IEC
Institute of Electronics and Computer

Journal of Artificial Intelligence and Systems, 2020, 2, 191–206
https://iecscience.org/journals/AIS
ISSN Online: 2642–2859

A Novel Data Instance Reduction Technique using Linear Feature Reduction

Arpita Joshi[1,2,*], **Nurit Haspel**[2]

[1]Department of Computer Science, Purdue University Northwest, Hammond, IN, USA
Email: arpitamhow@gmail.com/joshi137@purdue.edu
[2]Department of Computer Science, University of Massachusetts, Boston, MA, USA
Email: nurit.haspel@umb.edu
*Corresponding Author: Arpita Joshi, Email: arpitamhow@gmail.com

How to cite this paper: Arpita Joshi and Nurit Haspel (2020). A Novel Data Instance Reduction Technique using Linear Feature Reduction. Journal of Artificial Intelligence and Systems, 2, 191–206.
https://doi.org/10.33969/AIS.2020.21012

Received: June 16, 2020
Accepted: June 30, 2020
Published: July 2, 2020

Abstract

Representation of structurally significant data is indispensable to modern research. The need for dimensionality reduction finds its foray in varied genres viz-a-viz, Structural Bioinformatics, Machine Learning, Robotics, Artificial Intelligence, to name a few. The number of points required to effectively capture the essence of a structure is an intuitive decision. Feature reduction methods like Principal Component Analysis (PCA) have already been explored and proven to be an aid in classification and regression. In this work we present a novel approach that first performs PCA on a data set for reduction of features and then attempts to reduce the number of points itself to get rid of the points that have nothing or very little new to offer. The algorithm was tested on various kinds of data (points representing a spiral, protein coordinates, the Iris dataset prevalent in Machine Learning, face image) and the results agree with the quantitative tests applied. In each case, it turns out that a lot of data instances need not be stored to make any kind of decision. Matlab and R simulations were used to assess the structures with reduced data points. The time complexity of the algorithm is linear in the degrees of freedom of the data if the data is in a natural order.

Keywords

Dimensionality Reduction, PCA, Protein Conformations, Non-linear feature reduction, Data instance reduction

1. Introduction

Dimensionality reduction is a process of expressing high-dimensional data using a low-dimensional representation, while trying to preserve the variance in the data as much as possible. Principal Component Analysis is now one of the age-old and well-established methods for feature reduction [1]. It takes as input a data matrix with M observations and N features. The algorithm performs an orthogonal linear transformation and produces the *principal components*, which are the eigen-vectors of the covariance matrix of the original data that in decreasing order represent the variance captured by them. The components are obtained from the Singular Value Decomposition of the data matrix. More details of the

core algorithm can be found in [2, 3].

There are many approaches to obtaining these principal components, some of which are categorized as *robust*. Such algorithms, take into consideration the sparseness and possible corruption in the values of the data matrix [4]. We use one such approach called the *spherical* PCA. The data points in this approach are projected onto a sphere. The sphere is centered around the biggest cluster in the dataset and the radius is so chosen that most data points are covered. It offers a clever way of dealing with the outliers inherently present in the data. A single outlier affects the average of the data tremendously and ultimately the direction of principal components [5].

Real world structurally significant data is often so huge that its processing and analysis is long and tedious. Such datasets are also intensive on the system's memory. Also, in higher dimensions, the data becomes sparse. Consider the ratio of the areas of a square of length r and that of a circle of radius r, which is about 0.3183. Now consider the ratio of the volume of a cube and the volume of a sphere in three dimensions, the ratio now becomes approximately 0.2387. This ratio continues to decrease as the dimensions increase, which indicates that in high dimensional data the significant information moves towards the boundaries. The notion is referred to as the *curse of dimensionality*. Therefore, there arises a need for an algorithm that would make a decision as to whether a particular data instance in high dimensionality contributes to the shape of a structure or not. In particular, the motivation for this work was forked off because of dealing with humongous molecular data sets[6, 7]. These data sets are often in the form of text files of the order of 25 MB. Each row of these files represents a conformation that the molecule can have. Each conformation is represented by a set of attributes. These attributes can be normalized and reduced using various feature reduction techniques like PCA itself. Studies have shown that not all of these conformations need to be worked with to understand macro-molecular dynamics [8–12]. Our algorithm reduces the number of these conformations in a way that reduces the vast conformation space of proteins.

1.1. Contribution of the Work

The goal of this work is to obtain a reliable method to reduce the number of observations for data sets while losing as little information as possible. It finds relevance in the fact that lesser points would need less storage space and also reduce computing times of algorithms used to analyze these data sets. PCA processing in this work is used so as to obtain three distinctive features of the data that among themselves preserve the most variance. The algorithm then assesses the information content offered by each point. The assessment is done by the relative positioning of the points in the space defined by these first three principal components. As the Abstract claims, the algorithm has been shown to produce noteworthy results in eclectic data sets. In image processing, the classical techniques of edge detection and image recognition are of paramount importance. The algorithm can be an aid for all of these procedures. Details of more widespread applications of image processing with efficient data reduction and its importance in the medical world can be found in [13] and [14].

1.2. Literature Survey

Existing algorithms for data instance reduction are broadly divided into, incremental, decremental, batch and mixed [15–19]. The incremental algorithms begin with a null set and data instances are added to it depending on the result of the algorithm. The decremental algorithms, on the contrary, begin with the entire set of instances and depending on the

decision offered by the selection algorithms, instances are taken out from the set one had at the beginning. The batch algorithms function in a way that each instance is first analyzed and then a decision is made as to which ones to keep. Mixed algorithms begin with a preselected set of instances and the process then continues to figure whether instances should be deleted or added. The proposed algorithm falls into the decremental bracket.

An evaluation of the age-old techniques of instance reduction is explained in [15, 20]. Another work that performs an elaborate evaluation of existing algorithms and presents two ways based on Locality Sensitive Hashing [17] is presented in [16]. Another novel approach based on similarity calculation between instances and then clustering is presented in [18]. Another novel entropy based approach is presented in [19]. All of these algorithms are aimed to produce the best training set to produce accurate classification of a dataset. The algorithm proposed in this work explores a range of datasets and has been shown to work in each scenario. Comparative results and their analyses are provided in section 3.3.

