

# CC-226

# Introdução à Análise de Padrões


Prof. Carlos Henrique Q. Forster

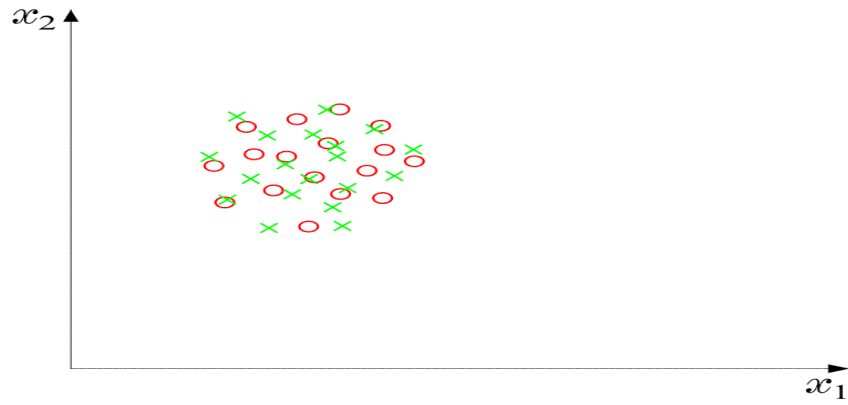
Seleção de feições



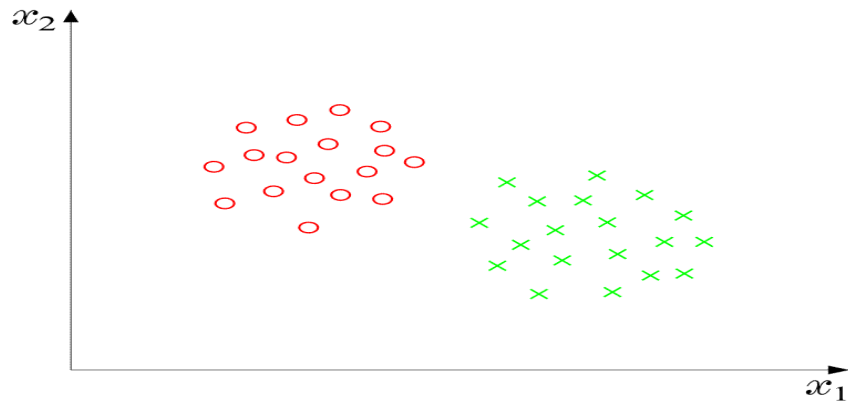
# FEATURE SELECTION

- The goals:
  - Select the “optimum” number  $l$  of features
  - Select the “best”  $l$  features
  
- Large  $l$  has a three-fold disadvantage:
  - High computational demands
  - Low generalization performance
  - Poor error estimates

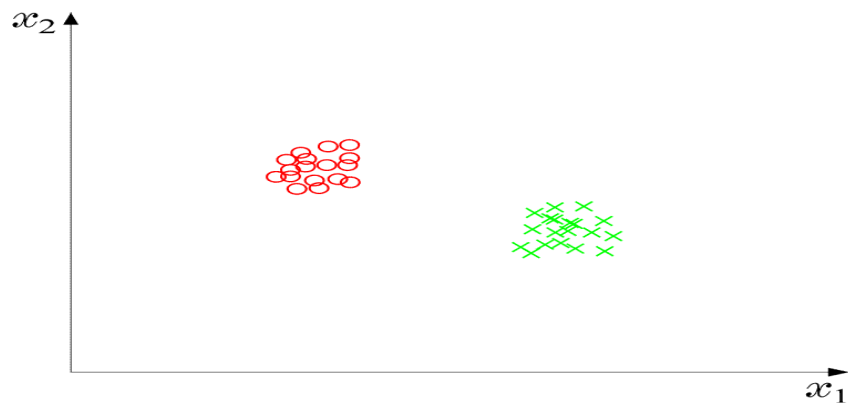
- 
- Given  $N$ 
    - $l$  must be **large enough** to learn
      - what makes classes **different**
      - what makes patterns in the same class **similar**
    - $l$  must be **small enough** not to learn what makes patterns of the same class **different**
    - In practice,  $l < N/3$  has been reported to be a sensible choice for a number of cases
  
  - Once  $l$  has been decided, choose the  $l$  most informative features
    - Best: **Large** between class distance,  
**Small** within class variance




**Bad choice**



**Not bad choice**



**Good choice**

- 
- The basic philosophy
    - Discard individual features with **poor** information content
    - The remaining information rich features are examined **jointly** as vectors





# Class Separability Measures

Considering features individually cannot take into account existing correlations among the features. That is, two features may be rich in information, but if they are highly **correlated** we need not consider both of them. To this end, in order to search for possible correlations, we consider features **jointly** as elements of **vectors**. To this end:

- Discard poor in information features, by means of a statistical test.
- Choose the maximum number,  $l$ , of features to be used. This is dictated by the specific problem (e.g., the number,  $N$ , of available training patterns and the type of the classifier to be adopted).

- 
- Combine remaining features to search for the “best” combination. To this end:

- Use different feature combinations to form the feature vector. Train the classifier, and choose the combination resulting in the best classifier performance.

A major disadvantage of this approach is the high complexity. Also, local minima, may give misleading results.

- Adopt a class separability measure and choose the best feature combination against this cost.

# Divergence

To see the rationale behind this cost, consider the two – class case. Obviously, if on the average the

value of  $\ln \frac{p(\underline{x} | \omega_1)}{p(\underline{x} | \omega_2)}$  is close to zero, then should be a

poor feature combination. Define:

$$\blacksquare D_{12} = \int_{-\infty}^{+\infty} p(\underline{x} | \omega_1) \ln \frac{p(\underline{x} | \omega_1)}{p(\underline{x} | \omega_2)} d\underline{x}$$

$$\blacksquare D_{21} = \int_{-\infty}^{+\infty} p(\underline{x} | \omega_2) \ln \frac{p(\underline{x} | \omega_2)}{p(\underline{x} | \omega_1)} d\underline{x}$$

$$d_{12} = D_{12} + D_{21}$$

$d_{12}$  is known as the **divergence** and can be used as a class separability measure.



- 
- For the multi-class case, define  $d_{ij}$  for every pair of classes  $\omega_i, \omega_j$  and the **average divergence** is defined as

$$d = \sum_{i=1}^M \sum_{j=1}^M P(\omega_i)P(\omega_j)d_{ij}$$

- Some properties:

$$d_{ij} \geq 0$$

$$d_{ij} = 0, \text{ if } i = j$$

$$d_{ij} = d_{ji}$$

- Large values of  $d$  are indicative of good feature combination.

# Matriz de espalhamento (scatter matrices) – Espalhamento intraclasses

- These are used as a measure of the way data are scattered in the respective feature space.

- **Within-class** scatter matrix 
$$S_w = \sum_{i=1}^M P_i S_i$$

where 
$$S_i = E \left[ (\underline{x} - \underline{\mu}_i)(\underline{x} - \underline{\mu}_i)^T \right]$$

and 
$$P_i \equiv P(\omega_i) \approx \frac{n_i}{N}$$

$n_i$  the number of training samples in  $\omega_i$ .

Trace  $\{S_w\}$  is a measure of the **average variance** of the features.

# Espalhamento inter-classe

$$S_b = \sum_{i=1}^M P_i (\underline{\mu}_i - \underline{\mu}_0)(\underline{\mu}_i - \underline{\mu}_0)^T$$

$$\underline{\mu}_0 = \sum_{i=1}^M P_i \underline{\mu}_i$$

Trace  $\{S_b\}$  is a measure of the average distance of the mean of each class from the respective global one.

