Métodos Numéricos
Primer Cuatrimestre 2023
Trabajo Práctico 2

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Procesamiendo de Imágenes, análisis de componentes principales y reducción de la dimensionalidad

---

Introducción

El procesamiento de imágenes es un campo relacionado al procesamiento de señales y el reconocimiento de patrones donde se trabaja con conjuntos de datos de imágenes, es decir arreglos numéricos del tipo matrices, cuyos elementos codifican los colores de los píxeles. Algunas de las tareas que tratan en este campo pueden ser, detección de rostros en imágenes, compresión de imágenes y cómputo de similitud entre imágenes.

En este trabajo práctico utilizaremos un conjunto de imágenes de rostros y realizaremos reducción de dimensionalidad de los datos mediante análisis de componentes principales. Esto permite proyectar el espacio de imágenes en una nueva base en la cual se organizan distintos atributos espaciales según su importancia. Esta representación sirve para hacer compresión de una imagen al descartar información que tiene poco impacto en la estructura de la imagen. También representando las imágenes en este espacio permite comparar imágenes y realizar análisis de similaridad buscando poder agrupar las imágenes por individuo.

Se cuenta con un conjunto de imágenes de rostros de N personas en escala de grises, todas con el mismo tamaño y resolución. A su vez, para cada una de las personas se cuentan con M imágenes distintas. Concretamente, se tiene un conjunto $S = \{(A_0, l_0), (A_1, l_1) \ldots (A_n, l_n)\}$ donde $A_i \in A \subseteq \mathbb{R}^{a \times b}$ el conjunto de imágenes que contiene N identidades con M imágenes por cada una siendo $|I| = MN$ y donde $l_i \in L \subseteq \mathbb{N}, l_i \leq N - 1$ es un identificador de identidad asociado al rostro de la imagen A_i

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

Teniendo en cuenta esto, una etapa habitual de preprocesamiento se basa en 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 diferenciar los objetos de la base de entrada. En esta dirección, estudiaremos dos método de reducción de dimensiones: *Análisis de Componentes Principales* (PCA, por su sigla en inglés) y una variante especial para tratar imágenes 2DPCA.

Métodos

Covarianza y correlación

Dados dos vectores $\mathbf{x}$ y $\mathbf{y}$ definimos la covarianza como

$$Cov(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \mu_x) \cdot (\mathbf{y} - \mu_y)}{n - 1} \tag{1}$$

Donde μ representa el valor medio del vector. Esta fórmula se puede interpretar como el producto interno de los vectores "centrados" y normalizada por la dimensión del vector menos uno.

Luego, definimos la correlación como una covarianza normalizada para que su rango sea entre -1 y 1.

$$Corr(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \mu_x) \cdot (\mathbf{y} - \mu_y)}{\sqrt{(\mathbf{x} - \mu_x) \cdot (\mathbf{x} - \mu_x)(\mathbf{y} - \mu_y) \cdot (\mathbf{y} - \mu_y)}} \tag{2}$$

Esta expresión es equivalente al coseno del ángulo entre los vectores $\cos\theta_{\mathbf{xy}}$. Alta correlación implica vectores paralelos, con correlación positiva para vectores en la misma dirección y negativa en direcciones opuestas, y correlación nula implica vectores perpendiculares.

Si se consideran los vectores columna $\mathbf{x_i} - \mu_{x_i}$ formando la matriz $\mathbf{X}$, la covarianza entre todas las columnas se llama *matriz de covarianza* y se expresa como:

$$\mathbf{C} = \frac{\mathbf{X}^t\mathbf{X}}{n - 1} \tag{3}$$

Esta matriz es simétrica y semidefinida positiva por lo cual sus autovalores son no negativos.

