Pattern Recognition using Singular Value Decomposition (SVD)

P Pankaj, Brijesh Kumar Markandey and Arun Kumar

October 26, 2020

Pattern Recognition using Singular Value Decomposition (SVD)

Pankaj
(MIT2020114)

Brijesh Kumar Markandey
(MIT2020115)

Arun Kumar
(MIT2020116)

M.Tech Information Technology
Indian Institute of Information Technology Allahabad

***Abstract*—This paper contains details about pattern recognition and singular value decomposition. We are explaining and analyzing the applications of singular value decomposition. It was found that the singular value decomposition is widely used in computer science fields like Machine learning and in transforming some curve. singular value decomposition is used in matrix factorization and recommendation systems.**

***Index Terms*—Matrix factorization, Recommendation System, Pattern recognition, singular value decomposition(SVD)**

I. Introduction

The matrix AA^T and A^TA are very special in linear algebra. Consider any m × n matrix A, we can multiply it with A to form AA^T and A^TA separately. Let's introduce some terms that frequently used in SVD. We name the eigenvectors for AA^T as u_i and A^TA as v_i here and call these sets of eigenvectors u and v the singular vectors of A. Both matrices have the same positive eigenvalues. The square roots of these eigenvalues are called singular values.

In linear algebra Singular value decomposition is used to transform the picture that is we can rotate and stretch using SVD. Consider a matrix A_{mXn}, SVD this matrix into two unitary matrices which are orthogonal matrices and ractangular diagonal of singular values.

$$A_{mXn} = U_{mXm}\Sigma_{mXn}V_{nXn}$$

SVD is used to transformation that is we can rotate and stretch the image. In the given equation matrix U and matrix V represents rotation of matrix while Σ represents stretching.

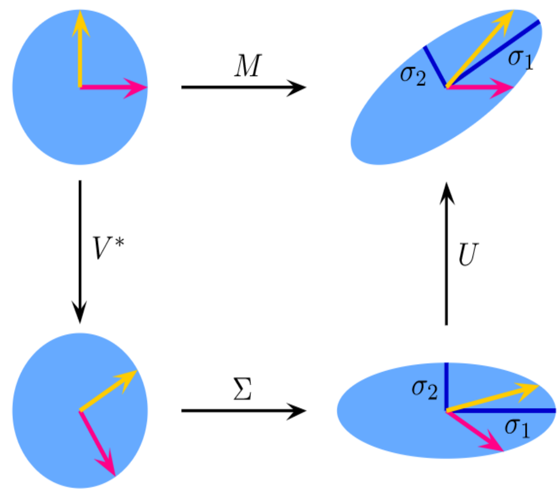
$$A^TA = (V\Sigma^TU^T)U\Sigma V^T$$

$$\mathbf{A^TA = V\Sigma^T\Sigma V^T}$$

$$AA^T = (U\Sigma V^T)(V\Sigma^T U^T)$$

$$\mathbf{AA^T = U\Sigma\Sigma^T U^T}$$

In the represented figure a circle is given, there are two vectors given that are yellow(Y) and red(R). When we are applying V on this circle, the circle is rotated after that we are applying Σ the circle is stretched horizontally and the circle becomes ellipse and at last we are applying U and this ellipse is again rotated and the resultant is an ellipse which is rotated.



$$M = U\cdot\Sigma\cdot V^*$$

Fig. 1.

To understand in better way let's consider an example -

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

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

We konw that -

$$U = AA^T$$

$$U = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}$$

Using characteristic equation -

$$(A-\lambda I) = 0$$

After solving the characteristic equation -

$$\lambda = 12,\ 10$$

Putting the values of λ -

$$U = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Applying Gram-Schimdt process -

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

We can calculate V also using similar process -

$$V = \begin{bmatrix} \frac{1}{\sqrt{16}} & \frac{2}{\sqrt{16}} & \frac{1}{\sqrt{16}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix}$$

Pattern Recognition is the act of taking a raw data and making an action based on the category of the pattern. Pattern Recognition can be defined as the classification of the data on the basis of the knowledge gained or on the basis of statistical information extracted from patterns and their representations.

Fig. 2.

Following are the application of Pattern recognition -
a. Biomedical Image Processing.
b. Optical Character Recognition.
c. Multimedia Document Recognition. etc etc. . . .
In layman's point of view, everything from your Netflix recommendations to your favorite shopping websites uses PR as a core concept. We will get a hang of all these things after we cover the general idealogy of PR.