2. Methods

As mentioned earlier, the first step is to perform spherical PCA on input data matrix. The data matrix could be a distance metric for all the data instances or a combination of attributes that measure the similarity of each data instance. If the data has non-numerical information, then that information needs to be translated into numerical data. Computing spherical principal components is a well known algorithm and we followed the procedure described in [4, 5, 21]. We wrote a short Matlab script for the purpose, it takes as input the data matrix (with only numeric features) and the number of dimensions (principal components) desired of all the data instances. The output is a matrix file that retains all the points with the number of desired principal components. We work with two and three dimensions, in order to better visualize structurally significant data. It is observed that over eighty percent of the variance is explained by the first three principal components in all kinds of data the algorithm was tested on, although it is not always the case. The method can be readily extended to higher dimensions. PCA redoes the features of the data set and produces a new set in decreasing order of variance captured by each of them. The relative values of these high variance capturing principal components is used to pick the relevant data instances. The process of collecting data for any experiment brings about a natural order among the instances. The proposed algorithm expects such order. If no such order exists, or there is very little information available about the dataset, a sort would need to be performed based on the first principal component.

2.1. General Outline

The data set at the beginning has N rows and M columns. The PCA processed data with N rows and three columns (first three principal components is fed to the algorithm). For a blind dataset (or a data set of known random nature), a sort should be performed based on the first principal component so that the points can be processed in order. Following is the pseudo-code that describes our approach:

Input: a numerical matrix of N data points

```
1. //2-D case: data set has N points
2. for i=2:N-2 {
3.     a = ratio of slope between i-1, i and i,i+1
4.     if(a < threshold)
5.        remove point i
6. }
```

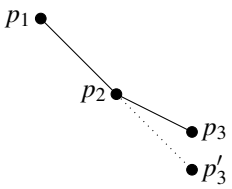


Figure 1. *Elucidation of the Algorithm. The middle point would be kept or rejected depending on where it lies with respect to the other two.*

```
7. //3-D case: the data set now has Q points (Q<N)
8. for q=2:Q-4 {
9.     P1 = position vector of point q-1
10.    P2 = position vector of point q
11.    P3 = position vector of point q+1
12.    P4 = position vector of point q+2
13.    N1 = cross product(P1-P2, P1-P3)
14.    b = angle between N1 and P4
15.    if(b<pi/5 || b>3*pi/5)
16.       remove point q+1
17. }
```

Output: a numerical matrix with a no. of data points that is <<< N

2.1.1. Projection Score

We seek an optimum value of the slope threshold (in line-4 of the algorithm above) and the angle between planes (line 14). Their optimality is decided by the what is called the *projection score* of a dataset. It implies how much a given point contributed to the variance of the entire data set. A mathematical formulation of a proof of correctness has been adapted from a method described in [22]. This work formulates a way to credit the informativeness of a variable in huge data sets. We use this formulation on data instances instead of features. For this, first the matrix is multiplied by its transpose to obtain an empirical covariance matrix. Let A be the data matrix of dimensions $N \times M$. The covariance matrix of dimension $N \times N$ is:

$$Cov(A) = A * A^T \tag{1}$$

Next, the eigen-values of the covariance matrix are computed. Let λ_x denote the eigen-value in x^{th} dimension. The ratio of the sum of these values in rejected data points to the total number of points is referred to as the projection score of these data instances. Let S denote the set of points eliminated by the algorithm, projection score α can be denoted by the following equation.

$$\alpha = \frac{\sum_{x \in S} \lambda_x}{\sum_{x=1}^{N} \lambda_x} \tag{2}$$

The lower this number is, the lower is the information content offered by this set of points. This number is found to be of the order of 10^{-3} at the very least in almost all data sets. The Projection Score column of **Table-1** shows these results.

2.2. 2D case

To decide whether a point can be removed from a projection, we measure how much information it adds to it. Intuitively, if three close by points are co-linear, the middle point

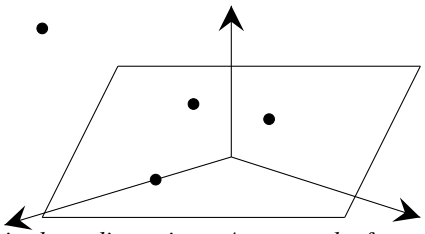


Figure 2. *The Algorithm in three dimensions. Amongst the four co-planar points the one encountered third would be rejected or kept depending on how much the plane is to be tilted in order to include the point that is out of the plane.*

does not add new information to the projection. To obtain information about these points, the first two principal components are traversed in the order in which they exist in the data file and the information content offered by each point is assessed as follows: Given three points p_1, p_2, p_3 at coordinates $(x_1, y_1), (x_2, y_2)$ and (x_3, y_3), respectively, then the threshold is defined as the ratio of the slopes of the lines $p_1 - p_2$ and $p_2 - p_3$:

$$\frac{\frac{y_2-y_1}{x_2-x_1}}{\frac{y_3-y_2}{x_3-x_2}} = \frac{y_2-y_1}{x_2-x_1} \times \frac{x_3-x_2}{y_3-y_2} \tag{3}$$

If this change of slope is above a threshold, the middle point is kept, otherwise it is discarded. **Figure-1** illustrates an example of three points. The middle point is the one being investigated. The method checks for collinearity. The ratio of slopes that constitutes collinearity depends on the kind of data. To find an optimal value we conducted a binary search starting from a given threshold, as detailed below. The threshold values for the various data sets are reported in the last column of **Table 1**. This value of the threshold maintains much of the variance in the data, while rejecting the most number of points. The process is explained in detail in Section 2.4. Decreasing this number any further may eliminate more points from the set but loses too much of its variance, according to the score described below.

