

# Ch6: Optimal Feature Generation

❖ In general, feature generation is a problem-dependent task. However, there are a few general directions common in a number of applications. We focus on three such alternatives.

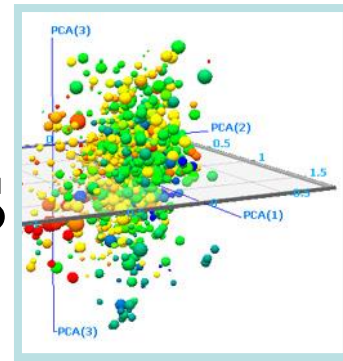
➤ Optimized features based on Scatter matrices (Fisher's linear discrimination).

- The goal: Given an original set of  $m$  measurements  $\underline{x} \in \mathfrak{R}^m$ , compute  $\underline{y} \in \mathfrak{R}^\ell$ , by the linear transformation

$$\underline{y} = A^T \underline{x}$$

so that the  $J_3$  scattering matrix criterion involving  $S_w$ ,  $S_b$  is maximized.  $A^T$  is an  $\ell \times m$  matrix.

# Principal Component Analysis



- Given  $N$   $m$ -dimensional samples  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ . Representing the set by  $\mathbf{x}_0$  (finding a vector  $\mathbf{x}_0$  such that the sum of the squared distances between  $\mathbf{x}_0$  and the various  $\mathbf{x}_k$  is as small as possible. We define the **squared-error criterion** function  $J_0(\mathbf{x}_0)$  by

$$J_0(\mathbf{x}_0) = \sum_{k=1}^N \|\mathbf{x}_0 - \mathbf{x}_k\|^2$$

and seek the value of  $\mathbf{x}_0$  that minimizes  $J_0$ .

- Solution:  $\mathbf{x}_0 = \mathbf{m}$ , where  $\mathbf{m}$  is the sample mean,

$$\mathbf{m} = \bar{\mathbf{x}} = \frac{1}{N} \sum_{k=1}^N \mathbf{x}_k$$

- **Proof:**

$$\begin{aligned}
 J_0(\mathbf{x}_0) &= \sum_{k=1}^N \|\mathbf{x}_0 - \mathbf{x}_k\|^2 = \sum_{k=1}^N \|(\mathbf{x}_0 - \mathbf{m}) - (\mathbf{x}_k - \mathbf{m})\|^2 \\
 &= \sum_{k=1}^N \|(\mathbf{x}_0 - \mathbf{m})\|^2 - 2 \sum_{k=1}^N (\mathbf{x}_0 - \mathbf{m})^T (\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^N \|(\mathbf{x}_k - \mathbf{m})\|^2 \\
 &= \sum_{k=1}^N \|(\mathbf{x}_0 - \mathbf{m})\|^2 - 2(\mathbf{x}_0 - \mathbf{m})^T \sum_{k=1}^N (\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^N \|(\mathbf{x}_k - \mathbf{m})\|^2 \\
 &= \sum_{k=1}^N \|(\mathbf{x}_0 - \mathbf{m})\|^2 + \underbrace{\sum_{k=1}^N \|(\mathbf{x}_k - \mathbf{m})\|^2}_{\text{independent of } \mathbf{x}_0} \rightarrow \text{Minimized by } \mathbf{x}_0 = \mathbf{m}
 \end{aligned}$$

- The sample mean is a zero-dimensional representation of data set. It does not reveal any of the variability in the data.
- We can obtain a more interesting, one-dimensional representation by projecting the data onto a line running through the sample mean,  $\mathbf{x}=\mathbf{m}+a\mathbf{e}$ , where  $\mathbf{e}$  is a unit vector in the direction of the line.
- If we represent  $\mathbf{x}_k$  by  $\mathbf{m}+a_k\mathbf{e}$ , we can find an “optimal” set of coefficient  $a_k$  by minimizing the squared-error criterion function

$$\begin{aligned}
J_1(a_1, \dots, a_n, \mathbf{e}) &= \sum_{k=1}^N \|(\mathbf{m} + a_k \mathbf{e}) - \mathbf{x}_k\|^2 = \sum_{k=1}^N \|a_k \mathbf{e} - (\mathbf{x}_k - \mathbf{m})\|^2 \\
&= \sum_{k=1}^N a_k^2 \|\mathbf{e}\|^2 - 2 \sum_{k=1}^N a_k \mathbf{e}^T (\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^N \|(\mathbf{x}_k - \mathbf{m})\|^2 \quad (82)
\end{aligned}$$

Recognizing that  $\|\mathbf{e}\|=1$ , partially differentiating with respect to  $a_k$ , and setting the derivative to zero, we obtain

$$a_k = \mathbf{e}^T (\mathbf{x}_k - \mathbf{m}) \quad (83)$$

Geometrically, this result says that we obtain a least-squares solution by projecting the vector  $\mathbf{x}_k$  onto the line in the direction of  $\mathbf{e}$  that passes through the sample mean.

