Principal Component Analysis

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

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

November 7, 2017

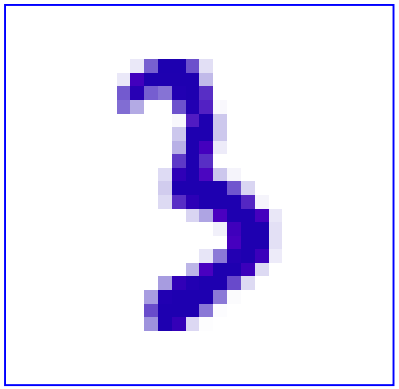
Contents

- 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 $\sigma^2 \rightarrow 0$, Model Selection

Following closely Chris Bishops' PRML book (Chapter 12)

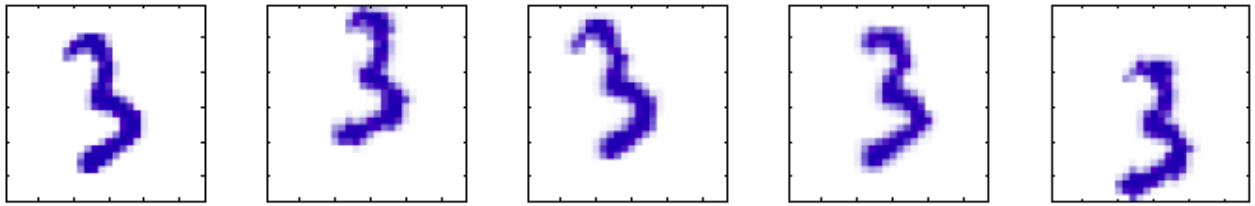
Continuous Latent Variables

- In many data sets, the data points lie close to a manifold of much lower dimensionality than that of the original data space.
- Consider a data set constructed by taking one of the off-line digits, represented by a 64×64 pixel grey-level image.
- Embed this in a larger image of size 100×100 by padding with pixels having the value zero (white pixels).
- Each image is represented by a point in $10{,}000$ dimensional space.



Continuous Latent Variables

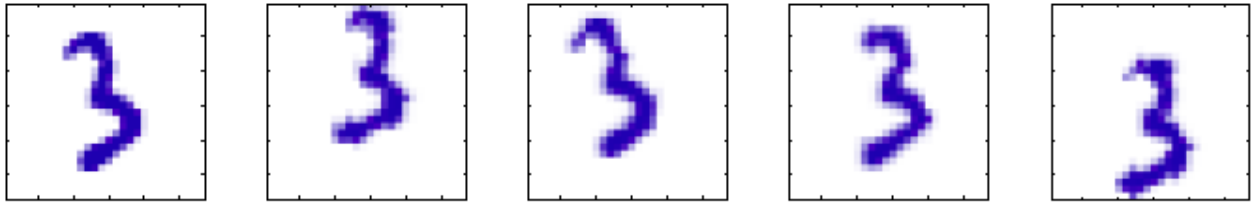
- We create multiple copies in which the location and orientation of the digit is varied at random.



- In this data set, there are 3 DOF of variability:
  - Vertical displacement
  - Horizontal displacement and
  - Rotation

Intrinsic Dimensionality of a Data Set

- The intrinsic dimensionality of the data set is three (vertical and horizontal displacement and rotation)



- The manifold is non-linear: when translating the digit past a particular pixel, that pixel value goes from 0 (white) to 1 (black) and back to zero again. This is a nonlinear function of the digit position.

- The translation and rotation parameters are latent variables: we observe only the image vectors without knowing the translation or rotation variables used to create them.

Other Latent Variables

- For real digit image data, additional DOF arise from
  - Scaling
  - Complex deformations due to the variability of an individual's writing style,
  - Etc.
- The number of such degrees of freedom will still be small compared to the dimensionality of the data set.
- For data compression, we are interested to explore the manifold structure.

Generative Point of View

- The data points are often not confined precisely to a smooth low-dimensional manifold.
- The departure of data points from the manifold is interpreted as 'noise'.
- **Generative view:**
  - (a) Select a point within the manifold according to some latent variable distribution $p(\boldsymbol{z})$ and
  - (b) Generate an observed data point by adding noise, drawn from some conditional distribution of the data variables given the latent variables $p(\boldsymbol{x}|\boldsymbol{z})$.

Generative Point of View

- **Example – Linear Gaussian latent variable model:**
  - (a) assume Gaussian distributions for the latent variables $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I})$ and the observed variables $p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2\boldsymbol{I})$
  - (b) make use of a linear-Gaussian dependence of the observed variables on the state of the latent variables, $p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2\boldsymbol{I})$
- This leads to a probabilistic formulation of principal component analysis (PCA).

Generative Point of View

Example – Factor Analysis:

- (a) assume Gaussian distributions for the latent variables

$$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma_0})$$

and the observed variables

$$p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi}),$$
$$\boldsymbol{x}_i \in \mathbb{R}^D, \boldsymbol{z}_i \in \mathbb{R}^M, \boldsymbol{W} \in \mathbb{R}^{D\times M}, \boldsymbol{\Psi} \in \mathbb{R}^{D\times D}$$

- (b) We take $\boldsymbol{\Psi}$ to be diagonal. This overall model is called **factor analysis** or **FA**.

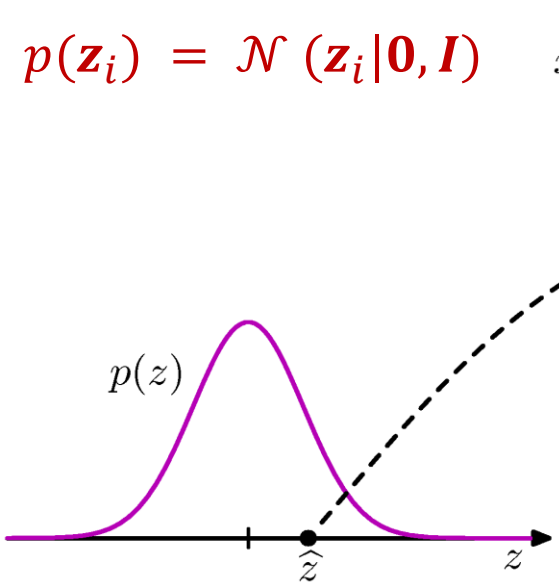
- The special case in which $\boldsymbol{\Psi} = \sigma^2\boldsymbol{I}$ is called **probabilistic principal components analysis** or **PPCA**.

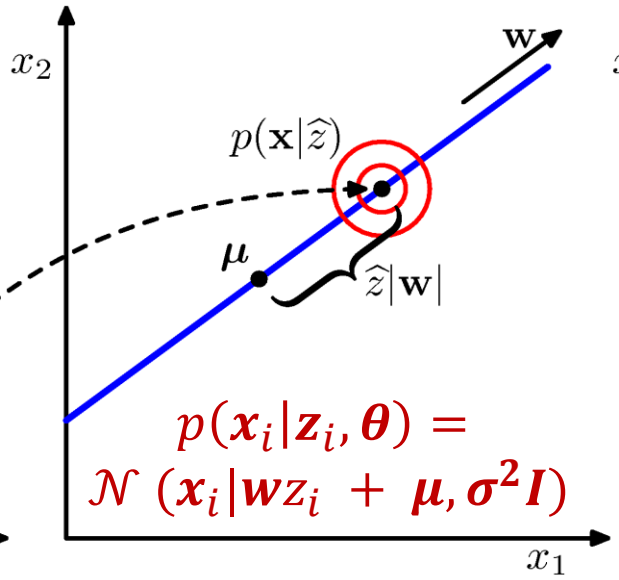
Generative Point of View

- PPCA is illustrated for $D = 2, M = 1$ (latent dimension).

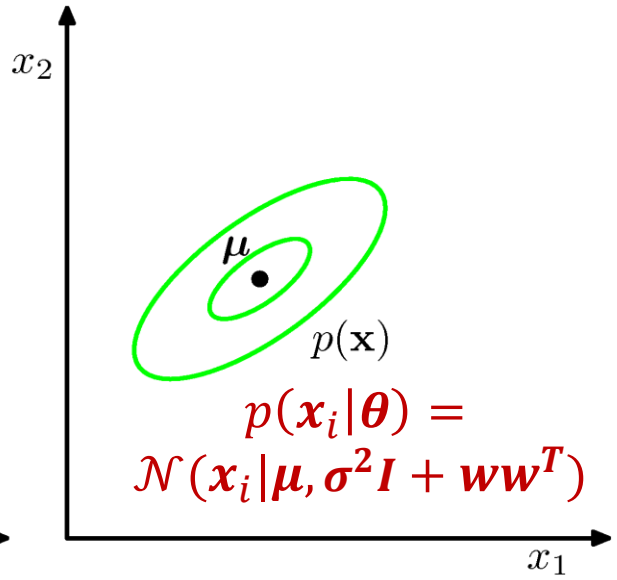
$$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \int \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi})\mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)d\boldsymbol{z}_i = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \boldsymbol{\Psi} + \boldsymbol{W}\boldsymbol{\Sigma}_0\boldsymbol{W}^T)$$

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





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



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

$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ $p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi})$ $p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \boldsymbol{\Psi} + \boldsymbol{W}\boldsymbol{\Sigma}_0\boldsymbol{W}^T)$

- Based on the obtained marginal, we see that without sacrificing generality, we can simplify as:

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

$p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}, \boldsymbol{\Psi} + \boldsymbol{W}\boldsymbol{W}^T)$

Inference on the Latent Factors

- Using $p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi})$, $p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, Bayes rule and standard eqs. for linear Gaussian systems, we can derive the posterior of the latent variables:

$$p(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{m}_i, \boldsymbol{\Sigma}_i),$$
$$\boldsymbol{m}_i = \boldsymbol{\Sigma}\left(\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}) + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right)$$
$$\boldsymbol{\Sigma} = \left(\boldsymbol{\Sigma}_0^{-1} + \boldsymbol{W}^{-1}\boldsymbol{\Psi}^{-1}\boldsymbol{W}\right)^{-1}$$

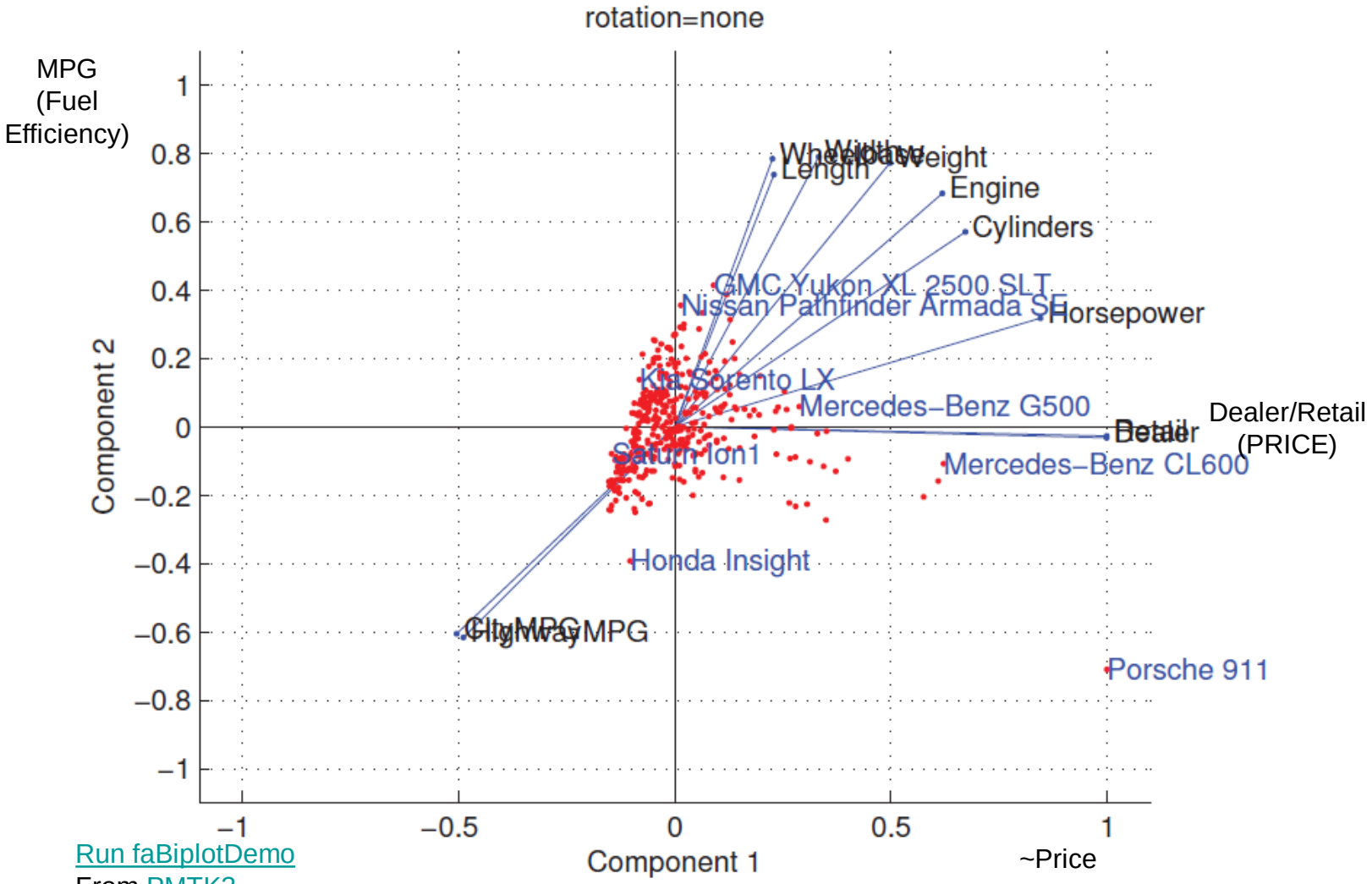
- Computing $\boldsymbol{\Sigma}$ takes $\mathcal{O}(M^3 + M^2D)$ time, and computing each $\boldsymbol{m}_i = \mathbb{E}[\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta}]$ takes $\mathcal{O}(M^2 + MD)$.

- The $\boldsymbol{m}_i$ are sometimes called the latent **scores**, or latent **factors**.

Example Based on Factor Analysis

- Consider a dataset of $D = 11$ variables and $N = 387$ cases describing aspects of cars (engine size, # of cylinders, MPG, etc.).
- We fit a $M = 2$ dimensional model. We plot the $\boldsymbol{m}_i = \mathbb{E}[\boldsymbol{z}_i | \boldsymbol{x}_i, \boldsymbol{\theta}]$ scores as points in $\mathbb{R}^2$, to visualize the data.
- We also project unit vectors corresponding to each of the feature dimensions, $\boldsymbol{e}_1 = (1, 0, \ldots, 0), \boldsymbol{e}_2 = (0, 1, 0, \ldots, 0)$, etc. into the low dimensional space (blue lines).
- We see that the horizontal axis represents price corresponding to the features labeled “dealer” and “retail” (expensive cars on the right).
- The vertical axis represents fuel efficiency (in terms of MPG) versus size: heavy vehicles are less efficient and are higher up, whereas light vehicles are lower down.
- Unfortunately, interpreting latent variable models is fraught with difficulties.

2D Projection of Car Data



Run faBiplotDemo
From PMTK3

Unidentifiability

- Consider an arbitrary rotation such that: $\widetilde{\boldsymbol{W}} = \boldsymbol{WR}$.
- We can immediately see that the likelihood function $p(\boldsymbol{x}_i|\boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{x}_i \middle| \boldsymbol{\mu}, \boldsymbol{\Psi} + \boldsymbol{WW}^{\boldsymbol{T}}\right)$ remains the same:

$$\widetilde{\boldsymbol{W}}\,\widetilde{\boldsymbol{W}}^{T} = \boldsymbol{WRR}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}} = \boldsymbol{WW}^{\boldsymbol{T}}$$

- We need to remove $M(M-1)/2$ DOF since that is the number of orthonormal matrices of size $M \times M$.
- Then the FA model has $D + MD - M(M-1)/2$ free parameters (excluding the mean), where the first term arises from $\boldsymbol{\Psi}$. This needs to be $\leq D(D+1)/2$, i.e. the number of parameters in an unconstrained symmetric covariance matrix.
- This gives us an upper bound on *M*: $M_{max} = \lfloor D + 0.5\left(1 -$

Unidentifiability

To address the unidentifiability the following options are available:

- Making $\boldsymbol{W}$ orthonormal: This is the case with PCA
  - columns arranged in order of decreasing variance
- Making $\boldsymbol{W}$ lower triangular: The first feature is then generated only by the 1st latent variable, the second by the first two, etc.
  - The number of parameters is equal to M_{max}.
  - One needs to properly select the first M visible variables as they affect the latent factors.

- Lopes, H. and M. West (2004). Bayesian model assessment in factor analysis. *Statisica Sinica 14*, 41– 67.

Unidentifiability

- Sparsity promoting priors on the weights: Use sparse factor analysis that forces some entries in $\boldsymbol{W}$ to zero. They may include $\ell_1$regularization, automatic relevance determination or spike and slab priors. But note that this does not ensure a unique MAP estimate.

- Choosing an informative rotation matrix $\boldsymbol{R}$: Choosing appropriate $\boldsymbol{R}$ to increase interpretable sparse $\boldsymbol{W}$.

- Using non-Gaussian priors for the latent factors in $p(\boldsymbol{z}_i)$: This leads to Independent Component Analysis (ICA).

- Zou, H. (2006). The adaptive Lasso and its oracle properties. *J. of the Am. Stat. Assoc.*, 1418–1429.
- Bishop, C. (1999). Bayesian PCA. In *NIPS*.
- Archambeau, C. and F. Bach (2008). Sparse probabilistic projections. In *NIPS*.
- Kaiser, H. (1958). https://www.dropbox.com/s/o4hejvydu1ntsk6/Kaiser1958.pdf?dl=0. *Psychometrika 23*(3).

Latent Models

- PCA arises as the Maximum Likelihood solution to a linear Gaussian Latent Variable Model.
- Probabilistic formulation allows:
  - Use of EM for parameter estimation
  - Using mixtures of PCA models
  - A Bayesian formalism in addition allows computing the number of principal components directly from the data.
- Non Gaussian latent variable models lead to Independent Component Analysis.
- Models can also be considered with a non-linear relation between latent and observed variables.

