



# Subspace Iteration Methods Application to Image Compression

## 1 Reminders and introduction

In the first session of this project, we have seen that the power method algorithm with deflation is not efficient in terms of performance and we have presented a more efficient method called **subspace iteration method**, based on an object called *Rayleigh quotient*.

With these latter method, we propose to apply it for image compression. As the images are described as rectangular matrices, this application relies on the Singular Value Decomposition and the best low-rank approximation theorem.



Examples in grayscale of comics images to compress

### Theorem: Singular Value Decomposition (SVD)

Let  $A \in \mathbb{R}^{q \times p}$  ( $q \geq p$ ) with full rank,  $\text{rank}(A) = p$  then we can decompose  $A$  as  $A = U\Sigma V^T$  with

- $U \in \mathbb{R}^{q \times q}$  is formed of  $q$  orthonormal eigenvectors associated to  $q$  eigenvalues of  $AA^T$ .
- $V \in \mathbb{R}^{p \times p}$  is formed of  $p$  orthonormal eigenvectors associated to the  $p$  eigenvalues of  $A^T A$ .
- $\Sigma \in \mathbb{R}^{q \times p}$  is a rectangular matrix whose non-zero elements on the "diagonal" are the singular values  $\sigma_i, i = \{1, \dots, p\}$  of  $A$  and are the square roots of the eigenvalues of both  $A^T A$  and  $AA^T$ . A common convention is to arrange the  $\sigma_i$  values in descending order, then the matrix  $\Sigma$  is uniquely determined by  $A$  (**but  $U$  and  $V$  are not**).

We have the following relations between  $u_i$  and  $v_i$ :

$$\text{for } i \leq p, v_i = \frac{1}{\sigma_i} A^T u_i \text{ and } u_i = \frac{1}{\sigma_i} A v_i \quad (1)$$

In image processing, the singular values represent the energy of the image. Indeed, the total energy of the image  $I \in \mathbb{R}^{q \times p}$  is represented by the following formula:

$$\|I\| = \text{Trace}(I^T I) = \sum_{i=1}^q \sum_{j=1}^p I_{ij}^2 = \sum_{i=1}^q \sigma_i^2$$

**Theorem: Best low-rank approximation**

Let  $A \in \mathbb{R}^{q \times p}$  which SVD is  $A = \sum_{i=1}^p \sigma_i u_i v_i^T$  with  $p = \text{rank}(A)$ .

If  $k < p$  and  $A_k = \sum_{i=1}^k \sigma_i u_i v_i^T$  then  $A_k$  is the best low-rank approximation of  $A$  of rank  $k < p$  i.e.:

$$\min_{\text{rank}(D)=k} \|A - D\|_F = \|A - A_k\|_F$$

## 2 Image Compression

Let  $I$  of size  $q \times p$  be the matrix describing the image.

Let  $k < p < q$ , we can rebuild the image  $I$  using its  $k$ -low-rank approximation  $I_k$ .

To compute the vectors and the singular values needed to achieve this reconstruction, we can either:

1. perform the *SVD* decomposition of  $I$
2. or, construct  $M = I I^T$  (or  $M = I^T I$  according to the sizes  $q$  and  $p$ ) and compute the  $k$  dominant eigenpairs of  $M$  then using relations (1), compute the other set of vectors (again according to the size  $p$  and  $q$ ).

Once we have the  $k$  singular values, the first  $k$  vectors of  $U$  and  $V$ , compute  $I_k$ .

We talk of image compression because instead of having the image represented by an array of size  $q \times p$ , it is now a triplet  $(\Sigma_k, U_k, V_k)$  with less values when  $k < p < q$ .

**Remark:**  $k$  could correspond to a number of eigenvalues to achieve a percentage of the trace.

**TODO :**

**Question 1:** what are the size of the elements the triplet  $(\Sigma_k, U_k, V_k)$ ? What happens when  $q < p$ ?

**Question 2:** Using the different functions that compute the eigenpairs of a matrix (*eig*, *power method*, the 4 versions of the *subspace iteration method*), implement the alternative 2 of the reconstruction while maintaining the displays and difference calculation:

- reconstruct the image, noted  $I_k$  based on the  $k$ -low rank approximation;
- compute the difference between  $I$  and  $I_k$  with the RMSE (Root Mean Square Error).

In your report, you will present the results of the different functions with some variation of parameters in a table and add some comments. Be careful to test your code with all the images (landscape and portrait layout)

**Bonus question :** Dealing with the case of color images