Finding the best direction  $\mathbf{e}$  for the line.  $\rightarrow$  Scatter Matrix  $\mathbf{S}$ .

$$\mathbf{S} = \sum_{k=1}^N (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T$$

- Scatter matrix  $\mathbf{S}$  is  $N-1$  times the sample covariance matrix. Using Eqs 82 , 83  $\rightarrow$

$$\begin{aligned}
 J_1(\mathbf{e}) &= \sum_{k=1}^N a_k^2 - 2 \sum_{k=1}^N a_k^2 + \sum_{k=1}^N \|(\mathbf{x}_k - \mathbf{m})\|^2 \\
 &= - \sum_{k=1}^N [\mathbf{e}^T (\mathbf{x}_k - \mathbf{m})]^2 + \sum_{k=1}^N \|(\mathbf{x}_k - \mathbf{m})\|^2 \\
 &= - \sum_{k=1}^N \mathbf{e}^T (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T \mathbf{e} + \sum_{k=1}^N \|(\mathbf{x}_k - \mathbf{m})\|^2 \\
 &= -\mathbf{e}^T \mathbf{S} \mathbf{e} + \sum_{k=1}^N \|(\mathbf{x}_k - \mathbf{m})\|^2
 \end{aligned}$$

- The vector  $\mathbf{e}$  that minimizes  $J_1$  also maximizes  $\mathbf{e}^T \mathbf{S} \mathbf{e}$ . We use the method of Lagrange multiplier to maximize  $\mathbf{e}^T \mathbf{S} \mathbf{e}$  subject to the constraint that  $\|\mathbf{e}\|=1$ .


Letting  $\lambda$  be the undetermined multiplier we differentiate

$$u = \mathbf{e}^T \mathbf{S} \mathbf{e} - \lambda(\mathbf{e}^T \mathbf{e} - 1)$$

with respect to  $\mathbf{e}$  and equating to zero to obtain

$$\frac{\partial u}{\partial \mathbf{e}} = 2\mathbf{S}\mathbf{e} - 2\lambda\mathbf{e} = 0 \quad \Rightarrow \quad \mathbf{S}\mathbf{e} = \lambda\mathbf{e} \quad \Rightarrow \quad \mathbf{e}^T \mathbf{S} \mathbf{e} = \lambda \mathbf{e}^T \mathbf{e} = \lambda$$

To maximize  $\mathbf{e}^T \mathbf{S} \mathbf{e}$  we want to select the **eigenvector** corresponding to the **largest eigenvalue** of the scatter matrix.



This interesting result can be extended from one-dimensional projection to a  $l$ -dimensional projection.

$$\mathbf{x} = \mathbf{m} + \sum_{i=1}^l a_i \mathbf{e}_i$$

Where  $l < m$ . It can be shown that

$$J_l = \sum_{k=1}^N \left\| \left( \mathbf{m} + \sum_{i=1}^l a_{ki} \mathbf{e}_i \right) - \mathbf{x}_k \right\|^2$$

is minimized when the vectors  $\mathbf{e}_1, \dots, \mathbf{e}_l$  are the  $l$  eigenvectors of the scatter matrix having the largest eigenvalues. Because the scatter matrix is real and symmetric, these eigenvectors are orthogonal.

They form a natural set of basis vectors for representing any feature vector  $\mathbf{x}$ . The *coefficients*  $a_i$  are the components of  $\mathbf{x}$  in that basis and are called *principal components*.



## ❖ Principal Components Analysis

### ❖ (The Karhunen – Loève transform):

- The goal: Given an original set of  $m$  measurements  $\underline{x} \in \mathfrak{R}^m$  compute  $\underline{y} \in \mathfrak{R}^\ell$  (the data samples have zero mean).

$$\underline{y} = A^T \underline{x}$$

for an **orthogonal**  $A$ , so that the elements of  $\underline{y}$  are **optimally mutually uncorrelated**.

That is

$$E[y(i)y(j)] = 0, i \neq j$$

- Sketch of the proof:

$$R_y = E[\underline{y}\underline{y}^T] = E[A^T \underline{x}\underline{x}^T A] = A^T R_x A$$

- If  $A$  is chosen so that its columns  $\underline{a}_i$  are the **orthogonal eigenvectors** of  $R_x$ , then

$$R_y = A^T R_x A = \Lambda$$

where  $\Lambda$  is **diagonal** with elements the respective **eigenvalues**  $\lambda_i$ .

