



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA



Métodos Numéricos

Trabajo práctico 2

Tu cara me suena

Resumen

El presente trabajo documenta un algoritmo de reconocimiento de rostros para sujetos pertenecientes a una base de datos. Hace uso de un Análisis de Componentes Principales (PCA según sus siglas en inglés) sobre un conjunto de imágenes digitales de rostros de sujetos. Se alega poder identificar con una tasa de error mínima una nueva imagen de un rostro de una persona perteneciente a la base de datos. Se explican los supuestos matemáticos asumidos en el algoritmo, se muestran resultados de diferentes pruebas de su efectividad y se llega a conclusiones sobre todo el proceso.

Integrante	LU	Correo electrónico
Danós, Alejandro	381/10	adp007@gmail.com
Gandini, Luciano	207/10	gl.gandini@gmail.com
Russo, Christian Sebastián	679/10	christian.russo@gmail.com

Palabras claves:

Reconocimiento caras. PCA. Power Method. Deflation. Autovalores. Autovectores. Matriz semi definida positiva.

Índice

1. Introducción Teórica	3
1.1. Metodo de la Potencia	3
1.2. Deflación	3
2. Desarrollo	4
2.1. Archivo de Entrada	4
2.1.1. Explicacion	4
2.2. Archivos de salida	4
2.3. Autovalores	4
2.4. Tiempos y porcentajes de acierto	4
2.5. Standard output	5
2.6. Metodo de la Potencia	5
2.6.1. Explicacion	5
2.6.2. Pseudocodigo	5
2.6.3. Ejemplo	6
2.7. Deflacion	10
2.7.1. Explicacion	10
2.7.2. Pseudocodigo	10
2.7.3. Ejemplo	10
2.8. Reconocimiento de caras	11
2.8.1. El vecino más cerca	11
2.8.2. Centros de masa	11
2.9. Algoritmo Principal	12
2.10. Demostraciones	13
3. Experimentación	15
4. Resultados	16
4.1. Experimentacion con Imagenes Reducidas	16
4.1.1. Metodo 0: Utilizando $A^t A$	16
4.1.2. Metodo 1: Utilizando AA^t	25
4.2. Experimentacion con Imagenes Full	34
4.2.1. Metodo 0: Utilizando $A^t A$	34
4.2.2. Metodo 1: Utilizando AA^t	34
5. Apéndice	43
5.1. Enunciado	43
5.2. Generador de Tests	44

5.3. Método de compilación	44
5.3.1. Método 1	44
5.3.2. Método 2	44
5.4. Script para correr casos	44
5.5. Equipo de pruebas	44
5.6. Referencias bibliográficas	45

1. Introducción Teórica

1.1. Metodo de la Potencia

El **método de la potencia** es una técnica iterativa que permite determinar el autovalor dominante de una matriz, es decir, el autovalor con mayor magnitud. Una ligera modificación en el método permite usarlo para determinar otros autovalores. Una propiedad útil del método de la potencia es que no solo produce un autovalor, sino también un autovector asociado.

De hecho, es frecuente que el método se aplique para calcular un autovector para un autovalor determinado por otros medios.

Para aplicar el método de la potencia supondremos que la matriz $A \in \mathbb{R}^{n \times n}$ tiene n autovalores $\lambda_1, \lambda_2, \dots, \lambda_n$ con un conjunto asociado de autovectores linealmente independientes $\{v_1, v_2, \dots, v_n\}$. Más aún, supondremos que A tiene exactamente un autovalor con módulo máximo, λ_1 , cuya magnitud es la mayor, por lo que

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| \geq 0$$

El procedimiento consiste en elegir un vector inicial $x \in \mathbb{R}^n$ y multiplicarlo por izquierda por la matriz A^k con $k \in \mathbb{N}$, y normalizar el vector resultante. Se puede demostrar que cuando $k \rightarrow \infty$, $\frac{A^k}{\|A^k\|}$ tiende al autovector asociado al autovalor dominante λ_1 . Para más información, referirse a [1]

1.2. Deflación

“Deflation techniques involve forming a new matrix B whose eigenvalues are the same as those of A , except that the dominant eigenvalue of A is replaced by the eigenvalue 0 in B .” - [1, p. 604]

En nuestro caso, para obtener la matriz B , le restaremos a la matriz A otra matriz formada por su autovalor dominante y el autovector asociado a éste, de la siguiente manera: $B = A - \lambda_1 v v^t$, siendo v el autovector y λ el autovalor asociado. Más adelante veremos este proceso en detalle.

2. Desarrollo

2.1. Archivo de Entrada

2.1.1. Explicacion

El ejecutable toma tres parámetros por línea de comando, que será el archivo de entrada, el archivo de salida, y el método a ejecutar (0 método estándar, 1 método alternativo). 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 de 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 especifica un número *ntest* de imágenes (preferentemente no contenidas en la base de imágenes) para las cuales se desea identificar a quien pertenecen. Cada una de ellas se especifica 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
```

2.2. Archivos de salida

2.3. Autovalores

El archivo pasado por parámetro, tendrá el vector solución con los k valores singulares de mayor magnitud, con una componente del mismo por línea.

2.4. Tiempos y porcentajes de acierto

Adicionalmente se creará otro archivo de salida, con mismo nombre pero finalizado en *.times*, donde se brindará la siguiente informacion en orden. Primero el k pasado por parámetro, la cantidad de personas, cantidad de muestras, el tiempo correspondiente a obtener la matriz con k vectores, el tiempo que tomo comparar las *ntest* imágenes con el método *Todos*, el tiempo que tomo comparar las *ntest* imágenes con el método *Centro*, el porcentaje de aciertos con el método *Todos* y el porcentaje de aciertos con el método *Centro*.

2.5. Standard output

Dado que con ciertos casos el tiempo de cómputo será muy extenso, se imprimirá por *stdOutput* cuantas iteraciones necesarias para calcular el *i*ésimo autovector.

2.6. Metodo de la Potencia

2.6.1. Explicacion

El método de la potencia es un método para obtener una estimación del autovector asociado al autovalor de módulo máximo de una matriz. Se basa en elegir un vector, que puede ser elegido aleatoriamente, y multiplicarlo iterativamente por izquierda por la matriz de la cual queremos obtener sus autovectores/autovalores. Se puede demostrar, referirse a [1, p. 576] por ejemplo, que cuando $k \rightarrow \infty$ entonces $A^k x$ tiende al autovalor dominante para cualquier x si la matriz cumple con las hipótesis. El autovector asociado a este autovalor sería el último vector resultante de multiplicar el inicial por la matriz iterativamente.

Se pide de esta matriz que la matriz sea cuadrada, tenga n autovalores con una colección asociada de autovectores linealmente independientes. Además, se pide que haya un autovalor dominante por sobre los demás. Es decir, $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$ para λ_i autovalor de A .

La matriz por sobre la cual estaremos aplicando la técnica es la matriz de covarianzas, que es cuadrada y simétrica ¹, por lo cual tiene además una base ortonormal de autovectores, que son linealmente independientes.

Sin embargo, no podremos asegurar que haya un autovalor dominante por sobre el resto, por lo que no estaríamos cumpliendo completamente con las hipótesis del método aquí enunciado. Queda como trabajo a futuro investigar cuáles serían las consecuencias de aplicarlo sobre matrices sin un autovalor dominante y, además, cómo identificarlas sin tener que encontrar sus autovalores.

Además, la cantidad de iteraciones necesarias para poder obtener los autovectores puede ser inviable para el cómputo de un programa si ésta es muy grande. Por eso, está la opción de obtener un autovector y autovalores aproximados antes de llegar a la convergencia. En caso contrario, si el vector $A^k x$ es igual, para algún criterio de igualdad a elección, a $A^{k-1} x$ entonces se para el algoritmo y se devuelven el autovector y autovalor encontrados. Nuestro criterio para diferenciar a un autovector del anterior es que haya una diferencia de 10^{-7} para alguna de sus coordenadas.

También es un dato importante el hecho de que la técnica de deflación a usar más adelante asume que el autovector es de norma uno, por eso cada vez que se multiplica por izquierda al vector obtenido en el paso anterior se lo divide por su norma para normalizarlo.

2.6.2. Pseudocodigo

Algorithm 1 Método de la Potencia(Matrix B, vector x_0 , int niters)

```

v ← x0
for i = 0 hasta niters do
  w ←  $\frac{Bv}{\|Bv\|}$ 
  if Es igual al anterior then
    break
  end if
end for
λ ←  $\frac{v^t B v}{v^t v}$ 
return λ, v

```

¹Más de esto en la sección del algoritmo principal

2.6.3. Ejemplo

$$A = \begin{bmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix}$$

Inicializando con el vector

$$x = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Fase 1:

$$y^{(1)} = Ax^{(0)} =$$

$$\begin{bmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix}$$

$$c_1 = 2 \text{ (componente dominante de } y^{(0)})$$

$$x^{(1)} = \frac{1}{2}y^{(1)} = \frac{1}{2} \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Fase 2:

$$y^{(2)} = Ax^{(1)} =$$

$$\begin{bmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \\ 3 \end{bmatrix}$$

$$c_2 = 3$$

$$x^{(2)} = \frac{1}{3} \begin{bmatrix} 3 \\ -2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{2}{3} \\ 1 \end{bmatrix} = \begin{bmatrix} 1,0 \\ -0,6667 \\ 1,0 \end{bmatrix}$$

Fase 3:

$$y^{(3)} = Ax^{(2)} =$$

$$\begin{bmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} 1,0 \\ -0,6667 \\ 1,0 \end{bmatrix} = \begin{bmatrix} 3,6667 \\ -3,3333 \\ 3,6667 \end{bmatrix}$$

$$c_3 = 3,6667$$

$$x^{(3)} = \begin{bmatrix} 1 \\ -0,9091 \\ 1 \end{bmatrix}$$

Fase 4:

$$y^{(4)} = Ax^{(3)} =$$

$$\begin{bmatrix} 3,9091 \\ -3,8181 \\ 3,9091 \end{bmatrix}$$

$$c_4 = 3,9091$$

$$x^{(4)} = \begin{bmatrix} 1 \\ -0,9767 \\ 1 \end{bmatrix}$$

Fase 5:

$$y^{(5)} = Ax^{(4)} =$$

$$\begin{bmatrix} 3,9767 \\ -3,9534 \\ 3,9767 \end{bmatrix}$$

$$c_5 = 3,9767$$

$$x^{(5)} = \begin{bmatrix} 1 \\ -0,9942 \\ 1 \end{bmatrix}$$

Fase 6:

$$y^{(6)} = Ax^{(5)} =$$

$$\begin{bmatrix} 3,9942 \\ -3,9883 \\ 3,9942 \end{bmatrix}$$

$$c_6 = 3,9942$$

$$x^{(6)} = \begin{bmatrix} 1 \\ -0,9985 \\ 1 \end{bmatrix}$$

Fase 7:

$$y^{(7)} = Ax^{(6)} =$$

$$\begin{bmatrix} 3,9985 \\ -3,9970 \\ 3,9985 \end{bmatrix}$$

$$c_7 = 3,9985$$

$$x^{(7)} = \begin{bmatrix} 1 \\ -0,9996 \\ 1 \end{bmatrix}$$

Fase 8:

$$y^{(8)} = Ax^{(7)} =$$

$$\begin{bmatrix} 3,9996 \\ -3,9993 \\ 3,9996 \end{bmatrix}$$

$$c_8 = 3,9996$$

$$x^{(8)} = \begin{bmatrix} 1 \\ -0,9999 \\ 1 \end{bmatrix}$$

Fase 9:

$$y^{(9)} = Ax^{(8)} =$$

$$\begin{bmatrix} 3,9999 \\ -3,9998 \\ 3,9999 \end{bmatrix}$$

$$c_9 = 3,9999$$

$$x^{(9)} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

Fase 10:

$$y^{(10)} = Ax^{(9)} =$$

$$\begin{bmatrix} 4 \\ -4 \\ 4 \end{bmatrix}$$

$$c_{10} = 4$$

$$x^{(10)} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = x^{(9)}$$

Entonces

$$\lambda_1 = \lim_j c_j = 4$$

Y su autovector asociado:

$$v = \lim_j x^{(j)} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

Confirmamos usando usando el software MatLab que los autovectores de la matriz son $\lambda_1 = 4, \lambda_2 = 3, \lambda_3 = 1$ que es compatible con el λ_1 encontrado por el método de la potencia.

2.7. Deflacion

2.7.1. Explicacion

Después de aplicar el Método de la Potencia obtenemos un autovector y su autovalor asociado con módulo máximo, pero para obtener los siguientes debemos recurrir a algún tipo de deflación. Este proceso consiste en anular el autovalor del autovector dominante para luego de que se corra el Método de la Potencia una segunda vez, se obtenga el segundo autovalor dominante.

Hay muchas técnicas de deflación, algunas de las más usadas se pueden encontrar en [1], pero la que aplicaremos en este trabajo consiste en restarle a una matriz A original una combinación entre el autovector dominante encontrado por el Método de la Potencia y su autovalor asociado.

Sea A la matriz original con por lo menos dos autovalores dominantes distintos, entonces la matriz $B = A - \lambda_1 v v^t$ tiene a los mismos autovectores que A y los mismos autovalores asociados para todo λ_i con $i \neq 1$. En el caso de λ_1 , ahora vale 0.²

El único *problema* de esto es que si aplicamos Método de la Potencia + Deflación k veces, entonces estaríamos asumiendo que hay por lo menos k autovalores dominantes distintos, que no siempre es el caso. Es una buena propuesta para un trabajo a futuro investigar más en detalle en qué situaciones no sería correcto asumir esto.

2.7.2. Pseudocodigo

Algorithm 2 Deflación(Matriz A , vector v , valor λ)

Require: λ autovalor de módulo máximo y v su autovector asociado

Require: $\|v\|=1$ y ortogonal al resto de los autovectores

Require: $A \in \mathbb{R}^{n \times n}$

Sea $B \in \mathbb{R}^{n \times n}$

for $i = 1$ hasta n **do**

for $j = 1$ hasta n **do**

$B_{ij} \leftarrow A_{ij} - \lambda v_i v_j$

end for

end for

return B

Ensure: v autovector de B con autovalor asociado 0

Ensure: Si v' autovector de A con autovalor asociado λ' , $v' \neq v$ y $\lambda' \neq \lambda$, también lo es para B

2.7.3. Ejemplo

Volviendo al mismo ejemplo que en el Método de la Potencia, teníamos a

$$A = \begin{bmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix}$$

Y habíamos calculado su autovector $\lambda_1 = 4$ dominante y al autovector $v = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$ asociado. Para deflación requerimos de un autovector con norma igual a 1, entonces lo dividimos por su norma:

$$v_1 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{3} \\ -1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix}$$

²demostrado en la sección Demostraciones

$$B = A - \lambda_1 v_1 v_1^t = \begin{bmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix} - 4/(\sqrt{3})^2 \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 5/3 & 1/3 & -4/3 \\ 1/3 & 2/3 & 1/3 \\ -4/3 & 1/3 & 5/3 \end{bmatrix}$$

Si calculamos los autovectores y autovalores de esta matriz, usando software específico a elección o usando lápiz y papel, obtendremos los siguientes:

Los autovalores quedarían: $[0, 1, 3]$ y sus respectivos autovectores por columna

$$\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

como esperábamos.

2.8. Reconocimiento de caras

Habiendo hecho un análisis de componentes principales, PCA según sus siglas en inglés, tenemos dos técnicas para poder reconocer un rostro nuevo y tratar de identificarlo según nuestra base de datos

2.8.1. El vecino más cerca

El procedimiento más intuitivo para reconocer una nueva persona consiste en buscar las coordenadas de la imagen a reconocer según el nuevo ejes de coordenadas dados por los autovectores y buscar entre todos los puntos de la base de datos, también en las nuevas coordenadas, el que esté a menor distancia de esta nueva imagen. La distancia usada para comparar es la distancia euclídea, o la norma dos de la resta de ambos vectores. Luego, identificar al sujeto del cual pertenece esta imagen *más cercana*.

Algorithm 3 Vecino Más Cerca(Matriz A_final , vector v , Integer muestras)

Require: El vector v ya ha sido restado el μ y ha sido dividido por $\sqrt{n-1}$

