



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

TP 2 - Tu cara me suena...

MÉTODOS NUMÉRICOS

Resumen

En este trabajo práctico, desarrollamos un método para el reconocimiento de caras basado en el enfoque de *eigenfaces*. Para ello, manipulamos la base de imágenes de entrenamiento matricialmente; obteniendo las componentes principales (autovectores) de su matriz de covarianzas a través del método de la potencia con deflación. Exploramos algunos métodos de clasificación, su tasa de efectividad y su relación con posibles variaciones en el conjunto inicial de entrenamiento.

Keywords: *Face Recognition, Componentes Principales, SVD*

Grupo 2

Integrante	LU	Correo electrónico
Franco Bartalotta		franco.bartalotta@hotmail.com
Fernando Gasperi Jabalera	56/09	fgasperijabalera@gmail.com
Ana Sarriés	144/02	abarloventos@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.exactas.uba.ar>

Índice

1. Introducción Teórica	3
2. Desarrollo	4
2.1. La implementación y algunas decisiones de diseño	4
2.2. Elección conveniente de la matriz de covarianzas para imágenes grandes	5
2.3. Comparación de la complejidad de los métodos	5
2.4. Algoritmos	7
2.5. Tests	7
2.5.1. Cantidad de componentes principales vs tasa de efectividad	7
2.5.2. Análisis de la tasa de eficiencia en función de la cantidad de personas	7
2.5.3. nimgp vs Tasa de efectividad	7
3. Resultados	8
3.1. Test de tasa de eficiencia en función de la cantidad de personas	8
3.2. Test de tasa de eficiencia en función de la cantidad de componentes principales consideradas	9
3.3. Test de tasa de eficiencia en función de nimgp	9
3.4. Test de tasa de eficiencia en función de la cantidad de componentes principales consideradas	10
4. Discusión	11
5. Conclusiones	12
6. Apéndices	13
6.1. Descripción formal del enfoque de <i>Eigenfaces for Recognition</i>	14
6.2. Enunciado de la Cátedra	15
Referencias	18

1. Introducción Teórica

El objetivo inicial de este trabajo es implementar un método que nos permita identificar personas casi en tiempo real. Para ello, nos basamos en el enfoque propuesto en “Eigenfaces for recognition” [1].

Según dicho enfoque, partimos de una base de datos de fotografías (conjunto inicial de entrenamiento), cada una asociada a un sujeto en particular y el objetivo es recibir una foto, que puede o no pertenecer a la base original, e identificar a qué sujeto pertenece. Aplicando el análisis de componentes principales, conocido como PCA por sus siglas en inglés (Principal Component Analysis), obtenemos una transformación de ese espacio inicial que permite reducir su dimensionalidad. A partir de entonces, pasamos a manipular tanto las imágenes de entrenamiento como las incógnitas, aplicándoles dicha transformación característica. De manera que los métodos de identificación (o clasificación) que desarrollamos operarán en ese espacio de dimensión reducida.

Para una descripción detallada del argumento teórico del método, ver 6.1

2. Desarrollo

2.1. La implementación y algunas decisiones de diseño

La implementación consta de 2 fases. Primero está la fase de entrenamiento:

1. se obtiene la matriz $I \in \mathbb{R}^{n \times m}$ a partir de todas las imágenes de la base de entrenamiento. Para $i = 1, \dots, n$, llamamos x_i a la i -ésima fila de I que se corresponde con la i -ésima imagen de nuestra base vectorizada por filas. Cada posición de x_i se corresponde con un pixel de la imagen a la cual corresponde. Además, sabemos que $n = \text{nimgp} * \#\text{sujetos}$, siendo nimgp la cantidad de imágenes que tengo por cada sujeto y $\#\text{sujetos}$ la cantidad de sujetos distintos a los que pertenecen las imágenes de mi base de entrenamiento, y $m = \text{alto} * \text{ancho}$, siendo alto y ancho los respectivos anchos y altos de las imágenes medidos en pixeles.
2. se calcular la matriz A a partir de la matriz I . Sus dimensiones son iguales. Obtenemos el vector $\mu = (x_1 + \dots + x_n)/n$ que es el promedio de las imágenes. Para calcular A , en primer lugar restamos a cada una de las filas de I el vector μ . Obtenemos I' que contiene en la i -ésima fila al vector $(x_i - \mu)^t$. En segundo lugar obtenemos A como: $A = I' * \sqrt{n-1}$.
3. calculamos la matriz de covarianzas $M_x = A^t * A$. (Para este punto y el siguiente efectuamos un análisis alternativo que se puede ver en 2.2)
4. aplicamos el método de la potencia con deflación k veces sobre la matriz M_x . k es el número de componentes principales que nos interesa obtener. Conseguimos una matriz $B \in \mathbb{R}^{m \times k}$ cuya columna b_i se corresponde con el autovector a_i de la matriz M_x asociado al autovalor λ_i tal que $\lambda_1 < \lambda_2 < \dots < \lambda_k \leq \lambda_{k+1} \leq \dots \leq \lambda_m$. Esto es así por que el método de la potencia, si encuentra un autovalor, es el de mayor módulo. [?, 1].
5. aplicamos la transformación característica a todas las imágenes de nuestra base de entrenamiento. La transformación característica de x_i está dada por

