Linear Models: PCA, FA

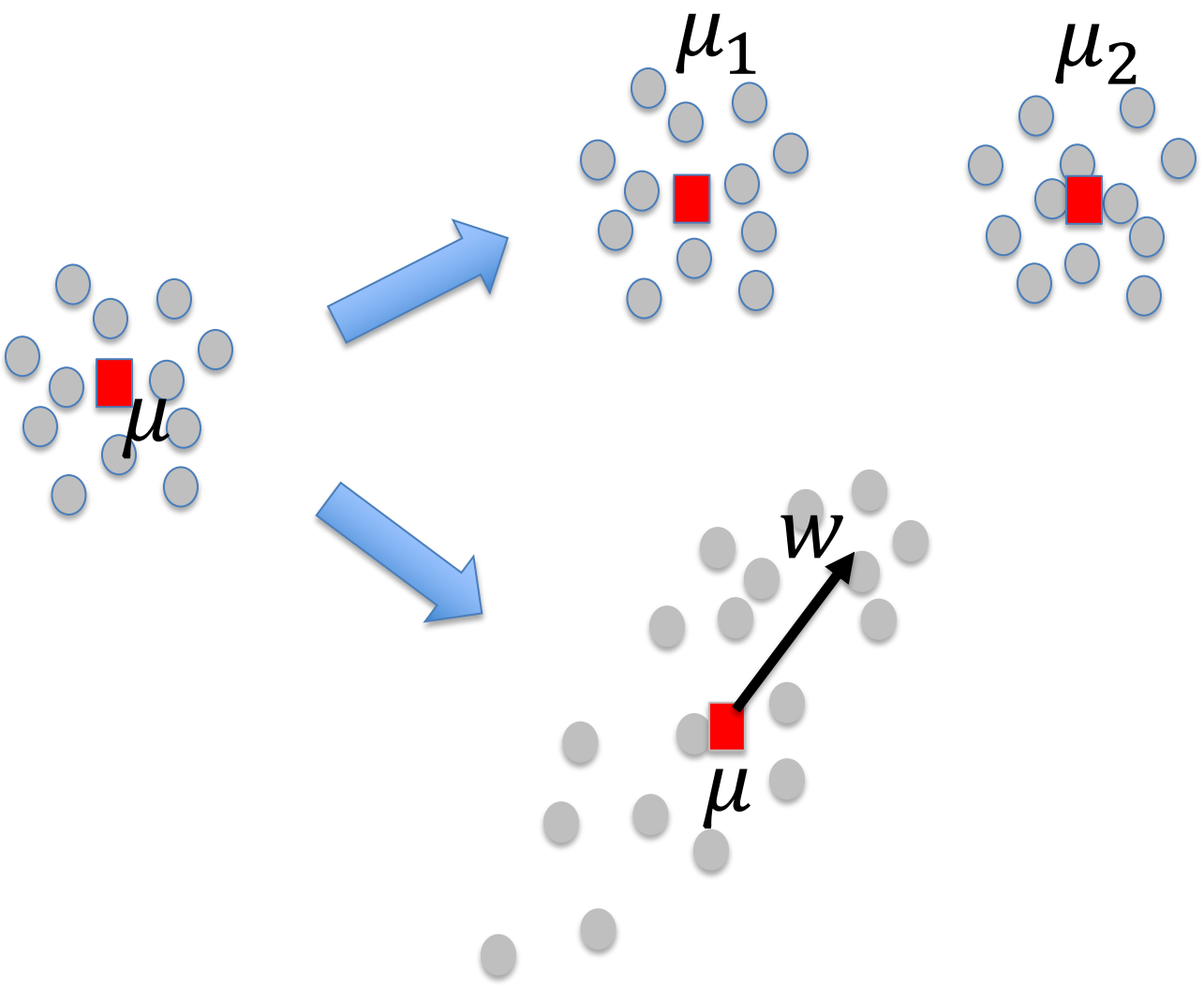
Shikui Tu

2018-03-22

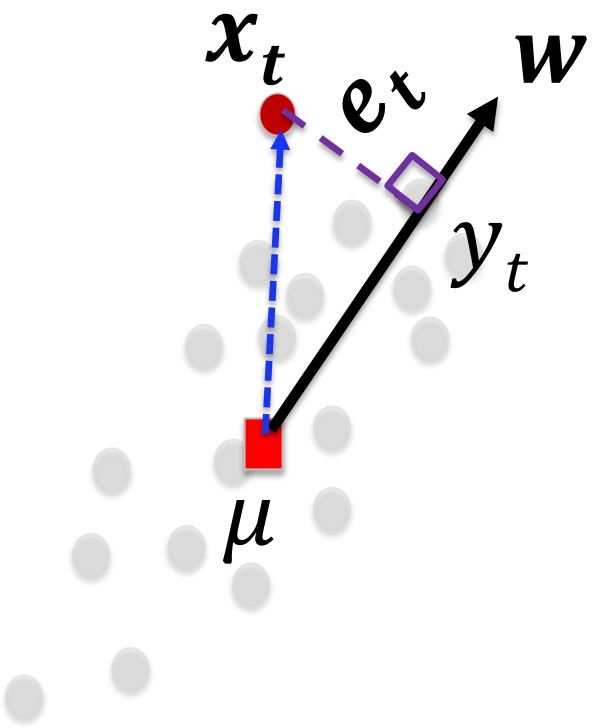
Outline

- Principal Component Analysis (PCA)
- From matrix factorization perspectives
- Hebbian learning, LMSER and PCA
- Probabilistic PCA, Factor Analysis (FA)

Model from “one point” to “one line”



Define the error



$$||w|| = 1$$

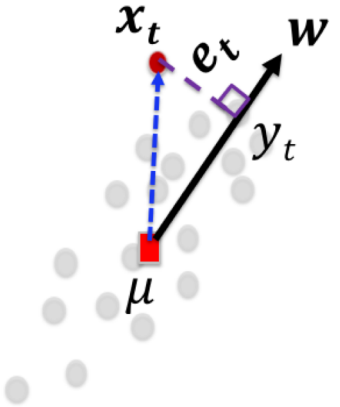
$$y_t = x_t^T w$$

$$e_t = ||x_t - y_t w||^2$$

$$J(w) = \frac{1}{N}\sum_{t=1}^{N} ||x_t - (x_t^T w)w||^2$$

Mean Square Error (MSE)

$$J(w) = \frac{1}{N}\sum_{t=1}^{N} ||\boldsymbol{x_t} - (\boldsymbol{x_t^T w})\boldsymbol{w}||^2$$



$$\boldsymbol{x_t^T x_t - (x_t^T w)w^T x_t - x_t^T (x_t^T w)w + (x_t^T w)w^T (x_t^T w)w}$$
$$\boldsymbol{= x_t^T x_t - w^T (x_t x_t^T) w}$$

$$||w|| = 1$$

Introduce a Lagrange multiplier λ

$$L(\{\boldsymbol{x_t}\}, \boldsymbol{w}) = J(\{\boldsymbol{x_t}\}, \boldsymbol{w}) - \lambda \cdot (\boldsymbol{w^T w} - 1)$$

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} - \lambda \cdot \frac{\partial (\boldsymbol{w^T w} - 1)}{\partial \boldsymbol{w}} = -2(\Sigma_x \boldsymbol{w}) - \lambda \cdot 2\boldsymbol{w} = \boldsymbol{0}$$

$$\Sigma_x = \frac{1}{N}\sum_{t=1}^{N} \boldsymbol{x_t x_t^T}$$

$$\boldsymbol{\Sigma_x w = (-\lambda) \cdot w}$$

Eigenvalues and Eigenvectors

Lagrange multiplier

From wiki

maximize $f(x, y)$
subject to $g(x, y) = 0$

Lagrange multiplier λ

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda \cdot g(x, y)$$

$$\nabla_{x,y} f = \lambda \nabla_{x,y} g$$

$$\nabla_{x,y} f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

$$\nabla_{x,y} g = \left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y} \right)$$

