Image Reduction Using Assorted Dimensionality Reduction Techniques

Augustine S. Nsang
Department of Computer Science
School of Inf Tech and Computing
American University of Nigeria
Yola By-Pass, PMB 2250, Yola, Nigeria
augustine.nsang@aun.edu.ng

Abdullahi Musa Bello
Department of Computer Science
School of Inf Tech and Computing
American University of Nigeria
Yola By-Pass, PMB 2250, Yola, Nigeria
abdullahi.bello@aun.edu.ng

Hammed Shamsudeen
Department of Computer Science
School of Inf Tech and Computing
American University of Nigeria
Yola By-Pass, PMB 2250, Yola, Nigeria
shamsudeen.hammed@aun.edu.ng

Abstract

Dimensionality reduction is the mapping of data from a high dimensional space to a lower dimension space such that the result obtained by analyzing the reduced dataset is a good approximation to the result obtained by analyzing the original data set.

There are several dimensionality reduction approaches which include *Random Projections*, *Principal Component Analysis*, the *Variance* approach, *LSA-Transform*, the *Combined* and *Direct* approaches, and the *New Random Approach.*

In this paper, we propose three new techniques, each of which will be a modified version of the last three techniques mentioned above (the *Combined* and *Direct* approaches, and the *New Random Approach*). We shall implement each of the ten reduction techniques mentioned, after which we shall use these techniques to compress various pictures. Finally, we shall compare the ten reduction techniques implemented in this paper with each other by the extent to which they preserve images.

Index Terms— dimensionality reduction, image compression, principal component analysis

1. Introduction

Given a collection of *n* data points (vectors) in high dimensional space, it is often helpful to be able to project it into a lower dimensional space without suffering great distortion (NR 2010a). In other words, it is helpful if we can embed a set of *n* points in *d*-dimensional space into a *k*-dimensional space, where $k << d$. This operation is known as *dimensionality reduction*.

There are many known methods of dimensionality reduction. These include *Random Projection (RP), Singular Value Decomposition (SVD), Principal Component Analysis (PCA), Kernel Principal Component Analysis (KPCA), Discrete Cosine Transform (DCT)* and *Latent Semantic Analysis (LSA)* (NR 2009). For each of these methods, each attribute in the reduced set is a linear combination of the attributes in the original data set.

Other dimensionality reduction methods, however, reduce a dataset to a subset of the original attribute set. These include the *Combined Approach (CA)*, the *Direct Approach (DA)*, the *Variance Approach (Var)*, *LSA-Transform*, the *New Top-Down Approach (NTDn)*, the *New Bottom-Up Approach (NBUp)*, the *Weighted Attribute Frequency Approach (WAF)* and the *Best Clustering Performance Approach (BCP)* (Nsang 2011).

Dimensionality reduction has several advantages, the most important of which is the fact that with dimensionality reduction, we could drastically speed up the execution of an algorithm whose runtime depends exponentially on the dimensions of the working space. At the same time, the solution found by working in the low dimensional space is a good approximation to the solution in the original high dimensional space.

One application of dimensionality reduction is in the compression of image data. In this domain, digital images are stored as 2D matrices which represent the brightness of each pixel. Usually, the matrix representing an image can be quite large, and for this reason it could be very time consuming querying this matrix to find out any information about the features of the image. In this paper, dimensionality reduction techniques are used to reduce the matrix representation of an image. This makes it possible to query the reduced matrix to get any information about the original image. Besides, we can use these techniques to compress all the pictures we have in a given folder, or website, thus conserving memory.

The rest of the paper is organized as follows. In Section 2, we shall examine the different dimensionality reduction techniques that shall be used to reduce the images. In Section 3 we shall look at the effects of reducing images using each of these techniques, and in Section 4 we shall compare the ten reduction techniques implemented in this project with each other by the extent to which they preserve images, and by their speeds of execution. Then we shall conclude this paper in Section 5.

