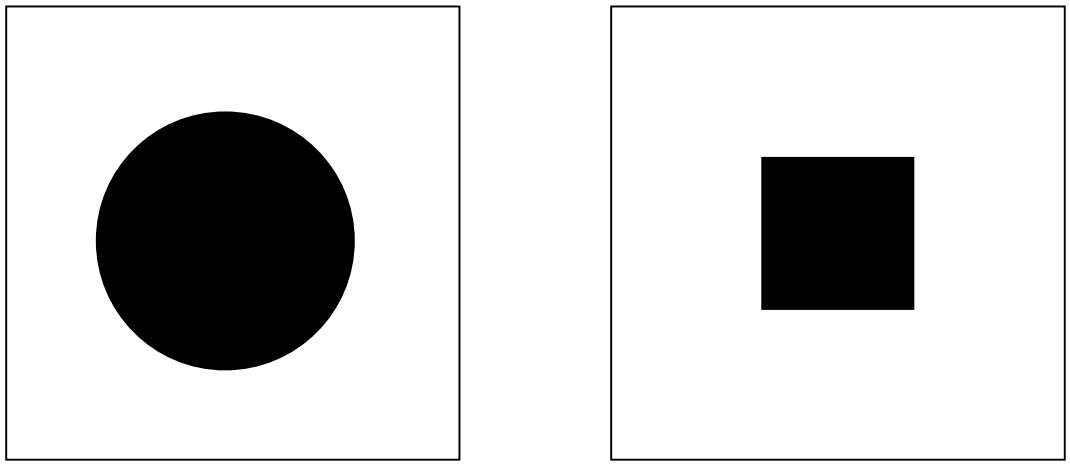
Chapter 6

Optimal feature selection

Simple recognition example with two objects (circle and square)



Assuming that the images are scanned with 512×512 pixels.

No one would use 512×512 pixels as features of the objects !!
(1 feature is sufficient: area)

Feature selection with linear transformations

The complexity of designing a classifier increases with dimension N of the feature space. The intention of the feature selection is to choose an appropiate subspace. The selected features must have high relevance for characterization of the classes, and at the same time guarantee a high capability to discriminate between classes. Thus they must vary little within a class (intra class distance), and guarantee great distances between the classes (inter class distance).

In general it does not make great sense to use the pixels of an image directly as a feature ($N=512^2=2^{18}=0{,}25$ mio. pixel). Generally there is high redundancy in the images since the pixels highly correlate.

It is also not very useful to enlarge a feature space by adding new features, if the new features highly correlate to the existing.

Idea: Transforming the original images into a new feature space (shifting and rotation of the coordinate system (unitary transformation)). And thus reduction to few features and - at the same time - information compression/condensation.

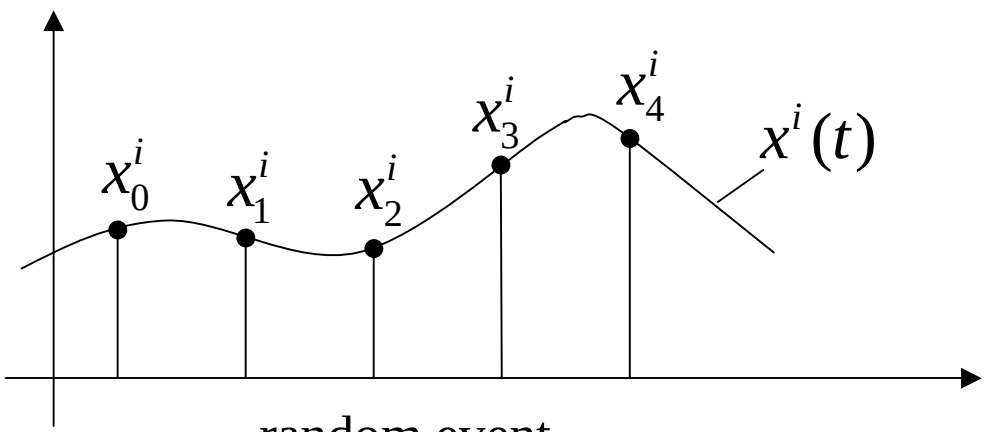
Characterizing random events in vector spaces

A *random event* $\mathbf{x}^i$ is an element of the vector space $\mathbb{X}$. For discrete spaces the elementary event consists of an ordered set of numerical values

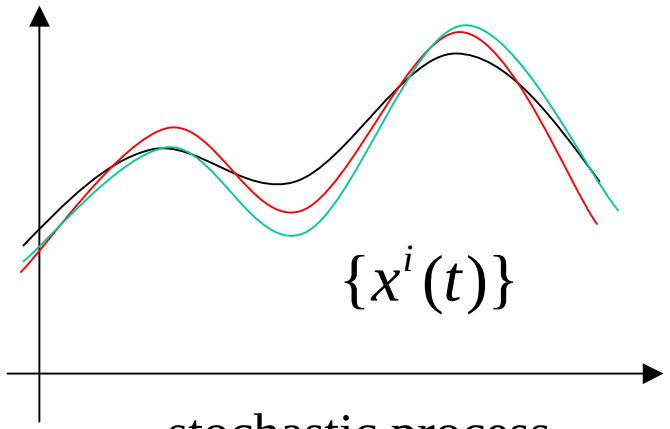
$\mathbf{x}^i := \{x_0^i, x_1^i, x_2^i, ..., x_{N-1}^i\}$

or for continuous spaces consists of a time- or position functions $x^i(t)$.

A *stochastic process* $\mathbf{x}$ consists of a set of events $\mathbf{x} := \{\mathbf{x}^j\}$.



random event



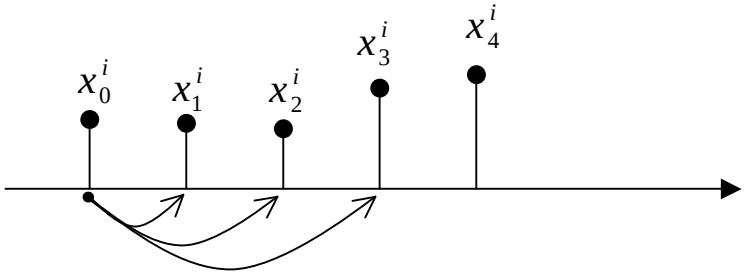
stochastic process

Statistical characteristics of a process

expected value: $\boldsymbol{\mu}_{\mathbf{x}} = \overline{\mathbf{x}} = E\{\mathbf{x}\} \approx \frac{1}{N}\sum_{i=0}^{N-1}\mathbf{x}^i$

autocorrelation: $\mathbf{K} = \mathbf{R}_{\mathbf{xx}} = E\{\underbrace{\mathbf{x}\mathbf{x}^T}_{\text{dyad. product!}}\} = \{E(x_i x_j)\} \approx \frac{1}{N}\sum_{i=0}^{N-1}\begin{bmatrix} x_0^i \cdot x_0^i & x_0^i \cdot x_1^i & x_0^i \cdot x_2^i \\ x_1^i \cdot x_0^i & x_1^i \cdot x_1^i & x_1^i \cdot x_2^i \\ x_2^i \cdot x_0^i & x_2^i \cdot x_1^i & x_2^i \cdot x_2^i \end{bmatrix}$

