**Bachelor’s Degree Report (BSc)**

**A Review of Dimensionality Reduction Methods and Their Applications**

***A Project Report***

**By**

**Raymond ULIOGHOGHA (A00015779)**

**Supervisor: Dr. Augustine NSANG**

**School of IT & Computing American University of Nigeria**

**Spring, 2018**

**A Review of Dimensionality Reduction Methods and Their Applications**

**Raymond ULIOGHOGHA**

***BSc Report***

**Project Report submitted in**

**Partial fulfillment of the requirements for the Degree of Bachelor in Computer Science**

**Supervisor: Dr. Augustine NSANG**

**School of IT & Computing American University of Nigeria**

**Spring, 2018**

### DECLARATION

I certify that this project work is carried out by myself and has not been previously submitted for the degree. And that the report is written unaided in my own words, apart from any quoted material which I identified clearly in the correct manner and fully acknowledged work by others. The work and the report were carried out under the guidance of Dr. Augustine Nsang.

Student's Name: Signature....................................

Raymond Ulioghogha

Supervisor's Name: Signature.....................................

Dr. Augustine Nsang

Dean's Name: Signature.....................................

Dr. Mathias Fonkam

SDP Coordinator’s Name: Signature.....................................

Dr. Abubakar Sadiq Hussaini

i

## ACKNOWLEDGEMENT

Firstly, I acknowledge The Almighty God for granting me the grace to carry out this project successfully. Furthermore, I give a lot of thanks to my project supervisor, Dr Augustine Nsang who provided me with valuable technical insights and moral support throughout this project.

## ABSTRACT

In the world we live in today, the reduction in data generally has seen a great rise. This is because of the numerous advantages that comes with working with smaller efficient data instead of the original large dataset. With this analogy, we can adopt Dimensionality reduction in computer science emphasizing on reducing computer memory in order to have more storage capacity on a computer. An example of this would be to reduce digital images which are then stored in 2D matrices.

Dimensionality reduction is a process where by given a collection of data points in a high dimensional Euclidean space, it is often helpful to be able to project it into a lower dimensional Euclidean space without suffering great distortion. The result obtained by working in the lower dimensional space becomes a good approximation to the original dataset obtained by working in the high dimensional space.

Dimensionality Reduction has two categories:

In the first category includes those in which each attribute in the reduced set is a linear combination of the attributes in the original dataset. These include *RP* and *PCA*. While the second category includes those in which the set of attributes in the reduced set is a proper subset of the attributes in the original dataset. These include all the other six techniques I implemented such as New Random Approach, Variance Approach, The first Novel Approach, The second Novel Approach, The Third Novel Approach and the LSA- Transform Approach.

Also, I compared these techniques mentioned above by how they preserve their images. Furthermore, I looked at the various applications we can use Dimensionality reduction example include:

* Image reduction
* Text data
* Nearest neighbor search
* K-nearest neighbor search
* Similarity search in a time series
* Clustering
* Classification

MATLAB programming language will be used to carry all forms of implementation in this project course.

KEYWORDS: Dimensionality reduction, MATLAB.

**Table of Contents**

[DECLARATION I](#_bookmark0)

[ACKNOWLEDGEMENT 2](#_bookmark1)

[ABSTRACT 3](#_bookmark2)

[CHAPTER 1 7](#_bookmark3)

* 1. [PROJECT OBECTIVE 7](#_bookmark4)
  2. [BACKGROUND 8](#_bookmark5)
  3. [ADVANTAGES 8](#_bookmark6)

[CHAPTER 2 10](#_bookmark7)

* 1. [DIMENSIONALITY REDUCTION TECHNIQUES 10](#_bookmark8)
  2. [RANDOM PROJECTION (RP) 10](#_bookmark9)
  3. [PRINCIPAL COMPONENT ANALYSIS 11](#_bookmark10)
  4. [NEW RANDOM APPROACH (NRA) 12](#_bookmark11)
  5. [SINGULAR VALUE DECOMPOSITION (SVD) 13](#_bookmark12)
  6. [VARIANCE APPROACH 15](#_bookmark13)
  7. [LATENT SEMANTIC ANALYSIS (LSA)-TRANSFORM 15](#_bookmark14)
  8. [FIRST NOVEL APPROACH 17](#_bookmark15)
  9. [SECOND NOVEL APPROACH 18](#_bookmark16)
  10. [THIRD NOVEL APPROACH 19](#_bookmark17)