```

minVal  $\leftarrow$   $\|fila_0(A\_final) - v\|$ 
minFila  $\leftarrow$  0
for  $i=0$  hasta  $cant\_filas(A)$  do
  if  $minVal > \|fila_i(A\_final) - v\|$  then
     $minVal \leftarrow \|fila_i(A\_final) - v\|$ 
     $minFila \leftarrow i$ 
  end if
end for
return  $(minFila/muestras)+1$ 
```

2.8.2. Centros de masa

La otra técnica también consiste en llevar todo al nuevo eje de coordenadas, pero no comparar la nueva imagen con todas las otras, sino que buscar un *punto medio* para todos los puntos de cada grupo de imágenes y buscar este punto medio, o centro de masa, que más se asemeje a la nueva imagen. Luego, identificar al sujeto por el cual ese centro de masa lo promedia.

Otra similitud es que de nuevo usamos la distancia euclídea para comparar vectores en el espacio de coordenadas.

Algorithm 4 Calcular Centros de Masa(Matriz A_final , Integer muestras, Integer sujetos)

```
centros  $\leftarrow$  Matriz  $\in \mathbb{R}^{sujetos \times cant\_columnas(A)}$ 
unaFila  $\leftarrow$  Vector  $\in \mathbb{R}^{cant\_columnas(A)}$ 
for i=0 hasta sujetos do
  unaFila  $\leftarrow$  filai*muestras( $A\_final$ )
  for j=0 hasta muestras do
    unaFila  $\leftarrow$  unaFila + filai*muestras+j( $A\_final$ )
  end for
  unaFila  $\leftarrow$  unaFila/muestras
  filai(centros)  $\leftarrow$  unaFila
end for
return centros
```

Algorithm 5 Buscar Centro Más Cercano(Matriz $centros$, vector v , Integer muestras, Integer sujetos)