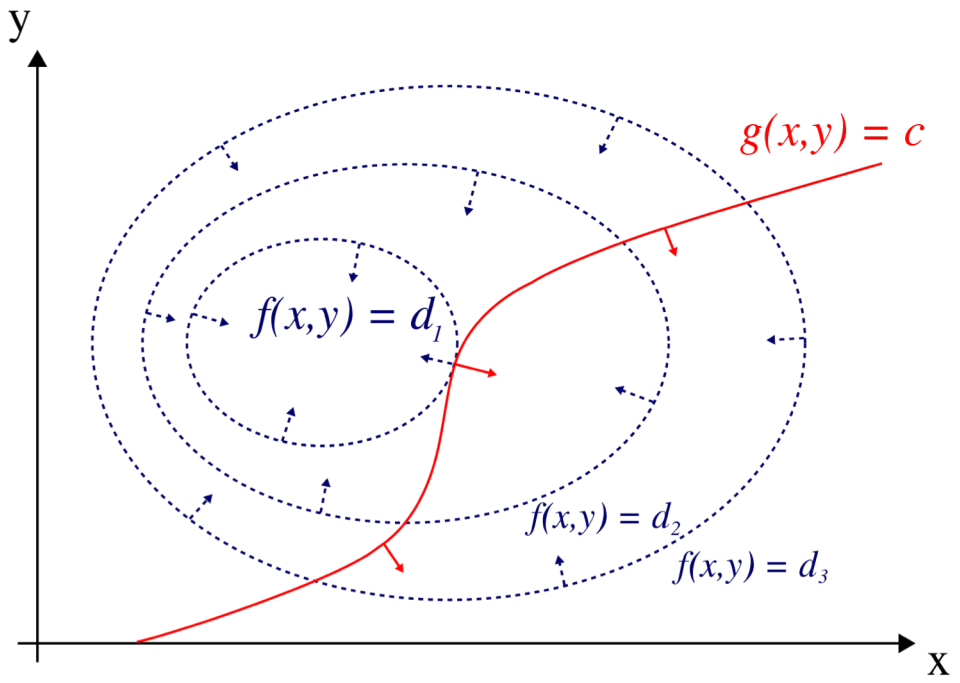
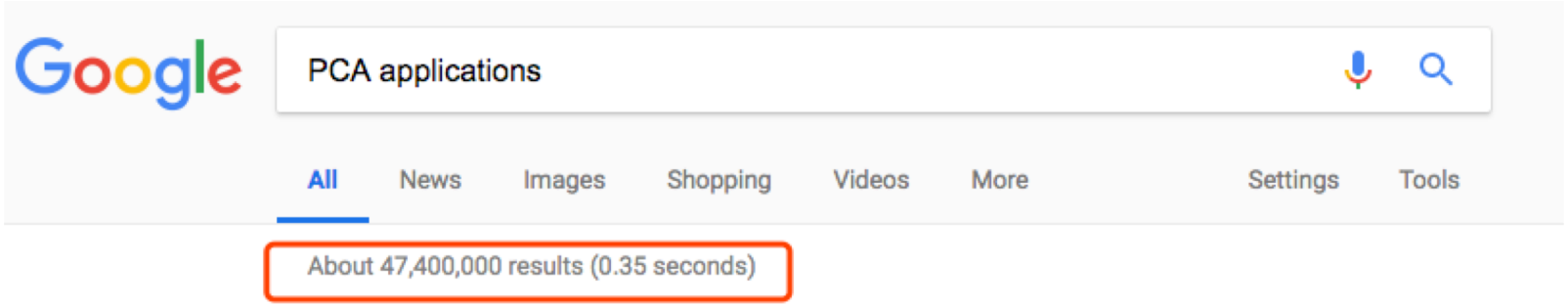


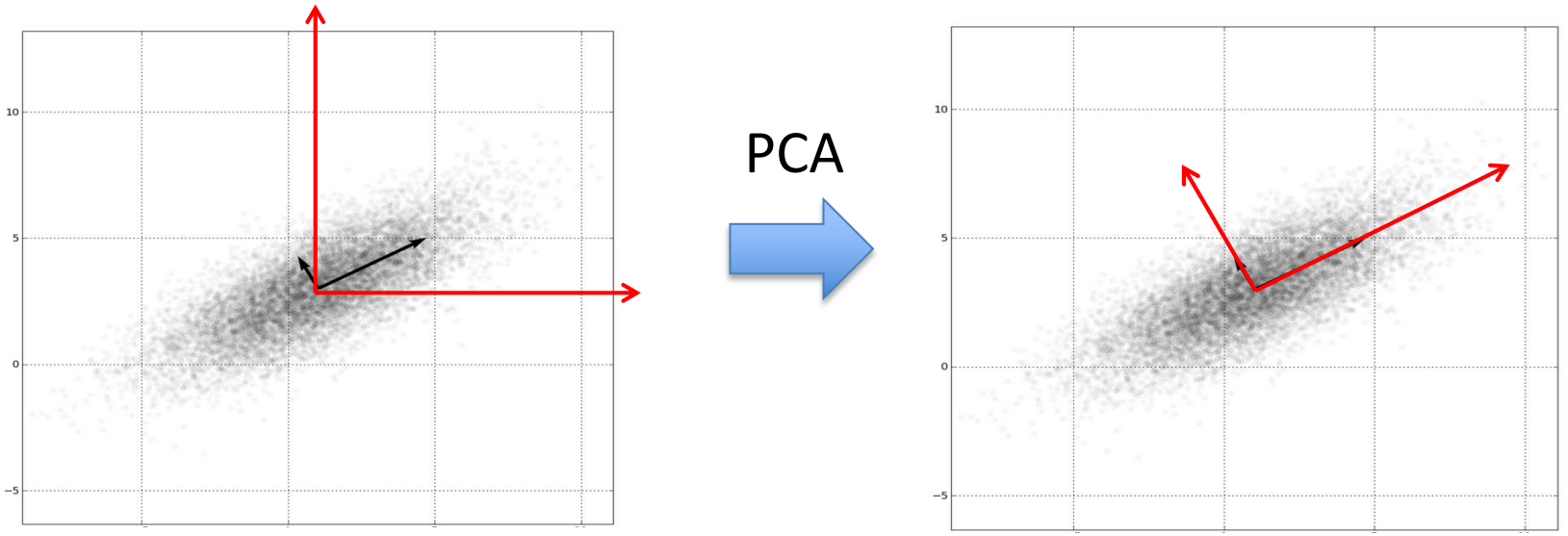
Figure 1: The red line shows the constraint $g(x, y) = c$. The blue lines are contours of $f(x, y)$. The point where the red line tangentially touches a blue contour is the maximum of $f(x, y)$, since $d_1 > d_2$.

Applications of PCA

- Popular in multivariate statistics, signal processing
- Reduce data noise, redundancy, correlation, …
- Dimensionality reduction
- Feature selection and feature extraction
- Data visualization (to 2D or 3D)

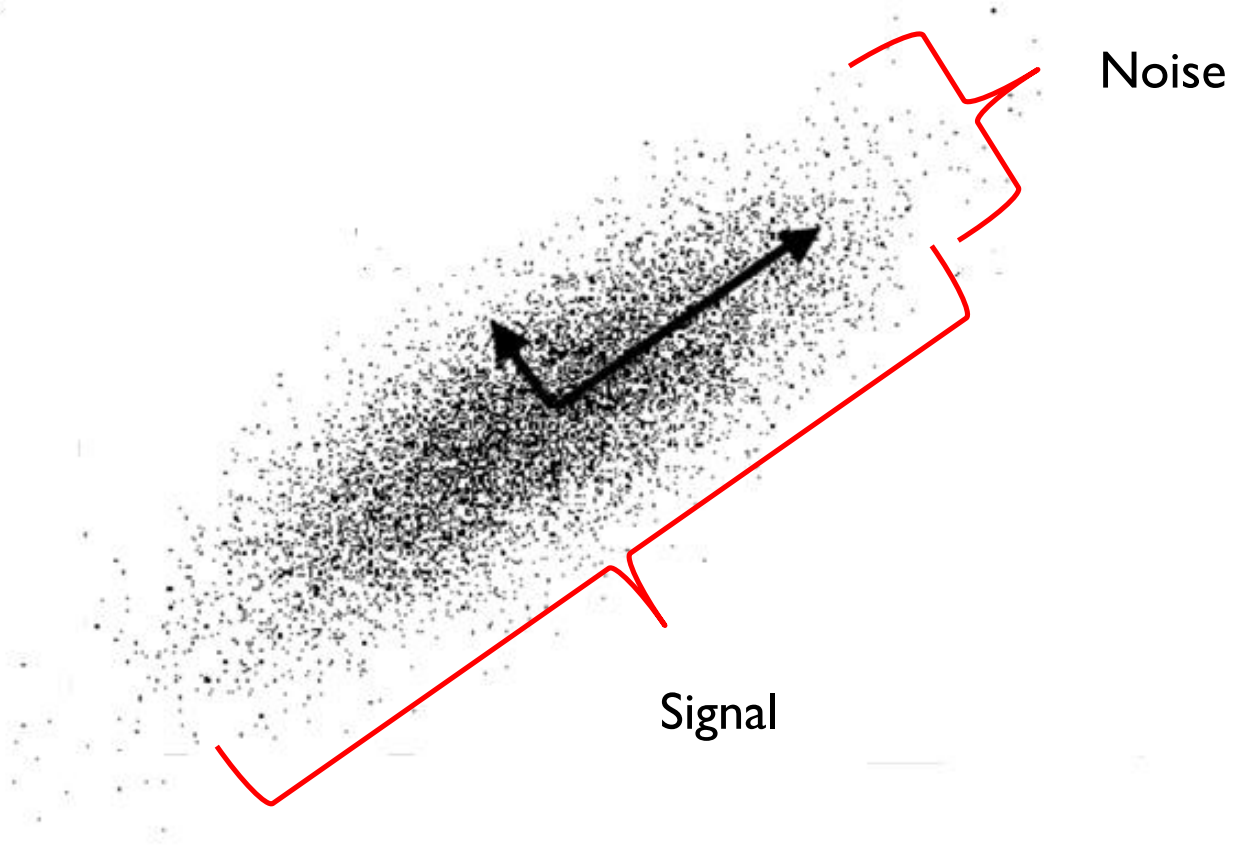


Data correlation



Correlation can be removed by rotating the coordinates to principal components.

