Math 521 Homework 2

Due on Tuesday, March 6, 2018

Kristin Holmbeck

Contents

Theory **2**
1. Unique Decomposition . . . 2
2. Matrix Bases . . . 2
3. Projections . . . 3
4. Orthonormal Basis Vectors . . . 4
5. SVD . . . 4

Computing **6**
1. Kohonen's Novelty Filter . . . 6
2. SVD on a boolean matrix . . . 7
3. (a) SVD on a higher-resolution image . . . 9
3. (b) Lower-Rank Approximations . . . 12

Code **14**
Gram-Schmidt . . . 14

List of Figures

1 Training Data for Kohonen . . . 6
2 Pattern to compare with training set from Figure 1 . . . 6
3 Novelty from Kohonen's filter . . . 7
4 Reconstruction of A by A_1 . . . 8
5 Test Image: Palos Verdes Half Marathon (960×1440) . . . 9
6 Singular Value Distribution (first 100 of 960) of Figure 5 . . . 10
7 Cumulative Energy of Figure 5 . . . 10
8 Rank-6 Approximation of Figure 5 . . . 11
9 Rank-47 Approximation of Figure 5 . . . 12
10 Lower-Rank Approximations of Figure 5 . . . 13

Theory

1. Unique Decomposition

Let W_1, 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 $\boldsymbol{x} \in W$ is not unique.

The requirements for a subspace $\hat{W}$ include :

- The zero vector is in $\hat{W}$
- If $\boldsymbol{u}, \boldsymbol{v} \in \hat{W}$, then $\boldsymbol{u} + \boldsymbol{v} \in \hat{W}$
- If $\boldsymbol{u} \in \hat{W}, c \in \mathbb{R}, c\boldsymbol{u} \in \hat{W}$

If we let $W = \{\begin{bmatrix} x & y & 0 \end{bmatrix}^T \ni x, y \in \mathbb{R}\} \subset \mathbb{R}^3$, then we can decompose W into $W_1 = \{\begin{bmatrix} x & 0 & 0 \end{bmatrix}^T \ni x, y \in \mathbb{R}\}$ and $W_2 = \{\begin{bmatrix} x & y & 0 \end{bmatrix}^T \ni x, y \in \mathbb{R}\}$. Then, we can express a vector $\boldsymbol{x} \in W$ non-uniquely. For example:

$$\boldsymbol{x} = \begin{bmatrix} 2 \\ 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 10 \\ 0 \end{bmatrix}$$

2. Matrix Bases

Consider the matrix

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

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

- The column space of A is the linearly independent columns in A. Since the second column is a scalar multiple of the first (by -1), the column space of A is:
  $\text{span}\left\{ \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \right\}$.
- The row space of A is the linearly independent rows of A. Notice that rows 2 and 3 are scalar multiples of the first row. Hence, the row space is:
  $\text{span}\left\{ \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\}$.

- The null space of A includes the vectors that solve $Ax = \mathbf{0}$. Then it is easy to see that $A \begin{bmatrix} x_1 \\ x_1 \end{bmatrix} = \mathbf{0}$ solves this. In other words,
  null space of $A = \text{span} \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$.

- The *left* null space of A are the vectors that solve $x^T A = 0$.
  $\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 2 & -2 \\ 3 & -3 \end{bmatrix} = \begin{bmatrix} x_1 + 2x_2 + 3x_3 & -x_1 - 2x_2 - 3x_3 \end{bmatrix} = \mathbf{0}$
  The most direct way to solve this is to convert $[A \quad | \quad I_3]$ to reduced-row echelon form. Performing this calculation, we obtain the basis for the left null space:
  $\text{span} = \left\{ \begin{bmatrix} 1 \\ 0 \\ -\frac{1}{3} \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ -\frac{2}{3} \end{bmatrix} \right\}$

3. Projections

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

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

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

For orthogonal projections, the Gram-Schmidt process is used – a method for generating an orthonormal basis from a set of vectors. Using the Gram-Schmidt algorithm (in the Code section of this document), we obtain the basis vectors:

$$e^{(1)} = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}, \qquad e^{(2)} = \frac{1}{3\sqrt{5}} \begin{bmatrix} -4 \\ 2 \\ 5 \end{bmatrix}$$