$$tc(x_i) = (b_1^t x_i, b_2^t x_i, \dots, b_k^t x_i) \in \mathbb{R}^k$$

y luego está la fase de identificación en la que recibimos una nueva imagen e identificamos a qué sujeto pertenece:

1. recibimos la imagen y la vectorizamos por filas de la misma forma que lo hicimos con las imágenes en nuestra base de entrenamiento en la primera fase y obtenemos $x_{\text{identificar}}$.
2. calculamos su transformación característica de la misma manera que lo hicimos en la fase de entrenamiento y con la misma matriz B

$$tc(x_{\text{identificar}}) = (b_1^t x_{\text{identificar}}, b_2^t x_{\text{identificar}}, \dots, b_k^t x_{\text{identificar}}) \in \mathbb{R}^k$$

3. decidimos a cuál sujeto pertenece por alguno de los dos métodos que describimos más adelante.

Los métodos que implementamos para decidir a qué sujeto pertenece una imagen luego de haber calculado su transformación característica son los siguientes:

Distancia mínima calculamos la distancia entre la transformación característica de la imagen que queremos identificar y todas las imágenes de nuestra base. Decidimos que la imagen nueva corresponde al sujeto al cual pertenece la imagen cuya transformación característica es la más cercana a la transformación característica de la nueva imagen. Es decir, con el de distancia mínima.

Distancia mínima al promedio calculamos el promedio de las distancias entre la transformación característica de la imagen que queremos identificar y las imágenes de cada uno de nuestros sujetos. De esta forma obtenemos una distancia promedio a los diferentes sujetos. Finalmente, decidimos que la imagen nueva corresponde al sujeto al cual pertenece el promedio de distancias más cercano a la transformación característica de la nueva imagen.

Elegimos como vector inicial para el método de la potencia $b_0 = (1, \dots, 1)$. La técnica de deflación arrastra el error generado por el método de la potencia, haciendo que los últimos autovalores tengan un error de cálculo mayor que los primeros. Este problema es mitigado por el hecho de que sólo tomamos los k mayores pero de todas formas probablemente quede reflejado en la experimentación.

2.2. Elección conveniente de la matriz de covarianzas para imágenes grandes

El método alternativo consiste en utilizar la matriz $A.A^t$ en lugar de la matriz A^tA a la hora de calcular los autovectores de la transformación característica.

Esto resulta posible, si advertimos que ambas matrices poseen los mismos autovalores. Veamos la demostración:

Sea u_i autovector de $A.A^t$ asociado a λ_i , entonces tenemos que:

$$(A.A^t).u_i = \lambda_i.u_i,$$

multiplicando por A^t a ambos lados de la igualdad, resulta:

$$A^t.(A.A^t).u_i = A^t\lambda_i.u_i,$$

que por propiedad de matrices y escalares, resulta:

$$(A^t.A).(A^t.u_i) = \lambda_i.(A^t.u_i),$$

Y observando esta última igualdad, notamos que llamando v_i a $A^t.u_i$ entonces v_i resulta autovector de $A^t.A$ asociado a λ_i .

De lo anterior, se desprende que todo autovalor de $A.A^t$ también lo es de $A^t.A$ (y tomando $A = A^t$ obtendríamos la vuelta). Este resultado nos permite elegir el producto de matrices que viva en un espacio de menor dimensión, y calcular sobre él, el método de la potencia, sabiendo que hallaremos los mismos autovalores.

Además, la última línea de la demostración también nos indica cuál será el método para calcular el autovector asociado: $v_i = A^t.u_i$.

2.3. Comparación de la complejidad de los métodos

El método 1 trabaja con la matriz AA^t y el método 2 trabaja con la matriz A^tA . La matriz $A \in \mathbb{R}^{n \times m}$ con $m = \text{pixels}(\text{img})$ y $n = \text{nimgp} \times \text{\#personas}$. Por lo tanto como $A^t \in \mathbb{R}^{m \times n}$ nos queda que $AA^t \in \mathbb{R}^{n \times n}$ y $A^tA \in \mathbb{R}^{m \times m}$.