- Observe that this is a **sufficient** condition but not **necessary**. It **imposes** a **specific orthogonal** structure on  $A$ .
- **Properties of the solution**
  - **Mean Square Error approximation.** (Fit the model with minimal reconstruction error)
  - Due to the orthogonality of  $A$ :

$$\underline{x} = \sum_{i=0}^{N-1} y(i) \underline{a}_i, \quad y(i) = \underline{a}_i^T \underline{x}$$

- Define a new vector in the  $m$ -dimensional subspace

$$\underline{\hat{x}} = \sum_{i=0}^{m-1} y(i) \underline{a}_i$$

- The Karhunen – Loève transform minimizes the square error:

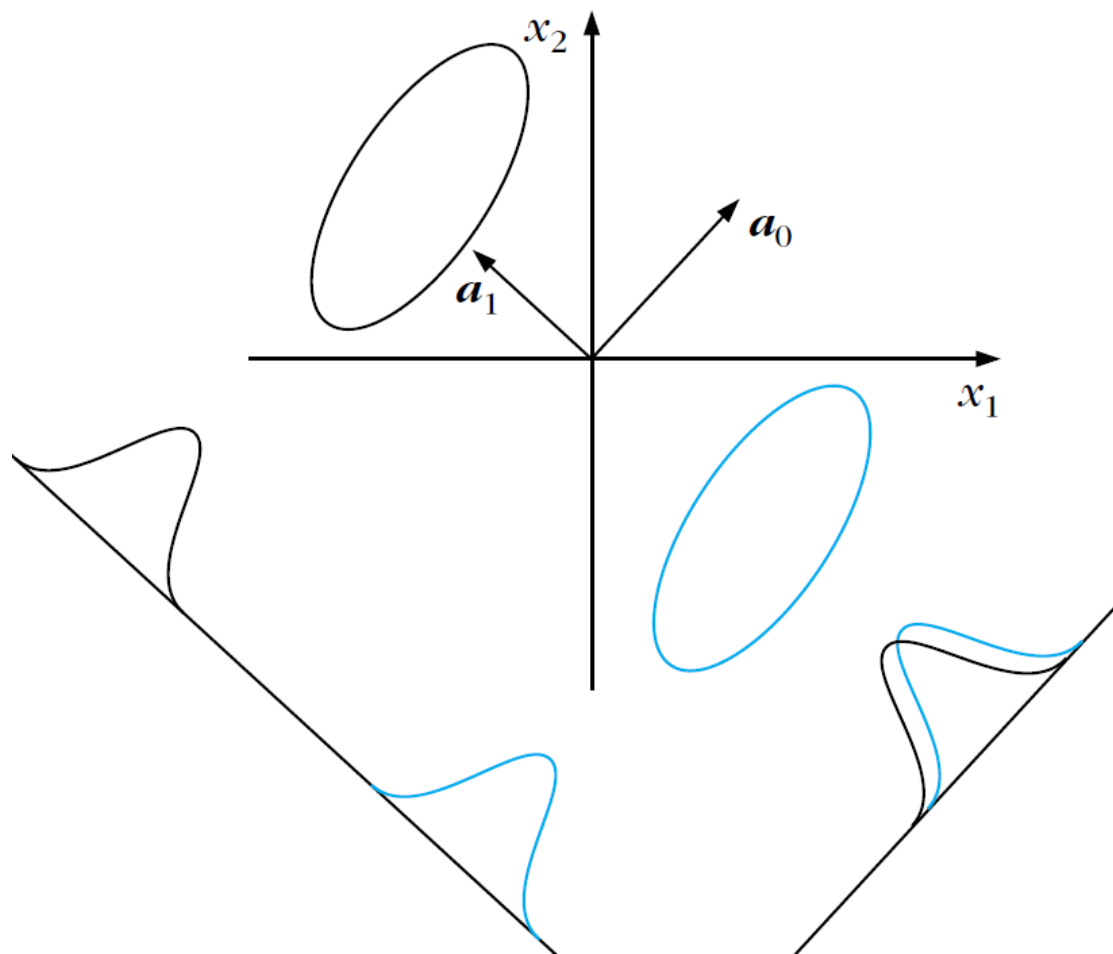
$$\begin{aligned} E \left[ \left\| \underline{x} - \underline{\hat{x}} \right\|^2 \right] &= E \left[ \left\| \sum_{i=m}^{N-1} y(i) \underline{a}_i \right\|^2 \right] = E \left[ \sum_i \sum_j (y(i) \underline{a}_i^T) (y(j) \underline{a}_j) \right] \\ &= \sum_{i=m}^{N-1} E \left[ y^2(i) \right] = \sum_{i=m}^{N-1} \underline{a}_i^T E \left[ \underline{x} \underline{x}^T \right] \underline{a}_i \end{aligned}$$

- The error is:

$$E \left[ \left\| \underline{x} - \underline{\hat{x}} \right\|^2 \right] = \sum_{i=m}^{N-1} \underline{a}_i^T \lambda_i \underline{a}_i = \sum_{i=m}^{N-1} \lambda_i$$

It can be also shown that this is **the minimum mean square error compared to **any** other representation of  $\underline{x}$  by an  $m$ -dimensional vector.**

- In other words,  $\hat{\underline{x}}$  is the **projection** of  $\underline{x}$  into the subspace spanned by the principal  $m$  eigenvectors. However, for Pattern Recognition this is not always the best solution.