The elements of the correlation matrix describe the correlation between the particular vector elements $\{x_0,x_1\}$, $\{x_0,x_2\}$ ... in time/local direction with increasing distance between elements:



autocovariance: $\mathbf{C}_{\mathbf{xx}} = E\{(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^T\} = \mathbf{R}_{\mathbf{xx}} - \overline{\mathbf{x}}\overline{\mathbf{x}}^T$

cross-correlation: $\mathbf{R}_{\mathbf{xy}} = E\{\mathbf{x}\mathbf{y}^T\}$

cross-covariance: $\mathbf{C}_{\mathbf{xy}} = E\{(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{y}-\overline{\mathbf{y}})^T\} = \mathbf{R}_{\mathbf{xy}} - \overline{\mathbf{x}}\overline{\mathbf{y}}^T$

Gaussian distributions stay so under *linear* transformations

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^N \det(\mathbf{C_{xx}})}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_\mathbf{x})^T \mathbf{C_{xx}}^{-1}(\mathbf{x}-\boldsymbol{\mu}_\mathbf{x})}$$

from $\mathbf{y} = \mathbf{A}\mathbf{x}$ follows for $p(\mathbf{y})$ a normal distribution with:

$$\boldsymbol{\mu}_\mathbf{y} = \mathbf{A}\boldsymbol{\mu}_\mathbf{x}$$

and:

$$\mathbf{C_{yy}} = E\{(\mathbf{y}-\bar{\mathbf{y}})(\mathbf{y}-\bar{\mathbf{y}})^T\} = \mathbf{A}E\{(\mathbf{x}-\bar{\mathbf{x}})(\mathbf{x}-\bar{\mathbf{x}})^T\}\mathbf{A}^T$$

$$\Rightarrow \mathbf{C_{yy}} = \mathbf{A}\mathbf{C_{xx}}\mathbf{A}^T$$

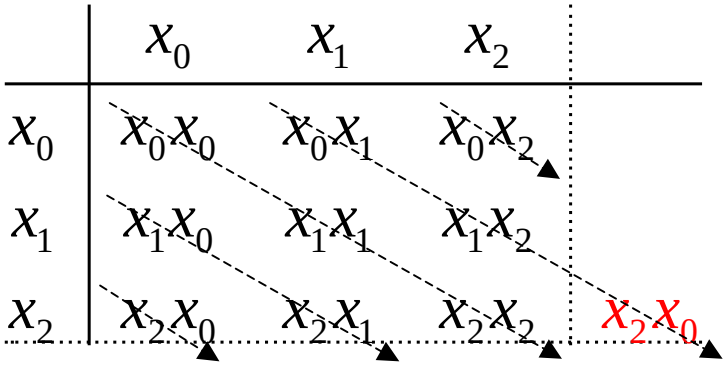
Calculating the ACF from the autocorrelation matrix

The values of the linear (cyclic) ACF result from diagonal sums of the (periodical continued) autocorrelation matrix:

linear ACF:

$$\begin{array}{lll} x_0 & x_1 & x_2 \\ x_0 & x_1 & x_2 \\ \hline x_0^2 + x_1^2 + x_2^2 \end{array} \qquad \begin{array}{lll} x_0 & x_1 & x_2 \\ x_0 & x_1 & x_2 \\ \hline x_0 x_1 + x_1 x_2 \end{array} \qquad \begin{array}{lll} x_0 & x_1 & x_2 \\ x_0 & x_1 & x_2 \\ \hline x_0 x_2 \end{array}$$

