsymbol{x}_l) - \frac{1}{N}\sum_{l=1}^{N}k(\boldsymbol{x}_l,\boldsymbol{x}_m) + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{l=1}^{N}k(\boldsymbol{x}_j,\boldsymbol{x}_l)$$

- We can express this in matrix notation as:

$$\widetilde{\boldsymbol{K}} = \boldsymbol{K} - \boldsymbol{1}_N\boldsymbol{K} - \boldsymbol{K}\boldsymbol{1}_N + \boldsymbol{1}_N\boldsymbol{K}\boldsymbol{1}_N$$

where $\boldsymbol{1}_N$ denotes the $N \times N$ matrix in which every element takes the value $1/N$.

Kernel PCA

$$\widetilde{\boldsymbol{K}} = \boldsymbol{K} - \boldsymbol{1}_N\boldsymbol{K} - \boldsymbol{K}\boldsymbol{1}_N + \boldsymbol{1}_N\boldsymbol{K}\boldsymbol{1}_N$$

- ❑ Thus we can evaluate $\widetilde{\boldsymbol{K}}$ using only the kernel function and then use $\widetilde{\boldsymbol{K}}$ to determine the eigenvalues and eigenvectors.

- ❑ Note that the standard PCA algorithm is recovered as a special case if we use a linear kernel $k(\boldsymbol{x},\boldsymbol{x}') = \boldsymbol{x}^T\boldsymbol{x}'$.

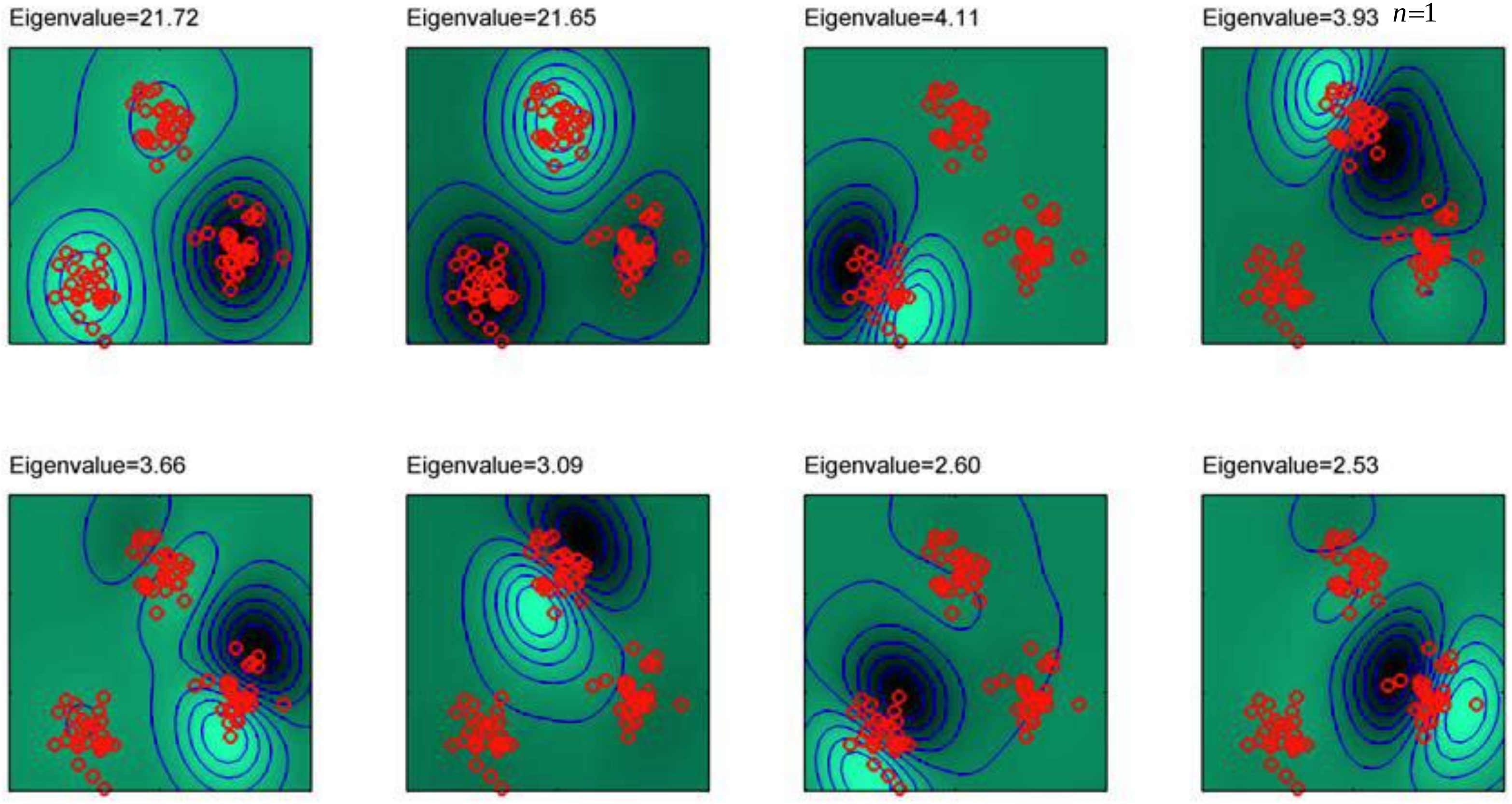
- ❑ We often use Gaussian kernels of the form:

$$k(\boldsymbol{x},\boldsymbol{x}') = \exp\left(-\frac{\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|}{2\sigma^2}\right)$$

Kernel PCA

- KPCA, with a Gaussian kernel ($2\sigma^2 = 0.1$) applied to a synthetic data in 2D, shows the first 8 eigenfunctions and eigenvalues.
- The lines along which the projection onto the corresponding principal component is constant are shown:

$$\boldsymbol{\phi}(\mathbf{x})^T \mathbf{v}_i = \sum_{n=1}^{N} a_{in} k(\mathbf{x}, \mathbf{x}_n)$$



Eigenvalue=21.72

Eigenvalue=21.65

Eigenvalue=4.11

Eigenvalue=3.93

Eigenvalue=3.66

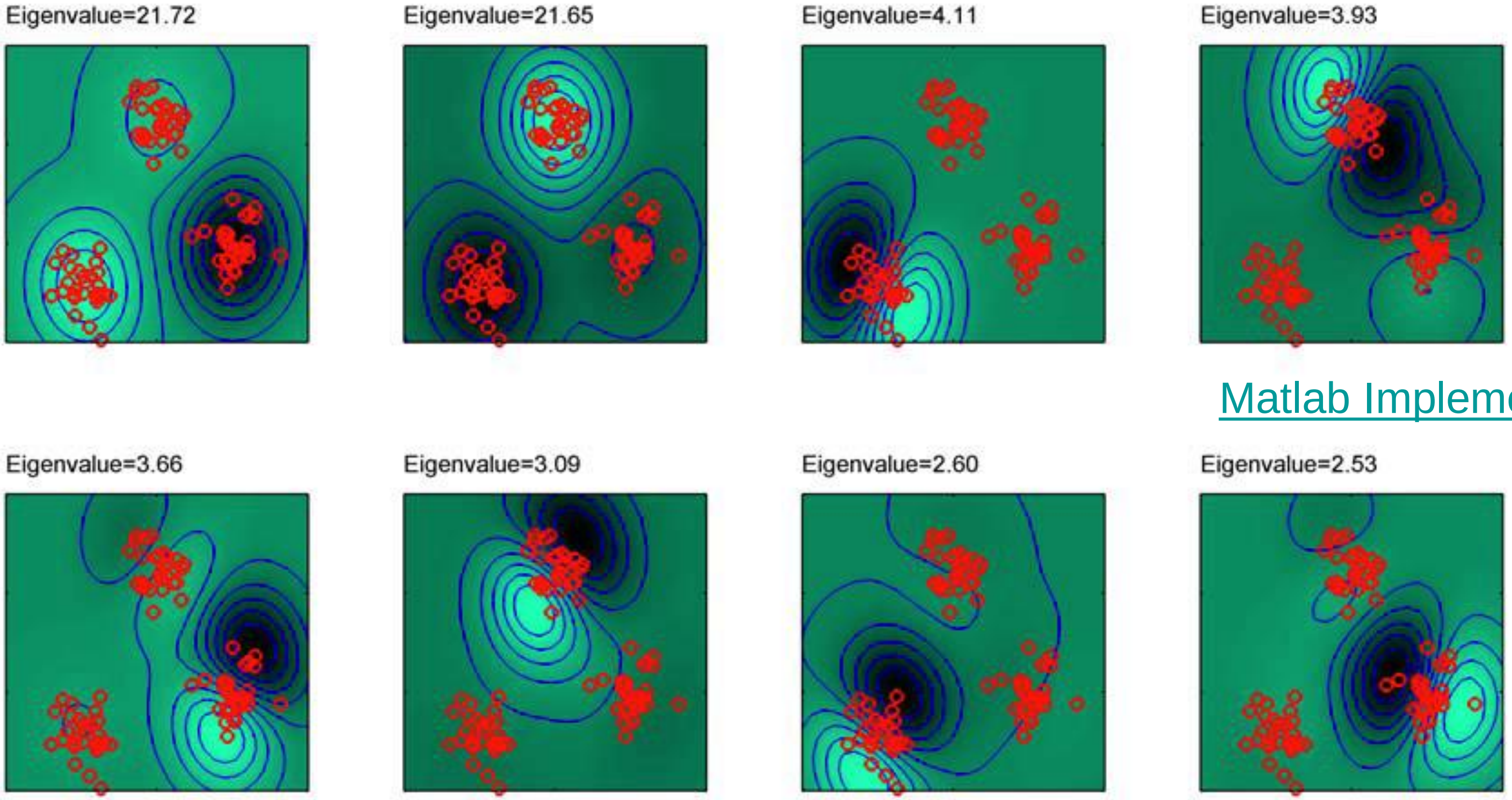
Eigenvalue=3.09

Eigenvalue=2.60

Eigenvalue=2.53

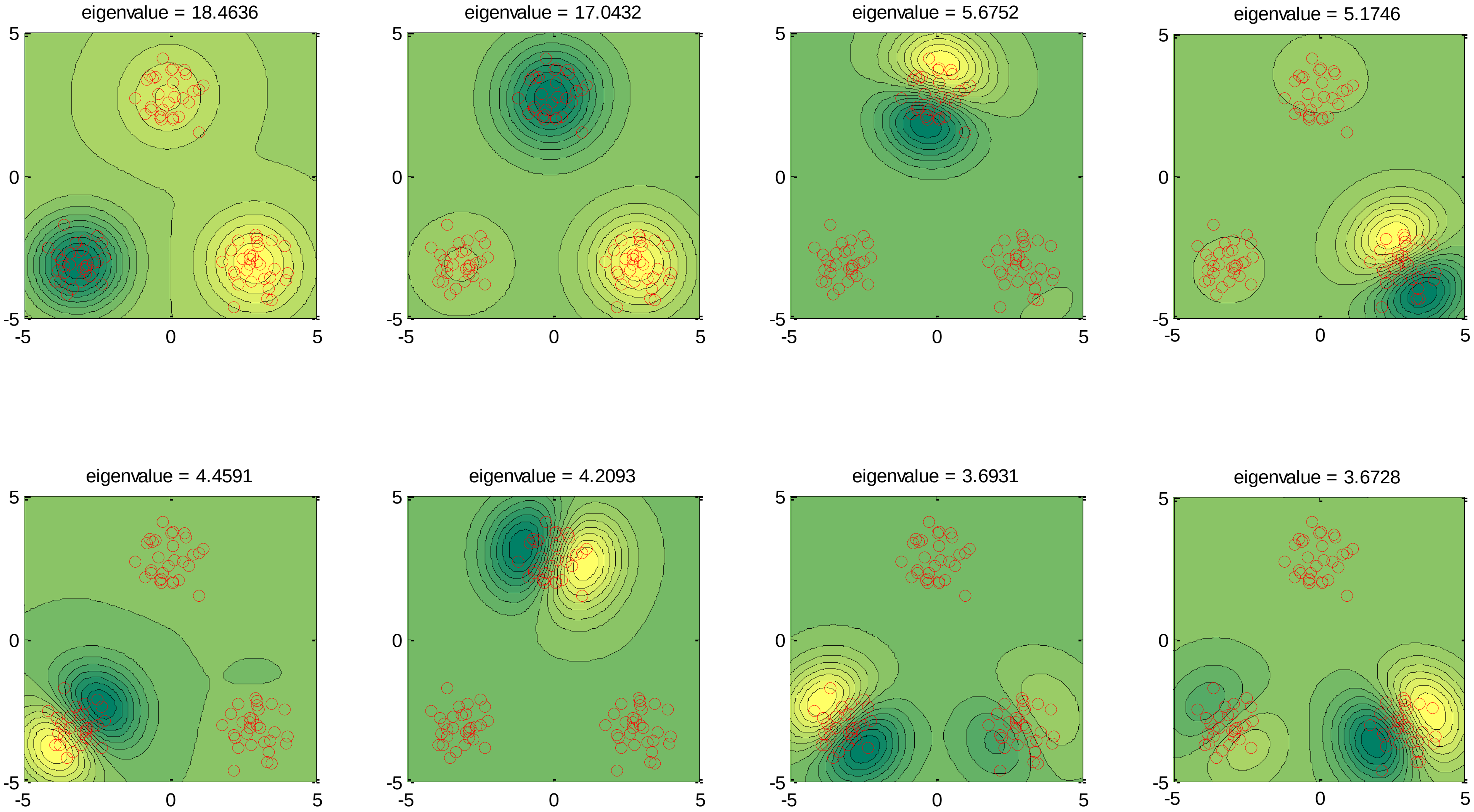
Kernel PCA

- Note the first 2 eigenvectors separate the 3 clusters, the next three eigenvectors split each of the cluster into halves, and the following three eigenvectors again split the clusters into halves along directions orthogonal to the previous splits.



