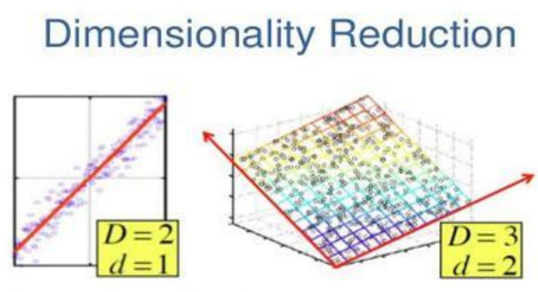
11/17/2020

PCA & SVD

NAME: N. I. MD. ASHAFUDDULA
STUDENT ID : 18204016
MSc. Student, DUET

INTRODUCTION:

Principal component analysis (PCA) & Singular value decomposition (SVD) both are widely popular feature reduction techniques. Feature reduction is the technique where we reduced input variables in such a way that could increase system performance. Suppose we have a dataset with 300 columns that means input variables so, our dataset has 300 dimensions. But to compute effectively and get an improved performance of our system we don't need all of those input variables we could reduce to feed our model with effective data and less input which increases the overall model performance. We use the terms Feature Reduction, Input variables, Columns in database interchangeably.



Fig[7]: Dimensionality Reduction

When we are dealing with high dimensional data, it is often useful to reduce the dimensionality by projecting the data to a lower dimensional subspace which captures the "essence" of the data.

Table 1: 5D input (Original data)

Record	F1	F2	F3	F4	F5
1	**1**	**1**	**1**	0	0
2	**2**	**2**	**2**	0	0
3	**3**	**3**	**3**	0	0
4	**4**	**4**	**4**	0	0
5	0	2	0	**4**	**4**
6	0	0	0	**5**	**5**
7	0	1	0	**2**	**2**

Dim. Reduction using PCA/ SVD

Table 2: 2D input (Reduced Dim)

Record	F1	F2
1	1.72	-0.22
2	5.15	-0.67
3	5.87	-0.89
4	8.58	-1.12
5	1.91	5.62
6	0.90	6.95
7	0.95	2.81

Uses of Feature Reductions:

- Fingerprint Recognition
- Recommendation System
- Image Classification
- Computer vision
- Dimension Reduction
- Analysis Input variables

MOTIVATION:

Dimensionality reduction helps our system to increase the system performance by compressing the data, removing redundant features, using less computer memory, speeding up the systems performance, visualizing data.

RECENT WORKS:

Paper [1] worked on the detection of Self-Care problems of children with physical and motor disability. They implanted several Machine Learning (ML) methods using PCA to analyze the performance of different ML methods for this dataset. They found the best result using KNN with PCA. Where they reduced the input vector from 205 to 53.

In paper [2] they used PCA to analyze the input feature to get better system performance in Likelihood Diabetes Prediction. Where no. of input variables were 15. They found the best result using CNN with PCA.

In paper [3] authors worked on improving the detection of Melanoma & Nevus skin cancer prediction using Deep Neural Network (DNN). They used a total of 2 datasets and got the best result using CNN with PCA.

In paper [4] authors reduced the dimension of image deep features using PCA. There the input dimension in their dataset was 3727 from where they reduced the dimension to 887 using PCA. Which produced better system performance.

In paper [4] authors were trying to analyze the Dimension Reduction techniques on Big Data. They reduced the dimension of the database from 36 to 26 using PCA. I found from their paper that the system shows better performance on Random Forrest classifiers using PCA for some specific kind of metrics.

In paper [6] authors used PCA of Multi-view Deep Representation to classify images.

Table 3: Recent works performances

Work	Original Input Features	Reduced Features	Method	Without Feature Reduction	With Feature Reduction
[1]	205	53	PCA, KNN	Accu: 81.43%	Accu: 84.29%
[2]	15	To Analyze	PCA, RF	-	Accu: 97.4%
[3]	2 Dataset	-	PCA, CNN	-	Accu: 96.8%
[4]	3727	887	PCA	Accu: 88.3%	Accu: 91.3%
[5]	36	26	PCA, RF	Acuu: 98.59%	Acuu: 98.3% (Performane of Other Metrices was better)
[6]	Multiple Image	-	PCA, Proposed	-	Accu: 85.33±0.47

PCA:

In PCA, we need to find a new set of transformed feature set which can explain the variance in a much better manner. The original 300 features are used to make linear combinations of the features in order to push all the variance in a few transformed features. These transformed features are called the Principal Components.