[CHAPTER 3 20](#_bookmark18)

* 1. [APPLICATIONS 20](#_bookmark19)
  2. [TEXT DATA 20](#_bookmark20)
  3. [NEAREST NEIGHBOR SEARCH 21](#_bookmark21)
  4. [SIMILARITY SEARCH IN A TIME SERIES 22](#_bookmark22)
  5. [CLUSTERING 22](#_bookmark23)
  6. [CLASSIFICATION 23](#_bookmark24)
  7. [K-NEARSEST NEIGHBOR 23](#_bookmark25)

[CHAPTER 4 26](#_bookmark26)

[4.0 IMPLEMENTATION AND RESULTS OF DIMENSIONALITY REDUCTION ON IMAGES 26](#_bookmark27) [4.1REDUCTION WITH PRINCIPAL COMPONENT ANALYSIS 26](#_bookmark28)

* 1. [REDUCTION WITH RANDOM PROJECTION 27](#_bookmark29)
  2. [REDUCTION WITH THE NEW RANDOM APPROACH 28](#_bookmark30)
  3. [REDUCTION WITH VARIANCE 29](#_bookmark31)
  4. [REDUCTION WITH THE FIRST NOVEL APPROACH 30](#_bookmark32)
  5. [REDUCTION WITH THE SECOND NOVEL APPROACH 31](#_bookmark33)
  6. [REDUCTION WITH THE THIRD NOVEL APPROACH 31](#_bookmark34)
  7. [REDUCTION WITH LATENT SEMANTIC ANALYSIS 33](#_bookmark35)

[CHAPTER 5 34](#_bookmark36)

* 1. [CONCLUSIONS 34](#_bookmark37)
  2. [MATLAB CODES USED 35](#_bookmark38)

[REFERENCES 41](#_bookmark39)

# CHAPTER 1

### PROJECT OBECTIVE

This project is mainly a survey on dimensionality reduction discussing different motives why we might want to reduce the dimensionality of a dataset. Outlining various works done, methods used and finally their applications in different domains of life. This project goes further in depth to look at different dimensionality reduction methods and ways in which we can implement a few of them. Finally, this project goes further to compare these techniques to the extent in which they preserve images and outlines the various applications in random projection.

### BACKGROUND

Assume a data set D contains n points in a high dimensional space, this can be mapped out onto a lower dimensional space with minimal distortion. (*see Nsang, Novel Approaches to Dimensionality Reduction and Applications*). For example, a data set with 30,000 columns will be difficult to inspect. Evidently, it will of great assistance to obtain 1,500 columns, which will make it a lot easier to analyze the result obtained compared to the original data set with 30,000 columns. Such that after conducting an analysis of the dataset, the result obtained is a good approximation when put in contrast to the result obtained by analyzing the original data set.

### ADVANTAGES

Some advantages of reduction of dimensionality d, of a given set of n points will include:

* + 1. Dimensionality reduction, will act as a catalyst in speeding up a given algorithm whose runtime depends exponentially on the dimension of the working space. For instance, if the dimensionality of the data set, d, is too large, a complex control system will be needed to avoid over fitting of the training data in machine learning.
    2. “High dimensionality data set may hinder variation of available choices of the data processing methods” (*see Nsang, Novel Approaches to Dimensionality Reduction and Applications*). These include image data, clustering and analysis of text files etc. In continuation to the given examples, dimensionality tends to be

large due to variety in various products, wide range of phraseology, or the large image window.

* + 1. Data sets in high dimensions have a tendency to displace sporadically. Thus, an algorithm will take a long time find any structure in the given data set.
    2. Dimensionality reduction carries along the noise and other irrelevant specs of the image reduced. It is due to the variety in high dimensional data set.
    3. Dimensionality reduction helps in making visualization of the data easier when its reduced to low dimensions such as 2D or 3D.
    4. Finally, dimensionality reduction helps us to conserve time and most importantly memory space.

Even though various expensive programmed methods can produce the similar models from the same type of high-dimensional datasets, reductions of dimensionality is still recommended as the initial process before any modeling of the data.

# CHAPTER 2

### Dimensionality Reduction Techniques

This chapter contains detailed explanations of various techniques of reduction in the dimensionality reduction process, which will be used in the deduction of matrices. They will show the effects of each technique represented by various data sets.

