Dos aplicaciones de la SVD al procesamiento de imágenes.

Jeffry Chavarría Molina

23 de septiembre de 2019

TEC | Tecnológico de Costa Rica

Sección 1 | Preliminares

SVD

Definición de SVD

Sean $m, n \in \mathbb{N}$ y $A \in \mathrm{Mat}(\mathbb{C}, m, n)$. Una descomposición en valores singulares de A es una factorización

$$A_{m\times n} = U_{m\times m} \cdot \Sigma_{m\times n} \cdot V^*_{n\times n}$$

donde U y V son unitarias y Σ es una matriz diagonal generalizada de entradas reales tal que $\sigma_{ij} = 0$, para $i \neq j$. Si A solo tiene entradas reales entonces la SVD se escribe:

$$A = U\Sigma V^{\mathbf{T}}$$

Como U y V^* son de rango completo (por ser unitarias y por ende invertibles), entonces $\mathrm{r}(A) = \mathrm{r}(U\Sigma V^*) = \mathrm{r}(\Sigma)$. Por ende, el rango de A (denótelo r, esto es $\mathrm{r}(A) = r$) será el número de entradas no nulas de la matriz Σ. La matriz A no tiene porqué ser de rango completo.

Reducción de la SVD

SVD reducida

Sea $A \in \mathrm{Mat}(\mathbb{C}, m, n)$. La descomposición **reducida** en valores singulares de A está dado por:

$$A = U_r \Sigma_r V_r^*,$$

donde $r = \mathrm{r}(A)$, que corresponde al número de valores singulares no nulos de A, que a su vez son los valores propios no nulos de las matrices AA^* y A^*A.

Las matrices U_r, Σ_r y V_r^* quedan definidas como sigue:

$$U = [U_r \mid U_{m-r}]; \quad V = [V_r \mid V_{n-r}] \text{ y } \quad \Sigma = \left(\begin{array}{c|c} \Sigma_r & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right)$$

Aproximación de rango reducido

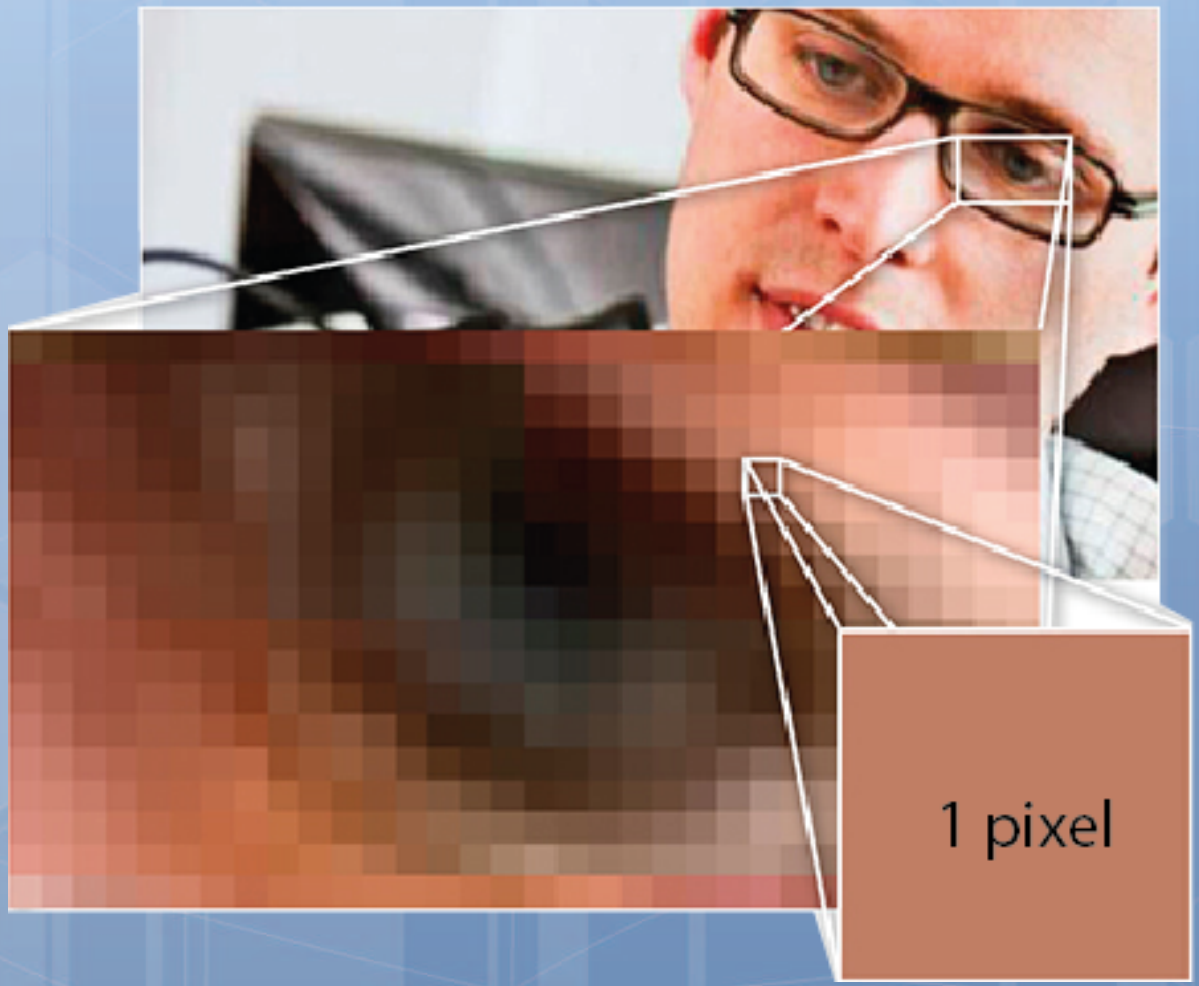
Sea $A \in \text{Mat}(\mathbb{C}, m, n)$ tal que $\text{r}(A) = r$ y sea r_0 un valor menor que r. Considere la forma reducida de la SVD para la matriz A dada por:

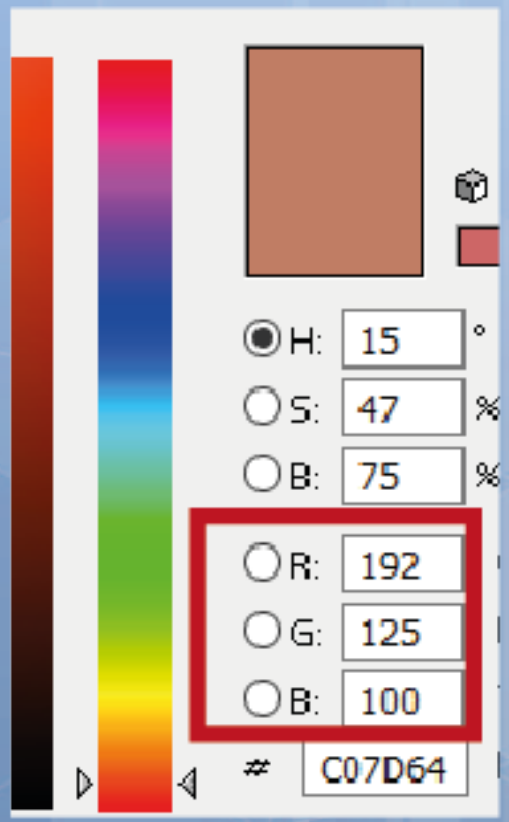
$$A = U_r \Sigma_r V_r^*$$

Y considere la matriz $A_0 = U_{r_0} \Sigma_{r_0} V_{r_0}^*$
Donde las matrices U_{r_0} y $V_{r_0}^*$ se definen de forma análoga a U_r y V_r y $\Sigma_r = \left(\begin{array}{c|c} \Sigma_{r_0} & \Sigma_{r_0 \times (r-r_0)} \\ \hline \Sigma_{(r-r_0)\times r_0} & \Sigma_{(r-r_0)\times(r-r_0)} \end{array} \right)$
A_0 así definida corresponde a la solución de aproximar la matriz de rango bajo de la matriz A, es decir, la solución al problema de minimización:

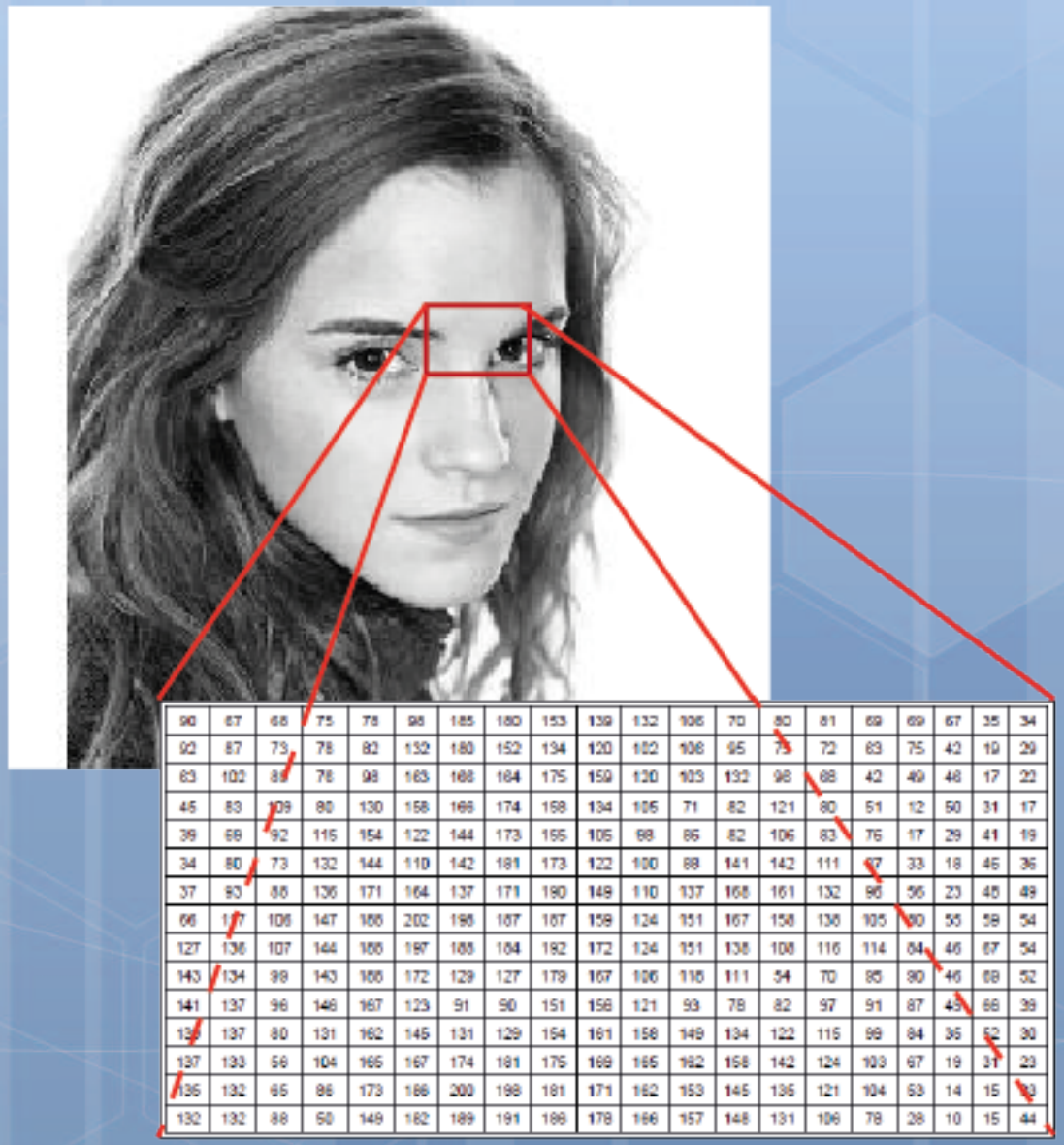
$$\min_{\text{r}(X) \leq r_0} ||A - X||_{fro}^2$$