The Principal Components have now got nothing to do with the original features. We will get 300 principal components from 300 features. Now here comes the impact of PCA. The newly formed transformed feature set or the Principal Components will have the maximum variance explained in the first PC. The second PC will have the second highest variance and so on.

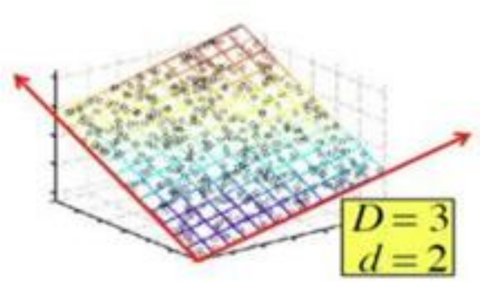
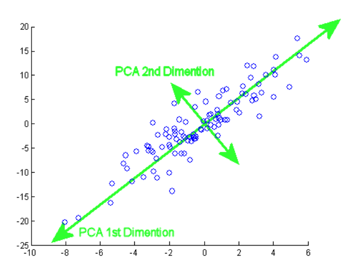


Fig [13]: PCA-1



Fig[13]: PCA-2

- **Variance:**

  Variance **(σ^2)** in statistics is a measurement of the spread between numbers in a data set. That is, it measures how far each number in the set is from the mean and therefore from every other number in the set.

  Formula:

$$\sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}$$

Where: x^i = the i^{th} data point

$\bar{x}$ = the mean of all data points

n = the number of data points

- **Eigenvector & Values:**

✓ Eigenvectors and values exist in pairs: every Eigenvector has a corresponding Eigenvalue.

✓ An Eigenvector is a direction, in the example the Eigenvector is the direction of the line (vertical, horizontal, 45 degrees etc.). An Eigenvalue is a number, telling us how much Variance there is in the data in that direction.

✓ In the example the Eigenvalue is a number telling us how spread out the data is on the line.

✓ The Eigenvector with the highest Eigenvalue is therefore the Principal Component.

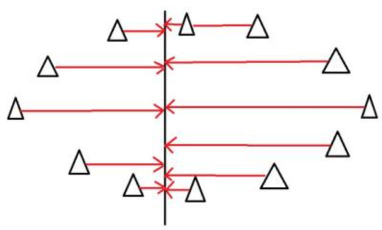


Fig [8]: Eigen vector

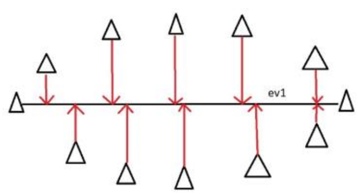
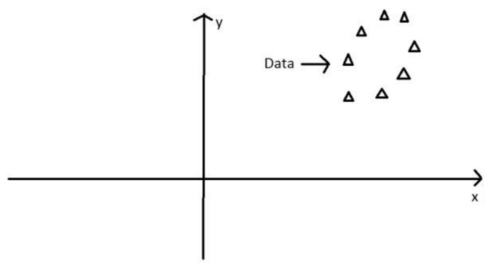


Fig [8]: Eigen vector

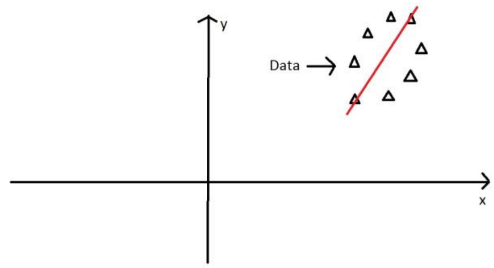
✓ If we take Age and Height of students, there are 2 variables, it's a 2-D data set, therefore there are 2 eigenvectors/values.

✓ Similarly If take Age, height, Class of students there are 3 variables, 3-D data set, so 3 eigenvectors/values.