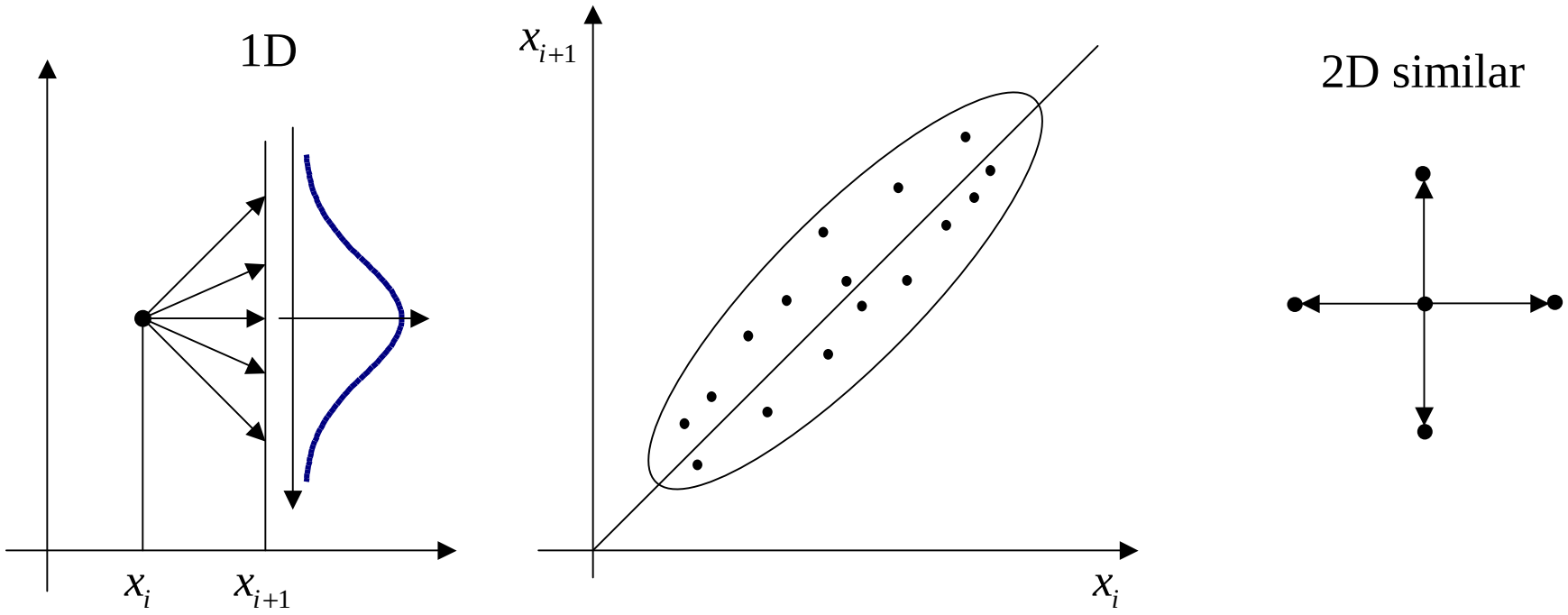
autocorrelation matrix



Decorrelation of neighbouring signal or pixel values in the vector space

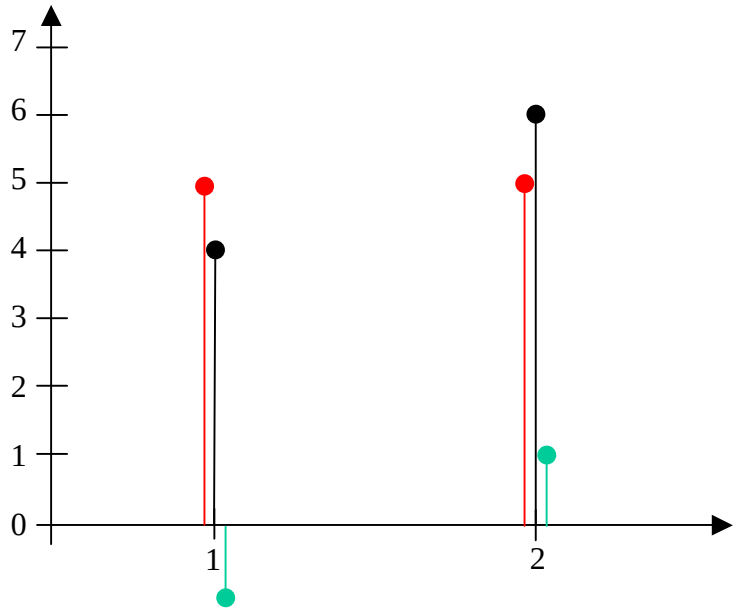
Given a signal or pixel value; what is the probability of the neighbouring signal amplitude taking similar values?

In general high correlation to neighbouring values! (1st angle bisection in vector space)



A simple example for the transition to a new feature space using an orthogonal transformation

Given a signal with two scan values



$$\mathbf{x} = \begin{bmatrix} 4 \\ 6 \end{bmatrix} = 4\begin{bmatrix} 1 \\ 0 \end{bmatrix} + 6\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$ original space

$$= 5\begin{bmatrix} 1 \\ 1 \end{bmatrix} + 1\begin{bmatrix} -1 \\ 1 \end{bmatrix} = 5\sqrt{2} \cdot \tfrac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \sqrt{2} \cdot \tfrac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ 1 \end{bmatrix}$$ feature space

Selecting only the first component (subspace) in the original space, results in an approximation performance of (*omittung one scan value strikes !!*):

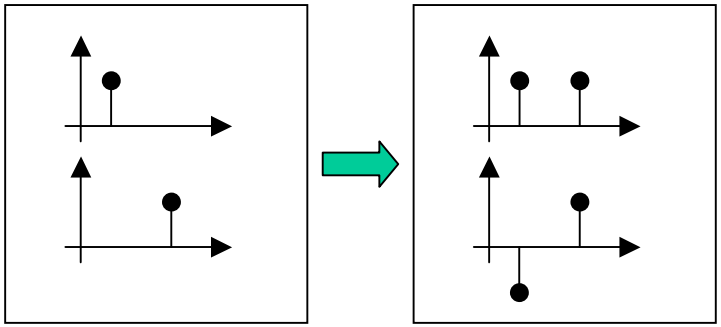
$$\frac{\left\| 4 \quad 0 \right\|}{\left\| 4 \quad 6 \right\|} = \frac{4}{7{,}21} = \boxed{55\%}$$

In contrast in the new feature space:

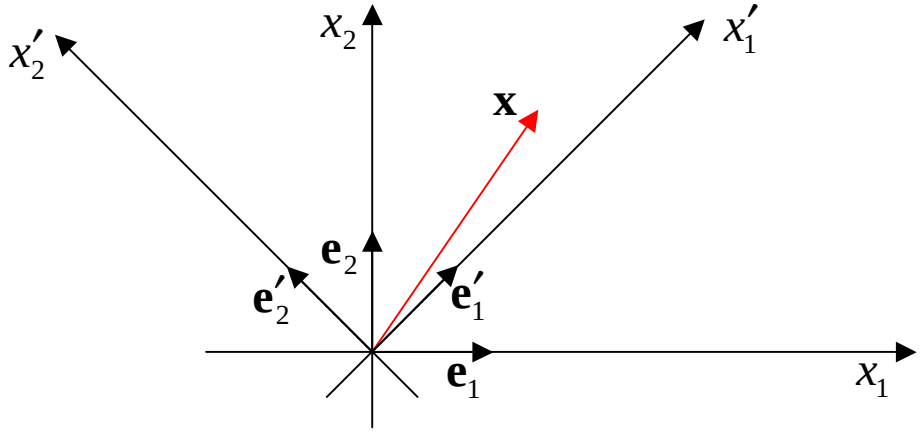
$$\frac{\left\| 5\sqrt{2} \quad 0 \right\|}{\left\| 5\sqrt{2} \quad \sqrt{2} \right\|} = \frac{7{,}07}{7{,}21} = \boxed{98\%}$$

Also: the new values do not correlate!

Representing in the vector space by rotation of the coordinate system with orthogonal transformation