Signal-noise ratio maximization



Keep one signal dimension, discard one noisy dimension.

Information redundancy

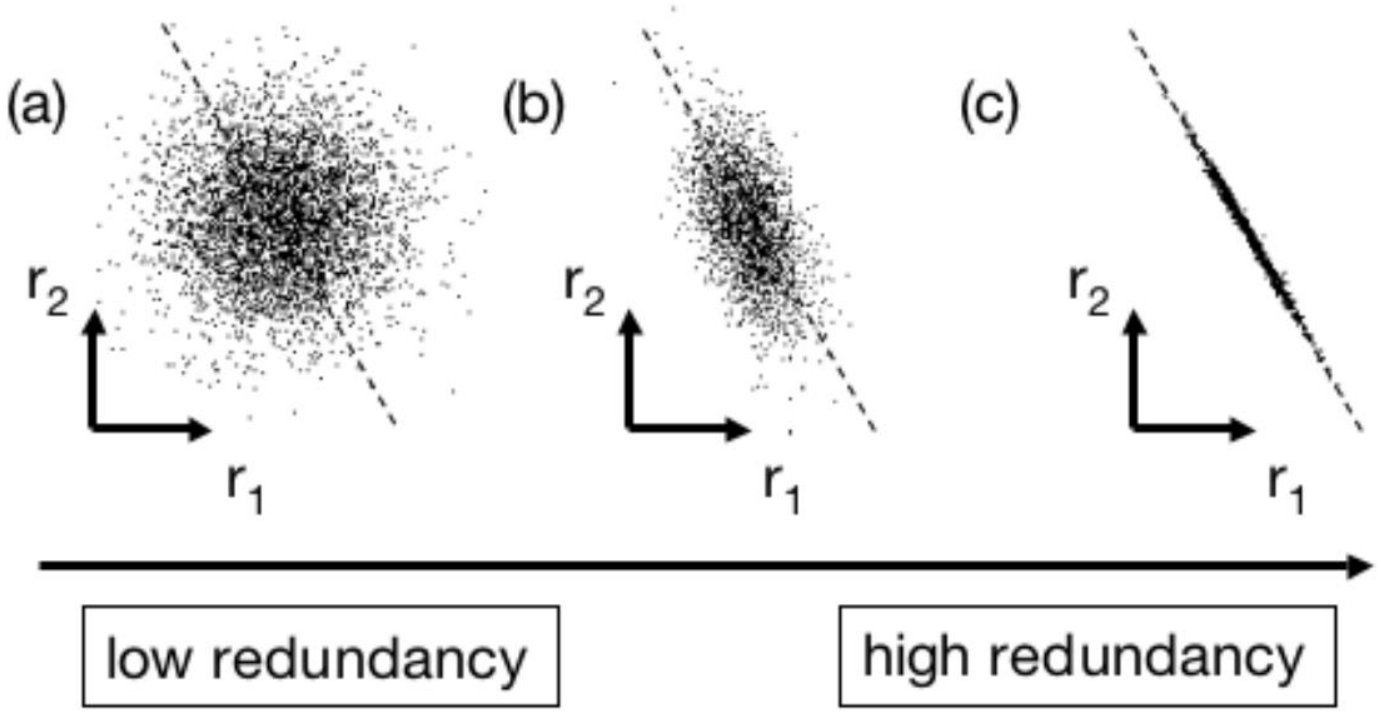


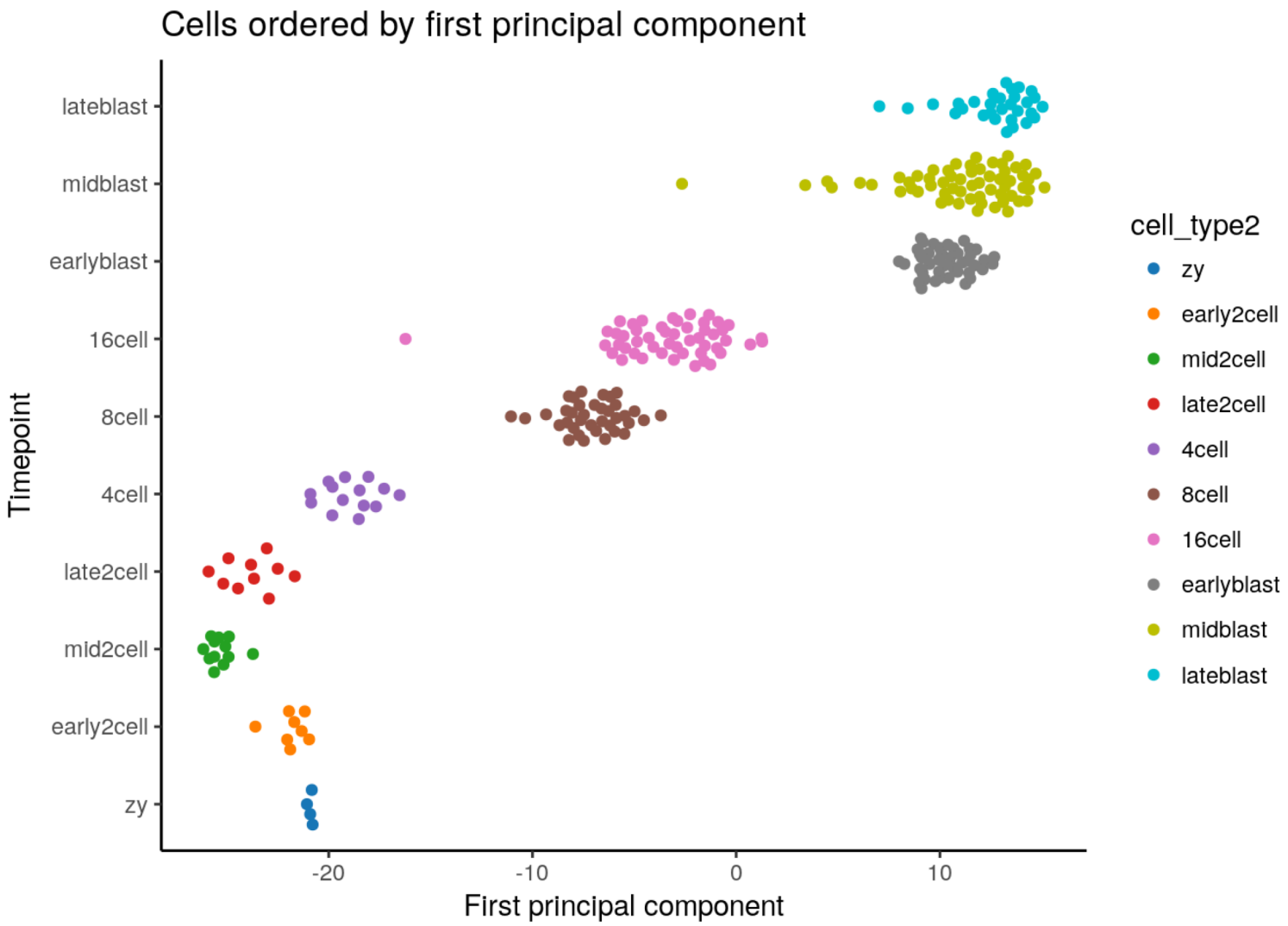
Image denoising by PCA

(a) Noisy image (b) PGPCA (PSNR=33.6) (c) PLPCA (PSNR=34.8)