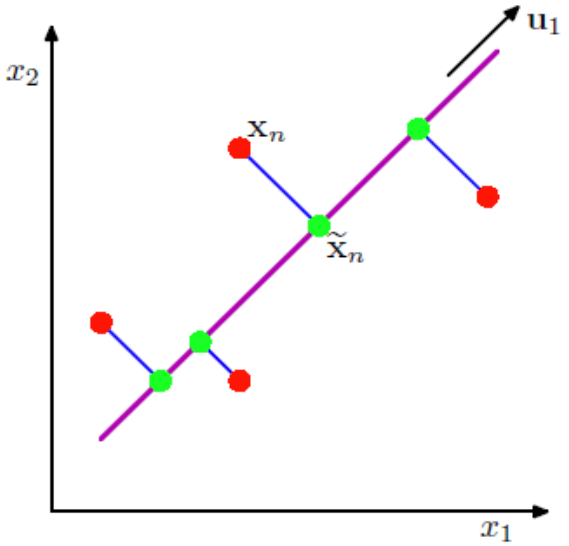
Principal Component Analysis

- Principal component analysis (PCA) – also known as Karhunen-Loeve expansion - is used widely for (Jolliffe, 2002)
  - Dimensionality reduction
  - Lossy data compression
  - Feature selection and
  - Data Visualization
- Principal component analysis seeks a lower dimensionality linear space (principal subspace) such that
  - The orthogonal projection of the data points onto this subspace maximizes the variance of the projected points (Hotelling, 1933)
  - An alternative definition of PCA is based on minimizing the sum-of-squares of the projection errors (Pearson, 1901)

- Jolliffe, I. T. (2002). *Principal Component Analysis* (Second ed.). Springer.
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* **24**, 417–441.
- Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, Sixth Series* **2**, 559–572.

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: Maximum Variance Formulation

- Consider a data set of observations $\{\boldsymbol{x_n}\}, n = 1, \ldots, N$, where $\boldsymbol{x_n}$ has dimensionality D.

- Our goal is to project the data onto a space having dimensionality $M < D$ (principal subspace) while maximizing the variance of the projected data (M here is fixed).

- Let $\{\boldsymbol{u}_i\}, i = 1, \ldots, M$ the basis vectors ($(D \times 1)$ vectors) of the principal subspace.

- We define the sample mean ($(D \times 1)$ vector) by: $\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n$

  and the sample covariance matrix ($(D \times D)$ matrix) by:

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

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: $\boldsymbol{u}_1^T \bar{\boldsymbol{x}}$
- Similarly, the variance of the projected data is:

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

- 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 $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M$ of the data covariance matrix $\boldsymbol{S}$ corresponding to the largest eigenvalues $\lambda_1, \ldots, \lambda_M$.

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

Total Variance

- Essentially PCA becomes the calculation of the eigenvectors of the data covariance matrix corresponding to the largest M eigenvalues.
- $\sum_{i=1}^{D} \lambda_i$ is the total variance.
- The contribution to the total variance by one component $\boldsymbol{u}_i$ is measured as:

$$\frac{\lambda_i}{\sum_{i=1}^{D} \lambda_i}$$

Proof by Induction: M-Dimensional Subspace

- We have shown our result for $M = 1$. Assume it is valid for projection space of dimensionality M. We will show that it is then valid for projection space of dimensionality $M + 1$.
- We need to maximize the following:

$$\max_{\boldsymbol{u}_{M+1}} \left\{ \boldsymbol{u}_{M+1}^T \boldsymbol{S} \boldsymbol{u}_{M+1} + \lambda_{M+1} \left(1 - \boldsymbol{u}_{M+1}^T \boldsymbol{u}_{M+1}\right) + \sum_{i=1}^{M} \eta_i \boldsymbol{u}_{M+1}^T \boldsymbol{u}_i \right\}$$

- The stationary points satisfy:

$$0 = 2\boldsymbol{S}\boldsymbol{u}_{M+1} - 2\lambda_{M+1}\boldsymbol{u}_{M+1} + \sum_{i=1}^{M} \eta_i \boldsymbol{u}_i$$

- Taking dot product with $\boldsymbol{u}_j^T, j = 1, \ldots, M$ gives $\eta_j = 0$. Thus

$\boldsymbol{S}\boldsymbol{u}_{M+1} = \lambda_{M+1}\boldsymbol{u}_{M+1} \Rightarrow \boldsymbol{u}_{M+1}$ eigenvector of $\boldsymbol{S}$ with the variance in this direction $\boldsymbol{u}_{M+1}^T \boldsymbol{S} \boldsymbol{u}_{M+1} = \lambda_{M+1}$

- λ_{M+1} needs to be the next largest eigenvalue after the first M ones → The argument by induction holds for any $M \leq D$.

PCA: Minimum Error Formulation

- This is based on projection error minimization.
- Consider a D-dimensional basis $\{\boldsymbol{u}_i\}$ where $i = 1,\ldots,D$ satisfying $\boldsymbol{u}_1^T\boldsymbol{u}_j = \delta_{ij}$
- Each data point $\boldsymbol{x}_n$ can be represented by:

$$\boldsymbol{x}_n = \sum_{i=1}^{D} a_{ni}\boldsymbol{u}_i \,, \text{ where: } a_{ni} = \boldsymbol{x}_n^T\boldsymbol{u}_i$$

- We approximate $\boldsymbol{x}_n$ using a representation with $M < D$ variables z_{ni} via a projection onto a lower-dim subspace:

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

where the $\{z_{ni}\}$ depend on the particular data point, whereas the $\{b_i\}$ are the same constants for all data points.

- We are free to choose the $\{\boldsymbol{u}_i\}, \{z_{ni}\}$, and $\{b_i\}$

PCA: Minimum Error Formulation

- Key idea of PCA: Minimize the distortion J introduced by the dimensionality reduction:

$$\min_{z_{nj},b_j,\boldsymbol{u}_i} J = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2, \quad \tilde{\boldsymbol{x}}_n = \sum_{i=1}^{M} z_{ni}\boldsymbol{u}_i + \sum_{i=M+1}^{D} b_i \boldsymbol{u}_i$$

- Setting the derivatives wrt $\{z_{ni}\}$, and $\{b_i\}$ to zero and using $\boldsymbol{u}_1^T \boldsymbol{u}_j = \delta_{ij}$ gives:

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

$$b_j = \bar{\boldsymbol{x}}^T \boldsymbol{u}_j, \; j = M+1,\ldots,D, \text{ where } \bar{\boldsymbol{x}} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n$$

- If we now substitute these expressions for $\{z_{ni}\}$, and $\{b_i\}$

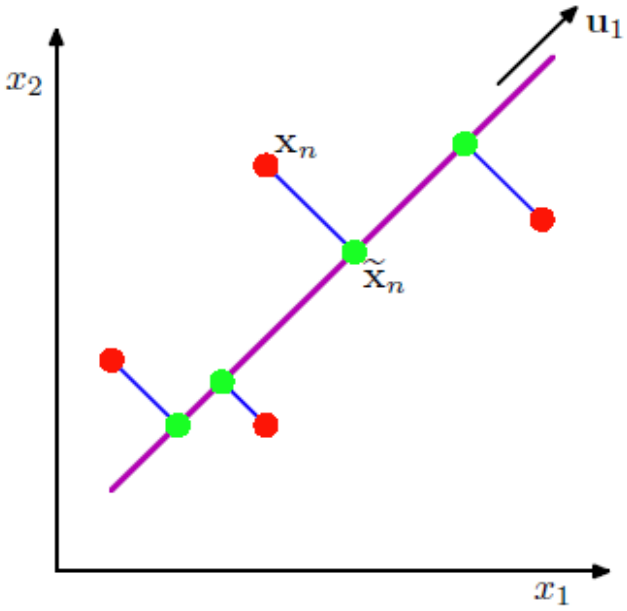
$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \sum_{i=1}^{D} (\boldsymbol{x}_n^T \boldsymbol{u}_i)\boldsymbol{u}_i - \sum_{i=1}^{M} (\boldsymbol{x}_n^T \boldsymbol{u}_i)\boldsymbol{u}_i - \sum_{i=M+1}^{D} (\bar{\boldsymbol{x}}^T \boldsymbol{u}_i)\boldsymbol{u}_i = \sum_{i=M+1}^{D} \{(\boldsymbol{x}_n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

PCA: Minimum Error Formulation

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

- We see that the displacement vector $\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n$ 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 $\tilde{\boldsymbol{x}}_n$ 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

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

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

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

- 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.

PCA: Minimum Error Formulation

- The distortion measure is given as

$$J = \sum_{i=M+1}^{D} \lambda_i$$

- Obtain the minimum value of J by selecting these eigenvectors corresponding to the $D - M$ smallest eigenvalues. The eigenvectors defining the principal subspace are those corresponding to the M largest eigenvalues.

- The two approaches discussed
  - Maximum Variance and
  - Minimum Error

both lead to the same main result of the PCA.

Compression of the Original Data Set

- Using the earlier equations, we can derive:

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

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

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

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

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

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

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

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

PCA Reconstruction

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

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

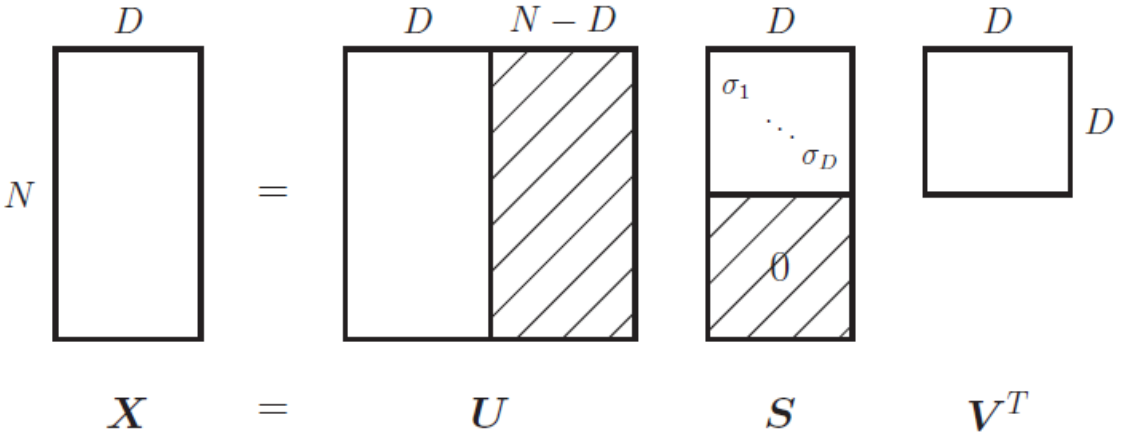
- If $M = D$, there is no dimensionality reduction but a rotation to align with the principal components e.g from $\{x_{n_1}, \dots, x_{n_D}\}$ to $\{a_{n_1}, \dots, a_{n_D}\}$ where:

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

PCA and Singular Value Decomposition

- We have defined the solution to PCA in terms of eigenvectors of the empirical covariance matrix (here denoted as $\widehat{\boldsymbol{\Sigma}}$). The same solution can be obtained based on SVD on the standardized data matrix $\boldsymbol{X}$.

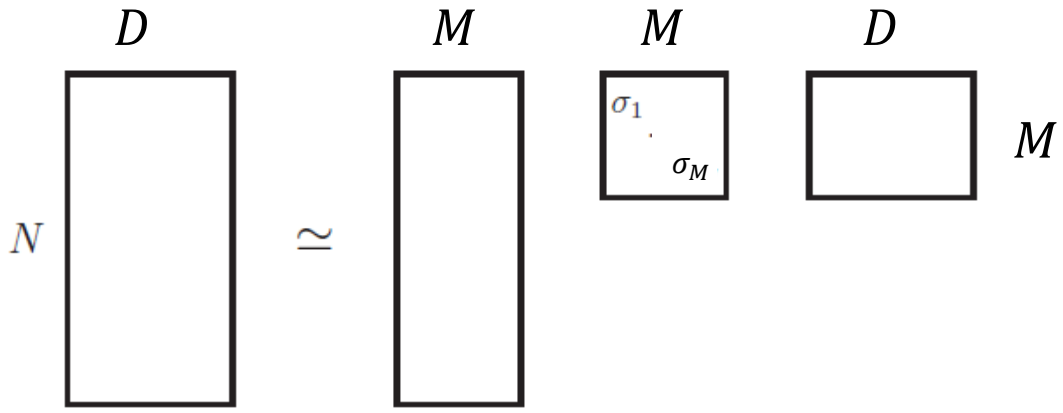
$\boldsymbol{U}, \boldsymbol{V}$: orthonormal
$\sigma_1, \sigma_2, .., \sigma_D$: singular values ($\geq 0$)



- The shaded entries in $\boldsymbol{S}$ and the off diagonal terms are zero. So we don't need to compute the shaded entries of $\boldsymbol{U}$ and $\boldsymbol{S}$.
- It can be shown that the eigenvectors of $\boldsymbol{X}^T\boldsymbol{X}$ are equal to $\boldsymbol{V}$ (right singular vectors of $\boldsymbol{X}$) and the eigenvalues of $\boldsymbol{X}^T\boldsymbol{X}$ equal to $\boldsymbol{D} = \boldsymbol{S}^2$ (squared singular values) $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^T \rightarrow (\boldsymbol{X}^T\boldsymbol{X})\boldsymbol{V} = \boldsymbol{V}\boldsymbol{D}$
- Similarly: $\boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T = U\boldsymbol{S}^2\boldsymbol{U}^T \rightarrow (\boldsymbol{X}\boldsymbol{X}^T)\boldsymbol{U} = \boldsymbol{U}\boldsymbol{D}$

PCA and Singular Value Decomposition

- If the singular values die off quickly, we can produce a rank M approximation of $\boldsymbol{X}$ as follows:



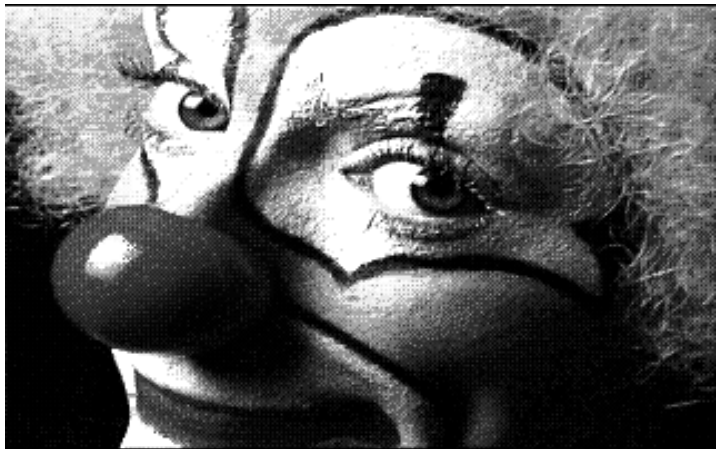
$$\boldsymbol{X} \approx \boldsymbol{U}_M \quad \boldsymbol{S}_M \quad \boldsymbol{V}_M^T$$

- The number of parameters for this truncated SVD are: $NM + M + MD$

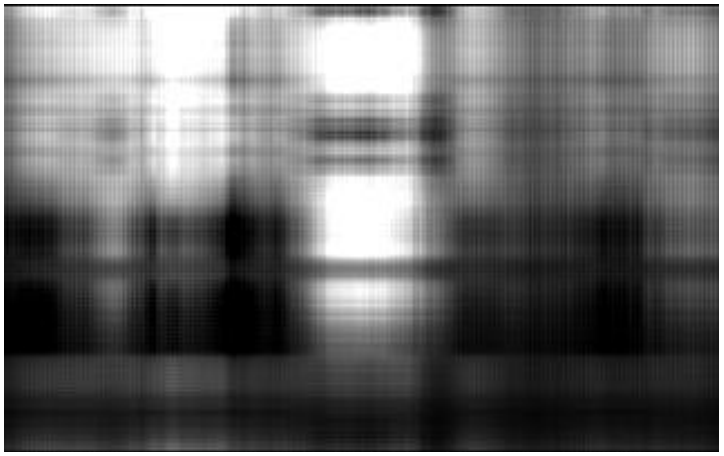
PCA and Singular Value Decomposition

- Consider the image shown of size 200×320 (rank 200) and the SVD approximations of rank $2, 5$, and 20.

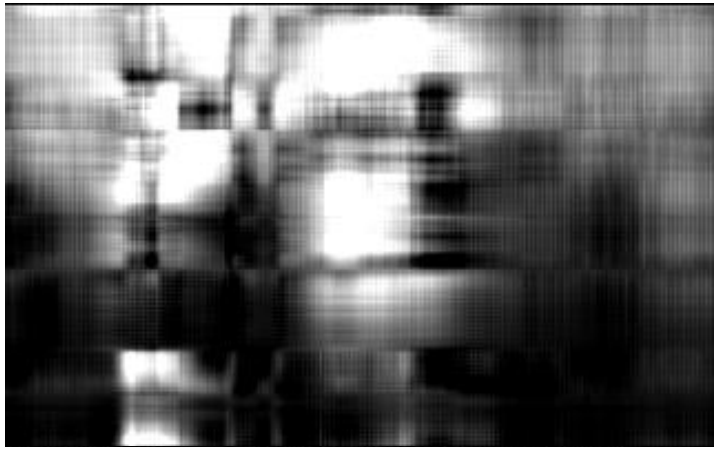
rank 200



rank 2



rank 5



rank 20

Run svdImageDemo
From PMTK3

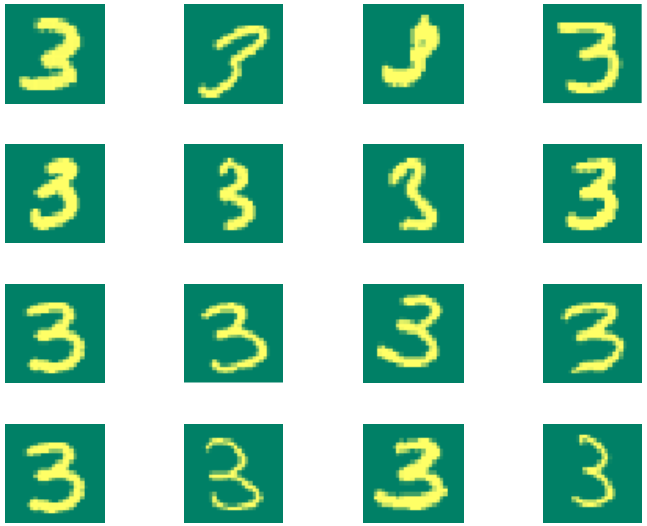
Canonical Correlation Analysis (CCA)

- CCA is a closely related linear dimensionality reduction technique.
- PCA works with a single random variable, CCA considers two (or more) variables.
- CCA finds a corresponding pair of linear subspaces that have high cross-correlation, so that each component within one of the subspaces is correlated with a single component from the other subspace.
- The CCA solution is expressed in terms of a generalized eigenvector problem.

- Bach, F. R. and M. I. Jordan (2002). Kernel independent component analysis. *Journal of Machine Learning Research* **3**, 1–48.
- Hotelling, H. (1936). Relations between two sets of variables. *Biometrika* **28**, 321–377.