original space feature space



$$\boxed{\mathbf{x}' = \mathbf{A}^T\mathbf{x}}$$

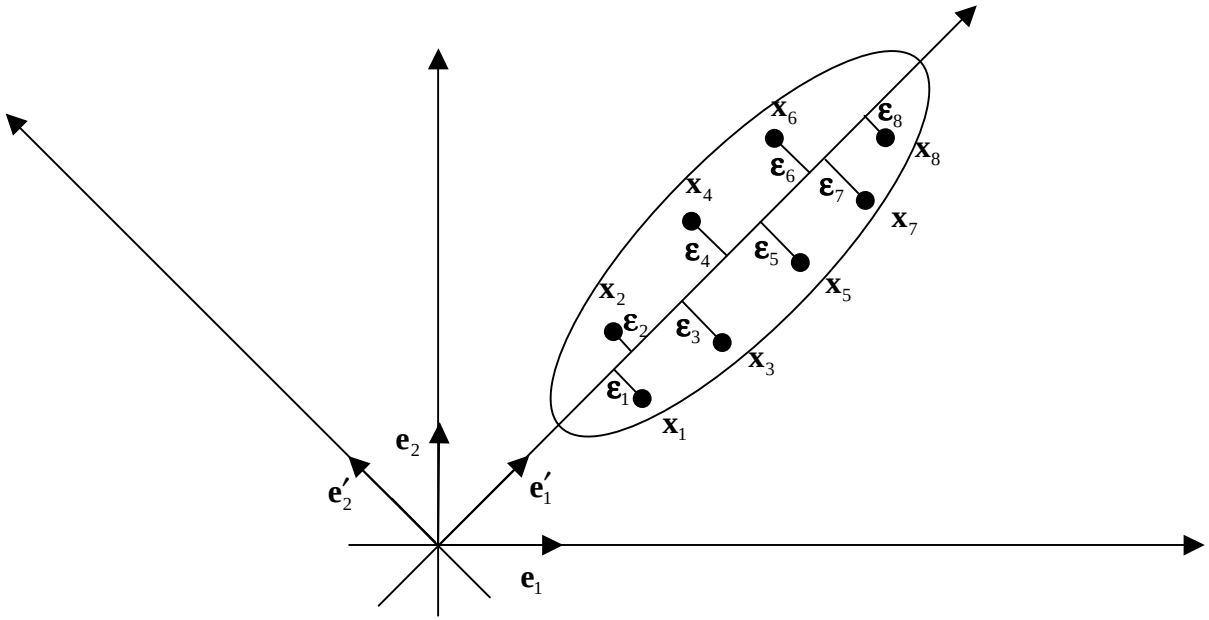
with: $\mathbf{A}^T = \frac{1}{\sqrt{2}}\begin{bmatrix} +1 & +1 \\ -1 & +1 \end{bmatrix}$ and: $\left(\mathbf{A}^T\right)^{-1} = \left(\mathbf{A}^T\right)^T = \mathbf{A} = \frac{1}{\sqrt{2}}\begin{bmatrix} +1 & -1 \\ +1 & +1 \end{bmatrix}$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 = x_1\begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_2\begin{bmatrix} 0 \\ 1 \end{bmatrix} = x_1'\mathbf{e}_1' + x_2'\mathbf{e}_2' = x_1'\tfrac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} + x_2'\tfrac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ 1 \end{bmatrix} = \underbrace{\frac{(x_1 + x_2)}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{\text{non-variable part}} + \underbrace{\frac{(x_2 - x_1)}{2}\begin{bmatrix} -1 \\ 1 \end{bmatrix}}_{\text{variable part}}$$

mit: $\mathbf{e}_1' = \mathbf{A}\mathbf{e}_1$ und $\mathbf{e}_2' = \mathbf{A}\mathbf{e}_2$

Optimal feature selection with unitary transformations

(Karhunen-Loeve or principal axis transformation)



task: find new base vectors

$$\{\mathbf{e}_i\} \xrightarrow{\mathbf{A}^T} \{\mathbf{e}'_i\}$$

$$\boxed{\mathbf{x}' = \mathbf{A}^T \mathbf{x}}$$

Unitary transformation, for real orthogonal transf. => rotation of the coordinate system

$$\langle \mathbf{Ax}, \mathbf{Ay} \rangle = \langle \mathbf{x}, \mathbf{A}^* \mathbf{Ay} \rangle \overset{!}{=} \langle \mathbf{x}, \mathbf{y} \rangle$$

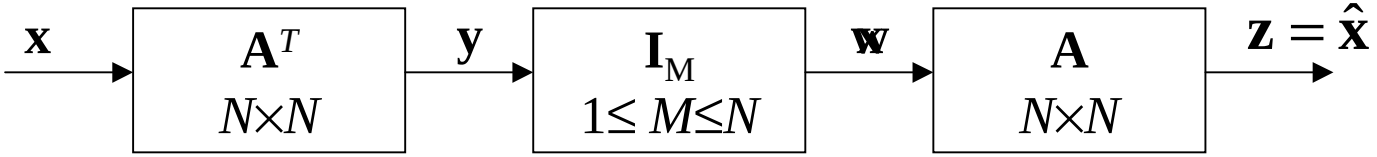
$$\Rightarrow \ \mathbf{A}^* \mathbf{A} = \mathbf{I} \ \Rightarrow \ \mathbf{A}^{-1} = \mathbf{A}^*$$

If the corresponding basis system is arbitrary (and is known by sender and reciever), a single vector element **x** can be characterized by a scalar value, if the first base vector $\mathbf{e}'_1$ is chosen in direction of **x** (element occurs or not):

$$\mathbf{x} = \alpha \frac{\mathbf{x}}{\|\mathbf{x}\|} + 0 \cdot \mathbf{e}'_2 + 0 \cdot \mathbf{e}'_3 + \ldots$$

In general the point is to find an optimal transformation into an appropiate coordinate system for a complete *ensemble of vectors*, in order to characterize the elements of the ensemble with as few coefficients as possible.

We start with determing the first new basis vector $\mathbf{e}_1'$, which we choose so that the *approximation error* for the ensemble of n vectors is *minimal*, or the sought space direction, that represents a *maximal information* of the ensemble. According to the projection sentence the smallest error results from projection to the subspace, which is represented by $\mathbf{e}_1'$; sought is the correct space direction. An optimal solution is sought starting from a quality factor.



$$\mathbf{w} = \mathbf{I}_M \mathbf{y} = y_1 \mathbf{e}'_1 + y_2 \mathbf{e}'_2 + \ldots + y_M \mathbf{e}'_M$$

the best approximation must apply for arbitrary M!

Starting from a square quality factor results:

$$J = \tfrac{1}{n}\{\|\boldsymbol{\varepsilon}_1\|^2 + \|\boldsymbol{\varepsilon}_2\|^2 + \ldots + \|\boldsymbol{\varepsilon}_n\|^2\}$$

$$= \tfrac{1}{n}\{\|\mathbf{x}_1 - \langle \mathbf{x}_1, \mathbf{e}'_1 \rangle \mathbf{e}'_1\|^2 + \|\mathbf{x}_2 - \langle \mathbf{x}_2, \mathbf{e}'_1 \rangle \mathbf{e}'_1\|^2 + \ldots\}$$

$$= E\{\|\mathbf{x} - \underbrace{\langle \mathbf{x}, \mathbf{e}'_1 \rangle \mathbf{e}'_1}_{\mathbf{Px}}\|^2\}$$

$\mathbf{Px}$
projection to
$S(\mathbf{e}'_1)$

for the ensemble: $\mathbf{x} := \{\mathbf{x}_i\} \quad i = 1, 2, \ldots, n$

ZR: For a orthogonal *projection* to a subspace **Px** applies according to Pythagoras:

