Image Compression using Singular Value Decomposition

Adam Abrahamsen and David Richards

December 14, 2001

Abstract

Do you store images on your computer? Do you need more room on your hard drive? Well, you could upgrade your hard drive. But, we would like to propose an alternative solution. Computer technology these days is most focused on storage space and speed. One way to help cure this problem is Singular Value Decomposition. We have put together this easy to read explanation of how to use Singular Value Decomposition to reduce the space required to store images. We assume that the reader has a basic knowledge of eigenvalues, eigenvectors, and matrix operations.

1. Introduction to SVD

Singular Value Decomposition (SVD) is said to be a significant topic in linear algebra by many renowned mathematicians. SVD has many practical and theoretical values, other than image compression. One special feature of SVD is that it can be performed on any real (m,n) matrix. It factors A into three matrices U, S, V, such that, $A = USV^T$. Where U and V are orthogonal matrices and S is a diagonal matrix.

2. Mathematics

We have stated that the purpose of (SVD) is to factor matrix A into USV^T. The matrix U contains the left singular vectors, the matrix V contains the right singular vectors, and the diagonal matrix S contains the singular values. Where the singular values are arranged on the main diagonal in such an order

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0,$$

where r is the rank of matrix A, and where $(_p)$ is the smaller of the dimensions m or n.

2.1. Arbitrary Example

We begin the process of Singular Value Decomposition by selecting the matrix A which has m rows and n columns. Now, we need to factor A into three matrices U, S, V^T.

First we will find V. If you multiply both sides of the equation $A = USV^T$ by A^T we get

$$A^TA = (USV^T)^T(USV^T) = VS^TU^TUSV^T.$$

Since $U^TU = I$ this gives

$$A^TA = VS^2V^T$$

Now we need to diagonalize A^TA. If you will notice, this is very similar to the diagonalization of matrix A into $A = Q\Lambda Q^T$. Except our symmetric matrix is not A, it is A^TA. To find V and S we need to find the eigenvalues and eigenvectors of A^TA. The eigenvalues are the square of the elements of S (the singular values), and the eigenvectors are the columns of V (the right singular vectors).

Eliminating V from the equation is very similar to eliminating U. Instead of multiplying on the left by A^T we will multiply on the right by A^T. This gives:

$$AA^T = (USV^T)(USV^T)^T = USV^TVS^TU^T.$$

Since $V^TV = I$, this gives

$$AA^T = US^2U^T$$

Again we will find the eigenvectors, but this time for AA^T. These are the columns of U (the left singular vectors).

Since A is $m \times n$, S is $m \times n$ and

$$A^TA$$

produces an $n \times n$ matrix, and:

$$AA^T$$

produces an $m \times m$ matrix,

$$A = \begin{pmatrix} u_1 & \cdots & u_r & \cdots & u_m \end{pmatrix} \begin{pmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & \ddots & \\ & & & & 0 \end{pmatrix} \begin{pmatrix} v_1^T \\ \vdots \\ v_r^T \\ \vdots \\ v_n^T \end{pmatrix}$$

Where U is $m \times m$, S is $m \times n$, V is $n \times n$.

2.2. 2×2 Example

Let :

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

$$A^T A = \begin{pmatrix} 2 & 1 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} 2 & -2 \\ 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 5 & -3 \\ -3 & 5 \end{pmatrix}$$

Subtracting $\lambda(I)$ from $A^T A$

$$\left| \begin{pmatrix} 5 & -3 \\ -3 & 5 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right| = 0$$

Therefore,

$$\begin{vmatrix} 5-\lambda & -3 \\ -3 & 5-\lambda \end{vmatrix} = 0$$

Now find λ,

$$(5-\lambda)(5-\lambda) - 9 = 0$$

$$\Rightarrow 25 - 10\lambda + \lambda^2 - 9 = 0$$

$$\Rightarrow \lambda^2 - 10\lambda + 16 = 0$$

$$\Rightarrow (\lambda - 8)(\lambda - 2) = 0$$

Therefore our eigenvalues are 8 and 2. We construct the matrix S^2 by placing the eigenvalues along the main diagonal in decreasing order.

$$S^2 = \begin{pmatrix} 8 & 0 \\ 0 & 2 \end{pmatrix}$$

Therefore, taking the square root of matrix S^2 gives,

$$S = \begin{pmatrix} 2\sqrt{2} & 0 \\ 0 & \sqrt{2} \end{pmatrix}$$

Now we need to find the eigenvectors of A^TA which are the columns of V.
First we'll show where $\lambda = 8$,

$$\left[\begin{pmatrix} 5 & -3 \\ -3 & 5 \end{pmatrix} - \begin{pmatrix} 8 & 0 \\ 0 & 8 \end{pmatrix} \right] \hat{v_1} = 0$$

$$\Rightarrow \begin{pmatrix} -3 & -3 \\ -3 & -3 \end{pmatrix} \hat{v_1} = 0$$

$$\Rightarrow \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \hat{v_1} = 0$$