Representación matricial de una imagen





Representación matricial de una imagen



TEC | Tecnológico de Costa Rica

Sección 2 | SVD en reconocimiento facial

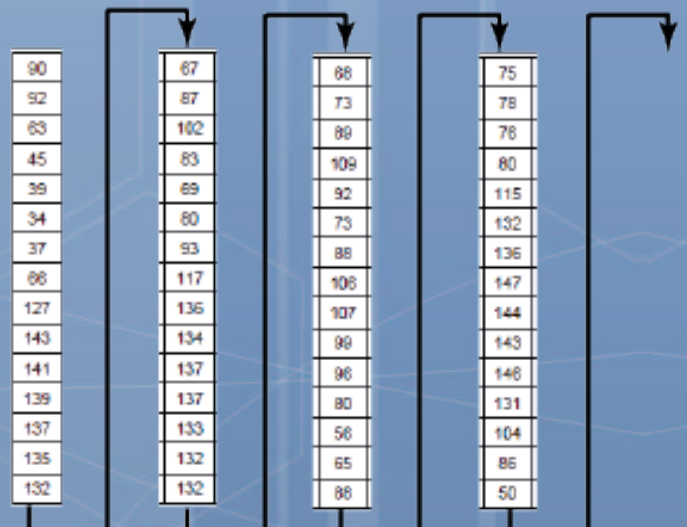
Primera aplicación: reconocimiento facial

Esta aplicación está basada en el artículo:
Zeng G. (2007). Facial Recognition with Singular Value Decomposition. In: Elleithy K. (eds) Advances and Innovations in Systems, Computing Sciences and Software Engineering. Springer, Dordrecht.

Planteamiento

Suponga que se tiene N imágenes de rostros de tamaño $m \times n$. Cada imagen $\mathbf{f}_i$ se puede reescribir como una matriz columna de tamaño $mn \times 1$.

90	67	68	75	78	98	165	180	153	139	132	108	70	80	81	69	69	67	35	34
92	87	73	78	82	132	180	152	134	120	102	106	95	75	72	63	75	42	19	29
63	102	89	76	98	163	166	164	175	159	120	103	132	96	68	42	49	46	17	22
45	83	109	80	130	150	166	174	156	134	105	71	82	121	80	51	12	50	31	17
39	69	92	115	154	122	144	173	155	105	98	86	82	106	83	76	17	29	41	19
34	80	73	132	144	110	142	161	173	122	100	66	141	142	111	67	33	16	46	36
37	93	88	136	171	164	137	171	190	140	110	137	168	161	132	96	56	23	48	49
66	117	106	147	166	202	186	167	167	159	124	151	167	156	136	105	60	55	59	54
127	136	107	144	188	197	188	184	192	172	124	151	138	108	116	114	84	46	67	54
143	134	99	143	166	172	129	127	179	167	106	116	111	54	70	95	90	46	69	52
141	137	96	146	167	123	91	90	151	156	121	93	78	82	97	91	87	45	66	39
139	137	80	131	162	145	131	129	154	161	158	149	134	122	115	99	84	35	52	30
137	133	56	104	165	167	174	181	175	169	165	162	158	142	124	103	67	19	31	23
135	132	65	86	173	188	200	198	181	171	162	153	145	135	121	104	53	14	15	33
132	132	88	50	149	182	189	191	186	178	166	157	148	131	106	78	28	10	15	44



TEC | Tecnológico de Costa Rica

Planteamiento

De esta manera la matriz:

$$S = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N]$$

corresponde a una matriz de tamaño $mn \times N$. Así, la $i-$ésima columna de S corresponde a la $i-$ésima cara escrita como un vector columna.
Considere el vector columna $\bar{\mathbf{f}}$, dado por:

$$\bar{\mathbf{f}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{f}_i$$

Este vector corresponde a la imagen promedio de S.

Considere $\mathbf{a}_i = \mathbf{f}_i - \bar{\mathbf{f}}$ para todo $i = 1, 2, \ldots, N$. De donde es posible definir la matriz A de tamaño $mn \times N$ de la siguiente forma:

$$A = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \ldots, \mathbf{a}_N]$$

Suponga que $\mathrm{r}(A) = r$, donde $r \leq N << mn$. Al calcular la SVD de A se tiene $A = U\Sigma V^*$, donde U y V son unitarias y:

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_r & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & \sigma_{r+1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & \sigma_N \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Las matrices $U_{mn\times mn}$ y $V_{N\times N}$ son matrices ortogonales de la forma:

$$U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r, \mathbf{u}_{r+1}, \ldots, \mathbf{u}_{mn}]$$

$$V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r, \mathbf{v}_{r+1}, \ldots, \mathbf{v}_N]$$

Donde $\mathbf{u}_i$ y $\mathbf{v}_i$ corresponden a los vectores singulares asociados al valor singular σ_i. Además, como U y V son

$$\mathbf{u}_i^{\mathbf{T}} \cdot \mathbf{u}_j = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases} \qquad \mathbf{v}_i^{\mathbf{T}} \cdot \mathbf{v}_j = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$$

Por otro lado, $A = U\Sigma V^{\mathbf{T}} \Rightarrow AV = U\Sigma$, de donde se deduce que:

$$A\mathbf{v}_i = \sigma_i \mathbf{u}_i$$

donde $\sigma_i \mathbf{u}_i = 0$ para todo $i > r$.

Se tiene que $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r\}$ forma una base ortonormal del espacio generado por las columnas de A.

En este caso, a cada vector $\mathbf{u}_i$ con $i \leq r$ se denomina cara-base del subespacio de caras.

Note lo siguiente: cada uno de los elementos $\mathbf{a}_i = \mathbf{f}_i - \bar{\mathbf{f}}$ se puede ver como un elemento de $\mathbb{R}^{mn}$ el cual es un espacio vectorial. Y el conjunto $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r\}$, es un conjunto de vectores ortogonales de $\mathbb{R}^{mn}$, es decir, el conjunto generado por esos vectores resulta ser un subespacio vectorial de $\mathbb{R}^{mn}$, el cual es el que llamaremos subespacio de caras, y se construye a partir del conjunto de imágenes de entrenamiento.