### RANDOM PROJECTION (RP)

This method compresses data from a high-dimension to a much lower dimension. It takes allocated data in high dimension and projects it onto a smaller dimensional subspace. It is derived by using a random matrix with its column(s) having unit lengths. For example, an original R x C dimensional dataset is projected to a subspace of R x L dimension through a random C x L matrix whose column are of unit lengths as well. If 𝑋𝑛 × 𝑑 is the original set of n d-dimensional observations, then;

𝑋𝑅𝑃𝑛 × 𝑘 = 𝑋𝑛 × 𝑑 𝑅𝑑 ×𝑘

will be the reduced data on lower k- dimensional subspace. (Nsang).

In simpler terms, “if points in a vector space are projected onto randomly selected sub- space of suitable dimensions, then the distances between the points are approximately preserved”, according to Johnson-Liden Strauss Lemma on his explanation of random mapping. Text documents cluster can be explained using the RP method. The p (initial

dimension) could be as massive as 6000, however the final endpoint k, will remain relative as big as 100 order. Due to this, computationally PCA tends to be expensive. RP are integrated as a kind of data pre-processing step. The product of the lower dimensional data is then applied to clustering. In this process, application areas that include Text data, Image data, and similarity search in a time series databases will be viewed.

Note: this technique cannot work with image compression due to the values being out of range (0-255). Additionally, the new matrix turns out to be the linear combination of the initial matrix.

N.B the MATLAB code that I used to implement RP is included in the appendix.

### Principal Component Analysis

Principal component analysis is the product of the limitation of a Singular Value Decomposition (SVD) to compress high dimensional data sets with odd numbers of rows and columns which brought about PCA. PCA is based on the covariance matrix of the variables and also a second order method. By locating a handful of orthogonal linear combinations of the original variable with the largest variances, PCA reduces the dimensions of the data.

Lets say n was a given data point as a matrix M, the most accurate q-dimensional approximation for the respective data set, (q x p) must be found. The PCA based dimensionality reduction technique is based on two steps:

1. Find the PCA of M. i.e. find U, D and V such that 𝑀 = 𝑈𝐷𝑉𝑇 where:
   * U is a *K x K* orthogonal matrix (i.e. UTU = IN) whose columns are the left singular vectors of *X.*
   * *V* is an *L x L* orthogonal matrix (i.e. VT V = IL) whose columns are the right singular vectors of X
   * *D* is a *K x L* diagonal matrix with diagonal elements, which are the singular values of *M*.
   * 𝑉𝑞 Is to be the matrix whose columns are unit vectors that corresponds with the *q*

largest left singular vectors of *X*. We note that 𝑉𝑞 is an *n* × *q* matrix.

1. The transformed matrix M is given by: 𝑀𝑃𝐶𝐴 = M𝑉𝑞

The PCA works with all types of matrices, including M as an n matrix with n ≠ p.

The most obvious component analysis (PCA) method has a computational complexity of of *O (p2n) + O (p3)*

Additionally, due to the compositional values in the new matrix being less than 95, the matrix representing the image is not preserved. Thus, representing the dark points on an image, which is also a linear combination of the original attributes.

The MATLAB code that we implemented PCA on is included in the appendix.

### NEW RANDOM APPROACH (NRA)

This method is based on the creation of random numbers. The algorithm utilizes generated numbers that are between n to n. then k unique numbers are added to an empty matrix (M). After the creation of a random number, the algorithm tries to locate it in M. If

a number is not found in matrix (M), then the algorithm adds the number to M. but if the number is already in the matrix (M), the algorithm then generates another number. Sequential search algorithm was implemented to find the elements in order to find the generated numbers that fill up the matrix. In other words, for a data set D, of dimensionality d to one of the dimensionality k, a new set, Sk is formed consisting of K numbers.

Selected at random from the set S given by: S = {x ϵ N | 1  x  d}

So, our reduced set, DR, will be given by:

DR = D(:, Sk)

That is, DR is a data set having the same number of rows as in D and *Ai* is the *ith*

attribute of DR that will be the *jth* attribute of D if j is the *ith* element of *Sk*.

N.B the MATLAB code that I used to implement NRA is included in the appendix.

