SVD for IMAGE and VIDEO FILTERING

Ashwin Swaminathan
ashwins@eng.umd.edu

In this project, we will study the application of SVD based techniques for image and video filtering. In the first couple of problems in this case study, you will learn to design filters using the SVD for a particular purpose. You will then use these filters for image denoising. The Truncated SVD method for image filtering is then presented. Finally, in the last few problems, you will learn to apply these techniques for video filtering.

Singular Value Decomposition for Filtering

As discussed in Chapter 5, the Singular Value Decomposition (SVD) is a very good tool for representation. The SVD of a $M \times N$ matrix A can be written as

$$A = U \Sigma V^T \tag{1}$$

where U and V are unitary matrices of dimension $M \times M$ and $N \times N$ respectively and Σ is a diagonal matrix of size $M \times N$. The superscript T on V stands for transpose. (Note: in the general case of complex matrices the transpose must be replaced by the conjugate transpose).

Let u_i be the columns of U and v_i be the columns of the matrix V. Expanding the product in equation 1, we obtain

$$A = \sum_{i=1}^{K} \rho_i u_i v_i^T \tag{2}$$

where ρ_i is a diagonal element of Σ. Hence, we see that we can represent an image A as a linear combination of the basis images $(u_i v_i^T)$. All the basis images are rank one and form an orthonormal basis for image representation. Here K is the rank of the matrix A. Without loss of generality, we will always assume that the ρ_i's are arranged in descending order. That is

$$\rho_1 \geq \rho_2 \geq \ldots \geq \rho_K \tag{3}$$

Note: The SVD is only one such tool for image representation. There are other transforms that be used to represent the image in terms of basis images. The difference between the SVD and the other transforms is that in SVD we choose the basis images to be dependent on the image. We will only consider the use of the SVD in this case study.

First, we shall consider various kinds of image filtering techniques to demonstrate the principle of using SVD for image filtering. We will show that the SVD can be used to design different kinds of filters (including both high pass and low pass filters).

The most prominent of these methods is *Linear Weighting* method.

Linear Weighting The linear weighting method can be used to design both low-pass and high pass filters. The SVD of the image is first obtained. A weighted averaging of the SVD components is done to obtain the filtered images. The filtering process can be mathematically expressed as

$$\hat{A} = \sum_{i=1}^{K} \rho_i \beta_i u_i v_i^T \tag{4}$$

where β_i are the weights. The type of the filtering operation and the kind of the resulting filter would depend on the relative values of the constants β_i. In the first problem, we would study the effect of these constants on the filtering operation.

Problem 1. In this problem, you will have to write MATLAB codes to implement the following. Attach the results of your simulation on the boat (boat.bmp) and the baboon image (baboon.bmp) along with your report.

1. First use the SVD function to obtain the basis images. Have a look at the basis images 1, 10, 100, 200. What do you observe ?
2. Plot the singular values of both the images? What do you observe? Explain your observations in terms of basis images.
3. Now implement the *linear weighting* method for (a) $\beta_i = i$ and (b) $\beta_i = (K - i + 1)/K$. What do you observe from your results? What kind of filters do you obtain in in (a) and (b)? (Note: You may have to appropriately scale your reconstructed image $\hat{A}$ for viewing purposes. You may use the imshowsc.m file for display purposes).

Non-Linear Weighting methods

Other kinds of filters exist depending on the kind of weighting applied to the singular values during the reconstruction process. Non-Linear weighting methods are another class of methods used for filtering purposes. In non-linear weighting methods, the singular values are raised to an exponential power on reconstruction. This is shown in following equation.

$$\hat{A} = \rho_1^{1-\alpha} \sum_{i=1}^{K} \rho_i^{\alpha} u_i v_i^T \tag{5}$$

Problem 2. In this problem, implement the non-linear filtering scheme for different values of $\alpha = 0.5, 1, 1.2$ on the boat image.