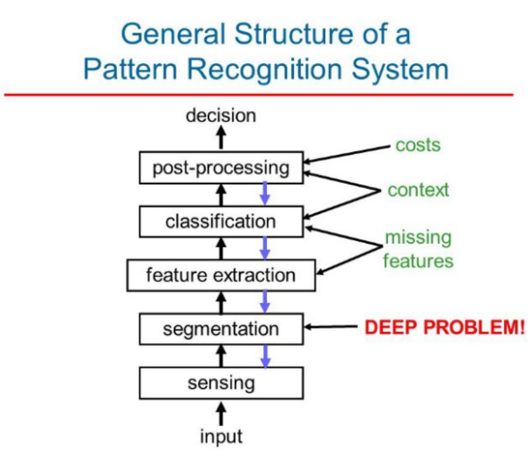


Fig. 3.

Explanation of Block Diagram -
1.First of all sensor senses the image or sound and convert it into signal data.
2.Segmentation is the part which enables segregating sensed object from the background noise.
3.Feature Extraction phase extracts the properties on which system will make patterns.
4.Next phase classifies the data into catagories.
5.Finally, the post-processor can take account of other considerations such as the cost of the error to decide appropriate action.

II. Proposed Model

In this report we are going to discuss about the Singular Value Decomposition (SVD). SVD method can transform a matrix A into product USV^T , which allows us to refactoring a digital image in to three matrices. The using of singular values of such refactoring allows us to represent the image with a smaller set of values, which can preserve useful features of the original image, but use less storage space in the memory, and achieve the image compression process.

Singular Value Decomposition (SVD) is said to be the important or best topic in linear algebra by many renowned mathematicians in all over the world. SVD has many practical and theoretical values special feature of SVD is that it can be performed on any real (m, n) matrix.

Let's say we have a matrix A with m rows and n columns, with rank r and r $\leq$ n $\leq$ m. Then the A can be factorized into three matrices:

$$A = USV^T$$

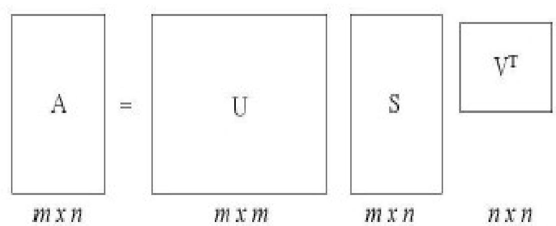


Fig. 4. Illustration of Factoring A to USV^T

Where Matrix U is an m × m orthogonal matrix,

$$U = [u_1, u_2, u_3, ...u_r, r_r + 1, ..., u_m]$$

column vectors u_i, for i = 1, 2, . . . , m, form an orthonormal set:

$$u_i^t u_j = \delta = \begin{cases} 1, ..., i = j 0, ...i \neq \text{j} \end{cases}$$

