Analytics for Managerial Decision Making
IBM 322

Sumit Kumar Yadav

Department of Management Studies

July 23, 2024

Course Logistics

- Book - No single text book for the course, various references will be provided
- TA-1 - Raja Babu Adityan, Research Scholar, DoMS - raja_ba@ms.iitr.ac.in
- TA-2 - Shivani Jaiswal, Research Scholar, DoMS - shivani_j@ms.iitr.ac.in
- Attendance - Institute Rules will be enforced
- Evaluation
  - MTE - 20%
  - Assignments/in-class quiz - 15%
  - Group Assignment - 15%
  - Course Participation/Attendance - 15%
  - ETE - 35%
- All course material/assignment submission etc. will be posted on MS Teams platform and https://tinyurl.com/autumnamdm

Variance and Mean when data is not one-dimensional

Applications??
Refer to Slides on "Who wrote the Royal Book of Oz?"

PCA Motivation

- To reduce the number of dimensions in the data
- To visualize the data
- To avoid over fitting

	Location	City	Society	Ambience	Airport
House 1	8	4	9	1	5
House 2	9	6	9	5	5
House 3	10	8	9	7	5
House 4	10	5	9	6	5
House 5	5	4	9	2	5
House 6	2	7	9	9	5
House 7	7	5	9	8	6
House 8	3	4	9	8	5
House 9	4	2	9	7	5
House 10	1	4	9	10	5

PCA Motivation

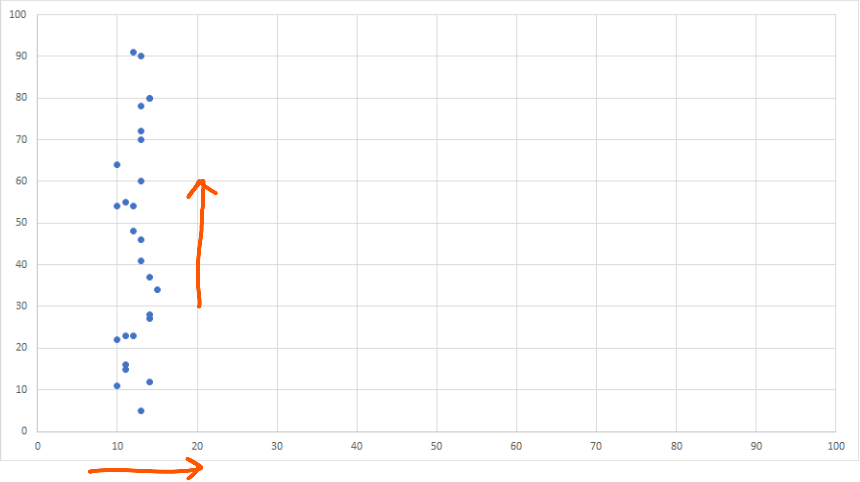
→ Feature "Extraction"

- To reduce the number of dimensions in the data
- To visualize the data
- To avoid over fitting

	Location	City	Society	Ambience	Airport
House 1	8	4	9	1	5
House 2	9	6	9	5	5
House 3	10	8	9	7	5
House 4	10	5	9	6	5
House 5	5	4	9	2	5
House 6	2	7	9	9	5
House 7	7	5	9	8	6
House 8	3	4	9	8	5
House 9	4	2	9	7	5
House 10	1	4	9	10	5

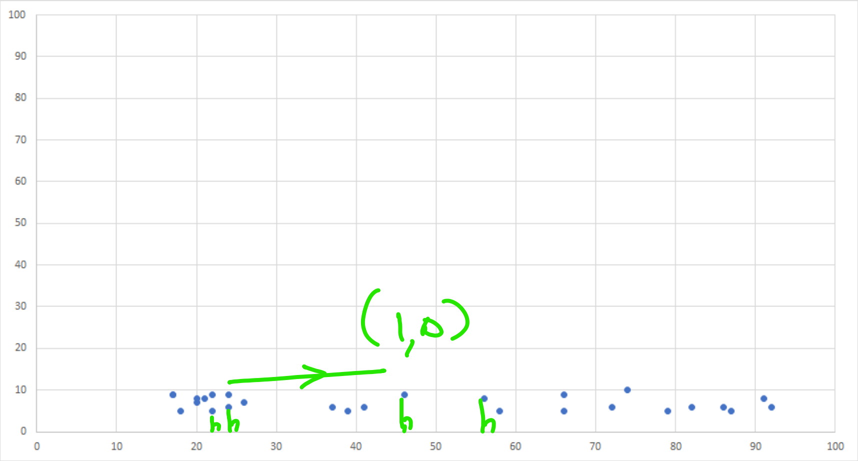
PCA Motivation

Which direction to skip?



