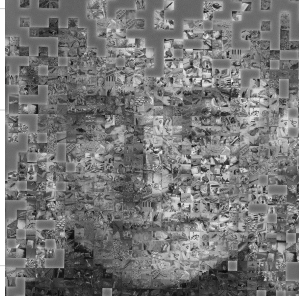


## RECONOCIMIENTO DE CARAS

CLASE 1 :

“Sistemas Estándares de Reconocimiento Eigen-space”

Profesor : Javier Ruiz del Solar

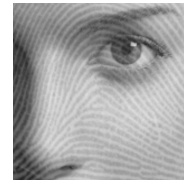


Procesamiento Avanzado de Imágenes

EL708

## Temario

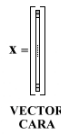
- Introducción
- Sistema Estándar de Reconocimiento
- Métodos de Proyección
- Medidas de Similitud
- Resumen



## Introducción

### - Representación :

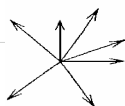
- Una cara está representada por los píxeles de su imagen digital.
- Una cara será tratada como un vector formado por todos los píxeles de su imagen.



- Las caras son un conjunto particular en el espacio de imágenes.

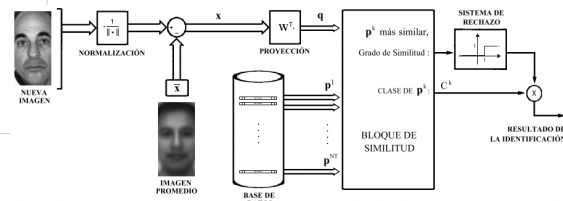
### - Supuesto :

- El espacio de caras es de menor dimensionalidad que el espacio de imágenes.



IMÁGENES

## Sistema Estándar de Reconocimiento



### - Etapas para el Reconocimiento :

- Normalización (señales con la misma energía)
- Media Nula (centrar las señales o quitar linealidad afín)
- Proyección Lineal (matriz de transformación  $W^T$ , reduce dimensión)
- Comparación con Base de Datos (vectores reducidos)
- Sistema de Rechazo (usando criterio óptimo bayesiano)

## Métodos de Proyección y Reconocimiento

### - Métodos de Proyección :

- PCA – Principal Component Analysis
- FLD – Fisher Linear Discriminant
- EP – Evolutionary Pursuit

### - Medidas de Similitud :

- Euclidean
- Cosine
- Mahalanobis (PCA-Whitening)
- SOM - Self Organizing Map

## Métodos de Proyección

### PCA – Principal Component Analysis

- Resumen : Buscar espacio reducido en que los vectores queden lo más separados unos de otros (maximizar varianza).

- Obtención de la Matriz de Correlación :  $R = E \{ x x^T \} = \frac{1}{N} \sum_{i=1}^N x^i (x^i)^T$
- Cálculo de valores y vectores propios de R :  $R w^k = \lambda_k w^k$

- vectores propios  $\Rightarrow$  componentes principales – Eigen-faces
- valores propios  $\Rightarrow$  varianzas de proyección asociadas

$$\sigma_{w^k}^2 = (w^k)^T R w^k = \lambda_k$$

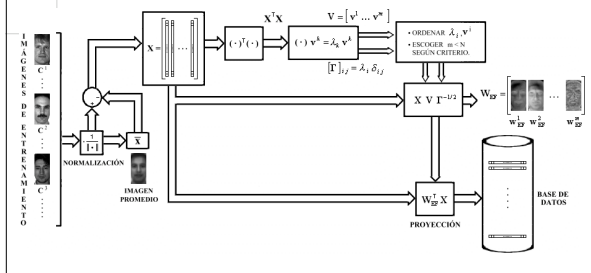
### - Problema con este método :

- Dimensión de R :  $N = (n^o \text{ píxeles}) \times (n^o \text{ píxeles})$   
Ejemplo: imagen de  $N=100 \times 200 \Rightarrow R$  de dimensión  $N \times N$  con  $N=20.000$  !
- Algoritmos para resolver problema de autovalores son  $O(N^3)$
- ¿ Cuántos valores propios no nulos tiene R ? ¿ N ?

## PCA – Principal Component Analysis

### - Solución del Problema :

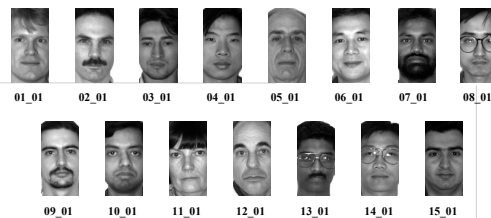
- Sea  $X = [x^1 \ x^2 \ \dots \ x^{NT}] \Rightarrow R = X X^T$
- Número de valores propios no nulos de  $R$  :  
NT = nº de imágenes de entrenamiento
- Resolver problema de autovalores para  $X^T X \in R^{NT \times NT}$



## PCA – Principal Component Analysis

### - Ejemplo :

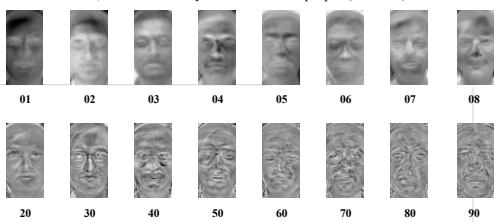
Base de datos de Yale. Se tienen 15 individuos (clases) con 10 caras por cada uno. La resolución fue reducida de 320x243 a 100x200.



