

## Exercício Programa 23

### MAC300 - Métodos Numéricos da Álgebra Linear

Fernando Omar Aluani 6797226

## 1 Decomposição por Valores Singulares

A Decomposição por Valores Singulares (SVD) é um método para fatorar uma matriz  $A$  ( $m \times n, m \geq n$ ) na forma  $A = U\Sigma V^T$ , onde:

- $U$  é  $m \times m$ , unitária. Colunas são os vetores singulares à esquerda;
- $\Sigma$  é  $m \times n$ , diagonal. Cada elemento da diagonal é um valor singular de  $A$ ;
- $V$  é  $n \times n$ , unitária. Colunas são os vetores singulares à direita;

Os vetores/valores singulares são relacionados aos autovetores/autovalores de uma matriz, o que faz esse método ser muito útil para achar os autovetores e seus respectivos autovalores de uma matriz  $A$ .

No EP, a decomposição SVD é executada em 2 passos. O propósito do primeiro passo é "simplificar" a matriz para a posterior diagonalização, e isso é feito com o algoritmo *Golub-Kahan*. Este passo transforma a matriz original  $A$  em  $A = UBV^T$ , onde

- $U$  é  $m \times m$ , unitária;
- $B$  é  $m \times n$ , bi-diagonal;
- $V$  é  $n \times n$ , unitária;

O segundo passo recebe  $U$ ,  $B$  e  $V$ , e aplicando o algoritmo *Golub-Reinsch* ele diagonaliza a matriz  $B$ , transformando-a na matriz  $\Sigma$  e atualizando  $U$  e  $V$  de tal forma que  $A = U\Sigma V^T$ . Terminando a decomposição SVD.

### 1.1 Melhorias ao método clássico

As melhorias desse método de decomposição Golub-Kahan-Reinsch do SVD em relação ao método clássico, que usa os autovetores e autovalores de  $AA^T$  e  $A^TA$ , são:

- É melhor para matrizes mal-condicionadas e de posto incompleto no geral;
- É sempre numericamente estável;
- É mais eficiente que a decomposição por autovalores de  $AA^T$  ou  $A^TA$ ;

### 1.2 Análise da Complexidade

Golub-Kahan é basicamente a aplicação de 2 householders ao mesmo tempo.

Golub-Reinsch vai, seguindo seu algoritmo, aplicando sucessivas rotações de Givens na matriz bidiagonal  $B$ , até ela "convergir" e ser diagonalizada.

## 2 Exemplo da Compressão de Imagens

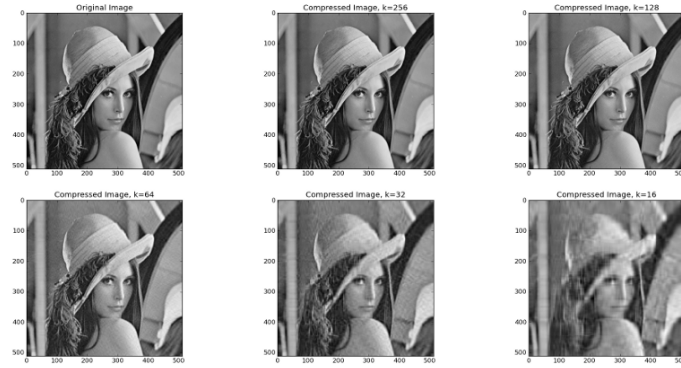


Figura 1: Comparação de uma imagem original em grayscale e diversas compressões.

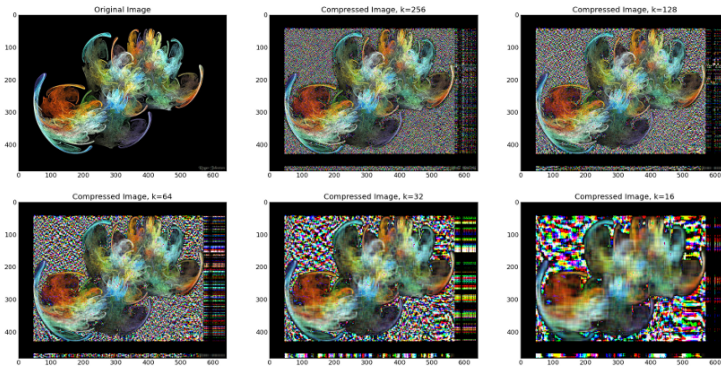


Figura 2: Comparação de uma imagem original em RGB e diversas compressões. Observação: rodando o EP em si para essa imagem o resultado da compressão não é o mostrado aqui. Por alguma razão que desconheço, o matplotlib desenha errado as imagens coloridas (nesse caso, o 'fundo' da imagem, que era preto, não passa a ser colorido de forma ruidosa assim)