### Singular Value Decomposition (SVD)

Singular value decomposition is in some way similar to PCA with the exception that to find the reduced dataset we use the transpose of the original dataset D to multiply the U(:,1:q);

Where q is the number of columns in the reduced set. To give;

Dr= DT\*U(:,1:q).

Given n data points in Rp as an N x P matrix X, we want to find the best q-dimensional approximation for the data (q < p). The SVD approach achieves this by computing

The SVD of X. In other words, it finds matrices U, D and V such that X = UDV(T) where:

* + - U is an N x N orthogonal matrix (i.e. UTU = In) whose columns are the left singular vectors of X;
    - V is a P x P orthogonal matrix (i.e. V T V = Ip) whose columns are the right singular vectors of X;
    - D is an N x P diagonal matrix with diagonal elements D1 x D2 x D3 = Dp 0 which are the singular values of X. Note that the bottom rows of D are zero rows.
    - Define Uq to be the matrix whose columns are unit vectors corresponding to the q largest left singular values of X. Uq is a n \_ q matrix. The transformed matrix is given by (Bingham E. 2001):

XSVD = XTUq

Since XT is a P x N matrix, and Uq is a N x Q matrix (as stated above), it is obvious that XSV D can only be meaningfully evaluated when X is a square matrix. If it is not, the result of the evaluation will be a P x Q matrix. This does not make sense because the number of rows is not supposed to change during a dimensionality reduction operation. Thus, if X is a N x P matrix (with n 6= p), finding the SVD decomposition of X will just be the first step towards reducing its dimensionality by the PCA method described next.

### VARIANCE APPROACH

When the final set of attributes is the most accurate subset of the original dataset of attributes, we can derive a true dimensionality reduction process known as the variance approach. It has a particular it works out:

N attributes are compressed to r<n attributes that make a subset of the initial attributes. The factor of the compressed data set projected from the RP approach and PCA approach are linear combinations of factors of original data set.

In order to accept this approach, the variance of all the columns is calculated respectively, and the resulting highest variances in the columns are selected. The variance formula can be derived below

Equation defining the variance.

Where

X is the data at current row

µ is the computed average for the column and N is the number of data in the column

N.B: The implementation of variance is included in the appendix.

### Latent Semantic Analysis (LSA)-Transform

This method of dimensional reduction preserves 100% of the image’s form while reducing the original matrix to a smaller matrix. This works out well due to PCA working

well for two dimensional data sets, especially when the amounts of principal components are above a certain starting point. In other terms, properties of the new matrix M have the exact same dimensions as the initial matrix M, given it no value since its dimensions remained the same. Regardless, this is not in line with our purpose.

The main reason for why LSA-Transform was employed is: According to Augustine Nsang, the author of Novel Approaches to Dimensionality Reduction and Applications stated that “LSA-Transform uses the redundancy of the given data in the matrices that represents images. in practice specifically, if *I* is to be an image, and *M* is to be the matrix of its pixel brightness values which represents *I. The* LSA-Transform simply picks out only the even columns and rows are in *M* to form *M1.*” We then can conclude that: Generally any point on any image is represented by a whole rectangle of values in an equivalent matrix. Let us take for example; the values below may be used to represent a dark point.

93 94 88 93

87 89 89 89

87 83 88 88

93 89 88 89

From that matrix above, we can see that all the values are less than 95. So, choosing the even rows from the matrix yields:

87 89 89 89

93 89 88 89

Also, selecting only the even columns yields: 94 93

89 89

83 88

89 89

Consequently, it shows the original sub matrix that had sixteen cells, is now compressed to form a smaller matrix of eight cells. The smaller matrix also represents a dark point on the image. We can see that, with LSA-Transform executed, the product of the matrix become 25% the size of the original matrix but they both represents matrix *I.* In summary, the LSA-Transform breakdown the number of rows and columns to half in order to produce a small amount of memory and still preserves the original image.

N.B MATLAB code that i used to implement this technique can be found in the appendix.

### First Novel Approach

To reduce the dimensionality of a dataset Dn x p from *p* to *k,* our first novel approach first determines the extent to which each attribute preserves the interpoint distances. In other words, for each attribute, *x,* in *D,* it computes gxm and gxM given by:

gxm = min{ ||

*f* (*u*)  *f* (*v*) ||2 }

|| *u*  *v* || 2

gxM = max{ ||

*f* (*u*)  *f* (*v*) ||2 }

|| *u*  *v* || 2

where u and v are any two rows of *D*, and f(u) and f(v) are the corresponding rows in the dataset reduced to the single attribute *x.* The average distance preservation for the attribute *x* is then computed as:

gxmid = (gxm + gxM)/2

To reduce the dataset *D* from *p* columns to *k* columns, this approach then selects the *k*

attributes of *D* of largest *gxmid* value.