## PCA – Principal Component Analysis

### - Ejemplo de Eigenfaces :

Eigenfaces obtenidas a partir de 100 imágenes de entrenamiento de la base de datos de Yale, en orden de mayor a menor valor propio (varianza) asociado.



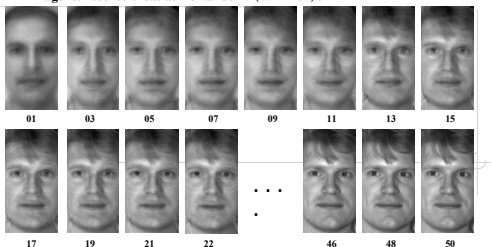
## PCA – Principal Component Analysis

### - Capacidad de Reconstrucción :

• Ecuación de Síntesis :  $x = \sum_{k=1}^m p_k w^k$

• Reconstrucción :  $x \approx \sum_{k=1}^m p_k w^k$

• Imágenes reconstruidas aumentando m (con NT=50) :



## ... Métodos de Proyección

### FLD – Fisher Linear Discriminant

- Resumen :** Buscar espacio reducido en que los vectores de una misma clase queden juntos y aquellos de distintas clases queden separados.
- Maximizar "Parámetro de Fisher" –  $\gamma$ .

$$\gamma(u) = \frac{u^T S_b u}{u^T S_w u} \leftarrow \text{separación entre elementos de clases distintas.}$$

- Matrices de Dispersión :  $S_b = \sum_{i=1}^{NC} P(C_i) (m^{(i)} - m)(m^{(i)} - m)^T$   
 $S_w = \sum_{i=1}^{NC} P(C_i) E[(x^{(i)} - m^{(i)})(x^{(i)} - m^{(i)})^T]$

- Resolver problema generalizado de autovalores :  $S_b w^k = \lambda_k S_w w^k$

vectores propios  $\Rightarrow$  Fisher - Faces

valores propios  $\Rightarrow$  parámetros de Fisher asociados a cada proyección  
 $\gamma(w^k) = \lambda_k$

## LDA – Linear Discriminant Analysis

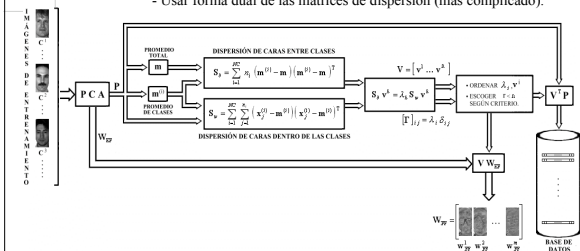
### - Propiedades Importante :

$(w^n)^T S_w w^m = S_w (w^n) \delta_{nm}$

• Vectores Base no son ortogonales :  $(w^n)^T S_b w^m = S_b (w^n) \delta_{nm}$

- Problemas :** - singularidad en la matriz  $S_w$ .
- dimensiones muy grandes en matrices de dispersión.

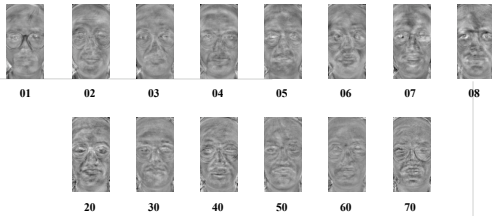
**Solución :** - Aplicar un bloque PCA como pre-procesamiento.  
- Usar forma dual de las matrices de dispersión (más complicado).



## LDA – Linear Discriminant Analysis

### - Ejemplo de Fisherfaces :

Fisherfaces obtenidas a partir de 100 imágenes de entrenamiento de la base de datos de Yale, en orden de mayor a menor valor propio ( $\gamma$ ) asociado.



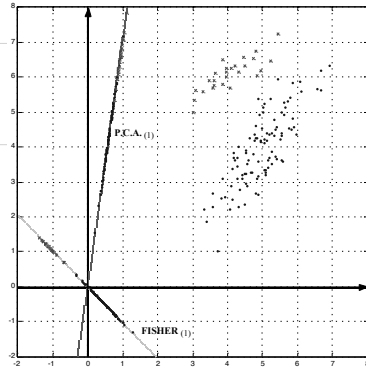
## ... Métodos de Proyección

### • PCA v/s FLD :

- PCA : - El estimador de R converge si N es grande.
  - En general esto se cumple, luego los modelos presentan poca varianza.
  - No usa información de las clases.
  - Buena capacidad de generalización (no puede sobreajustarse).
  - Capacidades de desempeño limitada.
- FLD : - Estimadores de las matrices de dispersión convergen si el número de imágenes por clase es grande.
  - En general esto no se cumple, luego los modelos presentan alta varianza.
  - Se usa información de las clases.
  - Mala capacidad de generalización (se ajusta a la información de clases).
  - Buen desempeño, si las estimaciones estadísticas son correctas.

