WELCOME TO

TECH I.S.

Principal Component Analysis (PCA)

Motivation

Consider the following 3D points, If each component is stored in a byte(6), we need 3 × 6 = <u>18 bytes</u>

- Looking closer, we can see that <u>all the points are related</u> - they are all <u>scaled by some factor</u>:

$$\begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}2\\4\\6\end{bmatrix} \quad \begin{bmatrix}4\\8\\12\end{bmatrix} \quad \begin{bmatrix}3\\6\\9\end{bmatrix} \quad \begin{bmatrix}5\\10\\15\end{bmatrix} \quad \begin{bmatrix}6\\12\\18\end{bmatrix} \longrightarrow$$

$$\begin{bmatrix}1\\2\\3\end{bmatrix} = 1 \times \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad \begin{bmatrix}4\\8\\12\end{bmatrix} = 4 \times \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad \begin{bmatrix}5\\10\\15\end{bmatrix} = 5 \times \begin{bmatrix}1\\2\\3\end{bmatrix}$$

$$\begin{bmatrix}2\\4\\6\end{bmatrix} = 2 \times \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad \begin{bmatrix}3\\6\\9\end{bmatrix} = 3 \times \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad \begin{bmatrix}6\\12\\18\end{bmatrix} = 6 \times \begin{bmatrix}1\\2\\3\end{bmatrix}$$

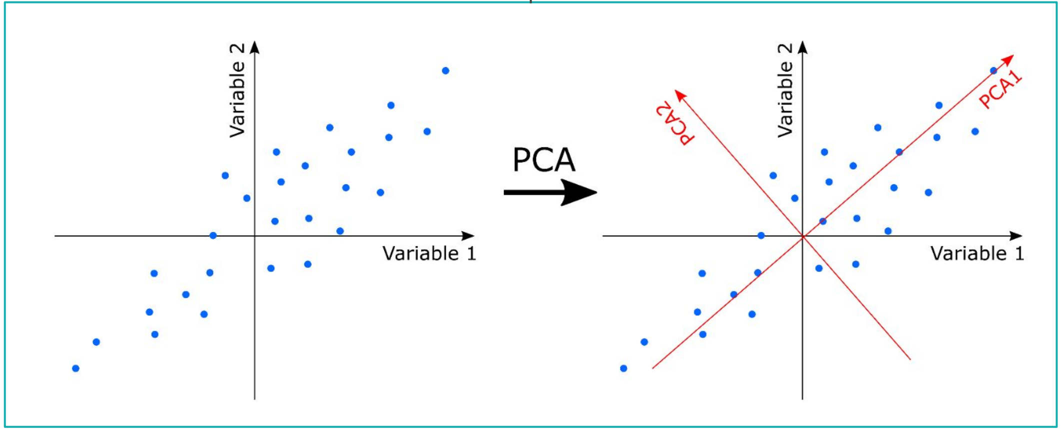
<u>They can be stored using only 9 bytes (50% savings!) :</u>

- Store one direction (3 bytes) + the multiplying constants (6 bytes)

Motivation - Contd

In previous example, we essentially rebuilt the coordination system for the data by selecting:

1. The direction with largest variance as the first new base direction.
2. The direction with the second largest variance as the second new base direction
3. And so on...



Principal Component Analysis (PCA)

PCA tries to identify the subspace (new coordinate system) in which the data approximately lies in.

PCA is usually a linear transformation that chooses a new coordinate system for the data set such that

- Greatest variance lie on the first axis (called the first principal component)
- The second greatest variance on the second axis, and so on

PCA can be used for reducing dimensionality by eliminating the later principal components.

How can we find the direction with largest variance?

By finding the eigenvector for the covariance matrix of the data.

Covariance Matrix

Covariance measures how one dimension varies w.r.t. another.
Suppose there are 3 dimensions, denoted as X, Y, Z.
The covariance matrix is:

$$COV = \begin{bmatrix} COV(X,X) & COV(X,Y) & COV(X,Z) \\ COV(Y,X) & COV(Y,Y) & COV(Y,Z) \\ COV(Z,X) & COV(Z,Y) & COV(Z,Z) \end{bmatrix}$$

We calculate the eigenvectors with the largest eigenvalues for the Covariance matrix, these eigenvectors (in order) are the principal component correspond to the dimensions that have strongest variation in the dataset.