Off-line Digit Images: An Example of PCA

- Let us apply PCA in the digit images discussed earlier.
- Consider a synthetic data set obtained by taking one of the digit images and creating multiple copies ($N = 10{,}000$) in each of which the digit has undergone a random displacement and rotation.
- The resulting images each have $D = 28 \times 28 = 784$ pixels.



MatLab Code

Off-line Digit Images: PCA Algorithm

- Our dataset is $\boldsymbol{X} \in D \times N$, where $D = 784$ (pixels) and $N = 10{,}000$ (images).

- Each column of the matrix $\boldsymbol{X}$ is a data point (image) $\boldsymbol{x}_n, (n = 1, 2, \ldots, 10{,}000)$ of size 784.

- We define the centered covariance matrix as:

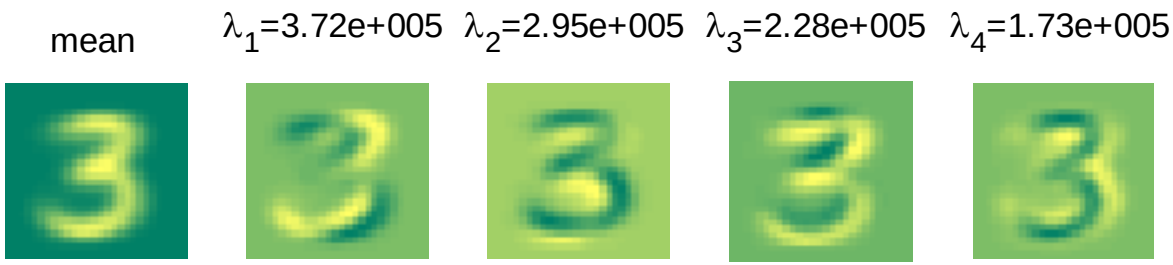
$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_n - \overline{\boldsymbol{x}}\right)^T$$

- To compute the principal components, we solve the following eigenvalue problem:

$$\boldsymbol{S}\boldsymbol{u} = \lambda\boldsymbol{u}$$

Off-line Digit Images: PCA Results

- Using the algorithm stated before, we apply PCA on the off-line digits dataset.
- Since each eigenvector of $\boldsymbol{S}$ is a vector in the original D-dimensional space, we can represent the eigenvectors as images of the same size as the data points (see Fig. below)
- Below is the mean vector $\bar{\boldsymbol{x}}$ along with the first four PCA eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_4$ for the digit 3 from the off-line digits data set, together with the corresponding eigenvalues.



MatLab Code

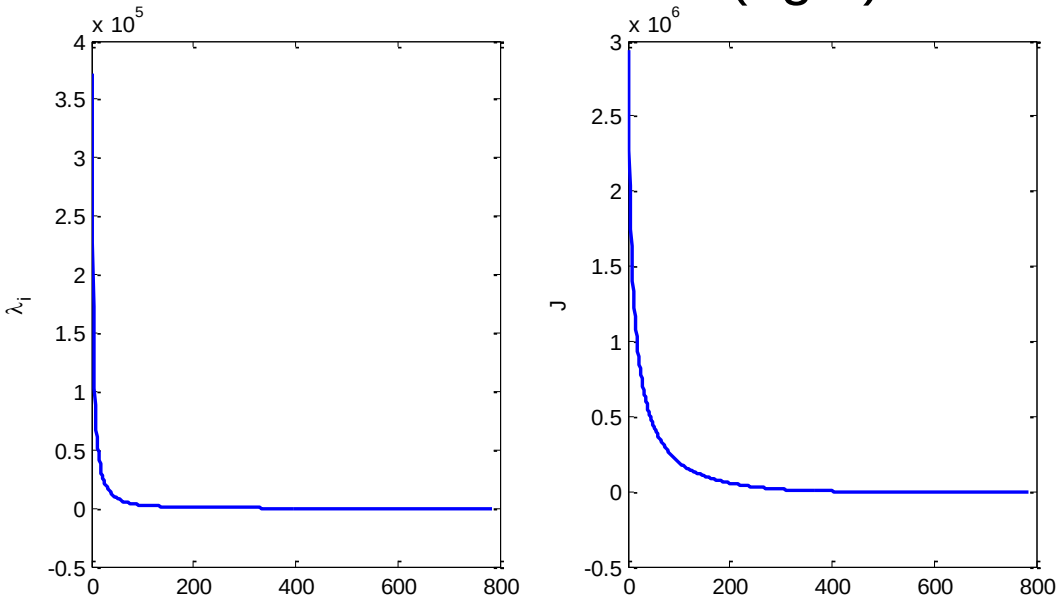
Off-line Digit Images: PCA Results

- The spectrum of eigenvalues, sorted into decreasing order, is shown below (left).

- The distortion measure J associated with choosing a particular value of M is given by

$$J = \sum_{i=M+1}^{D} \lambda_i$$

and is plotted for different values of M (right).



MatLab Code

Off-line Digit Images: PCA Reconstruction

- Below shows an original example of digit 3 from the off-line digits data set together with its PCA reconstructions obtained by retaining M principal components for various values of M.

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

- As M increases the reconstruction becomes more accurate and would become perfect when $M = D = 28 \times 28 = 784$.

MatLab Code

Off-line Digit Images: PCA Reconstruction

mean

principal basis 1

principal basis 2

principal basis 3

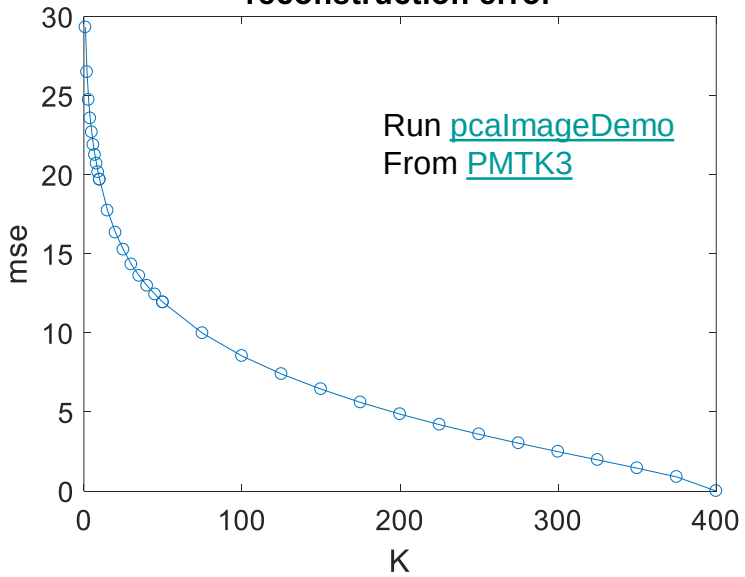
Using 5 bases

Using 10 bases

Using 20 bases

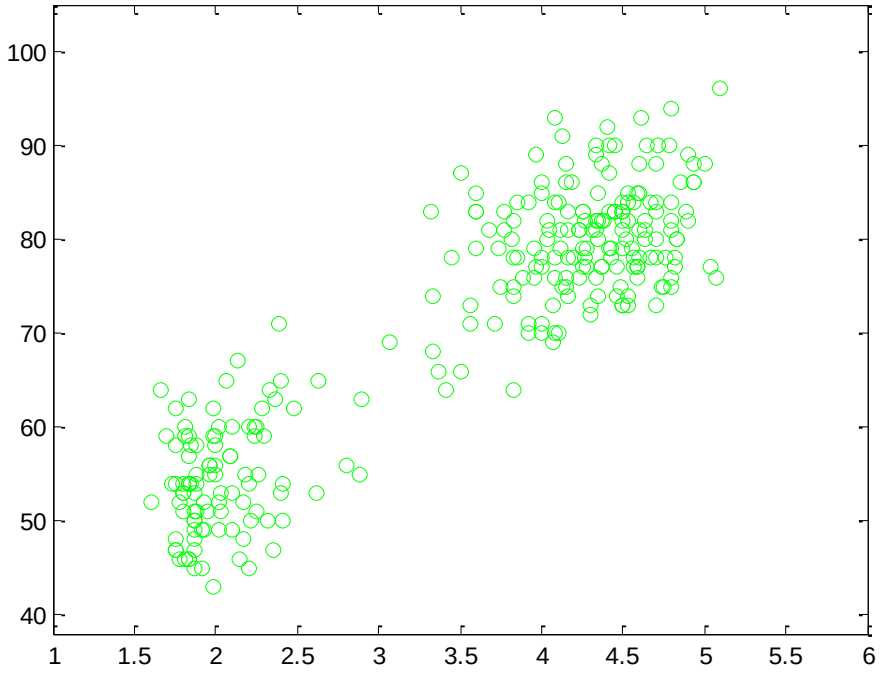
Using 400 bases

reconstruction error



Old Faithful Data Set: Whitening

- Another example is the linear pre-processing of the old faithful dataset.
- Here the goal is not dimensionality reduction but rather the transformation of a data set in order to standardize the data (zero mean/unit variance).
- The original dataset is shown below:



MatLab Code

Old Faithful Data Set: Standardizing

- Standardizing the data is useful for successful application of pattern recognition techniques.
- It is essential when the original variables are measured in various different units or have different scaling.
- In the Old Faithful data set, the time between eruptions is typically an order of magnitude greater than the duration of an eruption.
- Before applying classification techniques (e.g. K-Means) we perform linear re-scaling of the individual variables such that **each variable has zero mean and unit variance**.
- This is known as standardizing the data.

Old Faithful Data Set: Standardizing

- The covariance matrix for the standardized data has components

$$\rho_{ij} = \frac{1}{N}\sum_{n=1}^{N} \frac{x_{ni} - \bar{x}_i}{\sigma_i} \frac{x_{nj} - \bar{x}_j}{\sigma_j}, \sigma_i = std\ of\ x_i$$

- This is the correlation matrix of the original data.

- If two components x_i *and* x_j of the data are perfectly correlated, then $\rho_{ij} = 1$, and if they are uncorrelated, then $\rho_{ij} = 0$.

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:

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

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

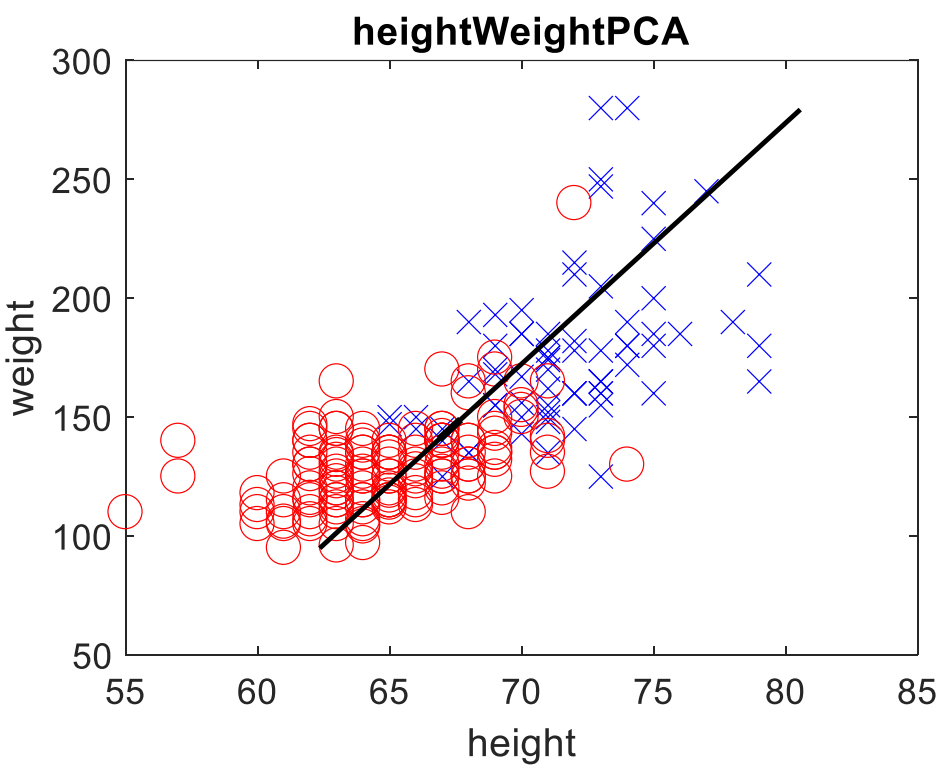
$$\text{Whitening of the data}: \boldsymbol{y}_n = \boldsymbol{L}^{-1/2}\boldsymbol{U}^T\left(\boldsymbol{x}_n - \bar{\boldsymbol{x}}\right)$$

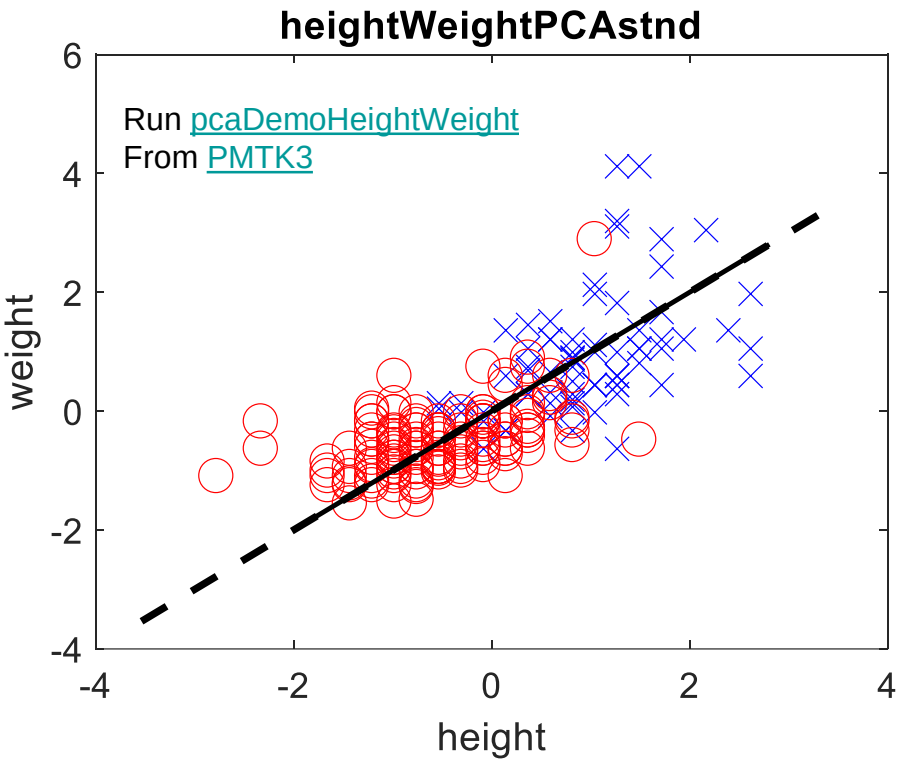
- The set $\boldsymbol{y}_n$ has zero mean and its covariance is the identity:

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

Old Faithful Data Set: PCA Example

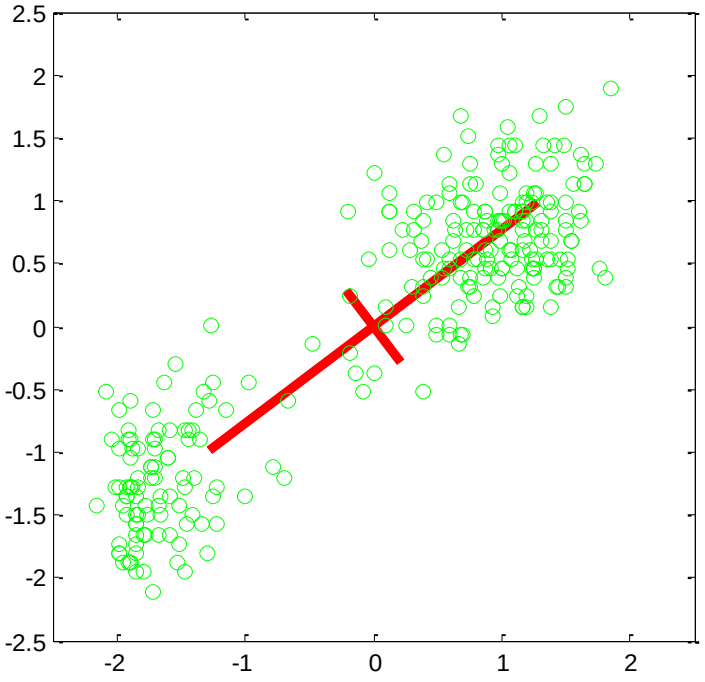
- It should be standard practice to standardize the data first before performing PCA. This is equivalent to working with correlation matrices instead of covariance matrices.
- (Left) PCA for raw data (Right) PCA of standardized data.

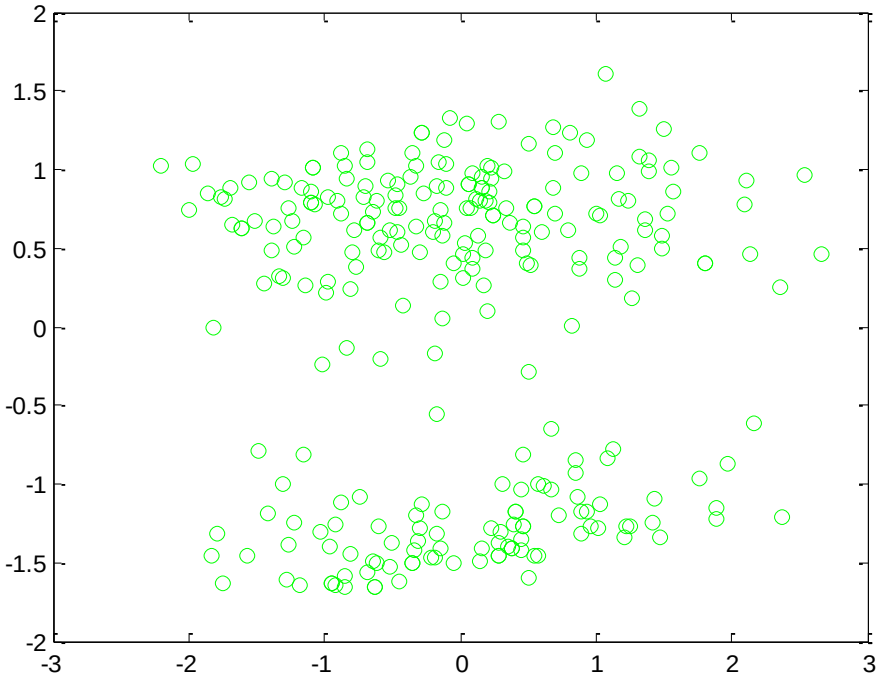




Old Faithful Data Set: PCA Example

- 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)

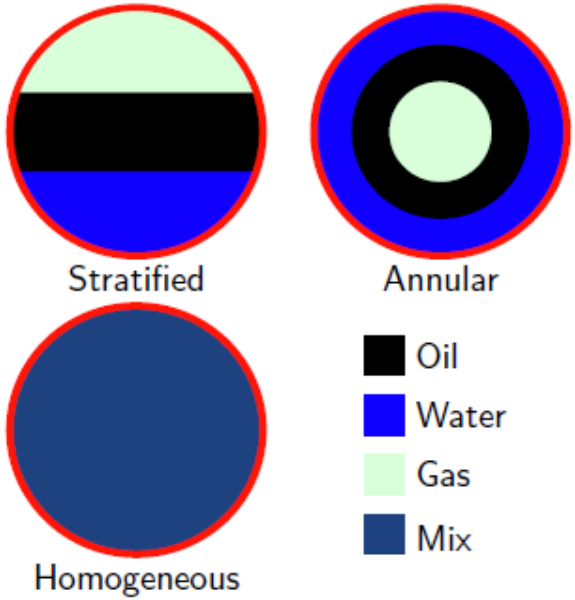




MatLab Code

Oil Flow Data Set

- Consider the oil flow data set.
- For a given geometrical configuration of the gas, water, and oil phases) there are only 2 DOF of variability -- the fraction of oil in the pipe and the fraction of water (the fraction of gas then being determined).
- The data space comprises 12 measurements.
- However, the points in the data set lie close to a 2D manifold embedded within this 12th dimensional space.