### • PCA v/s FLD :

EJES DE PROYECCIÓN P.C.A. & FISHER



## ... Métodos de Proyección

### EP – Evolutionary Pursuit (Seguimiento Evolutivo)

- **Resumen :** Buscar espacio reducido en que se logre una buena característica tanto en generalización (PCA) como en desempeño (FLD).
- Maximizar Función Objetivo :
 
$$\zeta(\mathbf{u}^i) = \zeta_d(\mathbf{u}^i) + \lambda \zeta_g(\mathbf{u}^i)$$

↓ medición del desempeño  
↑ indicador de generalización
- No existe un indicador estadístico directo sobre el desempeño,  
⇒ función objetivo no tiene una expresión analítica.
- **Problemas :**
  - Se requieren métodos de optimización para maximizar la función objetivo.
  - Restricción adicional dada por la alta dimensionalidad del espacio de soluciones posibles.
- **Solución Adoptada :** Usar Algoritmos Genéticos.

## EP – Evolutionary Pursuit (Seguimiento Evolutivo)

### • Etapas :

- Reducción de Dimensionalidad usando PCA.
- Transformada Whitening : Normaliza las varianzas de proyección al dividir cada componente por la raíz de su valor propio (varianza).

$$\tilde{\mathbf{A}}^{-1/2} = \begin{bmatrix} \lambda_1^{-1/2} & 0 & 0 \\ 0 & \lambda_2^{-1/2} & 0 \\ 0 & 0 & 0 \\ & 0 & \lambda_n^{-1/2} \end{bmatrix}$$

- Generar conjunto de soluciones formado por rotaciones y selección de algunas de las componentes de PCA-Whitening.
- Representación en base al siguiente Cromosoma :

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{n(n-1)/2} & a_1 & a_2 & \dots & a_n \end{bmatrix}$$

- ángulos  $\alpha$  representados por 10 bits.
- bits  $a_i$  indican las componentes seleccionadas por el cromosoma.

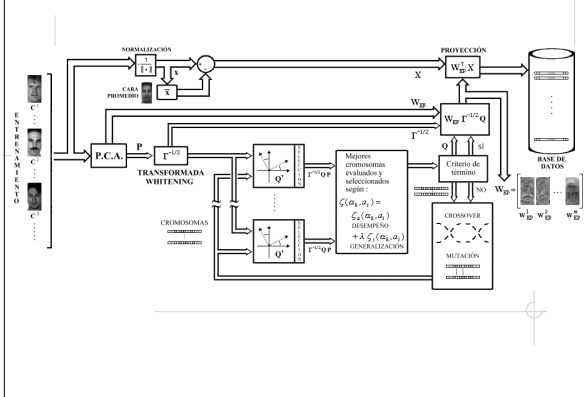
## EP – Evolutionary Pursuit (Seguimiento Evolutivo)

- Función Objetivo :  $\zeta(\alpha_i, a_i) = \zeta_d(\alpha_i, a_i) + \lambda \zeta_g(\alpha_i, a_i)$
- Desempeño se mide reconociendo las caras de la base de datos.
- Generalización se mide como la separación entre vectores de una misma clase:

$$\zeta_g(\alpha_i, a_i) = \sqrt{\sum_{i=1}^{NC} (\mathbf{m}^{(i)} - \mathbf{m})^T (\mathbf{m}^{(i)} - \mathbf{m})}$$

- Problema de la Función Objetivo :
  - Medición del desempeño exige tener menos ejes de proyección que número de clases. De otro modo el sistema "memoriza" las identidades de la base de datos y el desempeño siempre será 100%.
- Alternativas :
  - Usar conjunto de validación.
  - Hacer reconocimiento usando el "top 2".

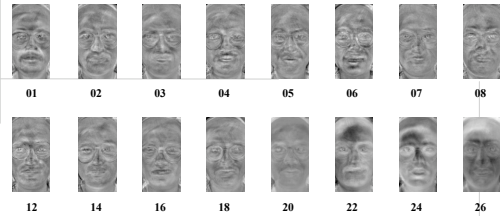
## EP – Evolutionary Pursuit



## EP – Evolutionary Pursuit (Seguimiento Evolutivo)

### - Ejemplo de EP-faces :

EP-faces obtenidas a partir de 100 imágenes de entrenamiento de la base de datos de Yale.



## Medidas de Similitud

• **Euclidean :**  $d(x, y) = \sqrt{(x - y)^T (x - y)}$

• Considera diferencia entre caras y error de reconstrucción.