El subespacio de caras

En este punto, se tiene que $\text{gen}(\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r\})$ es un subespacio vectorial del espacio de caras. Dada una cara nueva $\mathbf{f}$, que se representa como un vector de tamaño $mn \times 1$ es posible proyectar sobre el subespacio generado por U_r el vector $\mathbf{f} - \bar{\mathbf{f}}$.
Suponga que $\mathbf{x} = [x_1, x_2, \ldots, x_r]^\mathbf{T}$ denota las coordenadas del vector proyección respecto a la base, dichas coordenadas pueden ser calculadas por:

$$\mathbf{x} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r]^\mathbf{T}(\mathbf{f} - \bar{\mathbf{f}})$$

Luego se compara dichas coordenadas con las coordenadas de los vectores de la base, es decir, se compara con las coordenadas $\mathbf{x}_i$, donde:

$$\mathbf{x}_i = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r]^\mathbf{T}(\mathbf{f}_i - \bar{\mathbf{f}})$$

Validación de una cara nueva

La distancia entre los vectores de coordenadas puede ser considerada una medida de cercanía entre la cara nueva y las caras del conjunto de entrenamiento.

$$\varepsilon_i = \|\mathbf{x}_i - \mathbf{x}\|_2$$

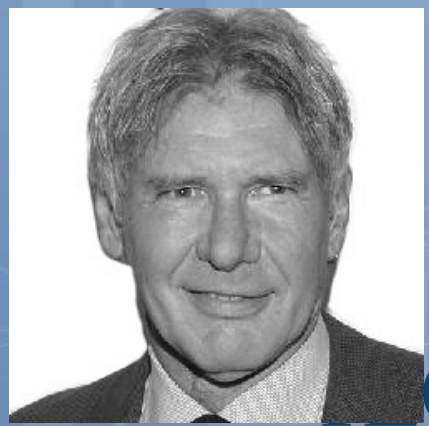
entonces la cara $\mathbf{f}$ será clasificada como el sujeto en la imagen $\mathbf{f}_i$ si ε_i cumple:

$$\varepsilon_i = \text{mín}\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N\}$$

y $\varepsilon_i \leq \varepsilon_0$, donde ε_0 es un umbral predefinido. En caso contrario $\mathbf{f}$ será clasificada como cara desconocida.

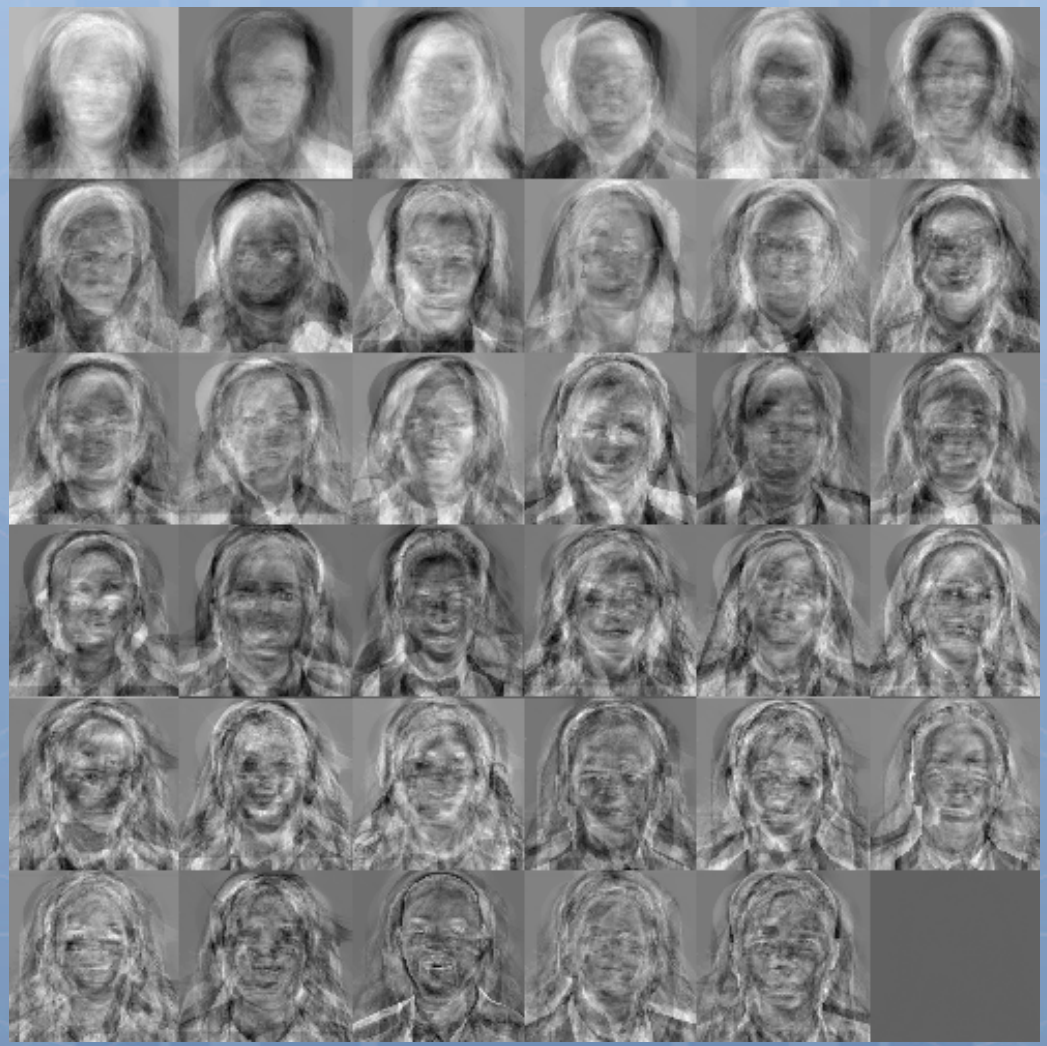
Ejemplo:

Para esta aplicación se ideó una pequeña base de datos con 36 rostros:



Ejemplo

Luego de ejecutar el algoritmo para las 36 caras se obtiene el subespacio de caras:



Ejemplo

Ver programa

Sección 3 | Segunda aplicación: Algoritmo GoDec

Segunda aplicación: Algoritmo GoDec

Esta aplicación de basa en los resultados del artículo:

Zhou, Tianyi & Tao, Dacheng. (2011). Bilateral Random Projections. 10.1109/ISIT.2012.6283064.

