

Eigenfaces

Anthony Huertas

Universidad Nacional de Ingeniería

Aplicación de Machine Learning, 2016

Outline

1 Introducción

- Conceptos Básicos
- Aprendizaje Automatizado
- Reconocimiento Facial

2 Reconocimiento Facial usando Eigenfaces

- Algoritmo de PCA

3 Eigenfaces

- Imágenes faciales
- Eigenfaces

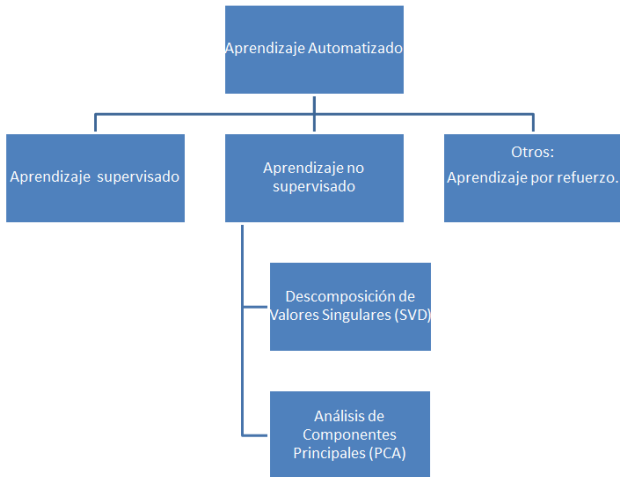
4 Reconocimiento Facial

- Clasificación de las Imágenes Faciales.
- Modelo del Algoritmo

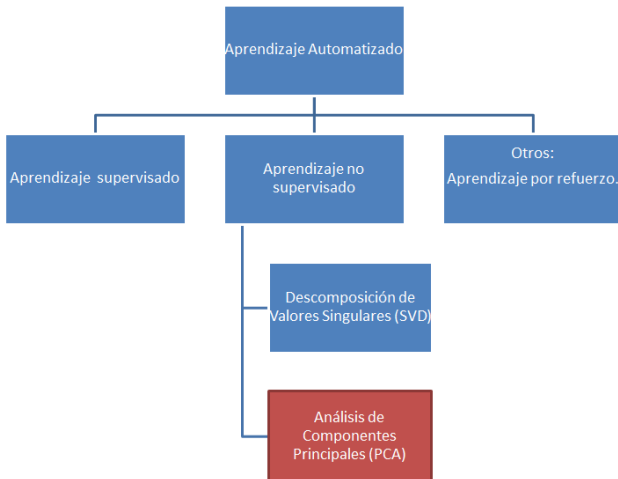
Matemática - Programación

- 1 Álgebra Lineal
- 2 Estadística y Probabilidades
- 3 Optimización
- 4 Programación

Tipos de Aprendizaje Automatizado



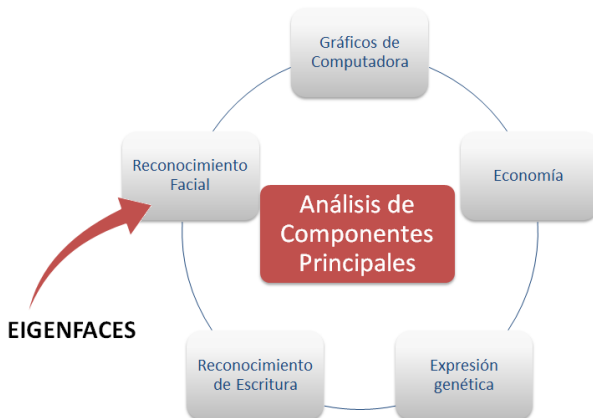
Aprendizaje No Supervisado



Análisis de Componentes Principales (Aplicaciones)



Análisis de Componentes Principales (Aplicaciones)



Usos

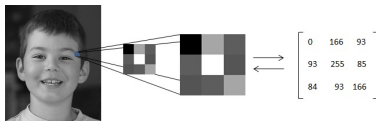


- Criminología
- Redes Sociales
- Implementación para secciones de una imagen facial:
ojos, boca, ...
- Huella digital

Modelo.

