University of Verona

A.Y. 2021-22

Machine Learning & Artificial Intelligence

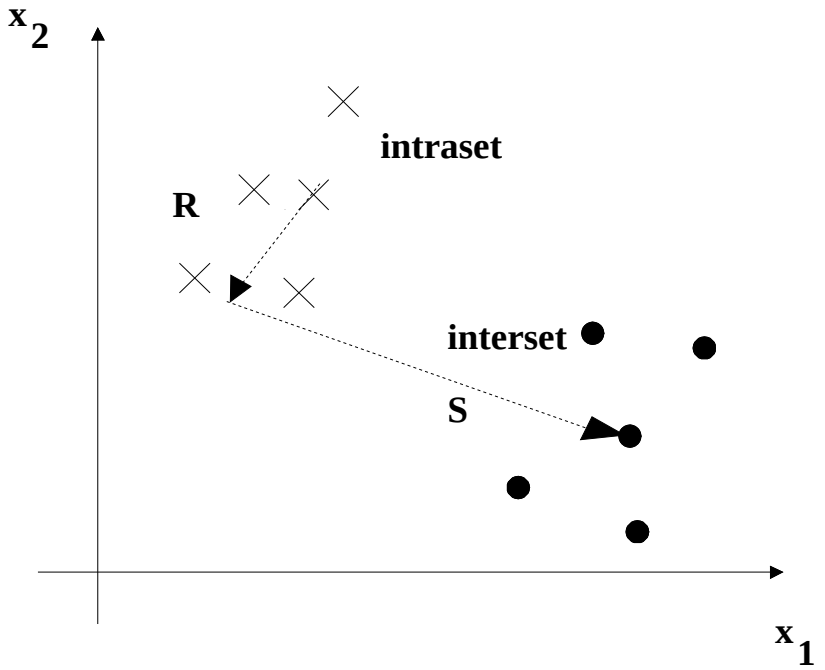
Linear transformations, Fisher's method
Feature extraction & selection
Principal Component Analysis

Vittorio Murino

More on Linear Transformations

- The basic idea is to look for a transformation $\mathbf{W}$ that leads from poorly structured $\mathbf{y}$ samples into a new simpler $\mathbf{x}$ set for classification:

$$\mathbf{x} = \mathbf{W}\,\mathbf{y}$$



- The problem is determining the transformation matrix $\mathbf{W}$.
- Two types of distances are considered:

$$\text{INTERSET} \Rightarrow S \qquad\qquad \text{INTRASET} \Rightarrow R$$

- The S parameter is defined as the average of all possible distances between two samples of different classes.

$$S = \frac{1}{M_1 M_2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} d^2\left[\mathbf{y}_i^{(1)}, \mathbf{y}_j^{(2)}\right]$$

- The R parameter, on the other hand, considers all possible distances within the same class.

$$R = \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j=1}^{M} d^2\left[\mathbf{y}_i, \mathbf{y}_j\right], \text{ with M equal to } M_1 \text{ or } M_2$$

- By performing a transformation $\mathbf{W}$, we want to make S maximum and R minimum in order to obtain a good separability of the classes.

- It can therefore be defined as a goal:

$$Q(\mathbf{x}) = \left. \frac{R_1 + R_2}{S} \right|_{min}$$

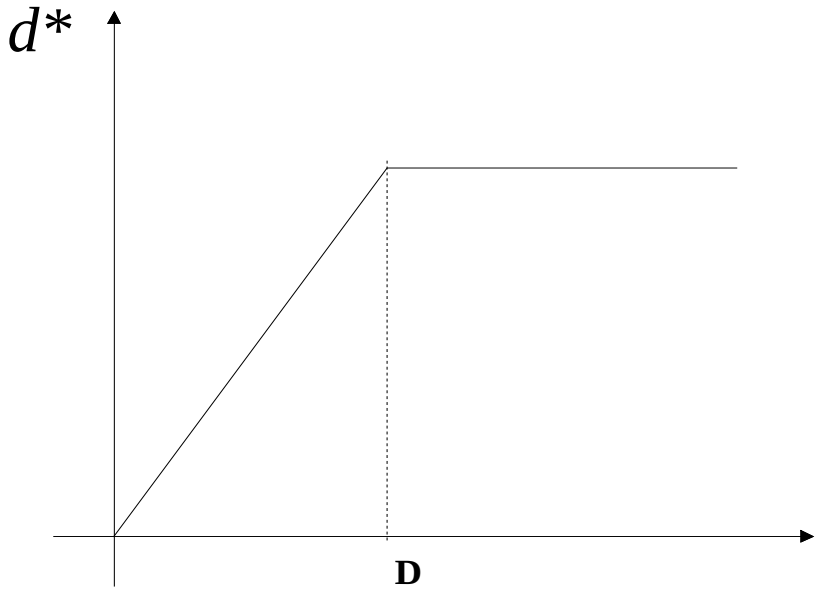
  or similar criteria that tend to maximize the Interset distance and minimize the Intraset distance.

- The transformation to have a minimal $Q(\mathbf{x})$ becomes nonlinear (and therefore very complicated).
- There are different types of distances:
  - Euclidean: it is not very good because it weighs much the more distant points (so we resort to a type of distance that tends to saturate);
  - saturation: the disadvantage is that with this type of distance discontinuities are introduced;
  - Continuous fittings: this is one of the simplest solutions.