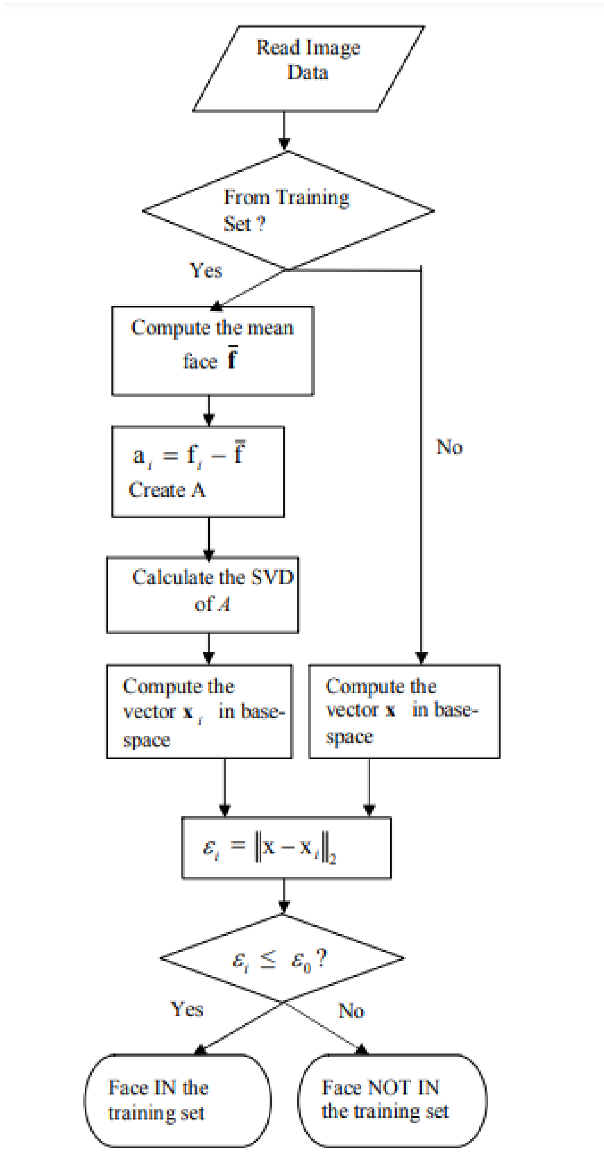


Fig. 5. **Flow chart of Face Recognition with SVD**

The above flow chart is used in face recognition.
First all raw data S of N faces is taken. Using this raw data we need to find the mean face $\overline{f}$ using S. Forms a matrix A with the computed $\overline{f}$. Calculate SVD of matrix A. For each known individual, compute the coordinate vector x_i. Choose a threshold ϵ_1 that defines the maximum allowable distance from face space. Determine a threshold ϵ_0 that defines the maximum allowable distance from any known face in the training set S. Now if ϵ_1 is less then or equal to ϵ_0 then Face is present in training set otherwise it is not present in training set.

III. Application

A. Dimensionality Reduction

The first and most important application is to reduce the dimensionality of data, the SVD is more or less standard for this, PCA is exactly the same as the SVD. You may want to reduce the dimensionality of your data because:
1. You want to visualize your data in 2d or 3d.
2. The algorithm you are going to use works better in the new dimensional space
3. Performance reasons, your algorithm is faster if you reduce dimensions.
In many machine learning problems using the SVD before a ML algorithm helps so it's always worth a try.

B. Multi-Dimensional Scaling

MDS is a dimensionality reduction technique similar to SVD but not confined to linear mappings; many manifold learning techniques are variations or different computational approaches to the same idea and method as MDS. (By the way, MDS is pretty good for data visualization after analysis.)

C. Pseudo-Inverse

Now, from SVD, we can clearly see that (not putting in the derivation), the Pseudoinverse looks like this -

$$A^+ = (A^T A)^{-1} A^T$$

And, let us also remember the basic rules of Matrices -
1. $(AB)^{-1} = B^{-1}A^{-1}$
2. $A^{-1}.A = I$
3. $A.I = A$
Now, let us assume A is square and non-singular. Then -

$$A^+ = (A^T A)^{-1} A^T$$
$$A^+ = A^{-1}(A^T)^{-1} \text{ (form1)}$$
$$A^+ = A^{-1}.I \text{ (form2)}$$
$$A^+ = A^{-1} \text{ (form3)}$$

D. Face Recognition

The SVD can be used to compress images, but there are some better algorithms of course.
Face recognition is one of the standard applications of PCA. Consider using the ORL face database -
1.Composed of 400 images with dimensions 112 x 92.
2.There are 40 persons, 10 images per each person.
3.The images were taken at different times, lighting and facial expressions.
4. The faces are in an upright position in frontal view, with a slight left-right rotation.

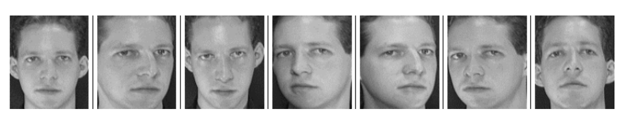


Fig. 6. **The ORL database, we can use k-fold cross validation with k=10**

1.We randomly divide the 400 images into 10 sets such that each contains 40 different persons.

2.For each iteration, 9 sets are used for training while the remaining set is reserved for testing.
3.There will be 10 recognition results and we can check for their consistency and compute the average recognition accuracy.