```
return Vecino Más Cerca (  $centros$ ,  $v$ , 1)
```

2.9. Algoritmo Principal

Teniendo ya todos los métodos necesarios, sólo faltaría enunciar cómo trabajarían entre ellos en este trabajo.

El primer paso consiste en leer las imágenes de la base de datos y guardarlas en una matriz poniendo a cada una como filas obteniendo una matriz $A_orig \in \mathbb{R}^{n \times m}$ siendo n la cantidad de imágenes y m las dimensiones de éstas.

Luego, buscamos la media μ para cada píxel de nuestra matriz para obtener a la matriz A_final definida como $A_final \in \mathbb{R}^{n \times m} = \frac{A_orig - \mu}{\sqrt{n-1}}$. Acá es donde aparece la primera decisión a tomar. El PCA estaba en un principio pensado para ser aplicado sobre la matriz de covarianzas de A_orig , que sería igual a $A_final^t A_final \in \mathbb{R}^{m \times m}$ una matriz simétrica. Sin embargo, si buscamos los autovalores y autovectores de la matriz $A_final A_final^t$, entonces serían los mismos autovalores que los de la matriz de covarianzas y para obtener sus autovectores asociados, simplemente multiplicamos por A^t a izquierda al autovector asociado al mismo autovalor en $A_final A_final^t$. La diferencia sería que en el caso de las imágenes, la cantidad de filas tiende a ser mucho más basta que la cantidad de columnas, ya que la primera es la dimensión de la imagen y la segunda la cantidad de imágenes en la base de datos.³

Una vez que hicimos nuestra primera decisión, aplicamos el método de la potencia + deflación sobre la matriz elegida para obtener la cantidad deseada de autovectores. Una vez obtenidos, se buscan las coordenadas de cada imagen sobre estos nuevos ejes que pueden ser mucho menores a los de la matriz original. Ése es uno de los objetivos de aplicar PCA, disminuir la dimensión de los datos minimizando el error, es decir, la diferencia entre las varianzas.

Ahora sólo quedaría identificar al sujeto de una nueva imagen. Llegó el momento de la segunda decisión, agrupar por el punto más cercano o por el centro de masa más cercano, dos técnicas previamente explicadas.

Una vez obtenido el resultado del agrupamiento, se devuelve el sujeto al que pertenece la nueva imagen según el método.

³Para una explicación matemática de todo esto, referirse a la sección de Demostraciones.

2.10. Demostraciones

El método de la potencia asume que tenemos un autovalor dominante y que todos los autovalores son mayores o iguales a 0. $A^t A$ es simétrica por lo que tenemos una base ortonormal de autovectores y además es semi-definida positiva, por lo que sus autovalores son positivos o 0. El problema es que no podemos asegurar que después de aplicar deflación la matriz seguirá siendo semi-definida positiva, aunque sí simétrica. Sin embargo, asumiremos estos supuestos de todas formas en este trabajo. Demostraremos algunas consecuencias de asumir esto.

Sean $A \in \mathbb{R}^{n \times m}$, $A^t A \in \mathbb{R}^{m \times m}$, $AA^t \in \mathbb{R}^{n \times n}$, $\lambda \in \mathbb{R}$, $v \in \mathbb{R}^m$, $u \in \mathbb{R}^n$, $\lambda' \in \mathbb{R}$, $B = A^t A - \lambda vv^t$, $\Sigma \in \mathbb{R}^{m \times n}$, $U \in \mathbb{C}^{m \times m}$, $V \in \mathbb{C}^{n \times n}$ y $A = U\Sigma V^t$, demostraremos algunas hipótesis que habíamos asumido para este trabajo.

Lema: Si v y v' son vectores de una base ortonormal de autovectores de $A^t A$ con autovalores asociados λ y λ' , entonces v es autovector de B con autovalor 0 y v' es autovector de B con autovalor λ'

- $Bv = 0v = 0$
- $Bv' = \lambda'v'$

Prueba:

$$\begin{aligned} Bv &= (A^t A)v - (\lambda vv^t)v = \lambda v - \lambda v(v^t v) = \lambda v - \lambda v = 0v = 0 \\ Bv' &= (A^t A)v' - (\lambda vv^t)v' = \lambda' v' - \lambda v(v^t v') = \lambda' v' - \lambda v(0) = \lambda' v' \end{aligned}$$

Lema: las matrices $A^t A$ y AA^t son simétricas.

Prueba:

$$(A^t A)^t = (A)^t (A^t)^t = A^t A \quad (AA^t)^t = (A^t)^t (A)^t = AA^t$$