A projection matrix $\mathbb{P}_i$ onto $e^{(i)}$, is given by $e^{(i)} {e^{(i)}}^T$. From this, we have the two projection matrices for the basis vectors above:

$$\mathbb{P}_1 = \frac{1}{5} \begin{bmatrix} 1 & 2 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbb{P}_2 = \frac{1}{45} \begin{bmatrix} 16 & -8 & -20 \\ -8 & 4 & 10 \\ -20 & 10 & 25 \end{bmatrix}$$

$$\text{then the total projection is: } \mathbb{P} = \mathbb{P}_1 + \mathbb{P}_2 = \frac{1}{9} \begin{bmatrix} 5 & 2 & -4 \\ 2 & 8 & 2 \\ -4 & 2 & 5 \end{bmatrix}$$

Lastly, note that $x = u^{(1)} + u^{(2)}$, in other words $x \in W$, so we expect the orthogonal projection of x onto W to be itself. Indeed it is, as $\mathbb{P}x = x$.

4. Orthonormal Basis Vectors

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.

Note that $UU^T = \mathbb{P}$ since

$$\begin{aligned} U &= \begin{bmatrix} e^{(1)} & e^{(2)} \end{bmatrix} \\ UU^T &= \begin{bmatrix} e^{(1)} & e^{(2)} \end{bmatrix} \begin{bmatrix} {e^{(1)}}^T \\ {e^{(2)}}^T \end{bmatrix} \\ &= \begin{bmatrix} e^{(1)} {e^{(1)}}^T + e^{(2)} {e^{(2)}}^T \end{bmatrix} = \mathbb{P} \qquad \text{by construction} \end{aligned}$$

As stated previously, the given vector $x = [0, 2, 1]^T$ solves $\mathbb{P}x = x$. In order to find a $y \ni \mathbb{P}y \neq y$, we will use a $y \notin W$. Let $y = [1, 1, 1]^T$. Then $\mathbb{P}y = \frac{1}{3}[1, 4, 1]^T \neq y$.

5. SVD

Determine the SVD of the data matrix

$$A = \begin{bmatrix} -2 & -1 & 1 \\ 0 & -1 & 0 \\ -1 & 1 & 2 \\ 1 & -1 & 1 \end{bmatrix}$$

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

We will compute the SVD of A "manually", that is, we will show the steps for how to compute the SVD, but leave the heavy lifting to MATLAB. First, consider the characteristic equation for $A^T A$:

$$\begin{aligned} A^T A &= \begin{bmatrix} 6 & 0 & -3 \\ 0 & 4 & 0 \\ -3 & 0 & 6 \end{bmatrix} \\ \rho(A^T A) &= |A^T A - \lambda I| = -\lambda^3 + 16\lambda^2 - 75\lambda + 108 = 0 \\ 0 &= (\lambda - 9)(\lambda - 4)(\lambda - 3) \qquad \text{by long division} \\ \lambda &= \{9, 4, 3\} \end{aligned}$$

Since our A matrix is 4x3, Σ will be the same size.

$$\Sigma = \begin{bmatrix} \sqrt{9} & 0 & 0 \\ 0 & \sqrt{4} & 0 \\ 0 & 0 & \sqrt{3} \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1.7321 \\ 0 & 0 & 0 \end{bmatrix}$$

We obtain the null spaces corresponding to the matrix $(A^TA - \lambda I)$, and use these to form the V matrix.

$$V = \begin{bmatrix} -0.7071 & 0 & -0.7071 \\ 0 & -1 & 0 \\ 0.7071 & 0 & -0.7071 \end{bmatrix}$$

Lastly, we begin construction of the U matrix using $u_i = \frac{1}{\sigma_i} Av_i$ as the first several column vectors.

$$AV\Sigma^{-1} = \hat{U} = \begin{bmatrix} 0.4082 & 0.5000 & 0.7071 & 0 \\ 0 & 0.5000 & 0 & 0 \\ -0.4082 & -0.5000 & 0.7071 & 0 \\ -0.8165 & 0.5000 & 0 & 0 \end{bmatrix}$$

The last column of U can be obtained via the same eigenvalue method on AA^T. We know that this corresponds to an eigenvalue of zero, so by computing the null space of AA^T, we obtain the full matrix for U:

$$U = \begin{bmatrix} 0.4082 & 0.5000 & 0.7071 & -0.2887 \\ 0 & 0.5000 & 0 & 0.8660 \\ -0.4082 & -0.5000 & 0.7071 & 0.2887 \\ -0.8165 & 0.5000 & 0 & -0.2887 \end{bmatrix}$$

Then A can be decomposed into $A = U\Sigma V^T$ by the singular value decomposition theorem. Checking this with MATLAB yields the same decomposition. (Note: the decomposition is not *exactly* the same, but differs in placement of some negative signs.)

