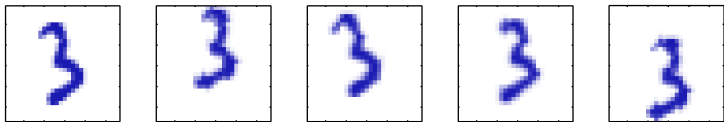


# Continuous Latent Variables

Christian Osendorfer

Technische Universität München



Synthetic data obtained by random translations and rotations of one digit image. Resulting Images are  $100 \times 100$  pixels.

Figure from Bishop's PRML

# Intrinsic dimensionality

10,000 dimensional data space, but only 3 *degrees of freedom*.

Data points live on a 3 dimensional (non-linear) subspace.

Translation and rotation parameters are *latent* variables.

*Distributed representation* (vs. local representation with MoG): all components of latent variable are used at the same time to represent the data.

# Factor Analysis (FA)

$$\mathbf{x} \in \mathbb{R}^d, \quad \mathbf{z} \in \mathbb{R}^l, \quad l \ll d$$

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}) \quad (1)$$

(*simple* latent distribution without loss of generality!)

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Psi}) \quad (2)$$

**W**: factor loading matrix,  $\mathbb{R}^{d \times l}$

**Ψ**: diagonal matrix (*uniquenesses*),  $\mathbb{R}^{d \times d}$

Matrix Factorization

An illustration of the generative process for a special form of Factor Analysis (probabilistic Principal Component Analysis)

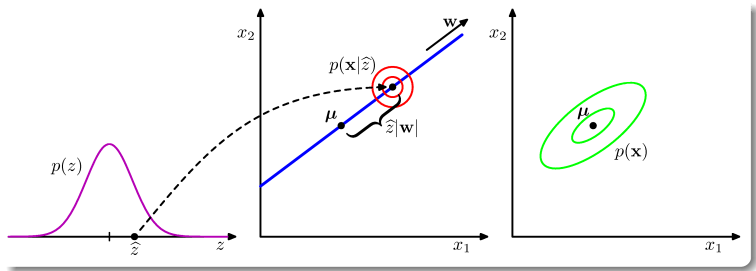


Figure from Bishop's PRML

## Low-rank approximation of a Gaussian

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C}) \quad (3)$$

with

$$\mathbf{C} = \mathbf{W}\mathbf{W}^T + \boldsymbol{\Psi} \quad (4)$$

why? (see *Multivariate Gaussians*)

or ( $\boldsymbol{\psi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$ ):

$$E(\mathbf{X}) = E(\mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\psi}) = \boldsymbol{\mu} \quad (5)$$

$$\text{Cov}(\mathbf{X}) = E((\mathbf{W}\mathbf{z} + \boldsymbol{\psi})(\mathbf{W}\mathbf{z} + \boldsymbol{\psi})^T) = E(\mathbf{W}\mathbf{z}\mathbf{z}^T\mathbf{W}^T + \boldsymbol{\psi}\boldsymbol{\psi}^T) = \mathbf{W}\mathbf{W}^T + \boldsymbol{\Psi} \quad (6)$$

Statistically more efficient than a full covariance Gaussian:  $O(d)$  vs.  $O(d^2)$ .

# Computations

FA defines a density on  $\mathbf{X}$ . Computing the likelihood for a data point  $\mathbf{x}$  requires  $\mathbf{C}^{-1}$  and  $|\mathbf{C}|$ .

$$\mathbf{C}^{-1} = \boldsymbol{\Psi}^{-1} - \boldsymbol{\Psi}^{-1}\mathbf{W}(\mathbf{I}_l + \mathbf{W}^T\boldsymbol{\Psi}^{-1}\mathbf{W})^{-1}\mathbf{W}^T\boldsymbol{\Psi}^{-1} \quad (7)$$

(via Sherman-Morrison-Woodbury formula.  $O(l^3)$ .)

$$|\mathbf{C}| = |\mathbf{I}_l + \mathbf{W}^T\boldsymbol{\Psi}^{-1}\mathbf{W}||\boldsymbol{\Psi}^{-1}| \quad (8)$$

(via matrix determinant lemma.  $O(l^3)$ .)

Computationally more efficient than a full covariance Gaussian.