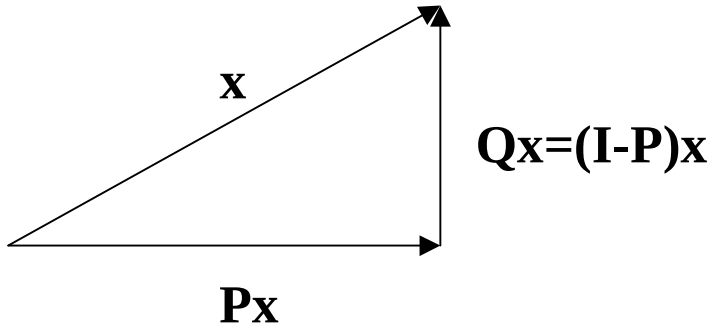
$$\|\mathbf{x}\|^2 = \|\mathbf{Px}\|^2 + \|(\underbrace{\mathbf{I}-\mathbf{P}}_{\mathbf{Q}})\,\mathbf{x}\|^2$$

$$\Rightarrow \boxed{\|\mathbf{x}-\mathbf{Px}\|^2 = \|\mathbf{x}\|^2 - \|\mathbf{Px}\|^2}$$

with: $\langle \mathbf{x}, \mathbf{e}_1' \rangle \mathbf{e}_1' = \underbrace{(\mathbf{e}_1' \mathbf{e}_1'^T)}_{\mathbf{P}}\,\mathbf{x}$

for real vectors applies:

$\langle \mathbf{a}, \mathbf{b} \rangle = \langle \mathbf{b}, \mathbf{a} \rangle$



and thus:

$$\langle\langle \mathbf{x},\mathbf{e}_1'\rangle\mathbf{e}_1',\langle \mathbf{x},\mathbf{e}_1'\rangle\mathbf{e}_1'\rangle = \langle \mathbf{x},\mathbf{e}_1'\rangle^2 \underbrace{\langle \mathbf{e}_1',\mathbf{e}_1'\rangle}_{=1}$$

$$J = E\{\|\mathbf{x}\|^2 - \overbrace{\|\langle \mathbf{x},\mathbf{e}_1'\rangle\mathbf{e}_1'\|^2}^{=1}\}$$

$$= E\{\|\mathbf{x}\|^2 - \langle \mathbf{x},\mathbf{e}_1'\rangle^2\}$$

(maximization of the squares of the FC)

$$= E\{\|\mathbf{x}\|^2 - \langle \mathbf{e}_1',\mathbf{x}\rangle\langle \mathbf{x},\mathbf{e}_1'\rangle\}$$

Useful formulas:

$$\langle \mathbf{a},\mathbf{b}\rangle\langle \mathbf{c},\mathbf{d}\rangle = \mathbf{a}^T(\overbrace{\mathbf{b}\mathbf{c}^T}^{\text{dyad. product}})\mathbf{d}$$

$$\mathbf{a}\langle \mathbf{b},\mathbf{c}\rangle = (\mathbf{a}\mathbf{b}^T)\mathbf{c}$$

$$\langle \mathbf{a},\mathbf{b}\rangle = \mathrm{tr}(\mathbf{a}\mathbf{b}^T)$$

The inner product can be calculated over the trace of the outer product!

$$\langle \mathbf{A},\mathbf{B}\rangle = \mathrm{tr}(\mathbf{A}\ \underbrace{\mathbf{B}^*}_{\mathbf{B}\ \text{adjoint}})$$

for image matrices

and thus:

$$J = E\{\|\mathbf{x}\|^2 - \mathbf{e}_1'^T(\mathbf{x}\mathbf{x}^T)\mathbf{e}_1'\}$$

$$= \underbrace{E\{\|\mathbf{x}\|^2\}}_{\sigma_x^2} - \mathbf{e}_1'^T \underbrace{E\{\mathbf{x}\mathbf{x}^T\}}_{\mathbf{R_{xx}}}\mathbf{e}_1'$$

σ_x^2: variance of x

$\mathbf{R_{xx}}$: auto-correlation matrix

$$\Rightarrow \boxed{J = \mathrm{tr}(\mathbf{R_{xx}}) - \mathbf{e}_1'^T\mathbf{R_{xx}}\mathbf{e}_1' \overset{!}{=} \min_{\mathbf{e}_1'}}$$

Side condition: the new basis vector is a unit vector :

$$\|\mathbf{e}_1'\|^2 = \langle \mathbf{e}_1', \mathbf{e}_1' \rangle = 1$$

The first term in J is constant and thus J is minimized, if the following term is maximized:

$$J' = \mathbf{e}_1'^T \mathbf{R}_{\mathbf{xx}} \mathbf{e}_1' \stackrel{!}{=} \max_{\mathbf{e}_1'}$$

Involving the side condition in the maximization of J' by a Lagrange approach:

$$J'' = \mathbf{e}_1'^T \mathbf{R}_{\mathbf{xx}} \mathbf{e}_1' + \lambda(1 - \langle \mathbf{e}_1', \mathbf{e}_1' \rangle)$$

using: $\frac{\partial \langle \mathbf{y}, \mathbf{y} \rangle}{\partial \mathbf{y}} = 2\mathbf{y}$ and: $\frac{\partial (\mathbf{y}^T \mathbf{R} \mathbf{y})}{\partial \mathbf{y}} = 2\mathbf{R}\mathbf{y}$ (if **R** symm.)

the neccessary constraint for an extremum results in:

$$\frac{\partial J''}{\partial \mathbf{e}_1'} = 2(\mathbf{R}_{\mathbf{xx}} \mathbf{e}_1' - \lambda \mathbf{e}_1') \stackrel{!}{=} \mathbf{0}$$

and thus the Eigenvalue equation:

$$\boxed{\mathbf{R}_{\mathbf{xx}} \mathbf{e}_1' = \lambda \mathbf{e}_1'}$$

Inserting in J' results in: $\boxed{J' = \lambda \mathbf{e}_1'^T \mathbf{e}_1' = \lambda}$

This term is maximized, if the maximal Eigenvalue $\lambda_1 = \lambda_{max}$ and the corresponding Eigenvector is chosen.

The one-dimensional subspace along $\mathbf{e}_1'$ is separated, within the remaining subspace one choses the second basis vector $\mathbf{e}_2' \Rightarrow \lambda_2$ and the second largest Eigenvalue, and so on.