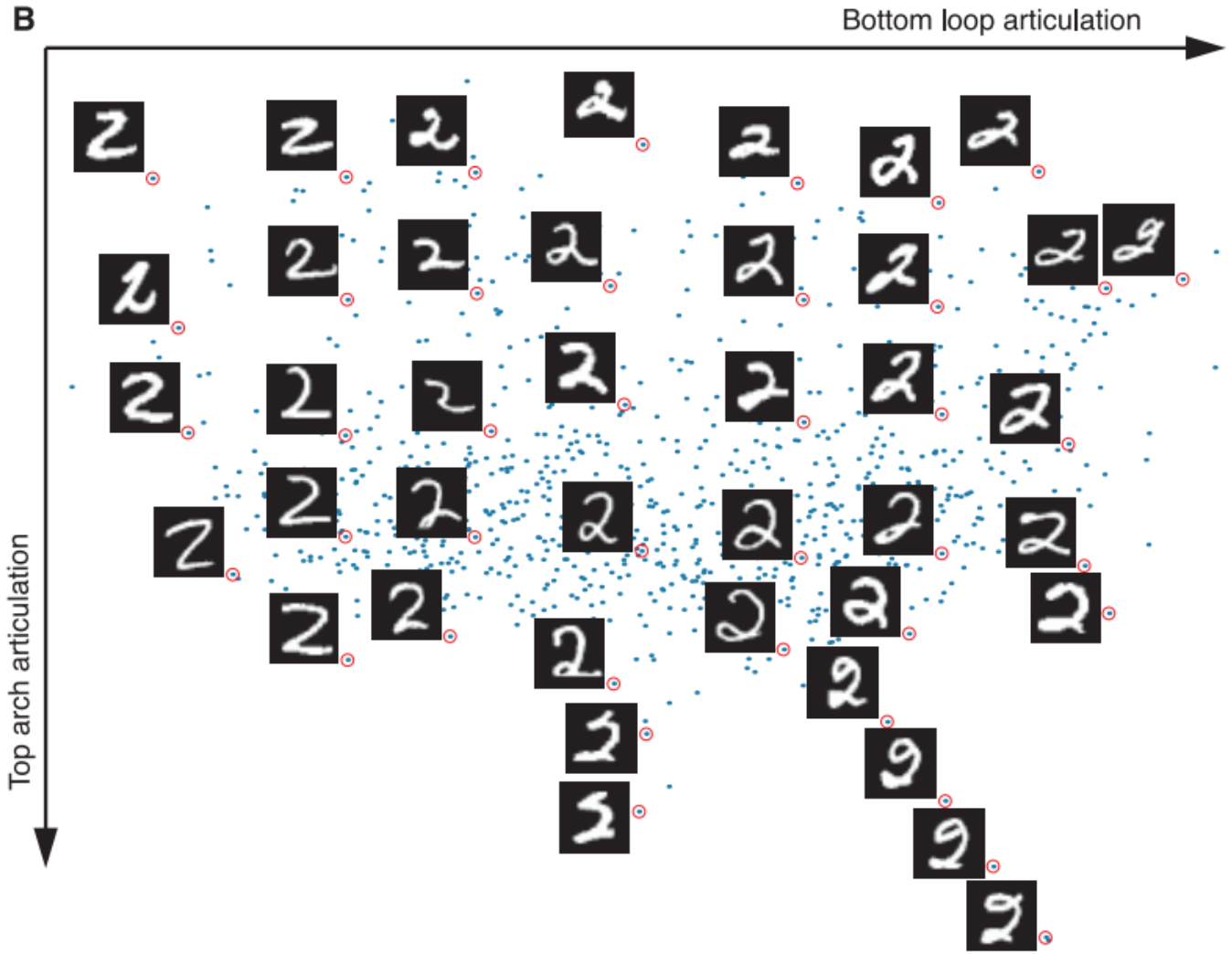
Figure 6: Visual evaluation of the denoising performance of PGPCA and PLPCA on an image (Barbara) damaged by an AWGN with noise level $\sigma = 10$, with their PSNR.

Charles-Alban Deledalle, Joseph Salmon, Arnak Dalalyan; BMVC 2011

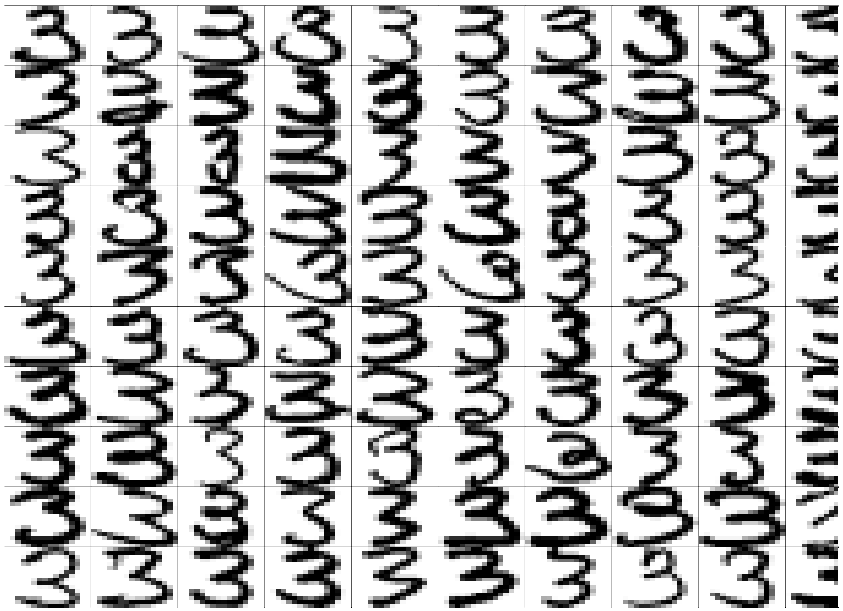
Order the cells by PCA



Visualize the hand-written digits



PCA of hand-written digits



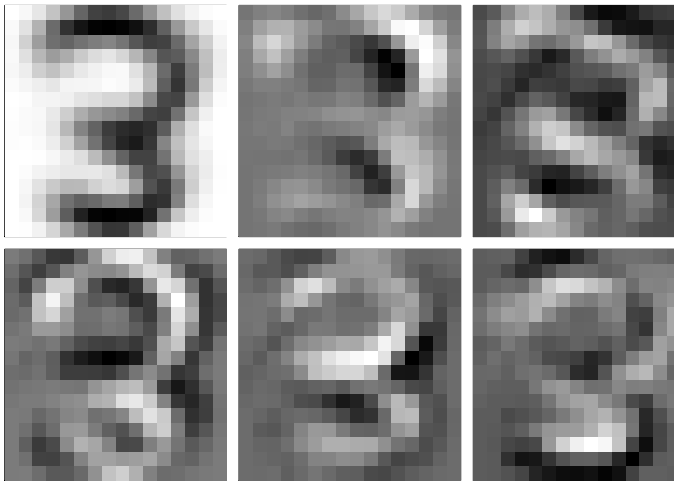


Figure 4: Digits: mean and eigenvectors

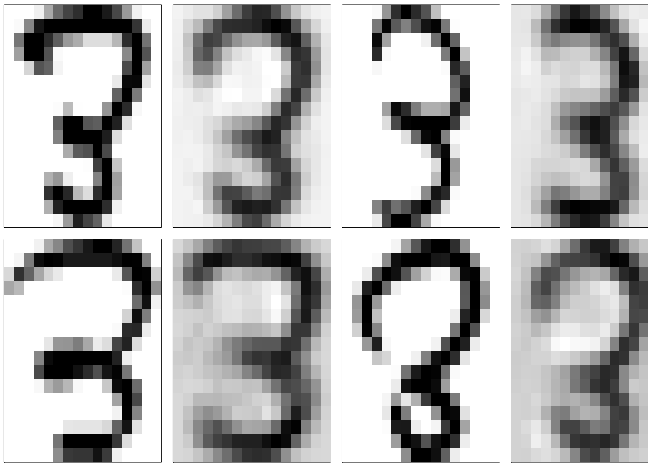


Figure 5: Digits data: Top: digits. Bottom: their reconstructions.

Eigen-face method

- Sirovich and Kirby (1987) showed that PCA could be used on a collection of face images to form a set of basis features.

- Given input image vector $U \in \Re^n$, the mean image vector from the database M, calculate the weight of the kth eigenface as:

  $$w_k = V_k^T (U - M)$$

  Then form a weight vector $W = [w_1, w_2, \ldots, w_k, \ldots, w_n]$

- Compare W with weight vectors W_m of images in the database. Find the Euclidean distance.

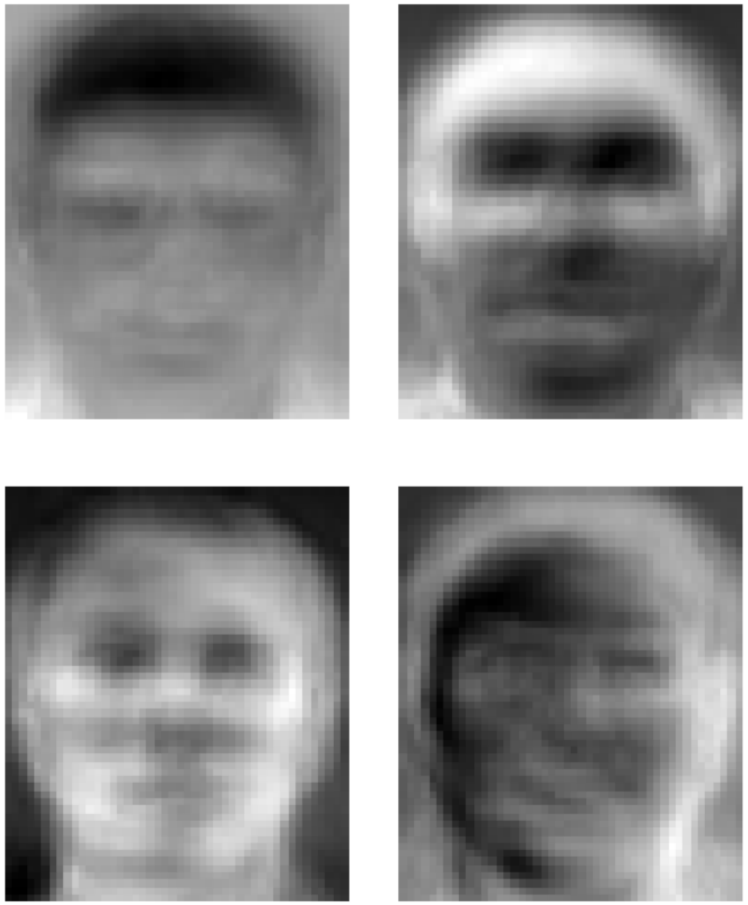
  $$d = ||W - W_m||^2$$

- If $d < \epsilon_1$, then the mth entry in the database is a candidate of recognition.
- If $\epsilon_1 < d < \epsilon_2$, then U may be an unknown face and can be added to the database.
- If $d > \epsilon_2, U$ is not a face image.