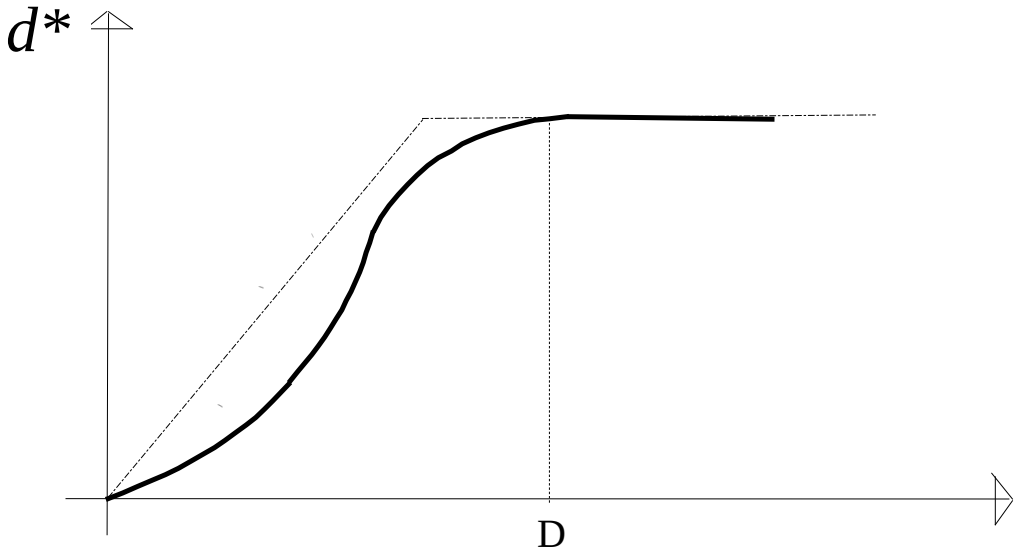
- Therefore, the best choice for the distance is to use a function with continuous first derivative approximating the Euclidean distance up to a certain threshold, and then to consider a fixed distance, in order not to weigh too much the very distant samples.
- A less coarse form of distance turns out to be a sigmoid-like function:

$$d^*(\mathbf{y_1},\mathbf{y_2}) = 1 - \exp\left\{-\frac{1}{2D^2} d^2(\mathbf{y_1},\mathbf{y_2})\right\}$$

where d^2 is the Euclidean distance, which is continuous and tends to saturate.



- The considerations on the type of distance apply for both R and S.
- All these considerations are fine if the samples are distributed in a statistical way (e.g., cloud) and not in a functional way (e.g., lamellar).



- Suppose we want to make a linear transformation $\mathbf{x} = \mathbf{W}\,\mathbf{y}$ between two spaces in order to verify the above criteria.
- In other words, we consider $\mathbf{W}$ matrix, then we operate the transformation reducing the size of the starting space ($\mathbf{y}$).

$\Rightarrow \quad \dim(\mathbf{y}) = m$

$\dim(\mathbf{x}) = n$, then $\dim(\mathbf{W}) = n \times m$

$$\mathbf{x} = \mathbf{W}\,\mathbf{y} \qquad \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{bmatrix} w_{11} & \cdots\cdots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots\cdots & w_{nm} \end{bmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

Then, we have

$$S = \frac{1}{M_1 M_2} \sum_{q=1}^{M_1} \sum_{p=1}^{M_2} \left\{ \mathbf{W} \left(\mathbf{y}_q^{(1)} - \mathbf{y}_p^{(2)} \right) \right\}^2$$

And developing:

$$S = \frac{1}{M_1 M_2} \sum_{q=1}^{M_1} \sum_{p=1}^{M_2} \sum_{i=1}^{n} \left\{ \sum_{j=1}^{m} w_{ij} \left(y_{q_j}^{(1)} - y_{p_j}^{(2)} \right) \right\}^2$$

$$R_1 = \frac{1}{M_1 (M_1 - 1)} \sum_{q=1}^{M_1} \sum_{p=1}^{M_1} \sum_{i=1}^{n} \left\{ \sum_{j=1}^{m} w_{ij} \left(y_{q_j}^{(1)} - y_{p_j}^{(1)} \right) \right\}^2$$

and similarly for the other class (ω_2).

- The unknowns are obviously the w_{ij}
- Whether the solution is easy or not depends on the objective function to be chosen.
- Let's consider the following cases:

1) **W** diagonal ($w_{ij} = 0$ if $i \neq j$ and $w_{ij} \neq 0$ if $i = j$)

Imposing as objective: $(R_1 + R_2)$ as minimum

and with the constraint $\Sigma_k \, w_{kk} = 1$, which is the Lagrangian minimum condition (otherwise called *constant perimeter* constraint), we get at the following solution (the latter constraint serves to prevent the arrival space from "exploding", that is, the arrival space $R_1 + R_2$ must be contained in the starting space):

$$\sum_k w_{kk} = 1 \Rightarrow w_{kk} = \frac{1}{\sigma_k^2 \sum_{j=1}^{n} \left(\frac{1}{\sigma_j^2} \right)}$$

with $\sigma_k^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(y_{i_k}^{(l)} - \bar{y}_k^{(l)}\right)^2$

and $N = M_1 + M_2$ if $(R_1 + R_2)$ is minimum

or $N = M_1$ or M_2 if R_1 or R_2 minimum, respectively; l label of the class.

- The term σ_k^2 is defined as the variance of the samples along the k-th component (*k-th feature* of the sample).
- In this way, a small variance implies that the k-th measurement is more reliable, and vice versa, such that the most reliable measurement is weighted more.
- Another constraint one can impose is $\prod w_{kk} = 1$ (constant volume constraint):

$$\prod_k w_{kk} = 1 \Rightarrow w_{kk} = \frac{1}{\sigma_k}\left(\prod_{j=1}^{n}\sigma_j\right)^{1/n}$$

which is inversely proportional to the standard deviation of the k-th measure.

2) **W** arbitrary

The objective R_1+R_2 minimum must now be reached with the constraint $R_1+R_2+S = constant$.

The procedure consists in defining two matrices whose coefficients can be calculated taking into account the intraset and interset parameters.