## Latent factors

Maybe  $\mathbf{x}$  reveals something about the latent  $\mathbf{z}$  ?

$$p(\mathbf{z}_i|\mathbf{x}_i) = \mathcal{N}(\mathbf{z}_i|\mathbf{m}_i, \mathbf{\Sigma}) \quad (9)$$

$$\mathbf{\Sigma} = (\mathbf{I} + \mathbf{W}^T \mathbf{\Psi}^{-1} \mathbf{W})^{-1} \quad (10)$$

$$\mathbf{m}_i = \mathbf{\Sigma}(\mathbf{W}^T \mathbf{\Psi}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})) \quad (11)$$

(Note the missing  $i$  at  $\mathbf{\Sigma}$ ).

$\mathbf{m}_i$  are sometimes called *scores*.

## Parameter estimation

FA is a latent variable model, use EM.

Previous slide: E step!

M step:

$$\boldsymbol{\mu} = \frac{1}{n} \sum_i \mathbf{x}_i \quad (12)$$

$$\mathbf{W} = \left[ \sum_i \tilde{\mathbf{x}}_i E(\mathbf{Z}_i)^T \right] \left[ \sum_i E(\mathbf{Z}_i \mathbf{Z}_i^T) \right]^{-1} \quad (13)$$

with  $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \boldsymbol{\mu}$  and  $E(\mathbf{Z}_i \mathbf{Z}_i^T) = \boldsymbol{\Sigma} + \mathbf{m}_i \mathbf{m}_i^T$ .

$$\boldsymbol{\Psi} = \frac{1}{n} \text{diag}(\mathbf{G}(\mathbf{W})) \quad (14)$$

$$\mathbf{G}(\mathbf{W}) = \sum_i (\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^T + \mathbf{W} E(\mathbf{Z}_i \mathbf{Z}_i^T) \mathbf{W}^T - 2\mathbf{W} E(\mathbf{Z}_i) \tilde{\mathbf{x}}_i^T) \quad (15)$$

# Probabilistic PCA

Constrain FA model:  $\Psi = \sigma^2 \mathbf{I}$  and  $\mathbf{W}$  is orthonormal.

Closed form solution for estimating  $\mathbf{W}$ :

$$\mathbf{W} = \mathbf{L}_l (\mathbf{\Lambda}_l - \sigma^2 \mathbf{I})^{\frac{1}{2}} \mathbf{R} \quad (16)$$

with  $\mathbf{L}_l$  the  $d \times l$  matrix whose columns are the first  $l$  eigenvectors of the empirical covariance matrix  $\mathbf{S}$  and  $\mathbf{\Lambda}_l$  the diagonal matrix of corresponding eigenvalues ( $\mathbf{R}$  is an arbitrary orthogonal matrix).

$$\sigma^2 = \frac{1}{d-l} \sum_{j=l+1}^d \lambda_j \quad (17)$$

PPCA captures (or models) variance along principal directions:  
Standardize data (unit of measurement)!

# FA vs. PPCA

Compare generated samples.

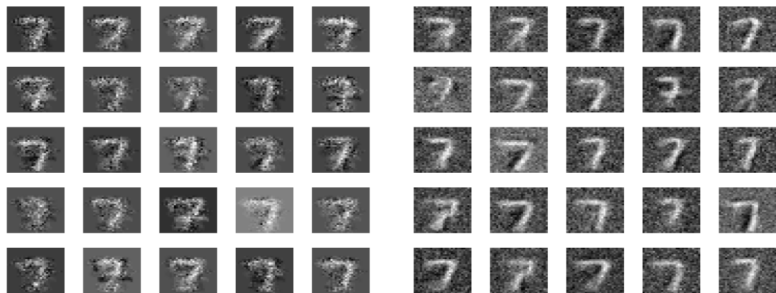
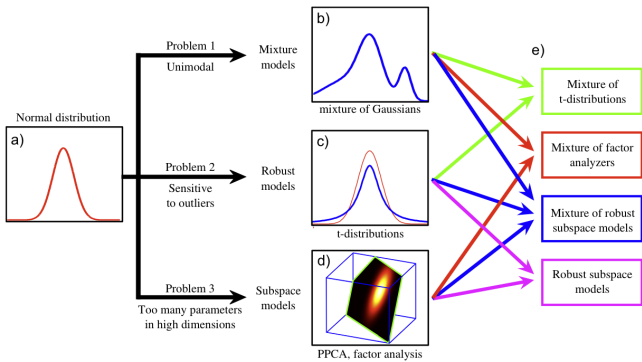


Figure from *Bayesian Reasoning and Machine Learning*, David Barber.



Today: Subspace models.