# Espalhamento Misto

- Mixture scatter matrix

$$S_m = E[(\underline{x} - \underline{\mu}_0)(\underline{x} - \underline{\mu}_0)^T]$$

It turns out that:

$$S_m = S_w + S_b$$

# Medidas sobre matrizes de espalhamento

$$J_1 = \frac{\text{Trace}\{S_m\}}{\text{Trace}\{S_w\}} \quad J_2 = \frac{|S_m|}{|S_w|} = |S_w^{-1} S_m|$$

□

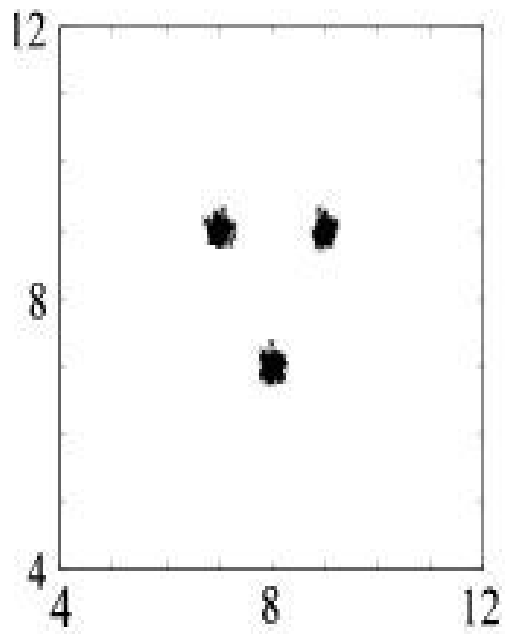
$$J_3 = \text{Trace}\{S_w^{-1} S_m\}$$

□

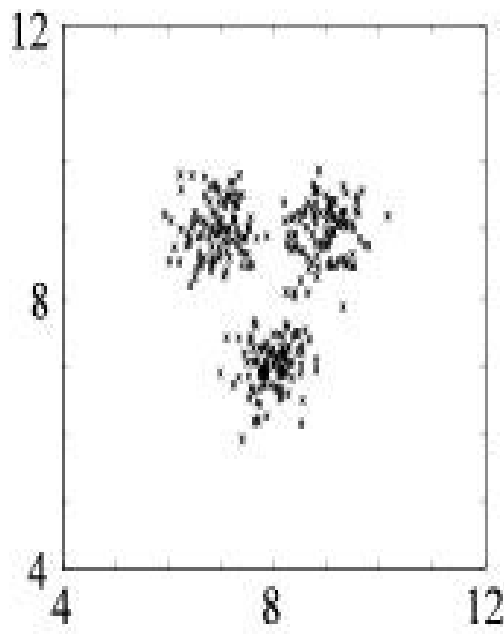
- Other criteria are also possible, by using various combinations of  $S_m$ ,  $S_b$ ,  $S_w$ .

The above  $J_1$ ,  $J_2$ ,  $J_3$  criteria take high values for the cases where:

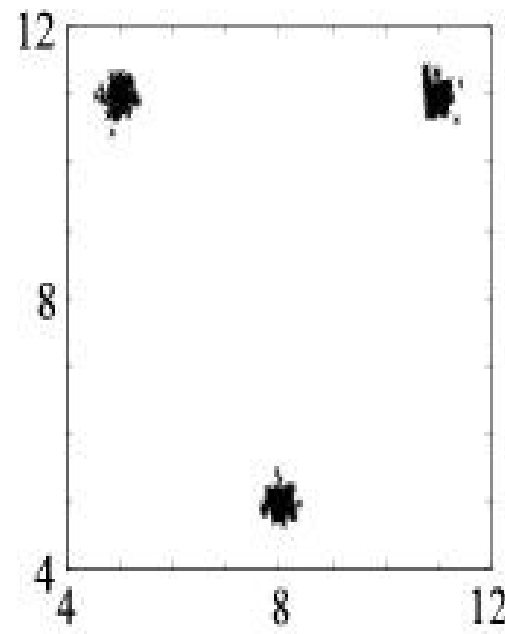
- Data are clustered together within each class.
- The means of the various classes are far.



(a)



(b)



(c)



# Razão discriminante de Fisher

- In **one** dimension and for **two** equiprobable classes the determinants become:

$$|S_w| \propto \sigma_1^2 + \sigma_2^2$$

$$|S_b| \propto (\mu_1 - \mu_2)^2$$

and

$$\frac{|S_b|}{|S_w|} = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

known as **Fischer's ratio**.

# Ways to combine features

Trying to form all possible combinations of features from an original set of  $m$  selected features is a computationally hard task. Thus, a number of **suboptimal** searching techniques have been derived.

- **Sequential forward selection.** Let  $x_1, x_2, x_3, x_4$  the available features ( $m=4$ ). The procedure consists of the following steps:
  - Adopt a class separability criterion (could also be the error rate of the respective classifier). Compute its value for **ALL** features considered **jointly**  $[x_1, x_2, x_3, x_4]^T$ .
  - Eliminate one feature and for each of the possible resulting combinations, that is  $[x_1, x_2, x_3]^T$ ,  $[x_1, x_2, x_4]^T$ ,  $[x_1, x_3, x_4]^T$ ,  $[x_2, x_3, x_4]^T$ , compute the class separability criterion value  $C$ . Select the best combination, say  $[x_1, x_2, x_3]^T$ .

- From the above selected feature vector eliminate one feature and for each of the resulting combinations,  $[x_1, x_2]^T$ ,  $[x_2, x_3]^T$ ,  $[x_1, x_3]^T$  compute **C** and select the best combination.

The above selection procedure shows how one can start from  $m$  features and end up with the “best” ones. Obviously, the choice is **suboptimal**. The number of required calculations is:

$$1 + \frac{1}{2}((m+1)m - \ell(\ell+1))$$

In contrast, a full search requires:

$$\binom{m}{\ell} = \frac{m!}{\ell!(m-\ell)!}$$

operations.

# Sequential backward selection

- Here the reverse procedure is followed.
  - Compute  $C$  for each feature. Select the “best” one, say  $x_1$
  - For all possible 2D combinations of  $x_1$ , i.e.,  $[x_1, x_2]$ ,  $[x_1, x_3]$ ,  $[x_1, x_4]$  compute  $C$  and choose the best, say  $[x_1, x_3]$ .
  - For all possible 3D combinations of  $[x_1, x_3]$ , e.g.,  $[x_1, x_3, x_2]$ , etc., compute  $C$  and choose the best one.

The above procedure is repeated till the “best” vector with features has been formed. This is also a **suboptimal** technique, requiring:

operations. 
$$lm - \frac{l(l-1)}{2}$$




# Floating Search Methods

The above two procedures suffer from the **nesting effect**. Once a bad choice has been done, there is no way to reconsider it in the following steps.

In the floating search methods one is given the opportunity in **reconsidering a previously discarded feature** or to **discard a feature that was previously chosen**.

The method is still **suboptimal**, however it leads to **improved performance**, at the expense of complexity.

- 
- Besides suboptimal techniques, some optimal searching techniques can also be used, provided that the optimizing cost has certain properties, e.g., monotonic.
  - Instead of using a class separability measure (filter techniques) or using directly the classifier (wrapper techniques), one can modify the cost function of the classifier appropriately, so that to perform feature selection and classifier design in a single step (embedded) method.
  - For the choice of the separability measure a multiplicity of costs have been proposed, including information theoretic costs.