2. Dimensionality Reduction Techniques

In this section, we shall examine the different reduction techniques that we will be using to reduce the images. They include the following:

2.1 Random Projection

In *Random Projection*, the original *d*-dimensional data is projected to a *k*-dimensional ($k << d$) subspace through the origin, using a random $d \times k$ matrix *R* whose columns have unit lengths (Bingham E. 2001). If X_{nxd} is the original set of *n* *d*-dimensional observations, then

$$X_{nxk}^{RP} = X_{nxd} R_{dxk}$$

is the projection of the data onto a lower k-dimensional subspace.

The key idea of random mapping arises from the Johnson Lindenstrauss lemma (Johnson W. B.) which states that if points in a vector space are projected onto a randomly selected subspace of suitably high dimension, then the distances between the points are approximately preserved.

2.2 Principal Component Analysis (PCA)

Given *n* data points in $\Re^P$ as an $n \times p$ matrix *X*, we want to find the best *q*-dimensional approximation for the data $(q << p)$. The PCA approach achieves this by first computing the Singular Value Decomposition of X. In other words, it finds matrices *U, D* and *V* such that $X = UDV^T$ where:

- U is an $n \times n$ orthogonal matrix (i.e. $U^TU = I_n$) whose columns are the left singular vectors of X;
- V is a $p \times p$ orthogonal matrix (i.e. $V^TV = I_p$) whose columns are the right singular vectors of X;
- D is an $n \times p$ diagonal matrix with diagonal elements $d_1 \geq d_2 \geq d_3 \ldots \geq d_p \geq 0$ which are the singular values of X. Note that the bottom rows of D are zero rows.
- Define U_q to be the matrix whose columns are unit vectors corresponding to the q largest left singular values of X. U_q is a $n \times q$ matrix.

The transformed matrix is given by (Bingham E. 2001):

$$X^{SVD} = X^T U_q$$

2.3 The Variance Approach (NR 2010b)

With the *Variance* approach, to reduce a dataset *D* to a data set D_R, we start with an empty set, *I*, and then add dimensions of *D* to this set in decreasing order of their variances. That means that a set *I* of *r* dimensions will contain the dimensions of top r variances. Intuitively, it easy to justify why dimensions of low variance are left out as they would fail to discriminate between the data. (Indeed, in an extreme case where all the values along a dimension are equal, the variance is 0, and hence this dimension cannot distinguish between data points). Thus, let

$$I_r = \{i_1, \ldots, i_r\} \subset \{1, \ldots, n\},$$

the collection of dimensions corresponding to the top *r* variances. That is i_1 denotes the dimension of largest variance, i_2 the dimension of next larger variance, etc. The reduced data base is obtained by extracting the data corresponding to the selected dimensions. That is, project *D* on I_r to obtain:

$$D_R = D(:, I_r),$$

where D_R has the same number of rows as D and r columns: the i^{th} column of D_R is the column of the original database with the i^{th} largest variance.

2.4 LSA-Transform (Nsang 2011)

LSA-Transform is probably the best technique for reducing image data. It makes use of the redundancy of the data in matrices that represent images, in practice. Specifically, if I is an image, and M is the matrix (of pixel brightness values) representing I, LSA-Transform simply selects only the even columns and rows of M to give M1. The simple explanation for this is as follows: one point on an image is usually represented by a whole rectangle of values in the corresponding matrix. For instance, a dark point maybe represented by the values:

93 94 88 93
87 89 89 89
87 83 88 88
93 89 88 89

Each of these values, as we can see, is less than 95. Selecting only the even rows leaves us with:

87 89 89 89
93 89 88 89

Similarly, selecting only the even columns leaves us with:

89 89
89 89

Thus the original sub-matrix of sixteen cells becomes reduced to a smaller matrix of four cells, which also represents one dark point on an image. After the execution of LSA-Transform, therefore, M and M1 become two matrices representing I, one a quarter the size of the other.

2.5 The *Combined* Approach (NR 2010b)

Like the two previous approaches, the *Combined* Approach is one approach which reduces a dataset *D* to a subset of the original attribute set.