https://en.wikipedia.org/wiki/Eigenface

Eigen-face examples

Eigen-face



Reconstruction by top-k eigenvectors

k=2 k=15 k=40

Some eigenfaces from AT&T Laboratories Cambridge

Outline

- Principal Component Analysis (PCA)

- From matrix factorization perspectives

- Hebbian learning, LMSER and PCA

- Probabilistic PCA, Factor Analysis (FA)

From matrix factorization perspectives

- Eigen-decomposition

$$\Sigma_x \approx U \Lambda U^T$$

$$U = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_k]^{\mathrm{T}}_{d \times N}$$

$$\Lambda = \mathrm{diag}[\lambda_1, \dots, \lambda_k]$$

Covariance matrix (assume $\boldsymbol{x}_t$ zero mean):

$$\Sigma_x = \frac{1}{N} \sum_{t=1}^{N} \boldsymbol{x}_t \boldsymbol{x}_t^T = \frac{1}{N} X X^T$$

$$X = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N]_{d \times N}$$

- Singular Value Decomposition (SVD)

$$X = U D V^T$$

$$X X^T = U D V^T \cdot V D U^T = U D^2 U^T$$

PCA and SVD

Data: *n* points in *p*-dim: $X = (x_1, x_2, \cdots, x_n)$

Covariance $C = XX^T = \sum_{k=1}^{p} \lambda_k u_k u_k^T$

Gram (kernel) matrix $X^T X = \sum_{k=1}^{r} \lambda_k v_k v_k^T$

Principal directions: u_k
(Principal axis,subspace)

Principal components: v_k
(projection on the subspace)

Underlying basis: SVD $X = \sum_{k=1}^{p} \sigma_k u_k v_k^T = U \Sigma V^T$

Methods of PCA utilization

Principal components (uncorrelated random variables):

$$X = (x_1, x_2, \cdots, x_n)$$

$$u_k = u_k(1) \cdot X_1 + \cdots + u_k(d) \cdot X_d$$

Dimension reduction: $$X = \sum_{k=1}^{p} \sigma_k u_k v_k^T = U \Sigma V^T$$

Projection to low-dim subspace

$$\tilde{X} = U^T X \qquad U = (u_1, \cdots, u_k)$$

Sphereing the data

Transform data to N(0,1)

$$\tilde{X} = C^{-1/2} X = U \Sigma^{-1} U^T X$$

From PCA to spectral clustering

Consider the kernel matrix: $W_{ij} = \left\langle \phi(x_i), \phi(x_j) \right\rangle$

In Kernel PCA we compute eigenvector: $Wv = \lambda v$

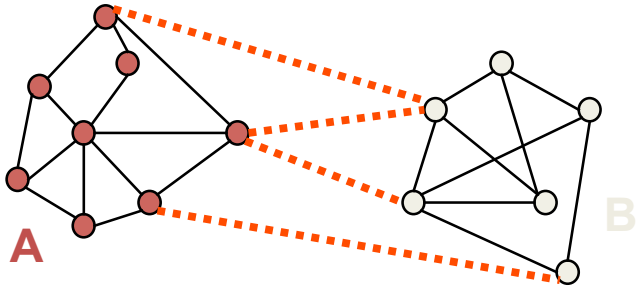
Generalized Eigenvector: $Wq = \lambda Dq$

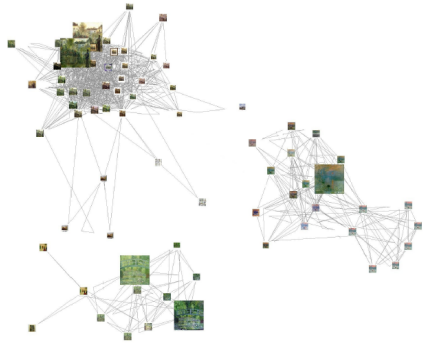
$$D = diag(d_1, \cdots, d_n) \qquad d_i = \sum_j w_{ij}$$

This leads to Spectral Clustering !

Spectral clustering

Group data points based on links in a graph:





Graph Laplacian matrix *L:*

$$L = D - W.$$

$$L_{\text{sym}} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$
$$L_{\text{rw}} := D^{-1} L = I - D^{-1} W.$$

Unnormalized spectral clustering

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k eigenvectors $u_1, \ldots, u_k$ of L.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | \, y_j \in C_i\}$.

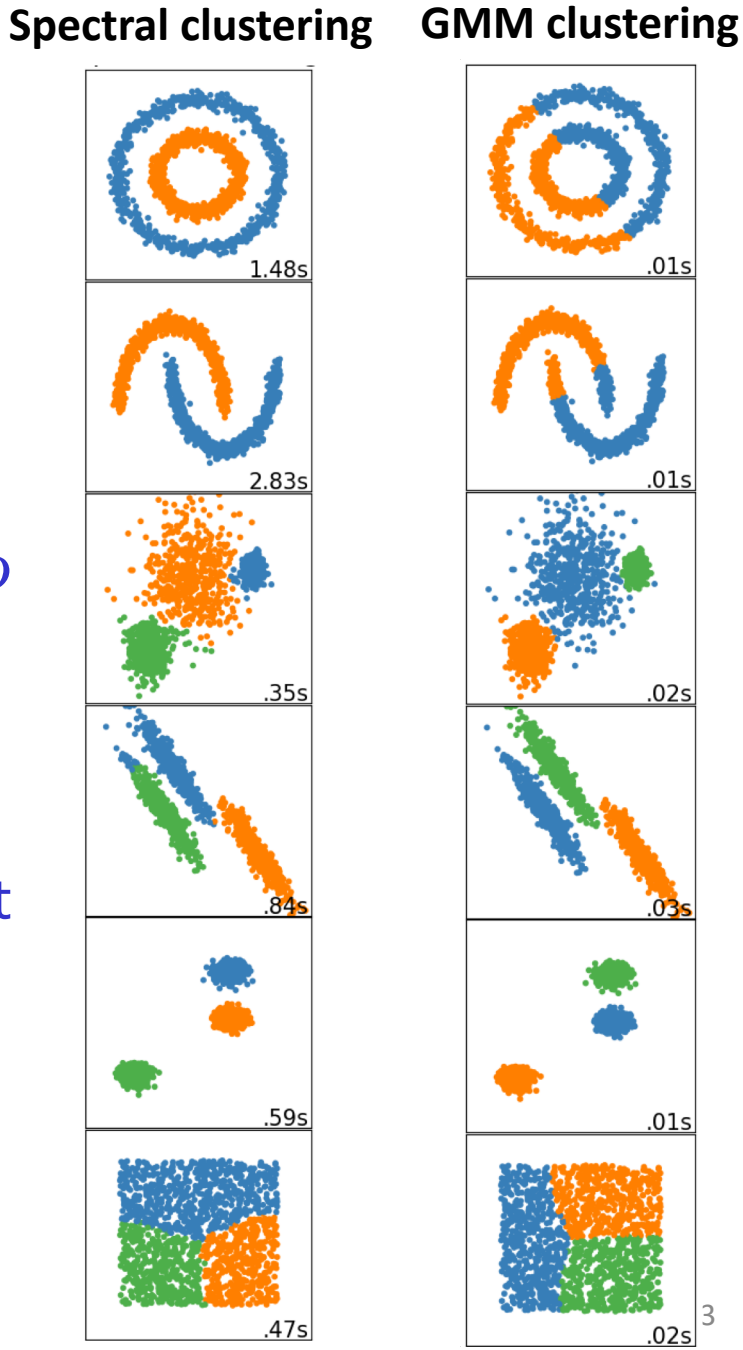
Scaled PCA for spectral clustering

PCA: $W = \sum_k v_k \lambda_k v_k^T$

Scaled PCA: $W = D^{\frac{1}{2}} \tilde{W} D^{\frac{1}{2}} = D \sum_{k=1} q_k \lambda_k q_k^T D$

$\tilde{W} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}, \quad \tilde{w}_{ij} = w_{ij} / (d_i d_j)^{1/2}$

$q_k = D^{-\frac{1}{2}} v_k$ scaled principal component



Outline

- Principal Component Analysis (PCA)
- From matrix factorization perspectives
- Hebbian learning, LMSER and PCA
- Probabilistic PCA, Factor Analysis (FA)

Hebbian learning

- Donald Hebb wrote in 1949:
  - When an axon in cell A is near enough to excite cell B and repeatedly and persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency in firing B is increased.

- The Hebbian synapse

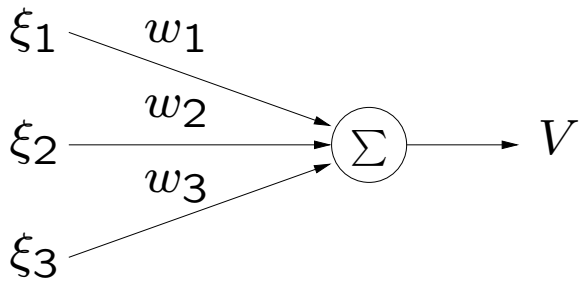
$$\Delta w_1(t) \propto x(t)\, y(t)$$

The Hebbian Neuron

A computational system which implements Hebbian learning.

