

# SVD y compresión de imágenes

Facultad de Matemática y Computación de la Universidad de La Habana

Pablo Gómez Vidal C211

La compresión de una imagen es minimizar el tamaño en bytes de un archivo gráfico sin degradar la calidad de la imagen a un nivel inaceptable. Existen diversas vías para realizar este proceso y una de esas vías es utilizando SVD. La descomposición de valor único se puede utilizar para comprimir imágenes reduciendo los rangos de la matriz de imágenes. Se puede preservar la mayor parte de la calidad de la imagen incluso después de comprimirla en una cantidad minúscula.

Observamos esto tanto en imágenes en escala de grises como en imágenes en color. La reducción en el tamaño del archivo permite almacenar más imágenes en una cantidad determinada de espacio en disco o memoria. También reduce el tiempo necesario para que las imágenes se envíen a través de Internet o se descarguen de la página web. Desde los albores de la generación de datos gráficos de Internet ha aumentado exponencialmente. La aparición de las redes sociales también ha desempeñado un papel muy importante en esto. Por lo tanto, se ha vuelto importante que optimicemos nuestros datos en términos de tamaño y comprimamos los archivos tanto como sea posible sin perder mucho valor.

Una imagen a color es una matriz de  $(n, m, 3)$  números donde a cada píxel se le asigna un vector en  $R^3$ , el vector representa la composición *RGB* del color. Este enfoque permite aplicar SVD al tratamiento de imágenes ya que toda imagen contiene información redundante, o sea, que puede ser eliminada sin que el efecto visual sea notable, se podría sustituir la matriz  $A$  por una matriz  $B$  de rango prefijado más pequeño.

Por ejemplo en una imagen de  $(600 \times 800)$  píxeles el rango será 600.

En vez de sumar todos los valores singulares, podemos reconstruirla hasta un número de valores singulares entre 1 y el rango de la matriz ( $1 < k < r$ ). Mientras mayor sea  $k$  mejor será la calidad de imagen pero menor la compresión, y viceversa.

En este sencillo proyecto utilicé las librerías de `Numpy` y `PIL` y la idea es bastante sencilla lo primero fue crear una función que abriera la imagen y la convirtiera en una matriz y la separara en tres matrices más para cada color RGB. Luego se creó la función que se encarga de comprimir la información en dependencia de la cantidad de valores singulares escogidos donde utilicé el método `numpy.linalg.svd` y se devuelve la matriz comprimida. Luego ya en la función principal se trata la imagen en cuestión (hay una foto de test adjunta junto con el proyecto) luego ya se divide la foto en 4 matrices, el original y uno para cada color RGB, luego se comprimen los canales R(rojo), G(verde), B(azul), y por último se combinan las tres nuevas imágenes con la original con la función `Image.merge` y el trabajo está terminado.