Lema: la matriz B es simétrica

Prueba:

$$B_{ij} = (A^t A)_{ij} - \lambda v_i (v^t)_j = (A^t A)_{ij} - \lambda v_i v_j \stackrel{(1)}{=} (A^t A)_{ji} - \lambda v_j (v^t)_i = B_{ji}$$

(1) $A^t A$ simétrica y v vector.

Lema: Los valores singulares de A son los mismos que los valores singulares de A^t .

Prueba: Los valores singulares de la matriz Σ son las raíces de los autovalores en orden decreciente

por la diagonal. Los autovalores están definidos como los valores que anulan a la función $\psi(\lambda) = \det(\lambda I - A)$. En el caso de la traspuesta, sus autovalores son los que anulan a la función $\psi(\lambda) = \det(\lambda I - A^t)$, pero $(\lambda I - A)^t = (\lambda I - A^t)$ y el determinante es invariante al traspasar una matriz. Entonces los autovalores son los mismos y, por ende, los valores singulares también.

Lema:

- $A^t = V\Sigma U^t$ con $A^t \in \mathbb{R}^{m \times n}$
- $AA^t = U\Lambda U^t$ con Λ la matriz con los autovalores de A y A^t en la diagonal y $AA^t \in \mathbb{R}^{n \times n}$.
- $A^tA = V\Lambda V^t$ con Λ la matriz con los autovalores de A y A^t en la diagonal $A^tA \in \mathbb{R}^{m \times m}$.

Prueba: El primero es inmediato de traspasar A y del hecho de que Σ es diagonal. Para el segundo y el tercero:

$$AA^t = U\Sigma V^t V\Sigma U^t \stackrel{(1)}{=} U\Sigma\Sigma U^t \stackrel{(2)}{=} U\Lambda U^t$$

$$A^tA = V\Sigma U^t U\Sigma V^t \stackrel{(1)}{=} V\Sigma\Sigma V^t \stackrel{(2)}{=} V\Lambda V^t$$

(1) U y V matrices ortogonales

(2) Λ diagonal con valores singulares en la diagonal.

Lema: Si u es autovector de AA^t con λ autovalor asociado, entonces $A^t u \in \mathbb{R}^m$ es autovector de A^tA también con λ autovalor asociado.

Prueba:

$$AA^t u = \lambda u \iff (1) A^t AA^t u = \lambda A^t u \iff A^t A(A^t u) = \lambda(A^t u)$$

(1) Se multiplica por A^t de ambos lados.

Por ende, $A^t u$ es autovector de A^tA .

3. Experimentación

Llegado el momento de la experimentación, se tenía una base de datos de imágenes divididas entre resoluciones de 112 x 92 píxeles y de 28 x 23 píxeles y se decidió dividir los tests de igual forma.

A los tests hechos sobre las imágenes de 28 x 23 se los subdividió en dos, tests con el método 0 y tests con el método 1. Esto hace referencia al método empleado para buscar los autovectores y autovalores de la matriz de covarianzas. El método 0 consistía en hacerlo sobre la matriz $A^t A$ mientras que el método 1 consistía en hacerlo sobre la matriz AA^t .

Por el lado de las imágenes de 112 x 92 píxeles se volvió inviable el método 0, dado que la dimension de la matriz terminaría siendo de $(112 * 92)^2 > 106$ millones de celdas mientras que la matriz del método 1 tendría **cómo máximo** $(41 * 10)^2 = 168100$ celdas si usamos a todas las imágenes.

En cada una de las subdivisiones se testeó lo siguiente al ir aumentando el K, o la cantidad de autovectores:

- **Mediciones de TK:** mide el tiempo de obtener la matriz de autovectores. Es decir:
 1. Restarle el μ y dividir por $\sqrt{n-1}$ a la matriz
 2. Tiempo de calcular los K autovectores
 3. Tiempo de multiplicar por izquierda al autovector de AA^t por A^t para obtener el autovector con el mismo autovalor pero de $A^t A$.

Se varía entre 1, 5 y 10 muestras para 11, 21 y 41 personas.

- **Mediciones de Ttodos:** el tiempo de identificar a todos los sujetos usando el método de identificación del vecino más cercano. Incluye el tiempo de restarle el μ y dividir por $\sqrt{n-1}$ a la matriz. Se varía entre 1, 5 y 10 muestras para 11, 21 y 41 personas.
- **Mediciones de Tcentro:** el tiempo de identificar a todos los sujetos usando el método de identificación de los centros de masa. Incluye el tiempo de restarle el μ y dividir por $\sqrt{n-1}$ a la matriz. Se varía entre 1, 5 y 10 muestras para 11, 21 y 41 personas.
- **Mediciones de HitsTodos:** el coeficiente entre los *hits* al identificar un sujeto y la cantidad de estos usando el método del vecino más cercano. Se varía entre 1, 3, 5, 7, 9 y 10 muestras para 11, 21 y 41 personas.