Approximation errors

The approximation error with M components ($1 \leq M \leq N$) results from:

$$J_M = E\{\|\mathbf{x}-\mathbf{z}\|^2\} = E\{\|\mathbf{x}-\underbrace{\mathbf{A}\mathbf{I}_M\mathbf{A}^T}_{\substack{\text{orthog.}\\ \text{projection } \mathbf{P}}}\mathbf{x}\|\} \quad \text{mit:} \quad \mathbf{A} = [\mathbf{e}'_1, \mathbf{e}'_2, \ldots, \mathbf{e}'_N]$$

to the subspace, which is spaned by the first M Eigenvectors

$$J_M = E\{\|\mathbf{x}-\mathbf{P}\mathbf{x}\|^2\} = E\{\|\mathbf{x}\|^2 - \|\mathbf{P}\mathbf{x}\|^2\} = E\{\|\mathbf{x}\|^2 - \langle\mathbf{P}\mathbf{x}, \mathbf{P}\mathbf{x}\rangle\}$$

applies:

$\mathbf{A}^T\mathbf{A} = \mathbf{I}$ ($\mathbf{A}^T$ is orthogonal)

$\mathbf{A}^T\mathbf{R}_{\mathbf{xx}}\mathbf{A} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$ with: $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \lambda_N$

the projection matrix is idempotent and symmetric

and thus:

$$\langle\mathbf{P}\mathbf{x}, \mathbf{P}\mathbf{x}\rangle = \langle\mathbf{x}, \mathbf{P}^T\mathbf{P}\mathbf{x}\rangle = \langle\mathbf{x}, \mathbf{P}^2\mathbf{x}\rangle = \langle\mathbf{x}, \mathbf{P}\mathbf{x}\rangle = \langle\mathbf{x}, \mathbf{A}\mathbf{I}_M\mathbf{A}^T\mathbf{x}\rangle$$

$$= \langle\mathbf{I}_M\mathbf{A}^T\mathbf{x}, \mathbf{A}^T\mathbf{x}\rangle = \text{tr}(\mathbf{I}_M\mathbf{A}^T(\mathbf{x}\mathbf{x}^T)\mathbf{A})$$

Approximation errors

and inserted into the quality measure:

$$
\begin{aligned}
J_M &= E\{\mathrm{tr}(\mathbf{x}\mathbf{x}^T)\} - E\{\mathrm{tr}(\mathbf{I}_M \mathbf{A}^T(\mathbf{x}\mathbf{x}^T)\mathbf{A})\} \\
&= \mathrm{tr}(\mathbf{R}_{\mathbf{xx}} - \mathbf{I}_M \underbrace{\mathbf{A}^T \mathbf{R}_{\mathbf{xx}} \mathbf{A}}_{\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)}) \\
&= \mathrm{tr}(\mathbf{R}_{\mathbf{xx}}) - \sum_{i=1}^{M} \lambda_i = \sum_{i=M+1}^{N} \lambda_i
\end{aligned}
$$

i.e., the approximation error corresponds to the sum of the Eigenvalues, that have not been considered.

The Karhunen-Loéve transformation (KLT)

The Karhunen-Loéve transform is defined as:

$$\boxed{\mathbf{y} = \mathbf{A}^T \mathbf{x}} \qquad \text{KLT}$$

$$\boxed{\mathbf{x} = \mathbf{A}\mathbf{y}} \qquad \text{KLT}^{-1}$$

and it applies:

$$\mathbf{R}_{\mathbf{yy}} = E\{\mathbf{y}\mathbf{y}^T\} = E\{\mathbf{A}^T(\mathbf{x}\mathbf{x}^T)\mathbf{A}\} = \mathbf{A}^T\mathbf{R}_{\mathbf{xx}}\mathbf{A} = \Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$$

with: $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \lambda_N$

The KLT can be calculated (as seen here) based on the *correlation matrix*, or based on the *autocovariance matrix* (the expected value is subtracted):

$$\boxed{\mathbf{y} = \mathbf{A}^T(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}})}$$

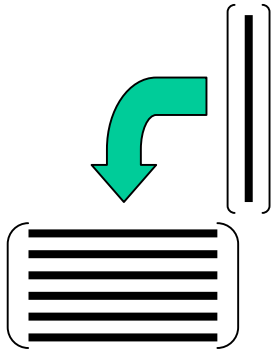
Interpretation of the KLT

Fourierseries, developing the $\mathbf{e}'_i$

$$\mathbf{z} = \overbrace{\mathbf{A}\,\mathbf{I}_M \quad \underbrace{\mathbf{A}^T\mathbf{x}}_{\text{calculating the Fouriercoefficients } \alpha_i \text{ projection to the space spaned by } \mathbf{e}'_i}}^{\text{projection to the subspace}} \quad \overbrace{\sum \alpha_i \mathbf{e}'_i}^{\text{Fourierseries, developing the } \mathbf{e}'_i}$$

projection to the subspace:

$$\mathbf{y} = \mathbf{I}_M \mathbf{A}^T \mathbf{x} = \begin{bmatrix} \langle \mathbf{e}'_1, \mathbf{x} \rangle \\ \langle \mathbf{e}'_2, \mathbf{x} \rangle \\ \vdots \\ \langle \mathbf{e}'_M, \mathbf{x} \rangle \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_M \end{bmatrix}$$



Fourierseries (developing the column vectors of **A**):

$$\mathbf{z} = \mathbf{A}\mathbf{w} = \sum_{i=1}^{M} \underbrace{\alpha_i}_{\text{FC}} \cdot \underbrace{\mathbf{e}'_i}_{\substack{\text{column vectors} \\ \text{of } \mathbf{A}}}$$

Minimizing the error is equal to maximizing the energy (length^2) in the transformed area or the maximization of the square sum of the Fouriercoefficients.

KLT for images (2D)

The above approach can be applied directly to a vector, which consists of *stacked row vectors* of an *image matrix* of dimension $N \times N$ (since we are only concerned about the complete sum error!). So now we have to find the Eigenvalue for a symmetrical matrix of dimension $N^2 \times N^2$. An Eigenvalue problem for matrix $N \times N$ requires $O(N^3)$ operations, so here: $\boldsymbol{O(N^6)}$

If an ensemble of images $\mathbf{X} := \{\mathbf{X}_i\}$ of dimension $N \times N$ can be modeled by the dyadic product of two one-dimensional ensembles of dimension $N \times 1$

$$\mathbf{x}^1 := \{\mathbf{x}_i^1\} \qquad \mathbf{x}^2 := \{\mathbf{x}_i^2\}$$

according to:

$$\mathbf{X} := \mathbf{x}^1 \mathbf{x}^{2^T}$$ i.e. $\mathbf{X}$ is separable!

KLT for images(2D)

Thus for every one-dimensional ensemble a KLT can be calculated and thus:

$$\mathbf{Y} = \mathbf{A}^{1T}(\mathbf{x}^1\mathbf{x}^{2T})\mathbf{A}^2 = \mathbf{A}^{1T}\mathbf{X}\mathbf{A}^2$$