Calculate the eigenvalues of $\mathbf{BC}^{-1}$ where

$$\mathbf{B} \quad \text{interset} \rightarrow b_{jk} = \frac{1}{M_1 M_2} \sum_{q=1}^{M_1} \sum_{p=1}^{M_2} \left(y_{q_j}^{(1)} - y_{p_j}^{(2)} \right) \cdot \left(y_{q_k}^{(1)} - y_{p_k}^{(2)} \right)$$

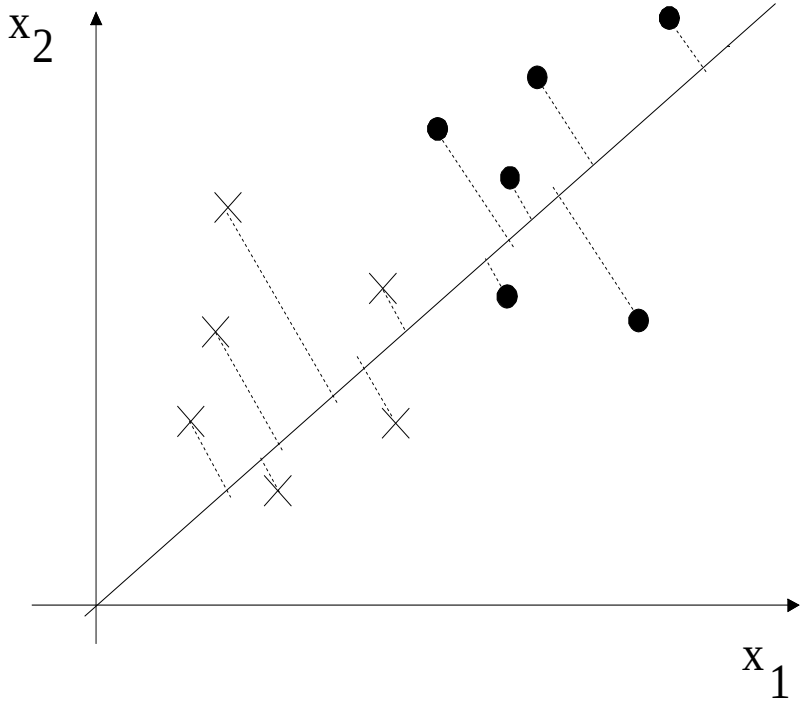
$$\mathbf{C} \quad \text{intraset} \rightarrow c_{jk} = \frac{2}{M(M-1)} \sum_{q=1}^{M} \sum_{p=1}^{M} \left(y_{q_j}^{(l)} - y_{p_j}^{(l)} \right) \cdot \left(y_{q_k}^{(l)} - y_{p_k}^{(l)} \right)$$

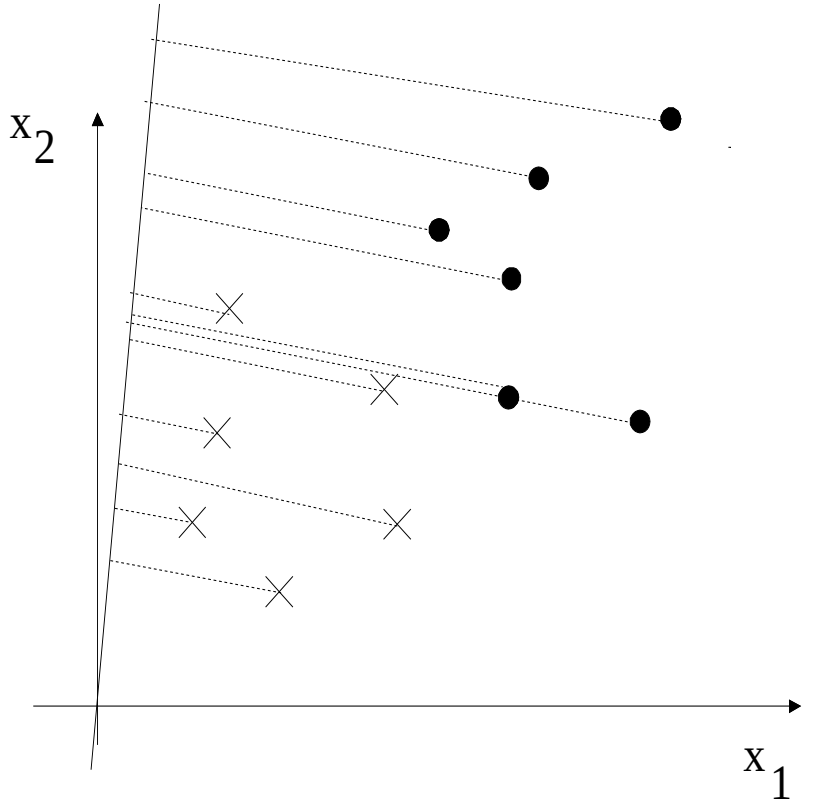
where $M = M_1 + M_2$ and $l = 1,2$

- We rank the eigenvalues: $\lambda_1 > \lambda_2 > \ldots$
- Calculate the corresponding eigenvectors $\mathbf{v}_1$, $\mathbf{v}_2$, ..., therefore: $\mathbf{w} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_n \end{bmatrix}$
- To reduce space, fewer eigenvectors can be considered.
- The eigenvector $\mathbf{v}_1$ is called the *Fisher direction* and allows to have the projection along an axis with the maximum interset distance and minimum intraset distance.
- You could also choose the objective function as $\max\{S\}$ with the constraint $R_1 + R_2 + S = constant$.
- In practice, however, the transformation involves calculations so demanding for significant constraints that it is not used.

The Fisher transform

- The problem is to reduce the dimensionality of the feature space in order to make the problem computationally manageable.
- It is essentially the projection of the *features* characterizing a sample on a straight line with a specific direction (from a *d-dimensional* problem to a *1-dimensional* problem).
- Obviously, if the classes were well separated in the d-dimensional space, typically they will not be in the *1*-dimensional case (because they will have overlapping elements), so the problem is to look for the orientation of the line for which the separation of the classes is better.