Figure from *Computer vision: models, learning and inference*, Simon J.D. Prince,

<http://computervisionmodels.blogspot.com/>

# PCA

Find an orthogonal set of  $l$  linear basis functions  $\mathbf{w}_j \in \mathbb{R}^d$  and corresponding low-dimensional projections  $\mathbf{z}_j \in \mathbb{R}^l$  such that the average *reconstruction error* is minimized:

$$J = \frac{1}{N} \sum_i \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \quad (18)$$

with  $\hat{\mathbf{x}}_i = \mathbf{W}\mathbf{z}_i + \boldsymbol{\mu}$ .

Equivalent (non obvious!) formulation: Find directions with *maximal projected variance*.

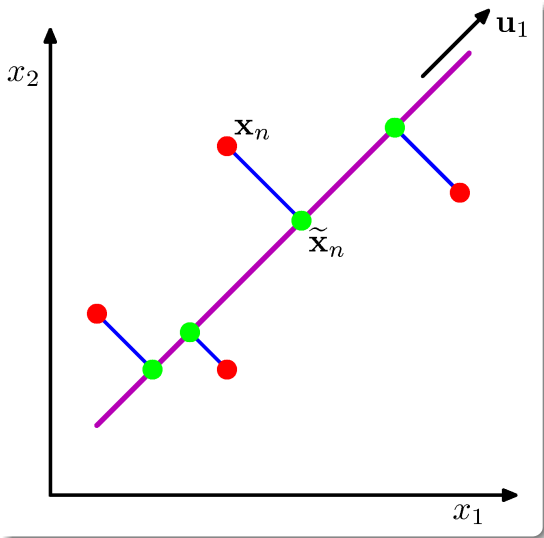


Figure from Bishop's PRML

Solution similar to pPCA:

$$\mathbf{W} = \mathbf{L}_l \quad (19)$$

$\mathbf{L}_l$  is the  $d \times l$  matrix whose columns are the first  $l$  eigenvectors of  $\mathbf{S}$ .

Note:  $\mathbf{L}_l^T \mathbf{L}_l = \mathbf{I}_l$  but not  $\mathbf{L}_l \mathbf{L}_l^T = \mathbf{I}$ , unless  $d = l$ .

$$J = \sum_{i=l+1}^d \lambda_i \quad (20)$$

What should be the dimensionality of the latent embedding?

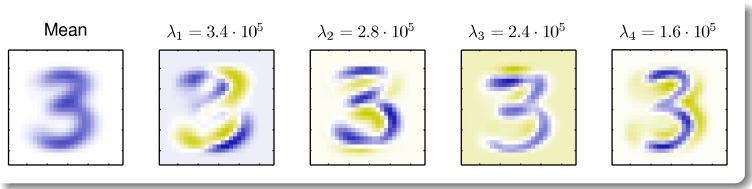


Figure from Bishop's PRML

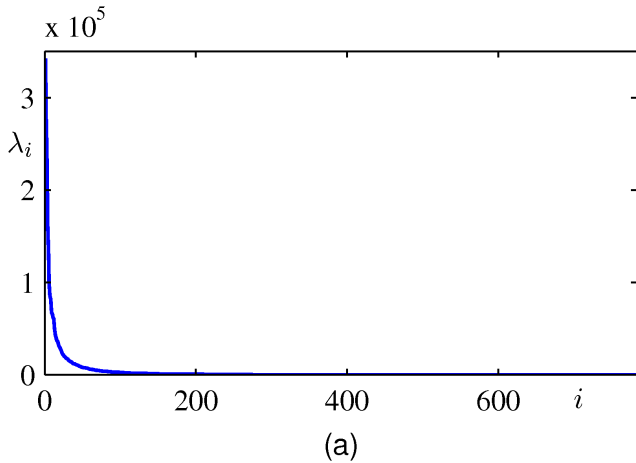


Figure from Bishop's PRML

# Eigenfaces

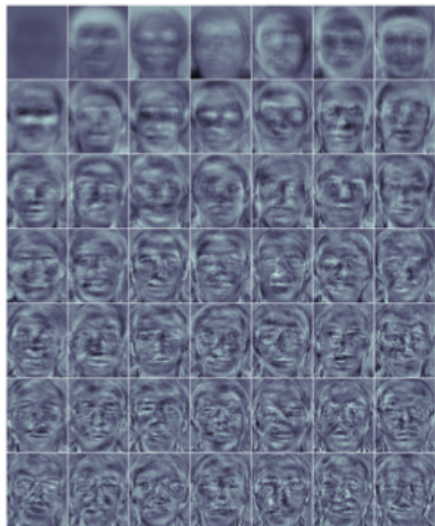


Figure from *Bayesian Reasoning and Machine Learning*, David Barber.