2.3. 3D case

A straightforward extension of this algorithm can be applied to three dimensions, where we seek to eliminate points that do not contribute to the projection by testing whether they are *co-planar* with three other close-by points. Since every three points define a plane, a fourth point on the plane does not contribute to the projection. The algorithm is performed on the reduced data set received from the two dimensional projection. As before, the points again are traversed in the order they appear in the dataset (or the sorted order based off of first principal component if the data was know to be random). In a sequence of four points, the threshold (the definition of threshold here is the same as in 2-D case) of angle required to tilt the plane on which the first three points lie helps in deciding whether the fourth point in this sequence is informative or not. If this change of angle is above a threshold, the fourth point is retained, otherwise it is rejected. The details of how this is achieved are in the next section. Figure-2 provides for visualization of this procedure in three dimensions.

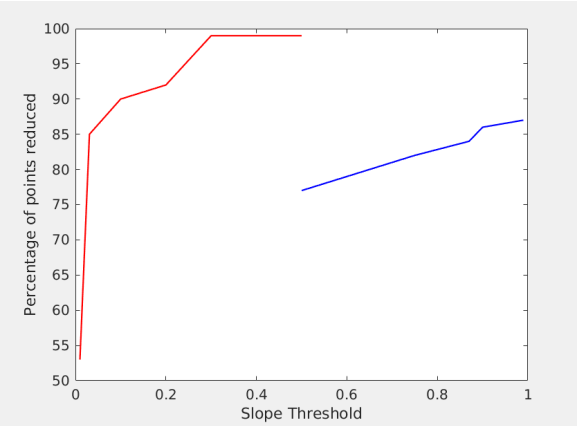
2.4. Determining the Threshold

As mentioned earlier, the *thresholding* here is paramount and depends on the kind of dataset being dealt with. For a new (blind) dataset, we start with an initial slope ratio value (for the 2-D case). Starting with 0.5 we assess the results based on the percentage of points eliminated and the projection score defined below. Assessing the results, there are three options, this threshold is either adequate, too high, or too low:

1. If the initial value 0.5 is too high, the reduced dataset will be too small. Per our definition, this means over 75% of the data is lost and the projection score is 10^{-2} or higher. A combination of the percentage of eliminated points and the projection score decide whether the subset obtained is acceptable or not. For example, in the Swiss Roll dataset, described in the following section, the percentage of points eliminated is at slope threshold of 0.5 was over 75%, but the projection score is much lower (**Table-1**, column 1) which meant that a slope threshold of 0.5 is not too hight this dataset. But for the datasets that do fall in this category, the next step would be to perform a binary search between 0 and 0.5, to obtain an optimal threshold. The next value of threshold to try would be 0.25, if the projection score is still high, try 0.125 and so on. The maximum number of trials in this study, 15 iterations, were needed for the face image dataset.
2. If the initial value 0.5 is too low, the dataset ends up retaining most of its points. By our standards, it happens if only 15% or fewer points of the original data are eliminated. In this case we perform a binary search between 0.5 and 1 until the convergence criterion mentioned above in the first case is achieved. All of the protein datasets in this study fall into this category, where (even though we began with a much lower threshold for experimentation purposes), the eliminated points were well above 60% and the projection scores as low as 10^{-5} were achieved.
3. The slope threshold of 0.5 is considered near adequate if the percentage of points eliminated are between 15 and 75. In this case, the threshold and the corresponding projection score should be considered for the values of slope threshold of 0.4 and 0.6.
    a) If the percentage of points decreased is less than 50 and the threshold of 0.6 results in an even smaller dataset while maintaining (or lowering) the projection score, we perform a binary search between 0.6 and 1. If not, the search should be between 0.5 and 0.6. As soon as the projection score becomes higher, stop and return the slope threshold for this dataset.
    b) If the percentage of points eliminated is over 50, we still try a higher slope threshold of 0.6 because the goal here is to be able to represent data with as fewer points as possible while capturing maximum variance. The spike in projection score is evident of the fact that informative point(s) have been removed from the dataset. Reducing the threshold to 0.4 results in a comparatively larger dataset while maintaining the projection score, we conduct a binary search between 0.4 and 0.5. If the projection score at 0.4 becomes lower, it is indicative of the fact that at the threshold of 0.5 there has been a loss of few informative points. The search should then be between 0 and 0.4. As soon as the projection score becomes higher, stop and return the slope threshold for this dataset.

    5-6 trials are usually enough for the purpose. Most machine learning datasets fall in this category.
    Figure-3 shows the process for the two extremes. In both the cases the stopping point for the threshold for the process is decided by the projection score at that point for the corresponding dataset.
4. As mentioned earlier, the next step is to try to eliminate even more points from the data set by progressing onto the 3-D version of the algorithm. The decision of rejecting a point, in this case, is determined by how much the plane created by the previous three points is to be tilted in order to accommodate the current point. The process of determining this *angle threshold* is the same as for the slope threshold. The angle that we begin with is $\pi/2$ (and $3 \times \pi/2$ to account for the negativeness of the angle). Depending on the size of the reduced data set and the projection score, a binary search can be started between π and $\pi/2$ or $\pi/2$ and 0.

Also, besides the projection score and the percentage of points eliminated, each dataset has its own method of evaluation that presents insight. The goal of *Supervised Learning* is to



;

Figure 3. *Slope Threshold versus the percentage of points reduced. In both the cases the stopping point for the threshold for the process is decided by the projection score at that point for the corresponding dataset.*

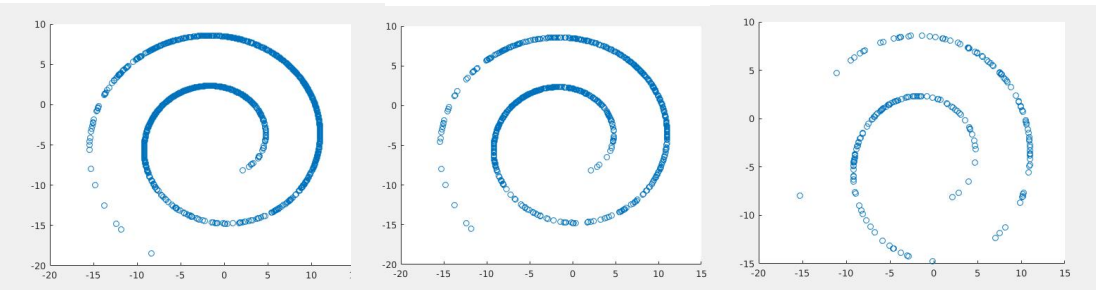


Figure 4. *2-D PCA projection of the Swiss Roll dataset with 1600 (left), 833 (middle) and 210 points (right).*

exploit the information known about the dataset. The datasets that represent a shape, like the Swiss Roll and Face Images, can be evaluated on the basis of their simulation as well. In the context of machine learning, the reduced dataset is treated as a *training set* to classify the rest of the data, details can be found in section 3.2.1.