- We will still have a loss, but among the possible transformations the Fisher's one is the best.

- Suppose you have a set of N *d*-dimensional samples $\mathbf{x}_1$, .., $\mathbf{x}_N$, of which N_1 classified as ω_1 and N_2 classified as ω_2.
- We want to look for a transformation $\mathbf{w}$, that is, a linear combination of the $\mathbf{x}$ components such as to generate the corresponding samples (scalars) y_1,..., y_N:

$$\mathbf{w}^t \mathbf{x} = y$$

- Geometrically, if the norm of $\mathbf{w}$ is equal to 1 (i.e., a degree of freedom corresponding to a line with generic direction), then every y_i is the projection of the sample $\mathbf{x}_i$ on the line of direction $\mathbf{w}$.
- The important aspect is the direction of $\mathbf{w}$ and not the amplitude (as it includes only a scaling).

- Since we want to separate the two classes even in the new one-dimensional space, then the difference in the means of the samples is considered as a measure of separation. Therefore:

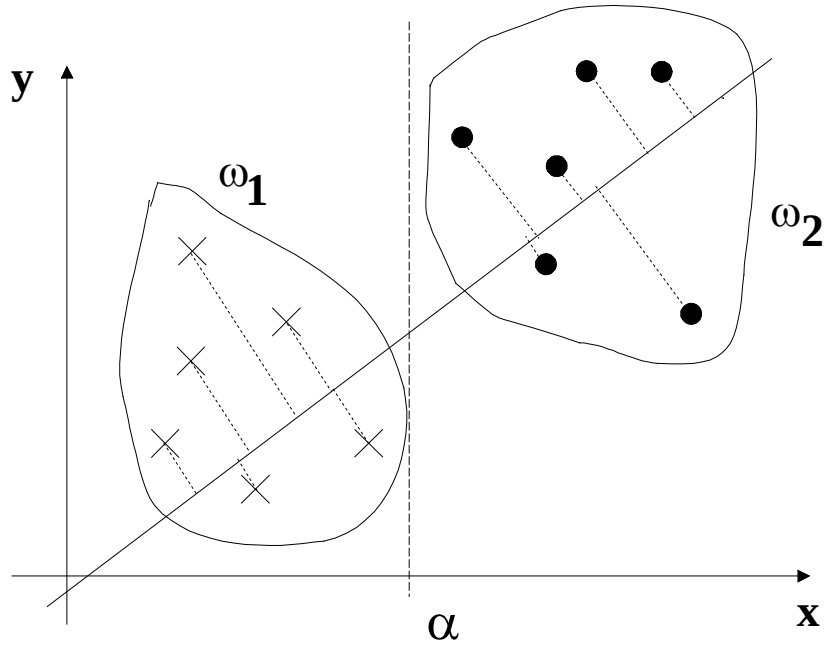
$$\tilde{m}_1 = \mathbf{w}^t \cdot \mathbf{m}_1$$

$$\tilde{m}_2 = \mathbf{w}^t \cdot \mathbf{m}_2$$

where

$$\left.\begin{aligned} \mathbf{m}_1 &= \frac{1}{N_1} \cdot \sum_{i=1}^{N_1} \mathbf{x}_i^{(1)} \\ \mathbf{m}_2 &= \frac{1}{N_2} \cdot \sum_{i=1}^{N_2} \mathbf{x}_i^{(2)} \end{aligned}\right\}$$ means before the transformation

$$\left.\begin{aligned} \tilde{m}_1 &= \frac{1}{N_1} \cdot \sum_{i=1}^{N_1} \mathrm{y}_i^{(1)} \\ \tilde{m}_2 &= \frac{1}{N_2} \cdot \sum_{i=1}^{N_2} \mathrm{y}_i^{(2)} \end{aligned}\right\}$$ means after the transformation



- We want to obtain that the difference between the means of the two (transformed) classes is large compared to the standard deviation of each class.
- Then, we define the Fisher linear discriminant as the linear function $\mathbf{w}^t\mathbf{x}$ for which the function J is maximum:

$$J(\mathbf{w}) = \frac{\left|\tilde{m}_1 - \tilde{m}_2\right|}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

where $\tilde{s}_1$ e $\tilde{s}_2$ are the *scatters* of the classified samples ω_1 and ω_2, respectively, defined as:

$$\tilde{s}_i^2 = \sum_{j=1}^{N_i} \left(y_j^{(i)} - \tilde{m}_i \right)^2$$

- We want that the dispersions are small enough, that is, that the samples of a class are quite concentrated around the mean value.
- To get J as an explicit function of $\mathbf{w}$, *scatter matrices* S_i and S_w are defined as follows:

$$S_i = \sum_{j=1}^{N_i} \left(\mathbf{x}_j^{(i)} - \mathbf{m}_i \right) \left(\mathbf{x}_j^{(i)} - \mathbf{m}_i \right)^t \qquad S_w = S_1 + S_2$$

- Analogously:

$$\tilde{s}_i^2 = \sum_{j=1}^{N_i} \left(y_j^{(i)} - \tilde{m}_i \right)^2 = \sum_{j=1}^{N_i} \left(\mathbf{w}^t \mathbf{x}_j^{(i)} - \mathbf{w}^t \mathbf{m}_i \right)^2 = \mathbf{w}^t S_i \mathbf{w}$$

- In this way:

$$\tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{w}^t S_w \mathbf{w} \qquad \left(\tilde{m}_1 - \tilde{m}_2 \right)^2 = \mathbf{w}^t S_B \mathbf{w}$$

where

$$S_B = (\mathbf{m}_1 - \mathbf{m}_2) \cdot (\mathbf{m}_1 - \mathbf{m}_2)^t$$

- So to get max $J(\mathbf{w})$, we express J as a direct function of $\mathbf{w}$ and then derive it wrt $\mathbf{w}$ and put equal to 0.