In order to determine low-rank approximations, we will denote the rank-k approximation to A, called A_k, by:

$$A_k = \begin{bmatrix} u_{11} & \cdots & u_{1k} \\ \vdots & \ddots & \vdots \\ u_{k1} & \cdots & u_{kk} \end{bmatrix} \begin{bmatrix} s_{11} & & 0 \\ & \ddots & \\ 0 & & s_{kk} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{1k} \\ \vdots & \ddots & \vdots \\ v_{k1} & \cdots & v_{kk} \end{bmatrix}^T$$

$$A_1 = \begin{bmatrix} -1.5 & 0 & 1.5 \\ 0 & 0 & 0 \\ -1.5 & 0 & 1.5 \\ 0 & 0 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} -1.5 & -1 & 1.5 \\ 0 & -1 & \epsilon \\ -1.5 & 1 & 1.5 \\ 0 & -1 & \epsilon \end{bmatrix} \quad A_3 = \begin{bmatrix} -2 & -1 & 1 \\ 0 & -1 & \epsilon \\ -1 & 1 & 2 \\ 1 & -1 & 1 \end{bmatrix}$$

Where $\epsilon = \mathcal{O}(1e-15)$, machine-precision zero with roundoff error. An interesting result from this is the spectral norm of the residual $A - A_k$ is equal to the $(k+1)$-th singular value of A. This theorem comes from [3].

$$||A - A_1||_2 = 2 = S_{22}$$
$$||A - A_2||_2 = 1.7321 = S_{33}$$
$$||A - A_3||_2 = 0 \qquad \text{is exact (up to machine precision), as expected}$$

Computing

1. Kohonen's Novelty Filter

Consider the training set consisting of the following three patterns consisting of 5×4 arrays of black squares (Figure 1). Using Kohonen's novelty filter, find and display (in terms of an image) the novelty in the pattern from Figure 2.

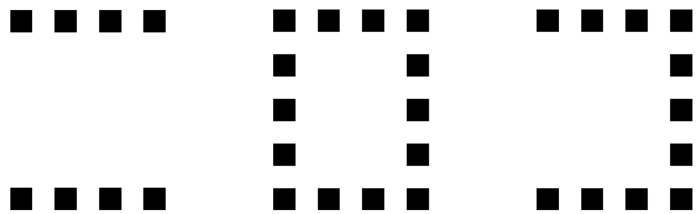

Figure 1: Training Data for Kohonen

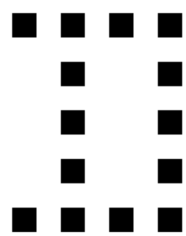

Figure 2: Pattern to compare with training set from Figure 1

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?

We can decompose a vector x into its orthogonal projection and the complement of its orthogonal projection, i.e. $x = w + w^T = \mathbb{P}x + (I - \mathbb{P})x$. From Kohonen [2], the novelty is the orthogonal complement, w^T.

In order to calculate the novelty, we find an orthonormal basis for the first three patterns. We create a matrix of size (20 x 3), and obtain an orthonormal basis W. The projection matrix then becomes $\mathbb{P} = WW^T$, and the novelty is then $(I - \mathbb{P})x$, where x is the (20 x 1) vector obtain from concatenating the columns of the pattern in Figure 2.

This gives us the novelty shown in Figure 3, corresponding to the matrix below:

$$(I - \mathbb{P})x = w^T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 3: Novelty from Kohonen's filter

Does this result make sense? Our comparison data is a linear combination of the training data, other than the vertical column appearing in the middle of the matrix. It makes sense then that our novelty, or difference, appears in that column.

2. SVD on a boolean matrix

Compute the SVD of the matrix A whose entries come from the first pattern in Figure 1 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.