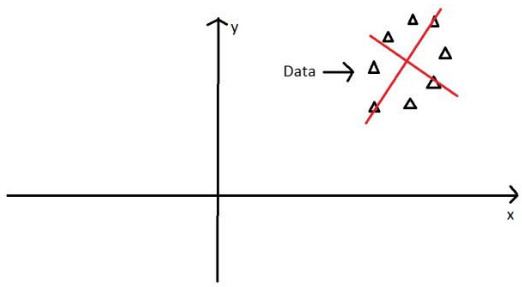
✓ The Eigenvectors have to be able to span the whole x-y area, in order to do this (most effectively), the two directions need to be orthogonal (i.e. 90 degrees) to one another.



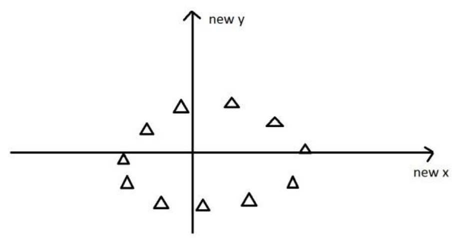
Fig[9]



Fig[9]



Fig[9]



Fig[9]

- ✓ The Eigenvectors gives us much more useful axis to frame the data in. So, We can now re-frame the data in these new dimensions.
- ✓ These directions are where there is most variation found, and that is where there is more information.
- ✓ Another way we can say, No variation of data means No information. In this case the Eigenvalue for that dimension would equal Zero, because there is no variations.

❑ **<u>Orthogonal [10]:</u>**

- ✓ Two vectors are Orthogonal if they are perpendicular to each other. (i.e. the dot product of the two vectors is 0)
- ✓ In Matrix, A square matrix with real numbers or elements is said to be an orthogonal matrix, if its transpose is equal to its inverse matrix or when the product of a square matrix and its transpose gives an identity matrix, then the square matrix is known as an orthogonal matrix.

  if, $A^T = A^{-1}$ is satisfied, then, $A\,A^T = I$

❑ **<u>Orthonormal [10]:</u>**

- ➢ A set of vectors is orthonormal if it is an orthogonal set having the property that every vector is a unit vector (a vector of magnitude 1).

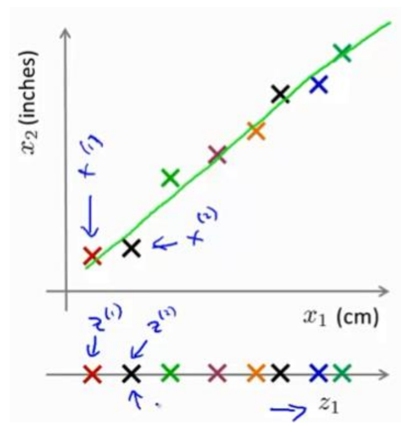
❑ Co-Variance & Correlation [11,12]:

- ✓ Both the terms measure the relationship and the dependency between two variables, suppose (x,y).
- ✓ "Covariance" indicates the direction of the linear relationship between variables.
- ✓ "Correlation" on the other hand measures both the strength and direction of the linear relationship between two variables.

PCA IMPLEMENTATION:

PCA stands for Principal Component Analysis. PCA finds the principal components of data. PCA finds the directions where there is the most variance, the directions where the data is most spread out.

1. Dimension Reduction tries to find a Line or Plane in which most of the data lies and project onto that line So that every projected data to the line is smallest.



Fig[7]: PCA 2D to 1D

2D to 1D

$X^1 \in R^2 \rightarrow Z^1 \in R^1$

$X^1 \in R^2 \rightarrow Z^1 \in R^1$

………………..

$X^m \in R^2 \rightarrow Z^m \in R^1$

$$Z^i = \begin{bmatrix} Z_1^i \end{bmatrix} = \begin{bmatrix} X^i \end{bmatrix}$$