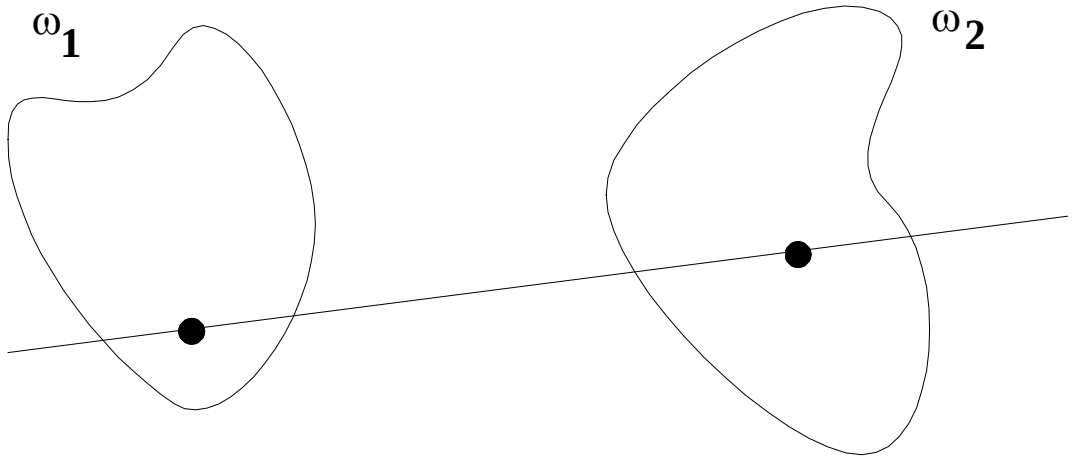
$$J(\mathbf{w}) = \frac{\left| \tilde{m}_1 - \tilde{m}_2 \right|}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{\mathbf{w}^t S_B \mathbf{w}}{\mathbf{w}^t S_w \mathbf{w}}$$

- Deriving we get:

$$\frac{\partial J}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = S_w^{-1}\left(\mathbf{m}_1 - \mathbf{m}_2\right)$$

  which is the Fisher transform.

- The demonstration starts with the assumption that $S_B\mathbf{w}$ it is always along the direction that connects the means of the two classes.



Curse of dimensionality

- In practical problems you can also find *training sets* of various kinds: few samples, few/many features, fixed features that cannot be selected, not-independent features

- There are 2 important problems for the design of a classifier
  - the computational complexity of the system,
  - the influence of the dimensionality (and cardinality) of the training set on the accuracy of the classification.

- If the features are statistically independent then it can be shown that optimal performance is achieved
- Example: pb. 2-class, normal, multivariate, equal covariance: $p(\mathbf{x}|\omega_j) = N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma})$
- It can be shown that, with equal *priors*, the probability of error is given by

$$P(e) = \frac{1}{\sqrt{2\pi}} \int_{r/2}^{\infty} e^{-\frac{u^2}{2}} du$$

where r^2 is the square of the Mahalanobis distance

$$r^2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t \, \boldsymbol{\Sigma}^{-1} \, (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

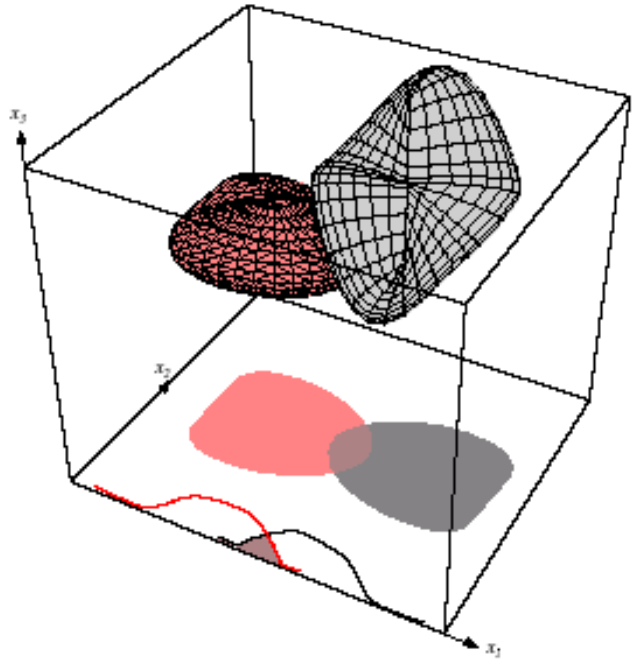
- Note that the probability of error decreases as *r* grows, tending to 0 when *r* tends to infinity

- In case $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2)$ (diagonal) then

$$r^2 = \sum_{i=1}^{d} \left(\frac{\mu_{i1} - \mu_{i2}}{\sigma_i} \right)^2$$

- Here, we see that each (independent) feature contributes to decreasing the probability of error, until it is arbitrarily small
- In general, if performance is inadequate, you can add features, but at the cost of complicating the classifier and the feature extractor
- However, even if the probabilistic structure is known, the Bayes risk may not vary even by adding features