*The KL transform is not always best for pattern recognition. In this example, projection on the eigenvector with the larger eigenvalue makes the two classes coincide. On the other hand, projection on the other eigenvector keeps the classes separated.* 12

- Total variance: It is easily seen that

$$\sigma_{y(i)}^2 = E \left[ y^2(i) \right] = \lambda_i$$

That is, the eigenvalues of the input correlation matrix are equal to the variances of the transformed features.

- Thus Karhunen – Loève transform makes the total **variance maximum**.

## Entropy

- The entropy of a process is defined as

$$H_y = -E \left[ \ln P_{\underline{y}}(\underline{y}) \right]$$

and it is a measure of the randomness of the process.

- Assuming  $\underline{y}$  to be a zero mean multivariate Gaussian, then the K-L transform **maximizes the entropy**:

$$H_y = -E \left[ \ln P_y(\underline{y}) \right] \quad \text{of the resulting } \underline{y} \text{ process.}$$

**Note:**

For a zero mean Gaussian multivariable  $m$ -dimensional process, the entropy becomes

$$H_y = \frac{1}{2} E[\mathbf{y}^T R_y^{-1} \mathbf{y}] + \frac{1}{2} \ln |R_y| + \frac{m}{2} \ln(2\pi)$$

$$E[\mathbf{y}^T R_y^{-1} \mathbf{y}] = E[\text{trace}\{\mathbf{y}^T R_y^{-1} \mathbf{y}\}] = E[\text{trace}\{R_y^{-1} \mathbf{y} \mathbf{y}^T\}] = \text{trace}(I) = m$$

$$\ln |R_y| = \ln(\lambda_0 \lambda_1 \dots \lambda_{m-1})$$

In words, selection of the  $m$  features that correspond to the  $m$  largest eigenvalues maximizes the entropy of the process. This is expected because variance and randomness are directly related.

❖ **Subspace Classification.** Following the idea of projecting in a subspace, the subspace classification **classifies** an unknown  $\underline{x}$  to the class whose **subspace is closer to  $\underline{x}$** . The following steps are in order:

- For **each class**, estimate the autocorrelation matrix  $R_i$ , and compute the  $m$  **largest eigenvalues**. Form  $A_i$ , by using respective eigenvectors as columns.
- Classify  $\underline{x}$  to the class  $\omega_i$ , for which the norm of the **subspace projection is maximum**

$$\|A_i^T \underline{x}\| > \|A_j^T \underline{x}\| \quad \forall i \neq j$$

According to Pythagoras theorem, this corresponds to **the subspace to which  $\underline{x}$  is closer.**

## Example 6.2

The correlation matrix of a vector  $\mathbf{x}$  is given by  $\mathbf{R}_x$ , Compute the KL transform of the input vector.

$$\mathbf{R}_x = \begin{bmatrix} 0.3 & 0.1 & 0.1 \\ 0.1 & 0.3 & -0.1 \\ 0.1 & -0.1 & 0.3 \end{bmatrix}$$

The eigenvalues of  $\mathbf{R}_x$  are  $\lambda_0 = \lambda_1 = 0.4$ ,  $\lambda_2 = 0.1$ . Since the matrix  $\mathbf{R}_x$  is symmetric, we can always construct orthonormal eigenvectors. For this case we have

$$\mathbf{a}_0 = \frac{1}{\sqrt{6}} \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{a}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}, \quad \mathbf{a}_2 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

The KL transform is then given by

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \end{bmatrix} = \begin{bmatrix} 2/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{3} & -1/\sqrt{3} & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix}$$

where  $y(0)$ ,  $y(1)$  correspond to the two largest eigenvalues.



### Example 6.3

Figure 6.2 shows 100 points in the two-dimensional space. The points spread around the  $x_2 = x_1$  line, and they have been generated by the model  $x_2 = x_1 + \epsilon$ , where  $\epsilon$  is a noise source following the uniform distribution in  $[-0.5, 0.5]$ .

