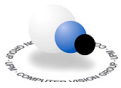
CVG-UPM

COMPUTER VISION

Machine Learning & Neural Networks

2.- Feature processing

by
Pascual Campoy
Grupo de Visión por Computador
U.P.M. - DISAM



CVG-UPM

COMPUTER VISION

Feature processing

- *Objectives*
- *Quality criteria*
- *Feature selection*
- *Unsupervised linear processing*
- *Supervised linear processing*

COMPUTER VISION

Objectives

1. *Mejorar los resultados del reconocimiento: disminuir las probabilidades de clasificación errónea o del riesgo.*
2. *Simplificar el reconocedor, tanto en la fase de aprendizaje como en la de ejecución*

Mathematical formulation:

Dado un conjunto de características de entrada al clasificador x (dim n), encontrar una función $y=F(x)$ de $R^n \Rightarrow R^m$, tal que optimice un criterio de calidad Q(y) definido sobre y

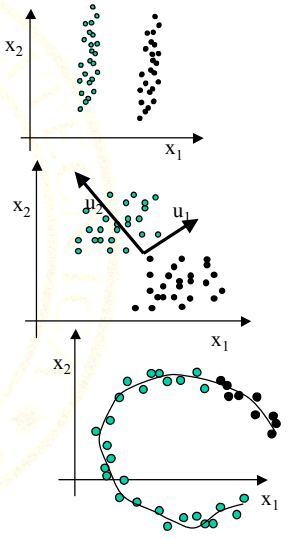
COMPUTER VISION

Processing types

Supresión de las características menos relevantes

Cálculo de nuevas características mediante combinación lineal

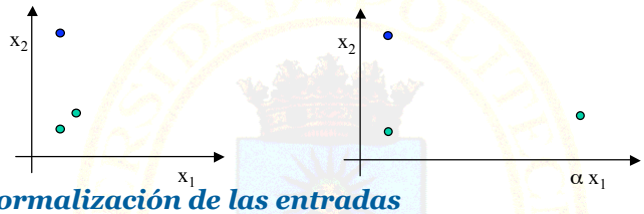
Cálculo de nuevas entradas mediante procesamiento no lineal



Quality criteria: methodologies

- ***Directos: minimización de la probabilidad de clasificación errónea o riesgo esperador***
  - óptimo: tiene en cuenta que un mayor número de características no necesariamente mejora la clasificación
  - dependientes del clasificador o suposición de un clasificador óptimo
  - resolución muy compleja
- ***Indirectos: aumento de la separabilidad entre las muestras pertenecientes a cada clase***
  - ligado al concepto de distancia
  - la reducción de la dimensión de las características siempre empeora el criterio
  - resolución más sencilla

Quality criteria: normalization



- ***Normalización de las entradas***
  - Cada componente por separado
    - $x'^{n}_i = (x^n_i - \underline{x}_i)/\sigma_i$

    consigue media 0 y σ=1

  Conjuntamente

  $x' = \Lambda^{-1/2} U^T (x^n - x)$

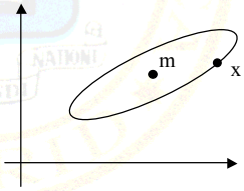
  siendo $\Lambda = diag\{\lambda_1 ... \lambda_D\}$; $U = [u_1 ... u_D]$

  donde λ_i u_i son los valores propios y U^T los vectores propios de la matriz de C de covarianzas

  consigue media 0 y C=I

Quality criteria: distances

- ***Distancias en el espacio de entradas***
  - Distancia de Manhatan
    - $d = \sum_i |x_i - m_i|$
  - Distancia euclidea
    - $d = \text{sqrt}\ (\sum_i (x_i - m_i)^2) = \text{sqrt}\ ((\mathbf{x}-\mathbf{m})^T\ (\mathbf{x}-\mathbf{m}))$
  - Distancia de Mahalanovich
    - $d = \text{sqrt}\ ((\mathbf{x}-\mathbf{m})^T\ \mathbf{C}^{-1}\ (\mathbf{x}-\mathbf{m}))$
      - siendo $C = \sum_i (\mathbf{x}_i-\mathbf{m})\ (\mathbf{x}_i-\mathbf{m})^T$ la matriz de covarianzas