Eigenvalues and Eigenvectors

Scalar λ and vector v are eigenvalues and eigenvectors of Matrix A:

Eigenvector of Matrix **A**

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

Eigenvalue of Matrix **A**

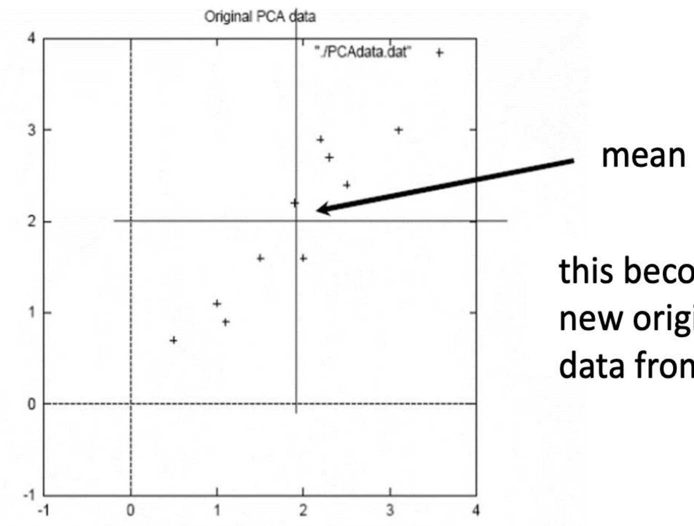
$$\underbrace{\begin{bmatrix} 1 & -3 & 3 \\ 3 & -5 & 3 \\ 6 & -6 & 4 \end{bmatrix}}_{A} \begin{bmatrix} 1/2 \\ 1/2 \\ 1 \end{bmatrix} = \underbrace{4}_{eigenvalue} \underbrace{\begin{bmatrix} 1/2 \\ 1/2 \\ 1 \end{bmatrix}}_{eigenvector}$$

Example – PCA

Consider the sample mean normalized Data for x and y:

DATA:

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9



this becomes the new origin of the data from now on

Example – STEP 1

1. Calculate the covariance matrix -> then find eigenvectors and eigenvalues
2. Now, we can choose to delete the smaller, less significant component of Eigenvector.

DATA:

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

$$\mathrm{Cov} = \begin{bmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555556 \end{bmatrix}$$

$$\text{eigenvalues} = \begin{bmatrix} 0.0490833989 \\ 1.28402771 \end{bmatrix}$$

$$\text{eigenvectors} = \begin{bmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{bmatrix}$$

$$\begin{bmatrix} -0.677873399 \\ -0.735178656 \end{bmatrix}$$

Example – STEP 2

<u>Deriving new data points</u> using the new found eigenvectors:

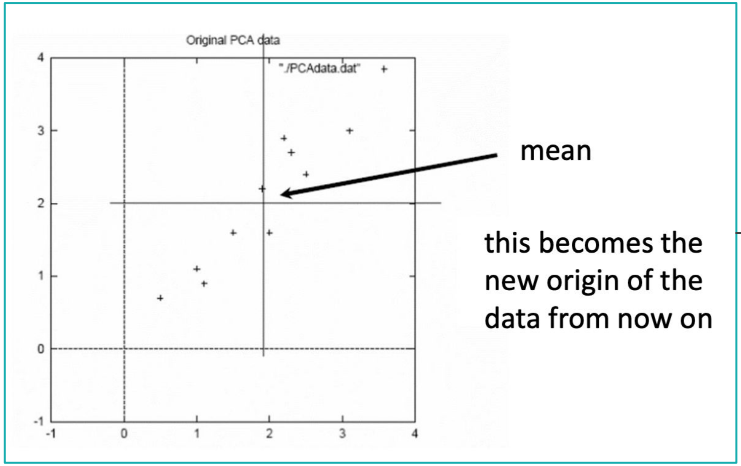
FinalData = RowZeroMeanData x RowFeatureVector

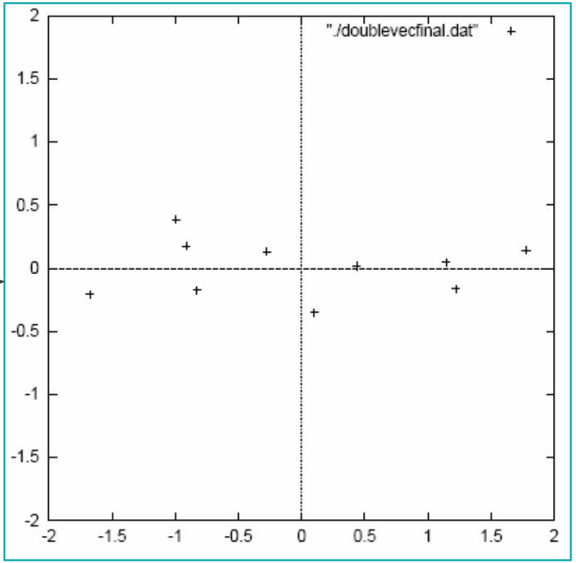
$$\text{FinalData}_{N\times d} = \begin{bmatrix} g\left(x^{(1)}\right)^{\mathsf{T}} \\ \vdots \\ g\left(x^{(N)}\right)^{\mathsf{T}} \end{bmatrix}_{N\times d} \begin{bmatrix} e_1^{\mathsf{T}} \\ \vdots \\ e_d^{\mathsf{T}} \end{bmatrix}_{d\times d}$$

- <u>RowZeroMeanData</u> is the mean-adjusted original data, i.e. the data items are in each row.
- <u>RowFeatureVector</u> is the matrix with the eigenvectors in the columns transposed so that the most significant eigenvector are at the top.

