**Bachelor’s Degree Report (BSc)**

**CATEGORIZATION OF DATA USING HIERARCHICAL**

**CLUSTERING**

**By**

**Nana Asmau Isa NASIR (A00017965)**

**Olumide Peter ERETAN (A00015889)**

**Supervisors: Dr. Augustine Nsang**

**School of Information, Technology & Computing American University of Nigeria**

**CATEGORIZATION OF DATA USING HIERARCHICAL**

**CLUSTERING**

**Nana Asmau Isa NASIR (A00017965)**

**Olumide Peter ERETAN (A00015889)**

***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 Information, Technology & Computing American University of Nigeria**

# DECLARATION

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

Nana Asmau Nasir Isa Signature.................................... Student

Olumide Eretan Signature..................................... Student

Dr. Augustine Nsang Signature..................................... Project Supervisor

Dr. Narasimha Rao Signature.................................... Program Chair

Dr. Mathias Fonkam Signature..................................... Dean

Dr. Abubakar Sadiq Hussaini Signature..................................... SDP Coordinator

# ABSTRACT

In this project, we shall implement the hierarchical clustering algorithm and apply it to various data sets such as the weather data set, the student data set, and the patient data set. We shall then reduce these datasets using the following dimensionality reduction approaches: *Random Projections* (RP), *Principal Component Analysis* (PCA), *Variance* (Var), the *New Random Approach* (NRA), the *Combined Approach* (CA) and the *Direct Approach* (DA).

The *rand* index and *ARI* will be implemented to measure the extent to which a given dimensionality reduction method preserves the hierarchical clustering of a data set. Finally, the six reduction methods will be compared by runtime*,* inter-point distance preservation, variance preservation and hierarchical clustering preservation of the original data set.

# ACKNOWLEDGEMENT

We thank our supervisor for being very patient with us and for making time out of his busy schedule to guide us.

We would also like to thank Usman Baba for helping us install the MATLAB software and for always being available whenever we needed his assistance with the it.

# DEDICATION

I, Nana Asmau Nasir Isa dedicate this to my parents for their constant prayers which were my light whenever things got dark, and Allah for everything he has done for me and for seeing me through.

I, Olumide Eretan dedicate this project to my wonderful parents for their unwavering support through my ups and my downs and most importantly to Jesus Christ, my lord and my savior.

Table of Contents

[DECLARATION i](#_bookmark0)

[ABSTRACT ii](#_bookmark1)

[ACKNOWLEDGEMENT iii](#_bookmark2)

[DEDICATION iv](#_bookmark3)

[LIST OF FIGURES vi](#_bookmark4)

[LIST OF TABLES vii](#_bookmark5)

1. [INTRODUCTION 1](#_bookmark6)
2. [HIERARCHICAL CLUSTERING 2](#_bookmark7)
	1. [SNIPPET OF CLUSTERED DATA 2](#_bookmark8)
3. [DIMENSIONALITY REDUCTION TECHNIQUES 4](#_bookmark9)
	* 1. [RANDOM PROJECTIONS (RP) 4](#_bookmark10)
		2. [PRINCIPAL COMPONENT ANALYSIS (PCA) 4](#_bookmark11)
		3. [NEW RANDOM APPROCAH 5](#_bookmark12)
		4. [VARIANCE 6](#_bookmark13)
		5. [COMBINED APPROACH 6](#_bookmark14)
		6. [DIRECT APPROACH 7](#_bookmark15)
4. [IMPLEMENTATION 9](#_bookmark16)
	* 1. [RANDOM PROJECTION (RP) 10](#_bookmark17)
		2. [PRINCIPAL COMPONENT ANALYSIS (PCA) 11](#_bookmark18)
		3. [NEW RANDOM APPROACH 12](#_bookmark19)
		4. [VARIANCE 13](#_bookmark20)
		5. [DIRECT APPROACH 14](#_bookmark21)
		6. [COMBINED APPROACH 15](#_bookmark22)
5. [RAND INDEX 16](#_bookmark23)
6. [CONCLUSION 17](#_bookmark26)
7. [REFERENCES 18](#_bookmark27)
8. [Appendix A MATLAB CODES USED FOR IMPLEMENTATION 19](#_bookmark28)