Quality criteria: objective functions

- ***Separabilidad de características***
  - Matriz de dispersión de un grupo
    - $S_i = \Sigma_j\ (x_j - m_i)\ (x_j - m_i)^T$
  - Matriz de dispersión intra-grupos
    - $S_W = \Sigma_i\ S_i$
  - Matriz de dispersión inter-grupos
    - $S_B = \Sigma_i\ n_i\ (m_i - m)\ (m_i - m)T$
  - Matriz de dispersión total

$$S_T = \Sigma_j\ (x_j - m)\ (x_j - m)^T = S_W + S_B$$

Feature processing

- Objectives
- Quality criteria
- Feature selection
- Unsupervised linear processing
- Supervised linear processing

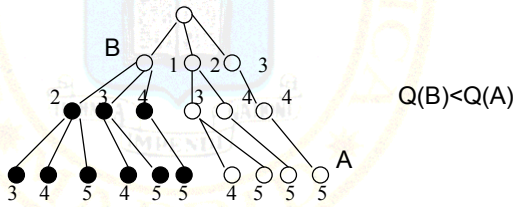
Feature selection ...

- Búsqueda exhaustiva: $O(2^n)$ ó $O(n!/(n-m)!m!)$

Brach and bound: Algoritmos de Dijstra, Floyd, A*

- óptimo
- ahorro computacional basado en el cálculo del criterio para conjuntos mayores, teniendo en cuenta el decrecimiento monótono de dicho criterio de calidad

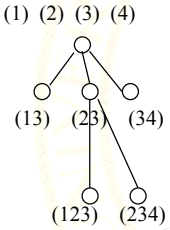
ejemplo: 2 mejores características de 5



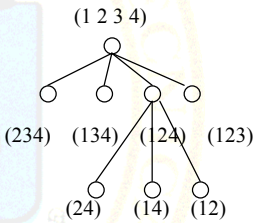
...feature selection

- **Búsqueda por técnicas heurísticas subóptimas:**

adición de características



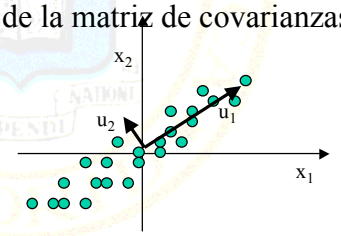
eliminación de características



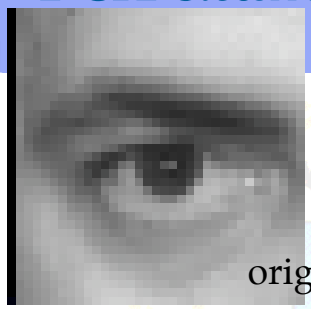
Unsupervised linear processing: PCA

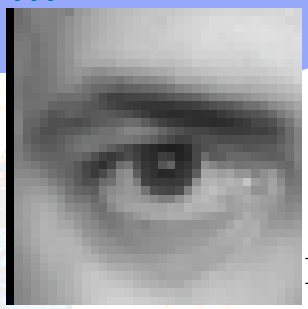
- **Análisis de componentes principales:**
  - Dado un conjunto de N vectores n-dimensionales x_i, donde $x_i = \sum_{j=1}^{n} a_{ij} u_j = \sum_{j=1}^{m} a_{ij} u_j + \sum_{j=m+1}^{n} a_{ij} u_j$ hallar una base ortonormal u_j tal que al seleccionar M vectores de base $x_i' = \sum_{j=1}^{m} a_{ij} u_j + \sum_{j=m+1}^{n} b_j u_j$ minimice $E = \sum_{i=1}^{N} (x_i - x_i')^2$