En el único punto que divergen los algoritmos es en el cálculo de los k mayores autovalores, y sus autovectores asociados, de $M_x = A^tA$, la matriz de covarianza. El método 1 aplica el método de la potencia con deflación directamente sobre la matriz M_x . El método 2 primero aplica el método de la potencia sobre la matriz $M_x^t = AA^t$ y después realiza un despeje para obtener los autovalores y autovectores de M_x . Por lo tanto tenemos que los dos métodos utilizan la misma función para calcular autovectores y autovalores: y el método 2 además utiliza una función para despejar cuyo

Algorithm 2.1 calcularAutovaloresYAutovectores(Matriz A)

```

for  $i = 1$  hasta  $k$  do
 $v_i, \lambda_i = \text{metodoDeLaPotencia}(A)$ 
 $A = \text{deflacionar}(A, v_i, \lambda_i)$ 
end for
```

pseudo-código podemos encontrar en la sección algoritmos. Por lo tanto, la complejidad del método 1 es (abusando de la notación asintótica):

$$O(\text{metodo1}) = k(O(\text{metodoDeLaPotencia}) + O(\text{deflacionar}))$$

y del método 2:

$$O(\text{metodo2}) = k(O(\text{metodoDeLaPotencia}) + O(\text{deflacionar})) + kO(\text{despejar})$$

La complejidad del método de la potencia es:

$$O(\text{metodoDeLaPotencia}) = \text{iter}(O(Ax) + O(\frac{y}{\|y\|}))$$

$A \in \mathbb{R}^{n \times n}$ y $x \in \mathbb{R}^n$ entonces:

$$O(Ax) = n(n \times \text{mult} + (n-1) \times \text{sums}) = (n^2) \times \text{mult} + n(n-1) \times \text{sums} = O(n^2)$$

Si $y \in \mathbb{R}^n \Rightarrow \frac{y}{\|y\|} = O(n)$ por lo tanto reemplazando por lo nuevos valores:

$$O(\text{metodoDeLaPotencia}) = \text{iter}(O(n^2) + O(n)) = \text{iter}O(n^2) = O(n^2)$$

porque *iter* es un valor fijo que elegimos de antemano. La complejidad de aplicar deflacionar con $v_i \in \mathbb{R}^n$ y $A \in \mathbb{R}^{n \times n}$ es:

$$O(\text{deflacionar}) = O(A - \lambda v_1 v_1^t) = n^2 \times \text{restas} + O(\lambda_i v_i v_i^t)$$

seguimos reemplazando:

$$n^2 \times \text{restas} + n^2 \times \text{mults} + O(v_i v_i^t) = n^2 \times \text{restas} + n^2 \times \text{mults} + n^2 \times \text{mults}$$

entonces nos queda que:

$$O(\text{deflacionar}) = n^2 \times \text{restas} + 2n^2 \times \text{mults} = O(n^2)$$

Por lo tanto la complejidad del método 1 queda de la siguiente forma:

$$O(\text{metodo1}) = k(O(\text{metodoDeLaPotencia}) + O(\text{deflacionar})) = k(O(n^2) + O(n^2)) = O(n^2)$$

y la complejidad del método 2 es:

$$O(\text{metodo2}) = O(\text{metodo1}) + O(\text{despejar})$$

Y la complejidad de despejar la planteamos como:

$$O(\text{despejar}) = O(\frac{\sqrt{\lambda_i}}{\lambda_i} A^t v)$$

con $v \in \mathbb{R}^n$ y $A \in \mathbb{R}^{n \times n}$ nos queda:

$$n \times \text{mults} + O(A^t v) = n \times \text{mults} + (n(n \times \text{mults} + (n-1) \times \text{sums}))$$

lo que nos deja con:

$$O(n) + nO(n) = O(n^2)$$

reemplazando en la complejidad del método 2 obtenemos que:

$$O(\text{metodo2}) = O(\text{metodo1}) + O(\text{despejar}) = O(n^2) + O(n^2) = O(n^2)$$

Concluimos que el método 1 y el método 2 tienen la misma complejidad $O(n^2)$ tomando a n como el tamaño de la matriz A de entrada. Como habíamos analizado en el comienzo el tamaño de la matriz A que recibe cada uno es diferente. Para el método 1 es $n = \text{pixels}(\text{img})$ y para el método 2 es $n = \text{nimgp} \times \text{\#personas}$. Por lo tanto cuál de los dos cueste menos va a depender de la relación entre la resolución de las imágenes de nuestra base de entrenamiento y la cantidad de imágenes que tenemos en total. En nuestro caso particular la resolución de las imágenes excede ampliamente a la cantidad de imágenes. Nuestra base de entrenamiento tiene 410 fotos y en general usamos una resolución de $112 \times 92 = 10304$. Por lo tanto:

$$\text{nimgp} \times \text{\#personas} \ll \text{pixels}(\text{img}) \Rightarrow O(\text{metodo2}) \ll O(\text{metodo1})$$

Algorithm 2.2 despejar(Matriz A , Vector v , Double λ)

$$u = v \frac{\sqrt{\lambda_i}}{\lambda_i} A^t$$

2.4. Algoritmos

2.5. Tests

En esta sección haremos una introducción a las experimentaciones que decidimos realizar.