- **Mediciones de HitsCentro** el coeficiente entre los *hits* al identificar un sujeto y la cantidad de estos usando el método de los centros de masa. Se varía entre 1, 3, 5, 7, 9 y 10 muestras para 11, 21 y 41 personas.

DECIIIIIIIIIIIIIIR QUÉ SIGNIFICAN LAS MUEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE-
TRAAAAAAAAAAAAAAAAAAAAAAS.

Siempre que se quiera identificar a alguien, se elije a una persona perteneciente a la base de datos usada para generar la matriz de autovectores.

Sobre estos tests esperamos encontrar que:

- Los tiempos son crecientes acorde al incremento de los autovectores buscados dado que mientras más grande sea la matriz de autovectores, más tiempo se tardará en encontrarla y, en el caso del método 1, se deberá multiplicar por izquierda a cada autovector para trasladarlo a un autovector de la matriz de $A^t A$.
- Los tiempos del método 1 deben ser mucho menores a los tiempos del método 0.
- El coeficiente de HitsTodos debe ser siempre 1 para los casos en los que se usen 10 muestras dado que toda muestra a identificar ha sido usada para generar la matriz de autovectores y, además, este método de identificación compara con todos los puntos, y entre ellos va a estar él mismo, que terminaría siendo el más cercano.
- Los coeficientes de HitsTodos y HitsCentro deben aumentar acorde aumenten la cantidad de muestras y la cantidad de autovectores.

4. Resultados

4.1. Experimentacion con Imagenes Reducidas

4.1.1. Metodo 0: Utilizando $A^t A$

Mediciones de TK

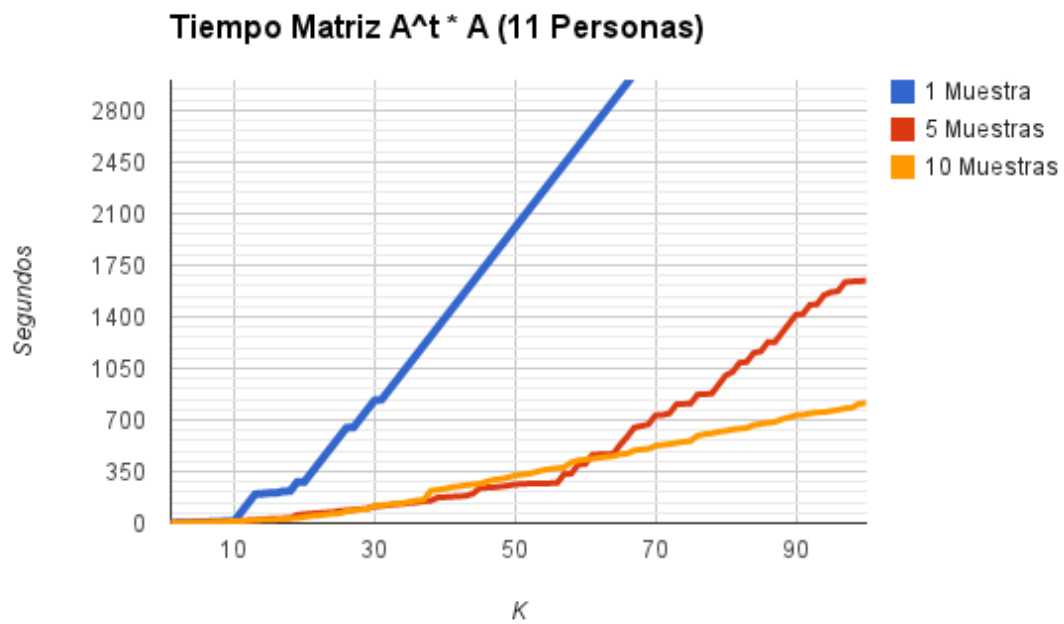
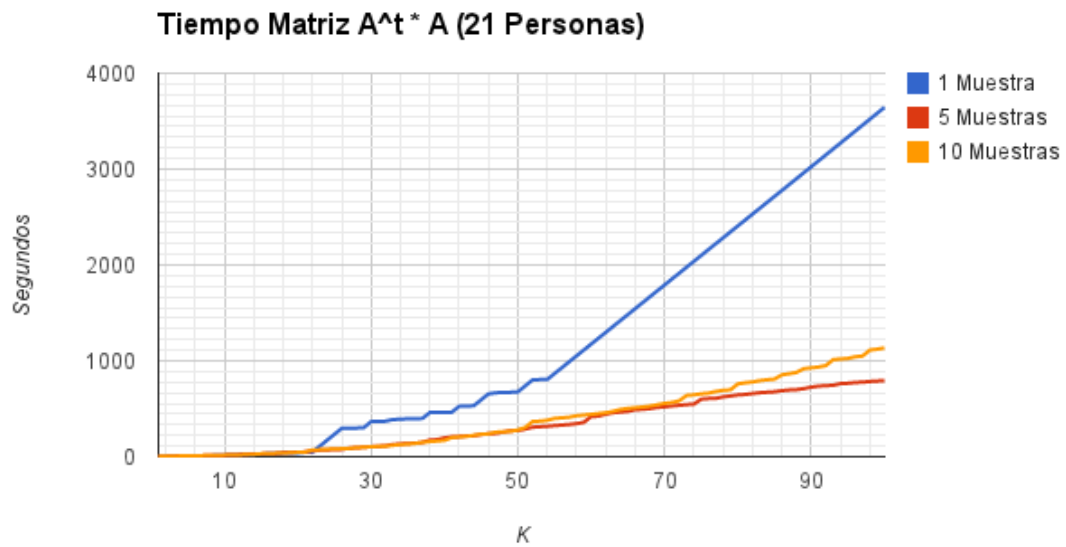
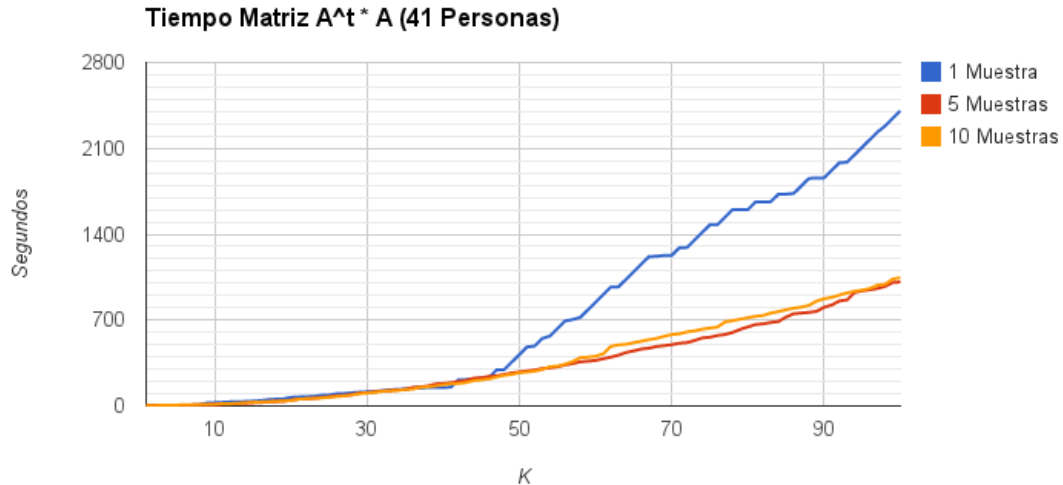
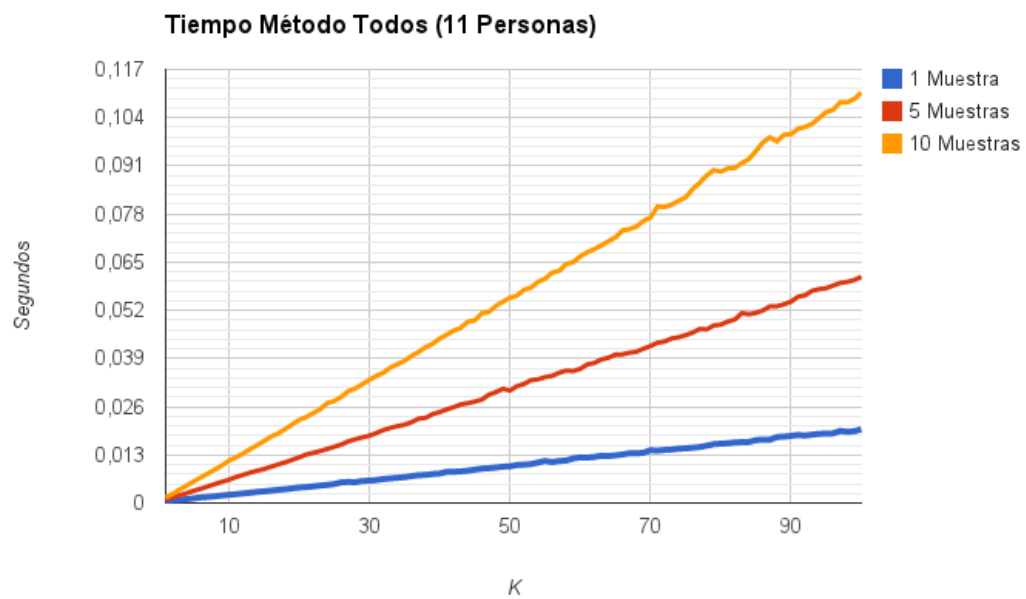
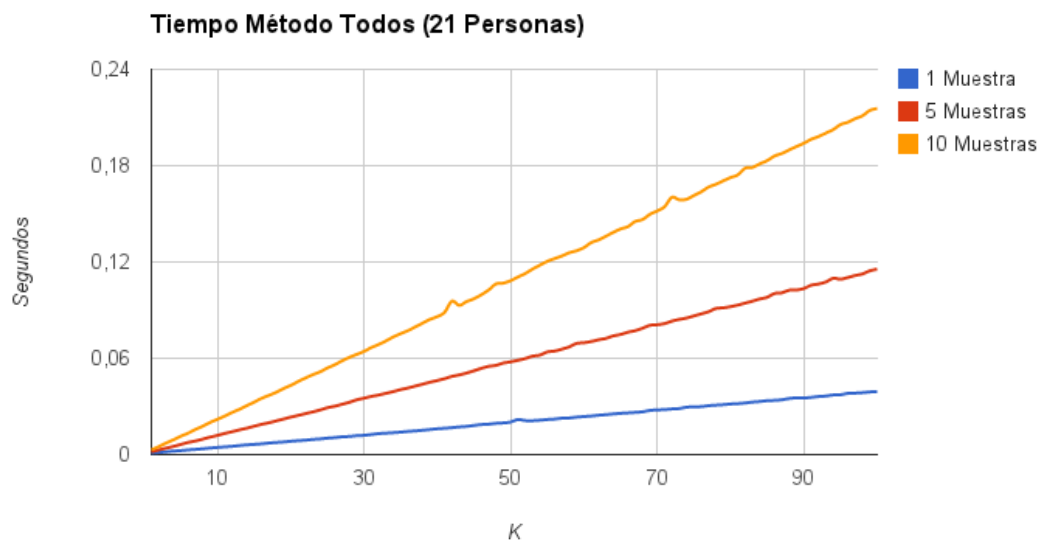


Figura 1: Tiempos Matrix $A^t A$ con 11 personas variando K

Figura 2: Tiempos Matrix $A^t A$ con 21 personas variando KFigura 3: Tiempos Matrix $A^t A$ con 41 personas variando K

Algo que en un principio parecería muy extraño es que el tiempo de calcular la matriz crece inversamente proporcional a la cantidad de muestras, sin ningún sentido obvio. Pero, si nos ponemos a analizar los tests, veremos que si sólo tenemos una muestra para identificar, eso significaría que tenemos 9 muestras para calcular los autovectores y por ende la matriz sería mucho mayor.

Además, el tiempo crece con los K, totalmente intuitivo. **Mediciones de Ttodos**

Figura 4: Tiempos Todos con Matrix $A^t A$ con 11 personas variando KFigura 5: Tiempos Todos con Matrix $A^t A$ con 21 personas variando K

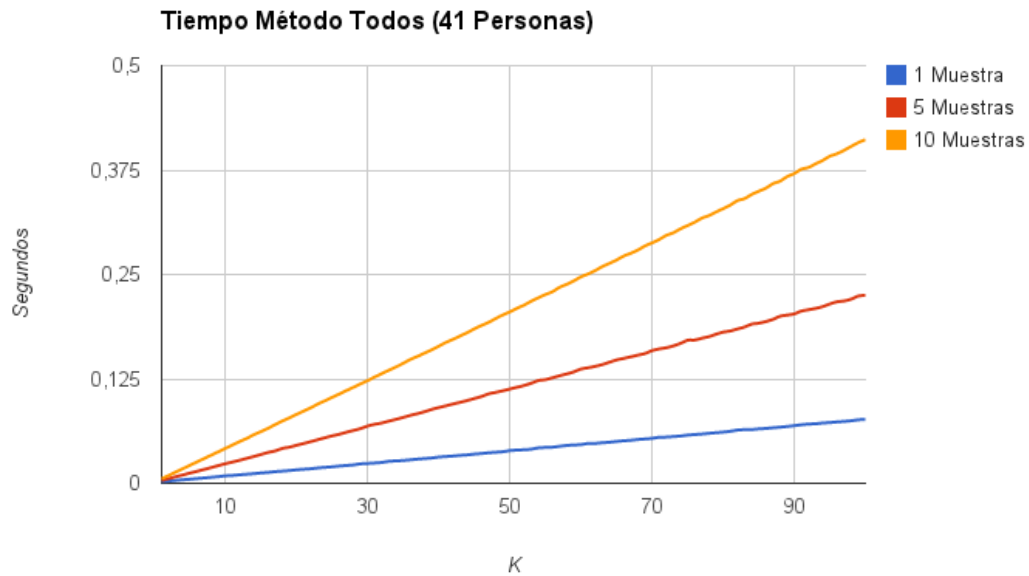
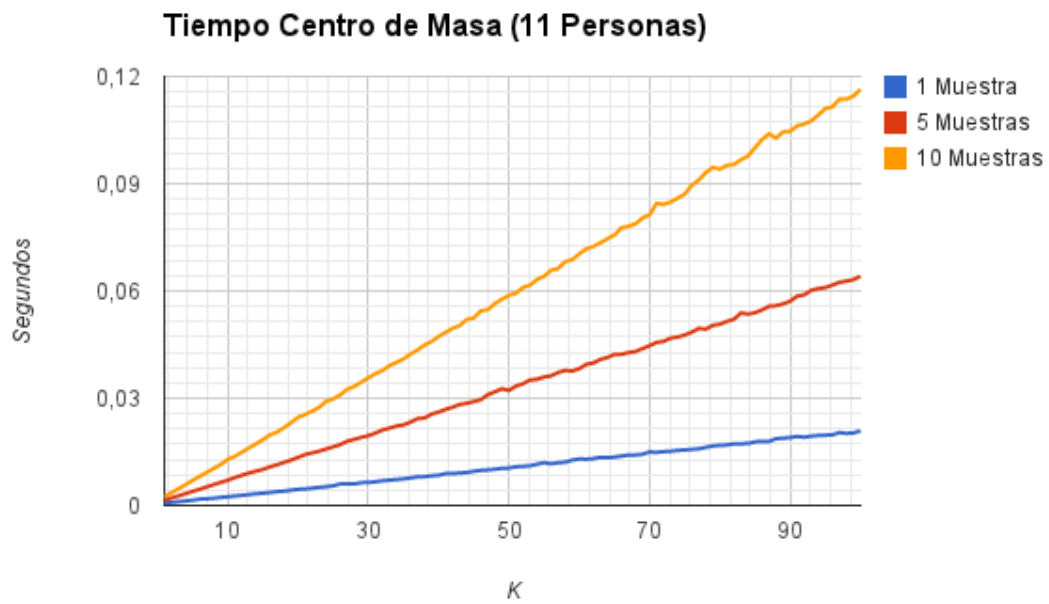
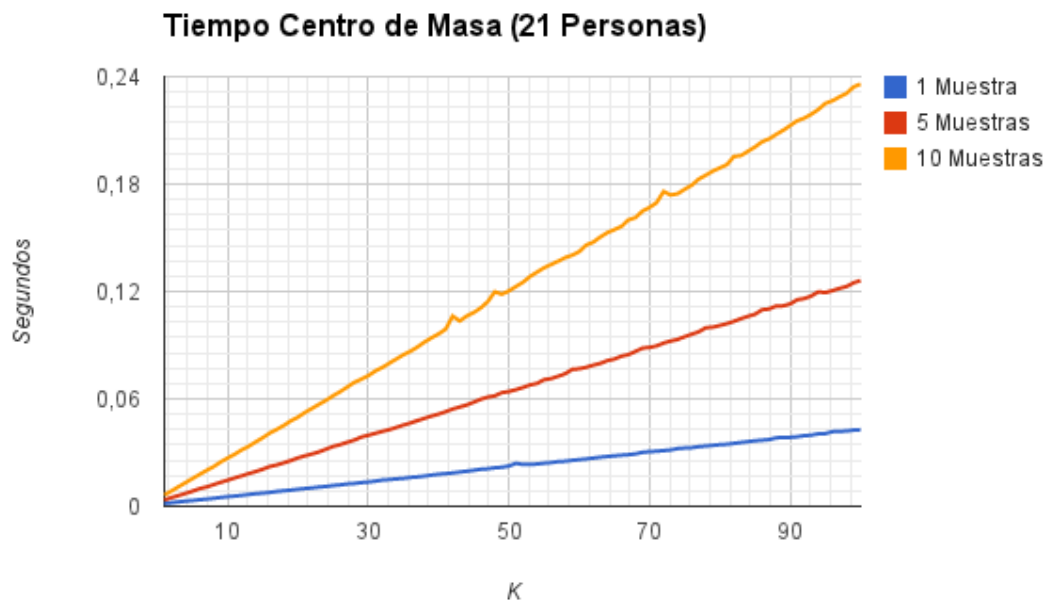


Figura 6: Tiempos Todos con Matrix $A^t A$ con 41 personas variando K

En el caso de los tiempos de encontrar al vecino más cercano, se ve que mientras mayor sea la cantidad de personas a identificar, mayor será el tiempo en el caso de usar 11, 21 y 41 personas.

Un dato interesante e inesperado son los pequeños saltos que se pueden apreciar en los tiempos del caso de usar 11 personas y que se acentúan acorde aumentan la cantidad de muestras. Esto parecería ser extraño porque no importa si dado un punto empezamos a recorrer los vecinos más cercanos posibles de éste y lo encontramos apenas empezamos, el algoritmo seguirá iterando hasta haber recorrido todos los puntos.

Mediciones de Tcentro

Figura 7: Tiempos Centro con Matrix $A^t A$ con 11 personas variando KFigura 8: Tiempos Centro con Matrix $A^t A$ con 21 personas variando K

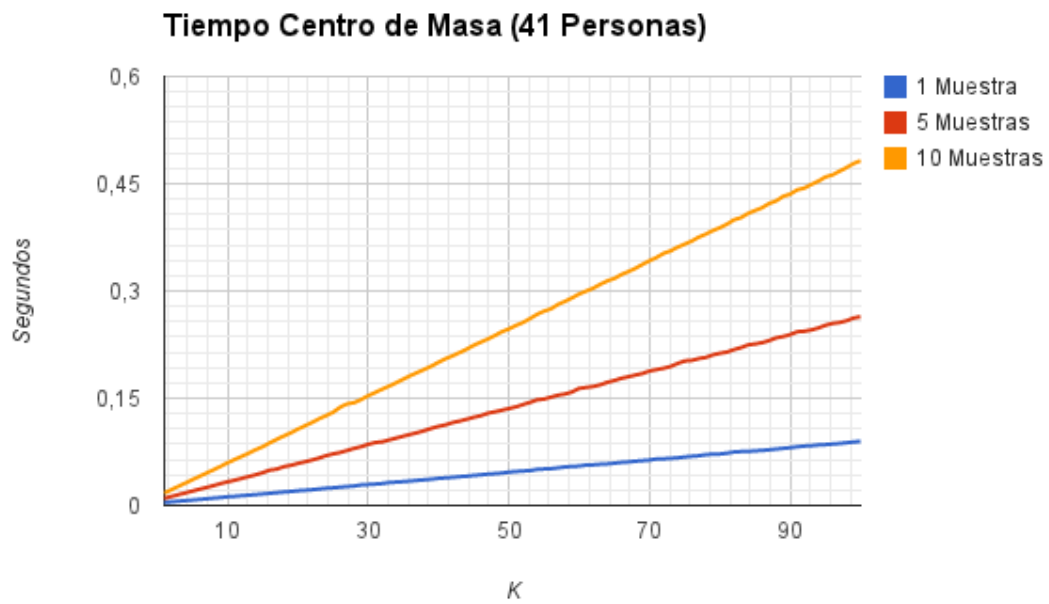


Figura 9: Tiempos Centro con Matrix $A^t A$ con 41 personas variando K

Aquí pasa algo muy parecido con el caso de las mediciones de TTodos (vecino más cercano). Los tiempos crecen a medida se incrementa la cantidad de muestras, y tenemos picos inexplicables en los casos con pocas personas.

Mediciones de HitsTodos