• **Coseno :**  $\cos(x, y) = \frac{x^T y}{\|x\| \|y\|}$

• Representa la distancia sobre la superficie de una hiper-esfera.

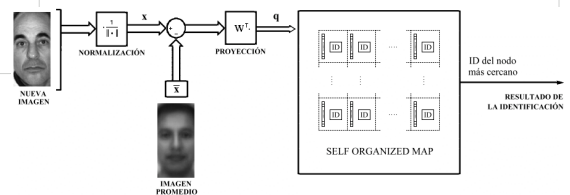
• **Mahalanobis :**  $d(x, y) = (x - y)^T R^{-1} (x - y)$

• Produce un efecto de escalamiento de la métrica.  
• Comprime las direcciones con mayor varianza, y  
• Estira las direcciones con menor varianza.  
• La métrica euclidiana permite identificar clusters esféricos. Mahalanobis permite localizar clusters elípticos.

## ... Medidas de Similitud

### • Self Organizing Map :

• Cambia la estructura del bloque de similitud.  
• Mejra la habilidad de generalización del sistema.



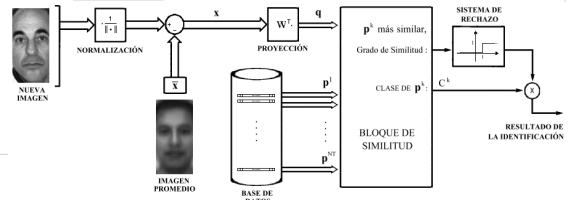
• Métodos de Entrenamiento :  
• Algoritmo SOM estándar.  
• Algoritmo Dot Product - SOM.

## ... Medidas de Similitud

### • Sistema Adaptivo SOM :

• Diseñado para adaptarse a los cambios en las caras:  
• las personas que se reconocen son vistas constantemente por el sistema.  
• Cuando llega una nueva cara:  
• el sistema neuronal realiza el reconocimiento.  
• la red SOM realiza una iteración de entrenamiento.

## Resumen



• **Métodos de Proyección :**  
• PCA – Principal Component Analysis  
• FLD – Fisher Linear Discriminant  
• EP – Evolutionary Pursuit

• **Medidas de Similitud :**  
• Euclidean  
• Coseno  
• Mahalanobis (PCA-Whitening)  
• SOM



CONTINUARÁ ...



Procesamiento Avanzado de Imágenes

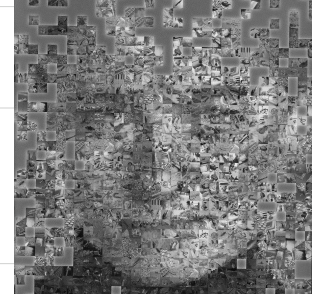
EL708

## RECONOCIMIENTO DE CARAS

CLASE 2 :

“ Sistemas Diferenciales  
de Reconocimiento  
Eigen-space ”

Profesor : Javier Ruiz del Solar

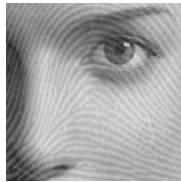


Procesamiento Avanzado de Imágenes

EL708

### Temario

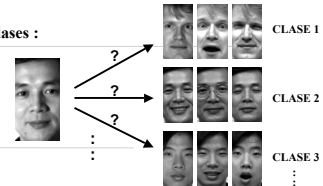
- Introducción
- Motivación de  
Métodos Diferenciales
- Sistemas Diferenciales  
de Reconocimiento
- Métodos de Proyección
- Métodos de Clasificación
- Esquema Unificado



### Motivación de Métodos Diferenciales

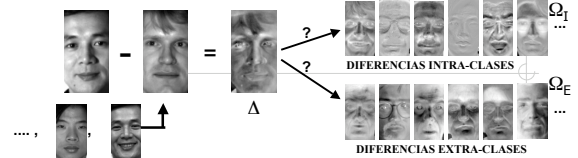
- Problema de NC clases :

muchas clases /  
pocas muestras  
por clase

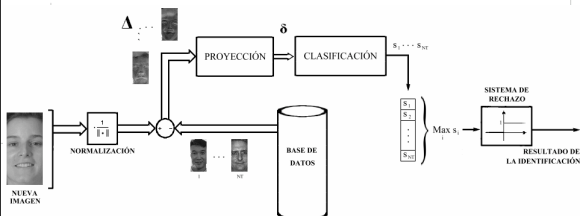


- Problema de 2 clases :

2 clases / muchas muestras por clase.