2.5.1. Cantidad de componentes principales vs tasa de efectividad

El objetivo de la siguiente experimentación es analizar cómo afecta el número de componentes principales utilizado a la tasa de efectividad que obtenemos al identificar fotos. Cuantas más componentes principales tomamos más información tenemos. Sin embargo, los k calculamos autovalores que tomamos son los k de mayor valor absoluto, es decir, los que más información reflejan. Por lo tanto es de esperar que cada componente principal que agregamos nos aporte menos información que la anterior. Es decir, que la tasa de efectividad aumente junto con la cantidad de componentes principales utilizadas pero a una velocidad sublineal. Además, sabemos que el método de la potencia con deflación arrastra error, esto implica que la exactitud de los autovalores obtenidos es cada vez menor $exactitud(\lambda_i) > exactitud(\lambda_{i+1})$. Para la experimentación vamos a mantener la cantidad de sujetos, 40, y la resolución de las imágenes, 112×92 , constantes. El análisis lo vamos a realizar considerando hasta 45 componentes principales. Los tests se correrán tomando $nimgp = 3, 6, 9$ caras de entrenamiento. Para cada combinación de k y $nimgp$ corremos el test 10 veces, tomando al azar las fotos de la base de entrenamiento y las utilizadas para identificación. Calculamos la tasa de efectividad para los 10 y nos quedamos con el promedio. Vamos a intentar identificar a todas las fotos restantes de la base para poder tener más casos.

2.5.2. Análisis de la tasa de eficiencia en función de la cantidad de personas

En este experimento se intenta analizar como influye la cantidad de personas en la tasa de eficiencia del reconocimiento de imágenes. Para esto vamos a concentrarnos solo en la variable de la cantidad de personas por lo que hay que fijar el resto de los parámetros en sus valores óptimos. La experimentación se va a hacer sobre la base de imágenes grandes, con cinco imágenes por persona (valor generoso pero dentro de los límites reales), y con quince componentes principales y con la técnica de reconocimiento más efectiva. Se espera que cuantas más personas haya (y en consecuencia más imágenes), la probabilidad de que haya imágenes similares de distintas personas o incluso personas similares es más alta, por lo que tasa de reconocimiento va a ser menor.

2.5.3. $nimgp$ vs Tasa de efectividad

En la siguiente experimentación queremos analizar cómo afecta a la efectividad la cantidad de imágenes que tomamos por sujeto, $nimgp$, para la base de entrenamiento a la tasa de efectividad. La base total con la que contamos contiene 41 sujetos y 10 fotos por cada uno de ellos. Es una base que consideramos pequeña por lo cual vamos a intentar de abarcar la mayor cantidad de combinaciones y así cubrir el mayor espectro de casos que podamos al menos dentro de nuestro pequeño universo. ¿Cómo vamos a variar las variables? ¿Qué esperamos obtener? ¿Por qué es interesante realizar este test? ¿Qué desventajas tenemos con la base dada?