# 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.

- The basic steps in the proof:
- $J_3 = \text{trace}\{S_w^{-1} S_m\}$ 
  - $S_{yw} = A^T S_{xw} A, S_{yb} = A^T S_{xb} A,$
  - $J_3(A) = \text{trace}\{(A^T S_{xw} A)^{-1} (A^T S_{xb} A)\}$
  - Compute  $A$  so that  $J_3(A)$  is maximum.

- The solution:

- Let  $B$  be the matrix that diagonalizes **simultaneously** matrices  $S_{yw}, S_{yb}$ , i.e:

$$B^T S_{yw} B = I, B^T S_{yb} B = D$$

where  $B$  is a  $l \times l$  matrix and  $D$  a  $l \times l$  diagonal matrix.

- Let  $C=AB$  an  $m \times \ell$  matrix. If  $A$  maximizes  $J3(A)$  then

$$\left( S_{xw}^{-1} S_{xb} \right) C = CD$$

- The above is an eigenvalue-eigenvector problem. For an  $M$ -class problem,  $S_{xw}^{-1} S_{xb}$  is of rank  $M-1$ .
  - If  $\ell=M-1$ , choose  $C$  to consist of the  $M-1$  eigenvectors, corresponding to the non-zero eigenvalues.

- The above guarantees maximum  $J3$  value. In this case:  $J3,x = J3,y$ .
- For a two-class problem, this results to the well known Fisher's linear discriminant

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

- For Gaussian classes, this is the optimal Bayesian classifier, with a difference of a threshold value .

$$\underline{y} = \left( \underline{\mu}_1 - \underline{\mu}_2 \right) S_{xw}^{-1} \underline{x}$$

- If  $\ell < M-1$ , choose the  $\ell$  eigenvectors corresponding to the  $\ell$  largest eigenvalues.
- In this case,  $J_{3,y} < J_{3,x}$ , that is there is loss of information.

- Geometric interpretation. The vector  $\frac{y}{\|y\|}$  is the projection of  $\frac{x}{\|x\|}$  onto the subspace spanned by the eigenvectors of  $S_{xw}$ .

$$S_{xw}^{-1} S_{xb}$$

```

❑ loadiris
❑ data=iris(:,1:4);
❑ m1=mean(data(1:50,:));
❑ m2=mean(data(51:100,:));
❑ m3=mean(data(101:150,:));

❑ m=(m1+m2+m3)/3;

❑ sb=(m1-m)*(m1-m)+(m2-m)*(m2-m)+(m3-m)*(m3-m);

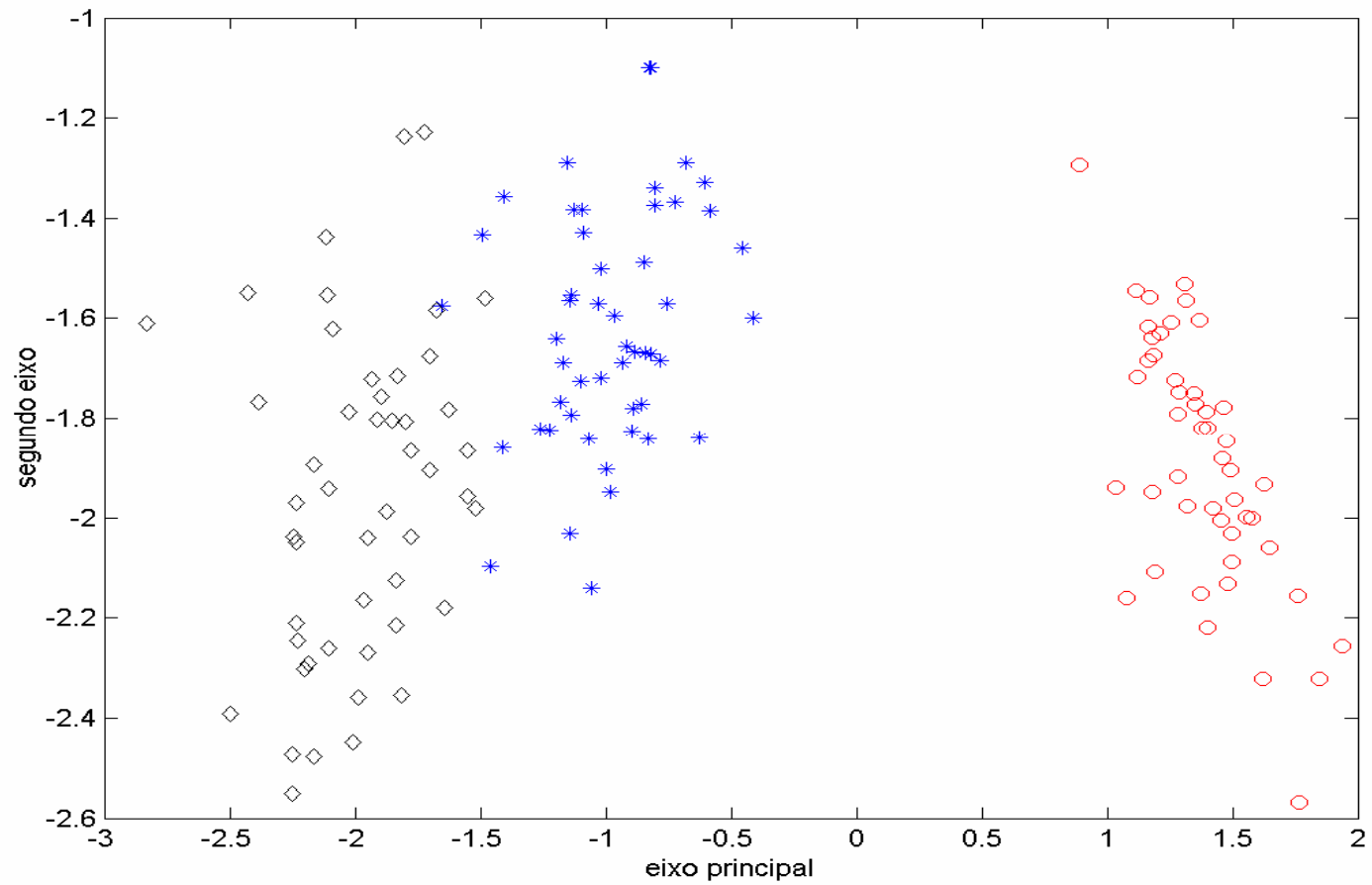
❑ s1=zeros(4,4);
❑ s2=s1;
❑ s3=s1;
❑ for i=1:50
❑     s1=s1+(data(i,:)-m1)*(data(i,:)-m1);
❑ end
❑ for i=51:100
❑     s2=s2+(data(i,:)-m2)*(data(i,:)-m2);
❑ end
❑ for i=101:150
❑     s3=s3+(data(i,:)-m3)*(data(i,:)-m3);
❑ end

❑ sw=s1+s2+s3;

❑ [v,d]=eig(inv(sw)*sb)
❑ w=[v(:,1), v(:,2)]
❑ k=w'*data';
❑ plot(k(1,1:50),k(2,1:50),'ro',k(1,51:100),k(2,51:100),'b*',k(1,101:150),k(2,101:150),'kd');
❑ xlabel('eixo principal');
❑ ylabel('segundo eixo');

```

# Resultado LDA - Iris





# Resultado LDA - Iris

□  $v =$

□	0.2049	-0.0090	0.3398	-0.6672
□	0.3871	-0.5890	0.1988	0.4427
□	-0.5465	0.2543	0.2728	0.4688
□	-0.7138	-0.7670	-0.8779	-0.3729

