

## Laboratorio de Métodos Numéricos - Primer Cuatrimestre 2017

### Trabajo Práctico Número 2: Cara a cara

---

#### Introducción

Como plan estratégico para el desarrollo del país, las autoridades nacionales están a días de inaugurar el prometedor *soja valley* que será el pilar económico de los años venideros. Dado que el rimbombante emprendimiento albergará a las industrias más importantes para el país, se necesita un sistema de reconocimiento de trabajadores que ingresen a las instalaciones dado que se estará manejando información confidencial a diario.

Si bien es muy factible que existan mejores formas y más económicas para reconocer diariamente a un trabajador, las autoridades están muy interesadas en desarrollar un sistema biométrico basada en reconocimiento facial por ciertos motivos. En primer lugar, la inauguración de *soja valley* se cubrirá por todos los medios hegemónicos de comunicación y un sistema de ingreso basado en reconocimiento facial es muy vistoso para mostrar publicamente, pudiendo centrarse en las cuestiones estéticas y evitando así entrar en detalles poco claros sobre el funcionamiento del proyecto integral. Además, el *Departamento de Nombres Pegadizos* está motivado con la utilización de un sistema de reconocimiento facial ya que la denominación *mirotucareta* para el sistema, rima perfectamente con otro discutible proyecto anterior.

En este contexto, las autoridades han contactado al Departamento de Computación para el desarrollo del sistema de reconocimiento facial que se utilizará en el flamante *soja valley*. Como punto de partida para la realización de un prototipo, nos han provisto de una base de datos preliminar para poder realizar una prueba de concepto.

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

Las imágenes a considerar se encuentran en un espacio de dimensión alta, lo cual puede traer dificultades para realizar cálculos y para diseñar un algoritmo de reconocimiento que pueda utilizar la información desperdigada por todas las dimensiones con un costo computacional aceptable.

Teniendo en cuenta esto, una alternativa interesante de preprocesamiento es buscar reducir la cantidad de dimensiones de las muestras para trabajar con una cantidad de variables más acotada y, simultáneamente, buscando que las nuevas variables tengan información representativa para clasificar los objetos de la base de entrada. En esta dirección, consideraremos un método de reducción de dimensiones: *Análisis de Componentes Principales* (PCA, por su sigla en inglés).

#### PCA

El método de análisis de componentes principales consiste en lo siguiente. Para  $i = 1, \dots, n$ ,  $x_i \in \mathbb{R}^m$  corresponde a 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

$$M = X^t X$$

a la matriz de covarianza de la muestra  $X$ . Siendo  $v_j$  el autovector de  $M$  asociado al  $j$ -ésimo autovalor al ser ordenados por su valor absoluto, definimos para  $i = 1, \dots, n$  la *transformación característica* de la cara  $x_i$  como el vector  $\mathbf{tc}(x_i) = (v_1^t x_i, v_2^t x_i, \dots, v_\alpha^t x_i) \in \mathbb{R}^\alpha$ , donde  $\alpha \in \{1, \dots, m\}$  es un parámetro de la implementación. Este proceso corresponde a extraer las  $\alpha$  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.

Finalmente, nos concentramos en la evaluación de los métodos. Dado que necesitamos conocer a qué persona corresponde una imagen para poder verificar la correctitud de la clasificación, una alternativa es particionar la base de entrenamiento en dos, utilizando una parte de ella en forma completa para el entrenamiento y la restante como test, pudiendo así corroborar la predicción realizada. Sin embargo, realizar toda la experimentación sobre una única partición de la base podría resultar en una incorrecta estimación de parámetros, como por ejemplo el conocido *overfitting*. Luego, se considera la técnica de *cross validation*, en particular el *K-fold cross validation*, para realizar una estimación de los parámetros del modelo que resulte estadísticamente más robusta.

## Enunciado

Se pide implementar un programa en C o C++ que lea desde archivos las imágenes de entrenamiento correspondientes a distintas personas y que, teniendo en cuenta 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 dos métodos 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 transformación característica, 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  $100 \times 100$  píxeles implicaría trabajar con matrices de tamaño  $10000 \times 10000$ . 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 la siguiente sugerencia y fundamentar como utilizarla en el contexto del trabajo.