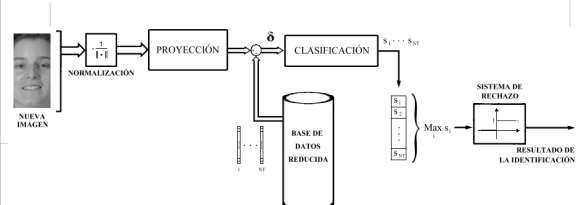
### Sistema Pre-diferencial de Reconocimiento



- Etapas para el Reconocimiento :

- Normalización ( señales con la misma energía )
- Diferencia de Imágenes ( $\Delta$ = imagen nueva - imagen de la base de datos )
- Proyección ( proyecta la imagen diferencial a un espacio reducido )
- Clasificación ( determina el grado de pertenencia a  $\Omega_i$  de la imagen diferencial )
- Sistema de Rechazo ( usando criterio óptimo bayesiano )

### Sistema Post-diferencial de Reconocimiento



- Etapas para el Reconocimiento :

- Normalización ( señales con la misma energía )
- Proyección ( proyecta la imagen diferencial a un espacio reducido )
- Diferencia de Proyecciones (  $\delta$  = proyección nueva - proyección de la base de datos )
- Clasificación ( determina el grado de pertenencia a  $\Omega_i$  del vector diferencial )
- Sistema de Rechazo ( usando criterio óptimo bayesiano )

## Métodos de Proyección y Clasificación

### Métodos de Proyección :

- PCA Simple
- PCA Dual

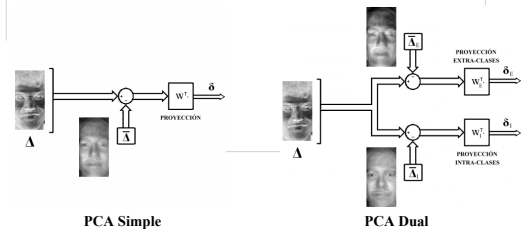
### Métodos de Clasificación :

- Clasificador de Bayes
- SVM – Support Vector Machine

## Métodos de Proyección

**PCA Simple** • Se proyectan las diferencias de caras usando PCA en forma análoga a los métodos estándares.

**PCA Dual** • Separa proyección de diferencias entre caras :  
 • pertenecientes a una misma clase (Intra-clases),  
 • pertenecientes a clases distintas (Extra-clases).  
 • Requiere calcular 2 matrices de proyección PCA.



## Métodos de Clasificación

### Clasificador de Bayes

• Se modela la distribución de probabilidad de las diferencias entre caras como una distribución normal :

$$P(\tilde{\mathbf{A}}/\tilde{\mathbf{U}}) = \frac{\exp\left[-\frac{1}{2}(\tilde{\mathbf{A}} - \tilde{\mathbf{A}})^T \mathbf{R}^{-1}(\tilde{\mathbf{A}} - \tilde{\mathbf{A}})\right]}{(2\pi)^{N/2} |\mathbf{R}|^{1/2}}$$

$\tilde{\mathbf{A}} \in \tilde{\mathbf{U}}_1$  : intra-clase

$\tilde{\mathbf{A}} \in \tilde{\mathbf{U}}_E$  : extra-clase

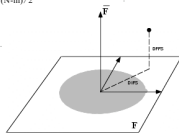
• La expresión se puede manejar al separar las componentes principales y el error de reconstrucción :

$$P(\tilde{\mathbf{A}}/\tilde{\mathbf{U}}) = P(\tilde{\mathbf{a}}/\tilde{\mathbf{U}}) P(\tilde{\mathbf{A}}/\tilde{\mathbf{a}}) = \frac{\exp\left[-\frac{1}{2} \sum_{j=1}^m \frac{\tilde{a}_j^2}{\lambda_j}\right]}{(2\pi)^{m/2} \prod_{j=1}^m \lambda_j^{1/2}} \cdot \frac{\exp\left[-\frac{\tilde{\mathbf{a}}^2(\tilde{\mathbf{A}}\tilde{\mathbf{a}})}{2\rho}\right]}{(2\pi\rho)^{(N-m)/2}}$$

$$\text{Ejes principales} \Rightarrow \tilde{\mathbf{A}} = \mathbf{W} \tilde{\mathbf{a}} + \tilde{\mathbf{A}}$$

$$\text{Error de reconstrucción} \Rightarrow \tilde{\mathbf{a}}^2(\tilde{\mathbf{A}}\tilde{\mathbf{a}}) = \|\tilde{\mathbf{A}} - \tilde{\mathbf{A}}\|^2 = \|\tilde{\mathbf{a}}\|^2$$

$$\text{Varianza promedio restante} \Rightarrow \rho = \frac{1}{N-m} \sum_{i=m+1}^N \lambda_i$$



## Clasificador de Bayes

• Usando el modelo probabilístico se puede aplicar la regla de Bayes para calcular la probabilidad *a posteriori* :

$$P(\tilde{\mathbf{A}} \in \tilde{\mathbf{U}}) = P(\tilde{\mathbf{U}}/\tilde{\mathbf{A}}) = \frac{P(\tilde{\mathbf{A}}/\tilde{\mathbf{U}}) P(\tilde{\mathbf{U}})}{P(\tilde{\mathbf{A}}/\tilde{\mathbf{U}}_1) P(\tilde{\mathbf{U}}_1) + P(\tilde{\mathbf{A}}/\tilde{\mathbf{U}}_E) P(\tilde{\mathbf{U}}_E)}$$