Análisis de componentes principales

Consideremos una matriz de datos $\mathbf{X} \in \mathbf{R}^{m \times n}$ cuyas filas representan m vectores de dimensionalidad n. El análisis de componentes principales (PCA[1]) busca encontrar un cambio de base de los vectores, de tal manera que las dimensiones se ordenen en componentes que expliquen la varianza en los datos de mayor a menor. Para hacer esto se realiza una descomposición en autovectores y autovalores de la matriz de covarianza de los datos:

$$\mathbf{C} = \mathbf{V}\mathbf{D}\mathbf{V}^t \tag{4}$$

Donde $\mathbf{V}$ es la matriz con los autovectores en las columnas y $\mathbf{D}$ una matriz diagonal con los autovalores. Estos autovalores son la varianza que captura cada nueva dirección dada por los

[1] Capítulo 12, Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.

autovectores. La matriz $\mathbf{V}$ permite transformar los datos $\mathbf{X}$ en la nueva base. Definimos a los nuevos datos rotados o transformados $\mathbf{R}$ mediante la operación

$$\mathbf{R} = \mathbf{X}\mathbf{V}$$

Dependiendo de la situación es posible reducir la dimensionalidad de los datos utilizando solo las k componentes principales que se deseen, es decir utilizando los primeros k vectores columna de la matriz $\mathbf{V}$.

El método de análisis de componentes principales aplicado al conjunto imágenes se puede expresar de la siguiente forma. Consideremos $x_i \in \mathbf{R}^n$ a la i-ésima imagen de nuestra base de datos almacenada como un vector de largo n la cantidad de píxeles. Podemos apilar todas las imágenes como filas en una matriz $\mathbf{X} \in \mathbf{R}^{m \times n}$. Luego sea $\mu_j = \sum_{i=1}^{m} X_{ij}/m$ el promedio de cada dimensión-pixel de las imágenes. Definimos la matriz centrada $\mathbf{X_c}$ que contiene en la i-ésima fila al vector $(x_i - \mu)^t$, y $\mathbf{C} = \mathbf{X_c}^t\mathbf{X_c}/(n-1)$ a la matriz de covarianza. Definimos a la imagen x_i proyectada en el espacio de menor dimensión como el vector $z_i = (v_1^t\, x_i, v_2^t\, x_i, \ldots, v_k^t\, x_i) \in \mathbf{R}^k$ donde v_j es el autovector de $\mathbf{C}$ asociado al j-ésimo autovalor al ser ordenados de mayor a menor y k define cuantas componentes se utilizan. En forma matricial $z_i = x_i V k$ donde Vk se obtiene a partir de las primeras k columnas de V la matriz de autovectores. Este proceso corresponde a proyectar la imagen sobre las k primeras *componentes principales*. La intención es que z_i resuma la información más relevante de la imagen, descartando las dimensiones de poca varianza que resultan en detalles o las regiones de la imagen que no aportan rasgos distintivos.

2DPCA

Una de las desventajas de aplicar PCA es que las imágenes se transforman en vectores 1D para construir la matriz de datos $\mathbf{X}$ de donde se computa la covarianza y los autovectores. Esto implica tener que hacer el calculo de autovectores sobre una matriz de dimensionalidad considerablemente alta (n píxeles).

La variante 2DPCA [2] considera una imagen en su espacio original como una matriz $A \in \mathbb{R}^{a \times b}$ y le asocia un *feature vector* $Y \in \mathbb{R}^a$, mediante la transformación lineal $Y = AX$ siendo $X \in \mathbb{R}^b$ el vector que maximiza la dispersión de los *features vectors* obtenidos para todo el conjunto de imágenes. Se puede ver que la elección óptima para el vector de proyección X se corresponde con el autovector de máximo autovalor de la denominada *image covariance matrix*

$$\mathbf{G} = \frac{1}{n}\sum_{j=1}^{n} (\mathbf{A_j} - \bar{\mathbf{A}})^{\mathbf{T}}(\mathbf{A_j} - \bar{\mathbf{A}}) \tag{5}$$

siendo $\bar{\mathbf{A}}$ la imagen promedio del conjunto de imágenes. El tamaño de esta matriz resulta menor que la de covarianza asociada en el cálculo de PCA.