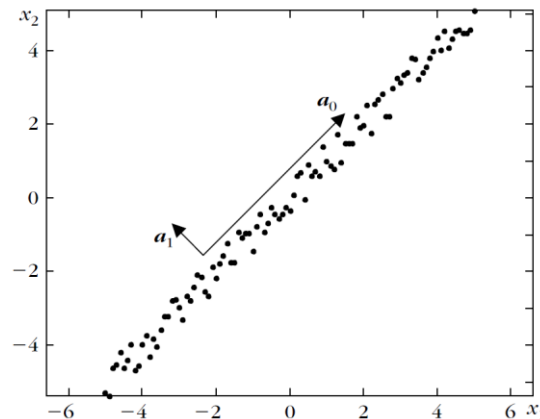
We first compute the covariance matrix and perform an eigendecomposition. The resulting eigenvectors are

$$\mathbf{a}_0 = [0.7045, 0.7097]^T, \quad \mathbf{a}_1 = [-0.7097, 0.7045]^T$$

corresponding to the eigenvalues

$$\lambda_0 = 17.26, \quad \lambda_1 = 0.04$$

respectively. Observe that  $\lambda_0 \gg \lambda_1$ . Figure 6.2 shows the two eigenvectors.  $\mathbf{a}_0$ , which correspond to the largest eigenvalue, points in the direction where data show maximum variability. Projecting along this direction retains most of the variance. Moreover, according to PCA, the dimensionality of the set is approximately one, due to the large gap between  $\lambda_0$  and  $\lambda_1$ , which is the correct answer. Also, note, that  $\mathbf{a}_0$ , is (approximately) parallel to the line  $x_2 = x_1$ .



# PCA Algorithm

## ❖ PCA algorithm:

- 1.  $\mathbf{X} \leftarrow$  Create  $N \times m$  data matrix, with one row vector  $x_n$  per data point
- 2.  $\mathbf{X} \leftarrow$  subtract mean  $\mathbf{x}$  from each row vector  $x_n$  in  $\mathbf{X}$
- 3.  $\Sigma \leftarrow$  covariance matrix of  $\mathbf{X}$
- 4. Find eigenvectors and eigenvalues of  $\Sigma$
- 5. PC's  $\leftarrow$  the  $l$  eigenvectors with largest eigenvalues

# PCA Algorithm in Matlab

```
% generate data
```

```
Data = mvnrnd([5, 5], [1 1.5; 1.5 3], 100);  
figure(1); plot(Data(:,1), Data(:,2), '+');
```

```
%center the data
```

```
for i = 1:size(Data,1)  
    Data(i, :) = Data(i, :) - mean(Data);  
end
```

```
DataCov = cov(Data); %covariance matrix
```

```
[PC, variances, explained] = pcacov(DataCov); %eigen
```

```
% plot principal components
```

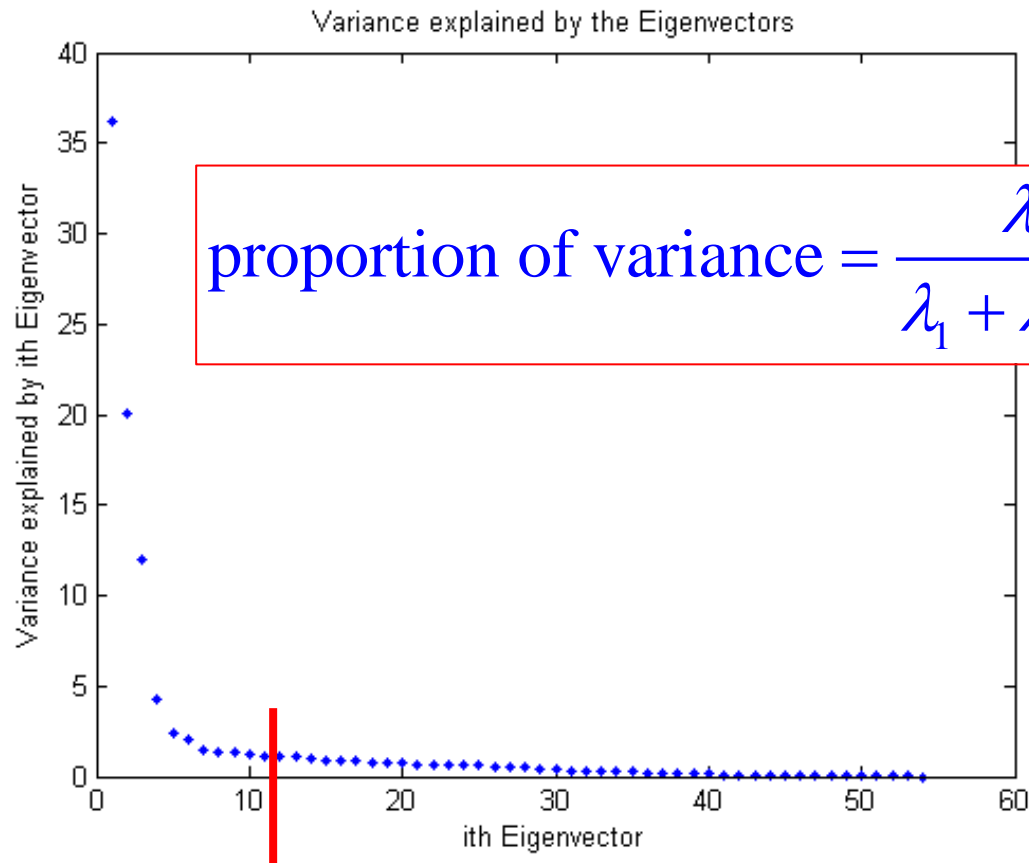
```
figure(2); clf; hold on;  
plot(Data(:,1), Data(:,2), '+b');  
plot(PC(1,1)*[-5 5], PC(2,1)*[-5 5], '-r')  
plot(PC(1,2)*[-5 5], PC(2,2)*[-5 5], '-b'); hold off
```

```
% project down to 1 dimension
```

```
PcaPos = Data * PC(:, 1);
```