• Para una imagen nueva buscamos las imágenes con las cuales forma diferencias intra-caras (misma clase). Estas cumplen :

$$P(\tilde{\mathbf{U}}_1/\tilde{\mathbf{A}}) - P(\tilde{\mathbf{U}}_E/\tilde{\mathbf{A}}) > 0$$

• Luego, el valor entregado por el clasificador debe ser :

$$S(\tilde{\mathbf{A}}) = P(\tilde{\mathbf{A}}/\tilde{\mathbf{U}}_1) P(\tilde{\mathbf{U}}_1) - P(\tilde{\mathbf{A}}/\tilde{\mathbf{U}}_E) P(\tilde{\mathbf{U}}_E)$$

$\tilde{\mathbf{A}}$  : diferencia entre imágenes (  $\tilde{\mathbf{a}}$  en el caso post-diferencial )

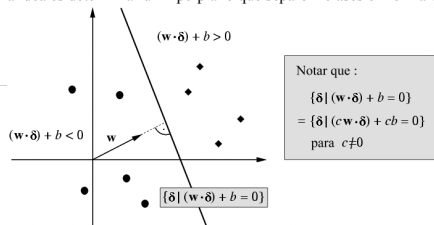
$\tilde{\mathbf{U}}_1$  : intra-clase

$\tilde{\mathbf{U}}_E$  : extra-clase

## ... Métodos de Clasificación

### SVM – Support Vector Machine

• La idea es determinar un hiperplano que separe 2 clases en forma óptima :



Notar que :  
 $\{\delta | (w \cdot \delta) + b = 0\}$   
 $= \{\delta | (c w \cdot \delta) + cb = 0\}$   
 para  $c \neq 0$

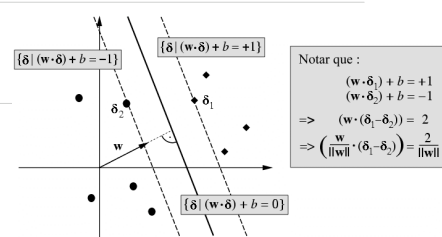
• El vector normal al plano,  $w$ , se define tal que las intra-caras cumplan :

$$w \cdot \tilde{\mathbf{a}} + b > 0 \Rightarrow \tilde{\mathbf{a}} \in \tilde{\mathbf{U}}_1$$

• Luego, el valor entregado por el clasificador debe ser :  $S(\tilde{\mathbf{a}}) = w \cdot \tilde{\mathbf{a}} + b$

## SVM – Support Vector Machine

• Se busca maximizar el margen, i.e. La distancia entre las fronteras más cercanas a ambas clases.



Notar que :  
 $(w \cdot \delta_1) + b = +1$   
 $(w \cdot \delta_2) + b = -1$   
 $\Rightarrow (w \cdot (\delta_1 - \delta_2)) = 2$   
 $\Rightarrow \left( \frac{w}{\|w\|} \cdot (\delta_1 - \delta_2) \right) = \frac{2}{\|w\|}$

ETIQUETAS:

$$y_i = +1 \Leftrightarrow \tilde{\mathbf{a}} \in \tilde{\mathbf{U}}_1$$

$$y_i = -1 \Leftrightarrow \tilde{\mathbf{a}} \in \tilde{\mathbf{U}}_E$$

PROBLEMA PRIMAL (CASO SEPARABLE):

$$\text{Min } \frac{1}{2} \|w\|^2$$

$$\text{sueto a: } y_i (w \cdot \tilde{\mathbf{a}}_i + b) \geq 1, \quad i = 1, \dots, NT$$

## SVM – Support Vector Machine

### LAGRANGIANO:

$$L(w, b, \alpha_i) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{NT} \alpha_i [y_i (w \cdot \tilde{a}_i + b) - 1] \quad \text{donde } \alpha_i \geq 0, \quad i = 1, \dots, NT$$

### DUALIDAD LAGRANGIANA:

$$\begin{aligned} \text{PRIMAL} \quad \min_{w, b} \quad & \max_{\alpha_i \geq 0} L(w, b, \alpha_i) \\ \text{DUAL} \quad \max_{\alpha_i \geq 0} \quad & \min_{w, b} L(w, b, \alpha_i) \end{aligned}$$

¡NO EN CUALQUIER CASO!  $J(\alpha_i)$

### PROBLEMA DUAL (CASO SEPARABLE):

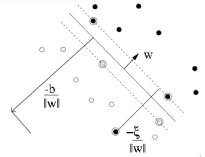
$$\begin{aligned} \frac{\partial}{\partial b} L(w, b, \alpha_i) &= 0 \\ \frac{\partial}{\partial w} L(w, b, \alpha_i) &= 0 \end{aligned} \quad \left\{ \begin{aligned} \max_{\alpha_i} J(\alpha_i) &= -\frac{1}{2} \sum_{i=1}^{NT} \sum_{j=1}^{NT} y_i y_j \alpha_i \alpha_j \tilde{a}_i \cdot \tilde{a}_j + \sum_{i=1}^{NT} \alpha_i \\ \text{sujeto a: } \sum_{i=1}^{NT} y_i \alpha_i &= 0 \\ \alpha_i &\geq 0 \quad i = 1, \dots, NT \end{aligned} \right.$$