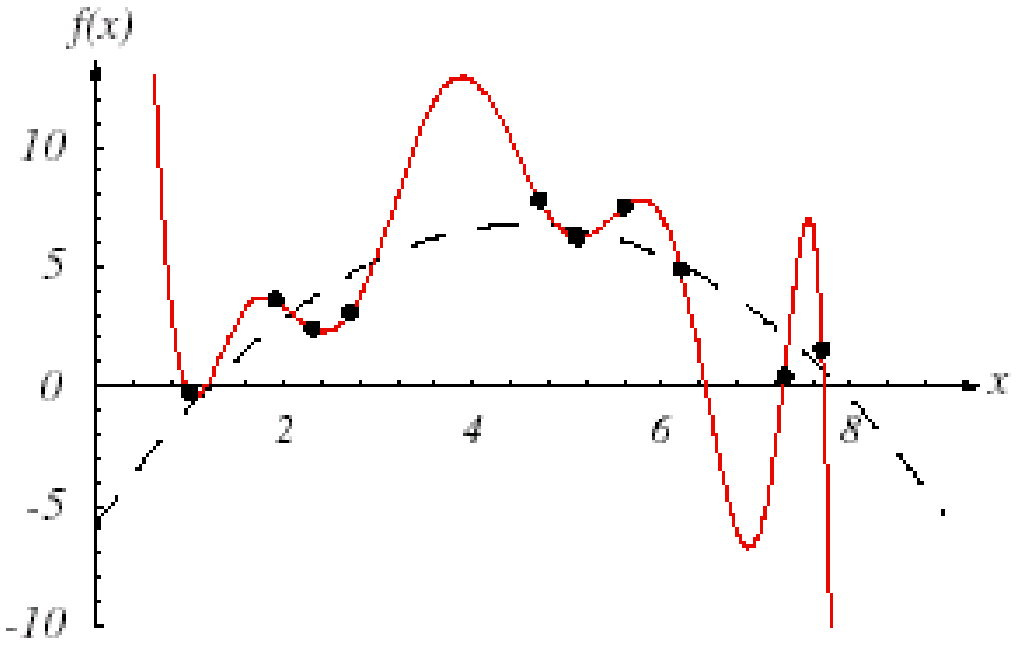
- In practice, adding features leads to worse performance. This is due to:
  - incorrect model (e.g., Gaussian hypothesis or conditional probability hypothesis);
  - insufficient number of samples, and therefore the distributions are not accurately estimated.



Overfitting

- When the number of samples is insufficient then:
  - try to reduce the dimensionality of features
  - combine features in some way (***feature extraction***)
  - look for a better estimate of the covariance matrix starting from a known estimate
  - threshold the covariance matrix or impose a diagonal matrix
- Doing this, however, we just imposes the independence of the features, but in reality they may NOT be independent!
- So, the performance may be only sub-optimal, and the motivation can still be attributed to the insufficient data

- The problem is similar to that of the data fitting (interpolation vs approximation)
- If you interpolate correctly you lose in generalization capacity (the parabola is the function that best approximates the data)



Main Component Method (PCA)
(Hotelling or Karhunen-Loève transform)

- Linear transformation originally introduced by Hotelling to decorrelate the elements of a random vector
- Karhunen & Loève later developed a similar transformation for continuous signals
- It is also called the method of the *principal components* or of the *eigenvectors*.
- Given a population of vectors of random variables $\mathbf{x}_i$, the base vectors of the KL transform are given by the orthonormalized eigenvectors of their covariance matrix (autocorrelation) $\mathbf{C}$

- Given a population of vectors of random variables (n x 1) of the type

$$\mathbf{x} = \{\mathbf{x}_1^{\mathrm{t}}, \mathbf{x}_2^{\mathrm{t}}, \ldots, \mathbf{x}_{\mathrm{M}}^{\mathrm{t}}\}$$

  The average is defined as

$$\mathbf{m}_{\mathbf{x}} = E\{\mathbf{x}\}$$

  where $E\{.\}$ is the Expected Value operator

- The covariance matrix of this population is defined as

$$\mathbf{C}_{\mathbf{x}} = E\{(\mathbf{x}\text{-}\mathbf{m}_{\mathbf{x}})\,(\mathbf{x}\text{-}\mathbf{m}_{\mathbf{x}})^{\mathrm{t}}\}$$

- Characteristics:
  - Since $\mathbf{x}$ is of dimensionality n, then $\mathbf{C}_{\mathbf{x}}$ is $n \times n$;
  - Elements c_{ii} are the variance of the i-th component of the set of $\mathbf{x}$;
  - c_{ij} are the related covariances between components;
  - $\mathbf{C}_{\mathbf{x}}$ is real and symmetrical; if x_i and x_j are unrelated, then $c_{ij} = c_{ji} = 0$.

- If the population is finite of size M, then

$$\mathbf{m}_{\mathbf{x}} = \frac{1}{M}\sum_{k=1}^{M}\mathbf{x}_k \qquad \mathbf{C}_{\mathbf{x}} = \frac{1}{M}\sum_{k=1}^{M}(\mathbf{x}_k - \mathbf{m}_{\mathbf{x}})(\mathbf{x}_k - \mathbf{m}_{\mathbf{x}})^t = \frac{1}{M}\sum_{k=1}^{M}\mathbf{x}_k\mathbf{x}_k^t - \mathbf{m}_{\mathbf{x}}\mathbf{m}_{\mathbf{x}}^t$$

- Since $\mathbf{C_x}$ is real and symmetrical, you can always find a set of n orthonormal eigenvectors
- Let $\mathbf{e_i}$ and λ_i, $i=1,2,\ldots,n$, be the eigenvectors and the related eigenvalues of $\mathbf{C_x}$, respectively, ordered in descending order, ie.,

  $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$
- Let $\mathbf{A}$ be a matrix whose rows are formed by the eigenvectors of $\mathbf{C_x}$ ordered as above, and use it to transform vectors $\mathbf{x}$ as follows

$$\mathbf{y} = \mathbf{A}\,(\mathbf{x} - \mathbf{m_x})$$

- The average of the vectors $\mathbf{y}$ is zero and the covariance matrix:

$$\mathbf{m_y} = \mathbf{0} \qquad\qquad \mathbf{C_y} = \mathbf{A}\,\mathbf{C_x}\,\mathbf{A}^{\mathrm{T}}$$

- In addition, $\mathbf{C_y}$ is a diagonal matrix with the eigenvalues of $\mathbf{C_x}$ in the diagonal.

$$\mathbf{C_y} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix}$$

- The elements of $\mathbf{y}$ are therefore decorrelated
- Since the lines of $\mathbf{A}$ are orthonormal vectors, then $\mathbf{A}^{-1} = \mathbf{A}^{\mathrm{T}}$ and each vector $\mathbf{x}$ can be calculated by $\mathbf{y}$:

$$\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{y} + \mathbf{m_x}$$

- Suppose you build $\mathbf{A}$ with the first k (larger) eigenvectors, $k < n$.

