PCA vs 2DPCA

Métodos Numéricos

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

May 19, 2023

PCA: ¿Cómo es la transformación característica?

Recordemos que tenemos $X \in R^{n \times m}$ con n igual a la cantidad de imágenes y m la cantidad de variables (alto $\times$ ancho).

- Si $B = (v_1, \ldots, v_n)$ es la base ortogonal de autovectores de $M_X = X^T X$ entonces la matriz cambio de base de B a la base canónica se escribe como la matriz cuyas columnas son los respectivos vectores:

$$C_{B,E} = V = \begin{bmatrix} & & \cdots & \\ v_1 & v_2 & \cdots & v_n \\ & & \cdots & \end{bmatrix}$$

PCA: ¿Cómo es la transformación característica?

Recordemos que tenemos $X \in R^{n \times m}$ con n igual a la cantidad de imágenes y m la cantidad de variables (alto $\times$ ancho).

- Si $B = (v_1, \ldots, v_n)$ es la base ortogonal de autovectores de $M_X = X^T X$ entonces la matriz cambio de base de B a la base canónica se escribe como la matriz cuyas columnas son los respectivos vectores:

$$C_{B,E} = V = \begin{bmatrix} & & \cdots & \\ v_1 & v_2 & \cdots & v_n \\ & & \cdots & \end{bmatrix}$$

- La matriz inversa de ésta da el cambio de base en la dirección opuesta. $C_{E,B} = C_{B,E}^{-1}$ y como nuestra base es ortogonal, tenemos

$$C_{E,B} = C_{B,E}^{T} = V^T = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

PCA: ¿Cómo es la transformación característica?

- Tenemos $X \in R^{n \times m}$

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x_n \end{bmatrix}$$

PCA: ¿Cómo es la transformación característica?

- Tenemos $X \in R^{n \times m}$

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x_n \end{bmatrix}$$

- Si x es una instancia de entrenamiento a la cual queremos cambiar de base, encontramos sus coordenadas usando $C_{B,E}^T = V^T$:

$$\overline{Z}_x = V^T x$$

PCA: ¿Cómo es la transformación característica?

- Tenemos $X \in R^{n \times m}$

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x_n \end{bmatrix}$$

- Si x es una instancia de entrenamiento a la cual queremos cambiar de base, encontramos sus coordenadas usando $C_{B,E}^T = V^T$:

$$\overline{Z}_x = V^T x$$

- Luego, si queremos cambiar de base cada instancia

$$V^T X^T = V^T \begin{bmatrix} x^{(1)} & \dots & x^{(n)} & \end{bmatrix} = \begin{bmatrix} V^T x^{(1)} & \dots & V^T x^{(n)} & \end{bmatrix}$$

PCA: ¿Cómo es la transformación característica?

- Tenemos $X \in R^{n \times m}$

$$X = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x_n \end{bmatrix}$$

- Si x es una instancia de entrenamiento a la cual queremos cambiar de base, encontramos sus coordenadas usando $C_{B,E}^T = V^T$:

$$\overline{Z}_x = V^T x$$

- Luego, si queremos cambiar de base cada instancia

$$V^T X^T = V^T \begin{bmatrix} x^{(1)} & \dots & x^{(n)} \end{bmatrix} = \begin{bmatrix} V^T x^{(1)} & \dots & V^T x^{(n)} \end{bmatrix}$$

- Para que cada imagen esté en una fila, trasponemos:

$$Z_X = (V^T X^T)^T = XV$$

PCA: Cómo recostruyo una imagen?

- Hay que reescibir cada imagen en el espacio de PCA.
- Dada una imagen x, $Z_x \in \mathbb{R}^m$ son las coordenadas en la base de autovectores V.

PCA: Cómo recostruyo una imagen?

- Hay que reescibir cada imagen en el espacio de PCA.
- Dada una imagen x, $Z_x \in \mathbb{R}^m$ son las coordenadas en la base de autovectores V.
- Entonces reconstruyo $x = \sum_{i=0}^{m} Z_x^i V_i$.

PCA: Cómo recostruyo una imagen?

- Hay que reescibir cada imagen en el espacio de PCA.
- Dada una imagen x, $Z_x \in \mathbb{R}^m$ son las coordenadas en la base de autovectores V.
- Entonces reconstruyo $x = \sum_{i=0}^{m} Z_x^i V_i$.
- Si tomamos $k \leq m$ tenemos $x \approx \sum_{i=0}^{k} Z_x^i V_i$.