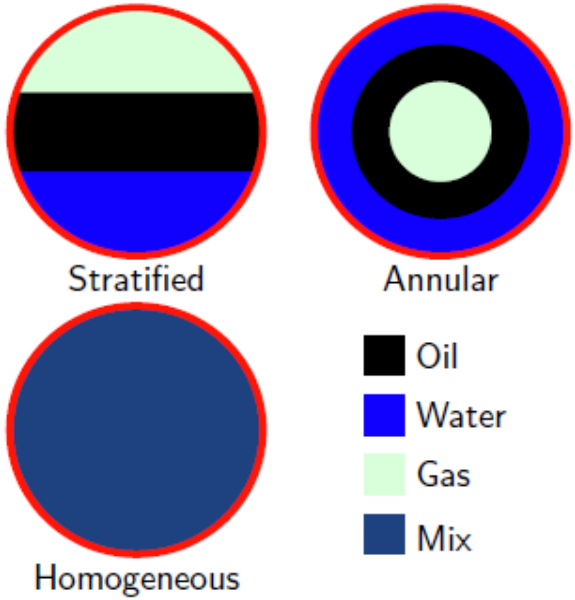


Flow Configurations

Red : 'laminar'
Blue: 'homogeneous'
Green: 'annular'

Oil Flow Data Set

- The manifold comprises several distinct segments corresponding to the different flow regimes, each such segment being a (noisy) continuous 2D manifold.
- For data compression, there is merit in exploiting this manifold structure.

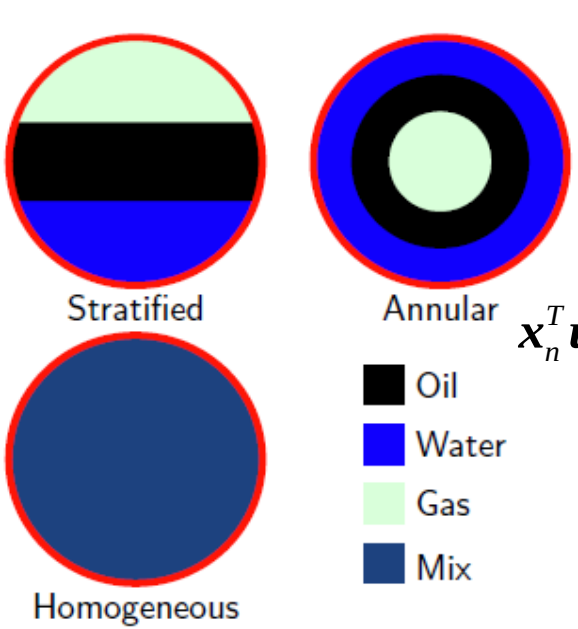


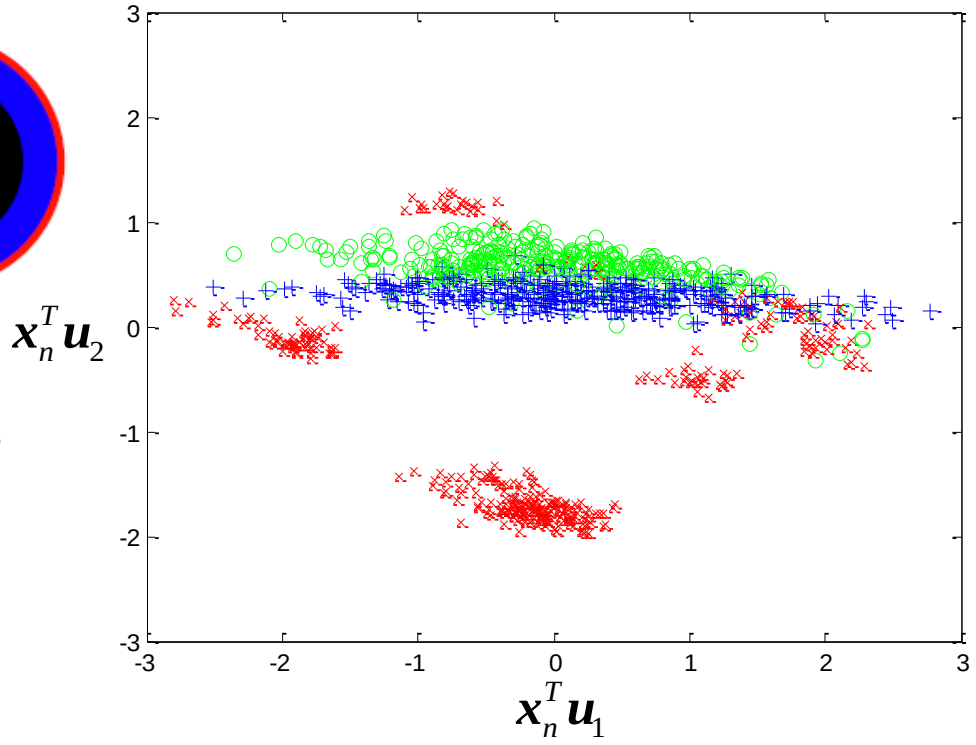
Flow Configurations

Red : 'laminar'
Blue: 'homogeneous'
Green: 'annular'

Oil Flow Data Set: PCA for Visualization

- Visualization of the oil flow data set obtained by projecting the data onto the first two principal components.



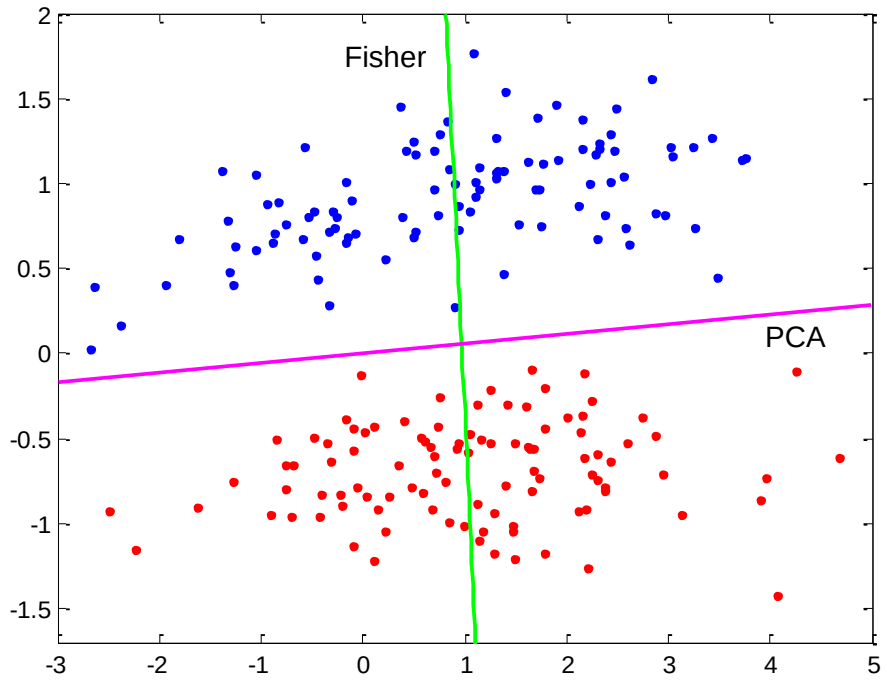


Flow Configurations
Red : 'laminar'
Blue: 'homogeneous'
Green: 'annular'

MatLab Code

PCA Vs. Fisher's Discriminant Method

- PCA and the Fisher linear discriminant method can be seen as linear dimensionality reduction techniques.
- PCA is unsupervised and depends only on $\boldsymbol{x}_n$ whereas Fisher linear discriminant also uses class-label information.



MatLab Code

- 2D data, 2 classes
- We are interested projecting the data in 1D.
- PCA chooses the direction of maximum variance (magenta curve) leading to to strong class overlap.
- The Fisher linear discriminant projects (green curve) to maximize class separation.

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 $N << D$, 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 $\left(\boldsymbol{x}_n - \bar{\boldsymbol{x}}\right)^T$.

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

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

- 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 $\boldsymbol{X}^T$: .

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

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

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

- 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

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

- Bayesian treatment of PCA -- the dimensionality of the principal subspace can be found automatically
- Probabilistic PCA can model class conditional densities and thus be used for classification
- Can be used generatively: samples from the distribution

- Tipping, M. E. and C. M. Bishop (1999a). Mixtures of probabilistic principal component analyzers. *Neural Computation* **11**(2), 443–482.

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

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

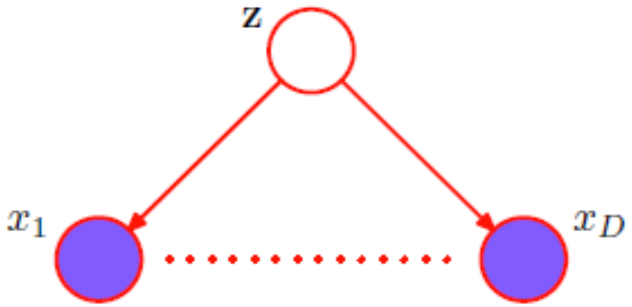
- The conditional distribution of the observed variable $\boldsymbol{x}$, conditioned on the value of $\boldsymbol{z}$, is again Gaussian:

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

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: Naïve Bayes Model

- This model $p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2\boldsymbol{I})$ is an example of Naive Bayes' model - Conditioned on $\boldsymbol{z}$, the components of the observed vector $\boldsymbol{x} = (x_1, \ldots, x_D)^T$ are assumed to be independent.



- We have 2 parameters in the model:
  - (a) $\boldsymbol{W}$, its columns span a linear subspace within the data space that corresponds to the principal subspace;
  - (b) σ^2 – variance of the conditional distribution.

- The prior $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I})$ is without loss of generality.

Probabilistic PCA – Generative View Point

- A latent variable model seeks to relate a D-dimensional observation vector $\boldsymbol{x}$ to a corresponding M-dimensional Gaussian latent variable $\boldsymbol{z}$

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

where

- $\boldsymbol{z}$ is an M-dimensional Gaussian latent variable
- $\boldsymbol{W}$ is an $(D \times M)$ matrix (the latent space)
- $\boldsymbol{\varepsilon}$ is a D-dimensional Gaussian noise
- $\boldsymbol{\varepsilon}$ and $\boldsymbol{z}$ are independent
- $\boldsymbol{\mu}$ is a parameter vector (non zero mean)

- Factor analysis:

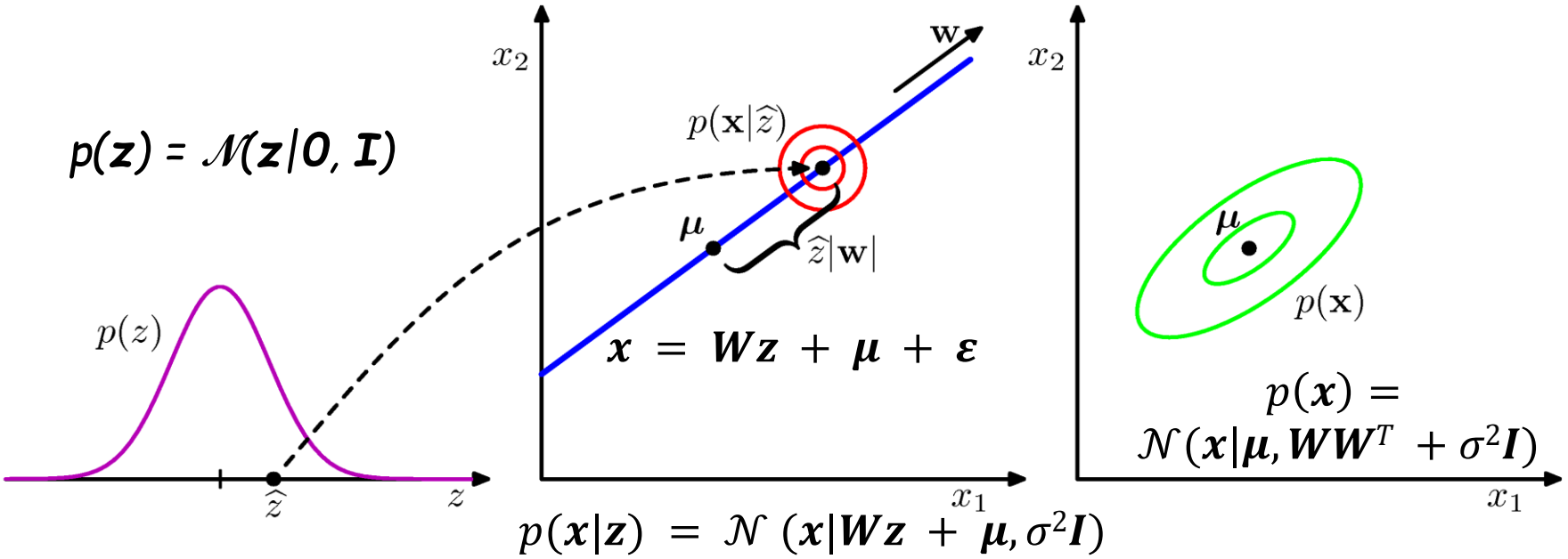
$$\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\psi})$$

- Probabilistic PCA:

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$$

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 $p(\boldsymbol{x})$.



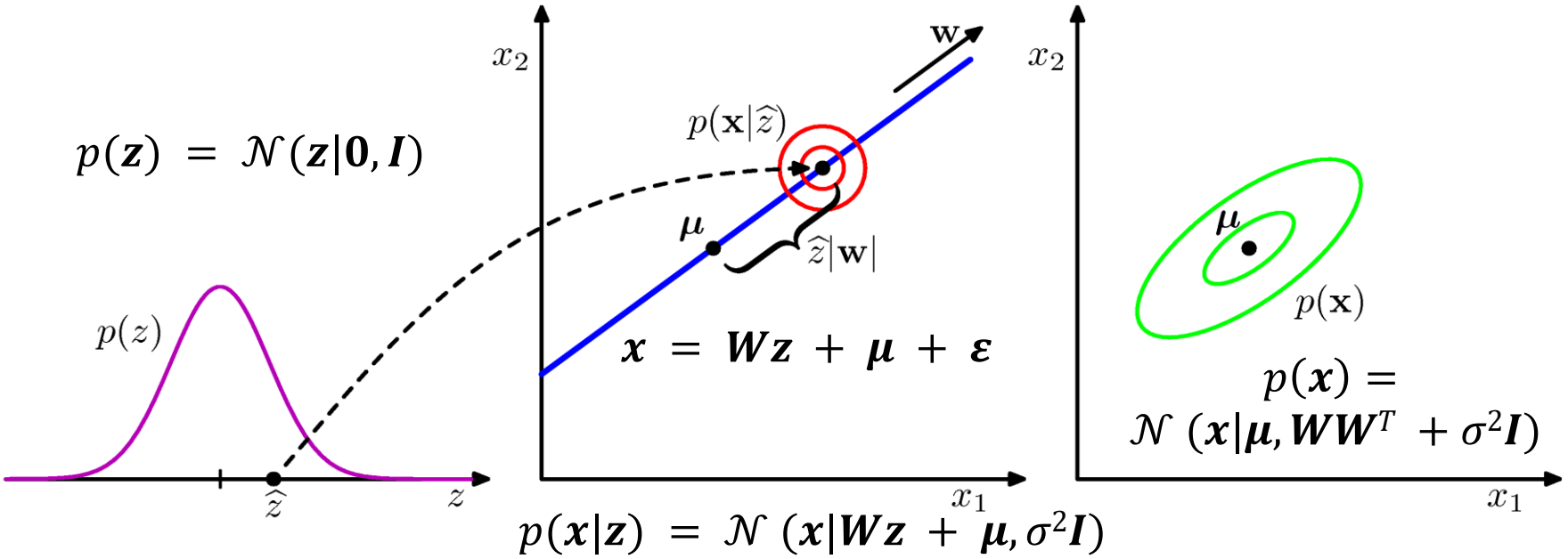
Probabilistic PCA – Generative View Point

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

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x} \mid \boldsymbol{z})\, p(\boldsymbol{z})\, d\boldsymbol{z}$$

- Using Eqs. for linear Gaussian models, this is given as:

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



Probabilistic PCA – Generative View Point

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

- This result can be derived directly by noting that the predictive distribution will be Gaussian and then evaluating its mean and covariance.

$$\mathbb{E}[\boldsymbol{x}] = \mathbb{E}[\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\varepsilon}] = \boldsymbol{W}\,\mathbb{E}[\boldsymbol{z}] + \boldsymbol{\mu} + \mathbb{E}[\boldsymbol{\varepsilon}] = \boldsymbol{\mu}$$

$$\operatorname{var}[\boldsymbol{x}] = \mathbb{E}\left[(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\varepsilon})(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\varepsilon})^T\right]$$

$$= \boldsymbol{W}\boldsymbol{W}^T \operatorname{var}[\boldsymbol{z}] + \mathbb{E}\left[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\right] = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 \boldsymbol{I}$$

- We used here that $\boldsymbol{z}$ and $\boldsymbol{\varepsilon}$ are independent, thus uncorrelated.