1. What kind of filter do you obtain in the three cases?
2. Which value of α is best for noise-removal applications?

A closer look at the two methods would reveal that we have designed various kinds of filters only by appropriately changing Σ. We have then reconstructed the filtered image as

$$\hat{A} = U\hat{\Sigma}V^T \tag{6}$$

Scheme 1: Truncated SVD methods

In both the earlier methods, we used all the values of the matrix Σ and thus all the basis images for reconstruction. In the subsequent methods, we will design filters particularly for noise removal purposes. This leads to truncated SVD methods for filtering.

Problem 3. In this problem, first show that for any matrix A,

$$||A||_F = ||\Sigma||_F \tag{7}$$

where $||.||_F$ denotes the Frobenius norm of the matrix and is given by

$$||A||_F = \left(\sum_{i=1}^{M} \sum_{j=1}^{N} |a(i,j)|^2 \right)^{1/2} \tag{8}$$

Now, use the above result to show that the best rank R approximation of the matrix A can be obtained by reconstructing the image based on the first R significant basis images. i.e.

$$\hat{A} = \sum_{j=1}^{R} \rho_j u_j v_j^T \tag{9}$$

Also show that Frobenius norm of the reconstruction error is given by

$$||E||_F = \left(\sum_{j=R+1}^{K} \rho_j^2 \right)^{1/2} \tag{10}$$

This gives rise to the truncated SVD filtering method. We approximate the filtered image using the first R basis images as shown in equation 9 above. The performance of the filtering method depends on the choice of the value of R used in the reconstruction.

Many algorithms exist to estimate the optimal value of R. In this case study, we will consider one of the main class of methods for estimating the value of R. In this class of methods, given the noisy version of the image, we estimate the SNR. We then compare the SNR with a predetermined threshold and find the optimal value of R so that the estimated SNR is as close to the threshold as possible. This method is discussed in the subsequent sections.

Estimating SNR based on SVD

In the first method, we estimate the SNR based on the SVD algorithm itself. To estimate the SNR of the signal in noise, we need to get a good estimate of the signal power and the noise powers. This problem is a classical problem of signal estimation in the presence of additive noise. Given the noisy signal Y, we need to estimate the original signal X where

$$Y = X + E \tag{11}$$

Here E is the additive noise. The method presented here is based on the fact that the signal matrix X is usually rank deficient. i.e. $rank(X) < min(N, M)$. We can also further assume that E is additive white gaussian noise and uncorrelated with the signal. That is

$$E^T E = \sigma^2 J \tag{12}$$

$$X^T E = 0 \tag{13}$$

J here stands for the identity matrix. We can express the SNR in terms of the Frobenius norm as

$$SNR = 20 \log_{10} \left(\frac{||X||_F}{||E||_F} \right) \tag{14}$$

We cannot directly compute the signal power $||X||_F$ and the noise power $||E||_F$ from our observations Y. So, we use approximate methods for estimating them. We will assume that the rank of the matrix X is R. Now, X can be written in terms of its SVD components as

$$X = [U_{x_1} U_{x_2}] \begin{bmatrix} S_x & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{x_1}^T \\ V_{x_2}^T \end{bmatrix} \tag{15}$$

where

$$\begin{aligned} U_{x_1} & \text{ is } M \times R, \\ U_{x_2} & \text{ is } M \times (M - R), \\ S_x & \text{ is } R \times R, \\ V_{x_1} & \text{ is } N \times R \text{ and} \\ V_{x_2} & \text{ is } N \times (N - R). \end{aligned}$$

Now, we will establish the mathematical basis for our SNR estimation algorithm.

Problem 4. In this problem, we will establish the mathematical basis for the SVD based SNR estimation algorithm.