Therefore,

$$\hat{v_1} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Since V has an orthonormal basis, v_1 needs to be of length one. We divide v_1 by it's magnitude to accomplish this. Thus,

$$\hat{v_1} = \begin{pmatrix} \frac{-\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

Similarly, we'll show where $\lambda = 2$

$$\left[\begin{pmatrix} 5 & -3 \\ -3 & 5 \end{pmatrix} - \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \right] \hat{v_2} = 0$$

$$\Rightarrow \begin{pmatrix} 3 & -3 \\ -3 & 3 \end{pmatrix} \hat{v_2} = 0$$

$$\Rightarrow \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix} \hat{v_2} = 0$$

Therefore,

$$\hat{v_2} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Similarly dividing by the magnitude to create the orthonormal basis gives,

$$\hat{v_2} = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

Now we need to construct the augmented orthogonal matrix V,

$$V = \begin{pmatrix} v_1 & v_2 \end{pmatrix}$$

and,

$$V = \begin{pmatrix} \frac{-\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}$$

Now we need to find the eigenvectors for AA^T. Since the eigenvalues for AA^T are the same as the eigenvalues for A^TA. We can go straight to finding the eigenvectors using the eigenvalues previously found.

First we'll show where $\lambda = 8$,

$$\left[\begin{pmatrix} 8 & 0 \\ 0 & 2 \end{pmatrix} - \begin{pmatrix} 8 & 0 \\ 0 & 8 \end{pmatrix} \right] \hat{u_1} = 0$$

$$\Rightarrow \begin{pmatrix} 0 & 0 \\ 0 & -6 \end{pmatrix} \hat{u_1} = 0$$

$$\Rightarrow \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \hat{u_1} = 0$$

Therefore,

$$\hat{u_1} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

Since $\hat{u_1}$ is already of length one, dividing by the magnitude we get the same vector back.

Similarly, we'll show where $\lambda = 2$

$$\left[\begin{pmatrix} 8 & 0 \\ 0 & 2 \end{pmatrix} - \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right] \hat{u_2} = 0$$

$$\Rightarrow \begin{pmatrix} 6 & 0 \\ 0 & 0 \end{pmatrix} \hat{u_2} = 0$$

$$\Rightarrow \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \hat{u_2} = 0$$

Therefore,

$$\hat{u_2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Setting up the augmented matrix U

$$U = \begin{pmatrix} u_1 & u_2 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

Therefore

$$A = USV^T$$

$$A = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2\sqrt{2} & 0 \\ 0 & \sqrt{2} \end{pmatrix} \begin{pmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}$$

3. Image Processing with SVD

What is the purpose of transforming the matrix A into USV^T? We want to approximate the $m \times n$ matrix A by using far fewer entries than in the original matrix. By using the rank of a matrix we remove the redundant information (the dependant entries) when $r < m$, or $r < n$.

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T + 0 u_{r+1} v_{r+1}^T + \cdots$$

Since the singular values are always greater than zero. Adding on the dependant terms where the singular values are equal to zero does not effect the image. The terms at the end of the equation zero out leaving us with:

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T$$

We can further approximate the matrix by leaving off more singular terms of the matrix A. Since the singular values are arranged in decreasing order, the last terms will have the least affect on the overall image. Doing this reduces the amount of space required to store the image on a computer.

4. Using SVD in Matlab

Matlab provides us with the ability to perform (SVD) on larger matrices. For an example we will use a 7×7 matrix with rank 5. Using the command:

```
A=randint(7,7,25,5)+25
```

provides us with a 7×7 matrix (A) with values between 0 and 50, and with a rank of 5.

$$A = \begin{pmatrix} 24 & 36 & 31 & 31 & 39 & 44 & 39 \\ 27 & 39 & 35 & 33 & 33 & 35 & 33 \\ 29 & 37 & 35 & 43 & 35 & 33 & 35 \\ 32 & 29 & 43 & 36 & 27 & 35 & 27 \\ 39 & 35 & 30 & 31 & 25 & 15 & 25 \\ 29 & 44 & 26 & 47 & 40 & 26 & 40 \\ 34 & 21 & 37 & 22 & 27 & 39 & 27 \end{pmatrix}$$

Now we can use the power of matlab again to perform (SVD) on the matrix A. Using the commands:

`[U,S,V]=svd(A)` factors A into USV^T.

After factoring A, using the command

```
svdimage(A,U,S,V,1,gray)
```

gives us the opportunity to view the matrix A as an image, and see each individual iteration of A using the rank approximation. A little explanation of the entries of this function require some explanation. The A,U,S, and V are self explanatory, but the 1 and the gray haven't been explained yet. The "1" starts

the program off with the first iteration and the"gray" uses the colormap that matlab defines as gray. Now each number of the matrix corresponds to the color that matlab has defined as the colormap(gray).