# LIST OF FIGURES

[Figure i Student data set and averages of each row 2](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825591)

[Figure ii Clusters of each row on the student data 3](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825592)

[Figure iii Averages of rows on the student data (snippet) 3](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825593)

[Figure iv Reduced from 15 to 10 columns (RP) 10](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825594)

[Figure v Reduced from 15 to 12 columns (RP) 10](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825595)

[Figure vi Reduced from 15 to 10 columns (PCA) 11](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825596)

[Figure vii Reduced from 15 to 12 columns (PCA) 11](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825597)

[Figure viii Reduced from 15 to 10 (NRA) 12](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825598)

[Figure ix Reduced from 15 to 12 (NRA) 12](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825599)

[Figure x Reducing from 15 to 12 (Variance) 13](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825600)

[Figure xi Reducing 15 to 10 columns (Variance) 13](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825601)

[Figure xii Reduced from 15 to 10 columns (Dir. App.) 14](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825602)

[Figure xiii Reduced from 15 to 10 (Dir. App.) 14](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825603)

[Figure xiv Reduced from 15 to 10 (Comb. Appr.) 15](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825604)

[Figure xv Reduced from 15 to 10 (Comb. Appr.) 15](file://localhost/C%3A/Users/User/Downloads/SDP%20Project%20Report_1st%20Draft.docx%23_Toc513825605)

# LIST OF TABLES

[Table i Comparison of DR Techniques (10 columns) 16](#_bookmark24)

[Table ii Comparison of DR techniques (12 columns) 16](#_bookmark25)

# INTRODUCTION

Given a data set containing *n* points in high dimensional space, it is often helpful if it can be projected onto a lower dimensional space without suffering great distortion. This process is called dimensionality reduction. Essentially, dimensionality reduction reduces the number of variables to be considered in a way that the relevant data is retained while reducing the amount of the data.

Dimensionality reduction helps to reduce the runtime of algorithms whose runtime depends on the dimensions of the working space. It also broadens the scope for the choice of method for data processing. It provides complexity control which avoids overfitting of the training data.

Dimensionality can be applied in several domains which include text data, image data, nearest neighbor search and in the domain of clustering and classification. 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. Clustering is a method of unsupervised learning. Classification, on the other hand, 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). As mentioned above, this project focuses on the categorization of data using hierarchical clustering.

# HIERARCHICAL 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.

With hierarchical clustering, the clusters are arranged in a hierarchy. The termination condition could be chosen as, for example, a desired number of clusters or a minimum distance between clusters.

In hierarchical clustering we begin with *n* clusters consisting of one point each.

Then the nearest neighbor clusters are combined until a termination criterion has been reached.

## SNIPPET OF CLUSTERED DATA



**Figure i Student data set and averages of each row**

Above is a snippet of the student data we used for this project. The averages of each row have been computed. After implementing the hierarchical clustering algorithm on the data set, the data is divided in 20 clusters.



**Figure iii Averages of rows on the student data (snippet)**

**Figure ii Clusters of each row on the student data**

Above, the averages of each row are put side by side with the cluster of each row. Notice that rows 1,2,4,6 and 12 are in cluster 1 and the averages of these rows are 73, 65, 83.8, 75.4, and 59.3. These numbers are within the range of 55-85. The range is large because we have a small number of clusters. If we had a higher number of clusters, the range would be a lot smaller

# DIMENSIONALITY REDUCTION

**TECHNIQUES**

## RANDOM PROJECTIONS (RP)

A dataset of M-dimensions is reduced to an L-dimensional dataset where (L << M) using a random M x L matrix R. The reduced dataset is given by:

BN x L = AN x M RM x L

Where:

* + - * BN x L is the reduced dataset
			* AN x M is the original high dimensional dataset and
			* RM x L is the Random Matrix

In this technique, a high dimensional data is compressed to a much lower dimensional data using a derived matrix R with M rows and L columns.