The Assumed Prior p(z) is Not Restrictive

- The prior we assumed is not restrictive. Indeed consider a more general prior of the form:

$$p(\boldsymbol{z}) = \mathcal{N}\left(\boldsymbol{z} \mid \boldsymbol{m}, \boldsymbol{\Sigma}\right)$$

- Then the marginal distribution is of the form:

$$p(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{\Sigma}^{-1}\boldsymbol{W}^{T} + \sigma^{2}\boldsymbol{I}\right)$$

- Thus the marginal distribution has an identical form as before:

$$p(\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x} \mid \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{W}}\tilde{\boldsymbol{W}}^{T} + \sigma^{2}\boldsymbol{I}\right), \qquad \tilde{\boldsymbol{\mu}} = \boldsymbol{W}\boldsymbol{m} + \boldsymbol{\mu}, \quad \tilde{\boldsymbol{W}} = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{W}$$

- We choose the simplest prior: $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{0}, \boldsymbol{I})$

Probabilistic PCA – Generative View Point

- There is redundancy in this parametrization.
- There is a whole class of $\boldsymbol{W}$'s differing by a rotation of the latent space coordinates that leads to the same predictive distribution. Indeed:
  - Let

$$\widetilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{R}, \ where \ \boldsymbol{R} = orthogonal \, matrix$$

  - Then

$$\widetilde{\boldsymbol{W}}\widetilde{\boldsymbol{W}}^T = \boldsymbol{W}\boldsymbol{R}\boldsymbol{R}^T\boldsymbol{W}^T = \boldsymbol{W}\boldsymbol{W}^T$$

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 – Predictive Distribution

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

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

- To derive this we used the Woodbury identity:

$$\left(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}\left(\boldsymbol{D} + \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{C}\boldsymbol{A}^{-1}$$

- This identity (easy to show with direct substitution) is used often when
  - ***A*** is large and diagonal (so easy to invert),
  - ***B*** has many rows but few columns  (and conversely for ***C*** ) so that the rhs is much cheaper to evaluate than the lhs.

Probabilistic PCA – Posterior Distribution

- The posterior distribution can be derived directly from earlier results on linear Gaussian models.
  - We know $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I})$
  - 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:

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

Here we used results from an earlier lecture:

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

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

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

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:

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

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

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

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

Maximum Likelihood PCA

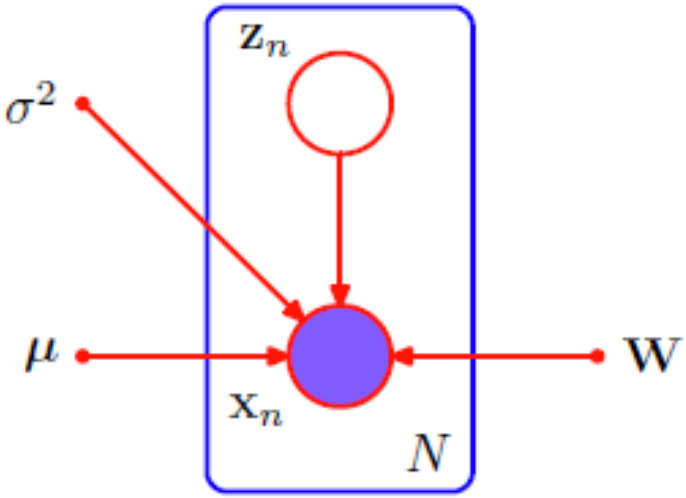
- The log-likelihood is then simplified as:

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

or

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

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



- 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 $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M$. The columns of $\boldsymbol{W}$ then define the principal subspace as in standard PCA.

Maximum Likelihood PCA

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

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

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

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

- For $\boldsymbol{R} = \boldsymbol{I}$, the columns of $\boldsymbol{W}$ are the principal component eigenvectors scaled by $(\lambda_i - \sigma^2)^{1/2}$.

- Interpretation: for the convolution of independent Gaussian distributions, here of $\boldsymbol{z}$ and the noise $\boldsymbol{\varepsilon}$, the variances are additive.

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

- The variance λ_i in the direction of an eigenvector $\boldsymbol{u}_i$ is composed of the sum of a contribution $\lambda_i - \sigma^2$ from the projection of the unit-variance latent space $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I})$ into data space through the corresponding column of $\boldsymbol{W}$, plus an isotropic contribution of variance σ^2 added in all directions by the noise model $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$.

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 $\boldsymbol{v}$ 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.

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

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:

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

  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

$$\boldsymbol{W}_{ML} = \boldsymbol{U}_M \left(\boldsymbol{L}_M - \sigma^2 \boldsymbol{I} \right)^{1/2} \boldsymbol{R} \qquad \sigma^2_{ML} = \frac{1}{D-M} \sum_{i=M+1}^{N} \lambda_i$$

- 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

- If we consider no dimensionality reduction ($M = D$), then

$$\boldsymbol{U}_M = \boldsymbol{U}, \quad \boldsymbol{L}_M = \boldsymbol{L},$$

and using $\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}$, $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$, we see that the covariance $\boldsymbol{C}$ of the marginal distribution for $\boldsymbol{x}$ becomes

$$\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2\boldsymbol{I} = \boldsymbol{U}\left(\boldsymbol{L} - \sigma^2\boldsymbol{I}\right)^{1/2} \boldsymbol{R}\boldsymbol{R}^T \left(\boldsymbol{L} - \sigma^2\boldsymbol{I}\right)^{1/2} \boldsymbol{U}^T + \sigma^2\boldsymbol{I} = \boldsymbol{U}\boldsymbol{L}\boldsymbol{U}^T = \boldsymbol{S}$$

- This result is the standard MLE solution for an unconstrained Gaussian distribution in which the covariance matrix is given by the sample covariance.

Maximum Likelihood PCA

- Since $\boldsymbol{C}$ is not full rank, we can use the matrix inversion lemma:

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

- Plugging the MLE estimates for $\boldsymbol{W}, \sigma^2$ gives: $\boldsymbol{W} = \boldsymbol{U}_M\left(\boldsymbol{L}_M - \sigma^2\boldsymbol{I}\right)^{1/2}$

$$\boldsymbol{C}^{-1} = \frac{1}{\sigma^2}\left[\boldsymbol{I} - \boldsymbol{U}_M\left(\boldsymbol{L}_M - \sigma^2\boldsymbol{I}\right)^{1/2}\left(\boldsymbol{L}_M - \sigma^2\boldsymbol{I} + \sigma^2\boldsymbol{I}\right)^{-1}\left(\boldsymbol{L}_M - \sigma^2\boldsymbol{I}\right)^{1/2}\boldsymbol{U}_M^T \right]$$

$$= \frac{1}{\sigma^2}\left[\boldsymbol{I} - \boldsymbol{U}_M\left(\boldsymbol{L}_M - \sigma^2\boldsymbol{I}\right)^{1/2}\boldsymbol{L}_M^{-1}\left(\boldsymbol{L}_M - \sigma^2\boldsymbol{I}\right)^{1/2}\boldsymbol{U}_M^T \right]$$

$$= \frac{1}{\sigma^2}\left[\boldsymbol{I} - \boldsymbol{U}_M \, diag\left(1 - \frac{\sigma^2}{\lambda_j}\right)\boldsymbol{U}_M^T \right]$$

- Similarly: $\log\left|\boldsymbol{W}\boldsymbol{W}^T + \sigma^2\boldsymbol{I}\right| = (D-M)\log\sigma^2 + \sum_{i=1}^{M}\log\lambda_i$

Use the matrix inversion lemma $det(A + UWV^T) = det(W^{-1} + V^T A^{-1} U) detW detA$

Maximum Likelihood PCA

- PCA is generally expressed as a projection of points from the D-dimensional dataspace onto an M-dimensional subspace.
- Probabilistic PCA is seen as the mapping $\boldsymbol{z} \rightarrow \boldsymbol{x}, \boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\varepsilon}$.
- The reverse mapping $\boldsymbol{x} \rightarrow \boldsymbol{z}$ is computed using the posterior:

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

- Every point in data space is characterized by its posterior mean and covariance in latent space:

$$\mathbb{E}\left[\boldsymbol{z} \mid \boldsymbol{x}\right] = \boldsymbol{M}^{-1}\boldsymbol{W}_{ML}^{T}(\boldsymbol{x} - \overline{\boldsymbol{x}}), \operatorname{var}\left[\boldsymbol{z} \mid \boldsymbol{x}\right] = \sigma^{2}\boldsymbol{M}^{-1}$$

- This projects to a point in data space given by:

$$\boldsymbol{W}\mathbb{E}\left[\boldsymbol{z} \mid \boldsymbol{x}\right] + \boldsymbol{\mu}$$

- This is identical form as regularized linear regression.

Maximum Likelihood PCA vs Standard PCA

- In the limit $\sigma^2 \to 0$, the posterior mean becomes:

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

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

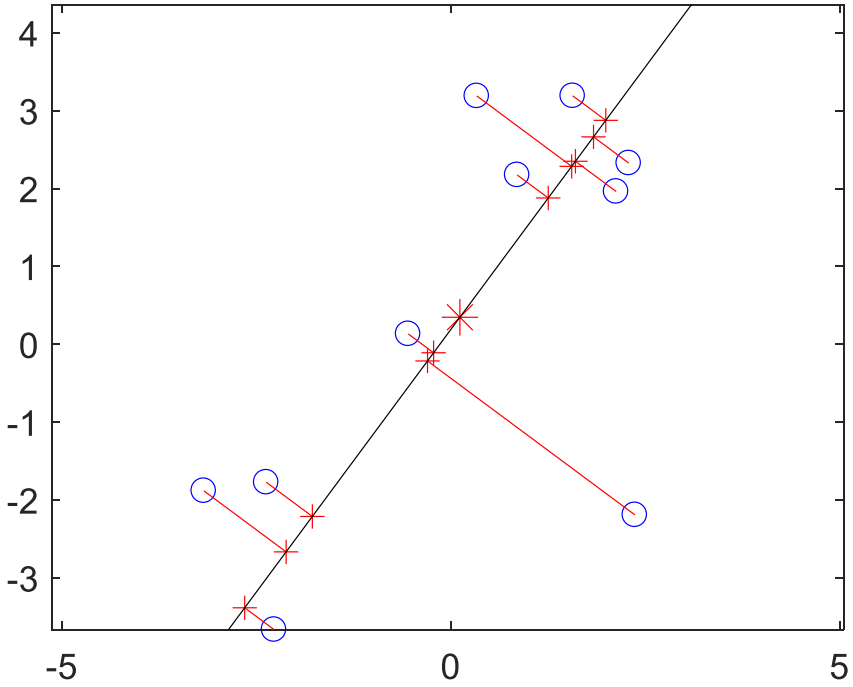
- Now substitute $\boldsymbol{W}_{ML} = \boldsymbol{U}_M\left(\boldsymbol{L}_M - \sigma^2\boldsymbol{I}\right)^{1/2}\boldsymbol{R} = \boldsymbol{U}_M\boldsymbol{L}_M^{1/2}$ for $\sigma^2 \to 0$ with $\boldsymbol{R} = \boldsymbol{I}$ (for consistency with PCA). Then

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

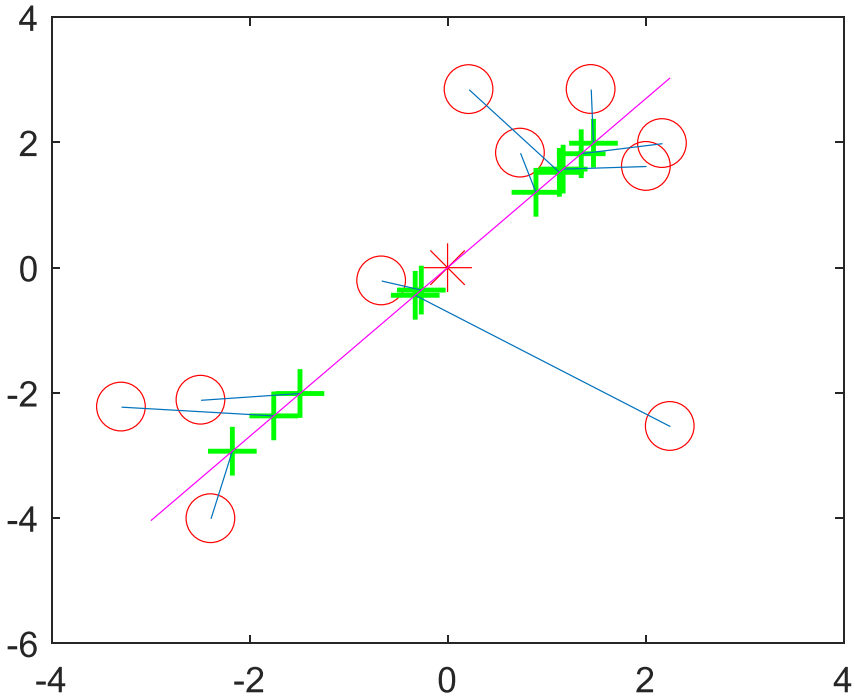
- This is an orthogonal projection of the data point onto the latent space, i.e. for the limit $\sigma^2 \to 0$, we recover the standard PCA model. The posterior covariance $\operatorname{var}\left[\boldsymbol{z} \mid \boldsymbol{x}\right] = \sigma^2\boldsymbol{M}^{-1} \to 0$ and the density becomes 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

- PPCA defines a multivariate Gaussian distribution in which the number of DOF (independent parameters) can be controlled while still capturing the dominant correlations in the data.

- A general Gaussian distribution has $D(D+1)/2$ independent parameters in its covariance matrix and D parameters in its mean.

  - The number of parameters scales as D^2.

- For a diagonal covariance matrix we have D independent parameters and the number of parameters grows linearly. However, the variables are independent and hence this model cannot express any correlations between them.

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

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

Probabilistic PCA: Number of DOF

- The number of DOF in $\boldsymbol{C}$ grows linearly in D for a given M:

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

- *M = D − 1*: recover the standard result for a full covariance Gaussian.
  - The variance along $D - 1$ linearly independent directions is controlled by the columns of $\boldsymbol{W}$, and
  - the variance along the remaining direction is given by σ^2.
- $M = 0$: equivalent to the isotropic covariance case.

EM Algorithms for PCA

- The PPCA model involves a marginalization over a continuous latent space $\boldsymbol{z}$. For each $\boldsymbol{x}_n$ there is a corresponding $\boldsymbol{z}_n$.
- Use EM to find the MLE of the model parameters.
- For high D, there are advantages using iteratively EM vs. working directly with the sample covariance.
- This approach can be extended to factor analysis for which there is no closed-form solution.
- Can also be used when values are missing & for mixture models.
- EM requires the complete-data log likelihood function:

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

- In the following, we substitute μ with the sample mean $\overline{\boldsymbol{x}}$.

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.

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

- 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

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

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

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

- This follows directly from

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

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

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

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

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

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

Proof of the M-Step Equations

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

- The two M-equations are derived by setting the derivatives wrt $\boldsymbol{W}$ and σ^2 equal to zero:

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

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

- 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

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

- Revise the parameter values in the M-Step.

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

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

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: Computational Cost

- The EM algorithm described here also does not construct the covariance matrix explicitly.

- Instead, the most computationally demanding steps are those involving sums over the data set that are $\mathcal{O}(NDM)$.

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

- For large D, and $M << D, \mathcal{O}(NDM)$ is much better compared to $\mathcal{O}(ND^2)$ (that we show is needed to compute the sample covariance)

  This offsets the iterative nature of the EM algorithm.

EM Algorithms for PCA: Online form

- The EM algorithm can be implemented in an on-line form: each D-dim data point is read/processed/then discarded before the next data point is considered.

- To see this, note that the quantities evaluated in the E-Step (an M-dimensional vector and an M × M matrix) can be computed for each data point separately.

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

- In the M-Step, we accumulate the sums over data points incrementally - advantageous if both N & D are large.

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

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

EM Algorithms for PCA: Missing Values

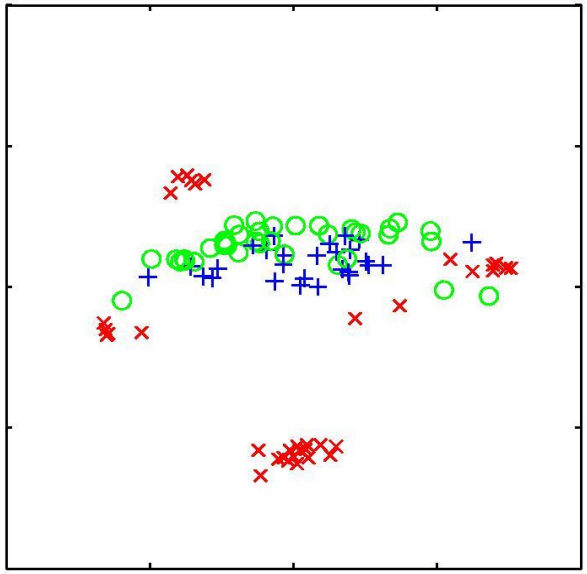
- Because we now have a fully probabilistic model for PCA, we can deal with missing data, provided that it is missing at random, by marginalizing over the distribution of the unobserved variables.

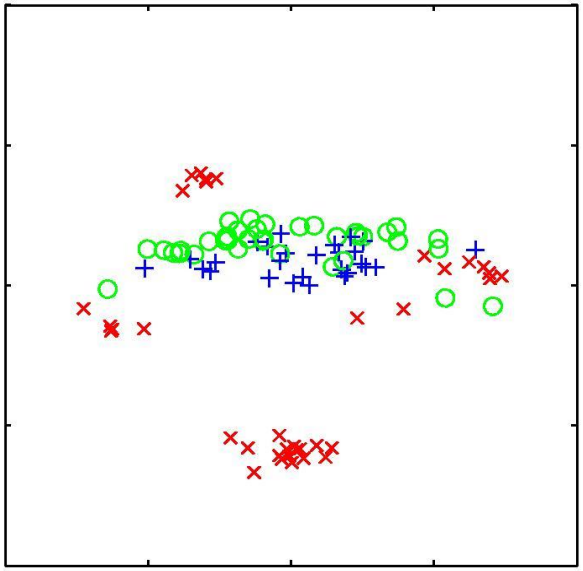
- Again these missing values can be treated using the EM algorithm.