To reduce a dataset D_{nxp} to a dataset containing *k* columns, the *Combined Approach* selects the combination of *k* attributes which best preserve the interpoint 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 $g_x m$ and $g_x M$ given by:

$$g_x m = \min\left\{ \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2} \right\}$$

$$g_x M = \max\left\{ \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2} \right\}$$

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:

$$g_x mid = (g_x m + g_x M)/2$$

To reduce the dataset D from p columns to k columns, this approach then finds the combination of k attributes whose average value of $g_x mid$ is maximum.

2.6 The *Direct* Approach (NR 2010b)

As with the *Combined Approach*, to reduce a dataset D_{nxp} to a dataset containing k columns, the *Direct Approach* selects the combination of k attributes which best preserve the interpoint distances, and reduces the original dataset to a dataset containing only those k attributes. To do so, it first generates all possible combinations of k attributes from the original p attributes. Then, for each combination, C, it computes $g_c m$ and $g_c M$ given by:

$$g_c m = \min\left\{ \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2} \right\}$$

$$g_c M = \max\left\{ \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2} \right\}$$

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:

$$g_c mid = (g_c m + g_c M)/2$$

To reduce the dataset D from p attributes to k attributes, this approach then finds the combination of k attributes whose value of $g_c mid$ is maximum.

As we can see, the difference between the *Combined* and *Direct* Approaches is that for the *Combined Approach*, we first find the average distance preservation for each attribute, and then, for any combination of attributes, we compute its average distance preservation by finding the averages of the distance preservations of the individual attributes. With the *Direct Approach*, on the other hand, to find the average distance preservation for any combination of attributes, C, we reduce the original dataset directly to the dataset containing only the attributes in C, and then compute the average distance preservation for this combination using the formulas above.

2.7 The New Random Approach

This is a technique suggested by Nsang, Maikori, Oguntoyinbo and Yusuf in (NMOY 2015). With this technique, to reduce a data set D of dimensionality d to one of dimensionality k, a set S_k is formed consisting of k numbers selected at random from the set S given by:

$$S = \{x \in \mathrm{N} \mid 1 \leq x \leq d\}$$

Then, our reduced set, D_R, will be given by:

$$D_R = D(:, S_k)$$

That is, D_R is a data set having the same number of rows as D, and if A_i is the i^{th} attribute of D_R, then A_i is the j^{th} attribute of D if j is the i^{th} element of S_k.

2.8 The Modified *Combined* Approach

As we saw in Section 2.5 above, the *Combined* Approach computes the average distance preservation of a combination of attributes by computing the average of their $g_x mid$ values. It's very clear that for any given attribute, x, $g_x mid$ is only an estimate of its average distance preservation, since it is computed as the midpoint between $g_x m$, the minimum distance preservation, and $g_x M$, the maximum distance preservation.

The modified version of the *Combined* Approach improves on the original version by computing the average distance preservation of a combination of attributes as the average of the **actual** distance preservations of each attribute. If x is an attribute of a dataset D, the actual distance preservation of x is computed as:

$$g_x = \frac{\sum_{u=1}^{n} \sum_{v=u+1}^{n} \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2}}{n_r}$$

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 n_r in this equation is the number of pairs of rows of D computed as:

$$n_r = {}^nC_2 = \frac{n(n-1)}{2}.$$

Thus, for any combination of attributes C of D, the average distance preservation of C is given as:

$$g_C = \frac{\sum_{x \in C} g_x}{n_C}$$

where n_C is the number of attributes in C. Therefore, to reduce a dataset D from p columns to k columns, the modified version of the *Combined* Approach finds the combination C of k attributes of D whose value of g_C is maximum.

2.9 The Modified *Direct* Approach

Like the *Direct Approach*, to reduce a dataset D_{nxp} to a dataset containing k columns, the modified version of the *Direct Approach* selects the combination of k attributes which best preserve the interpoint distances, and reduces the original dataset to a dataset containing only those k attributes. To do so, it first generates all possible combinations of k attributes from the original p attributes. However, for each combination, *C,* instead of estimating its average distance preservation using its $g_c$mid value, it computes the **actual** average distance preservation of C using the following formula:

$$g_C = \frac{\sum_{u=1}^{n} \sum_{v=u+1}^{n} \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2}}{n_r}$$

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 D reduced to the attributes of C. Once again, the term n_r in this equation is the number of pairs of rows of D computed as:

$$n_r = {}^{n}C_2 = \frac{n(n-1)}{2}.$$

Therefore, to reduce a dataset D from p columns to k columns, the modified version of the *Direct Approach* finds the combination C of k attributes of D whose value of g_C is maximum.