2D-KLT for separable images

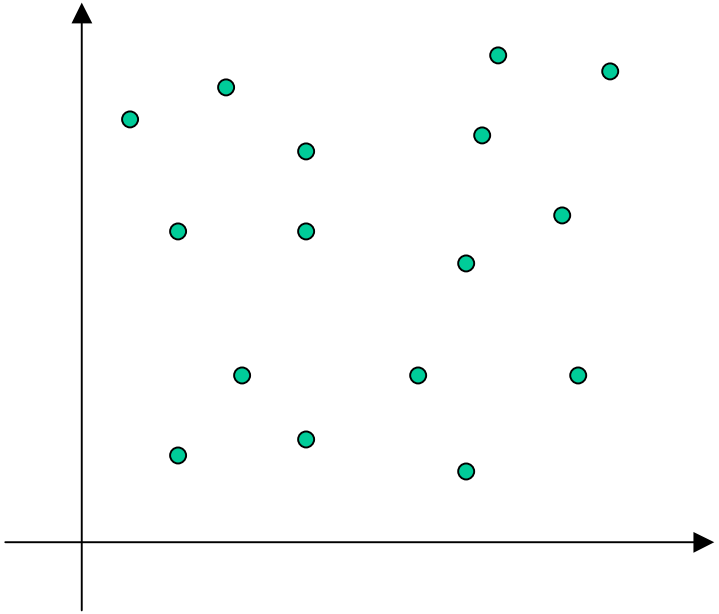
Thus only 2 Eigenvalues of dimension $N \times N$ are to be calculated. This results in a computing time improvement of: $O(N^6) / O(N^3) = O(N^3)$

The transformation with separable kernel can also be reduced, namely from $O(N^4)$ to $O(2N^3)=O(N^3)$.

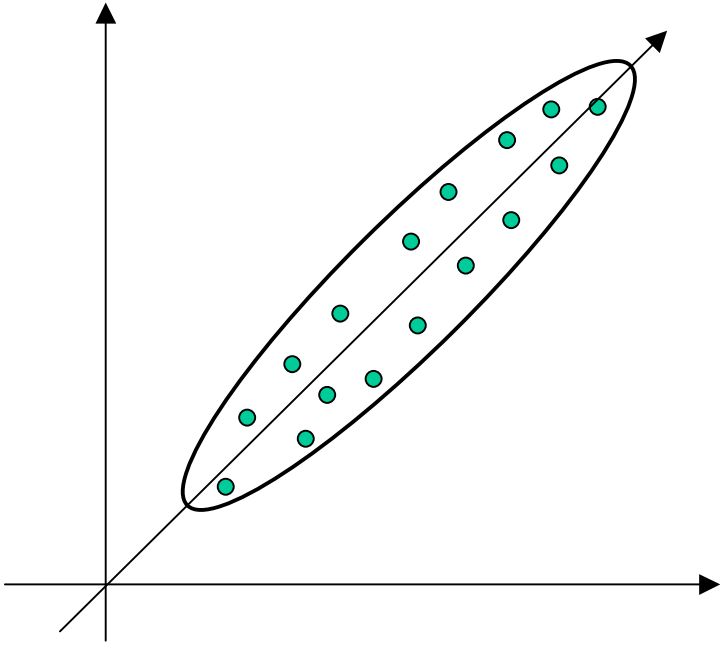
Properties of KLT

- advantages:
  - The KLT is optimal (wrt. the square error) in terms of best representation in subspaces with orthogonal basis.
    For highly correlating vector elements results a high information condensation in few elements of the KLT. The KLT profits from high correlations in the vector elements.
  - Since $\mathbf{R}_{yy}$ is a diagonal matrix, the values in y do not correlate!
- disadvantages:
  - The KLT is *data dependent* and must be calculated separately for every dataset.
  - Also there is *no fast* algorithm for KLT.

Data reduction depending on degree of correlation

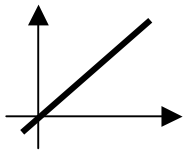


data un-correlated (white process)
KLT is of no significance

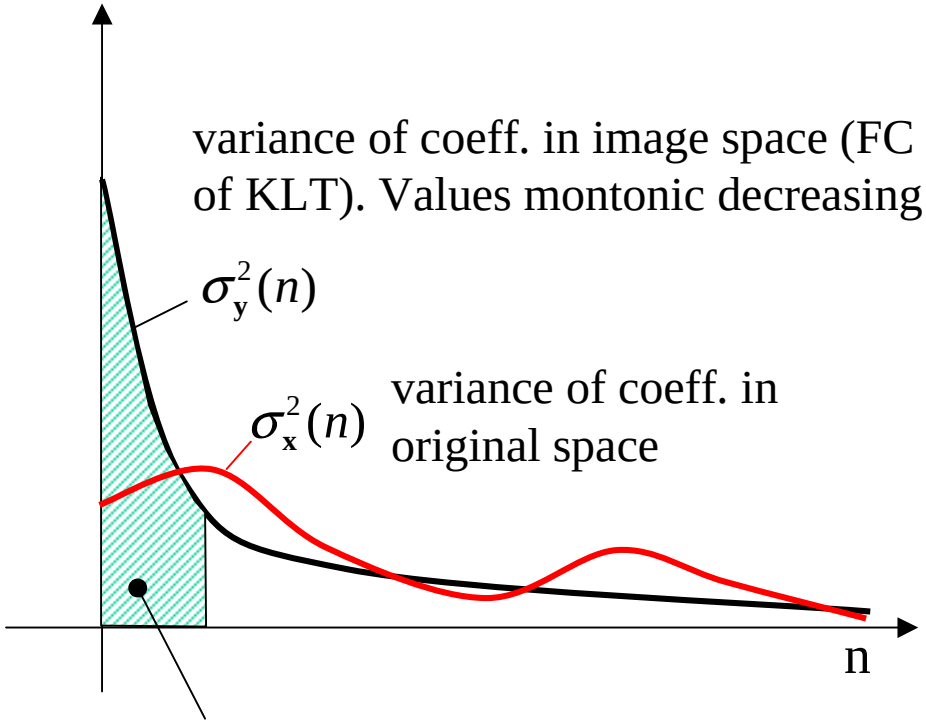


data highly correlated. KLT has high effect.

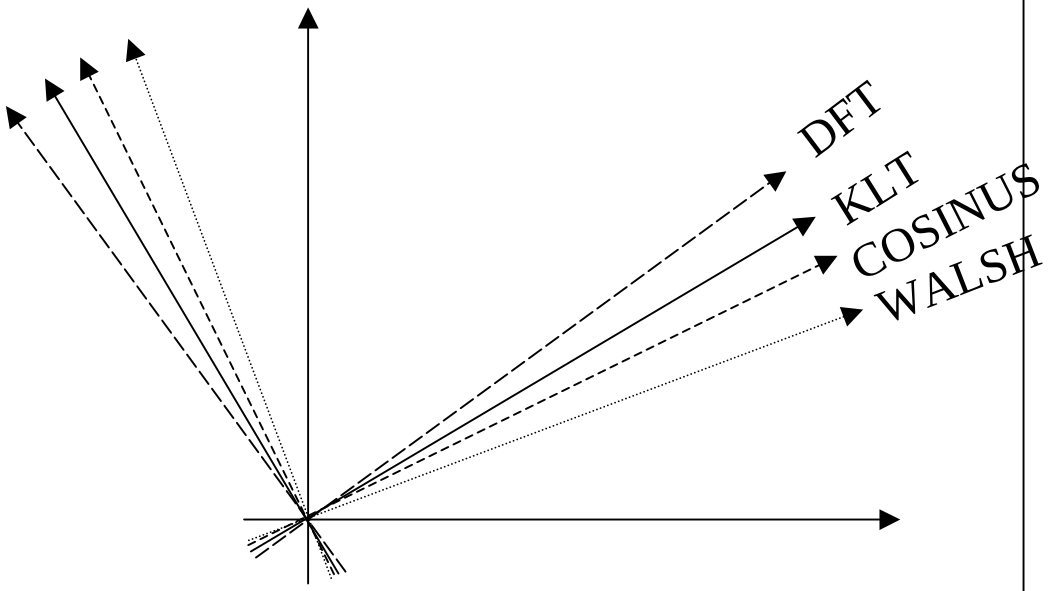
extreme case: images with constant gray value
(a vector is sufficient for representation)