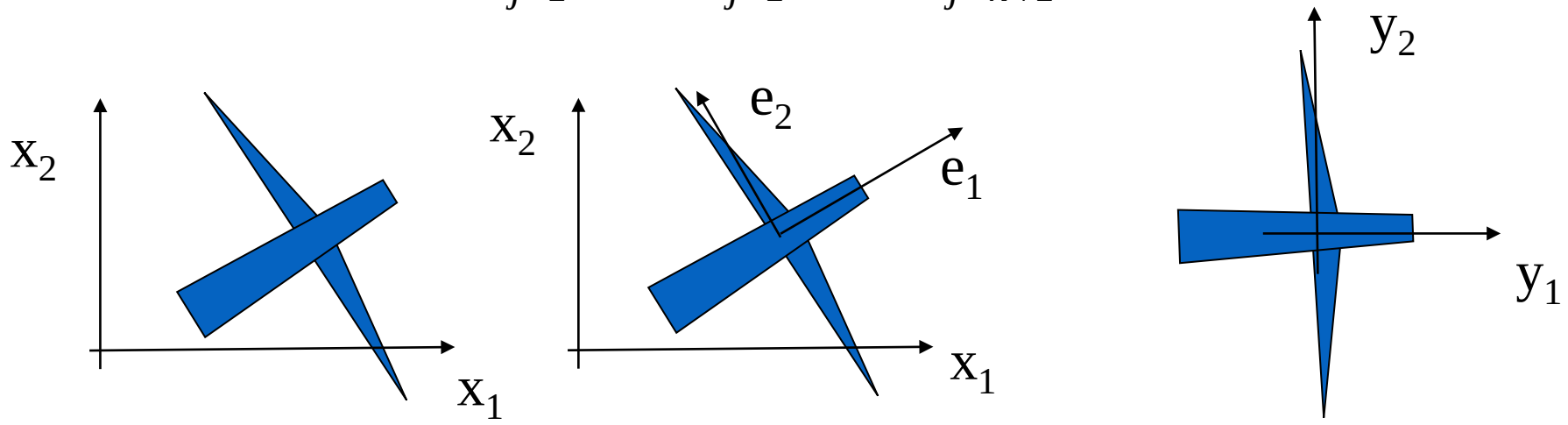
- Then $\mathbf{A}_k$ is a matrix of dimension $k \times n$ and vectors $\mathbf{y}$ have dimension k.
- Reconstructed vectors $\mathbf{x}$ will no longer be accurate and are given by:

$$\hat{\mathbf{x}} = \mathbf{A}_k^T \mathbf{y} + \mathbf{m}_\mathbf{x}$$

- It is shown that the Hotelling transform minimizes the mean quadratic error of the vectors $\mathbf{x}$

$$e_{ms} = \sum_{j=1}^{n} \lambda_j - \sum_{j=1}^{k} \lambda_j = \sum_{j=k+1}^{n} \lambda_j$$

- Example



Example: *appearance-based* recognition

- Recognition is based on sight or appearance.
- You use img as the basic components of models instead of features.
- Each object is represented by a set of views, theoretically taken from all possible points of view in all lighting conditions
- The identification of the object means finding the set that contains the img most similar to the obj to be recognized.
- It allows you to directly compare models with input data.
- The model database can get too big!