3. Results and Discussion

3.1. Simulation Results

3.1.1. Swiss Roll Dataset

Figure-4 (left) shows the PCA projection of a data set representing a Swiss Roll [23, 24]. It is a two dimensional projection that uses 1,600 points to represent the structure. After applying just the 2-D version of the algorithm, about half of these points were found to be redundant. **Figure-4** (right) shows the plot of the PCA projection of the remaining points. Only 833 points were used to construct this plot. As mentioned earlier, this is one of the datasets that represent a shape. Apart from the projection score and the percentage of elimination of data instances, the simulation results are quite instrumental in determining the validity of a reduced subset of points. In search of validation, a very different kind of structurally significant data set was chosen next.

Table 1. Projection Score trends in various data sets

Data Set	Projection Score	% of points eliminated	Reconstruction Error	Slope Threshold
Swiss Roll	$3.204 * 10^{-4}$	86.875	0.006	0.99
Human Galanin	$2.159 * 10^{-5}$	68.954	1.562	0.76
CDC42	$1.42 * 10^{-3}$	69.875	1.022	0.78
Vasopressin	$1.893 * 10^{-5}$	74.302	0.358	0.82
Lena Image	$2.03 * 10^{-5}$	53.36	0.13	0.00001
Iris Dataset	$7.37 * 10^{-9}$	52.67	0.02	0.16
Isolet Dataset	$6.4 * 10^{-3}$	40.23	0.14	0.46

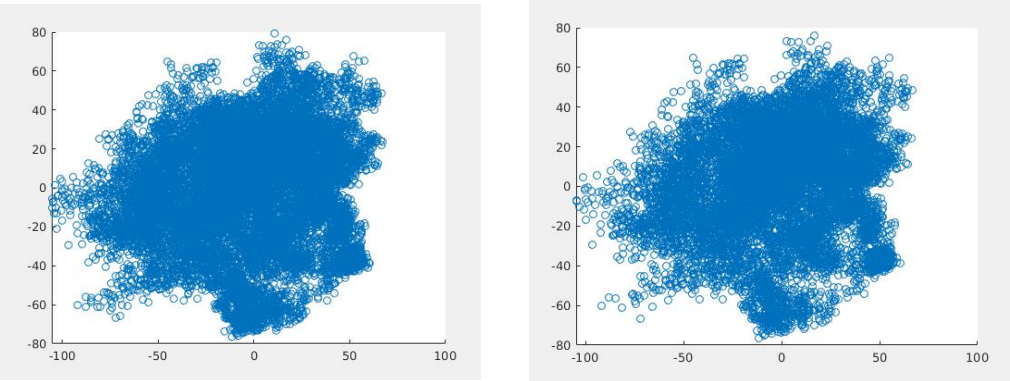


Figure 5. *2-D PCA projection of the Human Galanin dataset with 22750 (left) and 15680 (right) points respectively.*

3.1.2. Protein Datasets

We used simulations of protein structures. As mentioned in section 1, these molecular datasets are in the form of matrices, each row of which represents a conformation that the protein can attain in its trajectory. The data was created using Molecular Dynamics (MD) simulations [25]. If a molecule is composed of N atoms, it has $N \times 3$ attributes (columns), taking into account three-dimensional coordinates of each atom, that distinguish a conformation. The two structural extremes of the molecule are represented often with tens of thousands of conformations, making the data humongous. Human Galanin is one such example. It is a neuro-peptide and is a known heavy protein molecule with ample structural niceties. **Figure-5** (left) shows a two dimensional plot of the PCA projection of this macro-molecule with 22750 data points. The algorithm does away with about two-thirds of these points. **Figure-5** (right) is a PCA plot of the reduced data and was constructed with 15,680 points.

These results were obtained with just the application of the two-dimensional version of the algorithm. To do further tests, a set of medium to large protein molecules with well-established structures were chosen. **Figure-6** (left) shows the three dimensional plot of the PCA projection of one such protein, Cdc42, with 20,000 instances. Cdc42 is a protein from the Ras superfamily, involved in regulation of the cell cycle and has been shown to be involved in oncological processes [26]. When subjected to the two dimensional version of the algorithm, a data set of about 15,000 points was obtained. The number of points obtained subsequently by progressing to a plane was 4,985. **Figure-6** (right) shows a three-dimensional plot of these points. When the Swiss roll data set is again subjected to the progression of the algorithm, a set of 210 points was obtained. A two-dimensional plot of these points is shown in **Figure-4** (right). **Figures-7**- left and right show the same for Vasopressin, a hormone. These reduced forms of the protein datasets can be instrumental in isolation of protein conformations of interest and their trajectory simulation [27–29].