3D to 2D

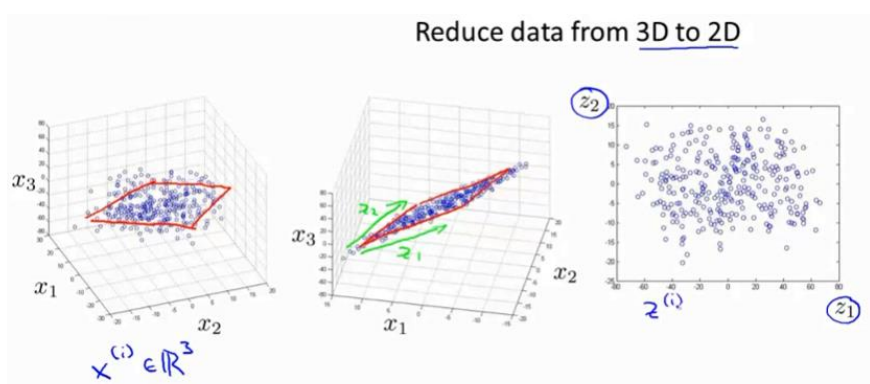
$X^1 \in R^3 \rightarrow Z^1 \in R^2$

$X^1 \in R^3 \rightarrow Z^1 \in R^2$

………………..

$X^m \in R^3 \rightarrow Z^m \in R^2$

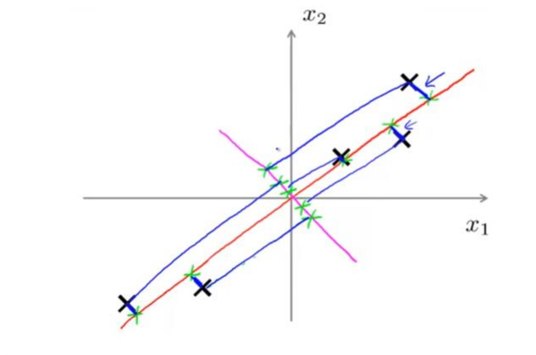
$$Z^i = \begin{bmatrix} Z_1^i \\ Z_2^i \end{bmatrix} = \begin{bmatrix} X^i \\ Y^i \end{bmatrix}$$



Fig[7]: PCA 3D to 2D

2. Wrong-Right projection line

- ✓ The projection of every point on the Magenta line (Fig: PCA (Wrong-Right projection line)) is huge and also greater than the Red line. So this is wrong line.



Fig[7]: PCA (Wrong-Right projection line)

3. Algorithm[7]:

1. Collect dataset

2. Preprocessing data (Feature Scaling/ Mean Normalization)

μ_j = (1/m)* $\sum_{i=1}^{m}(xij)$

Replace each x^i_j with (x_j - μ_j).

We scale data to have comparable range values. E.g if we have x_1 and x_2 features of data where,

x_1 = Size of house

x_2 = no. of bedroom

- ❑ Algorithm (Reduced N Dim to K Dim) [7]:

3. Compute "Co-variance matrix"

Sigma, Σ = (1/m)* $\sum_{i=1}^{n} xi\ (xi)T$ [Here, x^i (n*1) matrix]

In code, Sigma, Σ = (1/m) * X^T*X

4. Computer "Eigen Vectors" of matrix Sigma.

[U,S,V] = svd(Sigma) or U = eig(Sigma)

5. U is a (n*n) matrix. Where to reduce in K dim we need to choose K column from U matrix that is (n*k) matrix.

U_{reduce} = [:, 1:k]

6. Z_{new_space} = U^T_{reduce} * X_{input_set}

- ❑ **Algorithm** (Choose 'K' the no. of Principle Component) [7]:

PCA minimizes average projection Error such that,

Eq 1. Avg. projection Error = (1/m) * $\sum_{i=1}^{m}(x^{i} - x^{i}approx)^{2}$

Eq 2. Total variation in the data = (1/m) * $\sum_{i=1}^{m}(x^{i})2$

Now, we choose K to be smallest value so that,

$\frac{Eq.1}{Eq.2} \leq 0.01\ (1\%)$

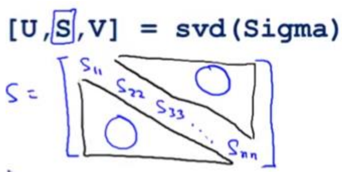
So, that we can maintain 99% variance or our desired variance

- **Algorithm (Choose 'K' the no. of Principle Component):**

  Previous method was not Efficient to choose 'K'

  Compute, [U,S,V] = svd(sigma)

  Where, S is a (r*r) diagonal matrix.