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

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

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
- $\tilde{\boldsymbol{X}}$ an $N \times D$ matrix whose n[th] row is given by $\boldsymbol{x}_n - \bar{\boldsymbol{x}}$.
- Ω a matrix $M \times N$ whose n[th] 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:

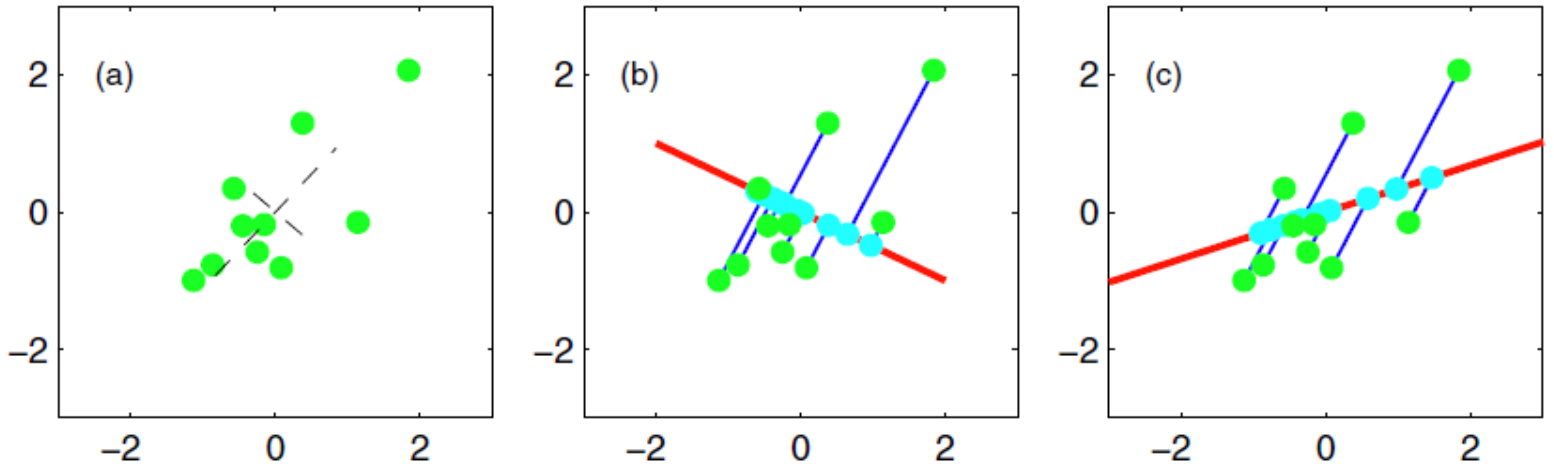
$$\boldsymbol{W}_{new} = \left[\sum_{n=1}^{N}\left(\boldsymbol{x}_n - \bar{\boldsymbol{x}}\right)\mathbb{E}\left[\boldsymbol{z}_n\right]^T\right]\left[\sum_{n=1}^{N}\mathbb{E}\left[\boldsymbol{z}_n \boldsymbol{z}_n^T\right]\right]^{-1} \Rightarrow \quad \boldsymbol{W}_{new} = \tilde{\boldsymbol{X}}^T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

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

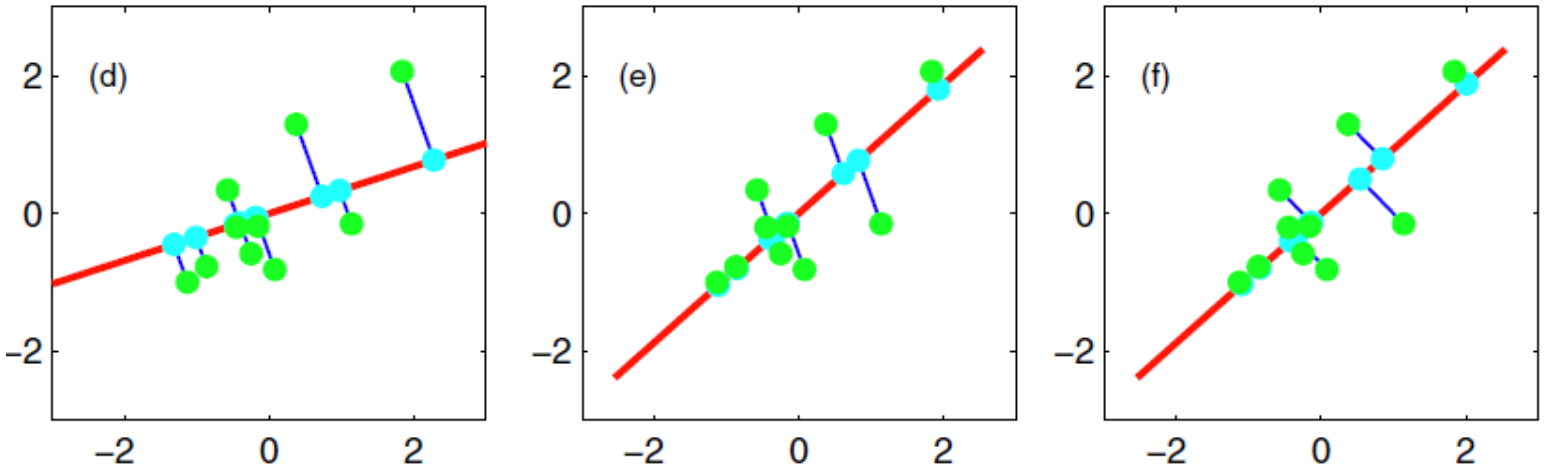
- Synthetic data illustrating the EM algorithm for PCA
  - (a) A data set $\boldsymbol{X}$ with the data points (green), together with the true principal components (eigenvectors scaled by the square roots of the eigenvalues).
  - (b) Initial configuration of the principal subspace defined by $\boldsymbol{W}$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

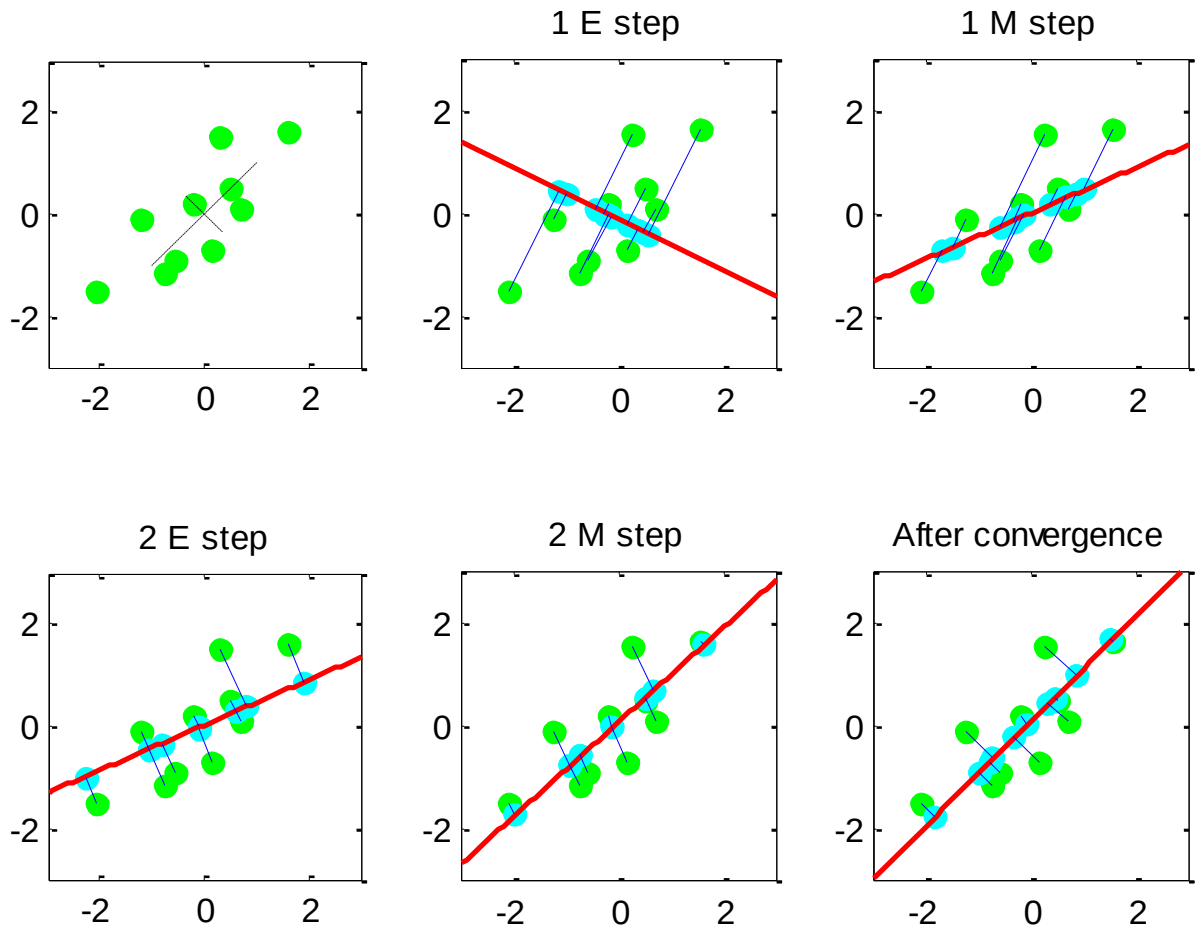
- Synthetic data illustrating the EM algorithm for PCA
  - (d) After the successive E-Step, the values of $\boldsymbol{Z}$ have been updated, giving orthogonal projections, with $\boldsymbol{W}$ held fixed.
  - (e) After the second M-Step.
  - (f) The converged solution.



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

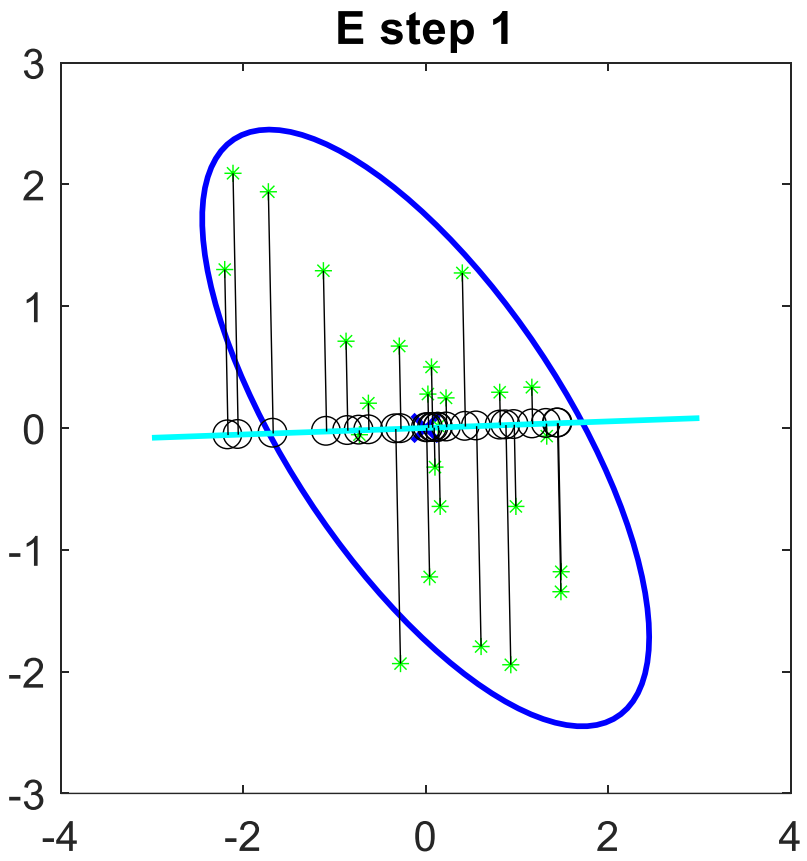
$$\boldsymbol{W}_{new} = \widetilde{\boldsymbol{X}}^T \boldsymbol{\Omega}^T \left(\boldsymbol{\Omega}\boldsymbol{\Omega}^T\right)^{-1}$$