Segunda aplicación: Algoritmo GoDec

Resultado de GoDec

Este algoritmo realiza la siguiente tarea: Dado un vídeo X con K frame, en donde se tiene un conjunto de objetos en movimiento sobre un fondo fijo, se pretende generar dos vídeos S y L de manera que S contenga a los objetos en movimiento sin fondo y L el fondo fijo pero sin los objetos en movimiento.

Segunda aplicación: Algoritmo GoDec

Formulación matemática

Considere una matriz X de tamaño $M \times N$ y sean r_0 y k_0 valores dados, tal que $r_0 << \mathrm{r}(X)$ y $k_0 << \mathrm{Card}(X)$, el problema queda resuelto al encontrar la solución del siguiente problema de optimización:

$$\min_{\mathrm{r}(L) \leq r_0;\ \mathrm{Card}(S) \leq k_0} \|X - L - S\|_{fr}^2$$

Segunda aplicación: Algoritmo GoDec

Análisis

El problema anterior pretende determinar L y S tal que $X \approx L + S$, donde $\mathrm{r}(L) \leq r_0$ y $\mathrm{Card}(S) \leq k_0$. Por lo que resolver el problema anterior sería equivalente a resolver simultáneamente los siguientes dos problemas:

- Determinar la matriz de bajo rango L de la matriz $X - S$.
- Determinar una matriz rala S de la matriz $X - L$.

El principal problema es la simultaneidad de los cálculos.

Primer problema

Matriz de bajo rango

Este problema está resuelto gracias a la SVD. Recuerde que:

$$(X - S)_{M\times N} = U_{M\times M}\Sigma_{M\times N}V^*_{N\times N} = U_r\Sigma_r V^*_r$$

donde $r = \mathrm{r}(X - S)$.
Por lo que determinar la matriz de rango bajo L de $X - S$, se puede hacer como sigue:

$$L = U_{r_0}\Sigma_{r_0}V^*_{r_0}$$

donde $r_0 \leq r$.

Segundo problema

Matriz rala o esparcida

Considere un operador de esparcidad de una matriz, denominado $\mathcal{P}$. Para el cual se tiene lo siguiente: Dada una matriz X de tamaño $M \times N$ y una constante k_0 tal que $k_0 < MN$, entonces:

$$\mathcal{P}(X, k_0) = X_{k_0}$$

donde X_{k_0} se obtiene de X dejando intactas las k_0 entradas más alejadas de cero y haciendo cero las restantes.
Por lo que:

$$S = \mathcal{P}(X - L, k_0)$$

Algoritmo GoDec

Algorithm 1 GoDec Clásico

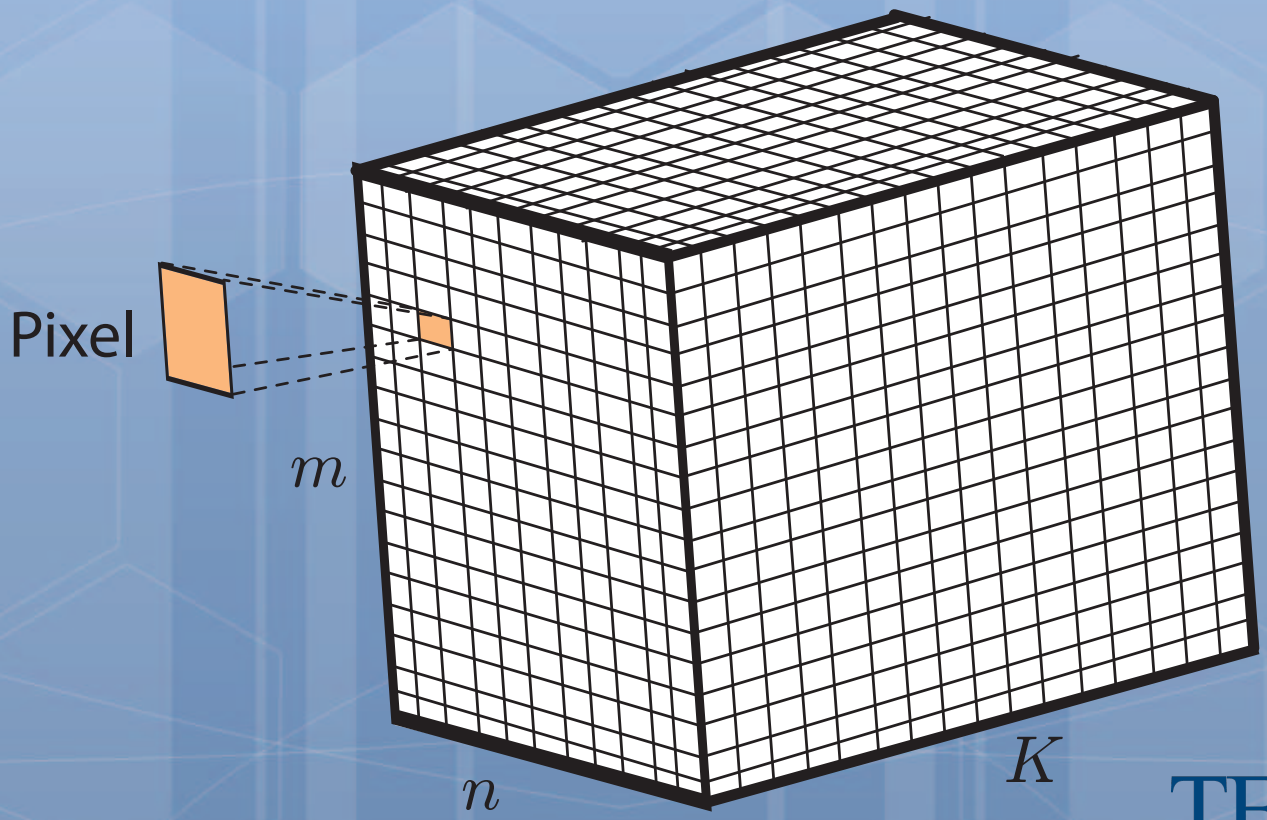
Input: $k_0, r_0, X_{M\times N}, \varepsilon$