  Now, for given 'K' we need to check if, $\frac{\sum_{i=1}^{k} s^{ii}}{\sum_{i=1}^{n} s^{ii}} \geq 0.99\ (99\%\ variance\ retained)$

Let, **Input:**

Dataset=np.array(
[[1, 1, 1, 0, 0], [3, 3, 3, 0, 0], [4, 4, 4, 0, 0], [5, 5, 5, 0, 0], [0, 2, 0, 4, 4], [0, 0, 0, 5, 5], [0, 1, 0, 2, 2]])

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler() #Standardize features by removing the mean and scaling to unit variance
scaler.fit(Dataset)
scaled_data=scaler.transform(Dataset) #making input arrays -> Tranpose

from sklearn.decomposition import PCA
pca = PCA(n_components = 2) #Here code suggests 25 is good #SCADI-paper took 53 features
pca.fit(scaled_data)
x_pca=pca.transform(scaled_data) #making input arrays -> reverse Tranpose = original

print('PCA on Dataset(7*5):\n\n',x_pca)
_plot_data(x_pca)
```

Output: (Reduced Dim.)

[[0.06153582 1.4876476] [-1.41476273 0.32780438] [-2.152912 -0.25211723] [-2.89106128 -0.83203884] [2.01452609 -0.69985245] [2.9755685 -0.71530184] [1.40710559 0.68385838]]

Shape: (7, 2)

SVD:

SVD stands for Singular Value Decomposition. SVD is the specific way to reduce input features from a dataset. SVD is nothing more than decomposing vectors onto orthogonal axes.

- **Singular Value Decomposition:**

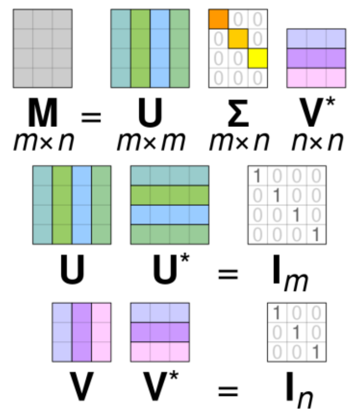
  $A_{m\times n} = U_{m\times m} \Sigma_{m\times n} V^T_{n\times n}$

  Where, $A_{m\times n}$ = input Matrix

  $U_{m\times m}$ = Orthogonal Matrix

  $\Sigma_{m\times n}$ = Diagonal Matrix

  $V^T_{n\times n}$ = Orthogonal Matrix



Fig[7]: SVD computation

- **Singular Value Decomposition Dimension reduced to 'k' Dimension:**

  $A_{m\times n} = U_{m\times m} \Sigma_{m\times n} V^T_{n\times n}$

  To find reduced dim of A matrix 'n' dimension to 'k' dimension,

  $Z_{reduced} = U[:,1:k] * \Sigma[1:k,1:k]$

SVD IMPLEMENTATION-1:

Input:**Dataset=np.array(**
[[1, 1, 1, 0, 0], [3, 3, 3, 0, 0], [4, 4, 4, 0, 0],[5, 5, 5, 0, 0], [0, 2, 0, 4, 4], [0, 0, 0, 5, 5], [0, 1, 0, 2, 2]])

```
from scipy.linalg import svd

n_elements = 2
U, S, V = svd(Dataset)
U = U[:,0:n_elements]
S = np.diag(S)
S = S[0:n_elements,0:n_elements]
rd = U.dot(S)
print('Original Data(7*5):\n\n',Dataset)
print('\nShort Hand SVD(7*2):\n\n',rd)
```

<u>Output:</u> (Reduced Dim.)

Short Hand SVD **<u>(7*2)</u>**<u>:</u>

[[-1.71737671 -0.22451218] [-5.15213013 -0.67353654] [-6.86950685 -0.89804872] [-8.58688356 -1.12256089] [-1.9067881 5.62055093] [-0.90133537 6.9537622] [-0.95339405 2.81027546]]

SVD IMPLEMENTATION-2:

```
from sklearn.decomposition import TruncatedSVD

#Dataset = A
# svd
svd = TruncatedSVD(n_components=2)
svd.fit(Dataset)
dataset_svd = svd.transform(Dataset)
print(dataset_svd.shape)
print(dataset_svd)