Example: *appearance-based* recognition

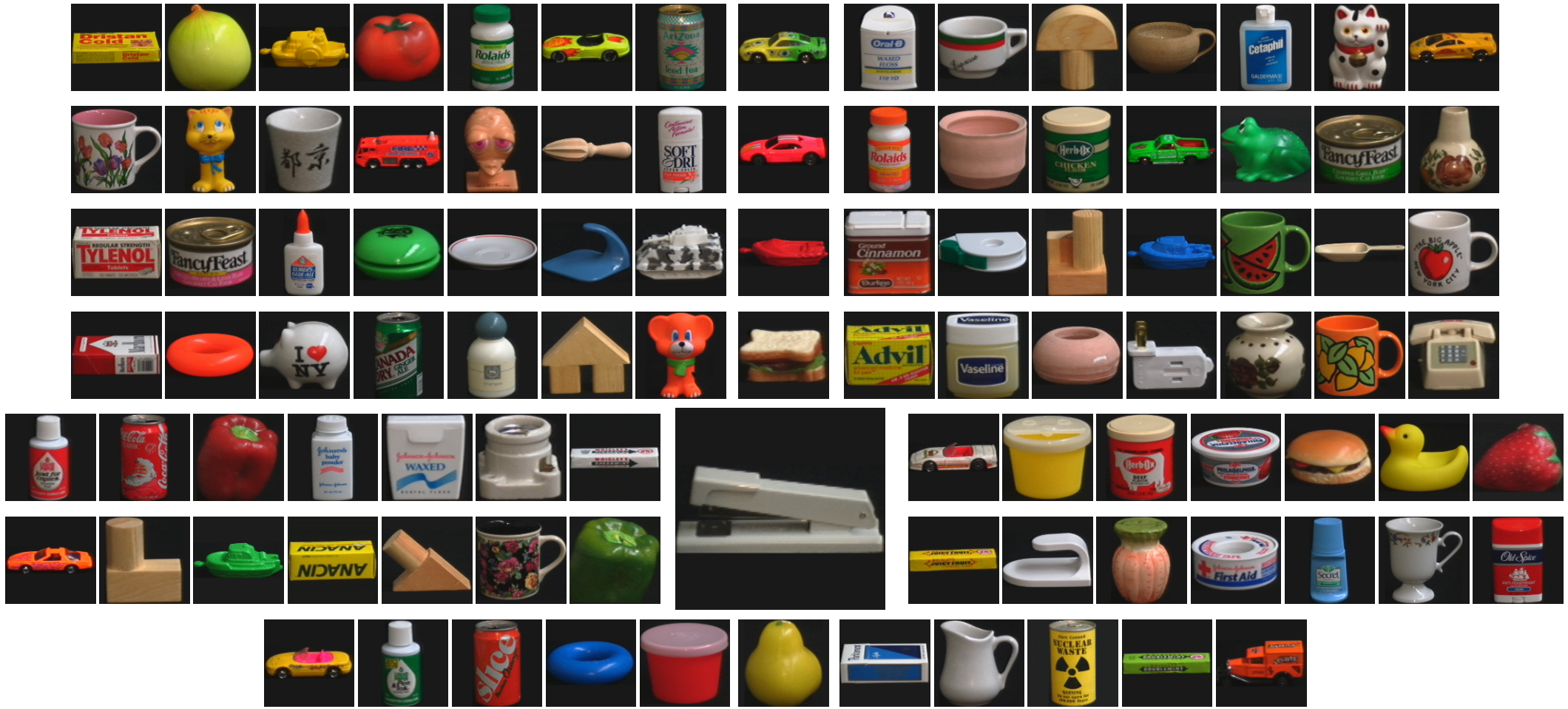


Image eigenspace

- View-based method
- Hypothesis:
  - each img contains only one obj
  - objs are captured by a fixed camera under weak perspective conditions
  - img normalized in size, i.e., the size of the img is the smallest rectangle that contains the largest view of the obj
  - the energy of each img is normalized to 1: $\sum_{i=1}^{N}\sum_{j=1}^{N} I(i,j)^2 = 1$
  - the obj is completely visible (not occluded)

- The comparison between img is performed by means of the *correlation* operation:

$$c = I_1 \otimes I_2 = \frac{1}{K}\sum_{i=1}^{N}\sum_{j=1}^{N} I_1(i,j) I_2(i,j), \quad K \text{ costante di normalizzazione}$$

- Algorithm
  - Given O obj, P points of view, L lighting directions, you have OPL img in the database
  - We calculate the covariance matrix Q, and represent every img $\mathbf{x}_{pl}{}^{o}$ with its eigenspace vector of coordinates $\mathbf{g}_{pl}{}^{o}$.
  - Only the components associated with the largest eigenvalues can be used to represent the images:

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n, \quad \lambda_i \cong 0 \quad \text{per} \quad i > k$$

$$\mathbf{x}_j \cong \mathbf{x}_m + \sum_{i=1}^{k} g_{ji}\mathbf{e}_i$$

    with $\mathbf{e}_i$ eigenvectors of matrix Q and g_{ij} are the coefficients associated with the img $\mathbf{g}$
  - If $k << n$ the representation is greatly reduced.
  - The set of views of an obj (by varying pose and direction of illumination) make the point $\mathbf{g}_{pl}{}^{o}$ in thw eigenspace to vary in a continuous way, so creating a *manifold*.

- The correlation between the images is carried out by calculating the distance between the images in the eigenspace.
- In the hypotheses of normalization of the images $\|\mathbf{x}_1\|^2 = \|\mathbf{x}_2\|^2 = 1$, we have $\|\mathbf{x}_1 - \mathbf{x}_2\|^2 = 2(1 - \|\mathbf{x}_1{}^T\mathbf{x}_2\|)$, i.e., maximizing the correlation means minimizing the distance:

$$\|\mathbf{x}_1 - \mathbf{x}_2\|^2 = \left\|\sum_{i=1}^{n} g_{1i}\mathbf{e}_i - \sum_{i=1}^{n} g_{2i}\mathbf{e}_i\right\|^2 \cong \left\|\sum_{i=1}^{k} g_{1i}\mathbf{e}_i - \sum_{i=1}^{k} g_{2i}\mathbf{e}_i\right\|^2 =$$

$$= \left\|\sum_{i=1}^{k} (g_{1i} - g_{2i})\mathbf{e}_i\right\|^2 = \sum_{i=1}^{k} (g_{1i} - g_{2i})^2 = \|\mathbf{g}_1 - \mathbf{g}_2\|^2$$

- The computational cost is $O(k)$ instead of $O(n)$.
- This procedure suggests how to identify an object:
  - given the images of a model, the points in the eigenspace and its interpolating curve can be calculated

- to identify an obj from a new img $\mathbf{y}$, $\mathbf{y}$ is projected in eigenspace (using the eigenvectors of the covariance matrix of all img OPL in the database), obtaining a point $\mathbf{g}_y$;
- you have to look for the curve $\mathbf{g}^o(\mathbf{p},\mathbf{l})$ ($= \mathbf{g}_{pl}{}^o$) closest to $\mathbf{g}_y$.

- How to estimate $\mathbf{g}_{pl}{}^o$:

$$\mathbf{g}_{pl}^{o} = \left[\mathbf{e}_1\ \mathbf{e}_2 \cdots \mathbf{e}_k\right]\left(\mathbf{x}_{pl}^{o} - \mathbf{x}_m\right)$$

- How to estimate the projection of $\mathbf{y}$ in eigenspace,$\mathbf{g}_y$:

$$\mathbf{g}_{y} = \left[\mathbf{e}_1\ \mathbf{e}_2 \cdots \mathbf{e}_k\right]\left(\mathbf{y} - \mathbf{x}_m\right)$$

- Considerations and comments:
  - finding the point of a curve closer to a given point is not always trivial;
  - it is not always true that $k << n$;
  - finding the eigenvalues of large matrices is computationally expensive;
  - segmentation between obj and background is not always a simple operation.