□  $d =$

□	0.6454	0	0	0
□	0	0.0056	0	0
□	0	0	0.0000	0
□	0	0	0	-0.0000

□  $w =$

□	0.2049	-0.0090	sepal length
□	0.3871	-0.5890	sepal width
□	-0.5465	0.2543	petal length
□	-0.7138	-0.7670	petal width
□	1º eixo	2º eixo	

# Principal Components Analysis

(The Karhunen – Loève transform):

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

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

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

That is

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

- 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.
  - Due to the orthogonality of  $A$ :

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

- Define

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

- The Karhunen – Loève transform minimizes the square error:

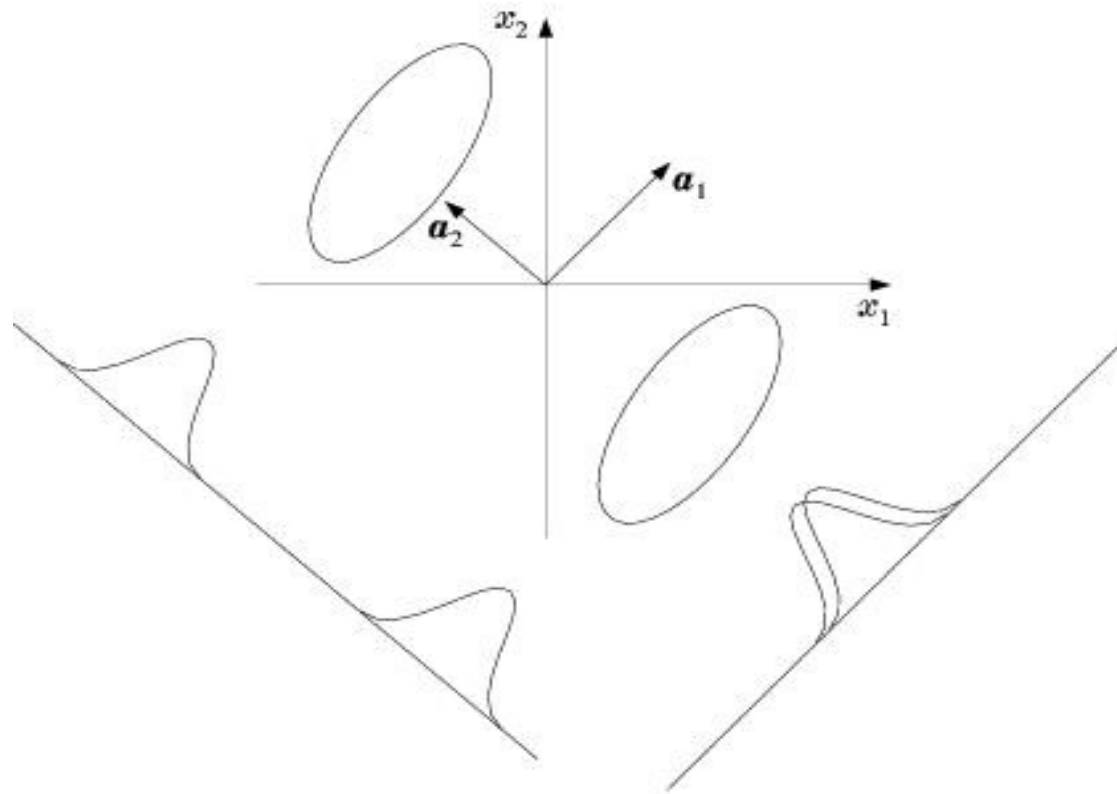
$$E \left[ \left\| \underline{x} - \underline{\hat{x}} \right\|^2 \right] = E \left[ \left\| \sum_{i=\ell}^m y(i) \underline{a}_i \right\|^2 \right]$$

- The error is:

$$E \left[ \left\| \underline{x} - \underline{\hat{x}} \right\|^2 \right] = \sum_{i=\ell}^m \lambda_i$$

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

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



- Total variance: It is easily seen that

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

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

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

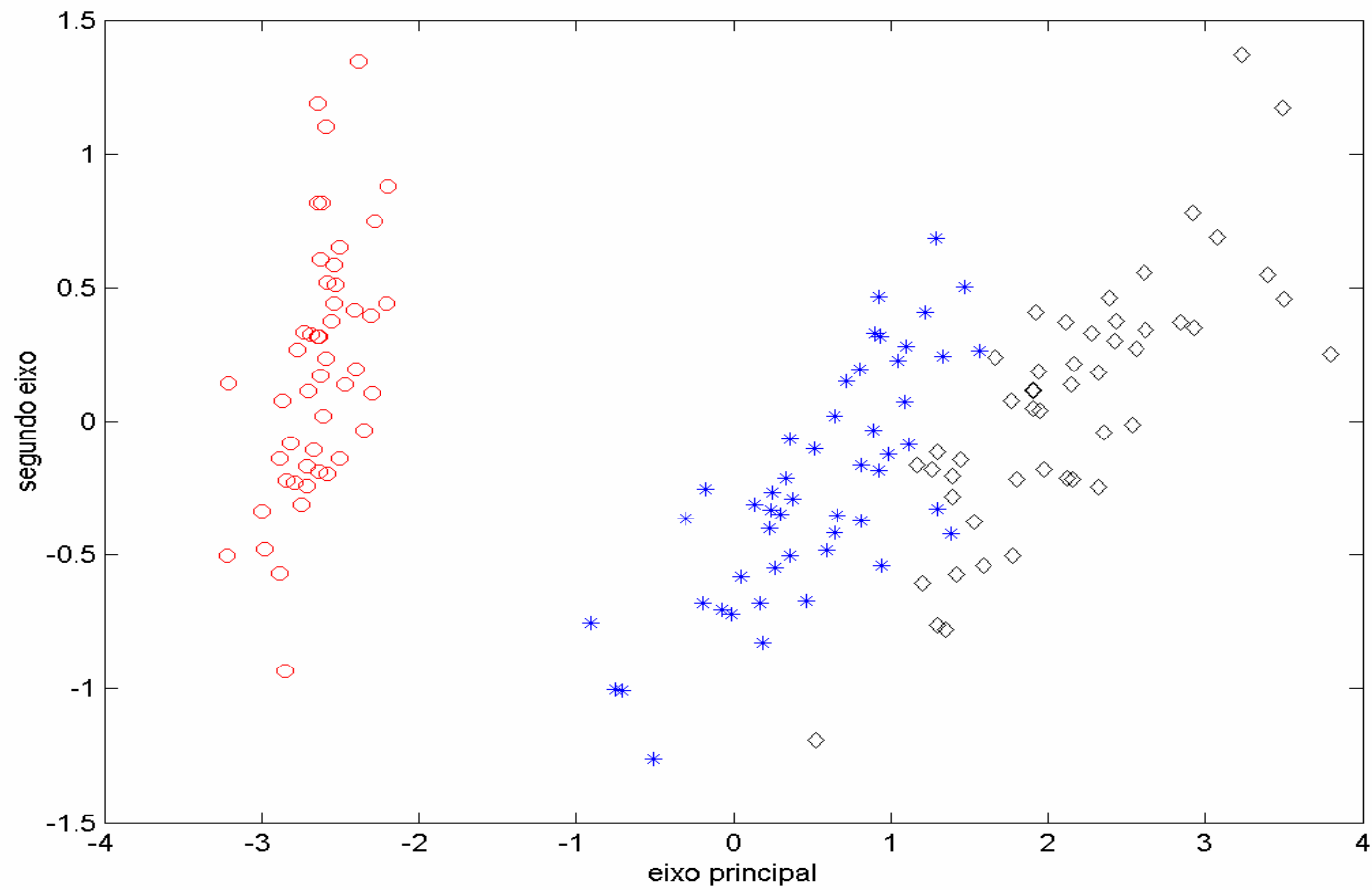
$$H_y = -E[\ln P_y(\underline{y})]$$

- of the resulting  $\underline{y}$  process.