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = USV^T \qquad \text{computed via MATLAB}$$

In particular, note the construction of the singular value matrix

$$S = \begin{bmatrix} 3.4641 & 0 & 0 & 0 \\ 0 & 1.4142 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

With this knowledge, we expect the rank-2 and greater approximations to be "very close" ($||A - A_k||$ is small) to the original matrix A. Indeed, this is the case

$$A_1 = \begin{bmatrix} 0.6 & 1.2 & 0.6 & 1.2 \\ 0.4 & 0.8 & 0.4 & 0.8 \\ 0.4 & 0.8 & 0.4 & 0.8 \\ 0.4 & 0.8 & 0.4 & 0.8 \\ 0.6 & 1.2 & 0.6 & 1.2 \end{bmatrix}$$

$$A_2 = A_3 = A_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \epsilon & 1 & \epsilon & 1 \\ \epsilon & 1 & \epsilon & 1 \\ \epsilon & 1 & \epsilon & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$||A - A_1|| = \sqrt(2) = 1.4142$$

$$||A - A_2|| = ||A - A_3|| = ||A - A_4|| = \epsilon$$

For brevity, let's look at the image representation of just A_1, which we know is not "good" compared to the other, higher-rank approximations.

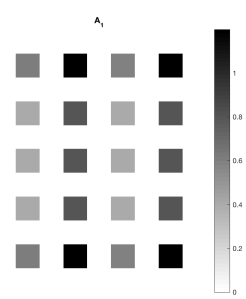

Figure 4: Reconstruction of A by A_1

You can see some residuals in spaces where we expect/want the matrix to be zero. Since this is a rank-1 approximation and σ_1 is not much greater than σ_2, the rank-1 approximation is not expected to be accurate.

3. (a) SVD on a higher-resolution image

Figure 5: Test Image: Palos Verdes Half Marathon (960 × 1440)

Computing the cumulative energy E via $E_k = \dfrac{\Sigma_{i=1}^{k} \sigma_i^2}{\Sigma_{i=1}^{k} \sigma_i^2}$ with $r = \text{rank(image)}$ and $k \leq r$, we obtain the *numerical rank* of the image in Figure 5 as 6; the number of singular values required to retain at least 95% of the energy in the original image. The cumulative energy is shown in Figure 7. What this actually looks like in terms of reconstruction is shown in Figure 8.

