



Application of singular value decomposition in image processing

Rajesh A. Jadav and Shailesh S. Patel

*Dept. of Mathematics, S.V.M. Institute of Technology, College campus, Old N.H.-8, Bharuch-392001, Gujarat, India
jdv_rajesh@yahoo.co.in, chintshail@yahoo.com*

Abstract

The purpose of this paper is to study an important application of Singular Value Decomposition (SVD) to image processing. The idea is that by using the smaller number of vectors, one can reconstruct an image that is closer to the original. The clarity of the image depends on how many singular values are used to reconstruct it. In this paper, SVD was applied to the image and also using the Matlab software we developed the code. We also demonstrated how the SVD is used to minimize the size needed to store an image.

Keywords: Singular value decomposition, image compression, image processing.

Introduction

The singular value decomposition (SVD), one of the most useful tools of linear algebra, is a factorization and approximation technique which effectively reduces any matrix into a smaller invertible and square matrix. One special features of SVD is that it can be performed on any real $m \times n$ matrix. It factors A into three matrices U, S, V , such that $A = USV^T$, (Ogden & Huff, 1997; Jody & Lynn 1998) where, U and V are orthogonal matrices and S is a diagonal matrix.

Singular value decomposition

Given a $m \times n$ matrix A , there exists decomposition (Ogden & Huff, 1997), such that $A = USV^T$, where U and V are orthogonal matrices and S is a diagonal matrix with nonnegative diagonal entries in decreasing order. The diagonal entries of S are the positive square roots of the eigen values of AA^T and are called the singular values of A . The decomposition $A = USV^T$ is called a singular value decomposition of A , and A can be written as,

$$A = s_1 u_1 v_1^T + s_2 u_2 v_2^T + s_3 u_3 v_3^T + \dots + s_n u_n v_n^T ,$$

where u_i and v_i are the i^{th} and j^{th} columns of the matrices U and V respectively, and $s_1 = s_2 = s_3 = \dots = s_n$ (Jain & Gunawardena, 2003).

Each of the nonzero terms in this representation of A is a rank-1 matrix, and if A is approximated by the sum of the first k terms, then this sum is called a rank- k approximation of A .

Theorem (SVD)

Let A be a $m \times n$ matrix, and $s_1, s_2, s_3, \dots, s_n$ be the singular values of A then $A = USV^T$ where, U and V are

orthogonal matrices and S is a diagonal entries $s_1, s_2, s_3, \dots, s_n$. (Equivalently,

$$A = s_1 u_1 v_1^T + s_2 u_2 v_2^T + s_3 u_3 v_3^T + \dots + s_n u_n v_n^T ,$$

where u_i and v_i are the i^{th} and j^{th} columns of the matrices U and V , resp., and $s_1 \geq s_2 \geq s_3 \geq \dots \geq s_n$, (Jain & Gunawardena, 2003).

Proof: Let $v_1, v_2, v_3, \dots, v_n$ be an orthonormal set of eigenvectors of AA^T corresponding to the eigenvectors $s_1^2, s_2^2, s_3^2, \dots, s_n^2$ written in decreasing order, where, possibly $s_{r+1} = \dots = s_n = 0$.