## PRINCIPAL COMPONENT ANALYSIS (PCA)

The Singular Value Decomposition (SVD) had an inability to successfully reduce higher dimensional data sets with unequal number of rows and columns, so the Principal Component Analysis was made to help solve this problem. This approach is second-order approach and is based practically on the covariance matrix of the variables. Another name for this approach is empirical orthogonal function (EOF) method or the SVD.

Given a data set D in a high dimension by matrix m x n to be reduced to a dataset of q columns, this approach finds the SVD of the data set by decomposing it into three matrices [U, S, V] such that D= USV transpose.

* U is an m x m orthogonal matrix (i.e. UTU = Im) whose columns are the left singular vectors of X;
* V is a n x n orthogonal matrix (i.e. VTV = In) whose columns are the right singular vectors of X;
* S is an m x n diagonal matrix with diagonal elements d1 ≥ d2 ≥ 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 m x q matrix.

The singular value decomposition of the original dataset, *D,* is obtained i.e. [U, S, V] = svd(D)

The reduced dataset becomes: Dr = D\*V(:,1:q)

where q is the number of columns in the reduced dataset.

## NEW RANDOM APPROCAH

*P* random numbers in the range 1..*m* are generated (if *m* and *p* are the numbers of dimensions in the original and reduced space respectively), making sure no random number is generated more than once. The columns corresponding to the random numbers form the columns of the reduced dataset.

This approach is basically about generating random numbers in a range then; unique numbers are added to an empty matrix (M). When a random number is generated the algorithm tries to locate the number in the matrix, and if the number is not there, it adds it to the matrix automatically. If the number is found in the matrix, it ignores it and then generates another number. After P random numbers are generated in the matrix, the algorithm terminates. The numbers in the matrix corresponding with the columns in the original dataset gives the reduced data set.

Let’s assume that reducing a data set D to a lower dimension of k columns, it will be given

by;

#### K = {x ϵ N | 1  x  D}

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

**DR = D(:, K)**

## VARIANCE

The variance approach is used to reduce a high dimensional data set to a proper subset of lower dimension. With this approach the variance of all the columns are calculated and the highest ones are picked depending on how many columns one wishes to reduce the data set to. Using the formula below, variance is calculated.



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

## COMBINED APPROACH

For each attribute, *x*, of a dataset *D,* g*x*mid (which is the approximate extent to which it preserves the interpoint distances) is computed. For a reduction of *D* from *p* columns to *k* columns, this approach selects the combination of *k* attributes of *D* whose average gxmid

value are maximum. This approach also reduces a high dimensional data to a proper subset

of lower dimension. The combined approaches selects the pair of columns that best preserves the inter-point distances between the tuples of the original data set. The best pairs of columns are the ones which have the highest value of gmid, the average of gm and gM. This goes to say that if we want to reduce a dataset Dnxp to a dataset containing k columns, the *Combined Approach* selects the combination of *k* attributes which best preserve the inter-point distances, and reduces the dataset to a dataset containing only those *k* attributes. To do so, it 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 }

*f* (*u*)  *f* (*v*) ||2

gxM = max{

|| *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

The ones with the highest gxmid are selected depending on how many columns one wishes to reduce the data to.

## DIRECT APPROACH

This approach is similar to combined approach. It selects the combination of *k* attributes which best preserve the inter-point distances. For a reduction of a data set *D* from *p* columns to *k* columns, the value of g*C*mid is computed for each combination, C, of k attributes of *D (*g*C*mid is the approximate extent to which *C* preserves the inter-point distances). The combination of *k* attributes of *D* with largest gCmid value is selected. Each combination, *C,* it computes gcm and gcM given by:

||

gcm = min{

||

*f* (*u*)  *f* (*v*) ||2

|| *u*  *v* ||2 }

*f* (*u*)  *f* (*v*) ||2

gcM = max{

|| *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 attributes in C. The average distance preservation for this combination of attributes is then computed as:

**i.**gcmid = (gcm + gcM)/2

The difference between these approaches is that in combined approach the gcmid of each combination C is calculated and we us one of the formulas above to calculate the average distance preservation of each combination. Whereas in direct approach the gxmid of each attribute is calculated and then for every combination of the attribute, the average distance preservation is calculated by finding the averages of the distance preservations of each attribute in the combinations

# IMPLEMENTATION

We use the student data set to implement the six dimensionality reduction algorithms as discussed earlier. The hierarchical clustering algorithm is implemented on the data set and set aside. In this research, we only group the data into 20 clusters. The data set is reduced to 10 and 12 columns using each of the dimensionality reduction techniques mentioned earlier. The hierarchical clustering algorithm is then run on the reduced data. The reduced clustered data and the original clustered data are compared using rand index. Each of the dimensionality reduction techniques will be compared by runtime, inter point distance preservation and variance preservation.

## RANDOM PROJECTION (RP)

As explained earlier, random projection uses a projecting matrix to multiply the original data set to produce a reduced data set. The reduced data sets are below.

**Figure iv Reduced from 15 to 10 columns (RP)**



**Figure v Reduced from 15 to 12 columns (RP)**

## PRINCIPAL COMPONENT ANALYSIS (PCA)

As previously explained, this approach finds the SVD of the data set by decomposing it into three matrices [U, S, V] such that D= USV transpose.



**Figure vi Reduced from 15 to 10 columns (PCA)**



**Figure vii Reduced from 15 to 12 columns (PCA)**

## NEW RANDOM APPROACH

As mention previously, with this approach, *P* random numbers in the range 1..*m* are generated (if *m* and *p* are the numbers of dimensions in the original and reduced space respectively). The columns corresponding to the random numbers form the columns of the reduced dataset.

**Figure viii Reduced from 15 to 10 (NRA)**



**Figure ix Reduced from 15 to 12 (NRA)**

## VARIANCE

With this approach the variance of all the columns are calculated and the highest ones are picked depending on how many columns one wishes to reduce the data set to.

**Figure xi Reducing 15 to 10 columns (Variance)**



**Figure x Reducing from 15 to 12 (Variance)**

## DIRECT APPROACH

As previously mentioned, for a reduction of a data set *D* from *p* columns to *k* columns, the value of g*C*mid is computed for each combination, C, of k attributes of *D* The combination of *k* attributes of *D* with largest gCmid value is selected.



**Figure xii Reduced from 15 to 10 columns (Dir. App.)**



**Figure xiii Reduced from 15 to 10 (Dir. App.)**

## COMBINED APPROACH

For each attribute, *x*, of a dataset *D,* is computed. For a reduction of *D* from *p* columns to *k* columns, this approach selects the combination of *k* attributes of *D* whose average gxmid value is maximum.

**Figure xiv Reduced from 15 to 10 (Comb. Appr.)**

5

**Figure xv Reduced from 15 to 10 (Comb. Appr.)**

# RAND INDEX

After reducing the data sets using the dimensionality reduction techniques, the hierarchical clustering algorithm is implemented on each reduced set.

Using Rand index, we compare all 6 reduction techniques for hierarchical clustering preservation.

**Table i Comparison of DR Techniques (10 columns)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Clustering Preservation | Runtime | Interpoint distance preservation | Variance Preservation |
| PCA | 100% | 0.06% | 99.72% | 99.85% |
| RP | 100% | 0.33% | 28.03% | 21.41% |
| Comb\_App | 100% | 6.50% | 81.39% | 79.79% |
| Dir\_App | 100% | 6.50% | 83.64% | 83.63% |
| Variance | 100% | 0.77% | 85.60% | 86.10% |
| NRA | 100% | 0.07% | 79.25% | 78.87% |

**Table ii Comparison of DR techniques (12 columns)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Clustering Preservation | Runtime | Interpoint distancepreservation | Variance Preservation |
| PCA | 100% | 0.06% | 98.50% | 99.24% |
| RP | 100% | 0.01% | 38.54% | 28.51% |
| Com\_App | 100% | 6.70% | 70.67% | 68.86% |
| Dir\_App | 100% | 6.70% | 69.25% | 68.41% |
| Variance | 100% | 0.03% | 72.43% | 73.65% |
| NRA | 100% | 0.02% | 68.58% | 69.34% |

# CONCLUSION

In conclusion, dimensionality reduction is a good way to reduce data so that they can be hierarchically clustered faster. Its saves time and memory. Hierarchical clustering is useful in several aspects of our daily lives (e.g. school results and medical records) and dimensionality reduction can improve the speed at which its implemented.