2DPCA: Derivación

- Sea $A \in \mathbb{R}^{a \times a}$ imágen cuadrada, $X \in \mathbb{R}^{a}$

2DPCA: Derivación

- Sea $A \in \mathbb{R}^{a \times a}$ imágen cuadrada, $X \in \mathbb{R}^{a}$
- Buscamos proyección $Y = AX$ tal que aplicada a toda la base de imágenes

$$argmax\, J(X) = Tr(S_x) \tag{1}$$

donde $S_x = E(Y - \bar{Y})(Y - \bar{Y})^T$ es la image covariance matrix.

2DPCA: Derivación

- Sea $A \in \mathbb{R}^{a \times a}$ imágen cuadrada, $X \in \mathbb{R}^{a}$
- Buscamos proyección $Y = AX$ tal que aplicada a toda la base de imágenes

$$argmax\ J(X) = Tr(S_x) \tag{1}$$

  donde $S_x = E(Y - \bar{Y})(Y - \bar{Y})^T$ es la image covariance matrix.
- S_x contiene un subconjunto de las covarianzas de la de PCA (retiene las covarianzas de las componentes por columna). [1]

[1] Extended Two-Dimensional PCA for Efficient Face Representation and Recognition

2DPCA: Derivación

- Sea $A \in \mathbb{R}^{a \times a}$ imágen cuadrada, $X \in \mathbb{R}^{a}$
- Buscamos proyección $Y = AX$ tal que aplicada a toda la base de imágenes

$$argmax\ J(X) = Tr(S_x) \tag{1}$$

  donde $S_x = E(Y - \bar{Y})(Y - \bar{Y})^T$ es la image covariance matrix.
- S_x contiene un subconjunto de las covarianzas de la de PCA (retiene las covarianzas de las componentes por columna). [1]
- En otras palabras, queremos que la dispersión de los vectores proyectados sea máxima.

[1] Extended Two-Dimensional PCA for Efficient Face Representation and Recognition

2DPCA: Derivación

- Sea $A \in \mathbb{R}^{a \times a}$ imágen cuadrada, $X \in \mathbb{R}^{a}$
- Buscamos proyección $Y = AX$ tal que aplicada a toda la base de imágenes

$$argmax\ J(X) = Tr(S_x) \tag{1}$$

  donde $S_x = E(Y - \bar{Y})(Y - \bar{Y})^T$ es la image covariance matrix.
- S_x contiene un subconjunto de las covarianzas de la de PCA (retiene las covarianzas de las componentes por columna). [1]
- En otras palabras, queremos que la dispersión de los vectores proyectados sea máxima.
- Se puede ver que $Tr(S_X) = X^T GX$ con $G = \sum_{j=1}^{N}(A_j - \hat{A})(A_j - \hat{A})^T$
- Lo anterior se minimiza tomando X como el autovector dominante de G.

[1] Extended Two-Dimensional PCA for Efficient Face Representation and Recognition

2DPCA: Recostruyendo una imagen

- $A \in \mathbb{R}^{a \times a}$ imagen cuadrada
- $X_j \in \mathbb{R}^a$ $j = 1...a$ las direcciones principales
- $Y_j \in \mathbb{R}^a$ $j = 1...a$ los features vectores de A

Entonces...

- Definimos $V = (Y_1|Y_2|\ldots|Y_a)$ y $U = (X_1|X_2|\ldots|X_a)$

2DPCA: Recostruyendo una imagen

- $A \in \mathbb{R}^{a \times a}$ imagen cuadrada
- $X_j \in \mathbb{R}^a$ $j = 1...a$ las direcciones principales
- $Y_j \in \mathbb{R}^a$ $j = 1...a$ los features vectores de A

Entonces...

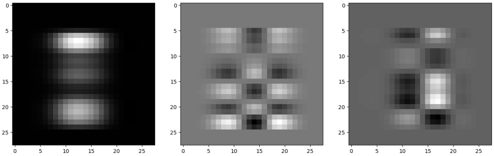
- Definimos $V = (Y_1|Y_2|\ldots|Y_a)$ y $U = (X_1|X_2|\ldots|X_a)$
- Notar que $V = AU$. Por ser U ortonormal $A = VU^T$ permite recuperar A.