### Second Novel Approach

Just like our first approach, to reduce the dimensionality of a dataset Dn x p from *p* to *k,* our second novel approach first determines the extent to which each attribute preserves the interpoint distances. This time, the actual distance preservation for each attribute is computed. If *x* is an attribute of the dataset D, the actual distance preservation of *x* is computed as:

  ||

*n*

*n*

*f* (*u*)  *f* (*v*) ||2

*gx* 

*u* 1 *v**u* 1

|| *u*  *v* ||2

*n*

*r*

where *n* is the number of rows of *D,* u and v are any two rows of *D*, and f(u) and f(v) are the corresponding rows in the dataset reduced to the single attribute *x.* The term nr in this equation is the number of pairs of rows of D computed as:

*n* *nC*

*r*

2

 *n*(*n* 1) *.*

2

To reduce the dataset *D* from *p* columns to *k* columns, this approach then selects the *k*

attributes of *D* of largest *gx* value.

### Third Novel Approach

Finally, our third approach selects the *k* attributes of *D* which best preserve *k-means clustering.* The extent to which each attribute preserves *k-means clustering* shall be computed using the *rand* index.

As detailed below:

Function DR = Novel\_App3f (D); k = 100

[r,c] = size (D); D1 = kmeans1 (D) AML = [];

For i = 1: c,

Di = kmeans1 (D (:, i))

AMi = testkmeans1 (D1, Di) AML = [AML, AMi];

end; AML

Ls = selectbestk (AML, k) Dr = D (:,l)

# CHAPTER 3

### APPLICATIONS

Areas where dimensionality reduction can be used are numerous in the world today. From some of the advantages pointed out earlier, we can see why dimensionality reduction is useful and essential to computing and other fields of study. Some applications of dimensionality reduction include:

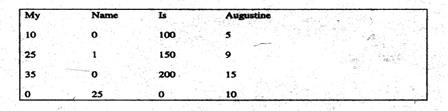
### TEXT DATA

Dimensional reduction can be used for information retrieval. This information can be retrieved using text data.

In this domain:

* + - Text data is presented in vector space
    - In this space, each document forms one d-dimensional vector where d is the vocabulary size
    - The ith element of the vector indicates some function of the frequency of the ith vocabulary item in the document.
    - Creating a text data document database this way helps us to group related documents together, and dimensionality reduction helps us to speed up this process.

Below is an example of Text data.

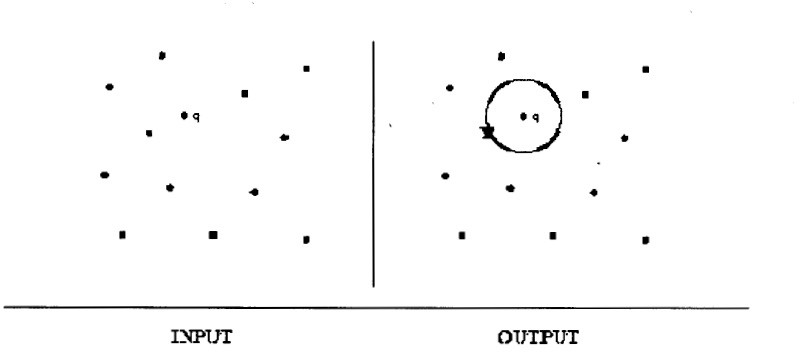


### NEAREST NEIGHBOR SEARCH

Dimensionality reduction can be used in nearest neighbor search by looking for the neighbor which is nearest to the data in particular and comparing them together. For example,

Consider a student database containing the scores and grades of fifty students out of the seventy students in a class. We can predict the grade of any of the other twenty students by simply finding its nearest neighbor. By nearest neighbor we mean the student who scores are as close as possible to that of the test student. If we have thirty scores for each

student. Finding the grade of the new student will be very time consuming thus it will be faster if we can reduce the dimensionality of the dataset to speed up this grading process. Below is an example of nearest neighbor search.