# LSI – Latent Semantic Indexing

$d$  is the number of words in a vocabulary  $\Rightarrow$   $\mathbf{x}$  represents a specific document (by word counts).

$$\mathbf{X} \cong \mathbf{Z} \times \mathbf{U} \quad (21)$$

$\mathbf{X}$  –  $n \times d$  matrix of word counts

$\mathbf{Z}$  –  $n \times l$  matrix latent document representation

$\mathbf{U}$  –  $l \times d$  matrix of  $l$  *eigendocuments*

Problem: Eigendocuments contain negative frequencies. Alternative methods (e.g. NMF) lead to more interpretable results.

Solve by SVD.

# Whitening/ZCA

PCA is often used as a preprocessing step.

$$\mathbf{z} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{L}_l^T (\mathbf{x} - \boldsymbol{\mu}) \quad (22)$$

$\mathbf{z}$  is *whitened*, that is, the empirical covariance matrix is  $\mathbf{I}$ .

*Zero Component Analysis (ZCA)* stays in the original domain:

$$\hat{\mathbf{x}} = \mathbf{L}_d (\mathbf{\Lambda} + \epsilon \mathbf{I})^{-\frac{1}{2}} \mathbf{L}_d^T (\mathbf{x} - \boldsymbol{\mu}) \quad (23)$$

# ICA – Independent Component Analysis

Generative model:

- ▶ some (unknown) data  $\mathbf{s} \in R^n$  ( $n$  independent *sources*).
- ▶ unknown *mixing* matrix  $\mathbf{A} \in R^{n \times n}$ .
- ▶ observed  $\mathbf{x} \in R^n$ :  $\mathbf{x} = \mathbf{A}\mathbf{s}$

Given  $m$  observations  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ , can we recover the  $\mathbf{s}$ 's? Yes, if we would know  $\mathbf{W} = \mathbf{A}^{-1}$ .

# ICA – Ambiguities

- ▶ Permutations
- ▶ Scaling
- ▶ Sources  $\mathbf{s}_j$  must be *non-gaussian*.

Distribution of  $\mathbf{S}$  must be non-gaussian. What does this mean for  $\mathbf{X}$ ?

$$p_{\mathbf{X}}(\mathbf{x}) = p_{\mathbf{S}}(\mathbf{W}\mathbf{x})|\mathbf{W}| = \prod_i p_{\mathbf{S}_i}(\mathbf{w}_i^T \mathbf{x})|\mathbf{W}|$$

where  $\mathbf{w}_i^T$  is the  $i$ -th row of  $\mathbf{W}$ .

## ICA – MLE

We can do MLE on  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$  if we specify densities for the individual sources (standard examples would be  $p_{s_i}(s) \approx \sigma'(s)$  or  $p_{s_i}(s) \approx -\tanh'(y)$ ).

The loglikelihood then is

$$\ell(\mathbf{W}) = \sum_{i=1}^m \sum_{j=1}^n \log p_{s_i}(\mathbf{w}_i^T \mathbf{x}) + \log |\mathbf{W}|$$

There is no closed form solution because of the involved non-linearities!  
We could find  $\mathbf{W}$  by e.g. gradient descent.

$$\nabla_{\mathbf{W}} \ell(\mathbf{W}) = \sum_{i=1}^m \left( \begin{pmatrix} 1 - 2g_1(\mathbf{w}_1^T \mathbf{x}_i) \\ 1 - 2g_2(\mathbf{w}_2^T \mathbf{x}_i) \\ \dots \\ 1 - 2g_n(\mathbf{w}_n^T \mathbf{x}_i) \end{pmatrix} \mathbf{x}_i^T + \mathbf{W}^{-T} \right)$$

$g_i$  is the antiderivative (*Stammfunktion* in German) of  $p_{s_i}$ .

# ICA and neural networks

Standard ICA requires an orthonormality constraint, which makes it difficult to learn overcomplete features, in particular it does not scale to large datasets.

Reconstruction ICA (RICA)

$$\frac{\lambda}{m} \sum_{i=1}^m \left( \|\mathbf{W}^T \mathbf{W} \mathbf{x}_i - \mathbf{x}_i\| \right) + \sum_{i=1}^m \sum_{j=1}^k \log g_j(\mathbf{w}_j^T \mathbf{x}_i)$$

see *ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning*, by Quoc V. Le et al., NIPS 2011.