

Trabajo Práctico 2

Reconocimiento de caras

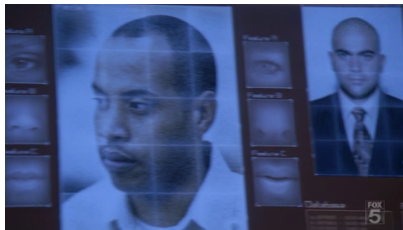
Métodos Numéricos

Primer cuatrimestre - 2018



Trabajo Práctico 2

Reconocimiento de caras - Aplicaciones



Trabajo Práctico 2

Reconocimiento de caras usando reducción de la dimensionalidad

- Datos: base de datos etiquetada de imágenes de caras de personas tomadas de una forma particular.
- Objetivo: dada una nueva imagen de una persona, ¿A cuál corresponde?

Sujeto 1:



Sujeto 2:



Sujeto 3:



Llegan nuevas imágenes, ¿a quiénes pertenecen?

Reconocimiento de caras

Contexto

Objetivo

Desarrollar (no solo en términos de implementación) un *clasificador* que permita reconocer a que persona de una base corresponde una cara.

Contexto

- ▶ Disponemos de una base de datos etiquetada que tendremos que utilizar tanto para entrenamiento como para testeo.
- ▶ Consideramos la base *ORL faces*. 41 sujetos, 10 imágenes por sujeto.
- ▶ Cada cara es una imagen en escala de grises de 92×112 . También disponemos de la misma base pero con imágenes reducidas.

Reconocimiento de caras

Vecino más cercano

Idea general (caso particular reconocimiento de caras)

- ▶ Consideramos cada imagen como un vector $x_i \in \mathbb{R}^m$, $m = 92 \times 112$, $i = 1, \dots, n$. Para las imágenes en la base de datos, sabemos además a que clase pertenece.
- ▶ Cuando llega una nueva imagen de una cara z , con el mismo formato, recorremos toda la base y buscamos aquella que minimice

$$\arg \min_{i=1, \dots, n} \|z - x_i\|_2$$

Luego, le asignamos la clase del representante seleccionado.

Generalización

Considerar más de un vecino.

Análisis de Componentes Principales

Resumen hasta acá

- ▶ Tenemos n muestras de m variables.
- ▶ Calculamos el vector μ que contiene la media de cada una de las variables.
- ▶ Construimos la matriz $X \in \mathbb{R}^{n \times m}$ donde cada muestra corresponde a una fila de X y tienen media cero (i.e., $x^{(i)} := (x^{(i)} - \mu) / \sqrt{n-1}$).
- ▶ Diagonalizamos la matriz de covarianzas M_X . La matriz V (ortogonal) contiene los autovectores de M_X .

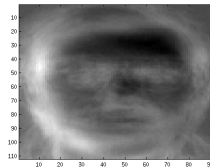
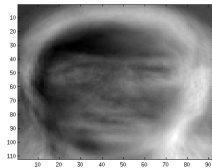
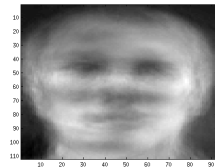
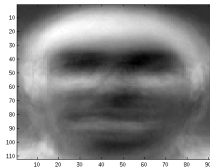
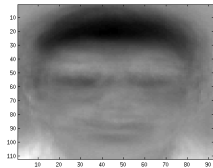
Propiedades del cambio de base

- ▶ Disminuye redundancias.
- ▶ El cambio de base $\hat{X}^t = PX^t = V^t X^t$ asigna a cada muestra un nuevo *nombre* mediante un cambio de coordenadas.
- ▶ Las columnas de V (autovectores de M_X) son las componentes principales de los datos.
- ▶ En caso de m grande, es posible tomar sólo un subconjunto de las componentes principales para estudiar (i.e., aquellas que capturen mayor proporción de la varianza de los datos).

Reconocimiento de caras

Autocaras (Eigenfaces)

Gráfico de los primeros 5 autovectores en V para la base ORL Faces.



Reconocimiento de caras

¿Cómo reconocemos una cara?