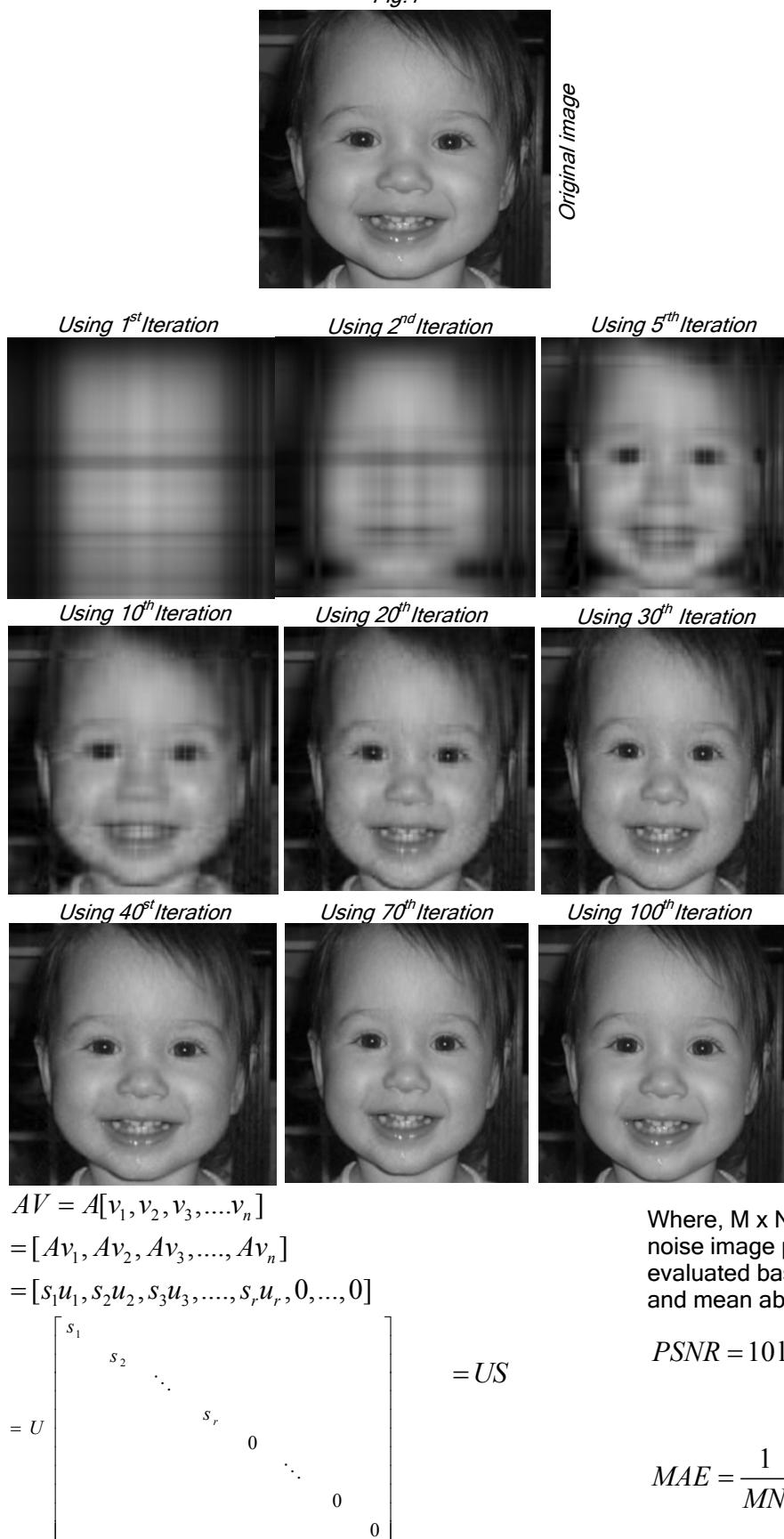
Let $u_i = \frac{1}{s_i} Av_i$, for $i=1, 2 \dots r$.

Now

$$u_i^T u_j = \frac{1}{s_i s_j} v_i^T A^T A v_j = \frac{1}{s_i s_j} v_i^T s_j^2 v_j = \frac{1}{s_i s_j} s_j^2 v_i^T v_j = 0 ,$$

$$u_i^T u_i = \frac{1}{s_i^2} v_i^T A^T A v_i = \frac{1}{s_i^2} v_i^T s_i^2 v_i = v_i^T v_i = 1$$

Thus, $u_1, u_2, u_3, \dots, u_r$, form an orthonormal set of vectors. If $r \leq m$, we extend the set $\{u_1, u_2, u_3, \dots, u_r\}$ to orthonormal set $\{u_1, u_2, u_3, \dots, u_r, \dots, u_m\}$ of vectors so as to form a basis of R^m . We will prove that $AV = US$, where S is the $n \times n$ diagonal matrix whose diagonal entries are $s_1, s_2, s_3, \dots, s_r, 0, \dots, 0$ and U and V are $m \times m$ and $n \times n$ orthogonal matrices, resp., given by $V = [v_1, v_2, v_3, \dots, v_n]$ and $U = [u_1, u_2, u_3, \dots, u_r, \dots, u_m]$.



Because, $v^T = v^{-1}$. Thus we obtain $A = USV^T$, as desired.

Experiments and results in Matlab

Matlab provides us with the ability to perform (SVD) on larger matrices. So Matlab is very helpful to do work with any size of matrices and its gives the result as fast as possible.

In this paper we give the example to show the effect of SVD (Jain, 2004), in Matlab using some basic operations.

For an example, Let I be the colour image of large size of matrices, so first we convert into gray scale image of size 497×498 matrix, and then we apply the singular value decomposition to the given image and try to reduce the size of the image i.e. to compress the size of given image. Hence we applied the Matlab software and developed the code. Fig. 1 shows the outcome of it.

As we see the 10th Iteration the image contain the 100 entries, also form the 30th Iteration we get the image near to original image, and form the 70th Iteration i.e. A 70×70 matrix, with 4900 entries is significantly reduced the original image of size 497×498 matrix, with 247506 entries. So, there is no need to go up to 100th Iteration. And image size is also compressed.