2.10 The Modified New Random Approach

This technique is suggested as an improvement of the *New Random Approach* discussed in Section 2.7 above. To reduce a dataset D_{nxp} from p attributes to k attributes using the modified version of the New Random Approach, we use the algorithm below. Clearly, the idea here is to generate a result which is less random (and thus more efficient) than the result of the *New Random Approach*. Note that m in the algorithm is the number of times the execution of the *New Random Approach* is repeated.

Algorithm

M = []
for i = 1 to m do

- Run the New Random Approach to generate k numbers at random in the range *1..p*
- Store the list of numbers generated as the i^{th} row of M

end

Generate the one-dimensional matrix M1 with p entries such that M1[p] holds the frequency of the number p in the matrix M

Finally, generate the matrix *Result* which contains the k entries in M of highest frequency, arranged in ascending order

Thus, if D is the original dataset, the result of reducing D using the modified version of the *New Random Approach* is given by:

$D_R = D(:, Result)$

Below is the result of a sample run of the program with D as given in Table 1 below (and with m = 4):

i) After the first run of *NRA:*

M = [9 2 3 7 5 1 10]

ii) After the second run of *NRA:*

$$M = \begin{bmatrix} 9 & 2 & 3 & 7 & 5 & 1 & 10 \\ 6 & 4 & 5 & 3 & 2 & 8 & 7 \end{bmatrix}$$

75	0	190	80	91	193	371	174	121	-16
56	1	165	64	81	174	401	149	39	25
54	7	172	95	138	163	386	185	102	96
55	14	175	94	100	202	380	179	143	28
82	21	197	87	88	181	360	177	103	-9
13	28	169	51	107	167	321	181	91	107
40	8	160	52	77	129	377	133	77	77
49	15	162	54	78	0	376	157	70	67
44	35	168	56	84	118	354	160	63	61
50	22	167	67	89	130	383	156	73	85
62	42	170	72	102	135	408	163	83	72
45	29	179	86	98	143	373	150	65	12
61	36	186	58	85	155	382	170	81	-24
30	49	177	73	105	180	355	164	104	68
51	43	174	88	112	158	399	184	94	46
47	50	150	48	75	132	350	169	72	36
68	56	171	59	82	145	347	176	61	84
46	57	158	65	70	120	353	122	52	57
73	63	193	63	119	154	392	175	90	73
57	64	166	79	96	188	406	158	79	-12
28	71	181	93	83	251	390	189	183	50
52	70	176	74	90	122	336	191	78	81
36	78	153	75	71	139	364	183	82	62

Table 1: The *D* Dataset

iii) After the third run of *NRA:*

$$M = \begin{bmatrix} 9 & 2 & 3 & 7 & 5 & 1 & 10 \\ 6 & 4 & 5 & 3 & 2 & 8 & 7 \\ 1 & 2 & 9 & 4 & 8 & 5 & 3 \end{bmatrix}$$

iv) After the fourth run of *NRA*

$$M = \begin{bmatrix} 9 & 2 & 3 & 7 & 5 & 1 & 10 \\ 6 & 4 & 5 & 3 & 2 & 8 & 7 \\ 1 & 2 & 9 & 8 & 4 & 5 & 3 \\ 2 & 5 & 6 & 8 & 1 & 7 & 10 \end{bmatrix}$$

Thus: M1 = [3 4 3 2 4 2 3 3 2 2] and

Result = [1 2 3 4 5 7 8]

Thus the result of reducing the dataset D using the modified version of the *New Random Approach* is the dataset D_R is given in Table 2 below.