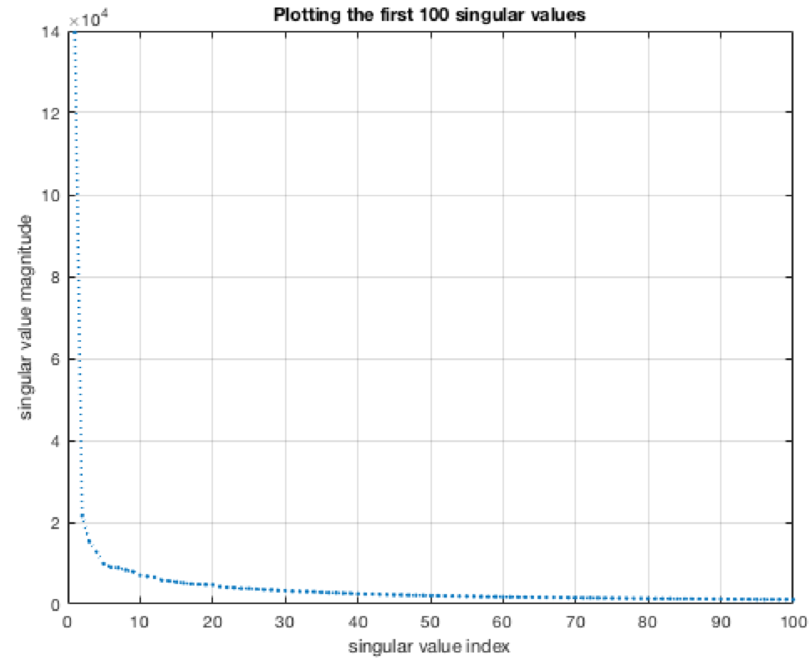

Figure 6: Singular Value Distribution (first 100 of 960) of Figure 5

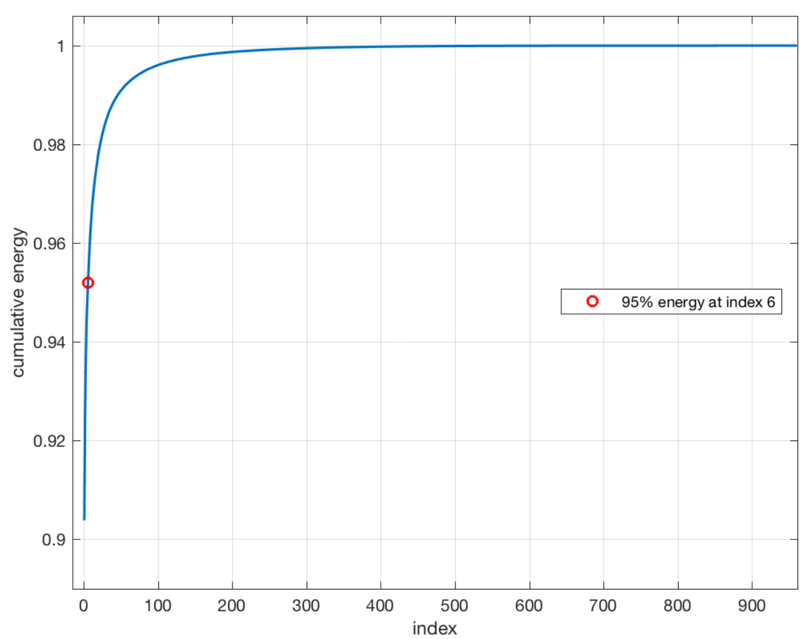

Figure 7: Cumulative Energy of Figure 5

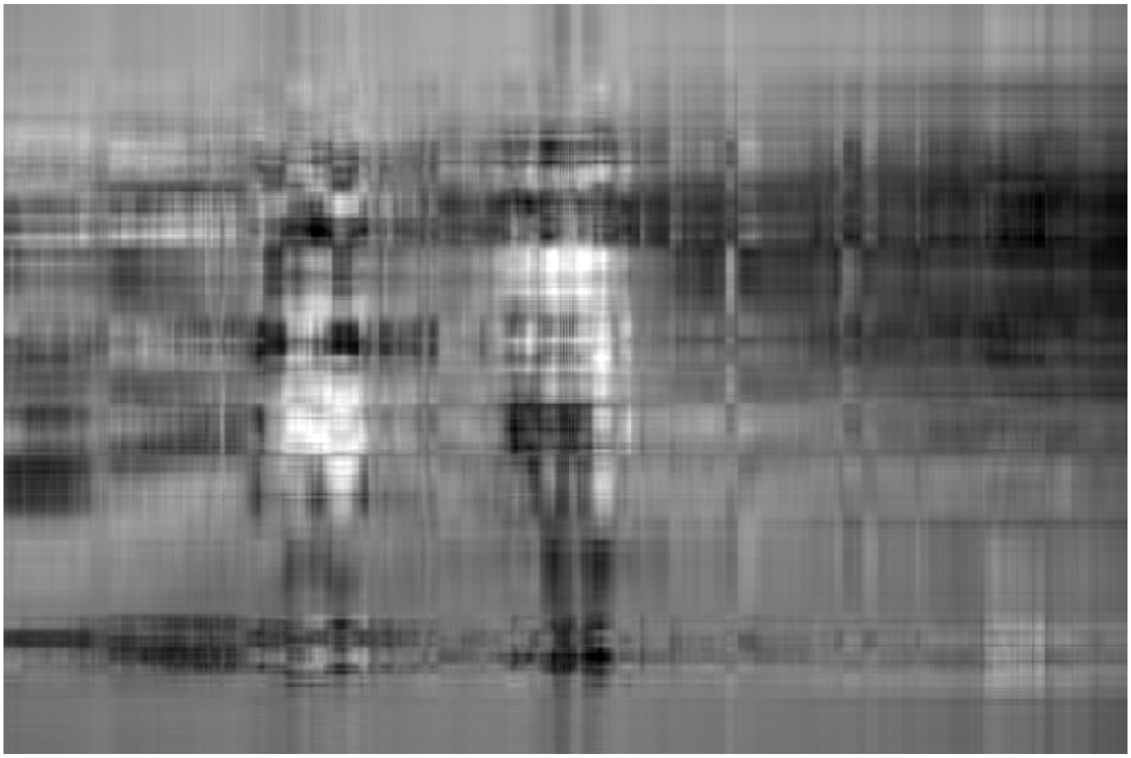

Figure 8: Rank-6 Approximation of Figure 5

In fact, in order to retain 99% of the energy, we need a rank-47 approximation. The resulting reconstruction is given in Figure 9.