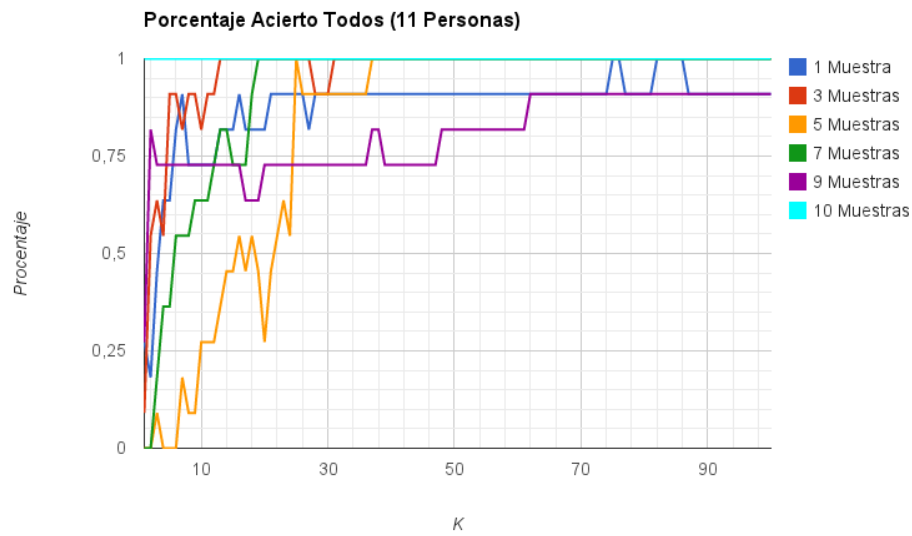


Figura 10: Coeficientes de efectividad vecino más cercano con Matrix $A^t A$ con 11 personas variando K

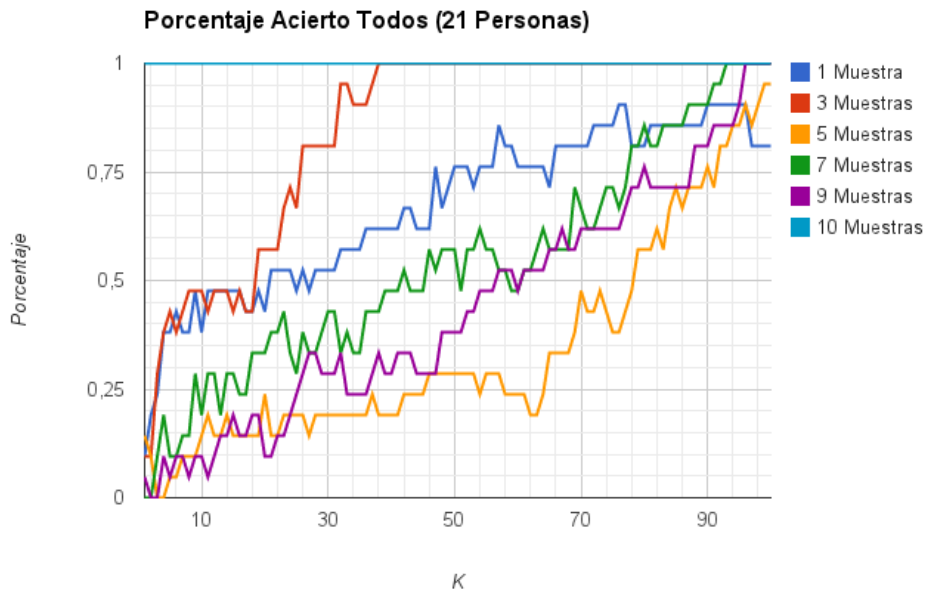


Figura 11: Coeficientes de efectividad vecino más cercano con Matrix $A^t A$ con 21 personas variando K

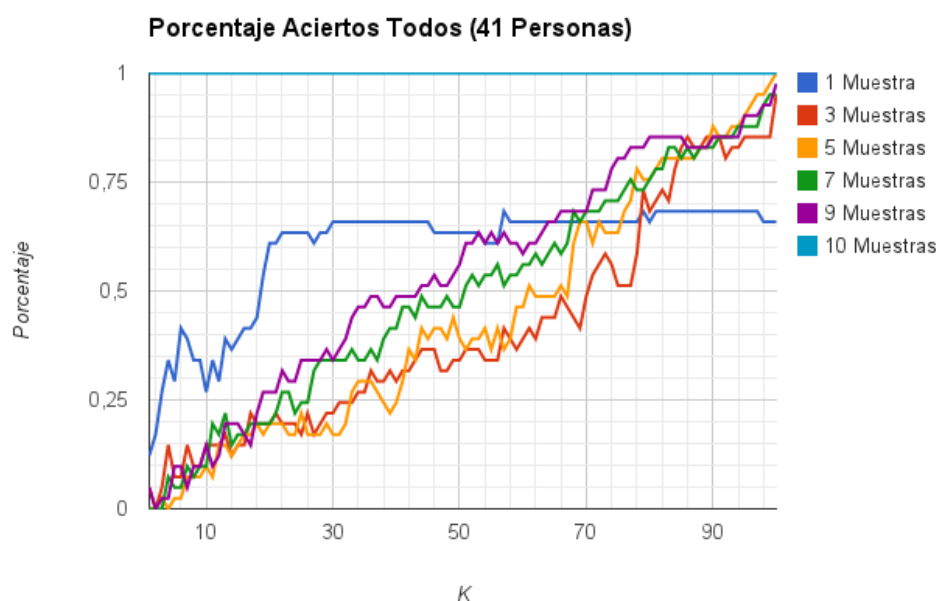


Figura 12: Coeficientes de efectividad vecino más cercano con Matrix $A^t A$ con 41 personas variando K

A simple vista, parecen gráficos extraños, con muchos saltos y con el caso de 10 muestras constante en 1. Lo primero es mucho más sorprendente en el caso con 11 personas, dado que los saltos con mucho mayores al ir aumentando la cantidad de autovectores. Esto significaría que con menos autovectores obtendríamos un mejor coeficiente y no parecen ser cantidad de autovectores aislados, dado que, por ejemplo en el pico de la medición con 9 muestras, el coeficiente empieza a caer en lo que parecería ser $K = 45$ y no se recupera hasta el $k = 58$, para volver a caer inmediatamente y desde ahí empezar a subir hasta el 1 y seguir en ese valor casi hasta el final cuando cae con $k = 90$.

En un principio no tendríamos una explicación para este fenómeno, dado que la técnica parecería ser consistente con la cantidad de autovectores.

Si aumentamos la cantidad de personas, se nota una tendencia más clara a aumentar la efectividad acorde aumenta la cantidad de autovectores. Siguen habiendo saltos pero no parecerían ser tan acentuados.

También es interesante ver que en todos los gráficos, las mediciones con todas las muestras menos la de una sola muestra llegan hasta el 100 % de efectividad en algún momento o muy cerca.

FALTA TERMINAR

Mediciones de HitsCentro

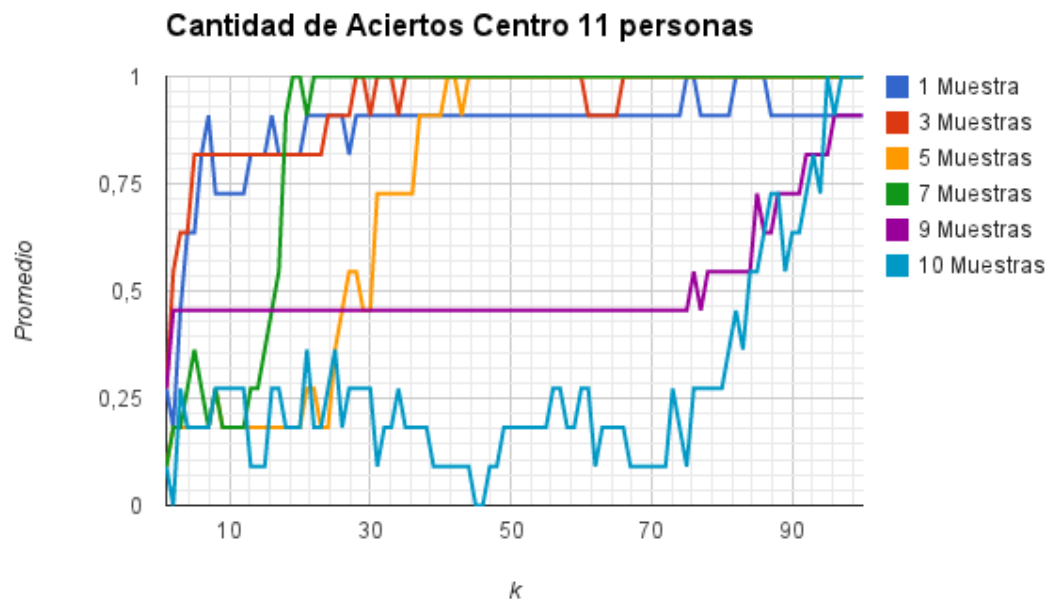


Figura 13: Coeficientes de efectividad de Centros de Masa con Matrix $A^t A$ con 11 personas variando K

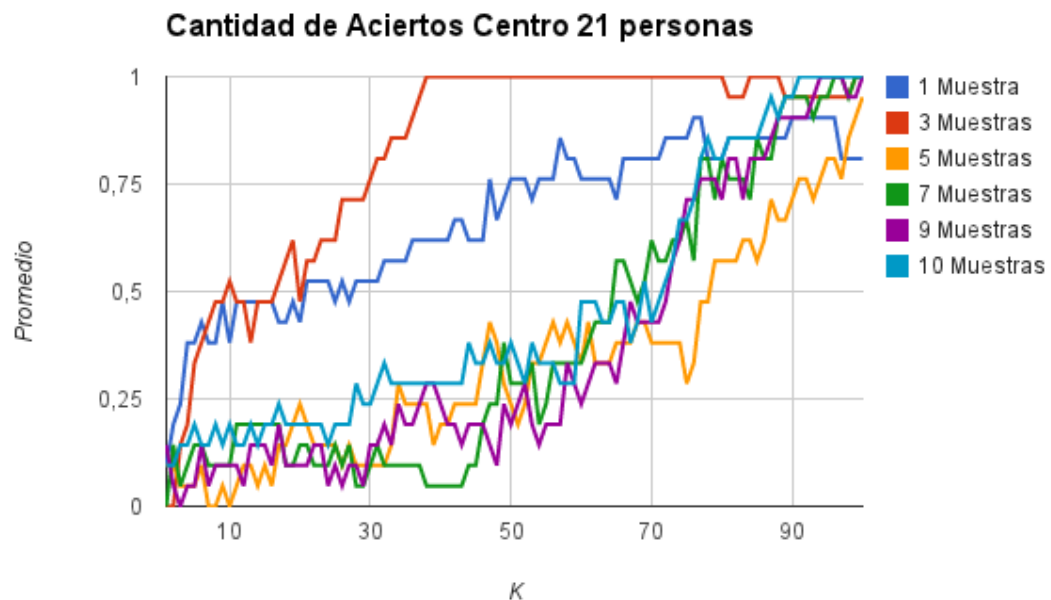


Figura 14: Coeficientes de efectividad de Centros de Masa con Matrix $A^t A$ con 21 personas variando K

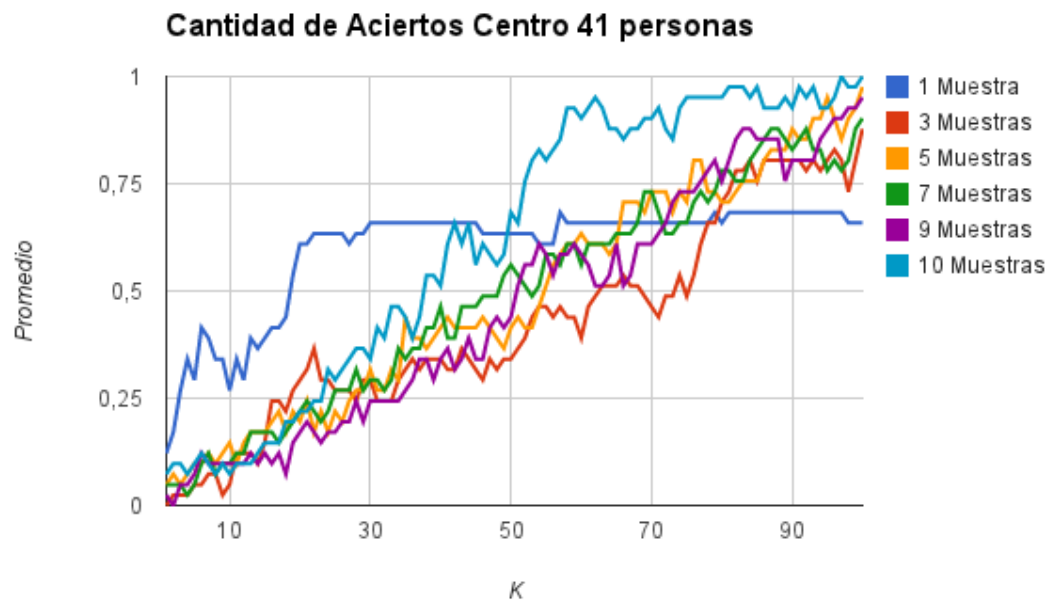


Figura 15: Coeficientes de efectividad de Centros de Masa con Matrix $A^t A$ con 41 personas variando K

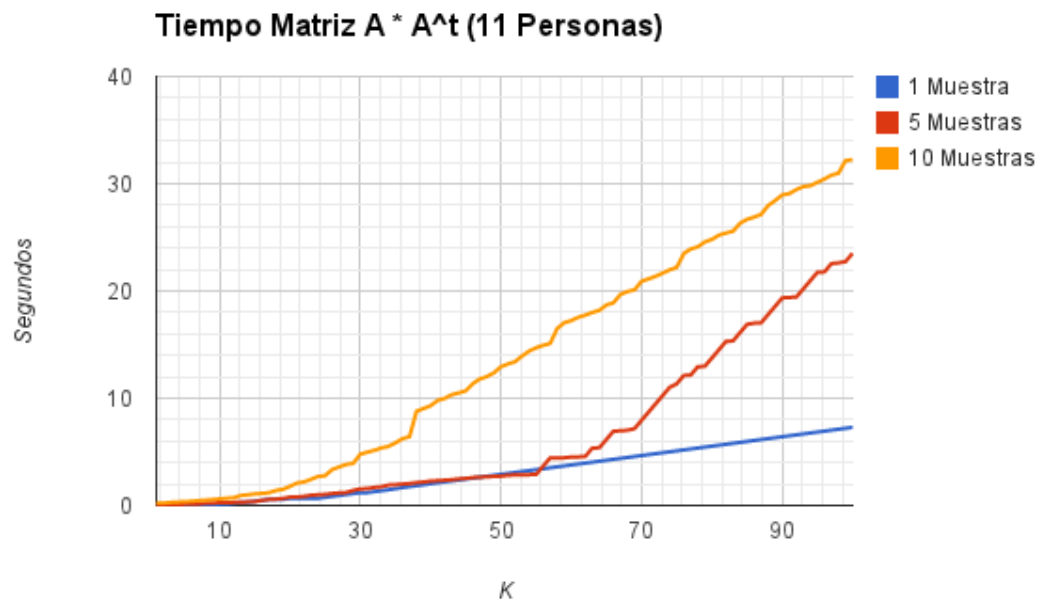
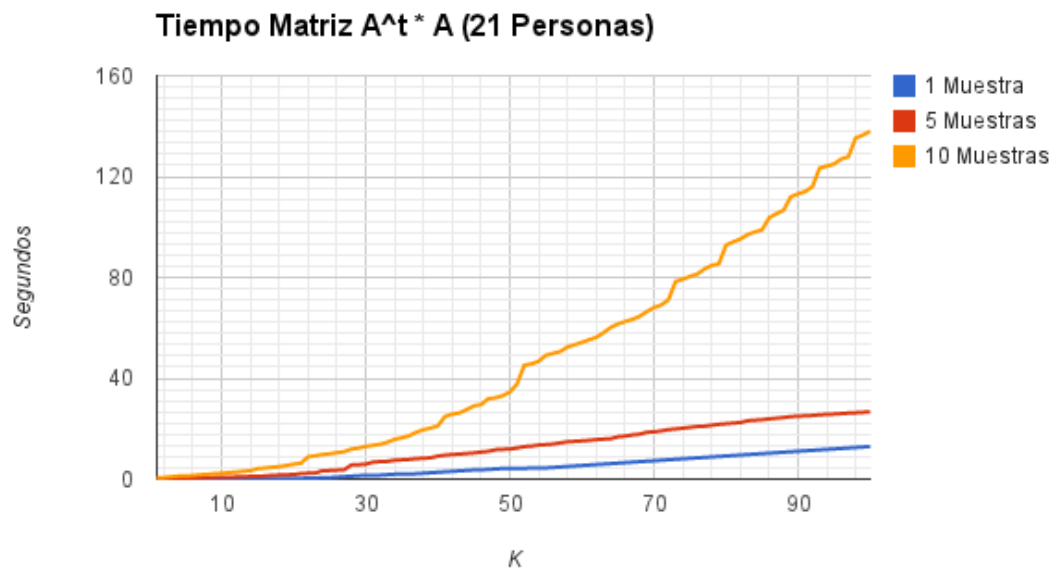
*) Muy irregulares *) Mientras mayor la cantidad de muestras, mayor la efectividad *) tienden a subir mientras sube el k

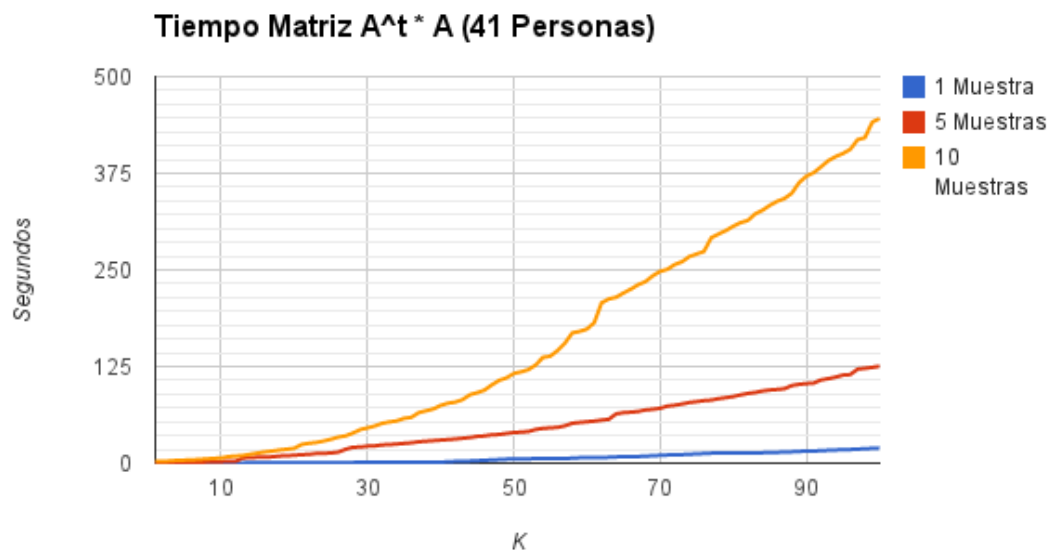
Conclusiones:

EXPLICAR

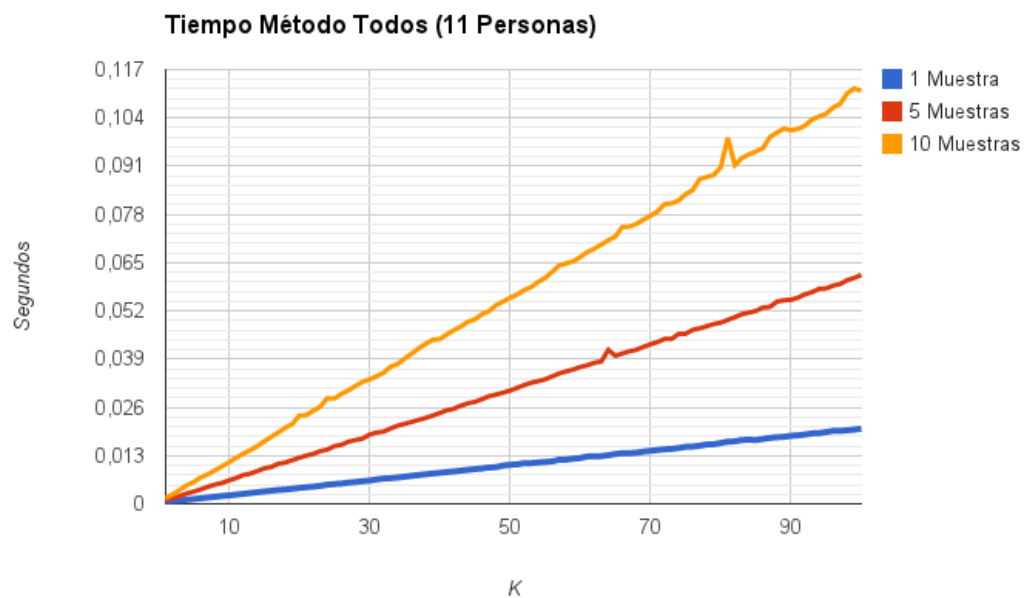
4.1.2. Metodo 1: Utilizando AA^t

Mediciones de TK

Figura 16: Tiempos Matrix AA^t con 11 personas variando KFigura 17: Tiempos Matrix AA^t con 21 personas variando K

Figura 18: Tiempos Matriz AA^t con 41 personas variando K

Mediciones de Ttodos

Figura 19: Tiempos Todos con Matriz AA^t con 11 personas variando K

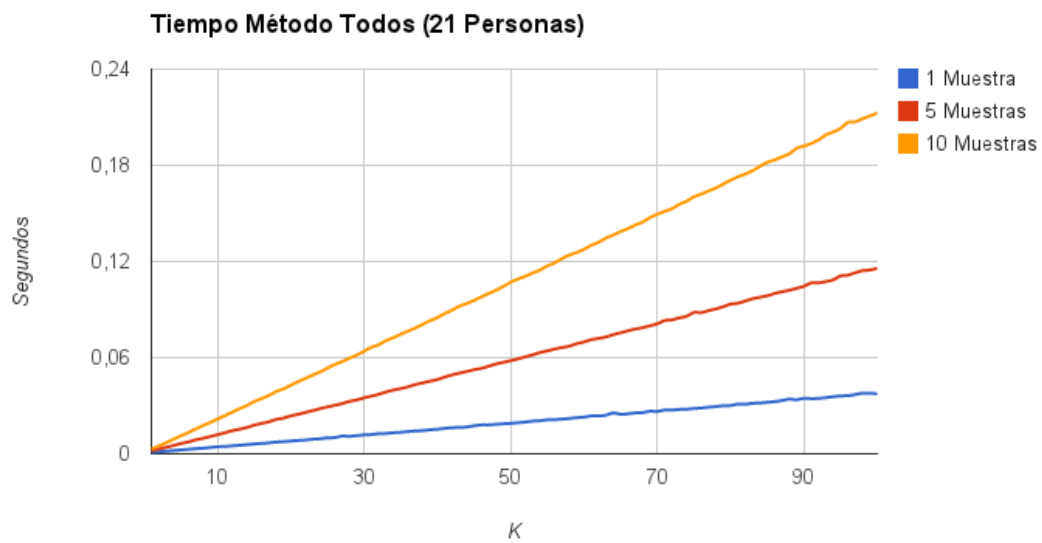


Figura 20: Tiempos Todos con Matrix AA^t con 21 personas variando K

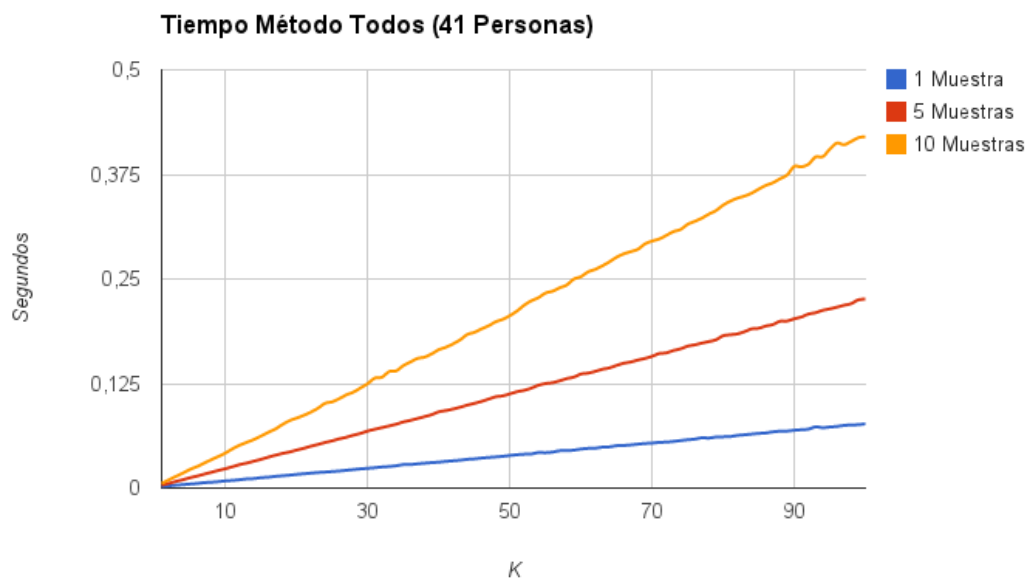
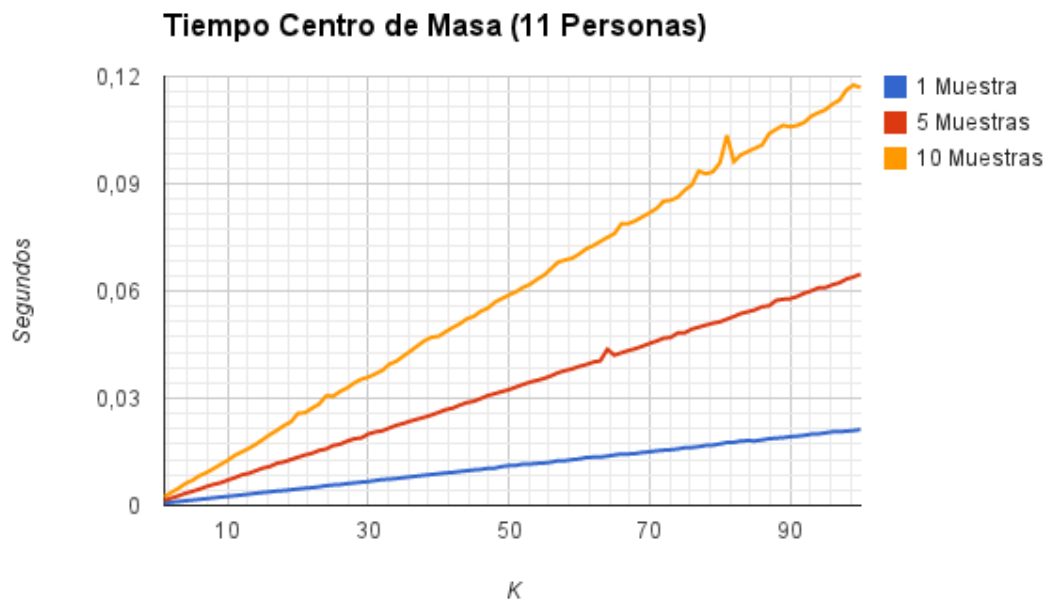
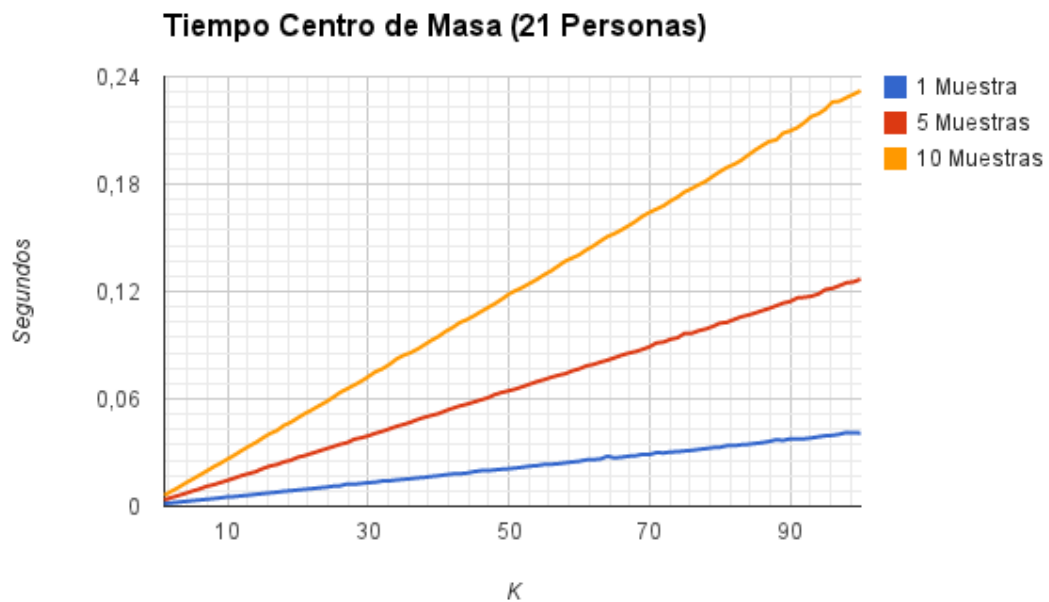


Figura 21: Tiempos Todos con Matrix AA^t con 41 personas variando K

Mediciones de Tcentro

Figura 22: Tiempos Centro con Matrix AA^t con 11 personas variando KFigura 23: Tiempos Centro con Matrix AA^t con 21 personas variando K

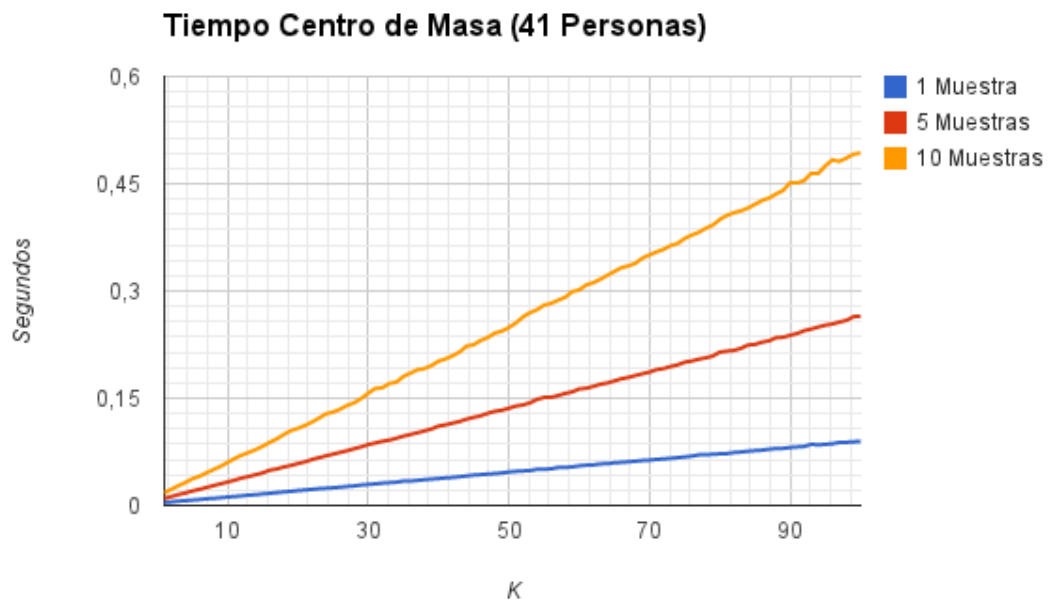


Figura 24: Tiempos Centro con Matrix AA^t con 41 personas variando K

Mediciones de HitsTodos

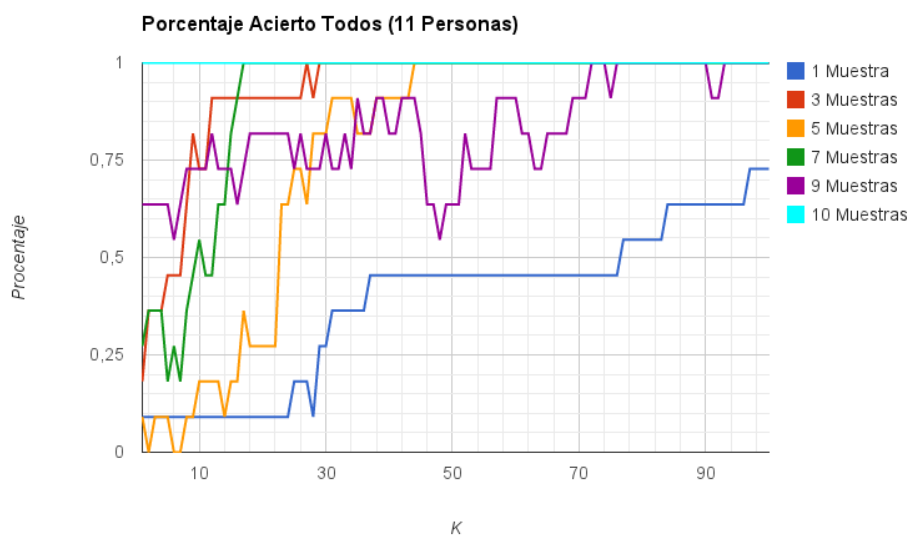


Figura 25: Coeficientes de efectividad vecino más cercano con Matrix AA^t con 11 personas variando K

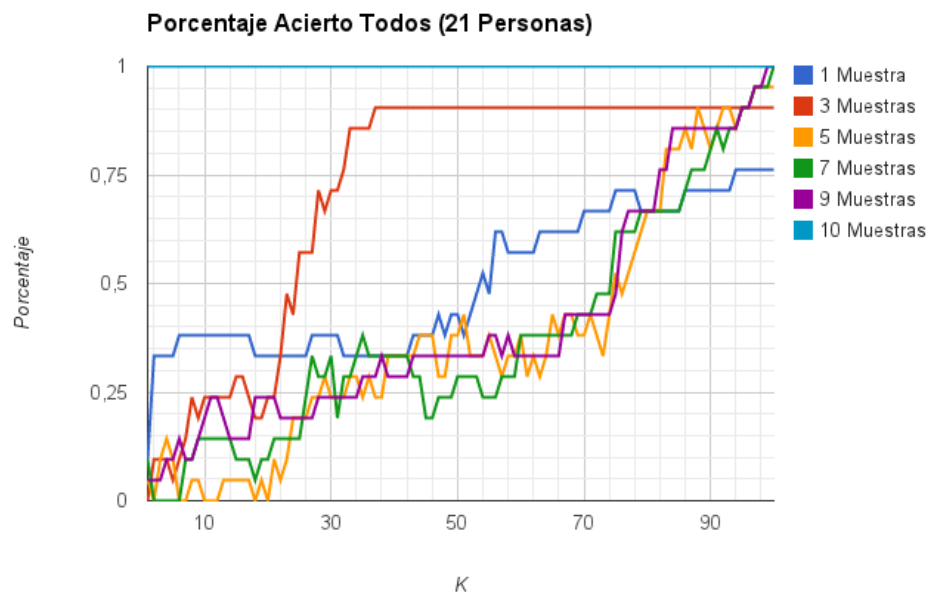


Figura 26: Coeficientes de efectividad vecino más cercano con Matrix AA^t con 21 personas variando K

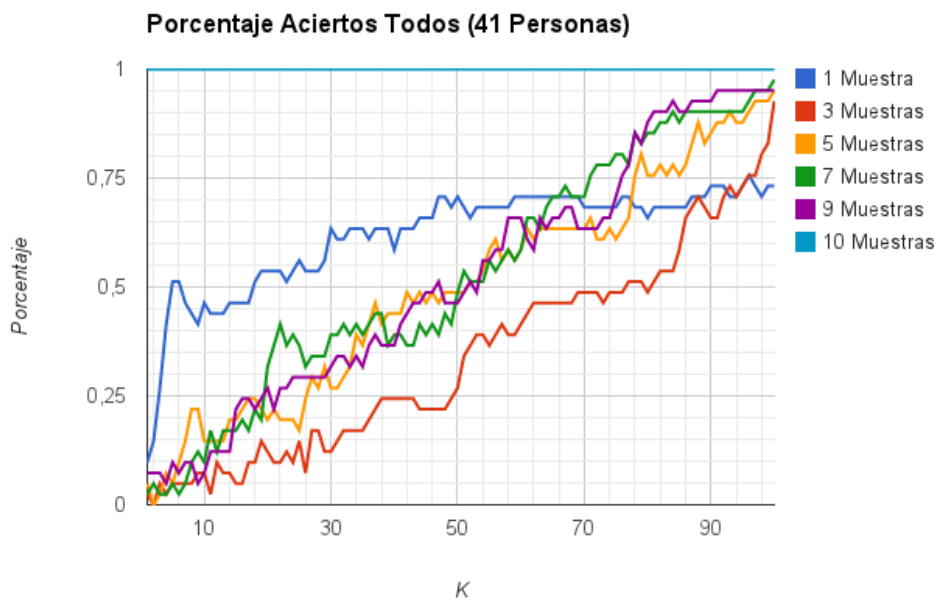


Figura 27: Coeficientes de efectividad vecino más cercano con Matrix AA^t con 41 personas variando K

Mediciones de HitsCentro

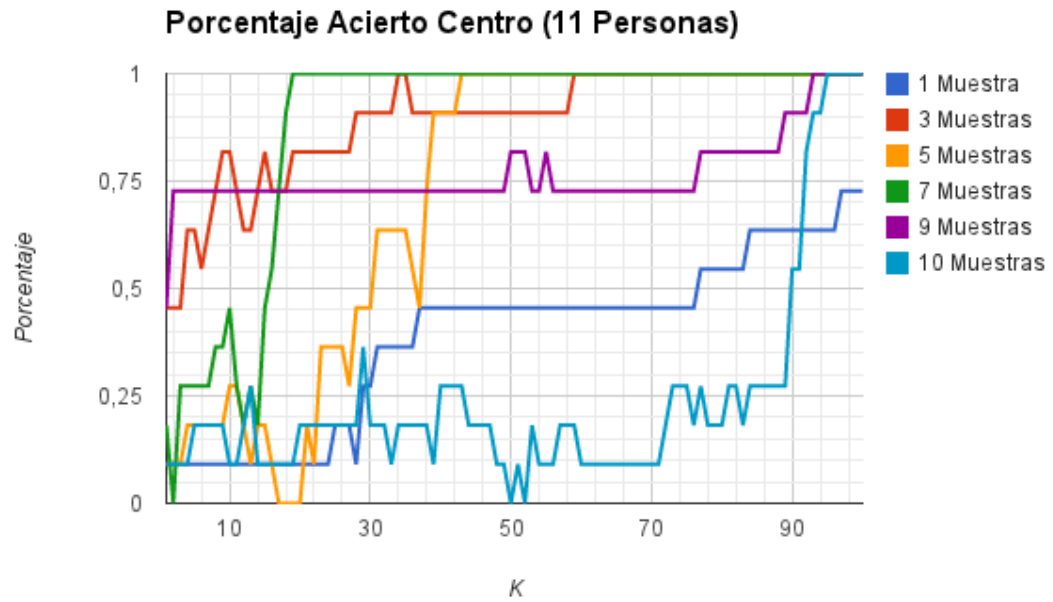


Figura 28: Coeficientes de efectividad de Centros de Masa con Matrix AA^t con 11 personas variando K

