SVD y compresión de imágenes

Salvador Pintos

abril/2003

Resumen

Singular Value Decomposition es la más importante factorización de una matriz. Es numéricamente estable y sus aplicaciones abarcan un extenso número de áreas, desde solución de sistemas lineales, Análisis de Componentes Principales (PCA), hasta Big Data. Aquí se resumen sólo las propiedades esenciales de SVD necesarias para su aplicación a la compresión de imágenes.

Revisión de propiedades de SVD

Definición

Singular Value Decomposition factoriza una matriz $A_{m\times n}$ en el producto de tres matrices, $A = UDV^T$, donde:

- $U_{m\times m}$ y $V_{n\times n}$ son ortogonales y $D_{m\times n}$ es diagonal y tiene la estructura siguiente:

$$\begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_k & 0 & 0 \\ 0 & 0 & \sigma_r & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- Los r *valores singulares*, σ_k, no nulos de la diagonal D cumplen:

$$\sigma_1 \geq \sigma_2 .... \geq \sigma_r > 0$$

Notación restringida

En lo que sigue se particiona U y V en dos matrices formadas con sus primeras r columnas y con las restantes:

$$U = [U_r \uparrow U_{m-r}] \quad V = [V_r \uparrow V_{n-r}]$$

Además, se define D_r la matriz $r \times r$ cuadrada formada con las primeras r filas y columnas de D. Se cumple que:

$$A = UDV^T = U_r D_r V_r^T$$

Descomposición en suma de matrices de rango uno

Este resultado puede expresarse como suma de matrices:

$$A = \sum_{k=1}^{r} \sigma_k u_k v_k^T$$

donde cada uno de los r productos, $u_k v_k^T$ es una matriz $m \times n$ de rango uno, ya que es el producto de un vector columna, u_k por un vector fila v_k^T. Para almacenar esas r matrices se necesita guardar , $r(m+n+1)$ números.

Norma Frobenius de una matriz $\|A\|_F$

Es la raíz cuadrada de la suma de los cuadrados de todos los elementos de la matriz. Donde:

$$\|A\|_F^2 = \sum_{i=1}^{m} \quad \sum_{j=1}^{n} \left|a_{i,j}\right|^2$$

Se relaciona con la factorización SVD ya que es igual a la suma de los valores singulares

$$\|A\|_F^2 = \sum_{h=1}^{r} \sigma_h^2$$

La mejor aproximación de rango s a una matriz A

Entre todas las matrices $B_{m\times n}$ de rango $s < r$, la mejor aproximación a A en el sentido de minimizar $\|A-B\|_F^2$ es:

$$B_s = \sum_{k=1}^{s} \sigma_k u_k v_k^T \tag{1}$$

recuérdese que es la suma de s matrices de rango 1. Y la aproximación de B a A medida por la norma frobenius es:

$$\|A-B\|_F^2 = \sum_{h=s+1}^{r} \sigma_h^2$$

Como $\sigma_1 \geq \sigma_2 .... \geq \sigma_r \geq 0$ se suman los σ más pequeños.

Compresión de imágenes

Una imagen en escala de grises es una matriz $A_{m\times n}$ y para guardarla se necesitan $m \times n$ pixels.

La figura 1 contiene una imagen cuyo tamaño es de 800×738 pixels. Es en escala de grises y se lee en octave, en una matriz $A_{800\times 738}$.

La mejor aproximación de rango s , ecuación 1, ocupa $s \times (m+n+1)$ pixels, luego la compresión de memoria es:

$$compresión = \frac{s \times (m+n+1)}{m \times n}$$

Figura 1: Foto de familia

En cuanto a la pérdida de calidad de la imagen el error relativo está dado por:

$$error = \frac{\|A - B\|_F}{\|A\|_F} = sqrt\left(\frac{\sum_{h=s+1}^{r} \sigma_h^2}{\sum_{h=1}^{r} \sigma_h^2}\right)$$

El proceso de selección de s debe iniciarse por graficar este error y determinar s de modo que la pérdida de calidad sea mínima, menor *error*, pero obteniendo una buena *compresión*. Las curvas presentadas en la figura 2 muestran que si se incluyen más valores singulares el error de la imagen aproximada decrece, pero es opuesto el comportamiento de la compresión lograda. En dicha figura se observa que es necesario acumular aproximadamente 57 componenente para que el error entre B_{57} y A sea menor a un 10%. Con 100 componentes, la aproximación B_{100} tiene un error relativo del 5%. Siempre se busca elegir el número de valores singulares compatibilizando ambos objetivos: que la imagen se aproxime a la original pero logrando al mismo tiempo una compresión adecuada.