Output: L, S

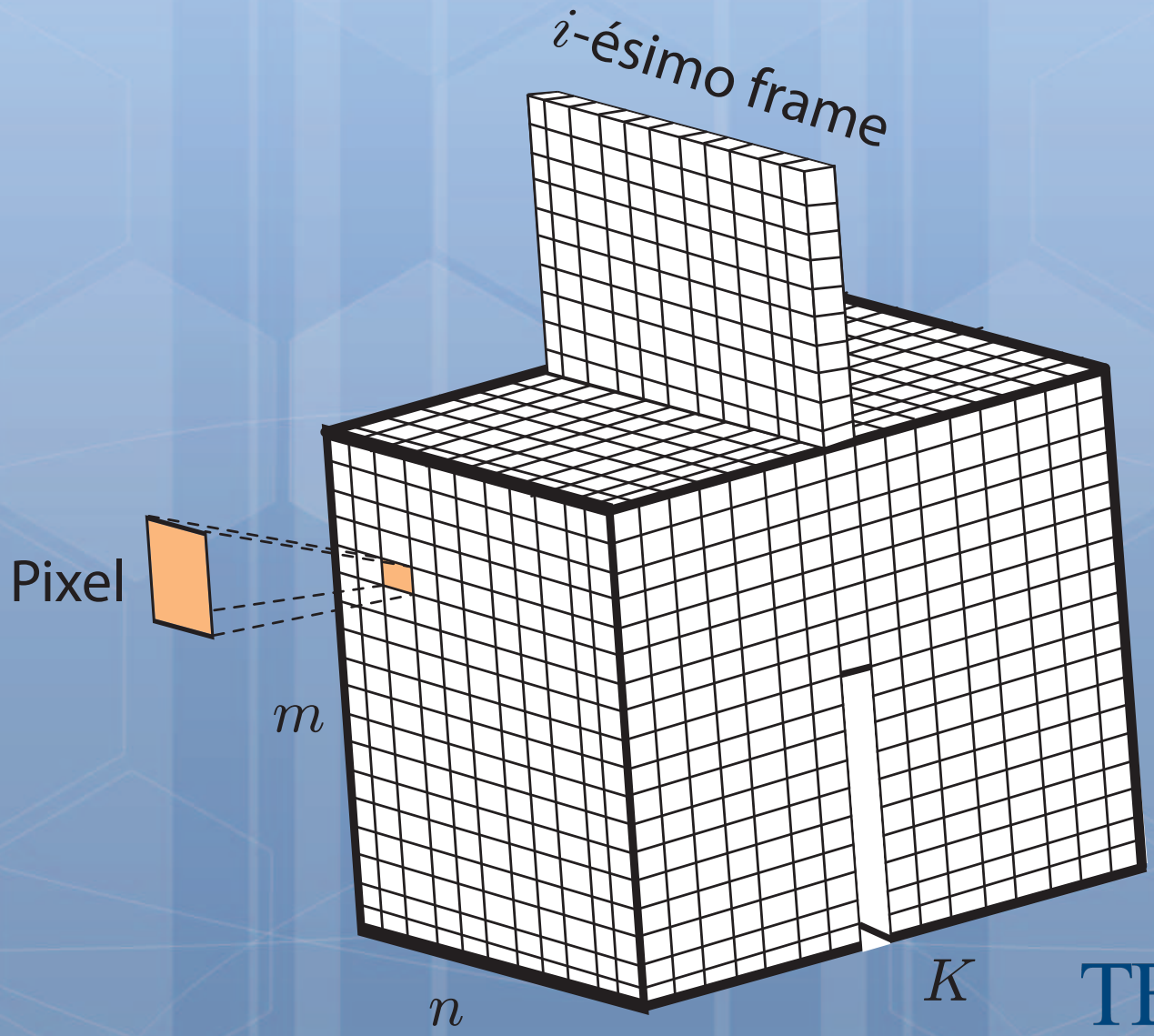
$L_0 := X, S_0 := \mathbf{0}_{M\times N}, t := 0$
while Verdadero **do**
  $t := t + 1$
  $[U, \Sigma, V^*] = \text{svd}(X - S_{t-1})$
  $L_t := U_{r_0} \cdot \Sigma_{r_0} \cdot V^*_{r_0}$
  $S_t = \mathcal{P}(X - L_t, k_0)$
  $E_t := \|X - L_t - S_t\|^2_{fr} / \|X\|^2_{fr}$
  if $|E_t - E_{t-1}| < \varepsilon$ **then**
    break
  end if
end while
retornar L, S

Tratamiento de los datos:

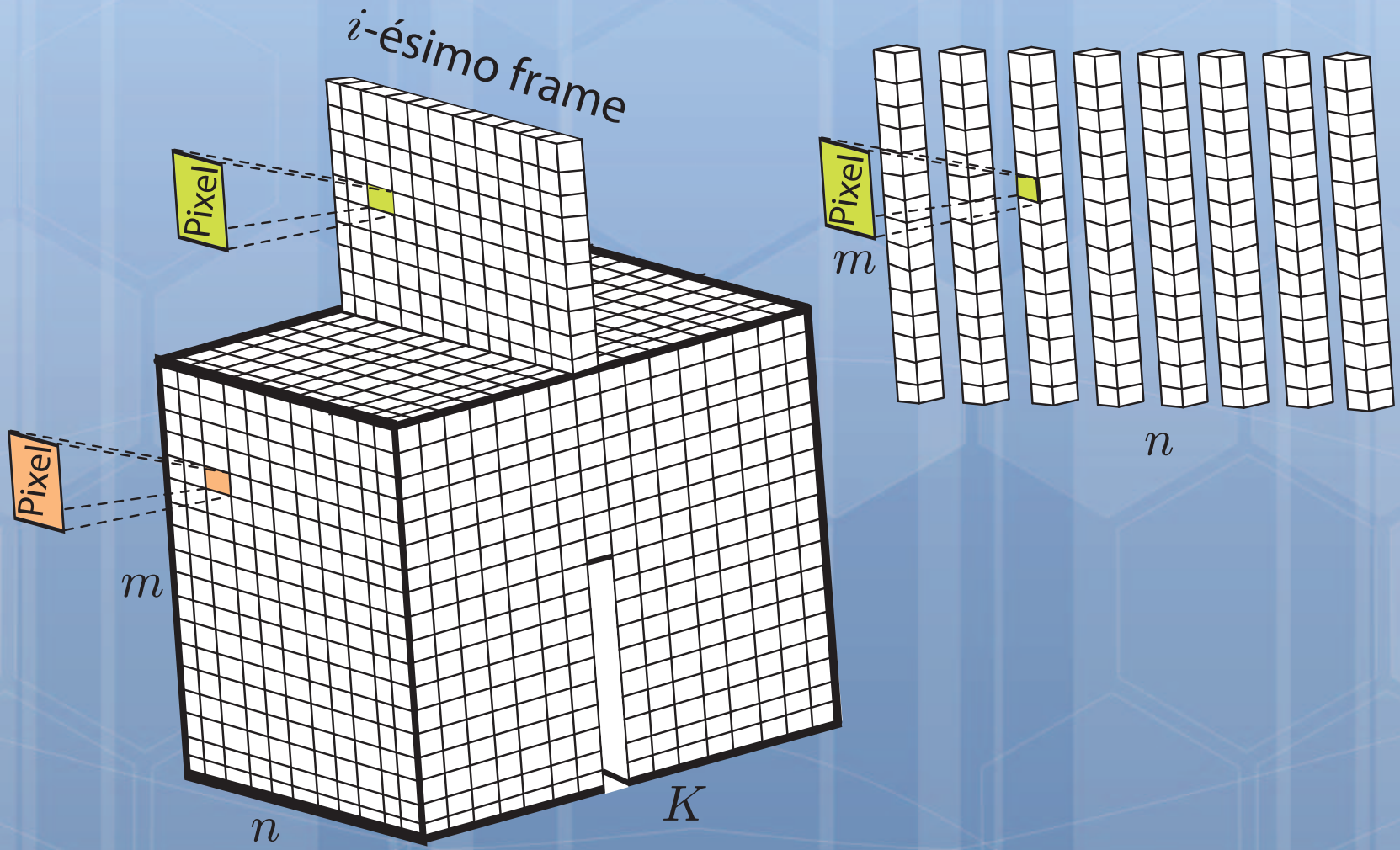
Considere un vídeo V con K frames, cada uno de tamaño $m \times n$ en el cual se muestra un conjunto de objetos en movimiento sobre un fondo fijo o inmutable. Considere la matriz X de tamaño $mn \times K$ construida como sigue:



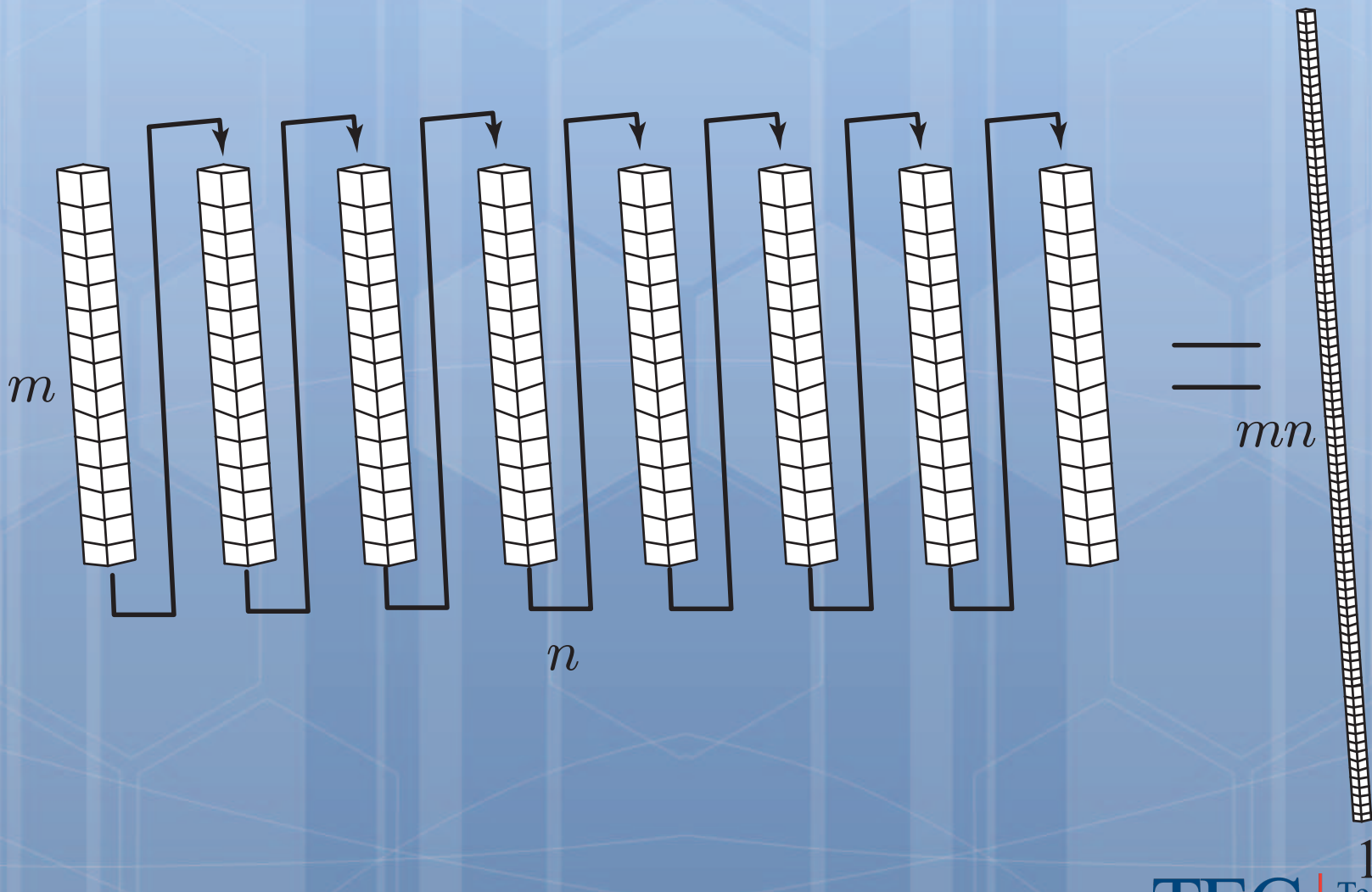
Tratamiento de los datos:



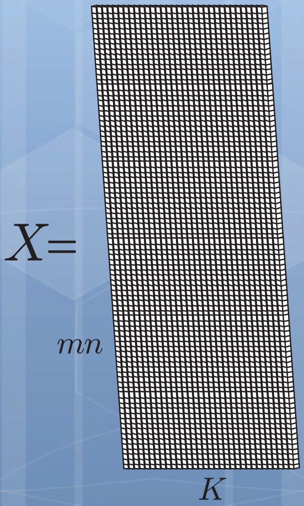
Tratamiento de los datos:



Tratamiento de los datos:



Tratamiento de los datos:



Tratamiento de los datos:

Resultados

Sección 4 | Mejora de GoDec

Principal desventaja de la SVD

El cálculo de las SVD no es computacionalmente eficiente. Si bien existen algoritmos iterativos que calculan o aproximan la SVD, el cálculo de dicha factorización es computacionalmente cara si se hace por el método tradicional, con la diagonalización de matrices.

Proyecciones bilaterales aleatorias

Suponga que se tiene una matriz densa L_0 de tamaño $m \times n$, considere además las matrices, denominadas proyecciones bilaterales, dadas por: $Y_1 = L_0 A_1$ y $Y_2 = L_0^T A_2$, donde $A_1 \in \mathbb{R}^{n \times r}$ y $A_2 \in \mathbb{R}^{m \times r}$, entonces la matriz L_0 puede ser aproximada por la matriz de rango a lo sumo r:

$$L = Y_1 \left(A_2^T Y_1 \right)^{-1} Y_2^T \tag{1}$$

Demostración:

Si A_2^T tiene entradas aleatorias. Y si $\mathrm{r}(L_0) = r$, entonces el algoritmo en (1) cumple que $L = L_0$ con probabilidad 1.

$$
\begin{aligned}
\Rightarrow \quad & L = Y_1(A_2^T Y_1)^{-1} Y_2^T \\
\Rightarrow \quad & L = (L_0 A_1)(A_2^T L_0 A_1)^{-1}(L_0^T A_2)^T \\
\Rightarrow \quad & L = L_0 A_1 (A_2^T L_0 A_1)^{-1} A_2^T L_0 \\
\Rightarrow \quad & A_2^T L = A_2^T L_0 A_1 (A_2^T L_0 A_1)^{-1} A_2^T L_0 \\
\Rightarrow \quad & A_2^T L = (A_2^T L_0 A_1)(A_2^T L_0 A_1)^{-1} A_2^T L_0 \\
\Rightarrow \quad & A_2^T L = A_2^T L_0 \\
\Rightarrow \quad & A_2^T (L - L_0) = 0
\end{aligned}
$$

En este punto se tiene dos posibilidades, o bien $L = L_0$ o A_2^T cae en el espacio nulo de $(L - L_0)^T$, pues $(L - L_0)^T A_2 = 0$, implicando que A_2^T pertenece a un conjunto de medida cero, lo cual tiene probabilidad cero de ocurrir. Finalmente se concluye que $L = L_0$ con probabilidad 1.

Por ahora se tiene que L_0 puede ser aproximada por:

$$L = Y_1 \left(A_2^T Y_1\right)^{-1} Y_2^T$$

donde $Y_1 = L_0 A_1$ y $Y_2 = L_0^T A_2$, y las matrices $A_1 \in \mathbb{R}^{n \times r}$ y $A_2 \in \mathbb{R}^{m \times r}$ pueden ser escogidas aleatoriamente.
Los autores del artículo propone escoger dichas matrices con el siguiente algoritmo para simplificar cálculos:

$$Y_1 = L_0(L_0^T L_0)A_1 \tag{2}$$
$$Y_2 = (L_0^T L_0)A_1 \tag{3}$$
$$A_2 = L_0 A_1 \tag{4}$$

De donde se tiene que: L se puede simplificar a:

$$L = (L_0 Q)Q^T$$

donde Q es $QR = qr(Y_2)$.

Algorithm 2 GoDecProyecto

Input: k, r, q, $X_{m\times n}$
Output: L, S

$L_0 := X$, $S_0 := \mathbf{0}_{m\times n}$, $t := 0$
while Verdadero **do**
  $t := t + 1$
  $L = X - S$
  $Y_2 = randn(n, r)$
  for i=1:q+1 **do**
    $Y_1 = L * Y_2$
    $Y_2 = L^T * Y_1$
  end for
  $QR = qr(Y_2)$
  $L_t := (L * Q) * Q^T$
  $S_t := \mathcal{P}_\Omega (X - L_t)$
  $E_t := \dfrac{\|X - L_t - S_t\|_{fr}^2}{\|X\|_{fr}^2}$
  if $|E_t - E_{t-1}| < tol$ **then**
    break
  end if
end while
retornar L, S