1. Using the definitions in equation 15 along with the assumptions in equations 12 and 13, verify that Y can be expressed as

$$\begin{aligned} Y &= U_y S_y V_y^T \quad \text{where} \\ S_y &= \begin{bmatrix} \sqrt{S_x^2 + \sigma^2 J_R} & 0 \\ 0 & \sigma J_{N-R} \end{bmatrix} \end{aligned} \tag{16}$$

   where J_i denotes an identity matrix of dimensions $(i \times i)$. Give explicit expressions for U_y and V_y.

2. Deduce expressions for the singular values of the observed matrix Y. Express the singular values of the matrix X in terms of the singular values of the matrix Y. What do the last $(N-R)$ singular values of the matrix Y indicate ?

3. Now, using the definition of the SNR given in equation 14 and the results from problem 3, give an expression that estimates the SNR in terms of the singular values of the matrix Y. (Note: Your final result for the SNR must be only in terms of the singular values of the matrix Y that is observed and must not contain terms involving the singular values of X).

Usually, due to other kinds of truncation and round-off errors, the last $(N-R)$ singular values of the matrix Y may not all be equal to σ. So, we instead estimate σ using

$$\sigma_{est}^2 = \frac{1}{N-R} \sum_{i=R+1}^{N} \rho_{y_i}^2 \tag{17}$$

where ρ_{y_i} denotes the i^{th} singular value of the matrix Y.

Problem 5. Using the equations, you derived in problem 4, now write a MATLAB code for the truncated SVD method. Apply the method to boatNoise.bmp and baboonNoise.bmp and attach your results along with your report. You may choose your truncation point R_{opt} so that the estimated SNR is as close as possible in the L_1-norm sense to the predetermined threshold (use a threshold of 20).

1. What is the optimal value of R that you obtained in the two cases?

2. Which image require more basis images to attain the same SNR and why? (Note: Both the images boatNoise.bmp and baboonNoise.bmp have been artificially synthesized by adding the same amount of noise on boat.bmp and baboon.bmp respectively)

Scheme 2: SVD based filtering using Image rotations

Though, the truncated SVD method helps in removing the noise from the image to a particular extent, we still can observe traces of noise in the final

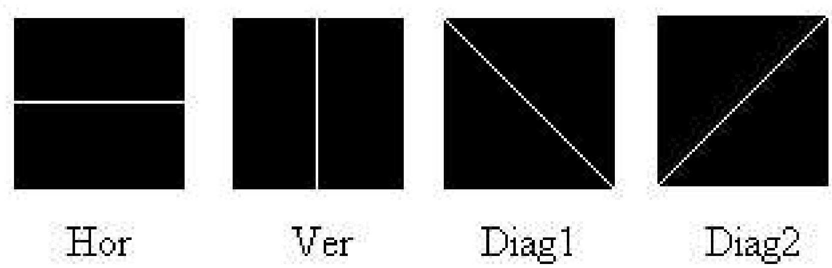

Figure 1: Images for Problem 6

output. In this section, we shall discuss another method based on truncated SVD for image filtering. The motivation behind the method would be clear from problem 6. This should provide you a better understanding on the kind of basis images produced by the SVD.

Problem 6. You are given a MATLAB data file (prob6data.mat). Load the file on MATLAB (You may use the load command in matlab to do this). This file has 8 images - Hor, Ver, Diag1, Diag2 (as shown in Fig. 1) and its noisy versions.

1. Do a singular value decomposition on all the 8 images and plot the singular values. (You may use the log scale for better clarity). What do you observe ?
2. Which image would require more number of basis vectors for its representation?
3. What is the effect of additive noise on the singular values?
4. What can you tell about the basis vectors produced by the SVD based your observations?
5. Now, use the truncated SVD method to reconstruct the image. You may use the 20 most significant basis images for reconstruction. For which image do you do you get the best results and why?

The truncated SVD based on image rotations is based on the principle of problem 6. As we have seen before, the SVD is a very good tool for image representation if the natural direction of the image is either horizontal or vertical. This must be clear from the results obtained in Problem 6. This means that we would require the least number of components if the image is aligned appropriately.