Let's assume a linear unit; experiment shows this is largely sufficient:

$$V = \sum_j w_j \xi_j = \bar{w}^T \bar{\xi} \tag{2}$$



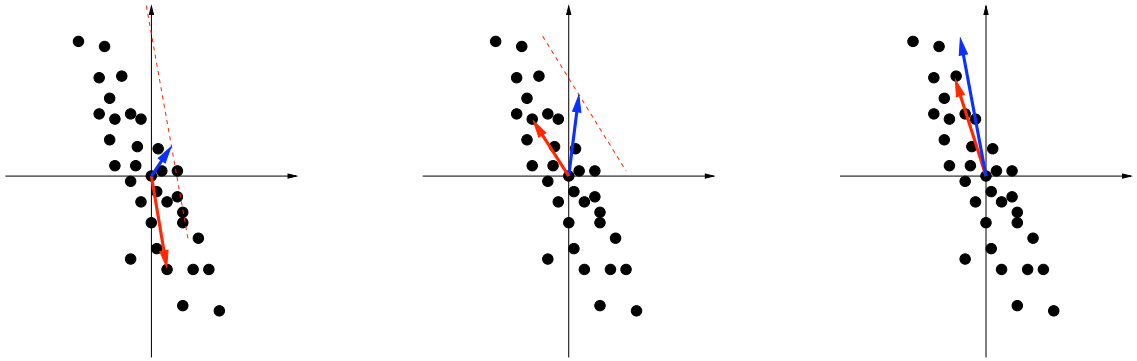
Plain Hebbian learning:

$$\Delta \bar{w} = \eta \, V \, \bar{\xi} \tag{3}$$

A adaptive learning rule

Hebbian learning implements PCA

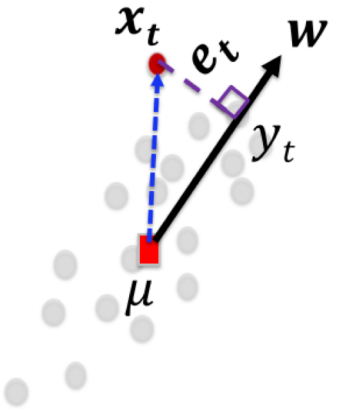
Recall that $\Delta\bar{w} = \eta V \bar{\xi}$ and that $V = \bar{w}^T \bar{\xi}$.



In this case, the weight vector will ultimately align itself with the *direction of greatest variance* in the data.

$$J(w) = \frac{1}{N}\sum_{t=1}^{N} ||\boldsymbol{x_t} - (\boldsymbol{x_t^T w})\boldsymbol{w}||^2$$

$$\boldsymbol{x_t^T x_t} - (\boldsymbol{x_t^T w})\boldsymbol{w^T x_t} - \boldsymbol{x_t^T}(\boldsymbol{x_t^T w})\boldsymbol{w} + (\boldsymbol{x_t^T w})\boldsymbol{w^T}(\boldsymbol{x_t^T w})\boldsymbol{w}$$
$$= \boldsymbol{x_t^T x_t} - \boldsymbol{w^T}(\boldsymbol{x_t x_t^T})\boldsymbol{w}$$



LMSER for PCA

Least Mean Square Error Reconstruction (LMSER)

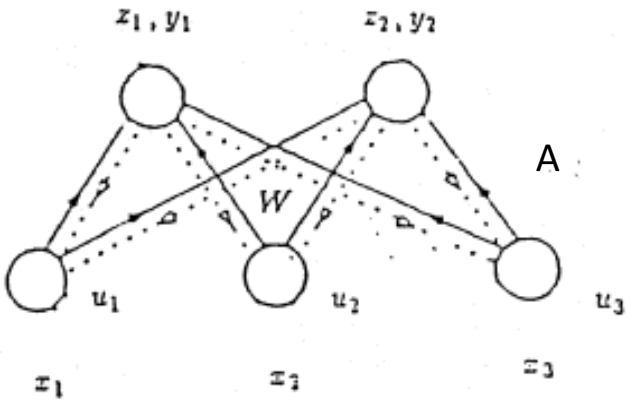
Xu, 1991, 93 *Proc. IJCNN 91,* Singapore, pp. 2362-2373
Neural Networks, vol. 6, pp. 627-648, 1993

$$s(\bar{y}) = [s(\bar{y}^{(1)}), \ldots, s(\bar{y}^{(m)})]^T$$

$$s(r) = \frac{1}{1+e^{-r}}, \quad \bar{y} = W(x-\mu)$$

$$J(W) = \frac{1}{N}\sum_{t=1}^{N} ||x_t - W^T W x_t||^2$$

W = A^T



A

$$\vec{y} = W\vec{x}, \vec{u} = W^t\vec{y}, \vec{y}^r = W\vec{u}$$

$$\bar{z} = S(\bar{y}) = A_m\bar{y}, A_m = diag\left[a_1, \cdots, a_{n_1}\right]$$

$a_1 > \cdots > a_{n_1}$ are all positive.

$$\tau^W \frac{dW}{dt} = \bar{z}\bar{x}^t - \bar{y}\bar{u}^t$$

$$\tau^W \frac{dW}{dt} = \bar{z}\bar{x}^t - \bar{y}\bar{u}^t + \bar{z}\bar{x}^t - \bar{y}^t\bar{x}^t$$

LMSER implements PCA/PSA

THEOREM 3. *Assume* $\lambda_1 \geq \lambda_2 \ldots \lambda_{n_1} > \lambda_{n_1+1} \geq \lambda_{n_0} > 0$ *and* $\Phi = [\vec{\phi}_1, \ldots, \vec{\phi}_{n_0}]$. *Then all the critical points of eqn* (10*b*) *given by Theorem* 2 *are saddle points of the energy landscape of* J, *except those with* $W = R'\Phi''^t$,

$R''^t R' = I$, $\Phi'^t \Phi' = I$, *where* $\Phi' = D\Phi'$, *and* $D = [D_1 | \vec{0}]$ *with* D_1 *being diagonal matrix and its diagonal elements being either* $+1$ *or* -1. *Furthermore, these* $W = R'\Phi''^t$ *let* $J = \frac{1}{2}E(\|\vec{x} - \vec{u}\|^2) = \frac{1}{2}E(\|\vec{x} - \mathbf{W}^t W\vec{x}\|^2)$ *reach its only local (also global) minimum* $J_{min} = \frac{1}{2} \sum_{i=n_1+1}^{n_0} \lambda_i$.

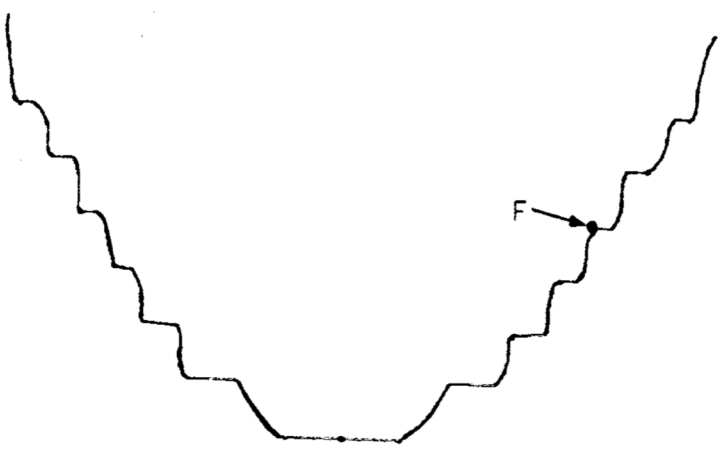


FIGURE 3. The landscape of J. There is only one unique minimum which is the flat bottom, and there are in total $\sum_{l=1}^{n_1} C_{n_0}^l - 1$ plateaux which are the saddle points.

Outline

- Principal Component Analysis (PCA)
- From matrix factorization perspectives
- Hebbian learning, LMSER and PCA
- Probabilistic PCA, Factor Analysis (FA)

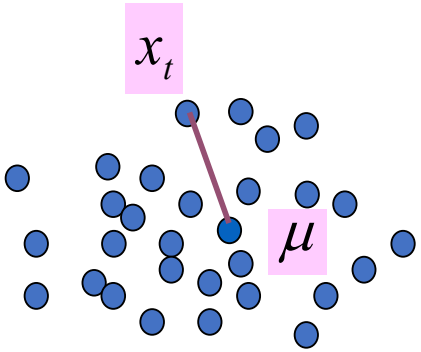
平方误差与高斯分布

$$e_t = x_t - \mu, \quad (x_t - \mu)^2 \Leftrightarrow p(x_t, \theta) \propto e^{-c(x_t-\mu)^2}, c > 0 \text{ Gaussian}$$

$$G(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$$

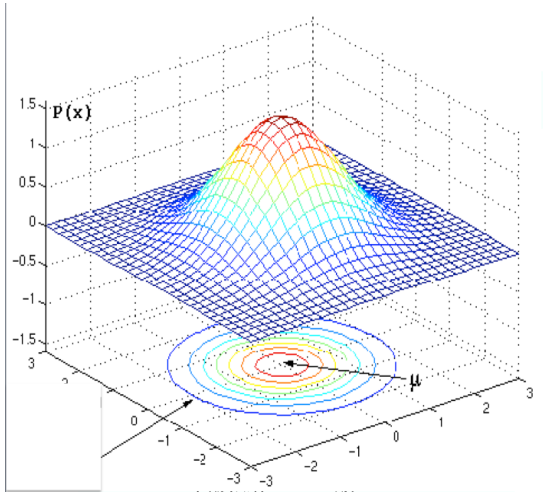
where σ^2 is the variance, and μ is the mean value.