Parámetros:  $w = \sum_{i=1}^{NT} \alpha_i y_i \tilde{a}_i$ ,  $b = y_k - w \cdot \tilde{a}_k$  Para algún  $\alpha_k > 0$  ( $\tilde{a}_k$ : support vector)

## SVM – Support Vector Machine

### PROBLEMA PRIMAL (CASO NO SEPARABLE):

$$\begin{aligned} \min_{w, b, \xi_i} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{NT} \xi_i \\ \text{sujeto a: } & y_i (w \cdot \tilde{a}_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, NT \\ & \xi_i \geq 0 \end{aligned}$$



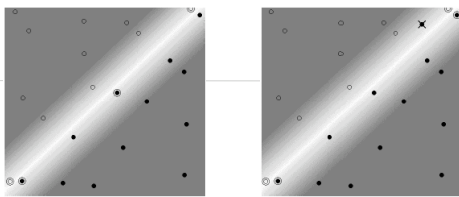
### PROBLEMA DUAL (CASO NO SEPARABLE):

$$\begin{aligned} \max_{\alpha_i} J(\alpha_i) &= -\frac{1}{2} \sum_{i=1}^{NT} \sum_{j=1}^{NT} y_i y_j \alpha_i \alpha_j \tilde{a}_i \cdot \tilde{a}_j + \sum_{i=1}^{NT} \alpha_i \\ \text{sujeto a: } \sum_{i=1}^{NT} y_i \alpha_i &= 0 \\ 0 \leq \alpha_i &\leq C \quad i = 1, \dots, NT \end{aligned}$$

Parámetros:  $w = \sum_{i=1}^{NT} \alpha_i y_i \tilde{a}_i$ ,  $b = y_k - w \cdot \tilde{a}_k$  Para algún  $\alpha_k > 0$  ( $\tilde{a}_k$ : support vector)

## SVM – Support Vector Machine

### - Ejemplo de Clasificador SVM Lineal :



CASO SEPARABLE

CASO NO SEPARABLE

¿ Caso No-Lineal ?

## SVM – Support Vector Machine

### “KERNEL TRICK”:

$$x \cdot y \rightarrow K(x, y)$$

· Si  $K(x, y)$  satisface las condiciones de Mercer,

$$\int g(x) K(x, y) g(y) dx dy \geq 0 \quad \forall g(x) \in L^2$$

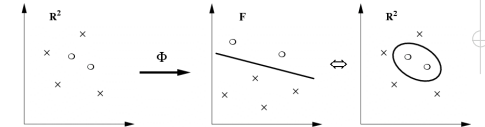
existe la expansión :

$$\begin{aligned} K(x, y) &= \sum \phi_i(x) \phi_i(y) \\ &= \tilde{\phi}(x) \cdot \tilde{\phi}(y) \end{aligned}$$

### Ejemplos:

$$\begin{aligned} K(x, y) &= (x \cdot y + 1)^d \\ K(x, y) &= e^{-\frac{\|x - y\|^2}{2\sigma^2}} \quad (\text{RBF}) \\ K(x, y) &= \tanh(k x \cdot y - c) \end{aligned}$$

· Se está haciendo un producto interno de los vectores mapeados a un espacio F usando la función:  $\Phi : R^N \rightarrow F$



## SVM – Support Vector Machine

### Clasificador SVM No-Lineal :

$$\begin{aligned} \max_{\alpha_i} J(\alpha_i) &= -\frac{1}{2} \sum_{i=1}^{NT} \sum_{j=1}^{NT} y_i y_j \alpha_i \alpha_j K(\tilde{a}_i, \tilde{a}_j) + \sum_{i=1}^{NT} \alpha_i \\ \text{sujeto a: } \sum_{i=1}^{NT} y_i \alpha_i &= 0 \\ 0 \leq \alpha_i &\leq C \quad i = 1, \dots, NT \end{aligned}$$

Parámetros:  $w = \sum_{i=1}^{NT} \alpha_i y_i \tilde{\phi}(\tilde{a}_i)$  NO SE CALCULA

$$b = y_k - \sum_{i=1}^{NT} \alpha_i y_i K(\tilde{a}_i, \tilde{a}_k) \quad \text{Para algún } k \text{ tal que } \alpha_k > 0 \quad (\tilde{a}_k : \text{support vector})$$