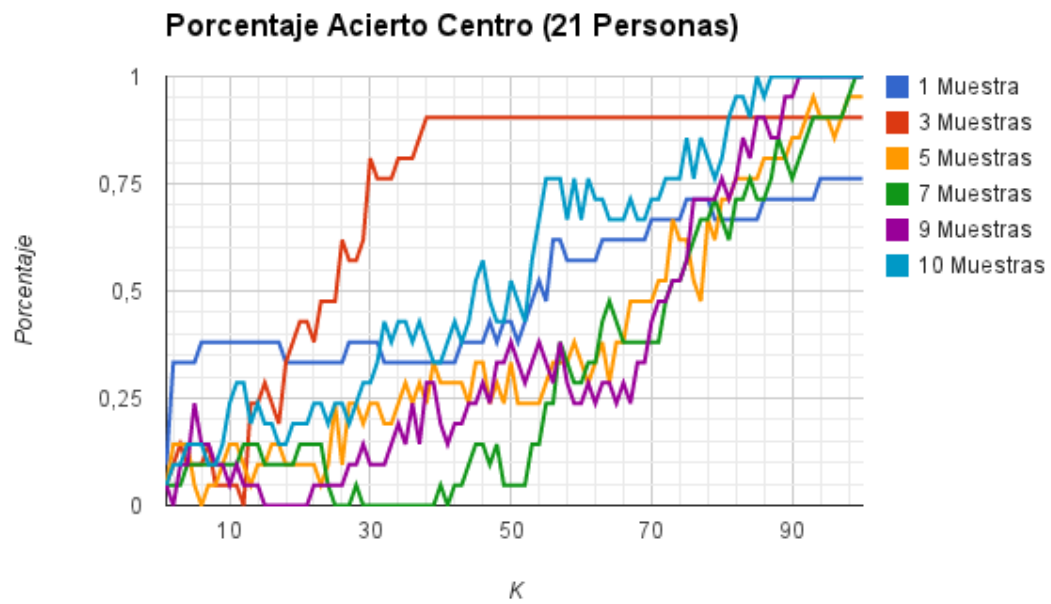


Figura 29: Coeficientes de efectividad de Centros de Masa Matrix AA^t con 21 personas variando K

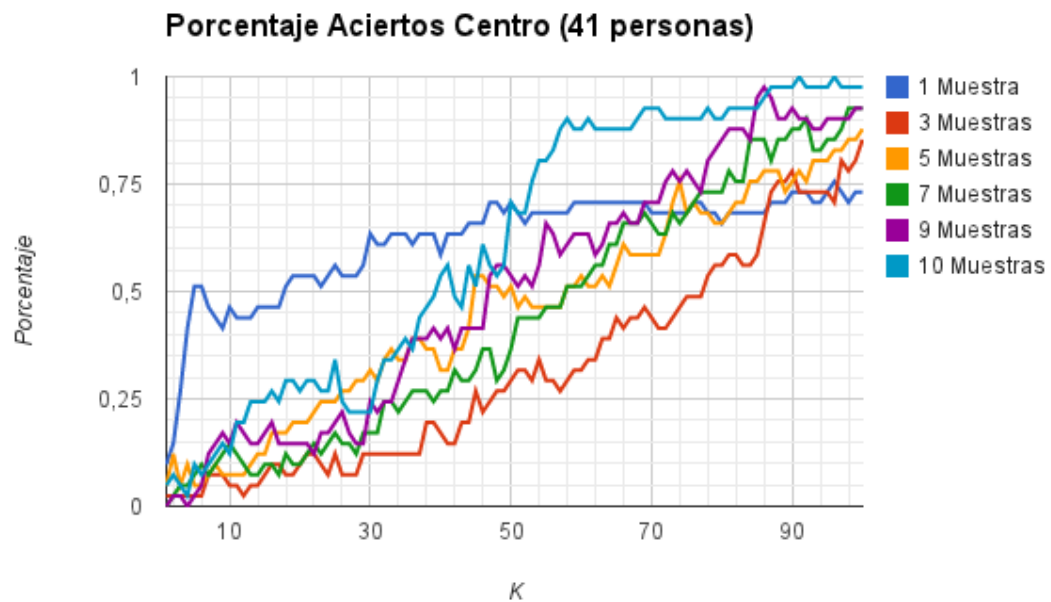


Figura 30: Coeficientes de efectividad de Centros de Masa con Matrix AA^t con 41 personas variando K

Conclusiones:**EXPLICAR****4.2. Experimentacion con Imagenes Full****4.2.1. Metodo 0: Utilizando $A^t A$**

Estos tests no los realizamos ya que el tiempo de ejecucion es muy elevado y ademas vamos a obtener los mismos resultados que a los aplicados con el **metodo 1** por lo demostrado en el lema:

Lema: Si u es autovector de AA^t con λ autovalor asociado, entonces $A^t u \in \mathbb{R}^m$ es autovector de $A^t A$ también con λ autovalor asociado.

el cual esta demostrado en la seccion **Demostraciones**

Notamos que el comportamiento es el mismo que en las imagenes reducidas.

4.2.2. Metodo 1: Utilizando AA^t

Mediciones de TK

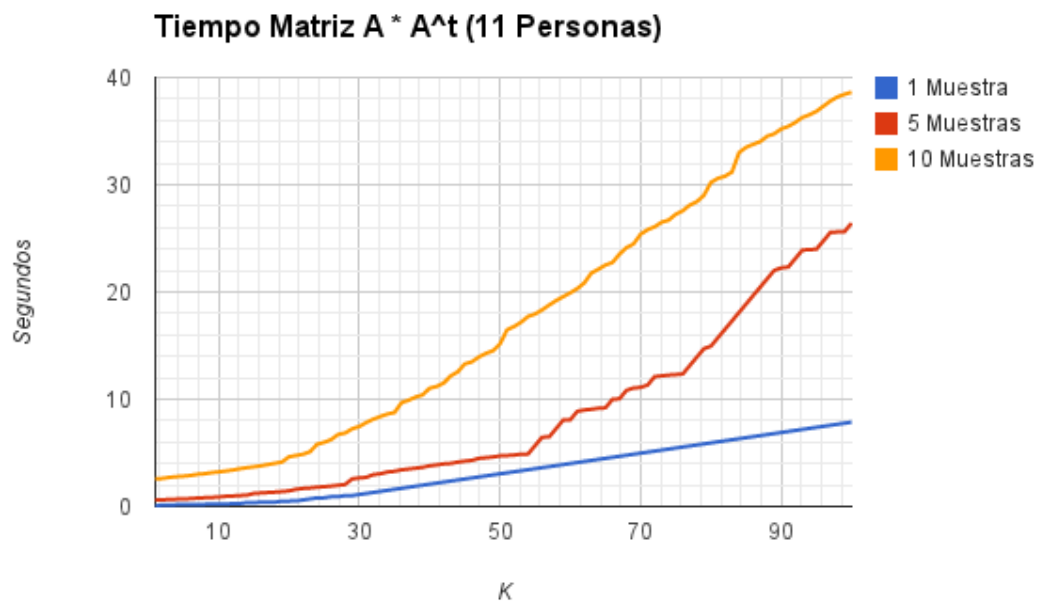


Figura 31: Tiempos de calcular la Matrix AA^t con 11 personas variando K

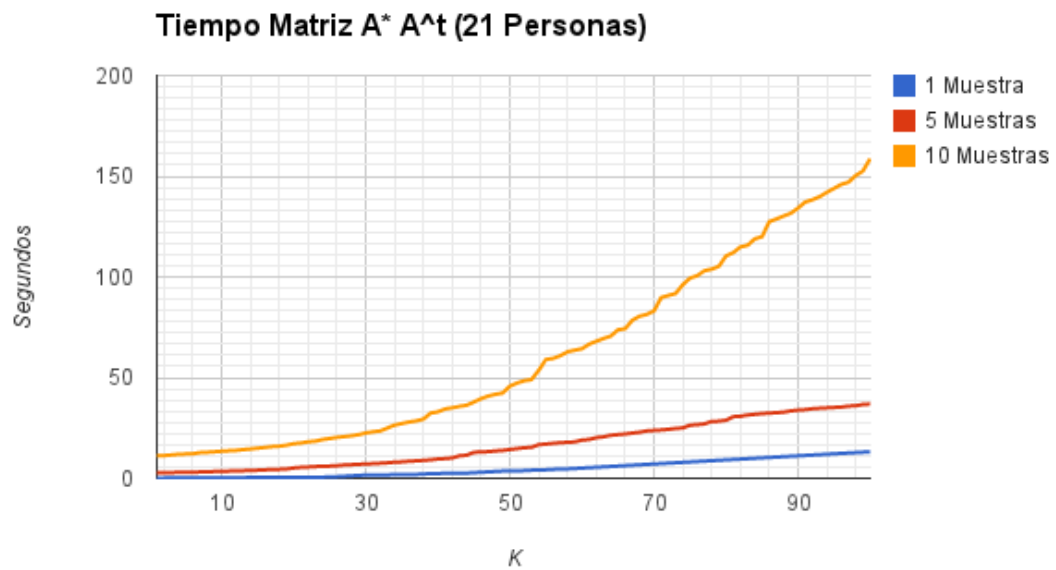


Figura 32: Tiempos de calcular la Matrix AA^t con 21 personas variando K

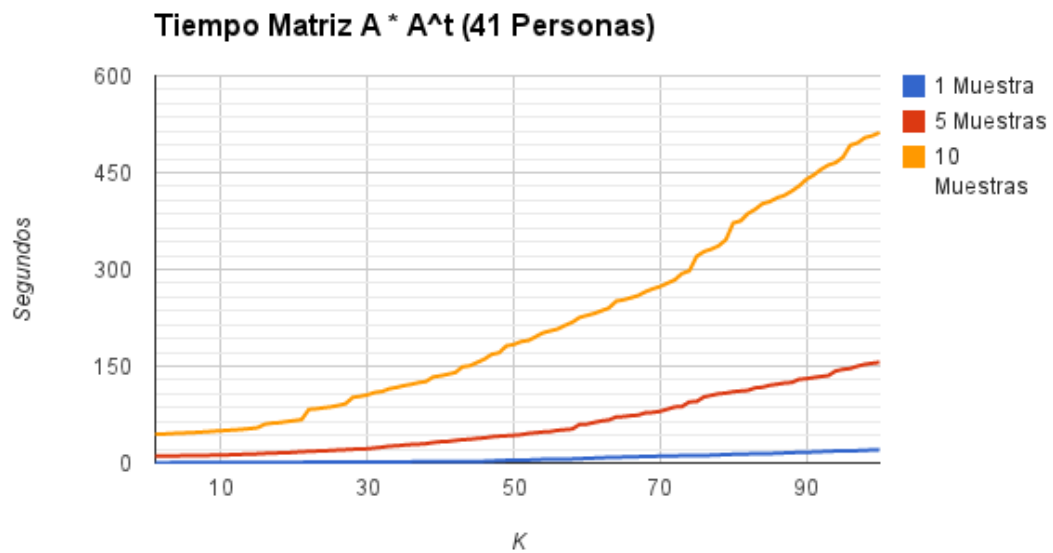


Figura 33: Tiempos de calcular la Matrix AA^t con 41 personas variando K

Mediciones de Ttodos

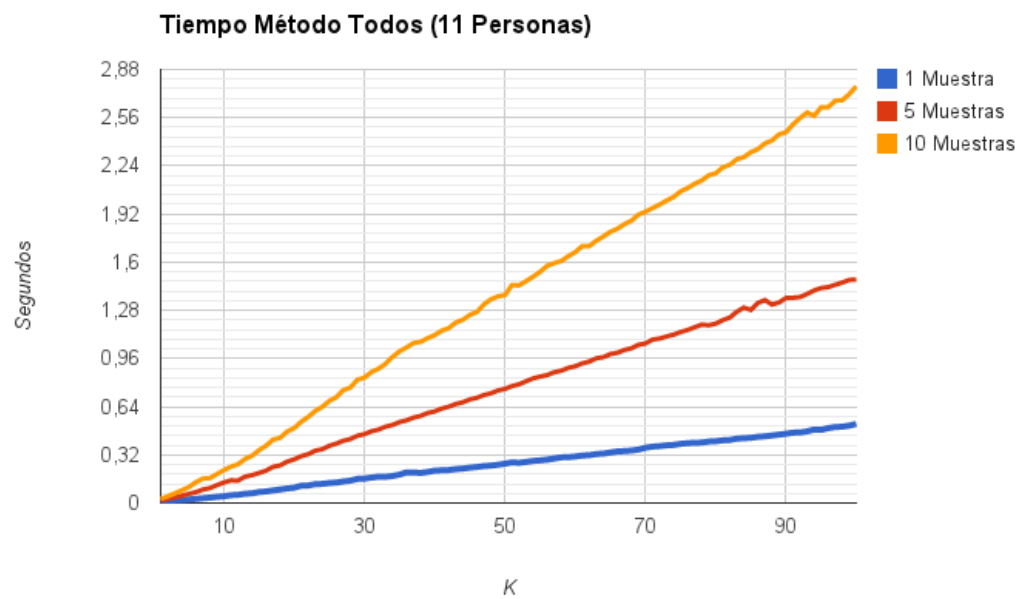


Figura 34: Tiempos de vecino más cercano con Matrix AA^t con 11 personas variando K

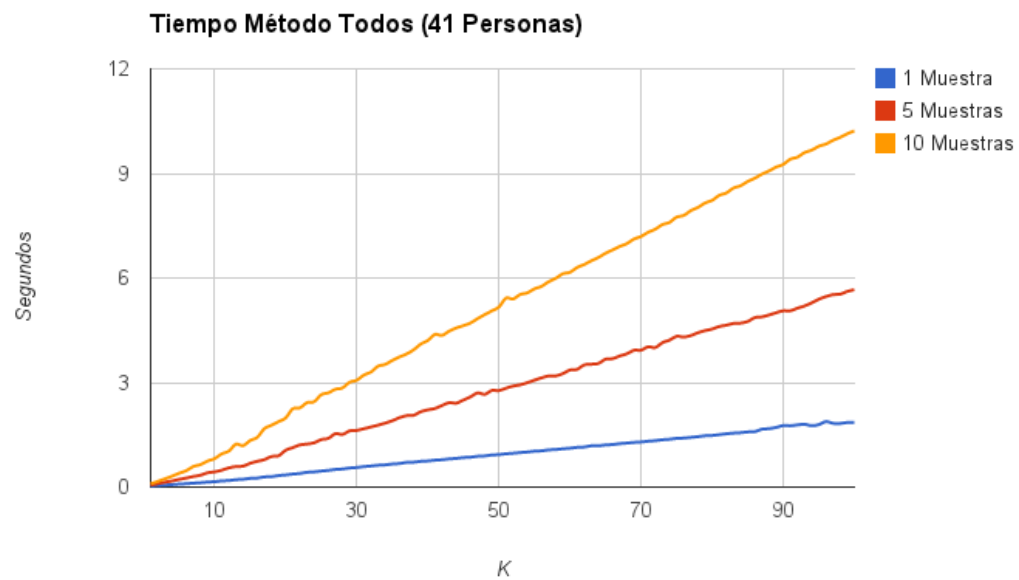


Figura 35: Tiempos de vecino más cercano Matrix AA^t con 21 personas variando K

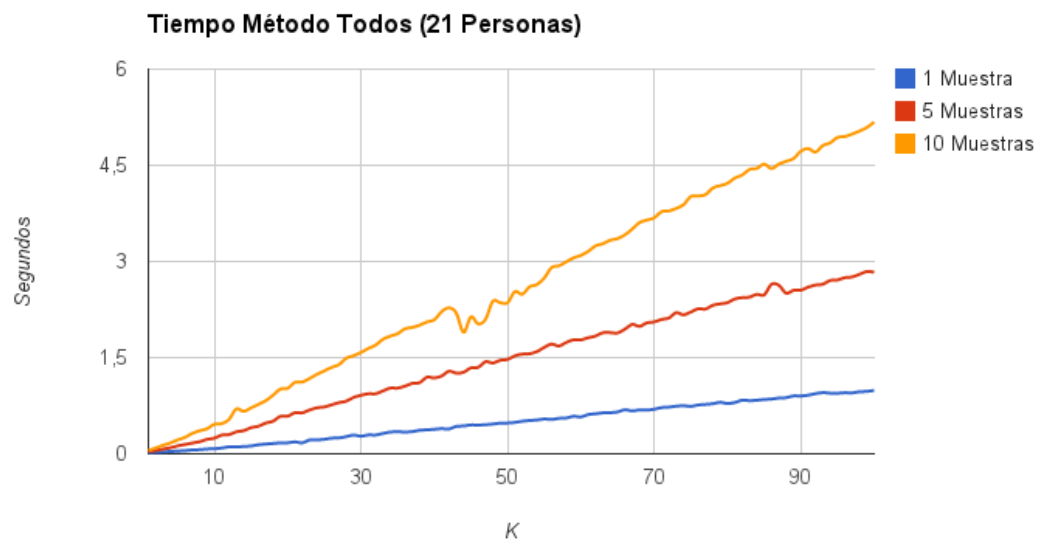


Figura 36: Tiempos de vecino más cercano Matrix AA^t con 41 personas variando K

Mediciones de Tcentro

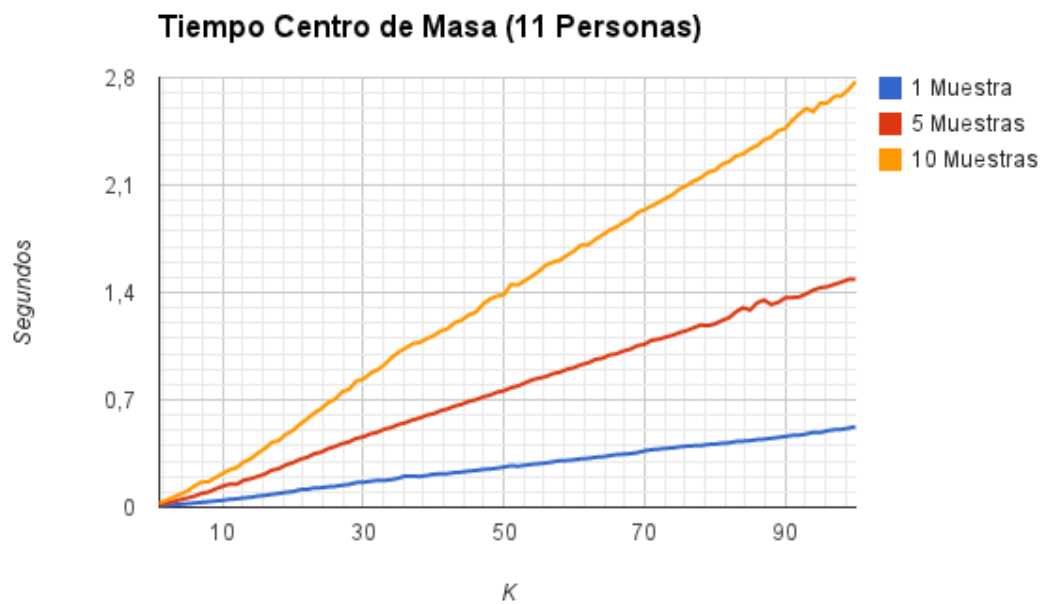


Figura 37: Tiempos de Centros de Masa Matrix AA^t con 11 personas variando K

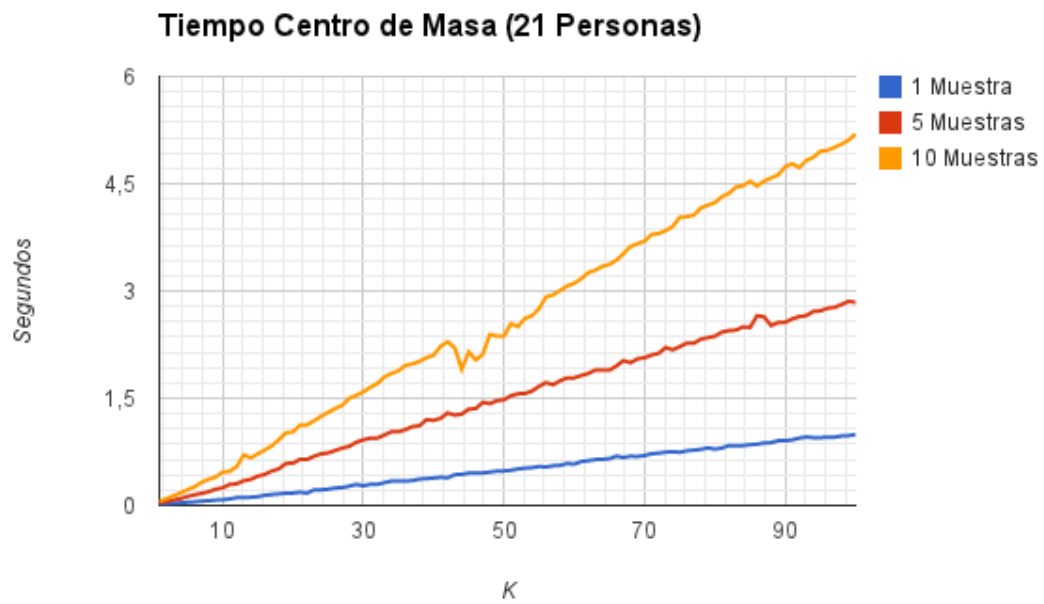


Figura 38: Tiempos de Centros de Masa con Matrix AA^t con 21 personas variando K

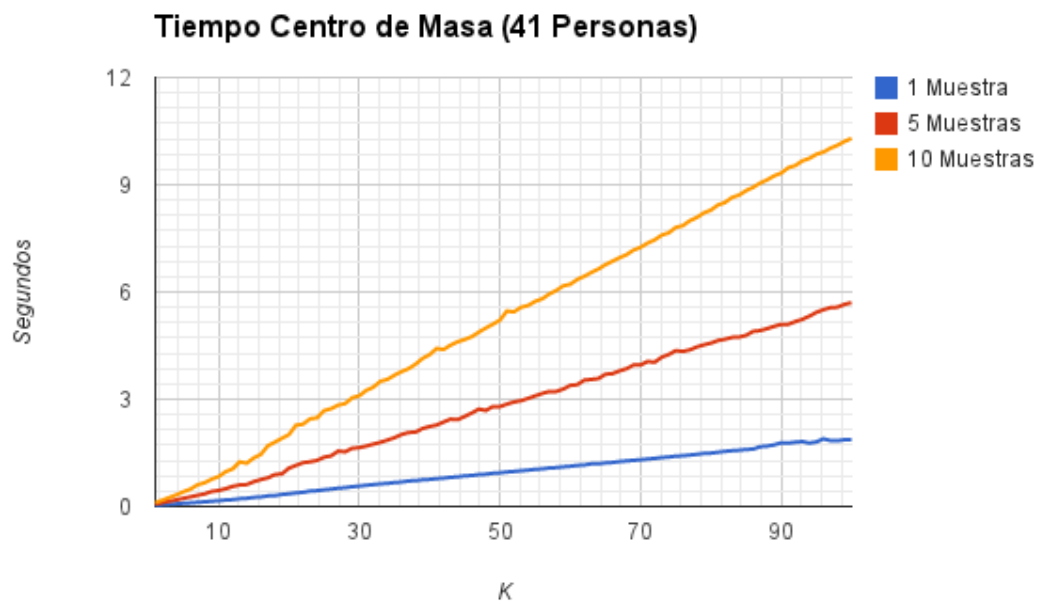


Figura 39: Tiempos de Centros de Masa con Matrix AA^t con 41 personas variando K

Mediciones de HitsTodos.

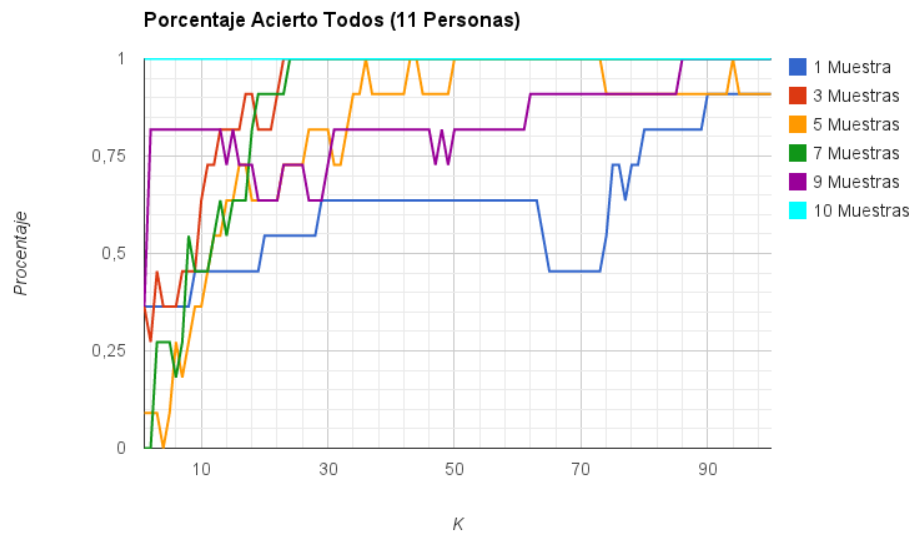


Figura 40: Coeficiente de efectividad de Vecino Más Cercano con Matrix AA^t con 11 personas variando K

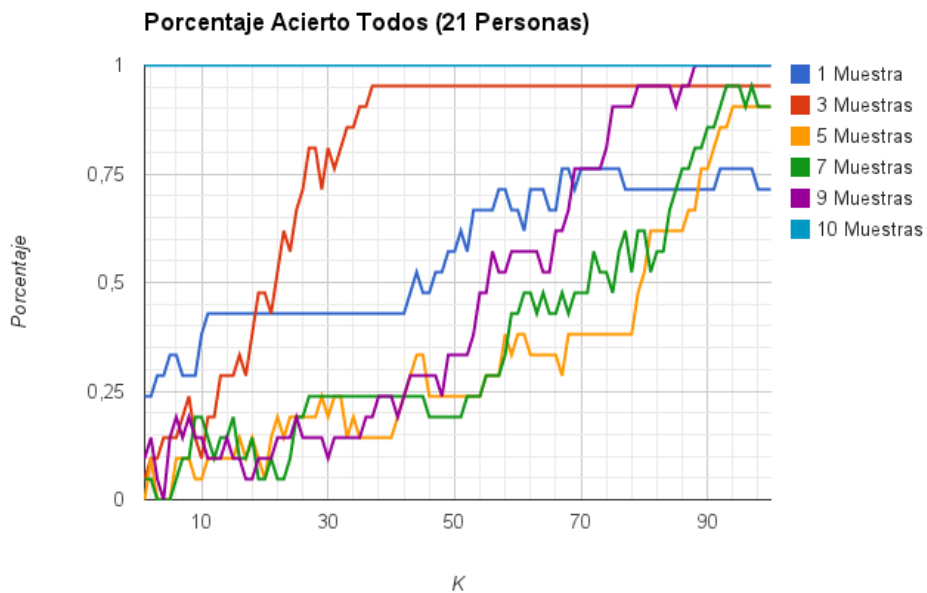


Figura 41: Coeficiente de efectividad de Vecino Más Cercano Matrix AA^t con 21 personas variando K

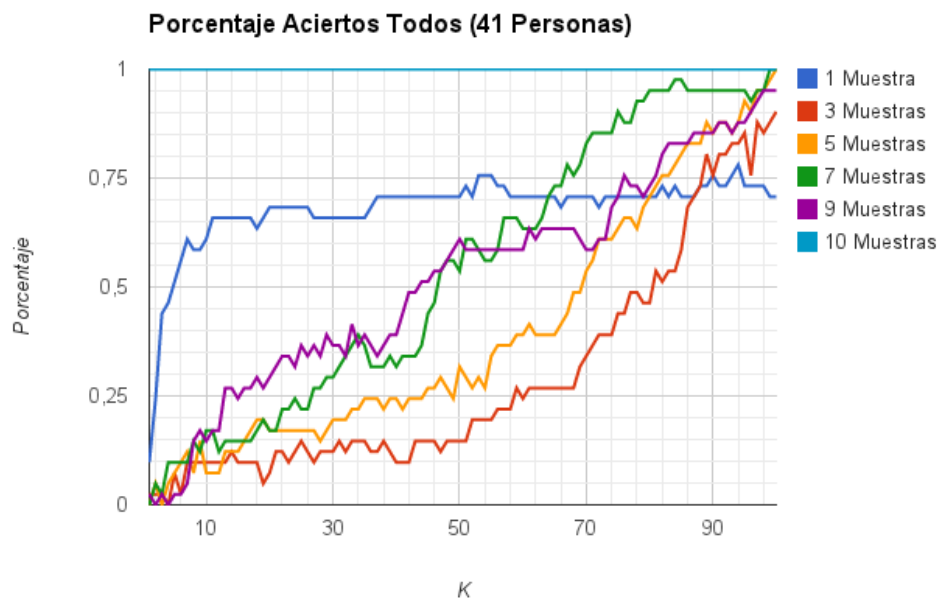


Figura 42: Coeficiente de efectividad de Vecino Más Cercano con Matrix AA^t con 41 personas variando K

Mediciones de HitsCentro

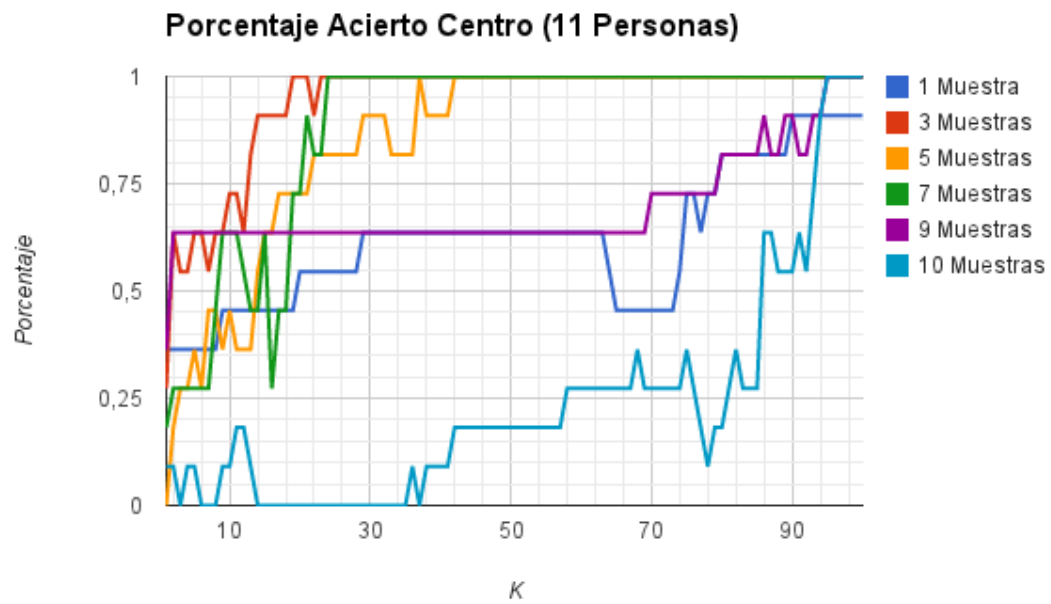


Figura 43: Coeficiente de efectividad de Centros de Masa con Matrix AA^t con 11 personas variando K

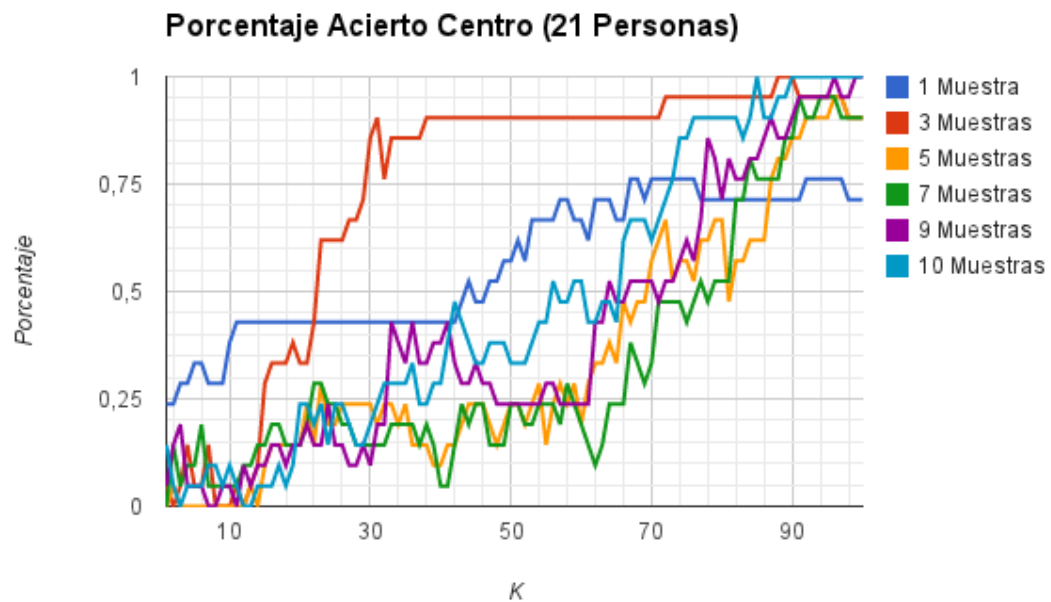


Figura 44: Coeficiente de efectividad de Centros de Masa con Matrix AA^t con 21 personas variando K

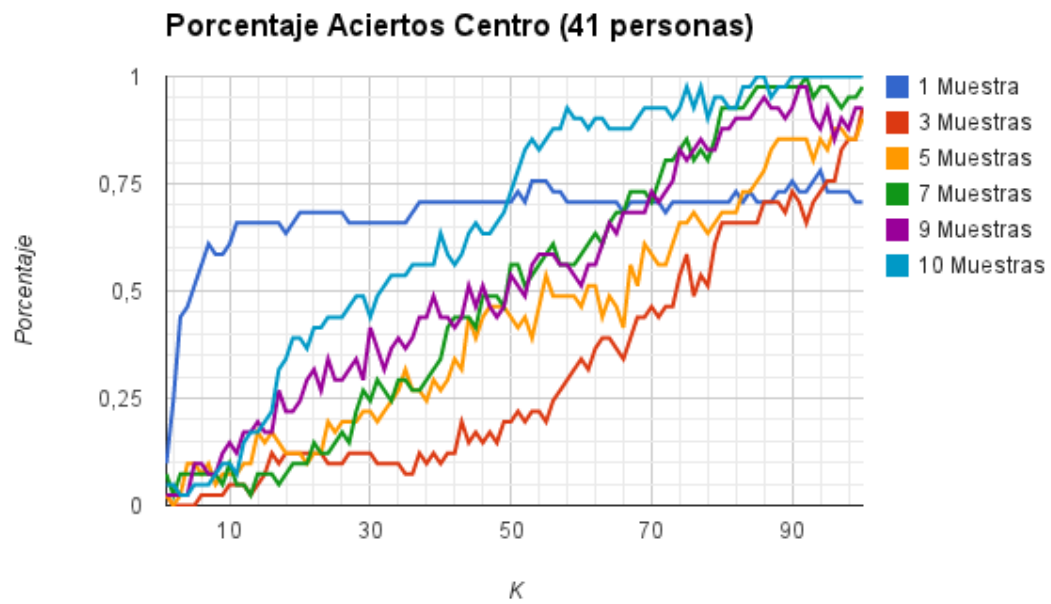


Figura 45: Coeficiente de efectividad de Centros de Masa con Matrix AA^t con 41 personas variando K

Conclusiones:

EXPLICAR

5. Apéndice

5.1. 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 que 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 ítem 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.

5.2. Generador de Tests

Para generar Tests realizamos un algoritmo en Python en el cual recibimos por parametros el k , la cantidad de personas y el metodo a aplicar. Luego variamos la cantidad de personas en un rango de $\{1,11,21,31,41\}$. Para cada una de estas variamos la cantidad de imagenes por persona de en el intervalo de 1 a 10 de manera random y comparamos contra una imagen que no sea una muestra (exceptuando el caso en el que cada persona tiene 10 muestras). El valor de k lo variamos desde adentro del codigo C++ con el fin de ahorrar calculos ya calculados anteriormente.

5.3. Método de compilación

5.3.1. Método 1

Parados en la carpeta /src del proyecto ejecutar