$$e_t = x_t - \mu, \quad \|x_t - \mu\|^2 \Leftrightarrow p(x_t, \theta) \propto e^{-c\|x_t-\mu\|^2}, c > 0 \text{ Gaussian}$$



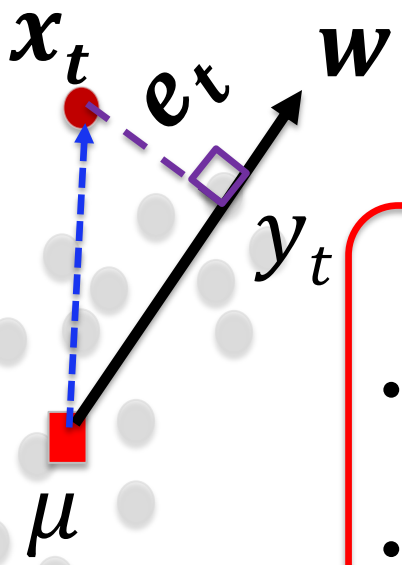
$$G(\mathbf{x} \mid \boldsymbol{\mu}, \sigma^2) = \frac{1}{(2\pi)^{d/2}\sigma^d} \exp\{-\frac{1}{2\sigma^2}(\mathbf{x}-\boldsymbol{\mu})^T(\mathbf{x}-\boldsymbol{\mu})\}$$

where σ^2 is the variance, and $\boldsymbol{\mu}$ is the mean vector.
d is the dimension.



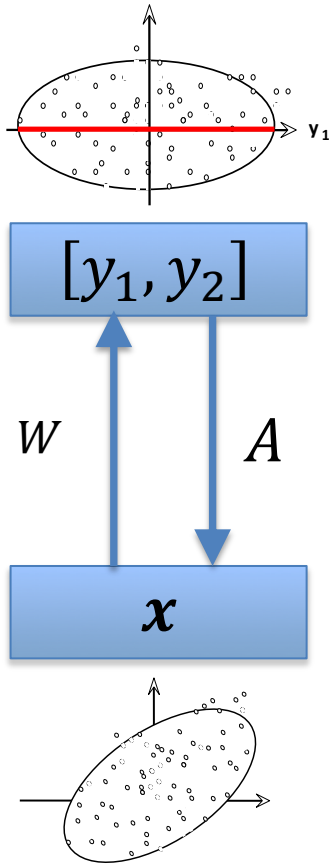
Generative model perspective

Continuous latent variable $\boldsymbol{y}$



For the t-th data point:

- Randomly sample a y_t:
  $\boldsymbol{y}_t \sim G(\boldsymbol{y}|\boldsymbol{0}, \Sigma_y)$;
- Randomly generate a noise e_t
  $\boldsymbol{e}_t \sim G(\boldsymbol{e}|0, \sigma^2 I)$
- Generate $\boldsymbol{x}_t$ by:
  $\boldsymbol{x}_t = A\boldsymbol{y}_t + \boldsymbol{\mu} + \boldsymbol{e}_t$

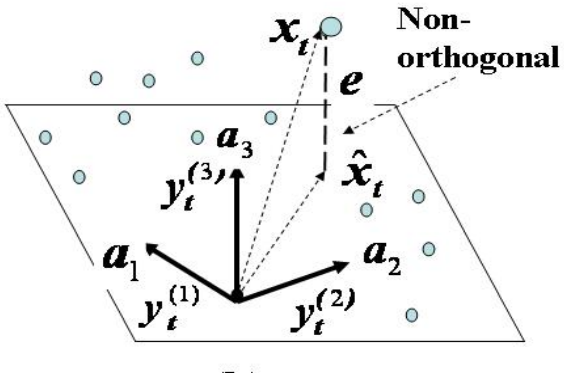


$$||\boldsymbol{w}|| = 1$$

$$y_t = \boldsymbol{x}_t^T \boldsymbol{w}$$

$$\boldsymbol{e}_t = ||\boldsymbol{x}_t - y_t \boldsymbol{w}||^2$$

Factor Analysis (FA) Model



$A^T A = I$ has been removed because it impedes $\sum_t \|e_t\|^2$ to reach its minimum

Indeterminacy

e) a rotation matrix since $A' = \Phi A$ spans the same subspace;

f) a diagonal D with $A' = AD$, $y' = D^{-1} y$.

Assumption

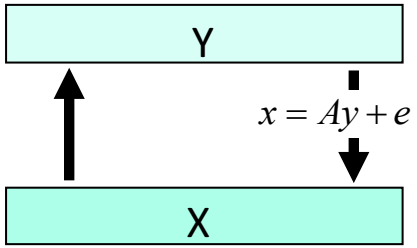
c) $\mathrm{E} e y^T = 0$ (i.e., not correlated) it is not longer a consequenc e;

f) $\mathrm{E} y y^T = \Lambda$ is diagonal !

$$q(y) = \prod_{j=1}^{k} G(y_j | 0,1) = G(y | 0, I)$$

Two Choices

$$q(y) = \begin{cases} G(y|0,I), \text{ choice (a)} \\ G(y|0,\Lambda), \text{ choice (b)} \end{cases}$$



g) a unknown allocation between th e two additive terms $Exx^T = A^T \Lambda A + Eee^T$.

- 样本方差之分割不变性（样本方差在子空间内外可任意分割）
- 超阈维数不变性（高于某维的子空间以零误差描述有限样本集）

EM algorithm for FA

E-Step: $p^{old}(\boldsymbol{y}|\boldsymbol{x}) = \dfrac{G(\boldsymbol{y}|0, I)G(\boldsymbol{x}|A\boldsymbol{y} + \boldsymbol{\mu}, \sigma^2 I)}{G(\boldsymbol{x}|\boldsymbol{\mu}, AA^T + \sigma^2 I)}$

$$E[\boldsymbol{y}|\boldsymbol{x}] = W\boldsymbol{x} \qquad W = A^T(AA^T + \sigma^2 I)^{-1}$$

$$E[\boldsymbol{y}\boldsymbol{y}^T|\boldsymbol{x}] = I - WA + W\boldsymbol{x}\boldsymbol{x}^T W^T$$

M-Step: max $Q\left(p^{old}(\boldsymbol{y}|\boldsymbol{x}), \Theta\right)$

$$Q = \int p^{old}(\boldsymbol{y}|\boldsymbol{x}) \cdot \ln[G(\boldsymbol{y}|0, I)G(\boldsymbol{x}|A\boldsymbol{y} + \boldsymbol{\mu}, \sigma^2 I)]\, d\boldsymbol{y}$$

$$A^{new} = \left(\sum_{t=1}^{N} \boldsymbol{x}_t (E[\boldsymbol{y}|\boldsymbol{x}_t])^T\right)\left(\sum_{t=1}^{N} E[\boldsymbol{y}\boldsymbol{y}^T|\boldsymbol{x}_t]\right)^{-1}$$

$$\sigma^{2^{new}} = \frac{1}{Nd} Tr\left\{\sum_{t=1}^{N} \{\boldsymbol{x}_t \boldsymbol{x}_t^T - A^{new} E[\boldsymbol{y}|\boldsymbol{x}_t]\boldsymbol{x}_t^T\}\right\}$$

Maximum likelihood FA implements PCA

$$p(\boldsymbol{y}) = G(\boldsymbol{y}|\boldsymbol{0}, I), \quad p(\boldsymbol{x}|\boldsymbol{y}) = G(\boldsymbol{x}|\boldsymbol{A}\boldsymbol{y} + \boldsymbol{\mu}, \Sigma_e),$$

$$p(\boldsymbol{x}|\Theta) = \int p(\boldsymbol{y})p(\boldsymbol{x}|\boldsymbol{y})d\boldsymbol{y} = G\left(\boldsymbol{x}|\boldsymbol{\mu}, \mathrm{AA}^{\mathrm{T}} + \Sigma_e\right),$$

$$\max_{\Theta} \log\left\{\prod_{t=1}^{N=1} p(\boldsymbol{x_t}|\Theta)\right\}$$