# PCA

- ❑ loadiris
- ❑ `data=iris(:,1:4)-repmat(mean(iris(:,1:4)),size(iris,1),1)`
- ❑ `[v,d]=eig(data'*data)`
- ❑ `w=[v(:,4), v(:,3)]`
- ❑ `k=w'*data';`
- ❑ `plot(k(1,1:50),k(2,1:50),'ro',k(1,51:100),k(2,51:100),'b*',k(1,101:150),k(2,101:150),'kd');`
- ❑ `xlabel('eixo principal');`
- ❑ `ylabel('segundo eixo');`

# Resultado PCA – Iris





# Resultados PCA – Iris

□ v =

□	-0.3173	0.5810	0.6565	0.3616
□	0.3241	-0.5964	0.7297	-0.0823
□	0.4797	-0.0725	-0.1758	0.8566
□	-0.7511	-0.5491	-0.0747	0.3588

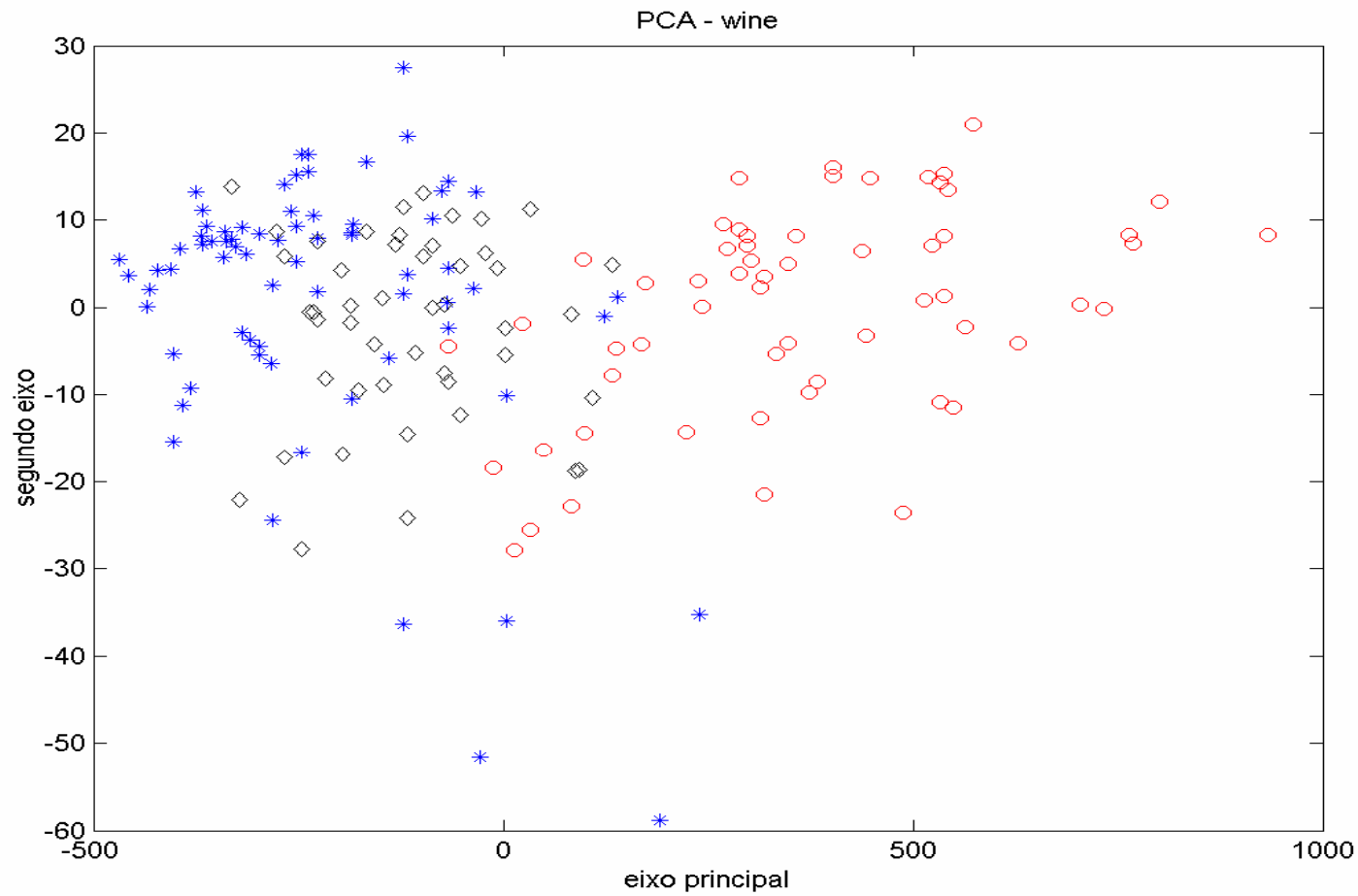
□ d =

□	3.5288	0	0	0
□	0	11.7001	0	0
□	0	0	36.0943	0
□	0	0	0	629.5013

□ w =

□	0.3616	0.6565	sepal length
□	-0.0823	0.7297	sepal width
□	0.8566	-0.1758	petal length
□	0.3588	-0.0747	petal width
□	1º eixo	2º eixo	