```
$ make
```

De esta forma se limpia, compila y ejecutan los test provistos por la cátedra. Para compilar por separado se puede hacer: **make data.o**, **make functions.o**, **make Matrix.o**, **make main.o**. O tambien se puede borrar haciendo **make clean**. Por defecto al ejecutar **make** el nombre del ejecutable es **caritas**

5.3.2. Método 2

Parados en la carpeta donde se encuentra el ejecutable (por ejemplo /src/)

```
$ ./ejecutable < PATH TEST IN > <PATH SALIDA> <METODO>
```

Donde METODO puede ser

- 0: Utiliza para los calculos $A^t A$
- 1: Utiliza para los calculos AA^t

Donde en PATH SALIDA se escriben los autovalores correspondientes.

5.4. Script para correr casos

Dado el tiempo de ejecución de los casos utilizados para este informe, realizamos un script en python, basadao en el provisto por la cátedra, que ejecuta secuencialmente los casos generados previamente y escribiendo el *stdOutput* de cada caso, en el correspondiente archivo *.console*. // El script ejecuta solo los casos que no tengan archivos de salida ya generados, por lo que se pueden correr múltiples intancias de este en paralelo.

5.5. Equipo de pruebas

Los casos se corrieron sobre un procesador intel i5 de cuatro núcleos sin *hyperthreading* y 16GB de memoria ram. Para no afectar las mediciones y optimizar el tiempo, se corrieron tres casos en paralelo continuamente, por lo que se pudo dar el 100 % de un núcleo a todas las pruebas.

5.6. Referencias bibliográficas

Referencias

- [1] Richard L. Burden and J. Douglas Faires *Numerical Analysis*. 2005.