Figure 9: Rank-47 Approximation of Figure 5

3. (b) Lower-Rank Approximations

Recall that the relative error of a rank-k approximation is given by σ_{k+1}/σ_k. As show in Figure 10, the relative errors τ_i are:

$$\begin{aligned}\tau_{10} &= 0.9579\\ \tau_{50} &= 0.9798\\ \tau_{100} &= 0.9816\\ \tau_{200} &= 0.9954\end{aligned}$$

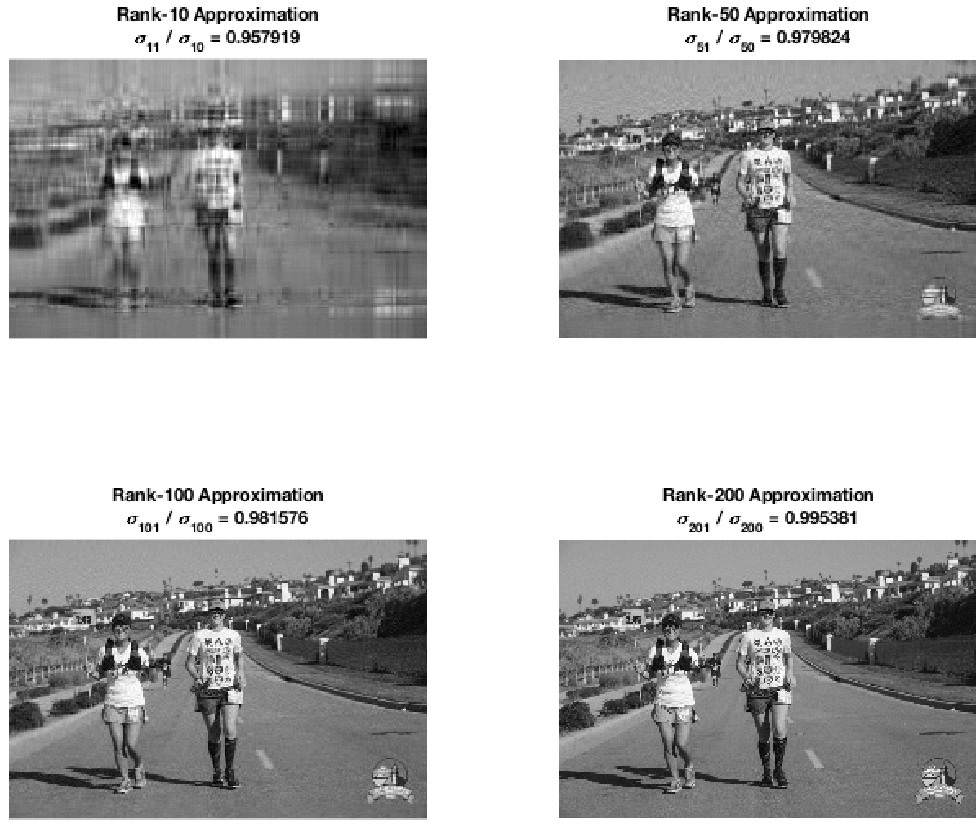

Figure 10: Lower-Rank Approximations of Figure 5

Code

Gram-Schmidt

```
clear;

A = [1 -1; 2 0 ; 0 1];
x = [0; 2; 1];

%
% Perform the Gram-Schmidt algorithm
%

% take the first normalized col vector to be the first orth. basis vector
u1 = A(:,1);
u1 = u1 ./ norm(A(:,1));

B = zeros(size(A));
B(:,1) = u1;

% Define the projection function
proj = @(u,v) (u'*v) / (u'*u) * u;

for ii = 2:size(A,2)
    tmp = 0;
    % sum the projections
    for jj = 2:ii
        tmp = tmp + proj(B(:,jj-1), A(:,ii));
    end
    % tmp = (current vector) - (sum of projections)
    tmp = A(:,ii) - tmp;

    % normalize, and set as new orth. basis vector
    B(:,ii) = tmp / norm(tmp);
end
```

References

[1] Chang, Jen-Mei. *Matrix Methods for Geometric Data Analysis and Recognition*. 2014.

[2] T. Kohonen. Self-Organization and Associative Memory. Springer-Verlag, Berlin, 1984.

[3] Manning, Christopher D. and Raghavan, Prabhakar, and Schtze, Hinrich. *Introduction to Information Retrieval*. Cambridge University Press. 2008. [online] Available at: https://nlp.stanford.edu/IR-book/html/htmledition/low-rank-approximations-1.html [Accessed 25 Feb. 2018].