# Resultados PCA – Wine



# Resultados PCA – Wine

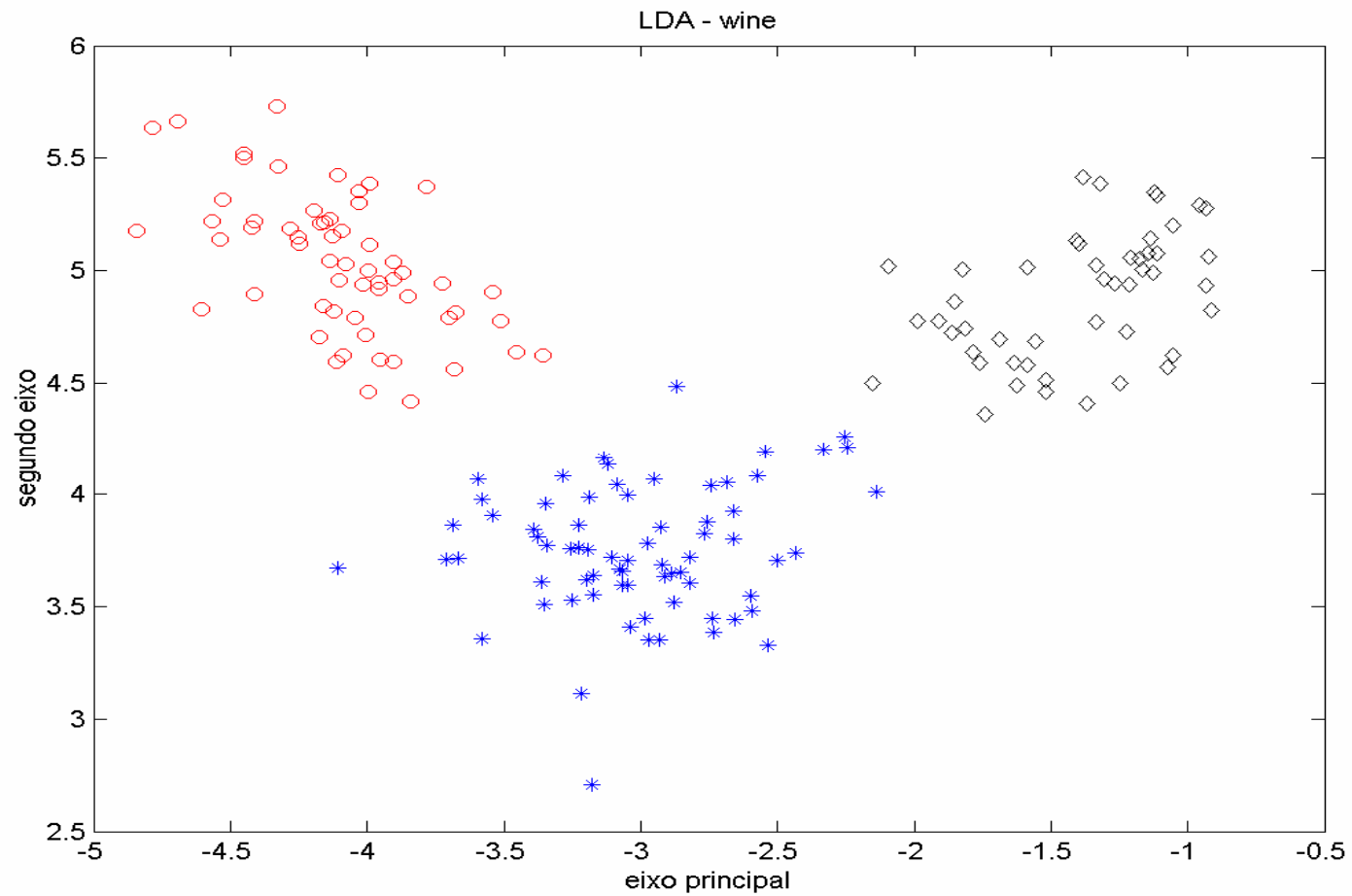
□ w =

- 0.0017 -0.0012
- -0.0007 -0.0022
- 0.0002 -0.0046
- -0.0047 -0.0265
- 0.0179 -0.9993
- 0.0010 -0.0009
- 0.0016 0.0001
- -0.0001 0.0014
- 0.0006 -0.0050
- 0.0023 -0.0151
- 0.0002 0.0008
- 0.0007 0.0035
- 0.9998 0.0178

□ wine\_fields =

- Origin
- Alcohol
- Malic acid
- Ash
- Alcalinity of ash
- Magnesium
- Total phenols
- Flavanoids
- Nonflavanoid phenols
- Proanthocyanins
- Color intensity
- Hue
- OD280/OD315 of diluted wines
- Proline

# Resultados LDA – Wine



# Resultados LDA – Wine

□ w =

- -0.1241 0.2644
- 0.0631 0.0878
- -0.0848 0.7003
- 0.0511 -0.0458
- -0.0008 -0.0001
- 0.2144 -0.0193
- -0.5869 -0.1194
- -0.5506 -0.4592
- 0.0409 -0.0930
- 0.1282 0.0694
- -0.3127 -0.4357
- -0.4017 0.0334
- -0.0009 0.0009

□ wine\_fields =

- Origin
- Alcohol
- Malic acid
- Ash
- Alcalinity of ash
- Magnesium
- Total phenols
- Flavanoids
- Nonflavanoid phenols
- Proanthocyanins
- Color intensity
- Hue
- OD280/OD315 of diluted wines
- Proline

# Subspace Classification

- 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.

# Independent Component Analysis

(ICA)

□ 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  $\underline{y} \in \mathfrak{R}^{\ell}$

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

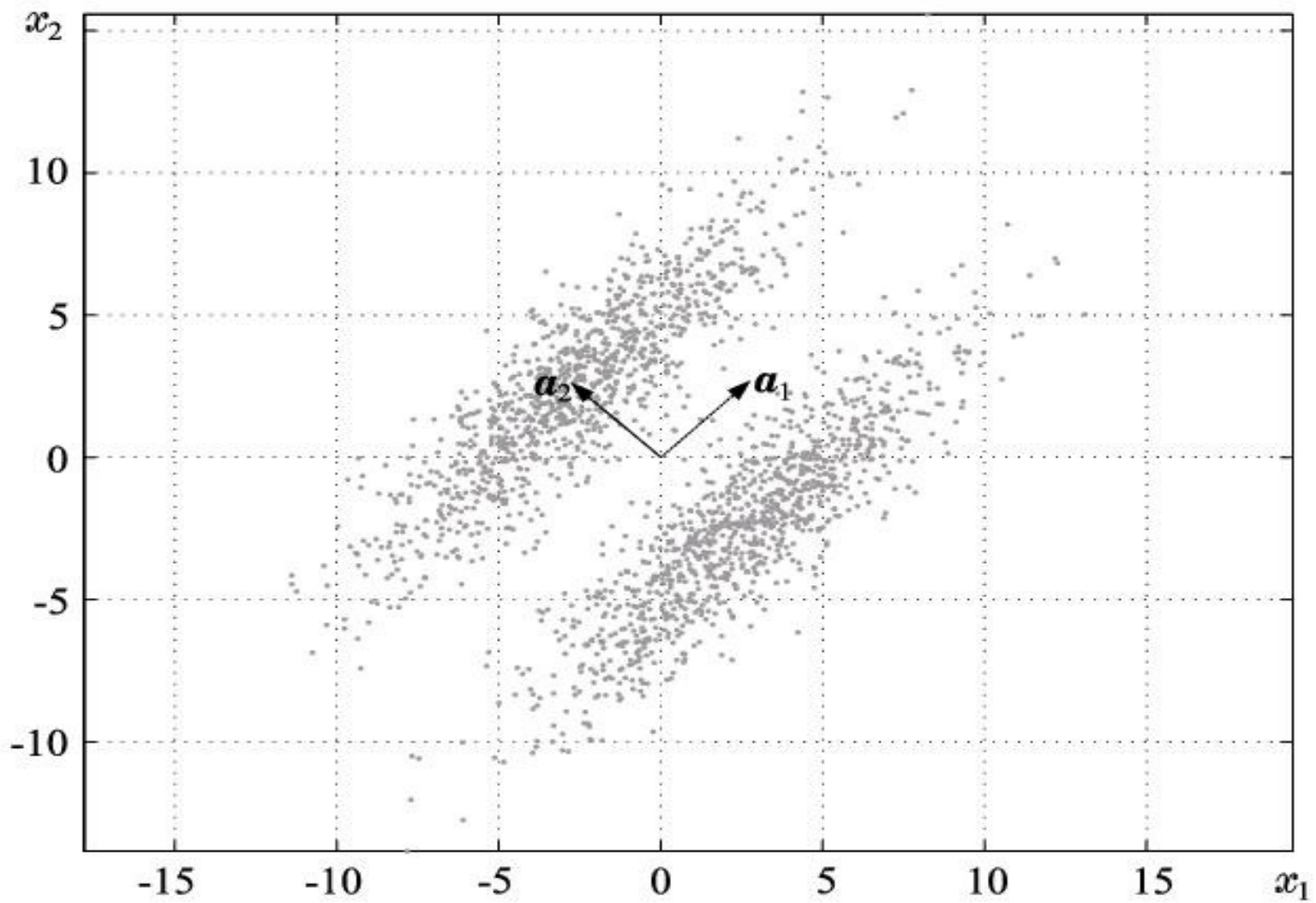
- so that the components of  $\underline{y}$  are statistically independent. In order the problem to have a solution, the following assumptions must be valid:

- 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.