- ➊ **INPUT:** ϕ_1, \dots, ϕ_K (K datos n dimensionales).
- ➋ Normalización de la media. $\Phi_i = \phi_i - \bar{\phi}$
- ➌ SVD aplicado a los datos normalizados
 - Eigenvalores de la matriz de covarianza de los datos (Varianza explicada)
 - Eigenvectores de la matriz de covarianza de los datos
- ➍ Determinación del número de componentes principales
- ➎ Determinación de los componentes principales u_1, \dots, u_m (eigenvectores que mantienen mayor varianza de los datos).
- ➏ Patrones de clasificación:

$$[\Theta_1 \quad \dots \quad \Theta_K] = \begin{bmatrix} u_1^T \\ \vdots \\ u_m^T \end{bmatrix} [\Phi_1 \quad \dots \quad \Phi_K].$$

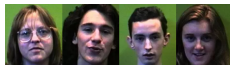


- Una imagen facial de tamaño $m \times n$ esta estructurada como una matriz de píxeles $n \times m$ dimensional o un vector de igual dimensión

$$\begin{matrix} & \text{matriz} \\ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ a_{m1} & \dots & \dots & a_{mn} \end{bmatrix} & \end{matrix} \quad n \times m$$

$$\begin{matrix} & \text{vector} \\ \equiv & \{a_{11}, \dots, a_{1m}, a_{21}, \dots, a_{nm}\} \end{matrix}$$

Se tomarán K imágenes faciales y, para un mejor lenguaje matemático, denominaremos “imagen facial” haciendo referencia a su matriz de píxeles correspondiente.



Código Python Imagen→ Escala de grises, ajuste de tamaño, matriz de píxeles

```
from PIL import Image
import numpy as np
def gray_size_pixel(imagen,pix_ancho=180,pix_largo=200):
    img = Image.open(imagen).convert('L')
    img = img.resize((pix_ancho,pix_largo), Image.ANTIALIAS)
    a=list(np.asarray(img))
    a=np.array(a)
    return img, a
```

Definición. (Media Facial)

Sean $\phi_1, \phi_2, \dots, \phi_K$ imágenes faciales, definimos la media facial Ψ , como

$$\Psi = \frac{1}{K} \sum_{i=1}^K \phi_i \quad (1)$$



Definición. (Contraste Facial)

Sean Φ_1, \dots, Φ_K imágenes faciales, con media facial Ψ , definimos los contrastes faciales como

$$\Phi_i = \phi_i - \Psi. \quad (2)$$



Definición. (Eigenfaces)

Se denominan eigenfaces a los eigenvectores, de la matriz de covarianza de las imagenes faciales ϕ_1, \dots, ϕ_K , que generan el subespacio de dimensión reducida usando el análisis de componentes principales.



Definición. (Face-space)

Se denomina face-space al subespacio generado por los eigenfaces.

Matriz de Covarianza.

$$C = \frac{1}{K} \begin{bmatrix} \Phi_1 & \Phi_2 & \dots & \Phi_K \end{bmatrix}_{(n \times m) \times K} \begin{bmatrix} \Phi_1^T \\ \Phi_2^T \\ \vdots \\ \Phi_K^T \end{bmatrix}_{K \times (n \times m)}$$

cuya dimensión es $(n \times m)^2$.

Matriz de Covarianza.