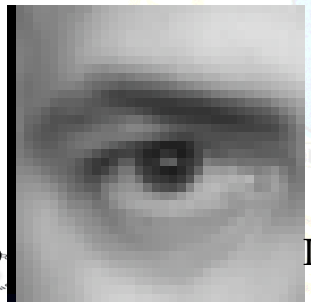
Solución: u_j son los vectores propios correspondientes a los mayores valores propios de la matriz de covarianzas y $E = \frac{1}{2} \sum_{i=m+1}^{n} \lambda_i$

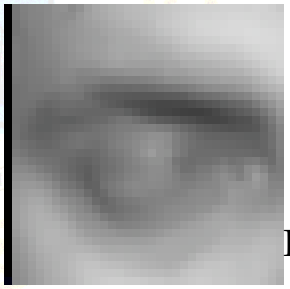


PCA example ...


original

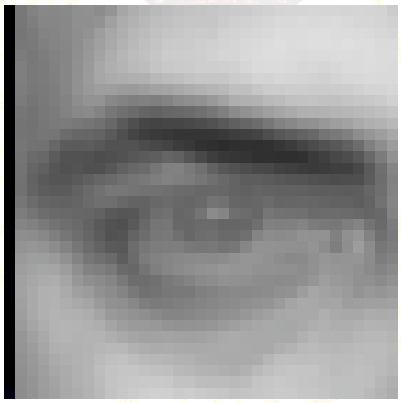

PCA 25


PCA 5


PCA 1

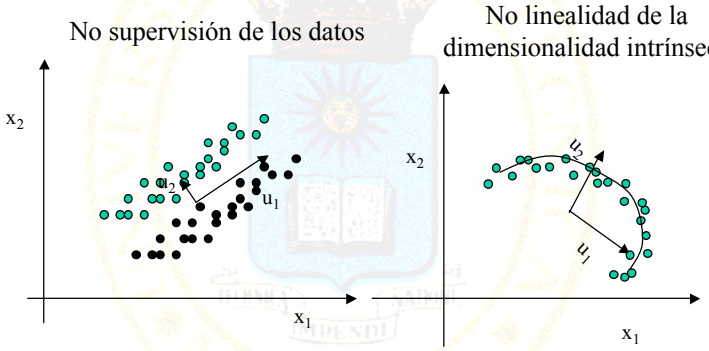
... PCA example

Imágenes sintéticas creadas sobre el subespacio de dimensión 1



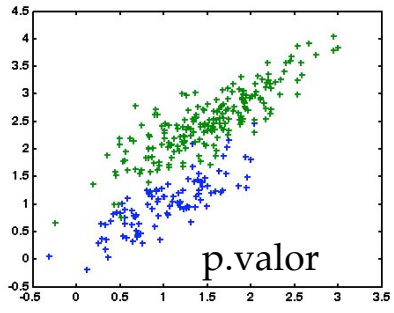
Unsupervised linear processing: Drawbacks

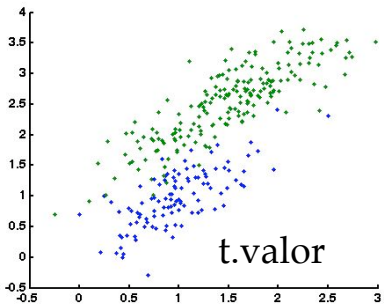
- ***Limitaciones del Análisis de Componentes Principales***



Example 3.1: data

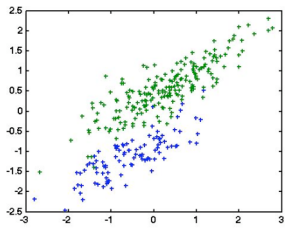
```
>> load datos_D2_C2.mat
>> plot(p.valor(1,1:100),p.valor(2,1:100),'+'); hold all;
>> plot(p.valor(1,101:300),p.valor(2,101:300),'+');
```