3. Resultados

3.1. Test de tasa de eficiencia en función de la cantidad de personas

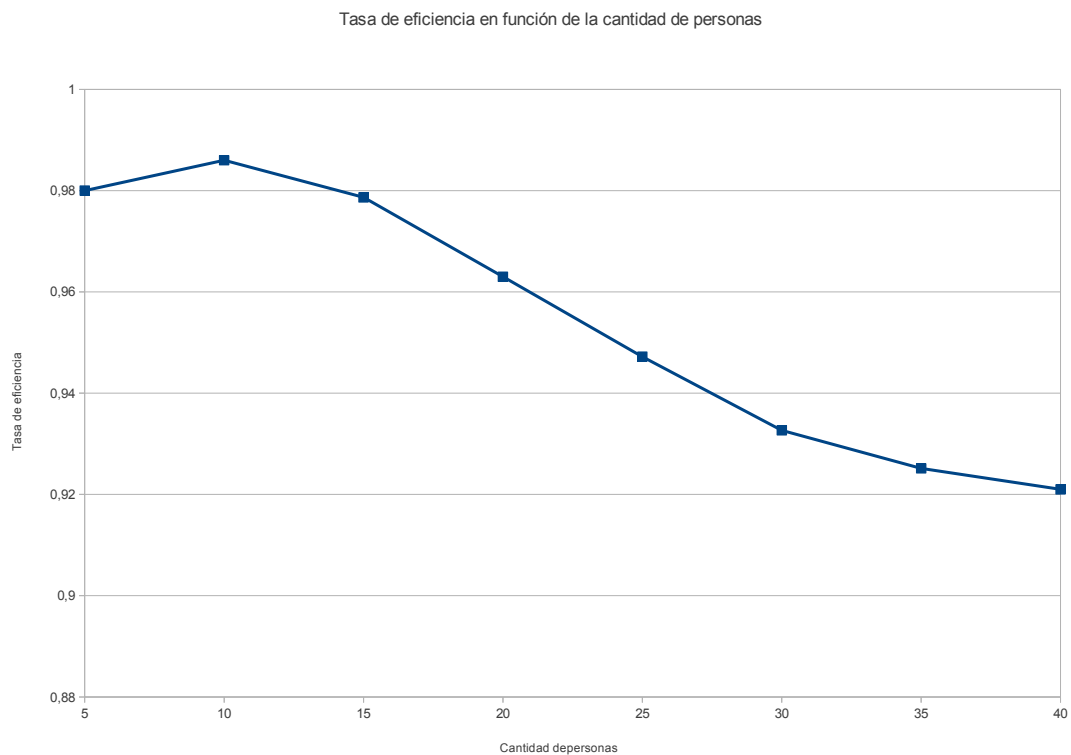


Figura 1: Tasa de eficiencia en función de la cantidad de personas.

3.2. Test de tasa de eficiencia en función de la cantidad de componentes principales consideradas

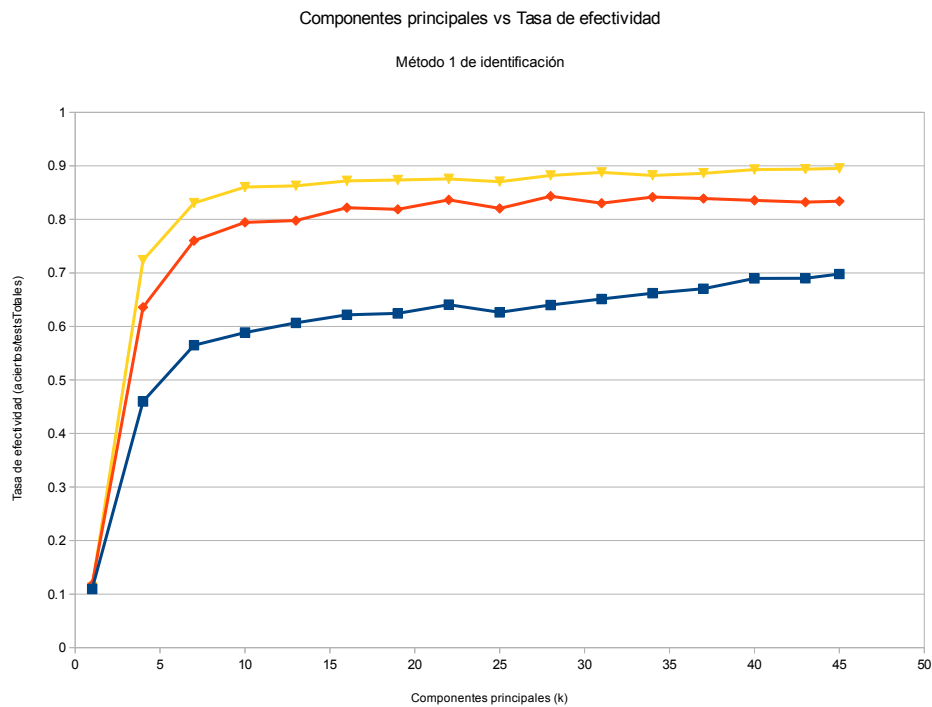


Figura 2: El método 1 de identificación se corresponde con el de Distancia Mínima