[2] Yang, J., Zhang, D., Frangi, A. F., & Yang, J. Y. (2004). Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE transactions on pattern analysis and machine intelligence, 26(1), 131-137.

Si tomamos $\mathbf{V} = [Y_1 \dots Y_b]$ la matriz de feature vectores para la imagen $\mathbf{A}$ y $\mathbf{U} = [X_1 \dots X_b]$ la base de autovectores ortonormales de $\mathbf{G}$ entonces $\mathbf{V} = \mathbf{AU}$ y por lo tanto $\mathbf{A} = \mathbf{VU}^T$ por ser U ortogonal. A su vez, como el producto de matrices puede descomponerse como suma de productos externos:

$$\mathbf{A} = \mathbf{VU}^T = \sum_{j=1}^{b} \mathbf{Y}_j \mathbf{X}_j^T \tag{6}$$

donde cada $\mathbf{Y}_j \mathbf{X}_j^T$ es una subimagen de $\mathbf{A}$. Al igual que en el método de PCA, en general se suelen considerar k componentes principales (ahora vectores de proyección en lugar de escalares) de manera de poder reconstruir aproximadamente la imagen original A a partir de k *subimagenes*. Es decir, $\mathbf{A} = \sum_{j=1}^{k} \mathbf{Y}_k \mathbf{X}_k^T$.

Por último, para calcular la representación en baja dimensionalidad de una imagen $\mathbf{A}_i$ computamos $\mathbf{Z}_i = (\mathbf{A}_i\mathbf{X}_1, \mathbf{A}_i\mathbf{X}_2, \dots, \mathbf{A}_i\mathbf{X}_k) \in \mathbb{R}^{a \times k}$ donde cada $\mathbf{A}_i\mathbf{X}_k$ representa el k-ésimo *feature vector* de la imagen $\mathbf{A}_i$. Notar que a diferencia de PCA en este caso la proyección genera matrices $a \times k$. Para utilizarlas y comparar dos imágenes utilizando esta representación consideraremos el vector $z_i \in \mathbb{R}^{ak}$ a partir de la matriz aplanada.

Matriz de similaridad

Dado un conjunto de datos $\mathbf{X} \in \mathbf{R}^{m \times n}$ con m datos y n atributos, se define una matriz de similaridad[3] a una matriz de $\mathbf{R}^{m \times m}$ que computa una función sobre cada par de datos ij.

$$D_{ij} = f(X_i, X_j) \tag{7}$$

La similaridad que vamos a utilizar viene dada por la matriz de correlación R a partir de la de covarianza.

$$\mathbf{C} = (\mathbf{X_c}\mathbf{X_c}^t))/(n-1) \tag{8}$$

$$R_{ij} = C_{ij}/\sqrt{C_{ii}C_{jj}} \tag{9}$$

Experimentación

Buscamos analizar y contrastar los métodos en estudio para conseguir representaciones que sean buenas para discriminar las identidades de los rostros de las imágenes y, a su vez, que sean útiles para codificar imágenes con menos información. Para esto vamos a explorar configuraciones de parámetros (k) y métodos (PCA vs 2DPCA) que obtenga un buen balance entre la calidad de las representaciones y el tiempo de cómputo. Nos va a interesar comparar también los resultados usando las imágenes en su representación original de pixeles.

[3]dependiendo el caso también se llama de disimilaridad o distancia

Enunciado

Implementación

1. Método de la potencia con deflación

   a) Implementar en `C++` el método de la potencia con deflación para el cómputo de autovectores y autovalores. Como argumentos de entrada debe tomar un archivo de texto con la matriz, la cantidad de iteraciones y la tolerancia para la convergencia. Como salida debe entregar un archivo de texto con los autovalores y otro con los autovectores como columnas de una matriz.

   b) Realizar tests para verificar la implementación del método en casos donde los autovalores y autovectores sean conocidos se antemano y reportar los resultados más destacados encontrados (incluso aquellos que puedan resultar anómalos). También, puede ser de utilidad contrastar con alguna librería de cálculo numérico (Numpy, Scipy).