Example 3.1: normalization

```
[D,N]=size(p.valor);  [D,Nt]=size(t.valor);
clear pn; clear meanp; clear stdp;
meanp=mean(p.valor')';  stdp=std(p.valor')';
 for i=1:N
    pn(:,i)=(p.valor(:,i)-meanp)./stdp;
 end
 for i=1:Nt
     tn(:,i)=(t.valor(:,i)-meanp)./stdp;
 end
```



alternativas:

```
[pn,meanp,stdp,tn,meant,stdt]=prestd(p.valor,t.valor);
[pn,meanp,stdp]=zscore(p.valor'); pn=pn';meanp=meanp';
                                  std=std'
```

Example 3.1: PCA

```
[transMatc,Diag]=eig(cov(pn')); ncompca=1;
for i=1:ncompca
transMat(i,:)=transMatc(:,D+1-i)';
end
error=Diag(1,1)/2;
```

alternativas:

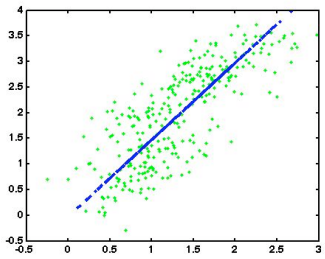
```
[ptrans,transMat]=prepca(pn,0.2);
[transMatc, ptransc] = princomp(pn'); transMatc=transMatc';
                                      ptrans=ptransc';
[residual,preconstructed]=pcares(pn',ncompca);
```

COMPUTER VISION

Exercise 3.1

>> load datos_D2_C2

1. Calcular los vectores de coordenadas reducidas de los puntos de test sobre el subespacio definido por el PCA (3 puntos)
2. Calcular los vectores de dimensión completa del apartado anterior y dibujarlos junto con los puntos de test. (4 puntos)
3. Calcular el ECM cometido mediante la aproximación PCA (3 puntos)



COMPUTER VISION

Supervised linear processing ...

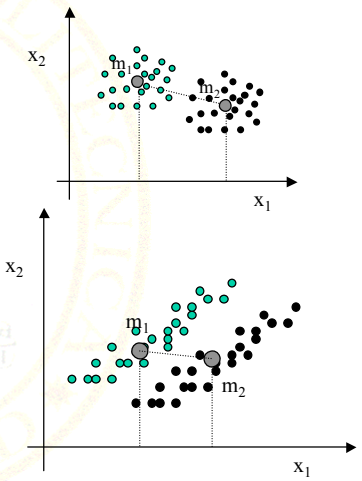
- **Discriminante bidimensional**
  - $y = \mathbf{w}^T \mathbf{x}$
  - idea inicial: maximizar $m_2 - m_1 =$
    $= \mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1) \Rightarrow \mathbf{w} \propto (\mathbf{m}_2 - \mathbf{m}_1)$
    no valida para distribuciones no isotrópicas

discriminante de Fisher:

$J(\mathbf{w}) = (m_2 - m_1)^2 / (s_1^2 + s_2^2) =$
$= (\mathbf{w}^T \mathbf{S}_B \mathbf{w}) / (\mathbf{w}^T \mathbf{S}_W \mathbf{w})$

maximización: $\mathbf{w} \propto \mathbf{S}_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1)$

isotrópico: $\mathbf{S}_W^{-1} = \mathbf{I}$



... supervised linear processing

- ***Discriminante de Fisher multidimensional:***
  - transformación: $\mathbf{y} = \mathbf{W}^T \mathbf{x}$
  - discriminante:

    $J(\mathbf{w}) = \text{Traza}\{s_W^{-1} s_B\} = \text{Traza}\{(\mathbf{W}^T S_W \mathbf{W})^{-1} (\mathbf{W}^T S_B \mathbf{W})\}$

  maximización:

  $\mathbf{W}$ vectores propios de $\mathbf{S}_W^{-1}\mathbf{S}_B$

  $\mathbf{W}$ tiene una dimensión máxima de c-1, siendo c el numero de clases

  isotrópico: $\mathbf{S}_W^{-1} = \mathbf{I}$