3.3. Test de tasa de eficiencia en función de nimgp

graphs/nimgpVsTasaDeEfectividad.pdf

Figura 3: Se graficó para 15, 30 y 45 componentes principales

3.4. Test de tasa de eficiencia en función de la cantidad de componentes principales consideradas

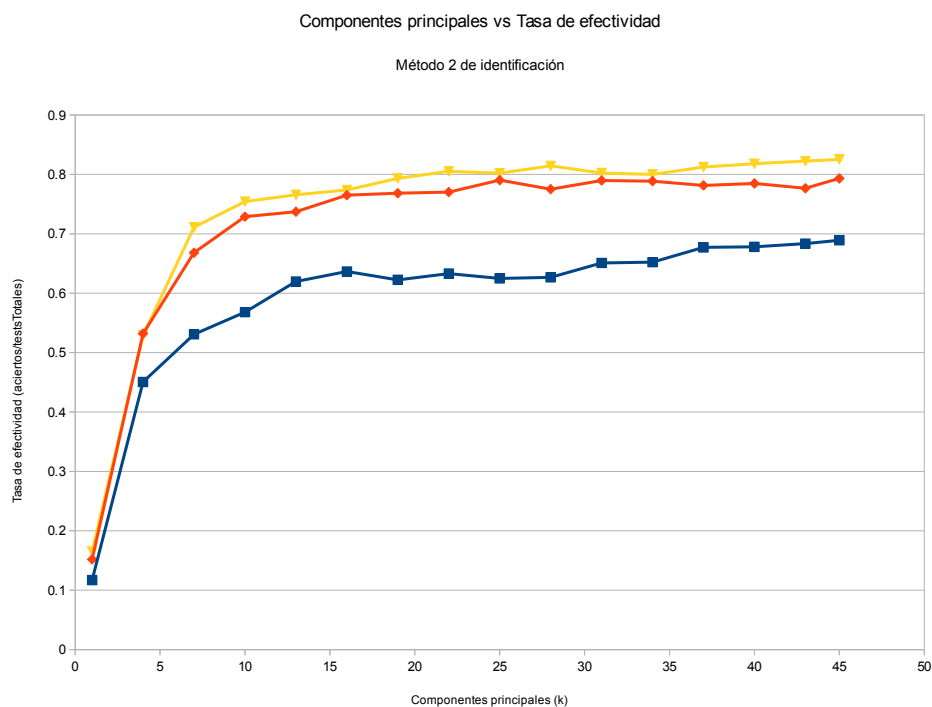


Figura 4: El método 2 de identificación se corresponde con el de Distancia Mínima Promedio

4. Discusión

5. Conclusiones

6. Apéndices

6.1. Descripción formal del enfoque de *Eigenfaces for Recognition*

Formalmente, partimos de n imágenes (almacenada por filas en un vector) que notamos $x_i \in \mathbb{R}^m$ con $i = 1, \dots, n$. A cada uno de nuestros x_i le restamos el vector $\mu = (x_1 + \dots + x_n)/n$ para que tengan media igual cero. De esta forma construimos la matriz X que tiene en la i -ésima fila al vector $(x_i - \mu)^t$. A partir de ésta, podemos construir la siguiente matriz M_x :

$$M_x = \frac{1}{n-1} \hat{X}^t \hat{X}$$

Esta matriz tiene la varianza de cada variable, que en nuestro caso son los pixeles de las imágenes, en la diagonal, y la covarianza entre ellos en las restantes posiciones, por lo que nos referimos a ella como la matriz de covarianzas. Esta matriz resulta ser simétrica y un teorema nos garantiza que toda matriz simétrica es diagonalizable en la forma $M_x = PDP^{-1}$ donde D es una matriz diagonal. Al diagonalizar, estamos buscando variables que tengan covarianza cero entre sí y la mayor varianza posible. Si aplicamos como transformación un cambio de base apropiado:

$$\hat{X}^t = P X^t$$

M_x nos queda:

$$M_x = \frac{1}{n-1} \hat{X}^t \hat{X} = \frac{1}{n-1} (P \hat{X}^t)(P \hat{X}) = P M_x P^t$$

Como M_x es simétrica existe V ortogonal tal que $M_x = V D V^t$:

$$M_x = P M_x P^t = P (V D V^t) P^t = (V^t V) D (V V^t) = D$$

habiendo tomado $P = V^t$. De esta forma vemos como las columnas v_j de la matriz V son los autovectores de M_x . Éstas van a ser las componentes principales de nuestros datos, la base de imágenes de entrenamiento. En la práctica, en caso de que la dimensión de M_x sea muy grande, es posible tomar sólo un subconjunto de las componentes principales, aquellas que capturen mayor proporción de la varianza de los datos.

De esta forma queda definida nuestra transformación característica por la aplicación del cambio de base a cada una de nuestras imágenes:

$$tc(x_i) = \bar{V}^t x_i = (v_1^t x_i, \dots, v_k^t x_i)$$

Ésta misma transformación característica se la aplicaremos a las imágenes que queramos identificar para evaluar a qué sujeto pertenecen.