# How many components?

- ❖ Check the distribution of eigen-values
- ❖ Take enough many eigen-vectors to cover 80-90% of the variance



## ❖ \*Independent Component Analysis (ICA) PR-Ch3-part4

In contrast to PCA, where the goal was to produce uncorrelated features, the goal in ICA is to produce statistically independent features. This is a much stronger requirement, involving higher to second order statistics. In this way, one may overcome the problems of PCA, as exposed before.

➤ The goal: Given  $\underline{x}$ , compute  $\underline{y} \in \mathbb{R}^{\ell}$

$$\underline{y} = W \underline{x}$$

so that the components of  $\underline{y}$  are statistically independent. In order the problem to have a solution, the following assumptions must be valid:

- Assume that  $\underline{x}$  is indeed generated by a linear combination of independent components

$$\underline{x} = \Phi \underline{y}$$

- $\Phi$  is known as the **mixing** matrix and  $W$  as the **demixing** matrix.
- $\Phi$  must be invertible or of full column rank.
- **Identifiability condition:** All independent components,  $y(i)$ , must be **non-Gaussian**. Thus, in contrast to PCA that can always be performed, ICA is meaningful for non-Gaussian variables.
- Under the above assumptions,  $y(i)$ 's can be uniquely estimated, within a scalar factor.

➤ **Common's method**: Given  $\underline{x}$ , and under the previously stated assumptions, the following steps are adopted:

- **Step 1**: Perform PCA on  $\underline{x}$  :

$$\underline{\hat{y}} = A^T \underline{x}$$

- **Step 2**: Compute a **unitary** matrix,  $\hat{A}$ , so that the **fourth order cross-cummulants** of the transform vector

$$\underline{y} = \hat{A}^T \underline{\hat{y}}$$

**are zero**. This is equivalent to searching for an  $\hat{A}$  that makes the squares of the **auto-cummulants** maximum,

$$\max_{\hat{A}\hat{A}^T=I} \Psi(\hat{A}) = \sum_{i=0}^{N-1} k_4(y(i))^2$$

where,  $k_4(\cdot)$  is the 4<sup>th</sup> order auto-cumulant.

- Step 3:  $W = (A\hat{A})^T$

➤ A hierarchy of components: which  $\ell$  to use? In PCA one chooses the principal ones. In ICA one can choose the ones with the least resemblance to the Gaussian pdf.

characteristic fun. of  $p(\mathbf{x})$ :  $\Phi(\Omega) = \int_{-\infty}^{+\infty} p(\mathbf{x}) \exp(j\Omega\mathbf{x}) d\mathbf{x} \equiv E[\exp(j\Omega\mathbf{x})]$

the moment generating function:

If  $j\Omega$  is changed into  $s$

$$\Phi(s) = \int_{-\infty}^{+\infty} p(\mathbf{x}) \exp(s\mathbf{x}) d\mathbf{x} \equiv E[\exp(s\mathbf{x})]$$

the 2<sup>nd</sup> characteristic function of  $\mathbf{x}$ :

$$\Psi(\Omega) = \ln \Phi(\Omega)$$

$$\frac{d^n \Phi(s)}{ds^n} \equiv \Phi^{(n)}(s) = E[x^n \exp(s\mathbf{x})]$$



the  $n$ th-order moment of  $\mathbf{x}$ :

$$\Phi^{(n)}(0) = E[x^n] \equiv m_n$$

the Taylor expansion of the second generating function results in:

$$\Phi(s) = \sum_{n=0}^{+\infty} \frac{m_n}{n!} s^n$$

$$\Psi(s) = \sum_{n=1}^{+\infty} \frac{\kappa_n}{n!} s^n$$

where  $\kappa_n \equiv \frac{d^n \Psi(0)}{ds^n}$

are known as the **cumulants** of the random variable  $\mathbf{x}$ .



