

## 1 Theory

- Let  $W_1$  and  $W_2$  be vector subspaces and  $W = W_1 + W_2$ ,  $W_1 \neq W_2$ . Show, by giving an example, that the decomposition of a vector  $\mathbf{x} \in W$  is not unique, i.e.,

$$\mathbf{x} = \mathbf{w}_1 + \mathbf{w}_2 = \mathbf{w}'_1 + \mathbf{w}'_2,$$

where  $\mathbf{w}_1 \neq \mathbf{w}'_1$ ,  $\mathbf{w}_2 \neq \mathbf{w}'_2$ ,  $\mathbf{w}_1, \mathbf{w}'_1 \in W_1$ ,  $\mathbf{w}_2, \mathbf{w}'_2 \in W_2$ .

- Consider the matrix

$$A = \begin{pmatrix} 1 & -1 \\ 2 & -2 \\ 3 & -3 \end{pmatrix}.$$

Determine **bases** for the column space, row space, null space, and left null space of  $A$ .

- Let  $V = \mathbb{R}^3$ , let

$$\mathbf{u}^{(1)} = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}, \quad \mathbf{u}^{(2)} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix},$$

and define  $W = \text{span}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)})$ . Find the orthogonal projection of  $\mathbf{x}$  onto  $W$ . Also find the projection matrix  $\mathbb{P}$  associated with this mapping.

- Reconsider Problem 3. Find vectors such that  $x = UU^T x$  and  $x \neq UU^T x$  where the matrix  $U$  consists of the **orthonormal basis vectors** of  $W$  from Problem 3.
- Determine the SVD of the data matrix

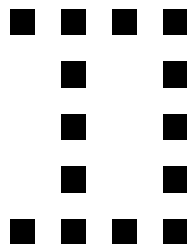
$$\begin{pmatrix} -2 & -1 & 1 \\ 0 & -1 & 0 \\ -1 & 1 & 2 \\ 1 & -1 & 1 \end{pmatrix},$$

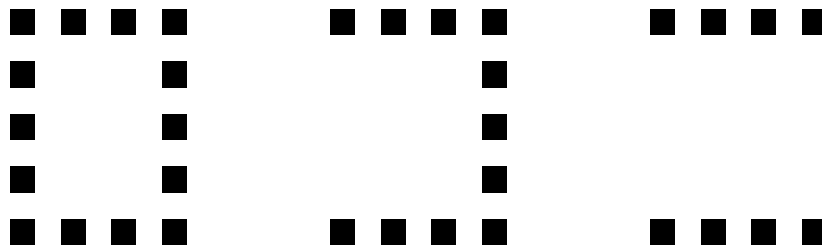
and compute the rank-one, -two, and -three approximations to  $A$ .

## 2 Computing

- Consider the training set consisting of the following three patterns consisting of  $5 \times 4$  arrays of black squares

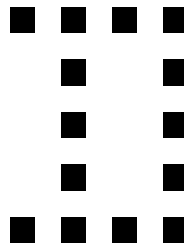
Using Kohonen's novelty filter, find and display (in terms of an image) the novelty in the pattern





Proceed by assuming that the black square entries have numerical value one and the blank entries have numerical value zero. Concatenate the columns of each pattern to make vectors in  $\mathbb{R}^{20}$ . Does your result make sense? Why or why not?

2. Compute the SVD of the matrix  $A$  whose entries come from the pattern



and display (in terms of an image) the reconstructions  $A_1, A_2, A_3, A_4$ . Again, treat the squares as ones and the blanks as zeros. Your reconstructions should be matrices with numerical values. Interpret your results.

3. This assignment requires the use of a MATLAB image. Choose your favorite image for this exercise (be sure to choose an image whose resolution is at least 300-by-300.). All the necessary MATLAB syntax is described as follows. To begin, load the MATLAB image into the matrix  $A$  using

```
>> A = imread('myImage.tif');
% adjust accordingly with the image extension
```

If your image is in color,  $A$  will have three dimensions where the first two give the resolution of the image and the last one contains a layer of red, a layer of green, and a layer of blue. To turn a color image into a monochrome one, use the MATLAB command

```
>> B = rgb2gray(A);
```

You don't necessarily have to work with a black and white picture, but it is definitely easier and computationally cheaper to start with one. The full and reduced SVD may then be executed simply by

```
>> A = double(A); % data matrix has to be in double precision
in order to perform mathematics on it
```

```
>> [U,S,V] = svd(A);
>> [U_thin,S_thin,V_thin] = svd(A,0);
%% the zero (not the letter "o") indicates economy size
%% 'double' converts any 8-bit single (uint8) into 16-bit
    double precision
```

where  $U$  (resp.  $U_{\text{thin}}$ ),  $S$  (resp.  $S_{\text{thin}}$ ), and  $V$  (resp.  $V_{\text{thin}}$ ) are the left-singular vectors, the singular values, and the right-singular vectors (resp. reduced). A rank- $k$  approximation of the image may be found via

```
>> A_k = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
```

The resulting image may be displayed using

```
>> imagesc(A_k);
% may use the option: axis off, axis square for better
    display
```

or

```
>> image(A_k);
```

For better viewing, one can reset the colormap to gray scales with the command

```
>> imagesc(A_k), colormap(gray)
```

Now, do the following:

- (a) Plot the singular value distribution of your image, where the  $x$ -axis is the counting index while the  $y$ -axis is the magnitude of the singular values. If we define the **cumulative energy** of  $A \in \mathbb{R}^{m \times n}$  with rank  $r$  to be

$$E_k = \frac{\sum_{i=1}^k \sigma_i^2}{\sum_{i=1}^r \sigma_i^2}, \text{ where } k \leq r,$$

identify the number of singular values ( $\sigma_i$ s) needed to retain at least 95% of the energy. This number is often called *numerical rank* of the matrix  $A$ . Is the numerical rank for your image large or small? Explain why.

- (b) Compute the rank-10, rank-50, rank-100, and rank-200 approximations to your chosen image along with the *relative errors* of approximation (use the title, xlabel, and ylabel commands to specify appropriate information), display them on the same figure (using subplot), and interpret your results. (Recall that the *absolute* error of a rank- $k$  approximation is measured by the  $k+1^{\text{th}}$  singular value, so the *relative* error is given by  $\sigma_{k+1}/\sigma_1$ .)