75	0	190	80	91	371	174
56	1	165	64	81	401	149
54	7	172	95	138	386	185
55	14	175	94	100	380	179
82	21	197	87	88	360	177
13	28	169	51	107	321	181
40	8	160	52	77	377	133
49	15	162	54	78	376	157
44	35	168	56	84	354	160
50	22	167	67	89	383	156
62	42	170	72	102	408	163
45	29	179	86	98	373	150
61	36	186	58	85	382	170
30	49	177	73	105	355	164
51	43	174	88	112	399	184
47	50	150	48	75	350	169
68	56	171	59	82	347	176
46	57	158	65	70	353	122
73	63	193	63	119	392	175
57	64	166	79	96	406	158
28	71	181	93	83	390	189
52	70	176	74	90	336	191
36	78	153	75	71	364	183

Table 2: The D_R Dataset

3.0 Reducing Images Using These Techniques

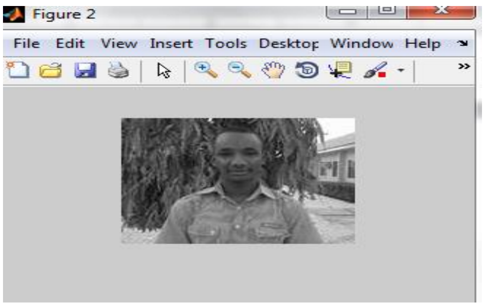
In this section, we shall use each of the techniques examined in Section 2 above to reduce images, and the effects of each reduction will be presented.

To achieve this aim, we shall make use of the MATLAB functions *imread* which converts an image into a matrix, and *imshow* which converts a matrix representing pixel brightness values into the image. (In other words, the function of *imread* is the reverse of the function of *imshow*.)

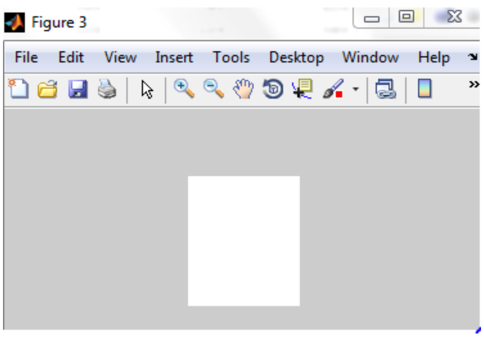
3.1 Random Projection

Random Projection is useless in preserving images. The simple reason is that when we multiply the matrix representing an image by a random matrix R, the resulting matrix typically has values outside the range of pixel brightness values. In our experiment, this is the result we obtained:

Original Image:



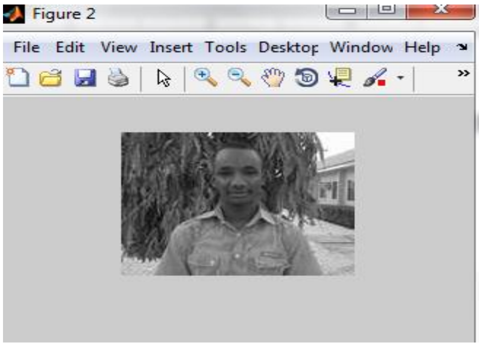
Reduced Image:



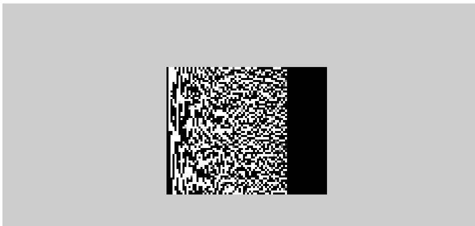
3.2 Principal Component Analysis (PCA)

Like RP, PCA is useless in preserving images. The following is the result obtained when we tried to reduce the result obtained using PCA:

Original Image:



Reduced Image:



3.3 Variance

Apart from RP and PCA, all the other methods we implemented were reasonably efficient in preserving images. With the *Variance* method, the results obtained are displayed below:

Original Image:

Reduced Image:

3.4 *Combined* Approach

Original Image:

Reduced Image:

3.5 *Direct* Approach

Original Image:

Reduced Image:

3.6 The Modified *Combined* Approach

Original Image:

Reduced Image:

3.7 The Modified *Direct* Approach

Original Image:

Reduced Image:

3.8 LSA-Transform

Original Image:

Reduced Image:

3.9 The New Random Approach

Original Image:

Reduced Image:

3.10 New Random Approach (Modified Version)

Original Image:

Reduced Image:

Remark:

As can be observed from the results above, some reduction methods (such as the *Combined* and *Direct* Approaches and their modified versions) maintain the sizes of the original image while others do not.