- Dada una matriz de covarianza  $M = X^t * X$ , encontrar una relación entre sus autovalores

y sus autovectores con los de la matriz  $\hat{M} = X * X^t$ .

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:  $92 \times 112$  y  $23 \times 28$  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 la performance, mediante las métricas presentadas a continuación, de los algoritmos 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.
- Para cada una de las formas para determinar a que persona corresponde una nueva cara, considerar 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.

La calidad de los resultados obtenidos será analizada mediante diferentes métricas. En particular, la métrica más importante que debe reportarse en los experimentos es la tasa de efectividad lograda, es decir, la cantidad de caras correctamente clasificadas respecto a la cantidad total de casos analizados.

Adicionalmente, se mencionan las siguientes métricas para su potencial uso en el análisis de los experimentos. Se deben utilizar al menos dos de las siguientes métricas, aunque no necesariamente para todos los experimentos realizados.

**Precision:** Es una medida de cuántos aciertos relativos tiene un clasificador dentro de una clase particular. Es decir, dada una clase  $i$ , la *precision* de dicha clase es  $\frac{tp_i}{tp_i + fp_i}$ .

En la anterior fórmula,  $tp_i$  son los *verdaderos positivos* de la clase  $i$ . Es decir, muestras que realmente pertenecían a la clase  $i$  y fueron exitosamente identificadas como tales. En contraposición,  $fp_i$  son los *falsos positivos* de la clase  $i$ . Son aquellas muestras que fueron identificadas como pertenecientes a la clase  $i$  cuando realmente no lo eran.

Luego, la *precision* en el caso de un clasificador de muchas clases, se define como el promedio de las *precision* para cada una de las clases.

**Recall:** Es una medida de qué tan bueno es un clasificador para, dada una clase particular,

identificar correctamente a los pertenecientes a esa clase. Es decir, dada una clase  $i$ , el *recall* de dicha clase es  $\frac{tp_i}{tp_i + fn_i}$ .

En la anterior fórmula,  $fn_i$  son los *falsos negativos* de la clase  $i$ . Es decir, muestras que pertenecían a la clase  $i$  pero que fueron identificadas con otra clase.

Luego, el *recall* en el caso de un clasificador de muchas clases, se define como el promedio del *recall* para cada una de las clases.

**F1-Score:** Dado que *precision* y *recall* son dos medidas importantes que no necesariamente tienen la misma calidad para un mismo clasificador, se define la métrica F1 para medir un compromiso entre el *recall* y la *precision*. La métrica F1 se define como  $2 * precision * recall / (precision + recall)$ .

**Kappa de Cohen:** Es una medida para indicar cuánto concuerdan dos clasificadores sobre un mismo set de datos. Dicha medida se define como  $\kappa = (p_o - p_a) / (1 - p_a)$ . Donde  $p_o$  es la probabilidad observada de que los dos clasificadores concuerden y  $p_a$  es la probabilidad aleatoria de que lo hagan.

Esta métrica puede utilizarse para determinar si el problema contiene ejemplos particularmente complicados, porque por ejemplo ningún clasificador lo reconoce correctamente.

Por último, se pide diseñar, implementar y experimentar sobre alguna aproximación para detectar si una imagen contiene o no una cara. Es decir, se quiere intentar responder la pregunta de si el sistema es capaz no solo de reconocer caras de personas con las que fue entrenado, sino si es posible discernir entre una imagen de una cara de una persona que no se encontraba en la base de entrenamiento y una imagen con un objeto o contenido que no sea una cara.

## Programa y formato de archivos

Se deberán entregar los archivos fuentes que contengan la resolución del trabajo práctico. El ejecutable tomará dos parámetros por línea de comando, que serán el archivo de entrada y el archivo de salida.

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  $112 \times 92$ , 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* numeros 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 *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 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 las  $k$  raíces cuadradas de los autovalores de mayor magnitud, con un valor 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 25 de Mayo de 2017, hasta las 23:59 hs., enviando el trabajo (informe+código) a [metnum.lab@gmail.com](mailto: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 26 de Mayo de 2017, de 17:30 a 18:00 hs.