Error approximation

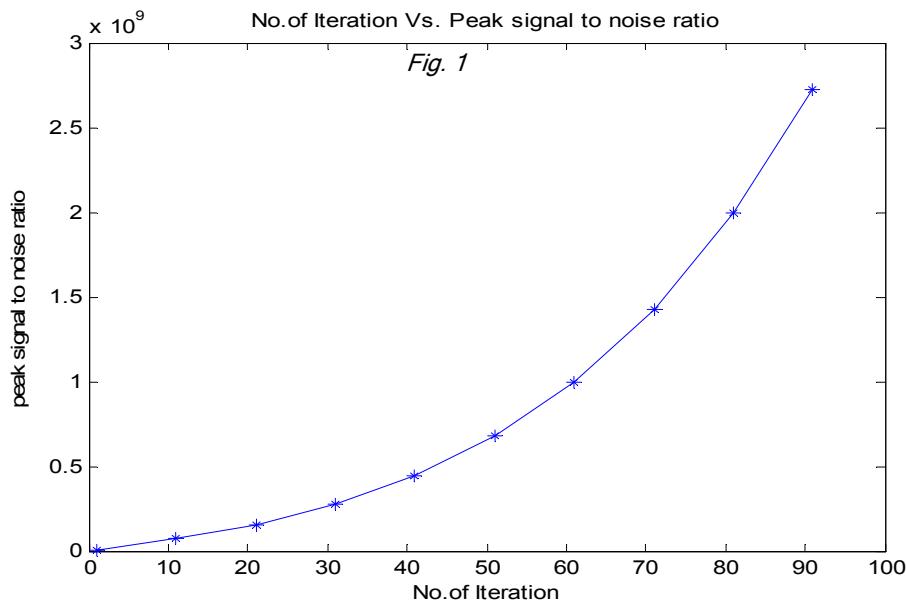
We also focus on Error in the output image; therefore we take the mean square error (Bakwad *et al.*, 2009), between the original image and noisy image that is compressed image.

$$\text{Error} = \frac{1}{MN} \left[\sum_{i=1}^M \sum_{j=1}^N z(i, j) - p(i, j) \right]^2$$

Where, $M \times N$ is the size of the original image $z(i, j)$ and noise image $p(i, j)$. The performance of this technique is evaluated based on peak signal to noise ratio (PSNR) and mean absolute error is given by

$$\text{PSNR} = 10 \log_{10} \frac{R^2}{\frac{1}{MN} \sum_{i,j=1}^{M,N} (f_{i,j}^{\wedge} - f_{i,j})^2}$$

$$\text{MAE} = \frac{1}{MN} \sum_{i,j=1}^{M,N} (f_{i,j}^{\wedge} - f_{i,j})^2$$



Where, $\hat{f}_{i,j}$ and $f_{i,j}$ denote the pixel values of the restored image i.e. compressed image and original image respectively.

The PSNR (Bakwad *et al.*, 2009), computes the peak signal-to-noise ratio between two images. This ratio is often used as a quality measurement between the original and a compressed image. The higher the PSNR the better the quality of the compressed image.

The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are the two error metrics used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower the error.

Conclusion

When we applied the image processing, a matrix can be compressed to a significantly smaller size that can be very useful tool to save storage space. We were able to get an image that is indistinguishable from the original image. We can also estimate the error caused due to the image compression. The clarity of the resulting images corresponding to the original image after the iterations. From the Fig. 2 one can conclude that as the no. of iteration increases, the image closer to original was obtained: besides the size of image also decreased. Fig. 3 reveals that as the number of iteration increases, the error is also reduced. It helps the image compression and face recognition.

References

1. Bakwad KM, Pattnaik SS, Sohil BS, Panigrahi BK, Sastry VRS and Gollapudi (2009) Bacterial foraging optimization technique cascaded with adaptive filter to enhance peak signal to noise ratio from signal image. *IETE J. Res.* 55(4), 173-179.
2. Jain AK (2004) Fundamental of digital image processing. Pearson Edu. India. p: 176-180.
3. Jain SK and Gunawardena AD (2003) Linear Algebra: An Interactive Approach. Thomson Asia Pvt. Ltd. pp: 216-220.
4. Jody S. Hourigan and Lynn V. McIndoo (1998) Singular value decomposition. *Lin. Algebra-Maths-45*, College of Redwoods.
5. Ogden CJ and Huff T (1997) The singular value decomposition and its application in image processing. *Lin. Algebra-Maths-45*, College of Redwoods.