Teorema

Si v es un eigenvector, con valor propio $\lambda \neq 0$, de $A^T A$, luego Av es un eigenvector con el mismo valor propio λ de AA^T .

$$D = \frac{1}{K} \begin{bmatrix} \Phi_1^T \\ \Phi_2^T \\ \vdots \\ \Phi_K \end{bmatrix}_{K \times (n \times m)} \begin{bmatrix} \Phi_1 & \Phi_2 & \dots & \Phi_K \end{bmatrix}_{(n \times m) \times K}$$

Por tanto siendo $v_1, v_2, \dots, v_{K'} (K' \leq K)$ eigenvectores de D , asociados a eigenvalores $\lambda_1, \dots, \lambda_{K'}$ distintos de cero, los eigenvectores de C asociados a los mismo autovalores distintos de cero son $\{Av_1, Av_2, \dots, Av_{K'}\}$ siendo $A = [\Phi_1, \dots, \Phi_K]$. Denotemos $u_i = Av_i$

Pérdida de Variabilidad

$$\sum_{i=1}^{K'} \lambda_i \quad (\text{Varianza total en } R^{K'}) \quad (3)$$

$$\sum_{i=1}^{n \times m} \lambda_i = \sum_{i=1}^{K'} \lambda_i \quad (\text{Varianza total en } R^{n \times m}) \quad (4)$$

Por tanto el porcentaje de varianza que se mantiene en el subespacio de dimensión $R^{K'}$ es

$$\frac{\text{Varianza total en } R^{K'}}{\text{Varianza total en } R^{n \times m}} \times 100\% = 100\%. \quad (5)$$

¿Podríamos entonces decir que los eigenvectores $\{u_1, u_2, \dots, u_{K'}\}$ son los eigenfaces?

En la práctica realizada por Sirivich y Kirby (1987), se observó que tomando los eigenvectores u_1, u_2, \dots, u_{115} de la matriz de covarianza de sus datos que mantengan el 100% de variabilidad, solo los primeros 40 eigenvectores fueron suficientes para una excelente representación pues se generaba una pérdida del 2% de variabilidad. Por este hecho, es útil el PCA para determinar el número de componentes principales que no genere pérdida de variabilidad significativa, digamos no más del 2%

Sea ϕ imagen de entrada, ésta puede ser clasificada como

- 1 **Reconocimiento de una imagen facial:**
- 2 **No reconocimiento de una imagen facial:**
- 3 **Imagen no facial**

De la imagen de entrada.

Normalizando: $\phi = \phi - \psi$. **Dato transformado:** $\Theta =$

$$\begin{bmatrix} u_1^T \phi \\ u_2^T \phi \\ \vdots \\ u_{K'}^T \phi \end{bmatrix}$$

De las K imágenes faciales.

Patrones de

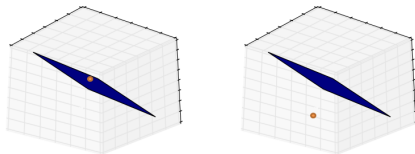
clasificación: $[\Theta_1 \quad \dots \quad \Theta_K] = \begin{bmatrix} u_1^T \\ \vdots \\ u_m^T \end{bmatrix} [\Phi_1 \quad \dots \quad \Phi_K]$

Imagen facial- No facial

Nuestra verificación inicial es determinar si la imagen de entrada es facial o no lo es, esto es si el face-space la clasifica como tal

$$\varepsilon^2 = \|\Phi - \text{proj}_F \Phi\|^2 \leq \delta$$

donde $\text{proj}_F(\Phi) = \sum_{i=1}^{K'} (u_i \cdot \Phi) u_i$.



(a) El subespacio clasifica el dato. (b) El subespacio no clasifica el dato

Figure: Ejemplo de un subespacio bidimensional, caracterizando un dato tridimensional.