A simple way to make the principle directions of the image horizontal or vertical is to do appropriate image rotations. We can then use a lower rank approximation of the rotated image A_R using the SVD.

The image is initially rotated by t degrees to obtain an image A_t. A singular value decomposition is done on the rotated image (A_t). The rotated image is then reconstructed based on the first R basis images (call it $A_t^{(rec)}$). The reconstructed image $A_t^{(rec)}$ is rotated back to obtain the image A^{rec}. This process is repeated for different values of angle of rotation t and the results are averaged to produce the final filtered image.

Problem 7. Write a MATLAB program to implement the truncated SVD based on image rotations. You may use $R = 8$ and a step size of 5 degrees for the angle t. Attach the results on boatNoise.bmp and baboonNoise.bmp along with your report. Compare the results obtained using this method to the results obtained in Scheme 1. Which algorithm is better and why?

VIDEO FILTERING

A video can be considered as a group of frames. Many of the video filtering algorithms are based on repeated application of the basic filtering algorithms on different frames of video. In this case study, we will however consider two kinds of video filtering methods. The first one is straightforward and is a simple extension of the image filtering schemes (Scheme-1 and Scheme-2) discussed in the previous sections.

Problem 8. In this problem, you will implement the first video filtering scheme in MATLAB. Use the provided program yuvRead.m to read the video from the input video stream carphoneNoise.yuv. Apply the algorithm you developed in Scheme 1 and Scheme 2 to filter the video (on a frame by frame basis). Observe the final filtered video. Which scheme gives better results. You will have to submit a the filtered video file along with your report. You can use the yuvWrite.m program to write the final filtered video. You can play the video using the playMovie.m program also given on the website.

Though, this filtering scheme would work, it is very time consuming. A typical 1-second video samples at around has around 16 frames (images) and therefore, you will have to run the program around 16 times to obtain the filtered version of a 1-second file. Moreover, exact filtering is not necessary for each video frame as you will hardly be able to observe any difference when you finally playback the video. So, instead of filtering every frame of the video, we could use motion compensation based techniques for video filtering.

Motion Estimation based SVD Video Filtering

Motion estimation algorithms are based on the principle that most parts of the subsequent video frames are almost similar. Very few parts of the video (like moving objects/persons) change. So one can estimate the motion of the objects based on information contained in subsequent frames and use the resultant motion vectors to represent the video.

We can remove the temporal redundancy using motion estimation (ME) and motion compensation (MC). The basic idea of ME/MC is that the content within a video shot is almost the same, except the objects motion and the global camera motion. We can find that some regions in the current frame have the corresponding ones in the previous and future frames. Therefore, we do not need to redo the computations. Instead, we can simply use motion vectors to describe the motion between these two frames. Motion estimation is to find out the motion vectors, and motion compensation is to construct the estimated current frame using a reference frame. As a sample illustration, the subsequent frames of the car image are shown in Fig. 2. We see from the images that they are almost similar. The background remains essentially the same but the car has moved slightly forward. The estimated motion vectors

Figure 2: Present and Next frames of the car image

are shown in Fig. 3.

There are 2 essential methods to do motion estimation. In both methods, you try to estimate the present frame in terms of the past frames. You can do an exhaustive search or a more sophisticated three step process. While the exhaustive search gives the best results, it is very slow. The three-step process is fast but not optimal. In this project, we will use the three-step process. You are given an implementation of the three-step motion estimation process on the website (motion3wayEstimate.m). For further details of the algorithm, please refer the book in the references.

In this case study, we will study one slight modification of the algorithm for video filtering. Using the redundancy of the subsequent frames and motion estimation algorithms, we first find the motion vectors. This gives us a good mapping of the blocks in the present frame in terms of the blocks in the previous frames. We will define two kinds of frames, "reference frames" and "estimated frames". The estimated frames correspond to the ones which have been estimated using the reference frames and motion estimation algorithms. Note that there could be more than one frame that is estimated based on one reference frame (In this case study, we will treat the number of estimated frames per reference frame as a variable call it N_{er}).