For the pictures in this presentation we used a program that we wrote with the help of David Arnold. We had to write this program in order to fit these images into the document. Using the commands,

```
close all
[A,map]=imread('lena.gif');
B=im2double(A,'indexed');
imshow(B,map)
[u,s,v]=svd(B);
 C=zeros(size(B));
  for j=1:k
   C=C+s(j,j)*u(:,j)*v(:,j).';
  end
C=floor(C);
imshow(C,map)
k=find(C<1);
C(k)=1;
set(gcf,'Unit','inches','Paperposition',[0,0,2,1])
print -djpeg 'lenak.jpg'
```

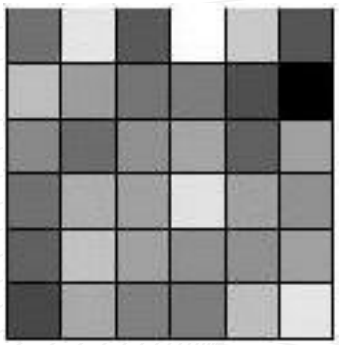
By changing the k values in the forloop we were able to construct the different iterations. You may notice the similarities between these two equations:

```
C=C+s(j,j)*u(:,j)*v(:,j).'
```

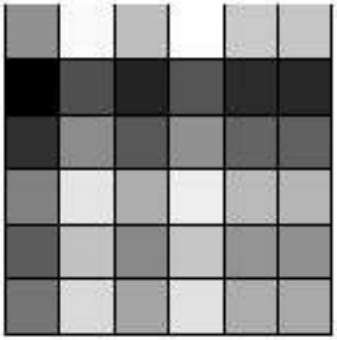
and

$$A = \sigma_1 u_1 v_1^T + r\sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T$$

Here are the images

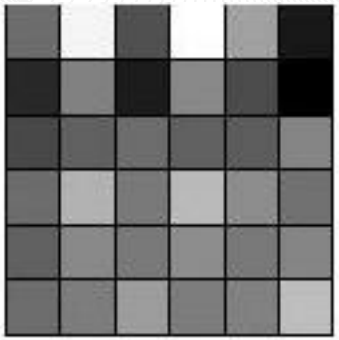


(a) Original Matrix

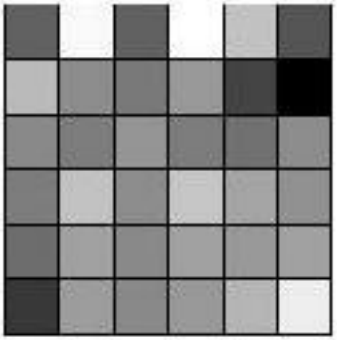


(b) 1 Iteration

As you can see here the first approximation is not very good. But we get the general idea of how it is supposed to look.

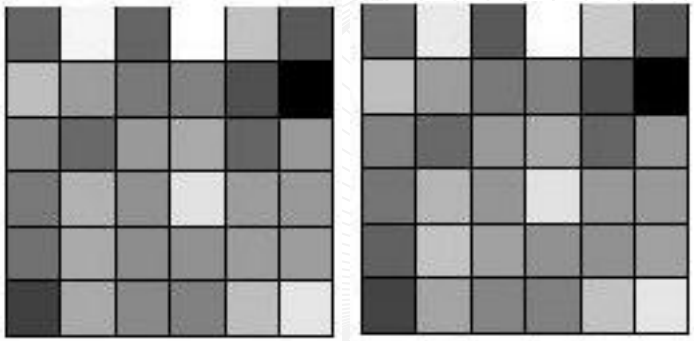


(c) 2 Iterations



(d) 3 Iterations

Now by the 3^{rd} iteration we are starting to see a very good approximation, the key thing here is to remember that the rank is 5.



(e) 4 Iterations

(f) 5 Iterations

As you can see the fourth iteration gives a near perfect image and the fifth iteration is the exact image, because of the rank of five we see the exact image after five iterations.

Now we will show the use of (SVD) on Lena which is one of the standardized images.

(g) Original Image

(h) 5 Iterations

(i) 10 Iterations

The first five iterations actually give the shape of the image, which is a decent approximation. This takes up 99.9% less storage space than the original image.

(j) 20 Iterations

(k) 60 Iterations

By the first sixty iterations we get a good approximation, we can identify the person with a substantial degree of detail. This image requires 84% less storage space than the original image. This is good.

(l) 100 Iterations

Finally, the first one-hundred iterations give a near perfect image, and yet requires 55% less storage space. This is very good!

5. Conclusion

Using (SVD) for image compression can be a very useful tool to save storage space. We were able to get an image that is indistinguishable from the original image, but only using 45% of the original storage space. The Singular Value Decomposition is not only used for image compression it has **many** other useful applications.

References

[1] Arnold, David. His matlab and LaTeX expertise.

[2] Kalman, Dan. **"A Singularly Valuable Decomposition." The College Mahtematics Journal. Vol.27_No.1_Jan 1998, 2-23.**

[3] Lay, David C.**Linear Algebra and Its Aplications.** Addison Wesley Longman, Inc., 1997

[4] Strang, Gilbert. **Introduction to Linear Algebra.** Wellesley-Cambridge Press, 1998.

[5] The Sauceman. His LaTeX expertise.