MATH 426 - Assignment 7 Part II: Image Compression using SVD

1 Preliminary Comments

Singular Value Decomposition (SVD) is a powerful idea from linear algebra which aims to extract useful information about a matrix. SVD has many theoretical and practical applications one of which we will see in this assignment. In what follows, we will use SVD in the context of image processing, where we read a gray-scale image into a matrix and will try to use SVD to analyze the image.



2 Background Information

In this section, we describe SVD briefly (for more information about SVD the reader can refer to any standard linear algebra text). The treatment that follows is based on that of [1], however, readers can refer to [2] as another resource.

We start by the following formal definition of SVD [2].

Definition: Let A be any $m \times n$ matrix. A singular value decomposition of A is a factorization of the following form:

$$A = U\Sigma V^T,$$

where

- U is an $m \times m$ orthogonal matrix,
- V is an $n \times n$ orthogonal matrix,
- Σ is an $m \times n$ diagonal matrix.

Moreover, it is assumed that diagonal entries of Σ are non-negative and we have $\Sigma_{ii}=\sigma_i$ with

$$\sigma_1 \ge \sigma_2 \ge \sigma_3 \ge \cdots \ge \sigma_p \ge 0,$$

where p = min(m, n). We call σ_i singular values of A.

That every $m \times n$ matrix has a singular value decomposition is a fundamental theorem in matrix analysis. For the sake of completeness, we state the main SVD theorem below [2].

Theorem (Singular Value Decomposition Theorem): Suppose A is a $m \times n$ matrix. Then A has a singular value decomposition. Moreover, singular values $\{\sigma_i\}$ of A are unique. If A is square and σ_i are distinct the left and right singular vectors $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ are uniquely determined up to complex signs.

The following result is a useful characterization of the rank of a matrix.

Theorem: Let A be an $m \times n$ matrix, with singular values

$$\sigma_1 \ge \sigma_2 \ge \sigma_3 \ge \cdots \ge \sigma_p \ge 0,$$

where p = min(m, n). Let $r \leq p$ be the number of non-zero singular values of A. Then, rank(A) = r.

In general, computing SVD of a matrix numerically is not an easy task. However, Matlab provides the **svd** function which computes the SVD of a given matrix. To find out more about Matlab's **svd** function the reader can refer to Matlab's help facility.

In our discussion of image processing we will need the following form of the SVD Theorem.

Theorem: Let A be a non-zero $m \times n$ matrix of rank r, with singular values $\{\sigma_1, \ldots, \sigma_r\}$, left singular vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$, and right singular vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$. Then,

$$A = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T.$$
(1)

Note that the above Theorem provides a way of getting approximations of A as follows. For $k \leq r$ (r is the rank of A), denote A_k by

$$A_k := \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^T.$$
⁽²⁾

Note that when σ_i is very small, the term $\sigma_i \mathbf{u}_i \mathbf{v}_i^T$ in the above sum becomes negligible. Moreover, since singular values of A are in descending order (stored as diagonal entries of Σ in $A = U\Sigma V^T$), once σ_k becomes sufficiently small, we can stop at that value of k and use A_k as a reasoable approximation to A. The definition of A_k in (2) will be important in the programming problem that will be specified in the next Section.

3 The Statement of the Problem

Your task is to write a Matlab function image_proc.m which loads the image date stored in A.mat (which you will download from the course web-page) into a matrix A and does the following:

- Compute and display rank of A (use the rank function).
- Compute the singular value decomposition of A.
- Compute rank k approximations A_k as explained in the previous section for six different values of k.
- For each A_k , display the corresponding (approxmate) image; you will need to report the rank k in the title of the corresponding figure. Organize your images in a 3×2 grid using subplots (Recall Matlab's subplot command).
- Also, in a separate figure, make a plot of singular values of A to see how the size of singular values drop at some point (this gives you an idea of for what k, A_k is a reasonably close approximation to A).

Organize your program in the following way:

• All code must be in a function called image_proc.m.

• You will need a sub-function called get_rank_k with the following interface:

```
Ak = get_rank_k(U, S, V, k)
```

The function get_rank_k returns the rank k approximation A_k given the singular value decomposition of A (the matrices U, S, and V) and the rank k. Note that the matrices U and V are the same as those in the singular value decomposition $A = U\Sigma V$ and the matrix S corresponds to Σ . Note that here is where you will need to use Equation (2). Note: The function get_rank_k must be vectorized (no loops).

In image_proc you will load the image data using the load command; To display the images you will need to use pcolor(A) (where A stores the image data).

4 Hints/Suggestions

As mentioned in the previous section, the function get_rank_k must be vectorized; however, as a first try you may want to implement the function using a for loop which is straightforward as the form of the Equation (2) suggests. After getting everything to work, you can vectorize the function (which makes it remarkably elegant).

5 Deliverables

Once you have written the code that solves the problem:

- Write a *brief* report where you explain the design of your code clearly. Also, list some observations based on your computational results. The report must be no longer than one page.
- The plots produced by your code
- All computer code used

6 References

- 1. Trefethen, L. and Bau, D. "Numerical Linear Algebra" (1997).
- 2. Watkins, D. "Fundamentals of Matrix Computations" (2006).