6.2. Enunciado de la Cátedra

Laboratorio de Métodos Numéricos - Primer Cuatrimestre 2014 Trabajo Práctico Número 2: Tu cara me suena

Introducción

A menos de cincuenta días del inicio de la gran fiesta del fútbol mundial, el Equipo de Desarrollos de Métodos Numéricos ha recibido un pedido reciente por parte del Comité de Seguridad de la Organización del Mundial Brasil 2014. Informaciones de último momento aseguran que una gran cantidad de *barras* se están organizando para *copar* las tribunas de los (aún no terminados) estadios del país organizador. Debido al retraso en la implementación del sistema FIFA Plus por cuestiones técnicas, el evento será cancelado a menos que nuestro equipo logre desarrollar un software de reconocimiento facial efectivo en los próximos 20 días.

El Comité Organizador ha puesto a nuestra disposición todos sus recursos a fin de evitar lo que sería la mayor vergüenza en la historia de la entidad. Entre otras cosas, se dispone de una base de datos de fotografías digitalizada de todas aquellas personas que alguna vez hayan ingresado a algún estadio de fútbol en algún país del planeta. El objetivo del trabajo es desarrollar un software que, tomando esta información como entrenamiento, pueda detectar en tiempo real si los asistentes a alguno de los partidos corresponde a un *barra*.

Como instancias de entrenamiento, se tiene un conjunto de N personas, cada una de ellas con M imágenes (eventualmente) distintas sus rostros en escala de grises del mismo tamaño y resolución. Cada una de estas imágenes sabemos a qué persona corresponde.

Para $i = 1, \dots, n$, sea $x_i \in \mathbb{R}^m$ la i -ésima imagen de nuestra base de datos almacenada por filas en un vector, y sea $\mu = (x_1 + \dots + x_n)/n$ el promedio de las imágenes. Definimos $X \in \mathbb{R}^{n \times m}$ como la matriz que contiene en la i -ésima fila al vector $(x_i - \mu)^t / \sqrt{n-1}$, y

$$A = U \Sigma V^t$$

a su descomposición en valores singulares, con $U \in \mathbb{R}^{n \times n}$ y $V \in \mathbb{R}^{m \times m}$ matrices ortogonales, y $\Sigma \in \mathbb{R}^{n \times m}$ la matriz diagonal conteniendo en la posición (i, i) al i -ésimo valor singular σ_i . Siendo v_i la columna i de V , definimos para $i = 1, \dots, n$ la *transformación característica* del dígito x_i como el vector $\mathbf{tc}(x_i) = (v_1^t x_i, v_2^t x_i, \dots, v_k^t x_i) \in \mathbb{R}^k$, donde $k \in \{1, \dots, m\}$ es un parámetro de la implementación. Este proceso corresponde a extraer las k primeras *componentes principales* de cada imagen. La intención es que $\mathbf{tc}(x_i)$ resuma la información más relevante de la imagen, descartando los detalles o las zonas que no aportan rasgos distintivos.

Dada una nueva imagen x de una cara, que no se encuentra en el conjunto inicial de imágenes de entrenamiento, el problema de reconocimiento consiste en determinar a qué persona de la base de datos corresponde. Para esto, se calcula $\mathbf{tc}(x)$ y se compara con $\mathbf{tc}(x_i)$, para $i = 1, \dots, n$, utilizando un criterio de selección adecuado.

Enunciado

Se pide implementar un programa en C o C++ que lea desde archivos las imágenes de entrenamiento correspondientes a distintas personas y que, utilizando la descomposición en valores singulares y el número de componentes principales k mencionado anteriormente, calcule la transformación característica de acuerdo con la descripción anterior. Se debe proponer e implementar al menos un método que, dada una nueva imagen de una cara, determine a qué persona de la base de datos corresponde utilizando la transformación característica.

Con el objetivo de obtener la descomposición en valores singulares, se deberá implementar

el método de la potencia con deflación para la estimación de autovalores/autovectores. En este contexto, la factibilidad de aplicar este método es particularmente sensible al tamaño de las imágenes de la base de datos. Por ejemplo, considerar imágenes en escala de grises de 100×100 píxeles implicaría trabajar con matrices de tamaño 10000×10000 . Una alternativa es reducir el tamaño de las imágenes, por ejemplo, mediante un submuestreo. Sin embargo, es posible superar esta dificultad en los casos donde el número de muestras es menor que el número de variables. Se pide desarrollar las siguientes sugerencias y fundamentar como utilizarlas en el contexto del trabajo.

- Dada una matriz y su descomposición en valores singulares $A = U\Sigma V^t$, encontrar la descomposición en valores singulares de A^t . Cómo se relacionan los valores singulares de A y A^t ?
- Dada la descomposición en valores singulares de A , expresar en función de U , Σ y V las matrices A^t , $A^t A$ y AA^t . Analizar el tamaño de cada una de ellas y deducir como relacionar las respectivas componentes principales. Combinar con el item anterior para el cómputo de los componentes principales.