As seen on the table above, all reduction techniques fully preserve the hierarchical clustering of the data set. The fastest reduction technique is Random Projections and it also preserves much of the inter-point distance and variance. Principle component Analysis is the fast and it also preserves inter-point distances and variance the highest.

# REFERENCES

* + - 1. *D. Achlioptas. Database-friendly random projections. In Proc. ACM Symp. on the Principles of Database Systems, pages 274–281, 2001.*
			2. *Augustine Nsang “Novel Approaches to Dimensionality Reduction and Applications”*

# APPENDIX

**MATLAB CODES USED FOR IMPLEMENTATION**

### HIERARCHICAL CLUSTERING

D = load('RS10 PCA.txt');

%X = XX(:,[2,3,4,5,7,9,10]);

%[U, S, V] = svd(XX);

%X = XX\*V(:,1:7);

[n, q] = size(D); k1 = 30;

% r = n - k1;

Y = zeros(n, n); for i = 1 : n

Y(i,1) = i;

end; Y

l = n;

while l > k1

P = Generatepairs(l); [r, c] = size(P)

Dmin = distance(D, Y, 1, 2);

Pmin = 1; for i = 1 : r,

DD = distance(D, Y, P(i,1), P(i,2)); if DD < Dmin

Dmin = DD;

Pmin = i;

end end;

Y = Merge(Y, P(Pmin,1), P(Pmin,2),l) l = l - 1

end; Clusters = Y

D2 = zeros(1,n); D2 = D2';

i = 1;

while Y(i,1) ~= 0 for j = 1 : n

x = Y(i,j); if x ~= 0

D2(x) = i

end;

end;

i = i + 1 end;

D2 Y

### RANDOM PROJECTION

*D = load('student\_data.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('student.txt') [U,S,V]=svd(D)

DR=D\*V(:,1:12)

### VARIANCE

M3 = load('student.txt') V = var(M3)

[r, n] = size(M3)

k = 10;

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)

### NEW RANDOM APPROACH

D = load('student.txt'); [r1,c1] = size(D);

M = []; i = 0;

k = 10;

while i < k found = 0;

x = round(c1\*rand(1)); [r, c] = size(M);

q = 1;

while (q <= c) & not(found) if (x == M(q))

found = 1; end

q = q + 1; end

if not(found) & (x ~= 0) M = [M,x];

i = i + 1; end

end

M = sort(M) DR = D(:,M)

### RAND INDEX

L = [1:100];

Mx = Combinations(L,2); M = sort2(Mx)

[r,c] = size(M);

M1 = load('D2H student.txt') M2 = load('D2HS10 Var.txt')

[r1,c1] = size(M1);

[r2,c2] = size(M2); count = 0;

for i = 1 : r,

LL = M(i,:); x = LL(1);

y = LL(2);

if (M1(x,c1) == M1(y,c1)) & (M2(x,c2) == M2(y,c2))

count = count + 1;

elseif (M1(x,c1) ~= M1(y,c1)) & (M2(x,c2) ~= M2(y,c2)) count = count + 1;

end; end;

count r

### DIRECT APPROACH

drnewx2;

[r,c] = size(M2x); i = 1; found = 0;

while (i <= r) && not(found) if M2x(i,k3) ~= Mc

i = i + 1; else

found = true; end

end

l = M2x(i,1:k)

### COMBINED APPROACH

drnewx2;

[r,c] = size(M3x); i = 1; found = 0;

while (i <= r) && not(found) if M3x(i,k3) ~= Md

i = i + 1; else

found = true; end

end

l = M3x(i,1:k)