Experimentación

2. Procesamiento de imágenes

   a) Implementar una función (Python) que dado el conjunto de imágenes obtiene la representación z en el espacio de menor dimensión para PCA y 2DPCA para un determinado k. Utilizar la implementación de autovectores previa. Notar que no es necesario volver a calcular todos los autovalores y autovectores para cada valor de k.

   b) Graficar el valor de los autovalores ordenados de mayor a menor en función de la componente principal para PCA y 2DPCA.

   c) Observar las denominadas "eigenfaces" para PCA y 2DPCA.

   d) Para algunas imágenes del conjunto de datos, regenerar los rostros para distintos valores de k para PCA y 2DPCA.

3. Análisis de similaridad

   a) Visualizar la matriz de correlación para el conjunto de datos original (centrado) y para los datos en el espacio z para dos valores de k distintos (uno bajo y otro más grande). Al usar las imágenes en el orden que vienen los nombres debe aparecer una estructura de bloques. Realizarlo para PCA y 2DPCA, teniendo en cuenta que para este último el espacio z son matrices, por lo que puede ser necesario tener que aplanarlas para poder luego evaluar la matriz de correlación.

   b) Nos interesa medir la calidad de la similaridad entre imágenes de una misma persona contra imágenes de distintas personas. Graficar en una misma figura los valores de las siguientes métricas para un rango interesante de valores de k para PCA y 2DPCA.

      1) Métrica mismo: El promedio de los valores de similaridad para las comparaciones entre imágenes de una misma persona.

2) Métrica distintos: Idem pero para todas las comparaciones de imágenes de diferentes personas

c) Para analizar la calidad de las imágenes comprimidas, proponer una métrica que permita evaluar el error de compresión (consultarla con los docentes previamente). Probar dejar fuera del conjunto a una persona antes de hacer PCA/2DPCA ¿Qué ocurre con la métrica si se utilizan imágenes de rostros fuera del conjunto que se usa para calcular?

d) Estudiar la complejidad de los métodos evaluando, principalmente, el costo computacional como el tiempo de ejecución de los métodos. Adicionalmente, pensar que ocurre con la cantidad de información que requiere almacenarse para guardar las representaciones en cada caso.

Importante

Las consignas de C++ están pensadas para resolverse utilizando la biblioteca de álgebra lineal *Eigen*. La forma de interfacear los programas en C++ con Python para poder hacer la experimentación es libre. La opción más directa es leyendo y escribiendo archivos de texto con las matrices de entrada y salida. Por ejemplo, se pueden generar datos en Python, escribir las matrices como archivos de texto, ejecutar el programa en C++ con el nombre del archivo de entrada y de salida, y luego en Python leer el archivo y armar los resultados como arreglos de Numpy. Otra alternativas más complejas, donde no se utilizan archivos de texto, y la interfaz es más transparente involucra usar *ctypes* o *pybind11*. Pueden consultar sobre esto si desean.

Forma y Fecha de entrega

1. Se debe entregar un informe hecho en LaTeX. Este debe incluir una introducción al problema, la solución a las consignas mostrando pseudocódigos con una explicación de su funcionamiento. Figuras o tablas para visualizar resultados junto con sus interpretaciones. Las figuras y tablas tienen que estar referenciadas en el texto y tener una descripción. Una sección de conclusiones con un resumen y de lo mostrado y los principales resultados.

2. Se debe entregar los códigos que generan los resultados, figuras, etc. Si se requieren procedimientos especiales para ejecutar el código deben ser descriptos.

3. La entrega es el Domingo 4 de Junio 23:59 h. Para la entrega se facilitará un formulario online donde deberán adjuntar el informe y los archivos.