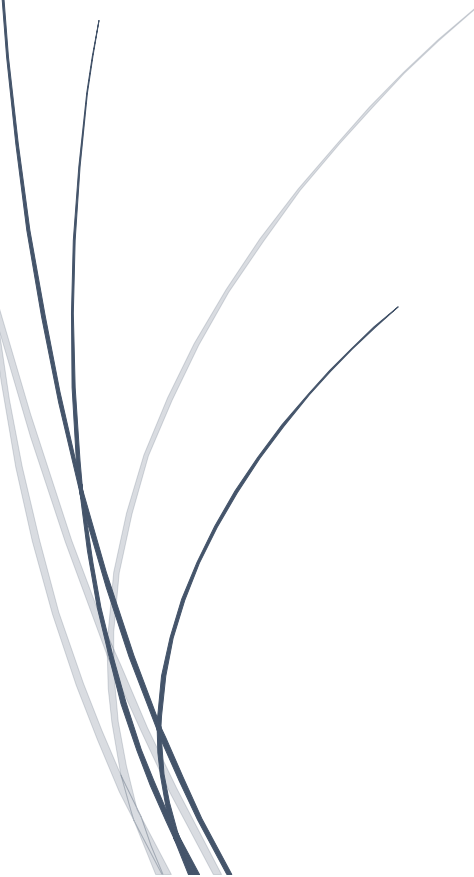


A dark blue vertical bar is on the left. A blue arrow points right from the bar, containing the date.

1/28/2021

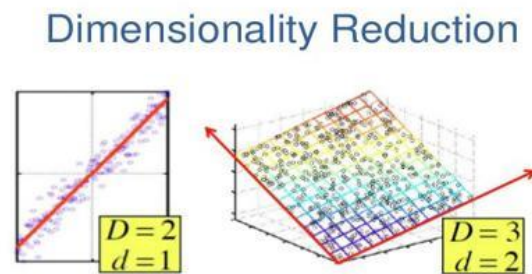
PCA & SVD - PROJECT

Several thin, curved lines in dark blue and light grey originate from the bottom left and curve upwards and to the right.

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

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.

PCA:

Principal component analysis (PCA). Linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space. The input data is centered but not scaled for each feature before applying the SVD. It uses the LAPACK implementation of the full SVD or a randomized truncated SVD by the method of Halko et al. 2009, depending on the shape of the input data and the number of components to extract.

METHOD DESCRIPTION:

PCA (n_components int, float or 'mle', default=None), Number of components to keep. If n_components is not set all components are kept.

In the code,

StandardScaler() method is used to standardize features by removing the mean and scaling to unit variance. Standardization of a dataset is a common requirement for many machine learning estimators, they might behave badly if the individual features do not more or less look like standard normally distributed data.

Scaler.fit() method computes the mean and std to be used for later scaling.

Scaler.transform() method performs standardization by centering and scaling.

pca.fit() method fits the model with scaled data.

pca.transform(scaled_data) applies dimensionality reduction to scaled data.

PCA IMPLEMENTATION:

```
1  from sklearn.preprocessing import StandardScaler
2
3  scaler = StandardScaler() #Standardize features by removing the mean and scaling to unit variance
4  scaler.fit(Dataset)
5  scaled_data=scaler.transform(Dataset) #making input arrays -> Tranpose
6
7  from sklearn.decomposition import PCA
8  pca = PCA(n_components = 2) #Here code suggests 25 is good #SCADI-paper took 53 features
9  pca.fit(scaled_data)
10 x_pca=pca.transform(scaled_data) #making input arrays -> reverse Tranpose = original
11
12 print('PCA on Dataset(7*5):\n\n',x_pca)
13 _plot_data(x_pca)
```

Output: (Reduced Dim.)

Output of this code contains shape (n, 2) as we chose n_component = 2. So k=15 dimensions is reduced to n=2 dimensions.

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.

METHOD DESCRIPTION:

n_elements = 2, is the number of features we want to keep and rest will be reduced named as feature reduction or dimension reduction.