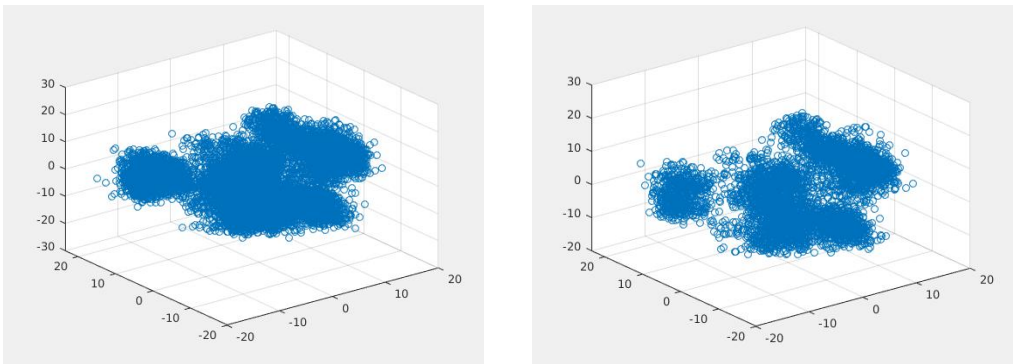


Figure 6. *3-D PCA projection of the CDC42 dataset with 20000 (left) and 4985 (right) points respectively.*

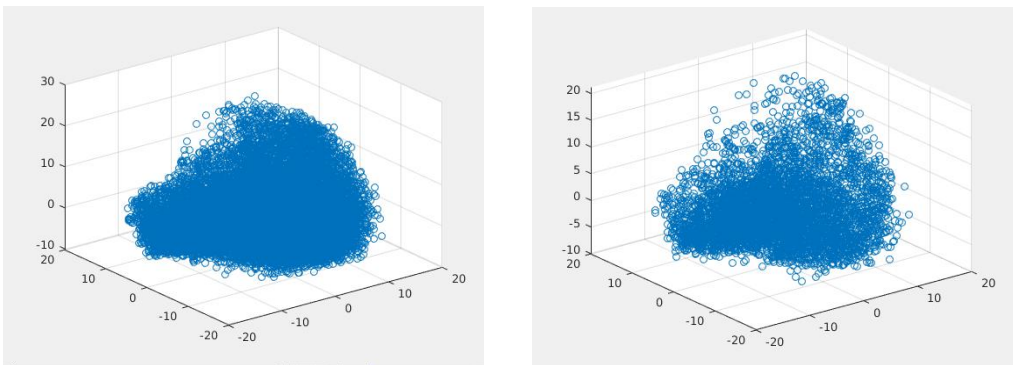


Figure 7. *3-D PCA projection of the Vasopressin dataset with 25000 (left) and 5320 (right) points respectively.*

3.1.3. Face Image

In light of the fact that the algorithm preserves a great deal of variance in structurally significant data, testing it on facial images seemed like a test that would consolidate its function. In image processing, efficient algorithms for de-noising an image are imperative. Many filtering processes exist for the purpose [30], [31]. The Lena image, shown in **Figure-8** was used as a test dataset here too. This image is pervasive and was procured form the Internet with a basic Google search. Its jpg file format was then converted into coordinate data using the method described in [32]. A two-dimensional projection of these points is shown in **Figure-9** (left). Once a coordinate matrix is obtained, it is PCA processed and the algorithm is performed as with other data sets. The reduced data set obtained here contained less than half the points in the original image and its plot is shown in **Figure-9** (right). In this data set, the algorithm discarded almost all the points (only three points were retained) for the values as low as 0.01. In this case the slope was decreased gradually until a considerable number of points were retained, which when plotted recreated the structure of the image. This number was 0.00001. Among the datasets used, this was the most complicated one, structurally. It had more nooks and crannies to cover in order to capture the variance and hence the slope threshold here is the lowest.

3.2. Quantitative Analysis

An advantage of using PCA processing presented itself in search of validation proofs. PCA is linear and the original data matrix can always be recreated. This fact was exploited to formulate a number termed the reconstruction error. First, the points eliminated by the


;

Figure 8. *The original Lena image. The standard test image widely used in the field of image processing since 1973.*

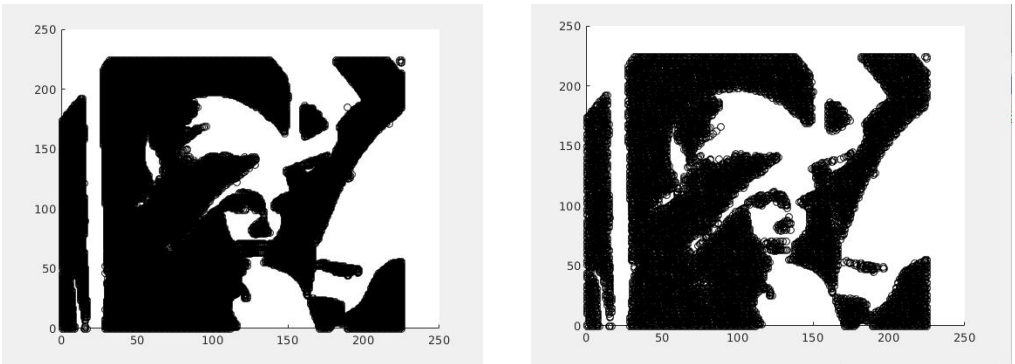


Figure 9. *2-D projection of Lena data set with 28,080 (left) and 13,097 (right) points respectively.*

algorithm are removed from the original data, this leaves the data matrix with as many points as the PCA projection returned by the algorithm (but all the original features/attributes). PCA projection of this smaller data set is then obtained. Then, the root mean square deviation (RMSD) of these two embeddings is calculated. This method first eliminates the translation component by shifting their center of mass to the same place, and then it finds the optimal rotation between the two sets using Singular Value Decomposition. The difference between the two structures is reported in the form of an error. Let the two matrices being compared be, A and B with dimensions $N \times M$. First, the centroid of the two is found, both the molecules are dragged to the origin by subtracting from each point the value of the centroid. The RMSD between the structures can then be calculated as under:

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\sum_{j=1}^{M}(a_{ij}-b_{ij})^2)} \tag{4}$$

Here, a_{ij} and b_{ij} are the corresponding elements of A and B respectively. The number returned by equation above indicates how similar the two structures are. It is enlisted for the various data sets in **Table 1** in the column named reconstruction error. Also, as mentioned earlier, in the data sets the algorithm was tested on, the first three principal components preserve over eighty percent of variance inherent in data. The PCA projection on the reduced data set follows these trends very closely. For example, in Cdc42, the first three principal