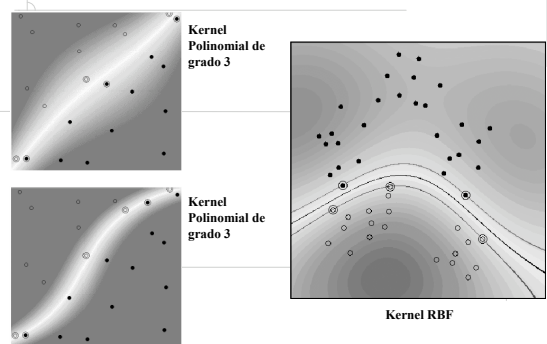
· Y el valor entregado por el clasificador es:

$$S(\tilde{a}) = \sum_{i=1}^{NT} y_i K(\tilde{a}_i, \tilde{a}) + b$$

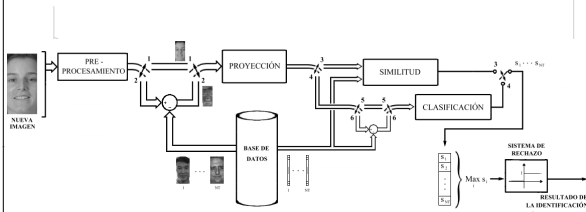
$\tilde{a}$  : nueva diferencia proyectada (o proyección de la diferencia en caso post-diferencial)  
 $\tilde{a}_i$  : diferencias proyectadas (o viceversa)  
 $y_i$  : etiquetas (+1: intraclasses, -1: extraclasses)  
 $i$  recorre solo los vectores de la base de datos para los cuales  $\alpha_i > 0$  (Support Vectors)

## SVM – Support Vector Machine

### - Ejemplo de Clasificador SVM No-Lineal :



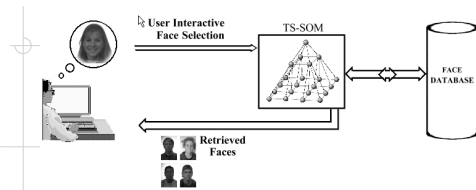
## Esquema Unificado



### - Sistemas de Reconocimiento :

- **Estándar** : switches en 1 y 3.
- **Pre-diferencial** : switches en 2, 4 y 5.
- **Post-diferencial** : switches en 1, 4 y 6.

## Busqueda de Caras en Bases de Datos



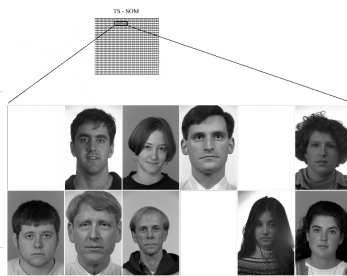
### - Sistemas de Búsqueda :

- **Selección** : El usuario selecciona caras parecidas a una cara que tiene en mente, dentro de una lista mostrada por el sistema.
- **Retorno** : El sistema entrega una nueva lista, de caras parecidas a las caras seleccionadas por el usuario.
- **Iteración** : El proceso se repite hasta que aparece la cara buscada por el usuario.

## Busqueda de Caras en Bases de Datos

### - Ordenamiento de Caras por Red SOM y proyecciones PCA :

- Entrenando una red SOM usando proyecciones PCA se ordenan las caras por parecido.
- Para que sea cierto, PCA debe definir parecidos de la misma forma que las personas.



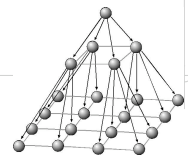
## Busqueda de Caras en Bases de Datos

### - Ordenamiento de Caras por Red SOM :

- Caras se ubican en una Red SOM 2D
- Ubicaciones de las caras seleccionadas por el usuario generan una matriz de selección.

### - Búsqueda Rápida en Red SOM :

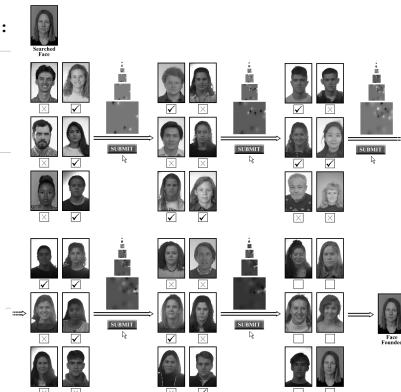
- Se usa una red TS-SOM (Tree-Structured SOM)
- Cada red se entrena usando información de niveles superiores (más rápido).
- Ubicación de caras en nodos inferiores  $O(\log N)$ .



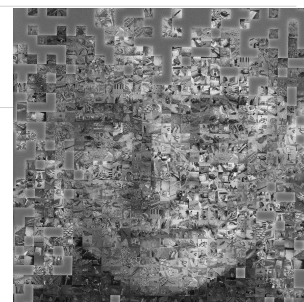
## Busqueda de Caras en Bases de Datos

### - Simulación :

- Base de Datos: FERET
- Se busca una cara dentro de 684 caras.
- Sistema converge en 5 iteraciones.



F I N



Procesamiento Avanzado de Imágenes

EL708