Maximum Likelihood

$$\boldsymbol{\Sigma}_e = \sigma_e^2 \mathbf{I}_n$$

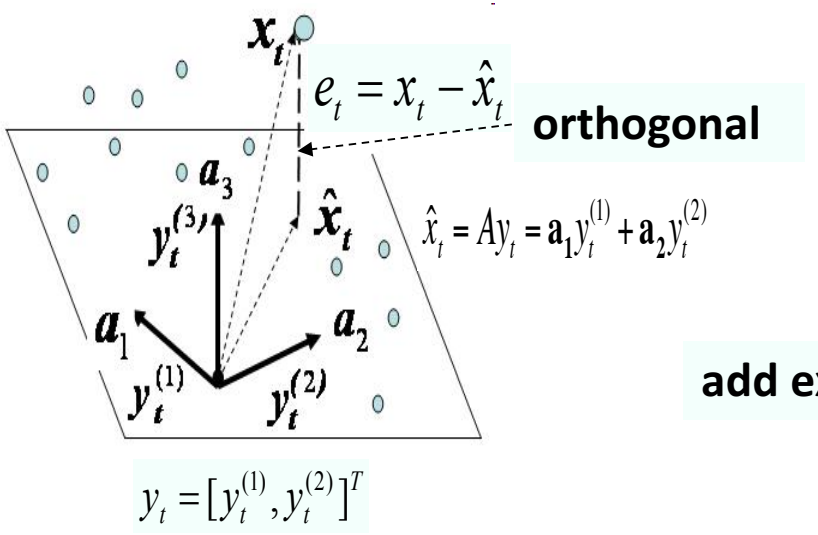
assume $\boldsymbol{\mu} = \mathbf{0}$

PCA

$$\begin{cases} \hat{\mathbf{A}}_{n\times m}^{ML} = \mathbf{U}_{n\times m}(\mathbf{D}_m - \hat{\sigma}_e^2)^{\frac{1}{2}}\mathbf{R}^T, \quad \mathbf{D}_m = diag[s_1, \ldots, s_m] \\ \hat{\sigma}_e^{2,ML} = \frac{1}{n-m}\sum_{i=m+1}^{n} s_i, \end{cases}$$

U is eigenvectors of sample cov.

Reparameterization of FA



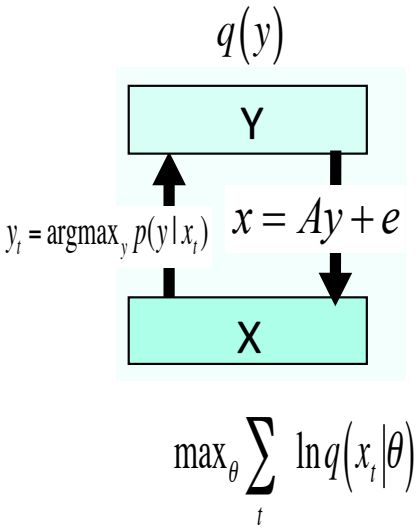
a) *e* and *y* are not correlated

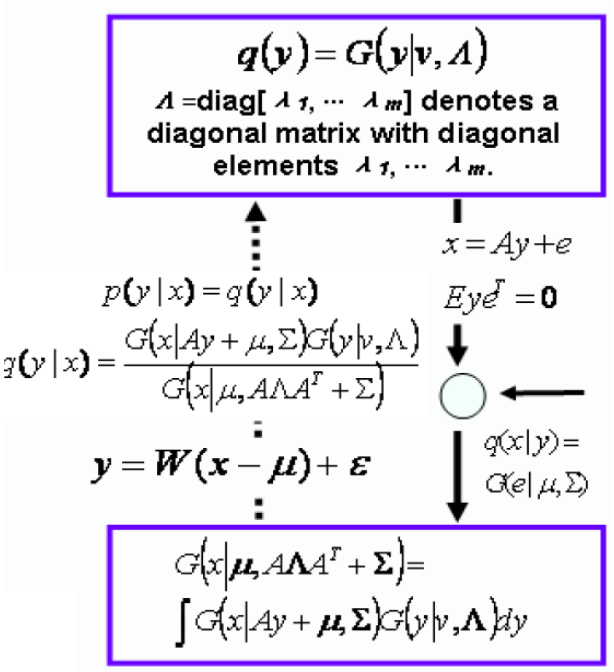
b) $\sum_t \|e_t\|^2$ reaches minimum

c) *e* is a Gaussian noise $G(e|0,\sigma^2 I)$

d) a unique plane but *A* not

add extra constraint I





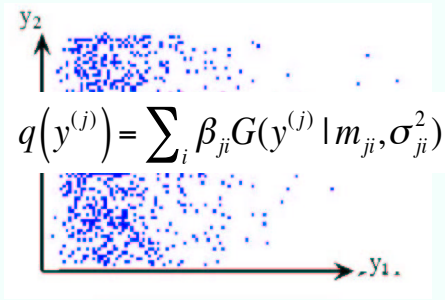
Two parameterizations of FA

	TYPE-A FA-A: $\Theta_m^a = \{\mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\Sigma}_e\}$	TYPE-B FA-B: $\Theta_m^b = \{\mathbf{U}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Sigma}_e\}$
$E[\mathbf{y}\mathbf{e}^T]$	$\mathbf{0}$ ($\mathbf{y}$ and $\mathbf{e}$ uncorrelated)	$\mathbf{0}$ ($\mathbf{y}$ and $\mathbf{e}$ uncorrelated)
$q(\mathbf{y}\|\boldsymbol{\Theta})$	$G(\mathbf{y}\|\mathbf{0}, \mathbf{I}_m)$	$G(\mathbf{y}\|\mathbf{0}, \boldsymbol{\Lambda})$, $\boldsymbol{\Lambda} = diag[\lambda_1, \ldots, \lambda_m]$
$\mathbf{A}$	any full column rank matrix	$\mathbf{A} = \mathbf{U}$, $\mathbf{U}^T\mathbf{U} = \mathbf{I}_m$
$q(\mathbf{x}\|\mathbf{y}, \boldsymbol{\Theta})$	$G(\mathbf{x}\|\mathbf{A}\mathbf{y} + \boldsymbol{\mu}, \boldsymbol{\Sigma}_\mathbf{e})$	$G(\mathbf{x}\|\mathbf{U}\mathbf{y} + \boldsymbol{\mu}, \boldsymbol{\Sigma}_\mathbf{e})$
$q(\mathbf{x}\|\boldsymbol{\Theta})$	$G(\mathbf{x}\|\boldsymbol{\mu}, \mathbf{A}\mathbf{A}^T + \boldsymbol{\Sigma}_e)$	$G(\mathbf{x}\|\boldsymbol{\mu}, \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T + \boldsymbol{\Sigma}_\mathbf{e})$

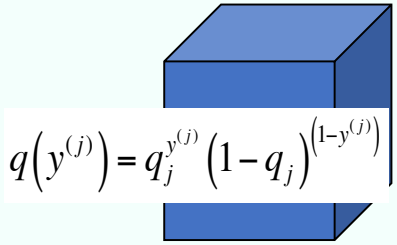
Which one is better?

Linear non-Gaussian

$$q(y) = \prod_{j=1}^{k} q(y^{(j)})$$

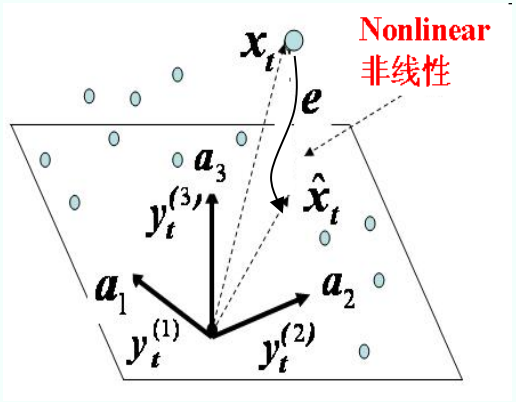


$$q\left(y^{(j)}\right) = \sum_i \beta_{ji} G(y^{(j)} \mid m_{ji}, \sigma_{ji}^2)$$

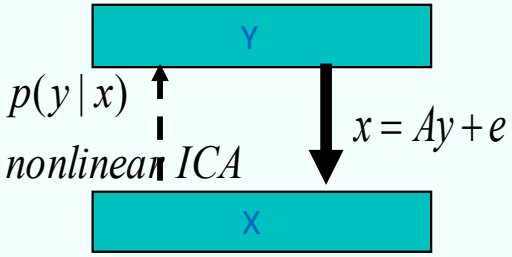


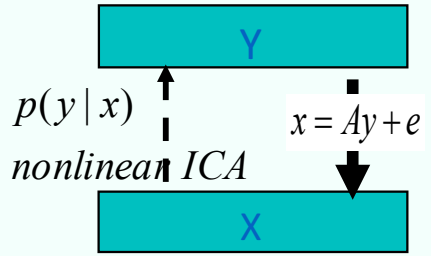
$$q\left(y^{(j)}\right) = q_j^{y^{(j)}} \left(1 - q_j\right)^{\left(1-y^{(j)}\right)}$$

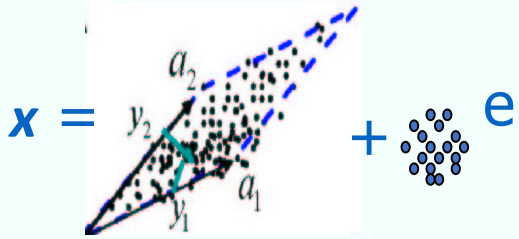
$y_j = 1 \text{ or } 0$



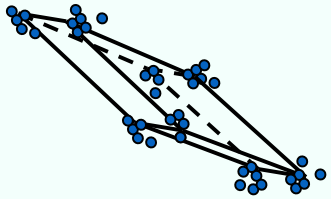
y_1 y_2 y_3 are real







NFA



BFA

Thank you!