Table 2. Residual Variance trends in original data sets

Data Set	Variance in 1st PC[1]	Variance in 2nd PC	Variance in 3rd PC
Swiss Roll	55.55	27.87	16.59
Human Galanin	41.22	34.05	24.72
CDC42	42.082	34.35	23.569
Vasopressin	47.75	33.23	19.012
Lena Image	52.88	35.11	13.02
Iris Dataset	78.37	13.59	8.04
Isolet Dataset	57.13	31.06	11.82

Table 3. Residual Variance trends in reduced data sets

Data Set	Variance in 1st PC	Variance in 2nd PC	Variance in 3rd PC
Swiss Roll	63.44	27.30	9.26
Human Galanin	43.112	29.43	27.46
CDC42	49.436	28.34	22.22
Vasopressin	53.68	28.98	17.33
Lena Image	56.07	32.23	10.63
Iris dataset	86.65	10.35	3.00
Isolet Dataset	58.59	31.01	10.40

components capture respectively 42.082, 34.35 and 23.569 percent of variance. In the reduced form of Cdc42, the first three principal components capture 49.436, 28.34 and 22.22 percent of residual variance respectively. This trend is observed in all of the other data sets too. It is reported in **Table 2** for the original data sets and in **Table 3** for the reduced form of these data sets. If the corresponding principal components capture a similar amount of variance, this indicates that the algorithm retains those points that actually contribute to generating the variance values in the original data.

3.2.1. Machine Learning Datasets

The slope threshold defined in Methods and enlisted for all data sets in **Table-1** plays a crucial role in the analysis of datasets for classification and regression. These datasets had non-numerical attributes and so were pre-processed to either convert them to numbers or just get rid of them depending on what they represented. Subsequently, the rest of the algorithm was applied similarly as with other datasets. These were the only datasets that didn't represent a structure and unlike the protein datasets, a better understanding of the

[1]Principal Component

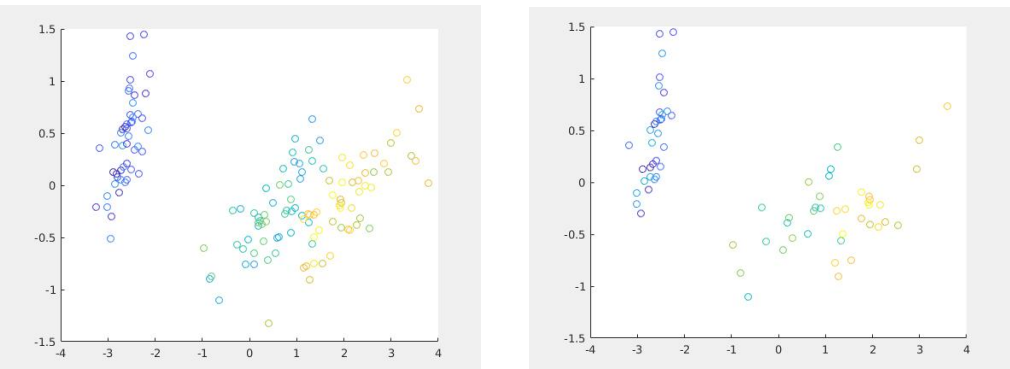


Figure 10. *Full and reduced Iris data set with 150 (left) and 71 (right) data instances respectively.*

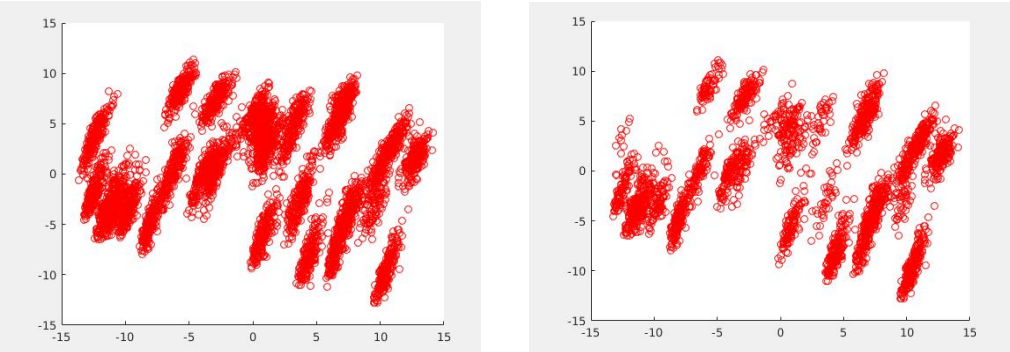


Figure 11. *Full and reduced Isolet data set with 6238 (left) and 3164 (right) data instances respectively.*

structure is not what was sought here. Nevertheless, like the protein datasets, a pictorial representation makes sense in terms of clusters of similar or close by points [33], after a projection of the original data is obtained (PCA, or any other feature reduction method for that matter). The first dataset used here was the Fisher's Iris dataset [34, 35]. It is a collection of three species of Iris flowers. It has 50 instances each of Iris Setosa, Iris Versicolour and Iris Virginica. Iris Setosa is the one that is linearly separable from the latter two. **Figure-10** (left) shows the PCA projection of the entire dataset. The instances differ on the basis of four attributes, namely, sepal length, sepal width, petal length and petal width, all in centimeters. Any value of the slope threshold here, as expected, gives two disjoint sets of data instances. **Table-1** reports this value for the Iris data set to be 0.16. The reduced data set so obtained is the smallest such set which when used as a *training set*, produces a 100 percent correct classification of Iris Setosa for the rest of the data set, which here is used as the *validation set*. **Figure-10** (right) shows the PCA projection of the reduced Iris data set. Perturbing the slope threshold further to include more data samples, as expected gives the same results. Increasing the slope threshold any further to obtain an even smaller training set mis-classifies a few data instances. A slope threshold of 0.17 produced a mis-classification of 1.92 percent. R simulations with five-fold cross validation using a Support Vector Machine with a linear kernel were used for verification purposes [36]. The reconstruction error in **Table-1** and the trends in residual variance of **Tables- 2 and 3** bear the same explanation as for the other datasets.