2DPCA: Recostruyendo una imagen

- $A \in \mathbb{R}^{a \times a}$ imagen cuadrada
- $X_j \in \mathbb{R}^a$ $j = 1 ... a$ las direcciones principales
- $Y_j \in \mathbb{R}^a$ $j = 1 ... a$ los features vectores de A

Entonces...

- Definimos $V = (Y_1 | Y_2 | \dots | Y_a)$ y $U = (X_1 | X_2 | \dots | X_a)$
- Notar que $V = AU$. Por ser U ortonormal $A = VU^T$ permite recuperar A.
- $VU^T = \sum_{j=1}^{a} Y_j X_j^T$ por propiedad de multiplicación matricial.



2DPCA: Recostruyendo una imagen

- $A \in \mathbb{R}^{a \times a}$ imagen cuadrada
- $X_j \in \mathbb{R}^a$ $j = 1...a$ las direcciones principales
- $Y_j \in \mathbb{R}^a$ $j = 1...a$ los features vectores de A

Entonces...

- Definimos $V = (Y_1|Y_2|\ldots|Y_a)$ y $U = (X_1|X_2|\ldots|X_a)$
- Notar que $V = AU$. Por ser U ortonormal $A = VU^T$ permite recuperar A.
- $VU^T = \sum_{j=1}^{a} Y_j X_j^T$ por propiedad de multiplicación matricial.

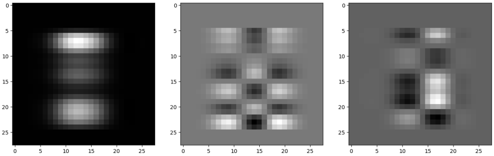


Figure: Ejemplo de las 3 *subimagenes* (i.e un término de la sumatoria) más relevantes.

- Si tomamos $k \leq a$ tenemos $A \approx \sum_{j=1}^{k} Y_j X_j^T$

2DPCA: Recostruyendo una imagen

- $A \in \mathbb{R}^{a \times a}$ imagen cuadrada
- $X_j \in \mathbb{R}^a$ $j = 1...a$ las direcciones principales
- $Y_j \in \mathbb{R}^a$ $j = 1...a$ los features vectores de A

Entonces...

- Definimos $V = (Y_1|Y_2|\dots|Y_a)$ y $U = (X_1|X_2|\dots|X_a)$
- Notar que $V = AU$. Por ser U ortonormal $A = VU^T$ permite recuperar A.
- $VU^T = \sum_{j=1}^{a} Y_j X_j^T$ por propiedad de multiplicación matricial.

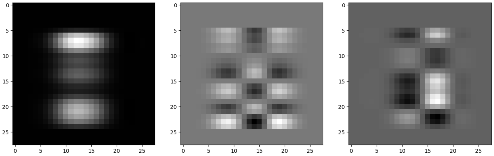


Figure: Ejemplo de las 3 *subimagenes* (i.e un término de la sumatoria) más relevantes.

- Si tomamos $k \leq a$ tenemos $A \approx \sum_{j=1}^{k} Y_j X_j^T$
- Equivalentemente, $A \approx VU_k^T$ con U_k las primeras k columnas de U.

2DPCA: Transformación característica

- 2DPCA proyecta la matriz que representa un espacio de menor dimensión.

2DPCA: Transformación característica

- 2DPCA proyecta la matriz que representa un espacio de menor dimensión.
- Como antes, por cada dirección principal se hace una proyección.

2DPCA: Transformación característica

- 2DPCA proyecta la matriz que representa un espacio de menor dimensión.
- Como antes, por cada dirección principal se hace una proyección.
- Ahora, cada proyección es un vector $Y \in \mathbb{R}^a$

Transformación característica

Dado $A \in \mathbb{R}^{a \times a}$ imagen cuadrada, V_k la matriz de k autovectores columna y $X_j \in \mathbb{R}^a$ una direcciones principal.

- PCA: $Z_A = \mathit{flatten}(A)V_k^T \in \mathbb{R}^k$
- 2DPCA: $Z_A = (AX_1| \dots |AX_k) = (Y_1| \dots |Y_k) \in \mathbb{R}^{a \times k}$