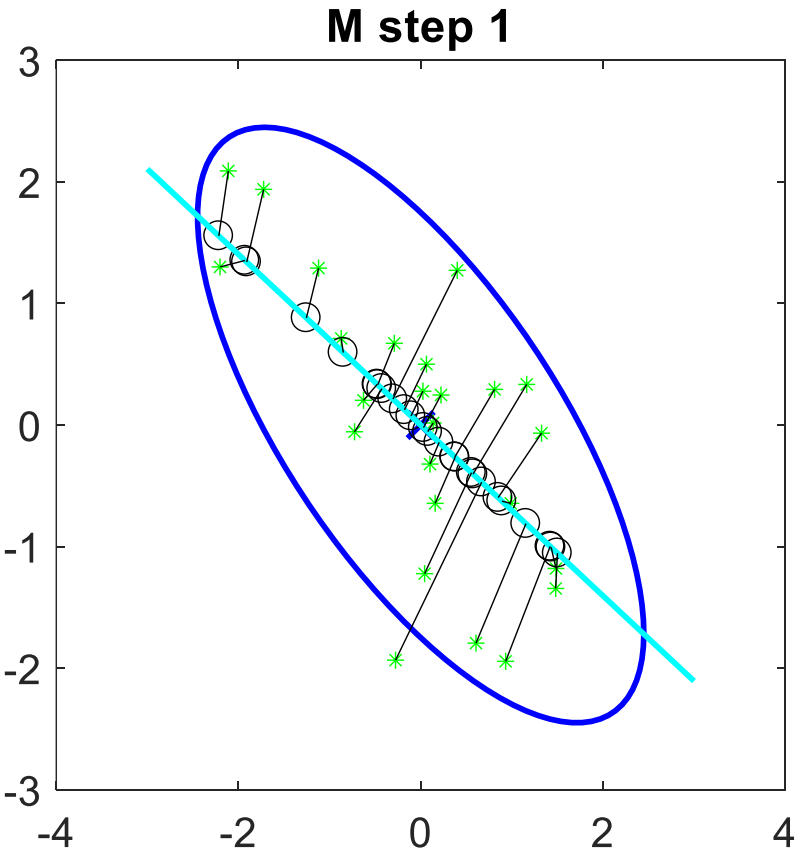
EM Algorithm for PCA: Example, D=2, M=1



Matlab Implementation

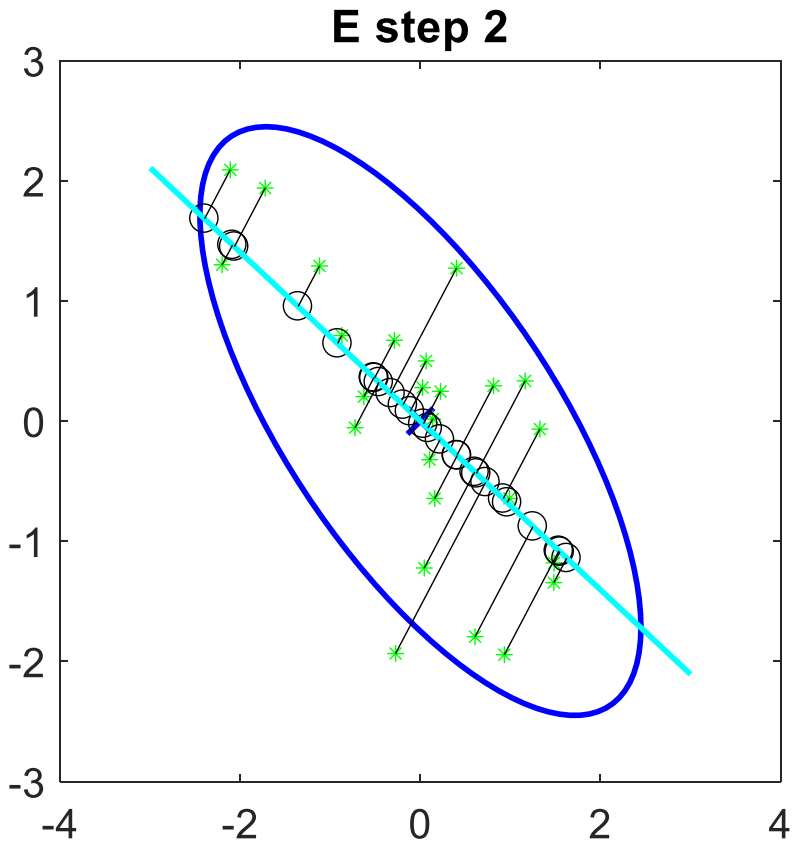
EM Algorithm for PCA: Example, D=2, M=1

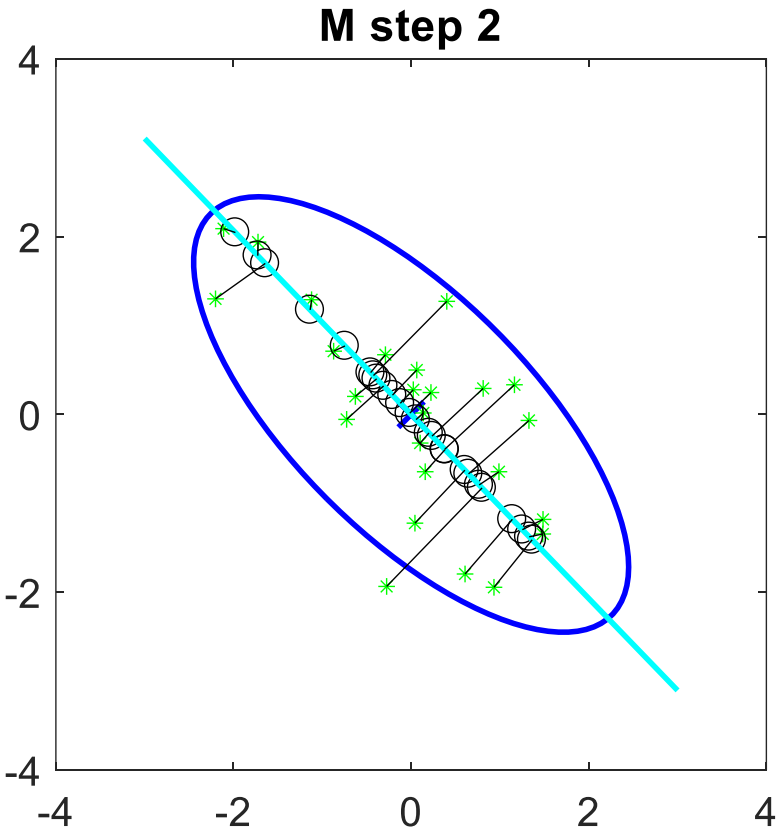




Run pcaEmStepByStep
From PMTK3

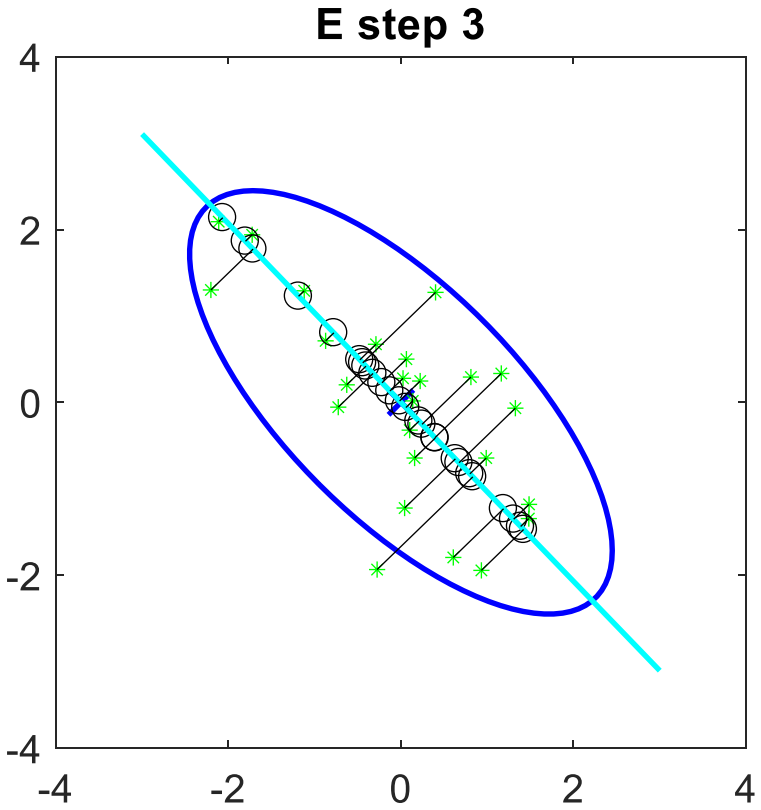
EM Algorithm for PCA: Example, D=2, M=1

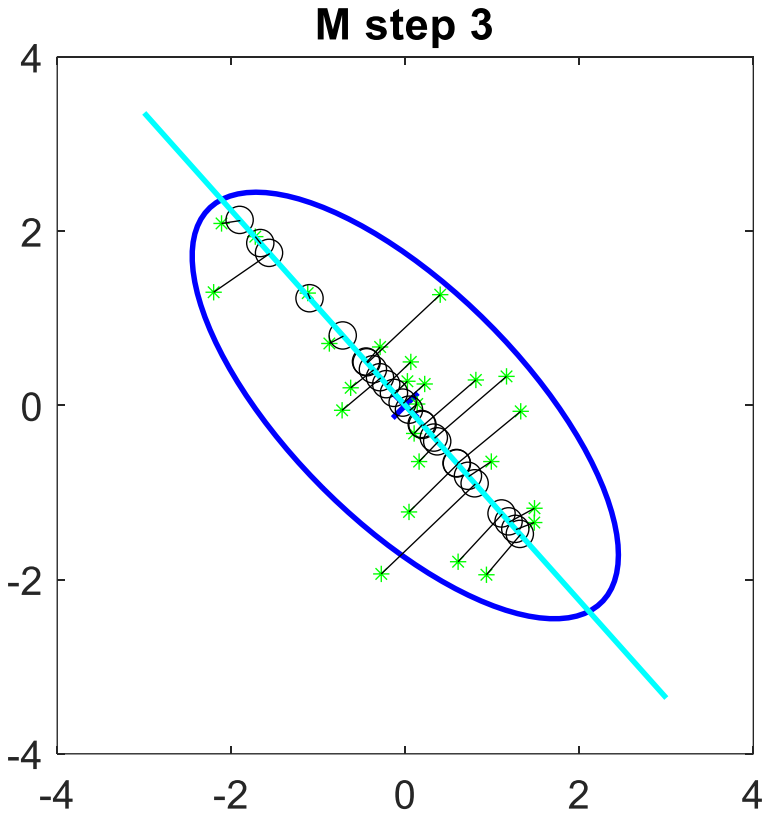




Run pcaEmStepByStep
From PMTK3

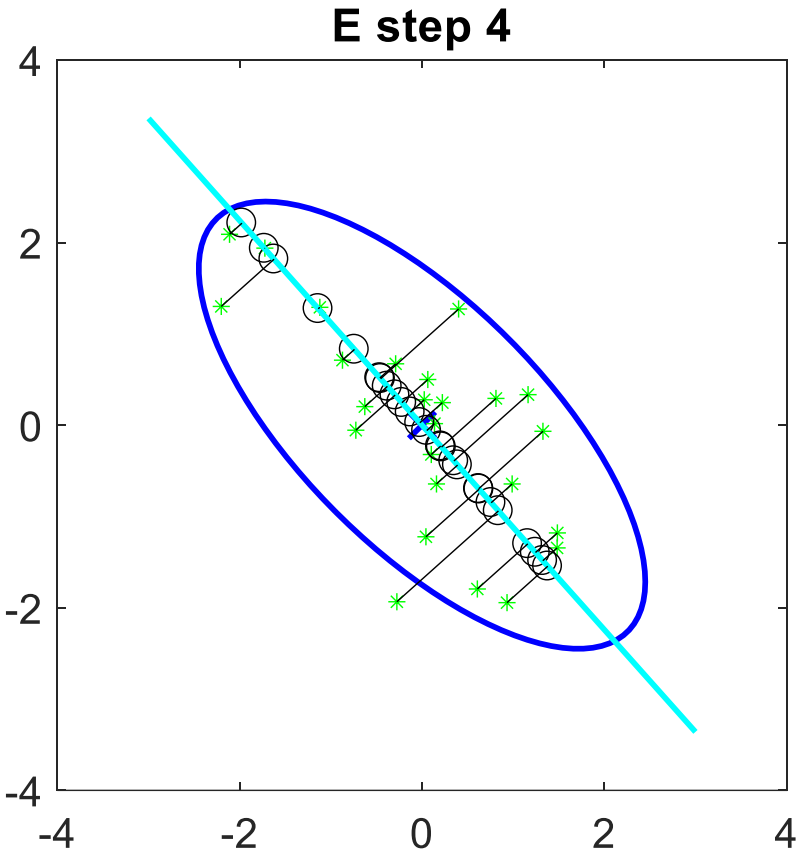
EM Algorithm for PCA: Example, D=2, M=1

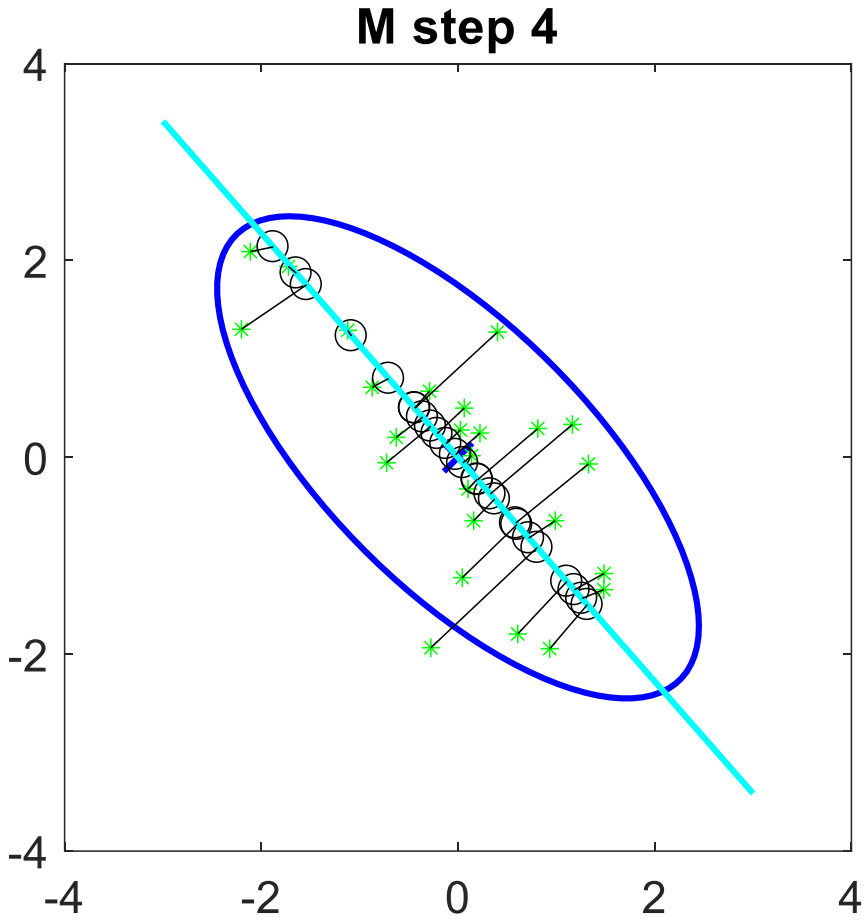




Run pcaEmStepByStep
From PMTK3

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





Run pcaEmStepByStep
From PMTK3

EM Algorithm and Standard PCA Revisited

- Let $\boldsymbol{W}$ be D × M whose columns define a linear subspace of dimensionality M embedded within a data set of dimensionality D.
- Let $\boldsymbol{\mu}$ a D-dimensional vector. We approximate the data points $\boldsymbol{x}_n, n = 1,..,N$ using a linear mapping from a set of D-dimensional vectors $\boldsymbol{z}_n$ as $\boldsymbol{W}\boldsymbol{z}_n + \boldsymbol{\mu}$.
- The sum of squares reconstruction error is:

$$J = \sum_{n=1}^{N} \left\| \boldsymbol{x}_n - \boldsymbol{\mu} - \boldsymbol{W}\boldsymbol{z}_n \right\|^2$$

- Minimizing wrt $\boldsymbol{\mu}$ gives:

$$0 = -\sum_{n=1}^{N} \left(\boldsymbol{x}_n - \boldsymbol{\mu} - \boldsymbol{W}\boldsymbol{z}_n \right) \Rightarrow \boldsymbol{\mu} = \bar{\boldsymbol{x}} - \boldsymbol{W}\bar{\boldsymbol{z}}$$

- This modifies the error as:

$$J = \sum_{n=1}^{N} \left\| \left(\boldsymbol{x}_n - \bar{\boldsymbol{x}} \right) - \boldsymbol{W} \left(\boldsymbol{z}_n - \bar{\boldsymbol{z}} \right) \right\|^2$$

EM Algorithm and Standard PCA Revisited

- Let $\boldsymbol{X}$ be a $N \times D$ matrix with nth row $\left(\boldsymbol{x}_n - \bar{\boldsymbol{x}}\right)^T$.
- Similarly let $\boldsymbol{Z}$ be a $N \times M$ matrix with nth row $\left(\boldsymbol{z}_n - \bar{\boldsymbol{z}}\right)^T$.
- The cost function can be written as:

$$J = \sum_{n=1}^{N} \left\| \left(\boldsymbol{x}_n - \bar{\boldsymbol{x}}\right) - \boldsymbol{W}\left(\boldsymbol{z}_n - \bar{\boldsymbol{z}}\right) \right\|^2 \Rightarrow \quad J = Tr\left\{ \left(\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^T\right)\left(\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^T\right)^T \right\}$$

- Setting the derivative wrt $\boldsymbol{Z}$ equal to $\boldsymbol{0}$ gives (using $\frac{\partial}{\partial \boldsymbol{A}} Tr(\boldsymbol{A}\boldsymbol{B}\boldsymbol{A}^T) = \boldsymbol{A}(\boldsymbol{B} + \boldsymbol{B}^T)$ and $\frac{\partial}{\partial \boldsymbol{A}} Tr(\boldsymbol{A}\boldsymbol{B}) = \boldsymbol{B}^T, \frac{\partial}{\partial \boldsymbol{A}} Tr(\boldsymbol{A}^T\boldsymbol{B}) = \boldsymbol{B}$) :

$$-2\boldsymbol{X}\boldsymbol{W} + 2\boldsymbol{Z}\boldsymbol{W}^T\boldsymbol{W} = 0 \Rightarrow \boldsymbol{Z} = \boldsymbol{X}\boldsymbol{W}\left(\boldsymbol{W}^T\boldsymbol{W}\right)^{-1}$$

This is simllar to the PCA E-Step: $\boldsymbol{\Omega} = \left(\boldsymbol{W}_{old}^T\boldsymbol{W}_{old}\right)^{-1}\boldsymbol{W}_{old}^T\tilde{\boldsymbol{X}}^T$

- Similarly derivative wrt $\boldsymbol{W}$ equal to $\boldsymbol{0}$ gives (analogous to the PCA M-Step: $\boldsymbol{W}_{new} = \tilde{\boldsymbol{X}}^T\boldsymbol{\Omega}^T(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)^{-1}$)

$$-2\boldsymbol{X}^T\boldsymbol{Z} + 2\boldsymbol{W}\boldsymbol{Z}^T\boldsymbol{Z} = 0 \Rightarrow \boldsymbol{W} = \boldsymbol{X}^T\boldsymbol{Z}\left(\boldsymbol{Z}^T\boldsymbol{Z}\right)^{-1}$$

Model Selection for FA and PPCA

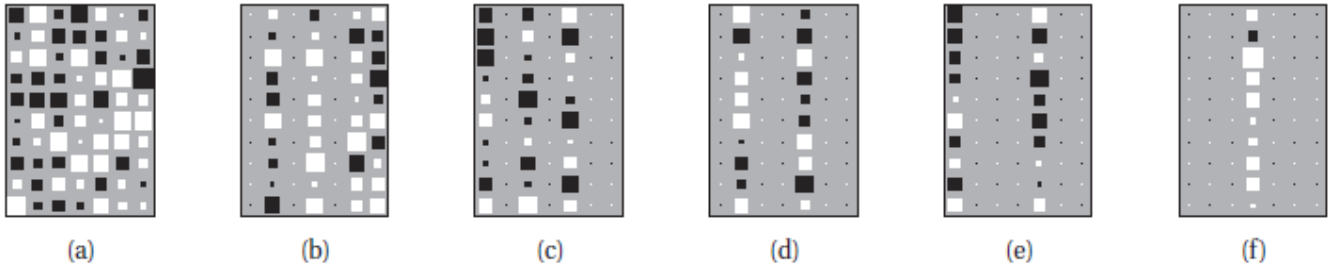
- We can in principle compute $M^* = argmax_M \, p(M|\mathcal{D})$.
- There are two problems with this.
  - Evaluating the marginal likelihood for LVMs is quite difficult.
    - ✓ One can use BIC or variational lower bounds.
    - ✓ Alternatively, we can use the cross-validated likelihood as a performance measure - this can be slow, since it requires fitting each model F times, where F is the number of CV folds.
  - The second issue is the need to search over a potentially large number of models.
    - ✓ The usual approach is to perform exhaustive search over all candidate values of M.
    - ✓ However, sometimes we can set the model to its maximal size, and then use automatic relevancy determination combined with EM to prune out irrelevant weights.

- Minka, T. (2000a). Automatical choice of dimensionality for PCA. Technical report, MIT.
- Bishop, C. (1999). Bayesian PCA. In *NIPS*.
- Ghahramani, Z. and M. Beal (2000). Variational inference for Bayesian mixtures of factor analysers. In *NIPS-12*.
- Lopes, H. and M. West (2004). Bayesian model assessment in factor analysis. *Statisica Sinica 14*, 41– 67.
- Paisley, J. and L. Carin (2009). Nonparametric factor analysis with beta process priors. In *Intl. Conf. on Machine Learning*.

Model Selection for FA and PPCA

- The ARD approach is shown in the figure using **Hinton diagrams** (for a mixture of FA using VBEM).
- The degree of sparsity depends on the amount of training data in accord with Occam's razor. When the sample size is small, the method prefers simpler models, but as the sample size gets sufficiently large, the method converges on the "correct" solution.

The data was generated from 6 clusters with intrinsic dimensionalities of 7, 4, 3, 2, 2, 1, which the method has successfully estimated.



number of points per cluster	intrinsic dimensionalities					
	1	7	4	3	2	2
8	2				1	
8	1		2			
16	1		4			2
32	1	6	3	3	2	2
64	1	7	4	3	2	2
128	1	7	4	3	2	2

Estimated number of clusters, and their estimated dimensionalities, as a function of sample size. The VBEM algorithm found two different solutions when $N = 8$. More clusters with larger effective dimensionalities are discovered as the sample sizes increases.

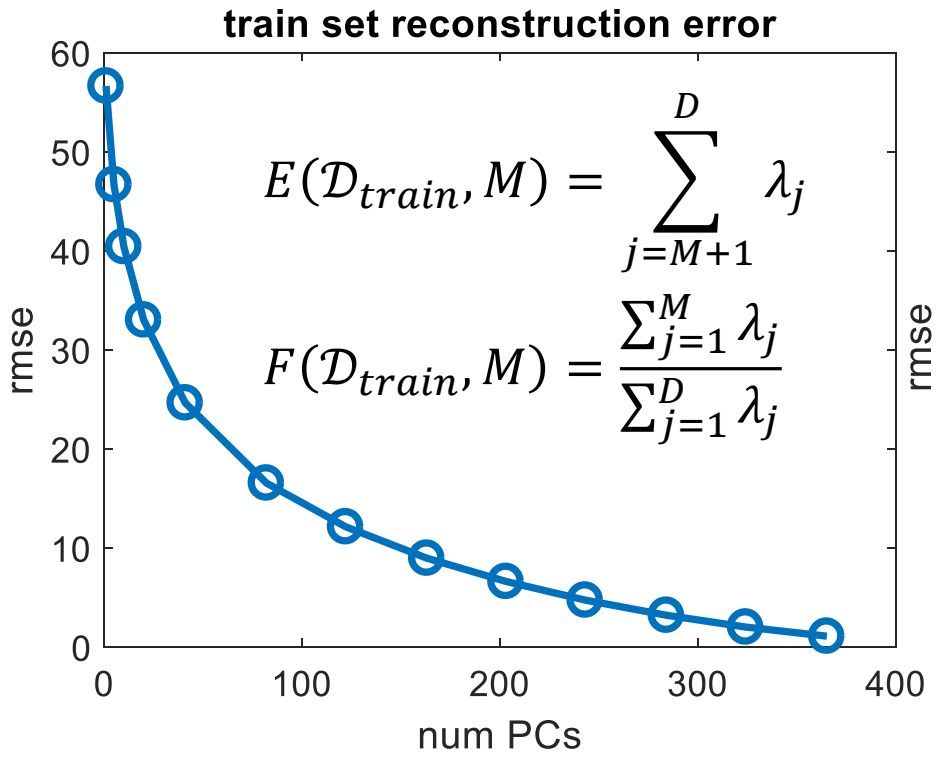
- Beal, M. (2003). *Variational Algorithms for Approximate Bayesian Inference*. Ph.D. thesis, Gatsby Unit.