# Measures of nongaussianity

- To use nongaussianity in ICA estimation, we must have a quantitative measure of nongaussianity of a random variable, say  $y$ . To simplify things, let us assume that  $y$  is centered (zero-mean) and has variance equal to one.

- **Kurtosis**

- The classical measure of nongaussianity is kurtosis or the fourth-order cumulant. The kurtosis of  $y$  is classically defined by

$$\text{kurt}(y) = E \{y^4\} - 3(E \{y^2\})^2$$

- **Negentropy**

- A second very important measure of nongaussianity is given by negentropy. Negentropy is based on the information-theoretic quantity of (differential) entropy.

- Negentropy  $J$  is defined as follows

$$J(\mathbf{y}) = H(\mathbf{y}_{\text{gauss}}) - H(\mathbf{y})$$

# Eigenfaces

Let  $X = \{x_1, x_2, \dots, x_n\}$  be a random vector with observations  $x_i \in \mathbb{R}^d$ .

1. Compute the mean  $\mu$

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

2. Compute the the Covariance Matrix  $S$

$$S = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T$$

3. Compute the eigenvalues  $\lambda_i$  and eigenvectors  $v_i$  of  $S$

$$Sv_i = \lambda_i v_i, i = 1, 2, \dots, n$$

4. Order the eigenvectors descending by their eigenvalue. The  $k$  principal components are the eigenvectors corresponding to the  $k$  largest eigenvalues.

The  $k$  principal components of the observed vector are then given by:


$$y = W^T(x - \mu)$$

where  $W = (v_1, v_2, \dots, v_k)$



# Eigenface recognition

- The Eigenfaces method then performs face recognition by:
- Projecting all training samples into the PCA subspace.
- Projecting the query image into the PCA subspace.
- Finding the nearest neighbor between the projected training images and the projected query image.



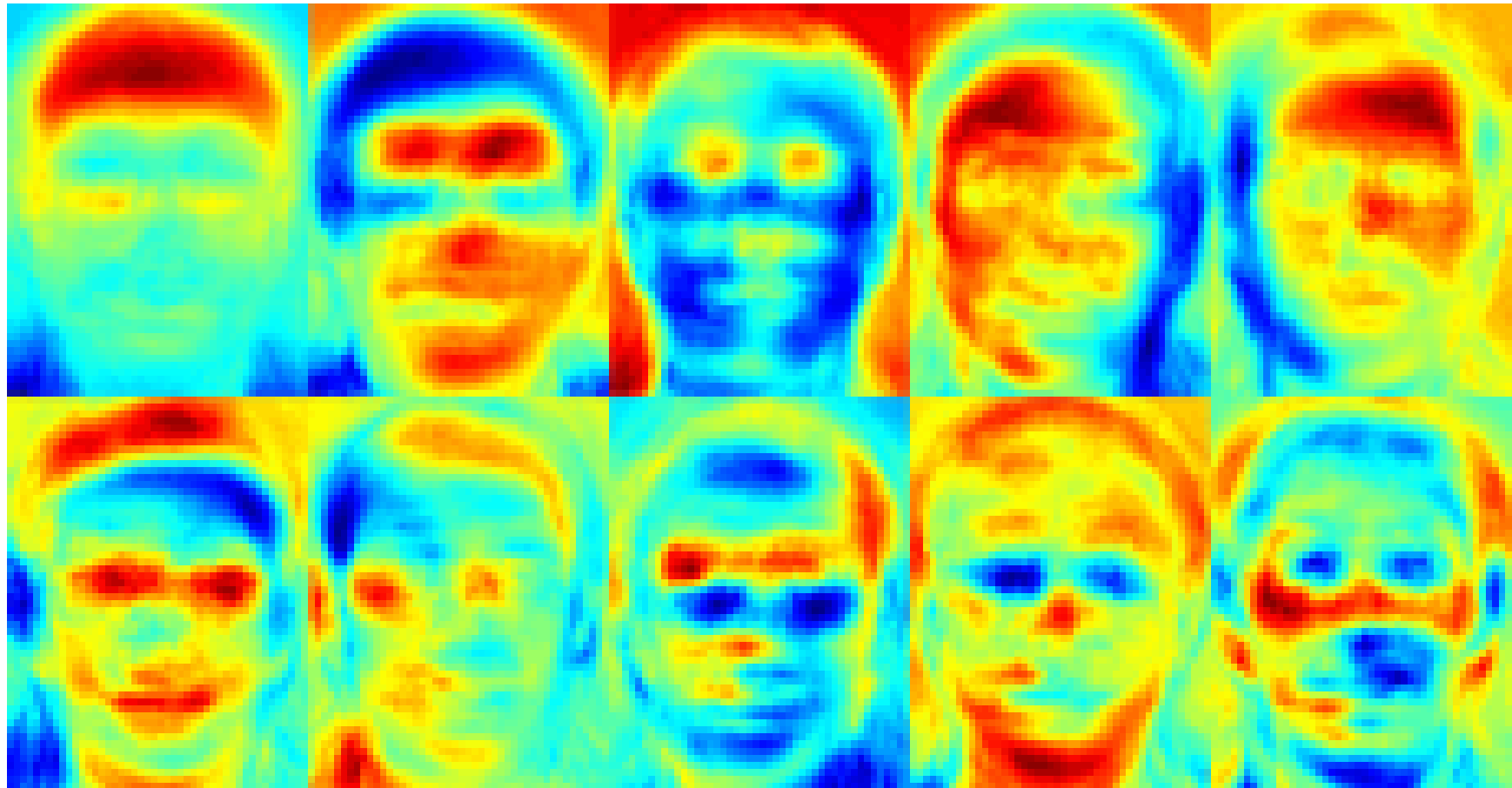
Still there's one problem left to solve. Imagine we are given 400 images sized  $100 \times 100$  pixel. The Principal Component Analysis solves the covariance matrix  $S = XX^T$ , where  $\text{size}(X) = 10000 \times 400$  in our example. You would end up with a  $10000 \times 10000$  matrix, roughly 0.8GB. Solving this problem isn't feasible, so we'll need to apply a trick. From your linear algebra lessons you know that a  $M \times N$  matrix with  $M > N$  can only have  $N - 1$  non-zero eigenvalues. So it's possible to take the eigenvalue decomposition  $S = X^T X$  of size  $N \times N$  instead:

$$X^T X v_i = \lambda_i v_i$$

and get the original eigenvectors of  $S = XX^T$  with a left multiplication of the data matrix:

$$XX^T (X v_i) = \lambda_i (X v_i)$$

# Imagem das eigenfaces





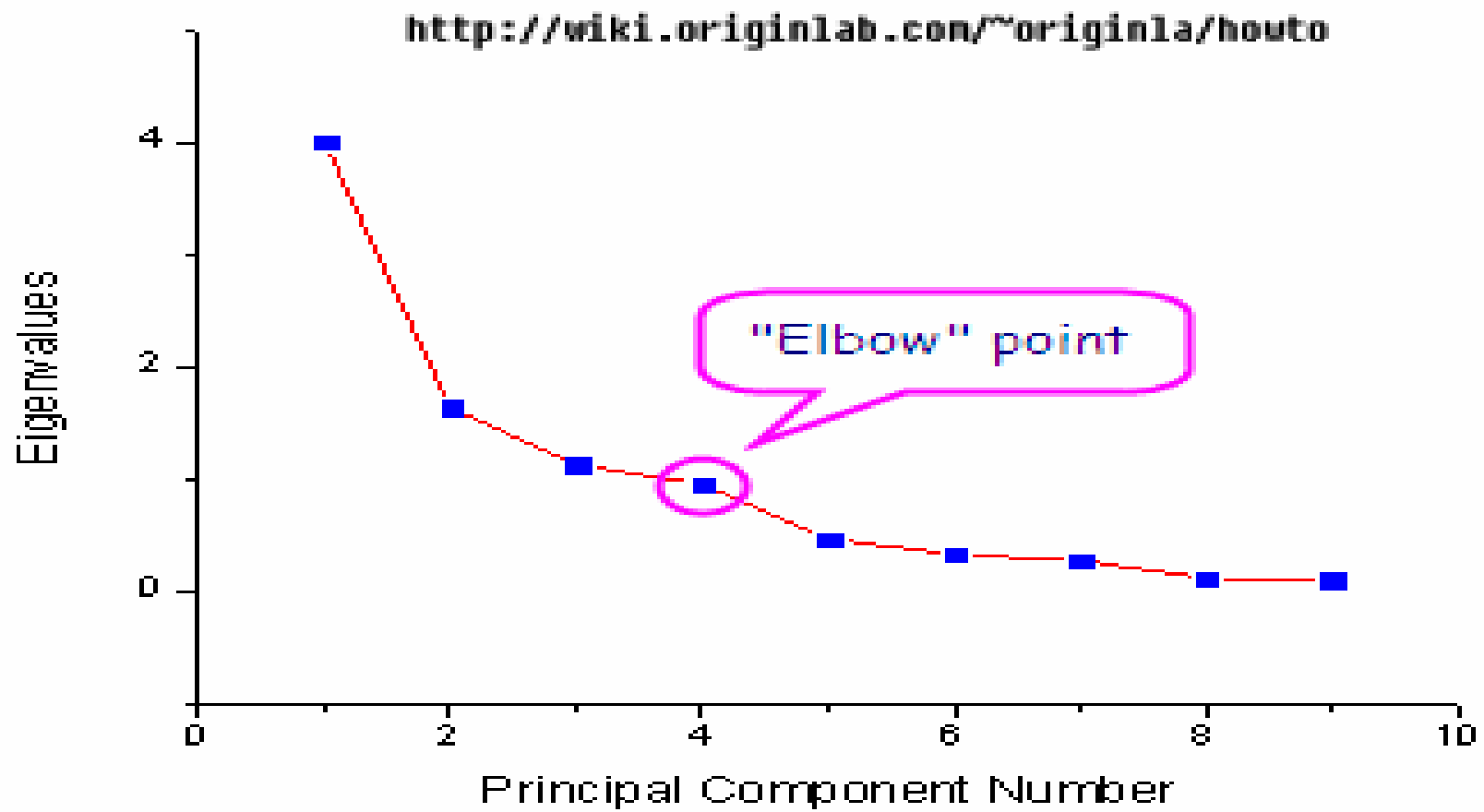


# Scree plot

- A plot, in descending order of magnitude, of the eigenvalues of a correlation matrix. In the context of factor analysis or principal components analysis a scree plot helps the analyst visualize the relative importance of the factors — a sharp drop in the plot signals that subsequent factors are ignorable.



# Scree plot





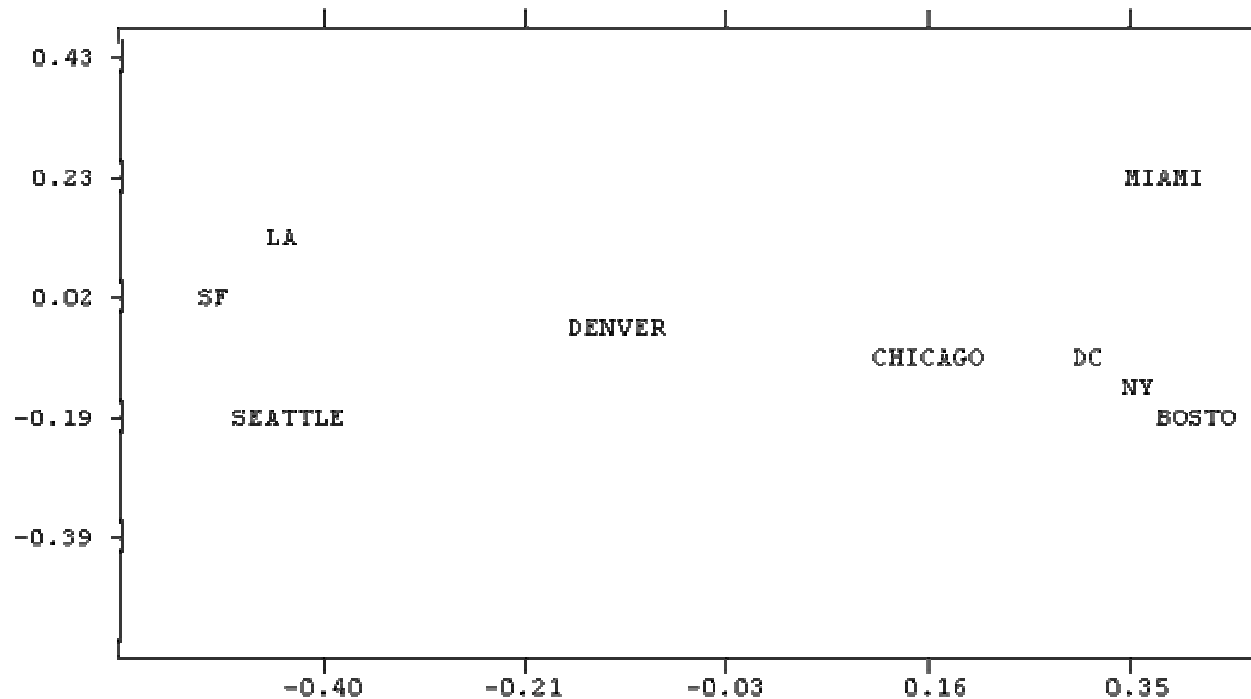
# Mais técnicas de redução de dimensionalidade

- ❑ MDS (multidimensional scaling)
- ❑ Nonlinear mappings
- ❑ Projection pursuit
- ❑ Grand tours
- ❑ Kernel PCA
- ❑ Autoencoder / Sparse coding
- ❑ Máquinas de Boltzman Restritas

□ <http://www.analytictech.com/networks/mds.htm>

# MDS

	1	2	3	4	5	6	7	8	9
	BOST	NY	DC	MIAM	CHIC	SEAT	SF	LA	DENV
	----	----	----	----	----	----	----	----	----
1 BOSTON	0	206	429	1504	963	2976	3095	2979	1949
2 NY	206	0	233	1308	802	2815	2934	2786	1771
3 DC	429	233	0	1075	671	2684	2799	2631	1616
4 MIAMI	1504	1308	1075	0	1329	3273	3053	2687	2037
5 CHICAGO	963	802	671	1329	0	2013	2142	2054	996
6 SEATTLE	2976	2815	2684	3273	2013	0	808	1131	1307
7 SF	3095	2934	2799	3053	2142	808	0	379	1235
8 LA	2979	2786	2631	2687	2054	1131	379	0	1059
9 DENVER	1949	1771	1616	2037	996	1307	1235	1059	0





# Acknowledgments

- Some of these slides are based on Theodoridis and Koutrombas' Pattern Recognition book slides.
- Some of these slides are based on OpenCV documentation