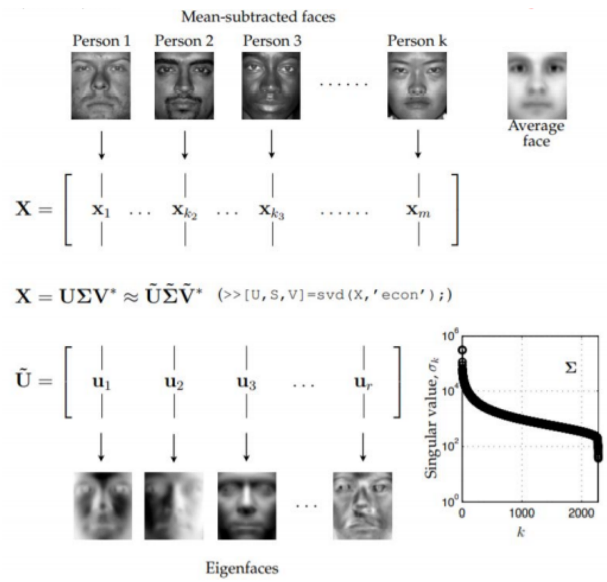


Fig. 7.

Final Result -

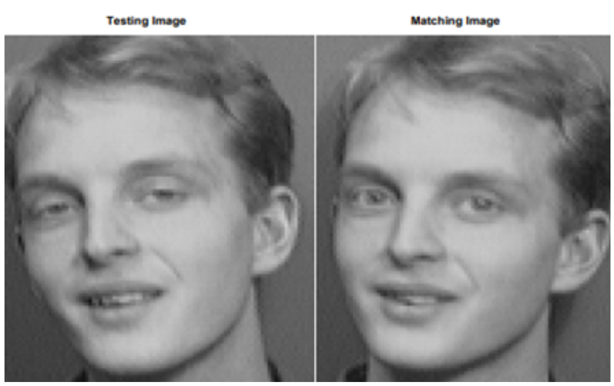


Fig. 8.

E. Image Compression

First decompose the image using SVD as usual into U, V and D matrices and then keep only k vectors.
The following Matlab code does this:
1.Percentage compute the SVD of I
2.[U,D,V] = SVD(I);
3.Percentage compute the k rank approximation of I
4.U_k = U(:,1:k);
5.V_k = V(:,1:k);
6.Σ_k = Σ(1:k,1:k);
7.Percentage of the compressed image
8.$I_k = U_k * \Sigma_k * (V_k)'$;

Conclusion

Thus it is observed that SVD gives good pattern recognition results with less computational complexity compared to other recognition techniques. A certain degree of compression as required by an application can be achieved by choosing an appropriate value of k (i.e. the number of eigen values). In other words, degree of compression can be varied by varying the value of k. However to achieve high value of compression ratio image quality is to be sacrificed. Therefore it is required to select proper value of k to choose between compression ratio and image quality. Once the value of k is selected for specific application or for specific video the same benchmark can be used for all the frames.

References

[1] Singular value decomposition: https://en.wikipedia.org/wiki/Singularvaluedecomposition
[2] Singular Value Decomposition Applied To Digital Image Processing : Lijie Cao
[3] Machine Learning — Singular Value Decomposition (SVD) Principal Component Analysis (PCA) : Jonathan Hui https://medium.com/@jonathanhui/machine-learning-singular-value-decomposition-svd-principal-component-analysis-pca-1d45e885e491
[4] Andrew Ng ML Course
[5] Hong X., Xu Y., Zhao G. (2017) LBP-TOP: A Tensor Unfolding Revisit. In: Chen CS., Lu J., Ma KK. (eds) Computer Vision – ACCV 2016 Workshops. ACCV 2016. Lecture Notes in Computer Science, vol 10116. Springer, Cham
[6] Division of Computing Studies Arizona State University Polytechnic Campus Mesa, Arizona 85212
[7] (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 7, 2012.
[8] International Journal of Advancements in Research Technology, Volume 2, Issue 8, August-2013 ISSN 2278-7763

APPENDIX

CODE:

```
function [ef, d] = svdRecognition0(newName, r, N, A, U, S, V, fbar, e0, e1)
    %newName = 'janet1.tiff'   r = # of sv
    chosen
    Ur = U(:, 1:r)
    X = Ur'* A
    fnew = imread(newName)
    fnew = imresize(fnew, [112, 92])
    f = reshape(fnew, 10304, 1)
    f0 = double(f)    fbar
    x = Ur'* f0   fp = Ur* x
    ef = norm(f0    fp)
    if ef < e1
        D = X    x*ones(1, N)
        d = sqrt(diag(D'*D))
        [dmin, indx] = min(d)
    if dmin < e0
        fprintf(['This␣image␣is␣face␣#', num2str(indx)])
    else
        fprintf('The␣input␣image␣is␣an␣unknown␣face')
    end
    else
        fprintf('The␣input␣image␣is␣not␣a␣face')
    end
```