_plot_data(dataset_svd)
```

<u>Output:</u> (Reduced Dim.)

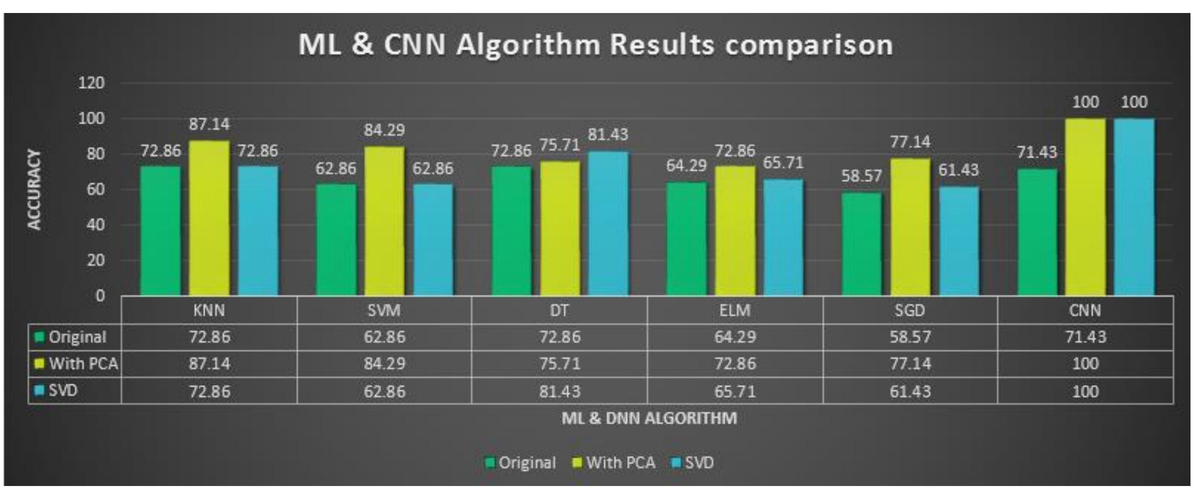
[[1.71737671 -0.22451218] [5.15213013 -0.67353654] [6.86950685 -0.89804872] [8.58688356 -1.12256089] [1.9067881 5.62055093] [0.90133537 6.9537622] [0.95339405 2.81027546]]

Shape: (7, 2)

RESULT COMPARISON of PCA & SVD WITH EXISTING WORKS:

From the table 4. We found for various machine learning algorithm using PCA and SVD, PCA performs better compare to SVD. Using feature reduction technique we choose only 15 features from 205 features in the dataset.

<u>Table 4: PCA and SVD performance comparison on SCADI[1] dataset.</u>



	KNN	SVM	DT	ELM	SGD	CNN
Original	72.86	62.86	72.86	64.29	58.57	71.43
With PCA	87.14	84.29	75.71	72.86	77.14	100
SVD	72.86	62.86	81.43	65.71	61.43	100

CONCLUSION:

As PCA and SVD performs better for specific kind of dataset we should careful to choose what feature reduction technique when to use. Feature selection task before applying PCA, SVD could serve us better.

REFERENCES:

[1] A Machine Learning Approach to Detect Self-Care Problems of Children with Physical and Motor Disability. (2018)

[2] Likelihood Prediction of Diabetes at Early Stage Using Data Mining Techniques. (2020)

[3] Improving detection of Melanoma and Naevus with deep neural networks. (2020)

[4] Dimension reduction of image deep feature using PCA. (2019)

[5] Analysis of Dimensionality Reduction Techniques on Big Data. (2020)

[6] Image Classification base on PCA of Multi-view Deep Representation. (2019)

[7] Coursera Machine Learning (PCA & Feature| Dimension Reduction)

[8] https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction

[9] https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/

[10] https://byjus.com/maths/orthogonalmatrix/#:~:text=If%20m%3Dn%2C%20which%20means,3%20rows%20and%203%20columns.

[11] https://towardsdatascience.com/let-us-understand-the-correlation-matrix-and-covariance-matrix-d42e6b643c22

[12] https://www.youtube.com/watch?v=xZ_z8KWkhXE

[13] https://dataconomy.com/2016/01/understanding-dimensionality-reduction/