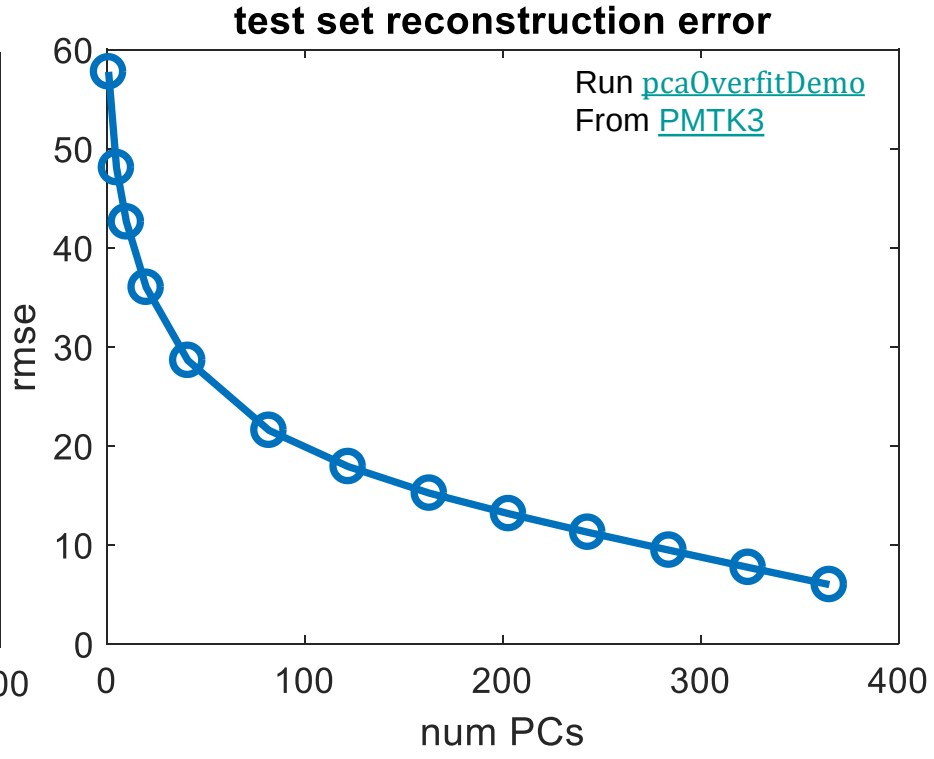
Model Selection for PCA

- Since PCA is not a probabilistic model, we cannot use any of the earlier discussed methods. An option is to use the reconstruction error:

$$E(\mathcal{D}, M) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2$$

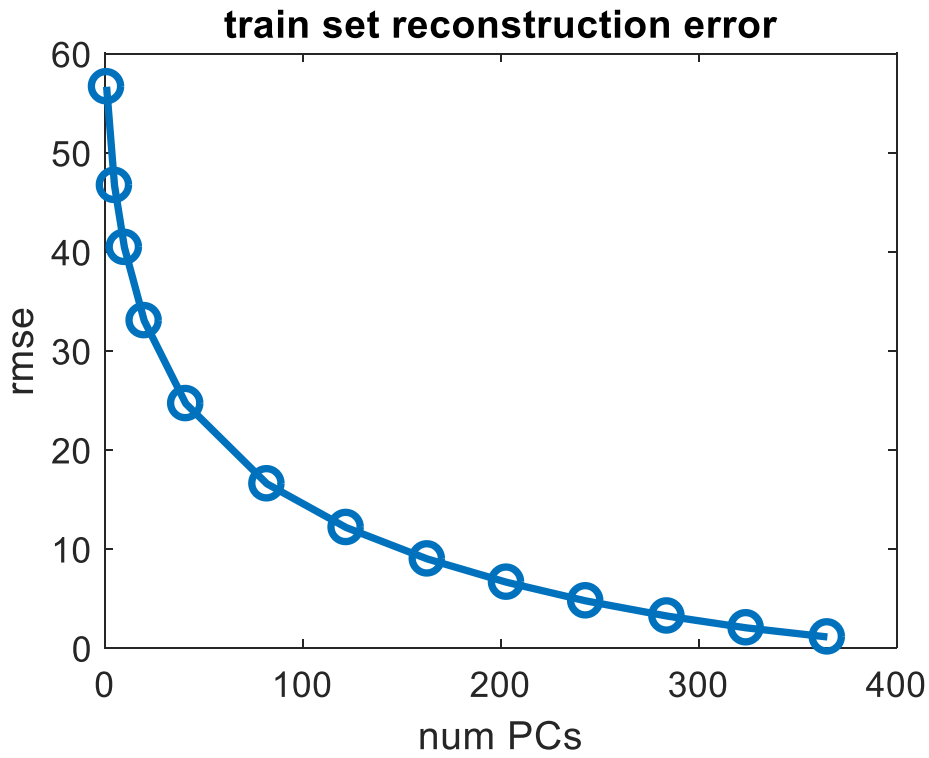
$$\hat{\boldsymbol{x}}_i = \mathbf{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{z}_i = \boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{\mu})$$

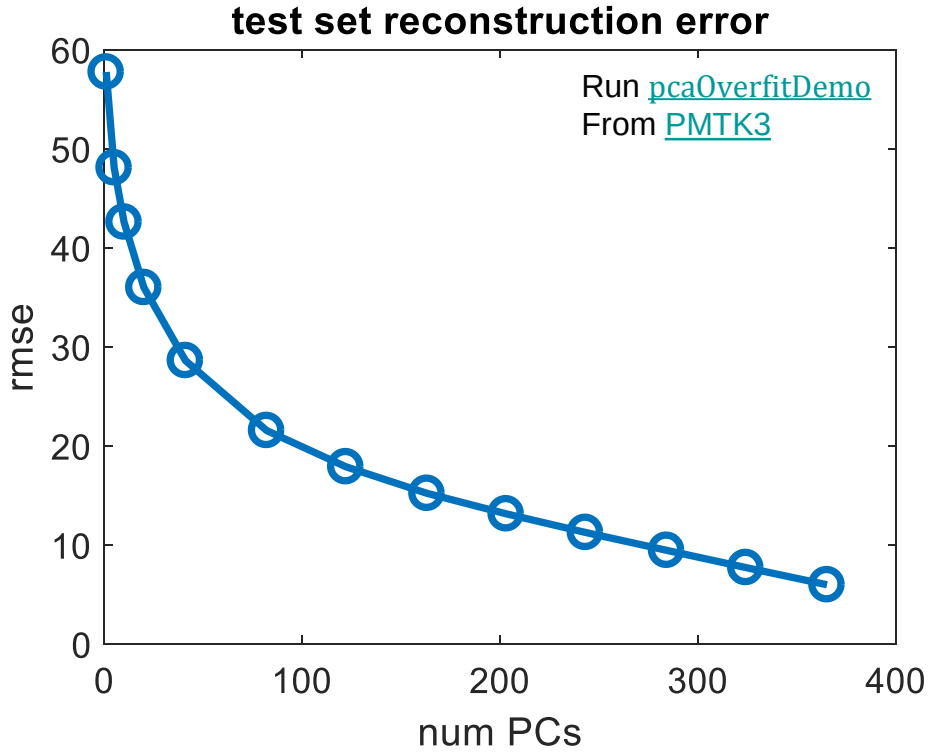




Model Selection for PCA

- If we use $M = rank(\boldsymbol{X})$, we get zero reconstruction error on the training set.
- We see that the test error continues to go down even as the model becomes more complex! Thus we do not get the usual U-shaped curve.
- The problem is that PCA is not a proper generative model of the data (a data compression technique). If you give it more latent dimensions it will be able to approximate the test data more accurately.

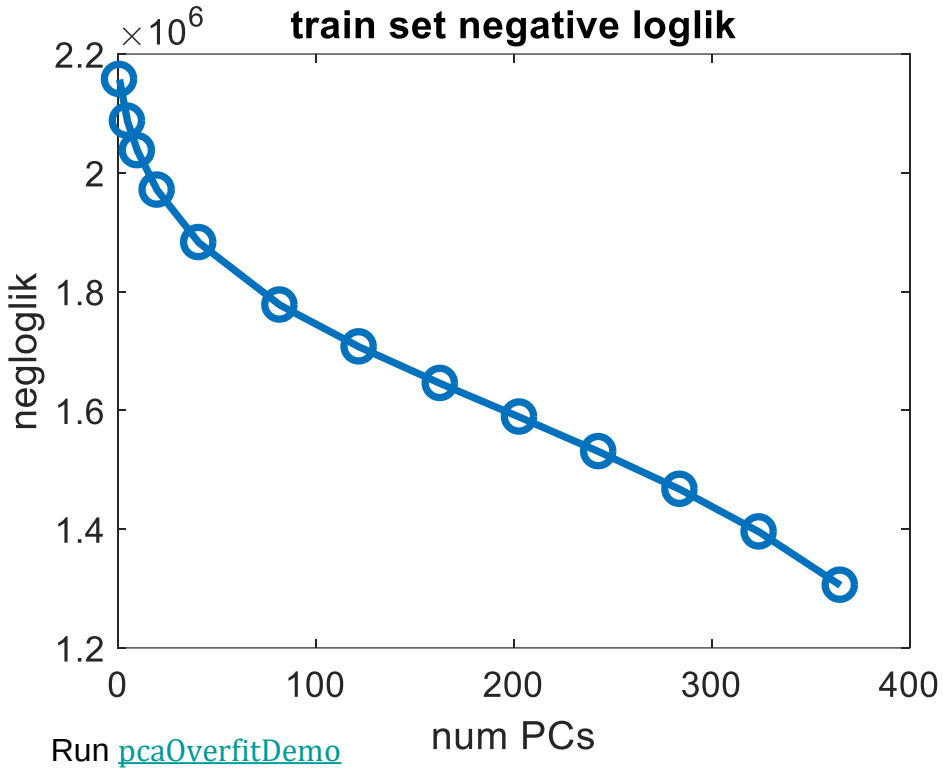


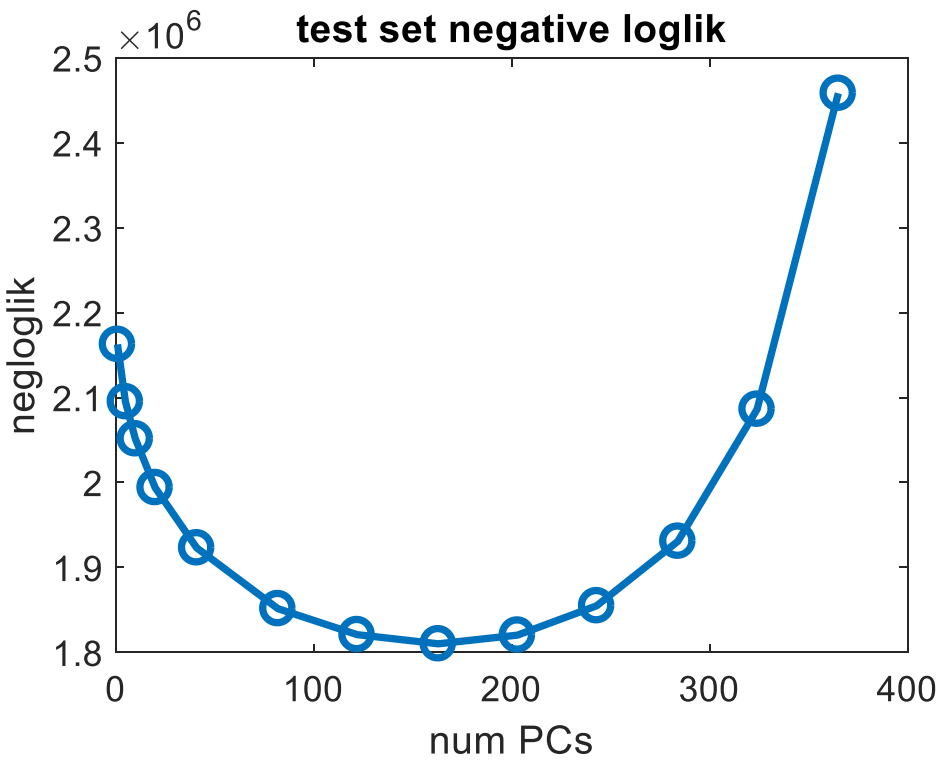


Model Selection for PCA

- When plotting the negative log likelihood on the test set we do indeed see the U shape.

- The nature of these results are similar to the ones for the K-means algorithm (also non-probabilistic method).





Run pcaOverfitDemo
From PMTK3

Profile Likelihood

- Although there is no U-shape, there is sometimes a change in the plots, from relatively large errors to relatively small.
- Let λ_k be a measure of the error incurred by a model of size k, such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{M_{max}}$. In PCA these are the eigenvalues.
- Partitioning these values into two groups, depending on whether $k < M$ or $k > M$, where M is some threshold which we will determine. To measure the quality of M, we will use a simple change-point model, where $\lambda_k \sim \mathcal{N}(\mu_1, \sigma^2)$ if $k \leq M$, and $\lambda_k \sim \mathcal{N}(\mu_2, \sigma^2)$ if $k > M$. (It is important that σ^2 be the same in both models, to prevent overfitting in the case where one regime has less data than the other)
- Within each regime, we assume the λ_k are iid. We can fit this model for each $M = 1 : M_{max}$ by partitioning the data and computing the MLEs, using a pooled estimate of the variance.

- Zhu, M. and A. Ghodsi (2006). Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics & Data Analysis 51*, 918– 930.

Profile Likelihood

- The MLE estimates are given as:

$$\mu_1(M) = \frac{\sum_{k<M} \lambda_k}{M}, \mu_2(M) = \frac{\sum_{k>M} \lambda_k}{N-M}$$

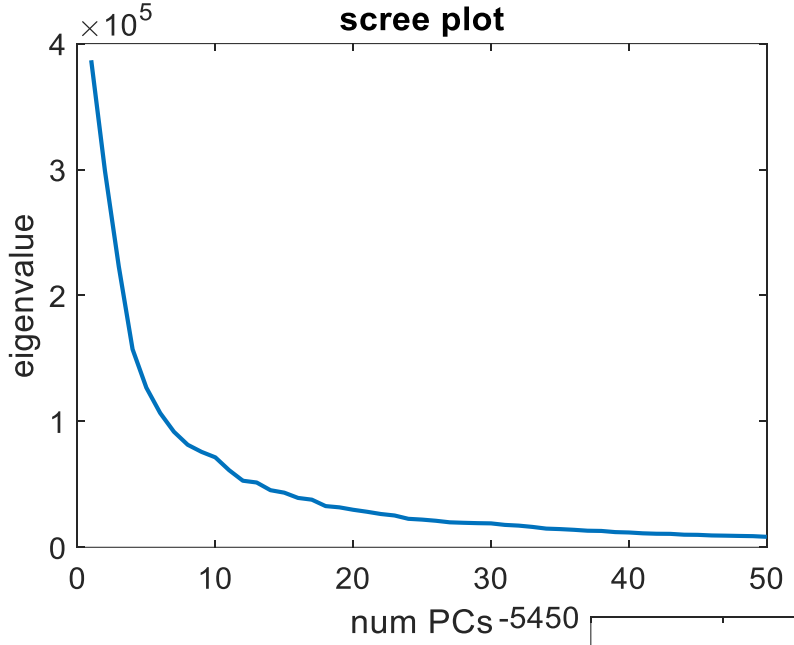
$$\ell(M) = \sum_{k=1}^{M} log\mathcal{N}(\lambda_k|\mu_1(M), \sigma^2(M)) + \sum_{k=M+1}^{K} log\mathcal{N}(\lambda_k|\mu_2(M), \sigma^2(M))$$

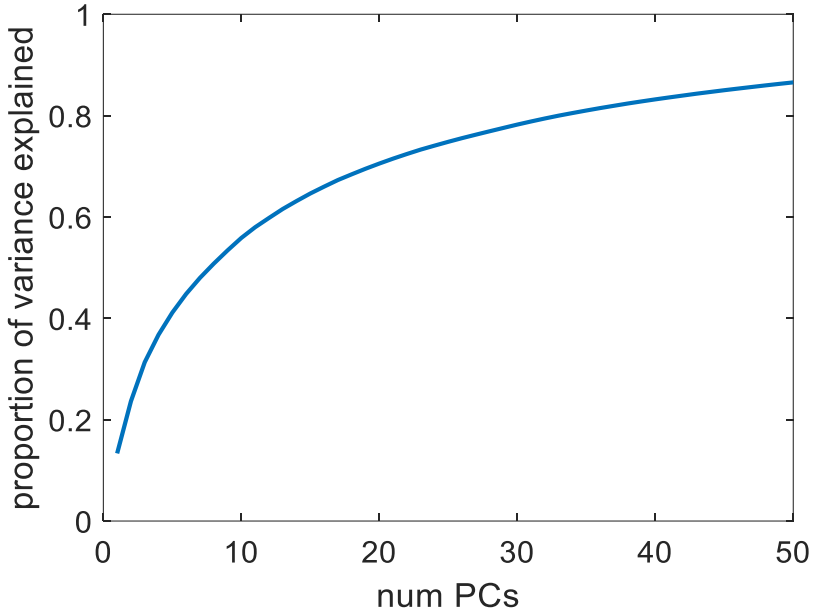
- Finally we choose:

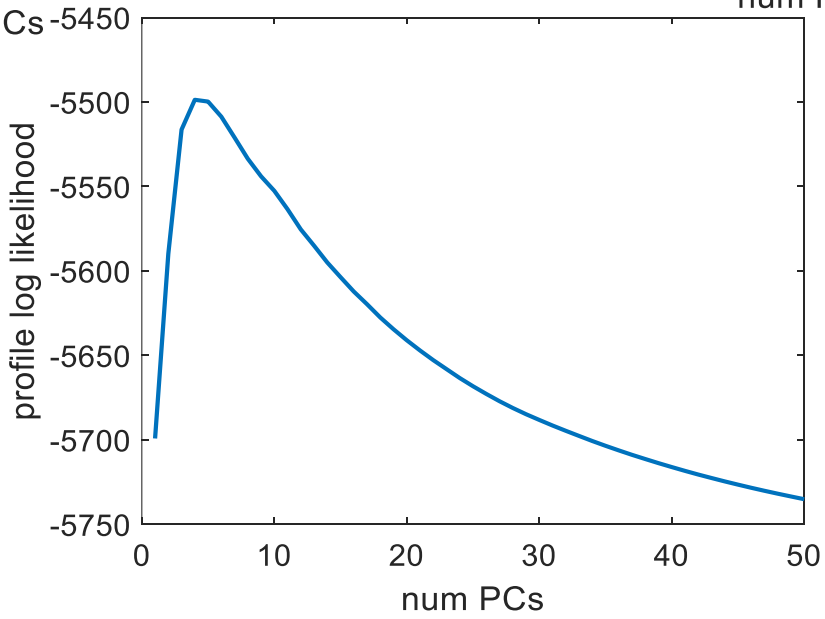
$$M^* = argmax\ell(M)$$

- Zhu, M. and A. Ghodsi (2006). Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics & Data Analysis 51*, 918– 930.

Profile Likelihood







Run *pcaOverfitDemo*
From PMTK3