En base a este análisis, se pide desarrollar una herramienta alternativa que permita trabajar bajo ciertas condiciones con imágenes de tamaño mediano/grande.

Junto con este enunciado se provee una base de datos de imágenes correspondiente a 41 personas, con 10 imágenes por cada una de ellas. Esta base de datos se encuentra disponible en dos resoluciones distintas: 92×112 y 23×28 píxeles por cada imagen. La segunda corresponde a un submuestreo de la base original. En relación a la experimentación, se pide como mínimo realizar los siguientes experimentos:

- Analizar para cada una de las variantes qué versión de la base de datos es posible utilizar, en base a requerimientos de memoria y tiempo de cómputo.
- Para cada una de las variantes propuestas, analizar el impacto en la tasa de efectividad del algoritmo de reconocimiento al variar la cantidad de componentes principales considerados. Estudiar también como impacta la cantidad de imágenes consideradas para cada persona en la etapa de entrenamiento.
- En caso de considerar más de una posibilidad para determinar a que persona corresponde una nueva cara, considerar para cada una la mejor configuración de parámetros y compararlas entre ellas.

El objetivo final de la experimentación es proponer una configuración de parámetros/métodos que obtenga resultados un buen balance entre la tasa de efectividad de reconocimiento de caras, la factibilidad de la propuesta y el tiempo de cómputo requerido.

Programa y formato de archivos

Se deberán entregar los archivos fuentes que contengan la resolución del trabajo práctico. El ejecutable tomará tres parámetros por línea de comando, que serán el archivo de entrada, el archivo de salida, y el método a ejecutar (0 método estándar, 1 método alternativo).

Asumimos que la base de datos de imágenes se encuentra organizada de la siguiente forma: partiendo de un directorio raíz, contiene un subdirectorio por cada una de las personas donde se encuentran las imágenes de entrenamiento. El nombre de las imágenes será un número y su extensión una archivo `.pgm` (ejemplo: `1.pgm`, `2.pgm`, etc.). De esta forma se puede saber a que persona corresponde cada imagen simulando un etiquetado de las mismas.

Para facilitar la experimentación, el archivo de entrada con la descripción del experimento sigue la siguiente estructura:

- La primera línea contendrá el *path* al directorio que contiene la base de datos, seguido de 5 números enteros que representan la cantidad de filas y columnas de las imágenes de la base, cuantas personas (p) y cuantas imágenes por cada una de ellas ($nimgp$), y cuántas componentes principales se utilizarán en el experimento (k). A continuación se muestra un ejemplo de una base de datos de imágenes de 112×92 , con 41 sujetos, 5 imágenes por sujeto y tomando 15 componentes principales.

```
../data/ImagenesCaras/ 112 92 41 5 15
```

- A continuación, el archivo contendrá p líneas donde en cada una de ellas se especificará la carpeta correspondiente a la p -ésima persona, seguido de $nimgp$ números enteros indicando las imágenes a considerar para el entrenamiento. La siguiente línea muestra como ejemplo 5 imágenes (2, 4, 7, 8 y 10) a considerar para la persona `s10`.

```
s10/ 2 4 7 8 10
```

- Finalmente, se especificará un número n_{test} de imágenes (preferentemente no contenidas en la base de imágenes) para las cuales se desea identificar a quien pertenecen. Cada una de ellas se especificará en una nueva línea, indicando el *path* al archivo seguido del número de individuo al que pertenece (relativo a la numeración establecida en el punto anterior). La siguiente línea muestra un ejemplo de una imagen a testear para el sujeto número 1:

```
../data/ImagenesCaras/s1/9.pgm 1
```

El archivo de salida obligatorio tendrá el vector solución con los k valores singulares de mayor magnitud, con una componente del mismo por línea. Adicionalmente, en caso de considerarlo conveniente, se podrán agregar archivos adicionales opcionales con más información que la descrita en la presente sección.

Junto con el presente enunciado, se adjunta una serie de scripts hechos en **python** y un conjunto instancias de test que deberán ser utilizados para la compilación y un testeo básico de la implementación. Se recomienda leer el archivo **README.txt** con el detalle sobre su utilización.

Fecha de entrega

- *Formato electrónico:* Jueves 15 de Mayo de 2014, hasta las 23:59 hs., enviando el trabajo (informe+código) a `metnum.lab@gmail.com`. El subject del email debe comenzar con el texto [TP2] seguido de la lista de apellidos de los integrantes del grupo.
- *Formato físico:* Viernes 16 de Mayo de 2014, de 17:30 a 18:00 hs.

Referencias

- [1] TURK, M., AND PENTLAND, A. Eigenfaces for recognition. *Journal of cognitive neuroscience* 3, 1 (1991), 71–86.