4.0 Comparisons Between the Different Dimensionality Reduction Techniques

As mentioned above, there are two types of reduction techniques: those in which each attribute of the reduced set is a linear combination of the attributes in the original data set; and those which reduce a dataset to a subset of the original attribute set**.** Of the ten techniques implemented in this paper, two of them (RP and PCA) belong to the first category, and as we have seen, they are both useless in preserving images. This applies to almost every technique in this category. Two exceptions in this regard include *Two Dimensional PCA* and *Discrete Cosine Transform* (Nsang 2011). All the other eight techniques implemented in this paper belong to the second category, and as we have seen, they are all efficient in preserving images.

Of these eight techniques, as mentioned above, LSA-Transform is probably the best in preserving images. Apart from the fact that the quality of the reduced image is practically the same as the quality of the original image, its speed of execution is very high. All the other seven techniques also significantly maintain the quality of the original image especially when most of the attributes of the matrix representing the original image are maintained – for instance when the number of attributes of the matrix representing the reduced image is at least 90% of the number of attributes of the matrix representing the original image. However, if we reduce the matrix representing the original image to 60% say (as in this case in this paper), as we can see, some of these methods are more efficient than others in preserving the original image. From the best to worst (as we can see from the results above), we have the *New Random Approach*, the *Modified New Random Approach*, and the *Variance Approach* followed by the *Direct* and *Combined* Approaches and their modified versions. Interestingly, these last four approaches are also the least time efficient. As a matter of fact, these last four approaches could take many days to run!

Because the *Combined* and *Direct* approaches and their modified versions have high run-time complexities, they are only suitable for reducing small images. The *Variance* approach on the other hand has the lowest run-time complexity (apart from *LSA-Transform*, of course), which makes it the most suitable for reducing large images. Obviously, the *New Random Approach* and its modified version are also suitable for reducing large images.

5.0 Conclusion and Future Work

In this paper, we have studied two categories of dimensionality reduction techniques: those in which each attribute in the reduced set is a linear combination of the attributes in the original set, and those which reduce a data set to a proper subset of the original attribute set. As we have realized, while most of the techniques in the first category are useless in preserving images, every technique in the second category can be used to preserve images. Image preservation is a very important application of dimensionality reduction as it enables us to conserve memory and to improve the speed of execution of any programs which use these images.

We also noticed that while *LSA-Transform*, the *New Random Approach*, the *Modified New Random Approach*, and the *Variance Approach* have low run-time complexities and can be used to reduce large images, the *Direct* and *Combined* Approaches and their modified versions have very large run-time complexities and can only be used to reduce small images.

We shall compare the extents to which each method discussed in this paper preserves the interpoint distances and *k-means* clustering of different datasets in another work. We shall also analyze the time complexity of each of the ten techniques implemented in this paper. We could not do these in this paper due to time and space constraints.

References

Nsang A., Ralescu A. 2010, 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), South Bend, IN.,* 70 - 77.

Nsang A., Ralescu A. 2009, A Review of Dimensionality Reduction and Their Applications. In *Proceedings of the Twentieth Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2009), Fort Wayne, IN.,* 118 - 123.

Nsang, A. 2011. *Novel Approaches to Dimensionality Reduction and Applications, An Empirical Study*. Lambert Academic Publishing.

Bingham E., Manilla H. 2001, Random Projections in Dimensionality Reduction: Applications to Image and Text Data. In *Conference on Knowledge Discovery Discovery in Data, Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 245-250.

Johnson W. B., Lindenstrauss J., 1984. Extensions of Lipshitz mapping into Hilbert Space. *Contemporary Mathematics.*

Nsang A., Ralescu A. 2010. More 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), South Bend, IN.*, 109-116.

Nsang A., Maikori A., Oguntoyinbo F., Yusuf H. 2015. A New Random Approach to Dimensionality Reduction, Unpublished.