Idea

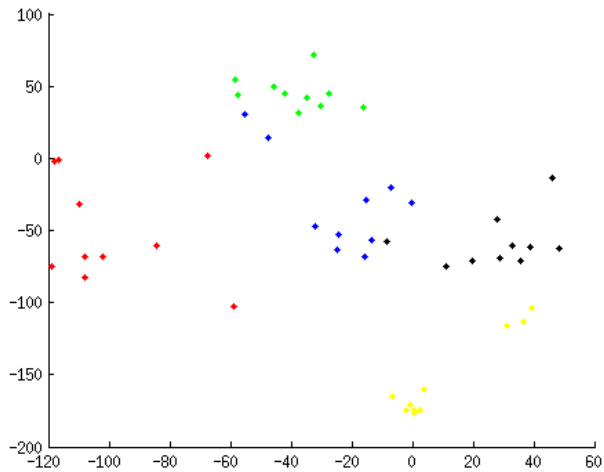
- ▶ Utilizar el cambio de base, transformando cada imagen convenientemente.
- ▶ Reducir la dimensión de los datos utilizando sólo algunas de las nuevas variables (eligiendo aquellas que capturan una fracción mayor de la varianza).

Procedimiento

- ▶ Reducción de la dimensión: parámetro de entrada que indica cuántas componentes principales considerar, α . Es decir, tomaremos $\bar{V} = [v_1 \ v_2 \ \dots \ v_\alpha]$.
- ▶ Transformación característica: Aplicamos el cambio de base a cada muestra $x^{(i)}$, definimos $tc(x^{(i)}) = \bar{V}^t x^{(i)} = (v_1^t x^{(i)}, \dots, v_\alpha^t x^{(i)})$.

Reconocimiento de caras

Reducción + Transformación ($k = 2$)



Reconocimiento de caras

¿Cómo reconocemos una cara?

Finalmente, dada una imagen de una cara que no se encuentra en la base:

- ▶ Vectorizamos la imagen en $x^* \in \mathbb{R}^m$.
- ▶ Definimos $\bar{x}^* = (x^* - \mu) / \sqrt{n-1}$.
- ▶ Aplicamos la transformación característica, $tc(\bar{x}^*)$ y buscamos (de alguna forma) a que sujeto pertenece.

Reconocimiento de caras

Metodología de evaluación

Como evaluamos si el método funciona?

- ▶ Como medimos la efectividad del método?

Reconocimiento de caras

Metodología de evaluación

Como evaluamos si el método funciona?

- ▶ Como medimos la efectividad del método?
- ▶ Tiene sentido probarlo sobre la base de training?

Reconocimiento de caras

Metodología de evaluación

Como evaluamos si el método funciona?

- ▶ Como medimos la efectividad del método?
- ▶ Tiene sentido probarlo sobre la base de training?
- ▶ De alguna forma defino una instancia, pruebo todas las combinaciones de parámetros sobre la misma. Es correcto? Puede surgir algún problema?

Reconocimiento de caras

Metodología de evaluación

Como evaluamos si el método funciona?

- ▶ Como medimos la efectividad del método?
- ▶ Tiene sentido probarlo sobre la base de training?
- ▶ De alguna forma defino una instancia, pruebo todas las combinaciones de parámetros sobre la misma. Es correcto? Puede surgir algún problema?

Idea

Utilizar la base de entrenamiento convenientemente para estimar y proveer suficiente evidencia respecto a la efectividad del método.

Algunas dificultades más

Ojo con $X^t X$.

Si la resolución de las imágenes es de 112×92 y hay ≈ 500 imágenes en la base:

- ▶ ¿Cuál es el tamaño de $X^t X$?

Alternativas

- ▶ Trabajar con imágenes más chicas (por ejemplo, 28×23 , un 25 % del tamaño original).
- ▶ Dada la matriz XX^t , relacionar los autovectores y autovalores de dicha matriz con la original.

¿Qué hay que hacer en el TP?

- ▶ Implementar el método de Análisis de Componentes Principales y el método de la potencia con deflacin.
- ▶ Experimentar con al menos dos métodos (con y sin PCA) para que, dada una nueva imagen de una cara, determine a que persona de la base de datos corresponde utilizando la transformación característica.
- ▶ Experimentar variando: k , cantidad de imagenes por cada persona en la base, *resolución de las imágenes en la base de datos*. Analizar los resultados en términos de las métricas presentadas sobre un conjunto de casos de prueba.

Fecha de entrega

- ▶ Formato electrónico: Viernes 25 de Mayo de 2018, hasta las 23:59 hs., enviando el trabajo (informe+código) a `metnum.lab@gmail.com`.
- ▶ Formato físico: Lunes 28 de Mayo de 2018 a las 18 hs. en la clase de laboratorio.