# FastICA

1. Centering  $\mathbf{x} = \tilde{\mathbf{x}} - \mathbf{m}_{\tilde{\mathbf{x}}}$
2. Whitening  $\mathbf{z} = \mathbf{V}\mathbf{x}, E\{\mathbf{z}\mathbf{z}^T\} = \mathbf{I}$
3. Choose  $m$ , No. of ICs to estimate. Set counter  $p \leftarrow 1$
4. Choose an initial guess of unit norm for  $\mathbf{w}_p$ , eg. randomly.
5. Let  $\mathbf{w}_p \leftarrow E\{\mathbf{z}[\mathbf{w}_p^T \mathbf{z}]^3\} - 3\mathbf{w}_p \|\mathbf{w}_p\|^2$
6. Do deflation decorrelation
$$\mathbf{w}_p \leftarrow \mathbf{w}_p - \sum_{j=1}^{p-1} (\mathbf{w}_p^T \mathbf{w}_j) \mathbf{w}_j$$
7. Let  $\mathbf{w}_p \leftarrow \mathbf{w}_p / \|\mathbf{w}_p\|$
8. If  $\mathbf{w}_p$  has not converged ( $|\langle \mathbf{w}_p^{k+1}, \mathbf{w}_p^k \rangle| \neq 1$ ), go to step 5.
9. Set  $p \leftarrow p+1$ . If  $p \leq m$ , go back to step 4.