Matlab Implementation

Kernel PCA



eigenvalue = 18.4636

eigenvalue = 17.0432

eigenvalue = 5.6752

eigenvalue = 5.1746

eigenvalue = 4.4591

eigenvalue = 4.2093

eigenvalue = 3.6931

eigenvalue = 3.6728

Matlab Implementation

Kernel PCA: Disadvatages

- KPCA involves finding the eigenvectors of the $N \times N$ matrix $\widetilde{\boldsymbol{K}}$ rather than the $D \times D$ matrix $\boldsymbol{S}$ of PCA.
  - In practice, for large N approximations are used.
- In PCA, we retain some reduced number $L < D$ of eigenvectors and then approximate a data vector $\boldsymbol{x}_n$ by its projection $\widehat{\boldsymbol{x}}_n$ onto the L-dimensional principal subspace, defined by

$$\widehat{\boldsymbol{x}}_n = \sum_{i=1}^{L} (\boldsymbol{x}_n^T \boldsymbol{u}_i)\, \boldsymbol{u}_i$$

- In KPCA, this in general is not possible.

Kernel PCA: Preimage problem

$$\widehat{\boldsymbol{x}}_n = \sum_{i=1}^{L} (\boldsymbol{x}_n^T \boldsymbol{u}_i)\, \boldsymbol{u}_i$$

- This is not possible in KPCA. Indeed note that the mapping $\boldsymbol{\phi}(\boldsymbol{x})$ maps the D-dimensional $\boldsymbol{x}$ space into a D-dimensional manifold in the M-dimensional feature space $\boldsymbol{\phi}$.

- The vector $\boldsymbol{x}$ is known as the pre-image of $\boldsymbol{\phi}(\boldsymbol{x})$. The projection of points in feature space onto the linear PCA subspace in that space will typically not lie on the nonlinear D-dimensional manifold and so will not have a corresponding pre-image in data space.

- Techniques have been proposed for finding approximate pre-images (Bakir et al., 2004)
  - Bakir, G. H., J. Weston, and B. Scholkopf (2004). Learning to find pre-images. In S. Thrun, L. K. Saul, and B. Scholkopf (Eds.), *Advances in Neural Information Processing Systems*, Volume 16, pp. 449–456. MIT Press.