Our filtering algorithm is as follows. First, given the video, we divide it into reference frames and estimated frames (done sequentially). For all the reference frames, we apply the truncated SVD algorithm and obtain its filtered version as we did for problem 8. We then consider a $N \times N$ block in the reference frame and do a Singular Value decomposition on it. We store the SVD coefficients.

Now, for the estimated frames, we use the three-step motion estimation algorithm to find the motion vectors. The $(i, j)^{th}$ of the present frame is expressed in terms of $(u_{ij}, v_{ij})^{th}$ block of the reference frame (u_{ij} and v_{ij} obtained using motion estimation algorithms). We now use the stored SVD coefficients of the (u_{ij}, v_{ij}) block of the reference frame. We approximate the (i, j) block of the

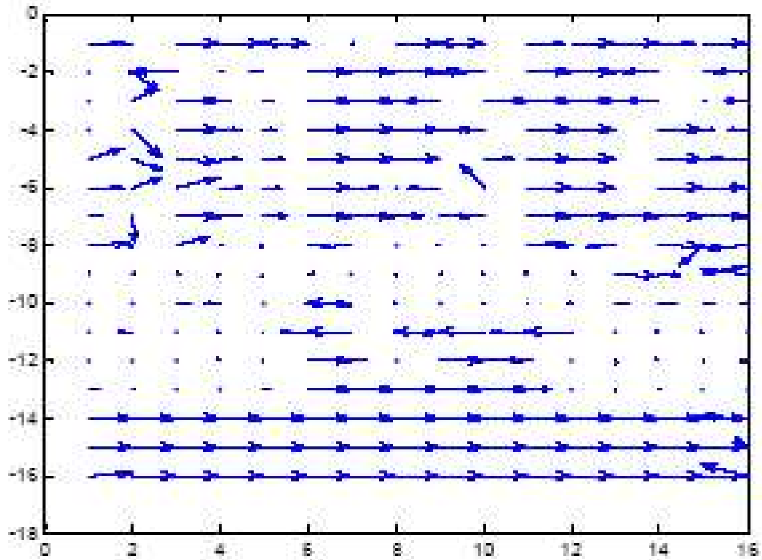

Figure 3: The estimated motion vectors

current frame by projecting this block on these basis vectors (obtained by SVD of the (u_{ij}, v_{ij}) block of the reference frame) and finding the best estimate of the present block in terms of the first K basis vectors of the corresponding block of the reference frame.

Problem 9. Now, implement the Video Filtering based on Motion Estimation algorithm described in this section. What are the trade-offs between this scheme and the previous scheme for video filtering. Explain (You may use N_{er} value of 3 or 4 for your results)

In this case study, we have explored a set of techniques for image and video filtering based on the Singular Value decomposition.

References

1. D. Muti, S. Bourennane, M. Guillaume, SVD-Based Image filtering Improvement by means of image rotations, IEEE conference on the Acoustic, Speech and Signal Processing (ICASSP), 2004.

2. J. Q. Zhang, S. J. Ovaska, ADC Characterization based on Singular Value Decomposition, IEEE Transactions on Instrumentation and Measurements, Feb 2002.

3. H. J. Andrews, C. L. Patterson, Singular Value Decompositions and Digital Image Processing, IEEE Transactions on Acoustic, Speech and Signal Processing, Feb 1976.

4. K. Konstantinides, G. S. Yovanof, Application of SVD Based Spatial Filtering to Video Sequences, IEEE conference on the Acoustic, Speech and Signal Processing (ICASSP), 1995.

5. Some parts of the description on Motion Estimation/Motion Compensation algorithms was adapted from ENEE631 (Digital Image Processing) Class lecture notes by Prof. Min Wu.

6. Details of the Motion Estimation/ Motion Compensation algorithm can also be found in the book - Video Processing by Wang.