Example – STEP 3

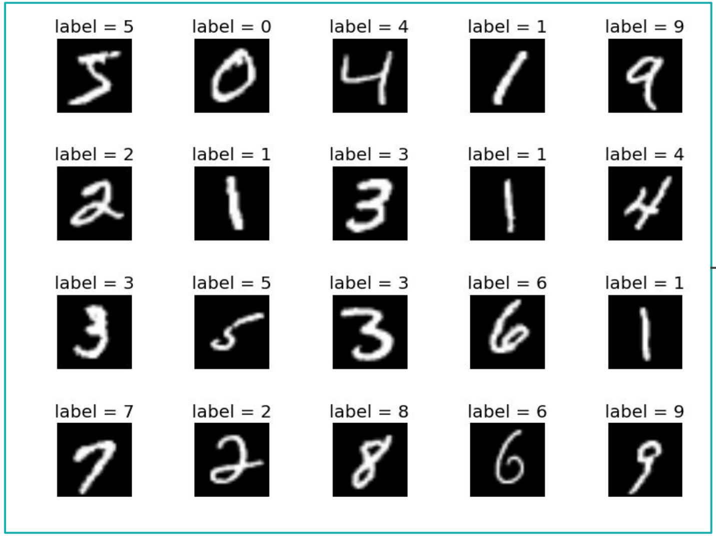
As shown below we can now discard y-axis for the PCA transformed data as, most of the data lies along x-axis:

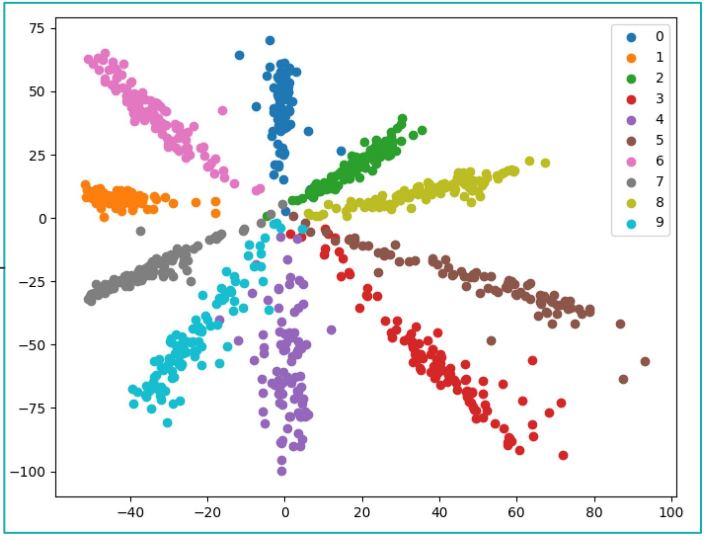




PCA – Image Compression

Original MNIST dataset is 700+ dimensions -> Using PCA it was reduced to 2 dimensions.





Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is an alternative method of matrix factorization for obtaining eigenvectors.

It states that any $m \times n$ matrix A can be written as the product of 3 matrices:

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

$$\underset{m \times n}{\overset{\text{A}}{\begin{pmatrix} x_{11} & x_{12} & & x_{1n} \\ & & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} = \underset{m \times m}{\overset{\text{U}}{\begin{pmatrix} u_{11} & & u_{m1} \\ & \ddots & \\ u_{1m} & & u_{mm} \end{pmatrix}}} \underset{m \times n}{\overset{\text{S}}{\begin{pmatrix} \sigma_1 & & 0 \\ & \ddots \; \sigma_r \; \ddots & \\ 0 & & 0 \end{pmatrix}}} \underset{n \times n}{\overset{\text{V}^\text{T}}{\begin{pmatrix} v_{11} & & v_{1n} \\ & \ddots & \\ v_{n1} & & v_{nn} \end{pmatrix}}}$$

U is $m \times m$ and its columns are orthonormal eigenvectors of AA^{T}
V is $n \times n$ and its columns are orthonormal eigenvectors of $A^{\mathsf{T}}A$

SVD - Application in RS

SVD in Recommendation System:

Suppose in the database of Walmart, there are m users and n items, and an $m \times n$ binary matrix A

- Each entry indicates whether a user has bought an item or not.
- As each user only buys very few items among all items, the matrix is very sparse

As the manager of Walmart, how can you predict the likelihood that a user will buy a given item? - Using SVD

$$\underbrace{A}_{m \times n} = U S V^T = \overbrace{\sigma_1 \underbrace{u_1}_{m \times 1} \underbrace{v_1^T}_{1 \times n}}^{\text{more significant}} + \dots + \overbrace{\underbrace{\sigma_r u_r v_r^T}_{m \times n}}^{\text{less significant}}$$

Much obliged.

TECH I.S.