La figura 3 muestra una imagen original y las aproximaciones B_{20}, B_{60}, y B_{100}. En el rincón arriba a la derecha se observa que con la aproximación B_{20} si bien sólo se necesita el 5% de los pixels, la imagen es borrosa. Abajo a la izquierda, B_{60} es una aproximación razonable y 16% de compresión. La B_{100} ocupa sólo la cuarta parte de los pixeles y prácticamente no se distingue de la original.

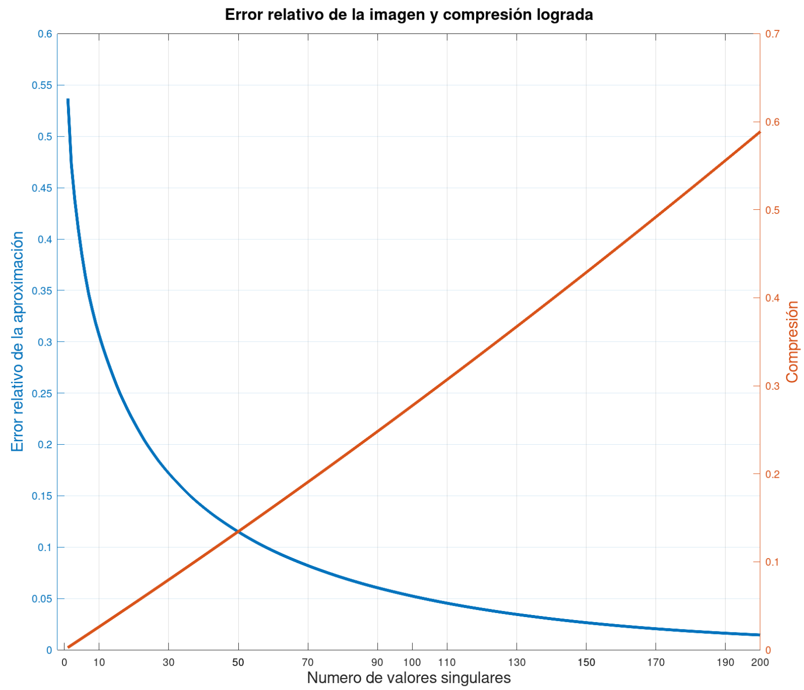

Figura 2: Error de la imagen y compresión lograda

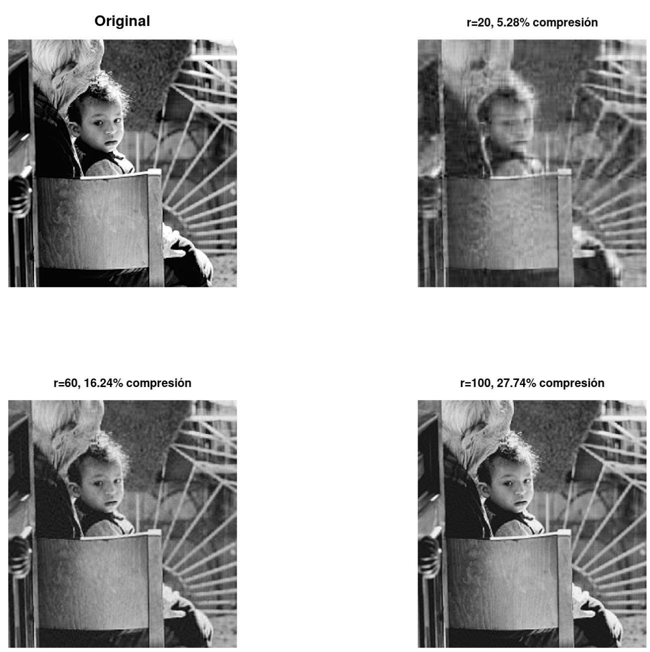

Figura 3: Imágenes con distinta resolución