In order to assess the scalability of the algorithm, another multivariate dataset was chosen to test the abilities of extraction of a suitable training set that produces better classification. The Isolet dataset was generated for speech recognition [37]. It contains 52 samples of voice for 150 subjects. Each person uttered a letter of the English alphabet twice. The dataset, as on the Machine Learning group's website of University of California, Irvine,[38] is divided into five groups of 30. Four of these are used as the training dataset and the fifth one as the validation set. The attributes of this dataset are described in the paper [37]. They include spectral coefficients like contour features, sonorant features etc, all real numbers describing a sample of a voice of a subject. The goal of this work was to produce a 26 way classification, to identify which alphabet was spoken by a subject. It achieves over 95% accuracy. When this dataset is subjected to the algorithm described in Methods, it again produces a reduced dataset eliminating 40.23% of the instances. The PCA projections of the full and reduced forms are shown in **Figures-11** left and right respectively. Evident from the projections, the reduced dataset clearly produces the same clusters. This dataset when used as the training set also produces the same classification. The procedure was the same as described for the Iris dataset. The Isolet dataset has more groups to classify than the Iris and many of these groups (each one representing an alphabet of the English language) are

Table 4. Comparison of the proposed algorithm with other instance reduction algorithms.

Algorithm	Average Accuracy for classification	% of points reduced
Instance Based Learning Algorithms [15]	78.85	16.13
DROP-3 [15]	81.14	14.31
Entropy based algorithm[19]	≈ 85	88.72
LSH based algorithm [16]	≈ 90	≈ 60
Instance reduction Algorithm [18]	78.23	≈ 50
Proposed Algorithm	≥ 90	≈ 70

phonetically more similar than others, so the percentage of points eliminated here should be lesser than the Iris dataset because the data here is more diverse. This was found to be true, and the algorithm distinguishes between these two very similar, real-valued, multivariate datasets by a margin of 12.44% in context of data instance elimination.

3.3. Comparison with Other Instance Reduction Methods

As pointed out in section 1.2, all of the other data instance reduction methods have only analyzed the machine learning datasets. Therefore, their method of validation is using the reduced dataset to produce a classification and measure its accuracy. The average accuracy of classification and the percentage of points eliminated for these methods versus the proposed algorithm is in **Table-4**. The datasets used to produce results presented in **Table-4** are the same as on UC Irvine's website, including the Iris and Isolet datasets described in the previous section. We chose the multivariate ones that have default task assigned as '*Classification*' and have only numerical attribute types, like the Letter Recognition, Breast Cancer Detection and E.Coli (for prediction of localization sites for proteins) to name a few. The values for the proposed algorithm in the table have been evaluated to produce the most accurate classification. If saving space is the motivation, the slope threshold can be compromised with the accuracy to produce even smaller datasets.

With the exception of LSH, all of the algorithms in **Table-4** are linearithmic (NlogN) at best. LSH, like the proposed algorithm (provided a sort on the data is not needed), is linear, but works in a very different way. It depends on formulation of hash functions over the dataset that are sensitive to data instances. In other words, a family of hash functions represents the various categories a data instance can belong to. Each data point's compatibility with each of these functions is evaluated. One point representative of each category is chosen to be a part of the reduced data set. The point chosen is the first one that makes its way into a certain category. The proposed algorithm works better for the purpose as it is adherent to the dataset. It does a better job in evaluating the information content offered by a particular data point. Only one point per category is not always enough to represent a category in its entirety. How many and which of the points of a certain category should be chosen is the strength of the proposed algorithm. Also, formulation of a class of functions is a task in itself and not having to do so, betters the time complexity. The proposed algorithm also performs better when the data is highly sparse, or in other words the categories for classification are of the same order as the number of data instances available.

4. Conclusions and Future Work

The methods and results above manifest the validation of this novel algorithm. The algorithm not only successfully reduces the data points required to represent a structure, it does so in a naturally cogent way. More complex the structure, lesser are the number of points reduced. In other words, the algorithm adheres to the complexities inherent in the data. Table-1 justifies this assertion. The two dimensional projection of the swiss roll data set as expected is just a spiral and it is safe to assume that not too many points are required

to assert that these points indeed represent a spiral. So, as expected, the algorithm eliminates the largest number of points. Amongst the macro-molecule datasets, Vasopressin (in these examples) is known to be less structurally complex than Human Galanin and Ccd42 and so, comparatively, ends up with a smaller portion of data points. A dataset representing the complexities of a visage intuitively has more intricacies to take care of and hence the Lena image data set ends up with a large portion of its original dataset. The Isolet dataset has 26 defined categories and so a training set for the best possible classification here eliminates lesser points than the Iris dataset.

The assessment of the information content offered by a data instance lets one decide whether to continue storing the particular data instance or not. The two Machine Learning datasets were used to draw conjectures of this sort. This could be an aid in determining and making sure that minimum number of data instances are stored. It would help in deciding whether new prospective data would affect the subsequent analysis or not. This is especially useful in large and complex data sets that require a lot of storage. If numerous attributes identify an instance, being able to do so is a boon. How this algorithm (or a variant of it) can be used for non-linear classification forms the basis for future work.

What lies ahead is to narrow down a way to obtain accurate projections of larger and relevant data sets using minimum possible data instances. A way to parameterize the dissimilarity of data instances would pave the way for non-linear feature reduction and then ultimately data reduction. One of our primary goals is to reduce protein data sets and use just enough points to isolate intermediate protein conformations [39, 40, 27]. Their isolation would be key to characterization of their conformational landscape which would pave the way for understanding the convoluted relation between protein structure, dynamics and function.

Conflicts of Interest

We declare that this work was not carried out in the presence of any personal, professional or financial relationships that could potentially be construed as a conflict of interest.

References

[1] P. K., "On lines and planes of closest fit to systems of points in space." *Mag A*, pp. 59–572, 1901.

[2] M. Benito and D. Pena, "A fast approach for dimensionality reduction with image data." *Pattern recognition*, pp. 2400–2408, 2005.

[3] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philos Trans A Math Phys Eng Sci.*, vol. 374, 2016. [Online]. Available: www.ncbi.nlm.nih.gov/pmc/articles/PMC4792409/

[4] E. J. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58(3), 2011.