For a zero mean random variable

$$\kappa_1(x) = E[x] = 0$$

$$\kappa_2(x) = E[x^2] = \sigma^2$$

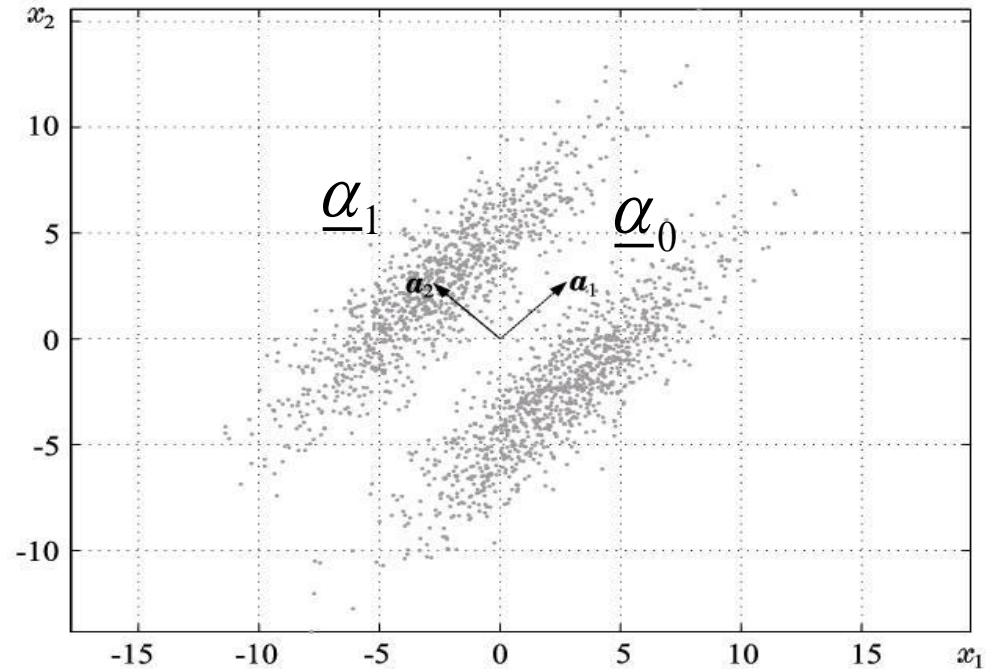
$$\kappa_3(x) = E[x^3]$$

$$\kappa_4(x) = E[x^4] - 3\sigma^4$$

*(kurtosis)*

➤ **Example:**

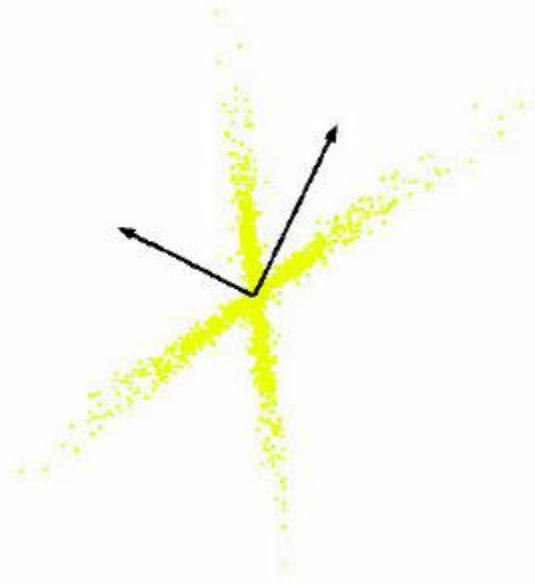
$$\begin{aligned} \boldsymbol{\mu}_1 &= [-2.6042, \quad 2.5]^T \\ \boldsymbol{\mu}_2 &= -\boldsymbol{\mu}_1 \end{aligned} \quad \Sigma = \begin{bmatrix} 10.5246 & 9.6313 \\ 9.6313 & 11.3203 \end{bmatrix} \rightarrow W = \begin{bmatrix} -0.7088 & 0.7054 \\ 0.7054 & 0.7088 \end{bmatrix} \equiv \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_0^T \end{bmatrix}$$



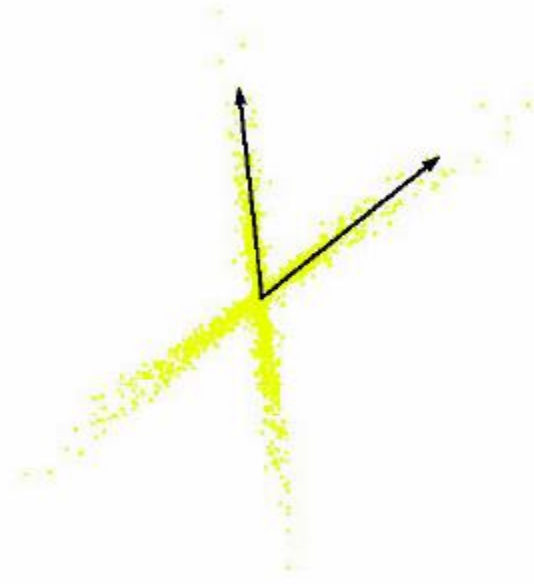
The principal component is  $\underline{\alpha}_0$ , thus according to PCA one chooses as  $y$  the projection of  $\underline{x}$  into  $\underline{\alpha}_0$ . According to ICA, one chooses as  $y$  the projection on  $\underline{\alpha}_1$ . This is the least Gaussian. Indeed:  $K_4(y_1) = -1.7$ ,  $K_4(y_2) = 0.1$

Observe that across  $\underline{\alpha}_1$ , the statistics is **bimodal**. That is, no resemblance to Gaussian.

# PCA vs ICA



PCA  
(orthogonal coordinate)



ICA  
(non-orthogonal coordinate)

## Difference with PCA

- ❖ It is not a dimensionality reduction technique
- ❖ There is no single (exact) solution for components; uses different algorithms (in R: FastICA, PearsonICA, MLICA)
- ❖ ICs are of course uncorrelated but also as independent as possible
- ❖ Uninteresting for Normally distributed variables

# Application domains of ICA

- ❖ Blind source separation (Bell&Sejnowski, Te won Lee, Girolami, Hyvarinen, etc.)
- ❖ Image denoising (Hyvarinen)
- ❖ Medical signal processing – fMRI, ECG, EEG (Mackeig)
- ❖ Modelling of the hippocampus and visual cortex (Lorincz, Hyvarinen)
- ❖ Feature extraction, face recognition (Marni Bartlett)
- ❖ Compression, redundancy reduction
- ❖ Watermarking (D Lowe)
- ❖ Clustering (Girolami, Kolenda)
- ❖ Time series analysis (Back, Valpola)
- ❖ Topic extraction (Kolenda, Bingham, Kaban)
- ❖ Scientific Data Mining (Kaban, etc)

# Image denoising

Original  
image



Noisy  
image



Wiener  
filtering



ICA  
filtering



## Other Feature Generation Methods

- ❖ The Singular Value Decomposition (SVD)
- ❖ The Discrete Fourier Transform (DFT)
- ❖ The Discrete Cosine And Sine Transforms
- ❖ The Hadamard Transform
- ❖ The Haar Transform
- ❖ Discrete Time Wavelet Transform (DTWT)
- ❖ The Multiresolution Interpretation
- ❖ Wavelet Packets
- ❖ Regional Features
- ❖ Features For Shape And Size Characterization
- ❖ Typical Features For Speech And Audio Classification
- ❖ ...