Reconocimiento Facial

determinaremos el patron Θ_i , que mejor clasifique a Θ , usando el mismo análisis anterior:

$$\epsilon^2 = \min\{\epsilon_i^2 : \|\Theta - \Theta_i\|^2 = \epsilon_i^2, i = 1, \dots, K'\}.$$

Dado este hecho, bajo medios empíricos, también se recomienda establecer un límite δ a esta cantidad.

$$\epsilon^2 \leq \delta$$

Por tanto si $\epsilon^2 \leq \delta$, el patrón Φ_i estableciéndose lo que denominamos **reconocimiento facial**, y por tanto la imagen facial ϕ matendría una alta relación con la imagen ϕ_i , debido a que se pierde muy poca variabilidad entre ellas.

En caso contrario no se establece un reconocimiento facial.

Ajuste del límite para la clasificación

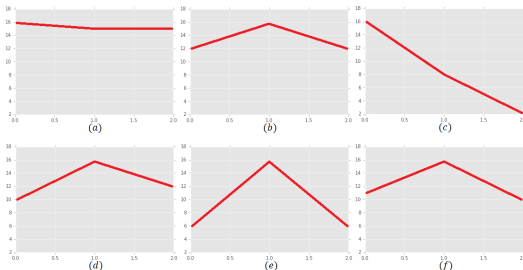


Figure: X: δ ; Y: Número de clasificaciones correctas. Tomándose 16 imágenes faciales, de las K imágenes faciales de entradas, pero variando su: (a) iluminación - (b) inclinación de su rostro - (c) tamaño de cabeza - (d) iluminación e inclinación - (e) iluminación y tamaño de cabeza - (f) inclinación y tamaño de cabeza.

Entrenamiento

- Muestra de K de imágenes faciales, como datos de entrenamiento para el aprendizaje
- Aplicar PCA para obtener los eigenvectores de la matriz D determinan no más del 2% de pérdida de variabilidad. Estos eigenvectores serán llamados *eigenfaces*.
- Determinar los patrones de clasificación transformados.

Tarea de reconocimiento

- Imagen de entrada
- Dado un límite δ , verificar si se cumple:

$$\epsilon^2 = \|\Phi - \text{proy}_F \Phi\|^2 \leq \delta$$

donde $\text{proy}_F(\Phi) = \sum_{i=1}^{K'} (u_i \cdot \Phi) u_i$.







- De cumplirse lo anterior se clasifica como una imagen facial, en caso contrario se clasifica como imagen no facial.
- Dado $\delta > 0$, verificar si se cumple:

$$\epsilon^2 = \min\{\epsilon_i^2 : \|\Theta - \Theta_i\|^2 = \epsilon_i^2, i = 1, \dots, K'\} \leq \delta.$$

- De cumplirse lo anterior se determina un reconocimiento facial y el patrón Θ_i que la caracterize, indicará la alta relación entre ϕ con la imagen facial ϕ_i , a causa de la poca pérdida de variabilidad. De no cumplirse no existe un reconocimiento facial.

Tareas adicionales

- De clasificar la imagen de entrada como imagen facial, estableciéndose o no el reconocimiento facial, entonces implementarla en el modelo actualizando el conjunto de las K imágenes faciales de entrenamiento.
- De clasificar la imagen de entrada como imagen facial, no estableciéndose el reconocimiento facial, entonces adquirir su dato transformado e implementar en la lista de patron de clasificación.

-  Cristopher, M. Bishop. Pattern Recognition and Machine Learning. Cambridge University, 2006.
-  Alex Smola and S.V.N. Vishwanathan. Introduction to Machine Learning. Cambridge University, 2008.
-  Arian Maleki and Tom Do. Review of Probability Theory. Stanford University,
-  Trevor Hastie, Robert Tibshirani, Jerome Friedman. The Elements of Statistical Learning. *Data Mining, Inference and Prediction*. Stanford University, 2008.
-  Arian Maleki and Tom Do. Probability Theory Review. Stanford, 2014.
-  Anderson, Sweeney y Williams. Estadística para Administración y Economía.



Ehran Cinlar. Introduction to Stochastic processes.
Princeton University, 2013.



Ehran Cinlar. Introduction to Stochastic processes.
Princeton University, 2013.