### SIMILARITY SEARCH IN A TIME SERIES

A time series database is a collection of data that are generated in series as time goes on. For example, suppose we have a patient database with the records of high blood pressure taken every thirty minutes in one day. Forty eight high blood pressure readings would have been taken for each patient each day. It will be a cumbersome process if we want to locate two patients with similar state of health. Reducing the dimensionality will help to speed up the process.

### CLUSTERING

Clustering is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense. If we have a database with high number of dimensions, and we want to group them in clusters it will be

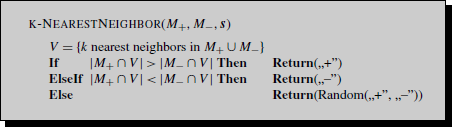
very cumbersome. If we reduce the dimensionality of the database using a suitable dimensionality technique it will be much faster grouping the rows of the dataset into different clusters.

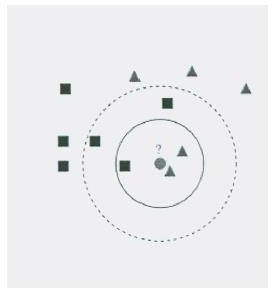
### CLASSIFICATION

Unlike clustering, classification is a method of supervised learning. The task of the supervised learner is to predict the value of the function for any valid input after having seen a number of training examples (i.e. pair of input and target output) in unsupervised learning, there are no training samples. The classification of the feature vectors must be based on the “similarity” between them.

### K-NEARSEST NEIGHBOR

The K-NEAREST NEIGHBOR classification algorithm makes a majority decision among the *k* nearest neighbors.



For example, the test sample which is the circle should be classified either to the first class of squares or to the second class of triangles.

If k = 3 it is classified to the second class because there are two triangles and only one square inside the inner circle.

If k = 5 it is classified to first class. Because they are three squares against two triangles inside the outer circle.

# CHAPTER 4

### IMPLEMENTATION AND RESULTS OF DIMENSIONALITY REDUCTION ON IMAGES

#### Reduction with Principal Component Analysis



* 1. ***Reduction with Random Projection***



#### Reduction with the New Random Approach.



* 1. ***Reduction with Variance.***



#### Reduction with the First Novel Approach.



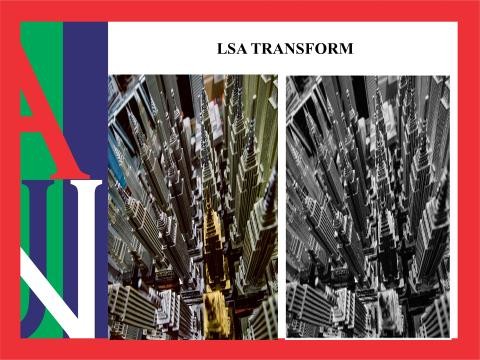
* 1. ***Reduction with the Second Novel Approach.***



#### Reduction with the Third Novel Approach.



* 1. ***Reduction with Latent Semantic Analysis.***



# CHAPTER 5

### CONCLUSIONS

To sum-up this project, I learnt a lot from learning dimensionality reduction. Firstly, those in the first category such as random projection and principal component analysis are not so good when used to reduce images. But when not used on images, RP has more advantages on the overall than PCA. Amongst these advantages is the fact that RP is less computationally expensive than PCA, and as such is more feasible on high dimensional data. This advantage alone clearly makes it a better dimensionality reduction method overall. Considering that, in practice, we will be dealing with data in very large dimensions most of the time.

In the second category of dimensionality reduction, *LSA-Transform* is the best in preserving images, considering both speed and quality of the images. Also, the three novel approaches and the *New Random Approach* are approximately as good as each other in preserving images. Finally, I discovered that the three novel approaches have very low speeds of execution and can only be used to reduce very small images.