Properties of KLT



Every subtotal is independent of M maximal!



behaviour of different unitary transformations

Further properties of KLT

The KLT guarantees, that the variances of the transformed features (principal diagonal elements of the covariance matrix) are maximal unbalanced (minimal entropy):

$$\mathbf{y} = \mathbf{A}^T(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})$$

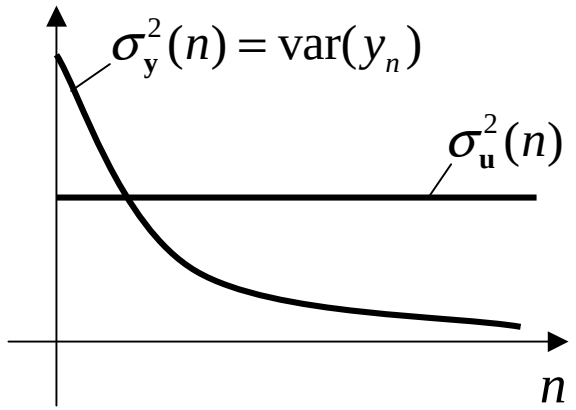
A maximization of the entropy, or a constant variance of all features can be obtained by a ***whitening transformation***:

$$\mathbf{u} = \Lambda^{-1/2}\mathbf{A}^T(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}) \qquad \Lambda^{-1/2} = \mathrm{diag}(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots, \lambda_N^{-1/2})$$

All features have same variance var(u_i)=1 (sterical invariant relations). The energy is distributed equally to all features.

The variables remain un-correlated when multiplied with a diagonal matrix!

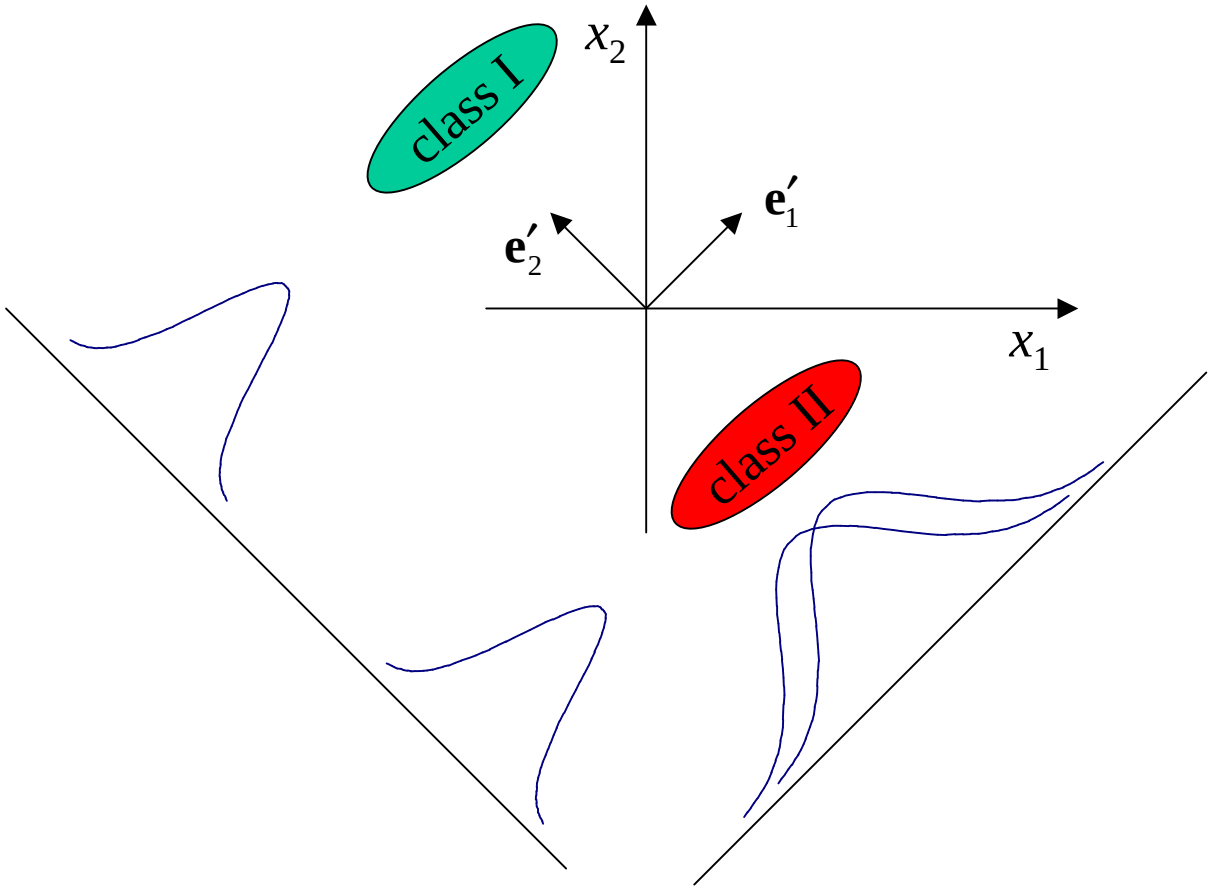
Whitening e.g. is needed to obtain a robustness as greatest as possible if a component is left out (e.g. in transfer mode systems).



Further properties of KLT

The optimality of KLT wrt. the minimal error square leads to a best information condensation of *one* ensemble and allows a selection of *M* dominant features of *N* observed values. Question: how can all data be *represented* in the best way?

This does not lead neccessarily to a best class representation, if several classes are too different. An optimization wrt. this leads to the so-called ***discriminantion analysis***. Question: how can the data be *discriminated* in the best way?



In this example, the features of the first Eigenvector overlap, while the feature of the second Eigenvector distinguishes the classes!

Assumption: variance along $\mathbf{e}_1'$ greater than variance along $\mathbf{e}_2'$.

covariance matrix is computed for the entire ensemble, because only one form of feature selection can be chosen!