[5] N. Locantore, J. Marron, D. Simpson, N. Tripoli, J. Zhang, and K. Cohen, "Robust principal component analysis for functional data," *Socicdad de Estadistica e Investigacion Operativa Test*, vol. 8, 1999.

[6] P. Das, M. Moll, H. Stamati, L. Kavraki, and C. Clementi, "Low dimensional, free energy landscapes of protein folding reactions by nonlinear dimensionality reduction," *Proc. Nat. Acad. Sci.*, vol. 103, no. 26, pp. 9885–9890, 2006.

[7] A. Vajdi and N. Haspel, "A new dynamic programming algorithm for comparing gene expression data using geometric similarity," *IEEE International Conference on Bioinformatics and Biomedicine*, pp. 1157 – 1161, 2016.

[8] N. Haspel, M. Moll, M. Baker, W. Chiu, and L. E. Kavraki, "Tracing conformational changes in proteins," *BMC Structural Biology*, vol. Suppl1, p. S1, 2010.

[9] B. Raveh, A. Enosh, O. Furman-Schueler, and D. Halperin, "Rapid sampling of molecular motions with prior information constraints," *Plos Comp. Biol.*, vol. 5(2), p. e1000295, 2009.

[10] A. Shehu and B. Olson, "Guiding the search for native-like protein conformations with an ab-initio tree-based exploration," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1106–1127, 2010.

[11] I. Al-Bluwi, M. Vaisset, T. Siméon, and J. Cortés, "Modeling protein conformational transitions by a combination of coarse-grained normal mode analysis and robotics-inspired methods," *BMC structural biology*, vol. 13, no. Suppl 1, p. S2, 2013.

[12] A. Joshi and N. Haspel, "Clustering of protein conformations using parallelized dimensionality reduction," *Journal of Advances in Information Technology*, 2019.

[13] M. Greberman, "Image processing applications: An overview," *Proc Annu Symp Comput Appl Med Care*, vol. Nov 7, pp. 257–259, 1984. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2578700/

[14] A. Bovick, "Handbook of image and video processing," *Academic Press New York*, 2000.

[15] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine Learning*, 2000.

[16] A. A. Gonzalez, J.-F. Diez-Pastor, J. J. Rodriguez, and C. G. Osorio, "Instance selection of linear complexity for big data," *Knowledge Based Systems*, 2016.

[17] S. Garcia, J. Derrac, J. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *IEEE's Transactions on Pattern Analysis and Machine Intelligence*, pp. 417–435, 2012.

[18] I. Czarnowski and P. Jedrzejowicz, "Instance reduction approach to machine learning and multi-database mining," *Annales UMCS Informatica*, 2006.

[19] S.-H. Son and J.-Y. Kim, "Data reduction for instance based learning using entropy-based partitioning," in *ICCSA: International Conference on Computational Science and Its Applications*, 2006, pp. 590–599.

[20] A. Joshi, "High performance computing techniques to better understand protein conformational space," Ph.D. dissertation, University of Massachusetts, Boston, 2019.

[21] J. Fujiki and S. Akaho, "Spherical pca with euclideanization," *ACCV'07 Workshop Subspace*, November 2007.

[22] M. Fontes and C. Sonesan, "The projection score – an evaluation criterion for variable subset selection in pca visualization," *BMC Bioinformatics*, vol. 12, p. 307, 2011.

[23] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction." *Science*, pp. 2319–2323, 2000.

[24] V. D. Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," *Advances in neural information processing systems*, 2003.

[25] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, and K. Schulten, "Scalable molecular dynamics with namd," *Journal of computational chemistry*, vol. 26, no. 16, pp. 1781–1802, 2005.

[26] K. Wennerberg, K. L. Rossman, and C. J. Der, "The ras superfamily at a glance," *Journal of Cell Science*, vol. 118, no. 5, pp. 843–846, 2005. [Online]. Available: http://jcs.biologists.org/content/118/5/843

[27] A. Joshi, N. haspel, and E. Gonzalez, "Sampling of intermediate protein conformations using efficient dimensionality reduction and topological analysis of structures," *submitted to IEEE Transactions on Computational Biology and Bioinformatics*, 2020.

[28] D. Luo and N. Haspel, "Multi-resolution rigidity-based sampling of protein conformational paths," pp. 787–793, September 2013.

[29] A. Vajdi, A. Joshi, and N. Haspel, "Integrating co-evolutionary information in monte carlo based method for proteins trajectory simulation," in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, ser. BCB '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 598–603. [Online]. Available: https://doi.org/10.1145/3307339.3343867

[30] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson, 2018.

[31] A. Joshi, A. Boyat, and B. K. Joshi, "Impact of wavelet transform and median filtering on removal of salt and pepper noise in digital images," *Proc. of International Conference on Issues and Challenges in Intelligent Computing Techniques*, 2014.

[32] A. H. Ashtari, "Pic/plot images to coordinate points." 2015, national University of Malaysia, Center for Artificial Intelligence Technology (CAIT), Faculty Of Information Science and Technology.

[33] P.-A. Cazade, W. Zheng, D. Prada-Gracia, G. Berezovska, F. Rao, C. Clementi, and M. Meuwly, "A comparative analysis of clustering algorithms: O2 migration in truncated hemoglobin i from transition networks," *The Journal of Chemical Physics*, vol. 142, no. 2, pp. –, 2015. [Online]. Available: http://scitation.aip.org/content/aip/journal/jcp/142/2/10.1063/1.4904431

[34] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annual Eugenics 7, Part II*, 1936.

[35] B. Dasarathy, "Nosing around the neighborhood: A new system structure and classification rule for recognition in partially exposed environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 1, 67-71,*, 1980.

[36] D. Simovici, *Mathematical Analysis for Machine Learning and Data Mining*. World Scientific, 2018.

[37] M. Fanty and R. Cole, "Spoken letter recognition." *Advances in Neural Information Processing Systems*, 1991.

[38] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[39] S. Kirkpatrick, C. D. G. Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[40] R. Vetro, N. Haspel, and D. Simovici, "Characterizing intermediate conformations in protein conformational space," *Proc. of the Ninth International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, July 2012.