### MATLAB CODES USED

##### RANDOM PROJECTION

*D = load('D.txt')*

*R1 = round(1000\*rand(38,20)); for j = 1 : 38,*

*for k = 1 : 20,*

*if (R1(j, k) >= 0) && (R1(j, k) < 666) R(j, k) = 0;*

*elseif (R1(j, k) >= 666) && (R1(j, k) < 833) R(j, k) = -1;*

*else*

*R(j, k) = 1;*

*end; end;*

*end;*

*DR = D \* R*

##### PRINCIPAL COMPONENT ANALYSIS

*D = load('BA.txt')*

*[U, S, V] = svd(D);*

*DR = D\*V(:,1:20)*

##### NEW RANDOM APPROACH

*D = load('BP.txt')*

*[nr, nc] = size(D) A = []*

*x = 1*

*while (x < 16) r = nc\*rand(1) r = ceil(r)*

*y = ismember(r, A) if (y == 0)*

*A = [A r]*

*x = x + 1 end*

*nc end*

*A = sort(A) DR = D(:, A)*

##### VARIANCE

*M3 = load('Dy.txt') V = var(M3)*

*[r, n] = size(M3) k = 8;*

*l = [];*

*l2 = []; t = 0;*

*while t < k max = 0;*

*for i = 1 : n x = V(i);*

*if (x > max) && not(ismember(i,l2)) max = x;*

*p = i; end*

*end*

*l = [l, max]*

*l2 = [l2, p] t = t + 1*

*end*

*l2 l*

*ls = sort(l2) DR = M3(:,ls)*

##### VARIANCE TRANSFORM

*A = imread('Godlove.jpg'); B1 = rgb2gray(A); imshow(A);*

*figure, imshow(B1); B = load('B.txt');*

*%figure, imshow(B); V = var(B);*

*[r,c] = size(B1); k = 525;*

*L = [];*

*cx = c; Vnew = V; for i = 1 : k*

*m = max(Vnew); l = locate(m, V); l1 = l + 1;*

*lc = locate(m, Vnew); if not(ismember(l,L))*

*L = [L, l];*

*else*

*l = locate2(m,V,l1); L = [L,l];*

*end;*

*l1 = lc - 1; l2 = lc + 1; if lc == cx*

*Vnew = Vnew(:,1:l1); else*

*Vnew = Vnew(:,[1:l1,l2:cx]); end;*

*cx = cx - 1; end*

*L*

*L = sort(L) M3R = B1(:,L);*

*figure, imshow(M3R);*

##### FIRST NOVEL APPROACH

*D = load('B.txt')*

*k = 400;*

*%D = D(:,9:15)*

*[r,c] = size(D);*

*M1 = []; L1 = [];*

*gmid = [] for i = 1 : c;*

*[g, G] = computegG(D, D(:,i));*

*% L1 = [i; g; G];*

*gmid(i) = (g + G)/2;*

*% M1 = [M1, L1];*

*end; gmid*

*l = selectbestk(gmid, k) Dr = D(:,l)*

##### SECOND NOVEL APPROACH

*function ls = Novel\_App2f(D);*

*k = 100;*

*D = D(:,481:640)*

*[r,c] = size(D);*

*M1 = []; L1 = [];*

*dp = dist\_preserve(D) ls = selectbestk(dp, k)*

*%Dr = D(:,l)*

##### THIRD NOVEL APPROACH

*function ls = Novel\_App3(D); D = load('B.txt');*

*k = 400*

*[r,c] = size(D); D1 = kmeans1(D) AML = [];*

*for i = 1 : c,*

*Di = kmeans1(D(:,i))*

*AMi = testkmeans1(D1, Di) AML = [AML, AMi];*

*end; AML*

*ls = selectbestk(AML, k)*

*%Dr = D(:,l)*

**REFERENCES**

* + 1. A. Nsang and A. Ralescu. A review of dimensionality reduction methods and their applications. In *Proceedings of the Twentieth Midwest Artificial Intelligence and Cognitive Science Conference* (MAICS 2009)*,* pages 118–123, 2009.
    2. A. Nsang and A. Ralescu. Approaches to dimensionality reduction to a subset of the original dimensions. *In Proceedings of the Twenty-First Midwest Artificial Intelligence and Cognitive Science Conference* (MAICS 2010), 2010.
    3. A. Nsang and A. Ralescu. More approaches to dimensionality reduction to a subset of the original attribute set. In *Proceedings of the Twenty-First Midwest Artificial Intelligence and Cognitive Science Conference* (MAICS 2010), 2010.
    4. Achlioptas, D. "Database friendly random projections." Achlioptas, D. *Journal of Computer and System Sciences*. 2003. 6671 - 678.
    5. Yu, Lei, Jeiping ye and Huan Liu. *Dimensionality Reduction for Data Mining - Techniques, Applications and Trends*. n.d. Binghamton University: State University of New York; Arizona State University.