We decomposed the dataset into **U, S, V matrix**. To compute the dimensionality reduction we used U and S matrix and n_elements.

SVD IMPLEMENTATION-1:

```
1  from scipy.linalg import svd
2
3  n_elements = 2
4  U, S, V = svd(Dataset)
5  U = U[:,0:n_elements]
6  S = np.diag(S)
7  S = S[0:n_elements,0:n_elements]
8  rd = U.dot(S)
9  print('Original Data(7*5):\n\n',Dataset)
10 print('\nShort Hand SVD(7*2):\n\n',rd)
```

OUTPUT:

Out of this code contains reduced dimension with shape (n,2). n= number of data item, n_elements = 2 or keeping features number = 2.

SVD IMPLEMENTATION-2:

Dimensionality reduction using truncated SVD. This transformer performs linear dimensionality reduction by means of truncated singular value decomposition (SVD). Contrary to PCA, this estimator does not center the data before computing the singular value decomposition. This means it can work with sparse matrices efficiently.

In particular, truncated SVD works on term count/tf-idf matrices as returned by the vectorizers in sklearn.feature_extraction.text. In that context, it is known as latent semantic analysis (LSA).

SVD suffers from a problem called “sign indeterminacy”, which means the sign of the components_ and the output from transform depend on the algorithm and random state. To work around this, fit instances of this class to data once, then keep the instance around to do transformations.

METHOD DESCRIPTIONS:

Parameters, `n_components`int, default=2. Desired dimensionality of output data. Must be strictly less than the number of features. The default value is useful for visualisation.

Svd.fit() method fits the dataset.

Svd.transform() method performs dimensionality reduction on dataset.

```
1  from sklearn.decomposition import TruncatedSVD
2
3  #Dataset = A
4  # svd
5  svd = TruncatedSVD(n_components=2)
6  svd.fit(Dataset)
7  dataset_svd = svd.transform(Dataset)
8  print(dataset_svd.shape)
9  print(dataset_svd)
10
11 _plot_data(dataset_svd)
```

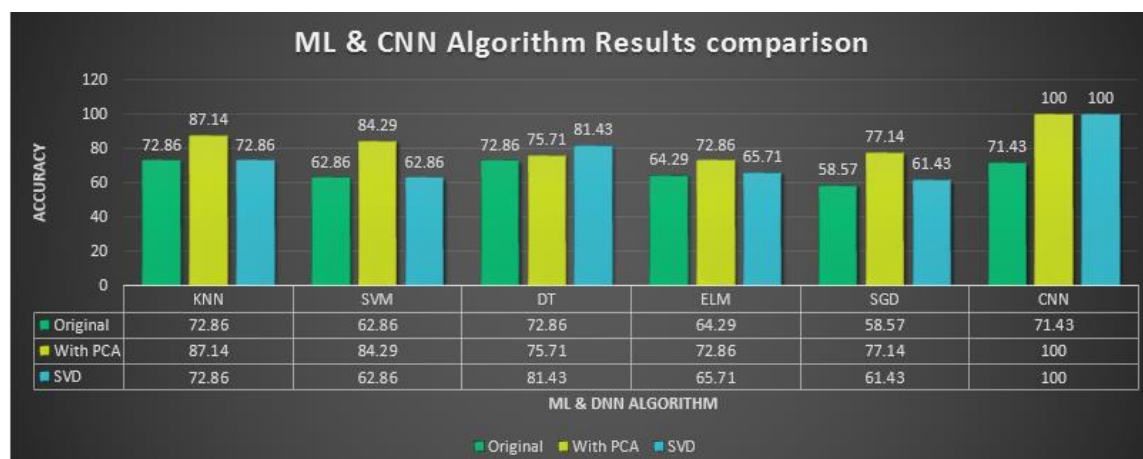
OUTPUT:

Out of this code contains reduced dimension with shape (n,2). n= number of data item, `n_components` = 2 or keeping features number = 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.

Table 4: PCA and SVD performance comparison on SCADI[1] dataset.



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)