PCA Motivation

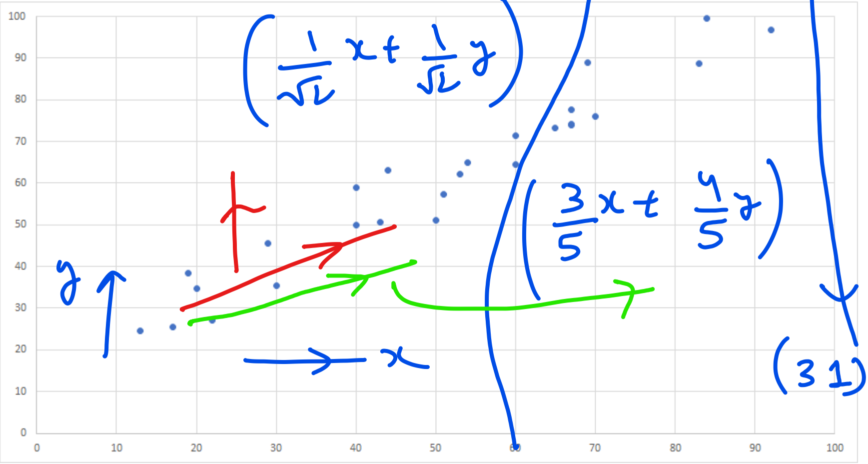
Which direction to skip?



PCA Motivation

$(25, 20)$

Which direction to skip?



PCA Motivation

Looks like we are trying to capture as much variance as possible while reducing the number of dimensions.

Linear Algebra Revisit

What is a vector?

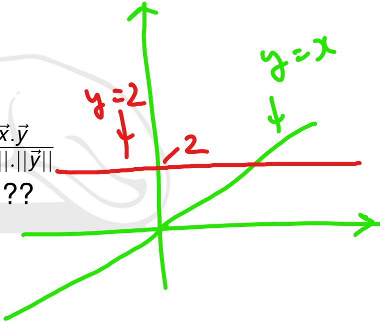
- a mathematical object that encodes a length and direction
- A vector is often represented as a 1-dimensional array of numbers, referred to as components and is displayed either in column form or row form
- Represented geometrically, vectors typically represent coordinates within a n-dimensional space

Vectors in R^n/Concept of Vector Space

(α, α)

$(1,1)$, $(-1,-1)$, $(4,4)$ (β, β)

- Vector Space - Closed under addition and multiplication $k(\alpha, \alpha)$ $(k\alpha, k\alpha)$
- Vectors on line $y = 2 * x + 1$ form a vector space??
- Addition/Subtraction (Graphical Representation)
- Multiplication by a scalar
- Dot Product (Inner Product)
- Length/Magnitude $\rightarrow (1,2,3,4,5)$
- Angle between two vectors - $\cos\theta = \frac{\vec{x}.\vec{y}}{||\vec{x}||.||\vec{y}||}$
- Dot product of perpendicular vectors = ??



$||\vec{x}||$

$(\alpha+\beta, \alpha+\beta)$

Linearly Independent and Dependent Vectors

$$\boxed{\alpha_1 = 1,\ \alpha_2 = -1,\ \alpha_3 = -1}$$

$$\begin{array}{cccc} \vec{a} & \vec{b} & \vec{c} & \vec{c} \\ \begin{pmatrix}1\\1\\1\end{pmatrix} & \begin{pmatrix}1\\1\\0\end{pmatrix} & \begin{pmatrix}0\\1\\0\end{pmatrix} & \begin{pmatrix}2\\2\\1\end{pmatrix} \end{array}$$

$$\begin{pmatrix}\alpha_1 + \alpha_2 + 2\alpha_3 \\ \alpha_1 + \alpha_2 + 2\alpha_3 \\ \alpha_1 + \alpha_3\end{pmatrix} = \begin{pmatrix}0\\0\\0\end{pmatrix}$$

- concept of zero vector

Consider m vectors in $\mathbb{R}^n$,
If $\alpha_1 \vec{v_1} + \alpha_2 \vec{v_2} + \cdots + \alpha_n \vec{v_n}$ implies
$\alpha_1 = \alpha_2 = \cdots = \alpha_m = \vec{0}$, then
the vectors are said to be linearly independent.

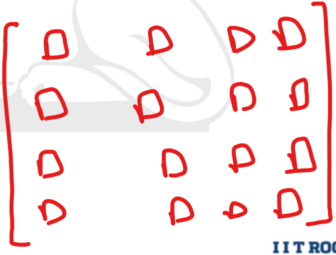
$$\begin{pmatrix}\alpha_1 + \alpha_2 \\ \alpha_1 + \alpha_2 + \alpha_3 \\ \alpha_1\end{pmatrix} = \begin{pmatrix}0\\0\\0\end{pmatrix}$$

$$\alpha_1 \begin{pmatrix}1\\1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}1\\1\\0\end{pmatrix} + \alpha_3 \begin{pmatrix}0\\1\\0\end{pmatrix} \begin{pmatrix}2\\2\\1\end{pmatrix} = \begin{pmatrix}0\\0\\0\end{pmatrix}$$

Matrices

In mathematics, a matrix (plural matrices) is a rectangular array or table of numbers, symbols, or expressions, arranged in rows and columns, which is used to represent a mathematical object or a property of such an object. - Wikipedia

- Matrix Size
- Representing any element of a matrix
- (1,n) and (m,1) matrices
- Square Matrix



Operations on Matrices

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

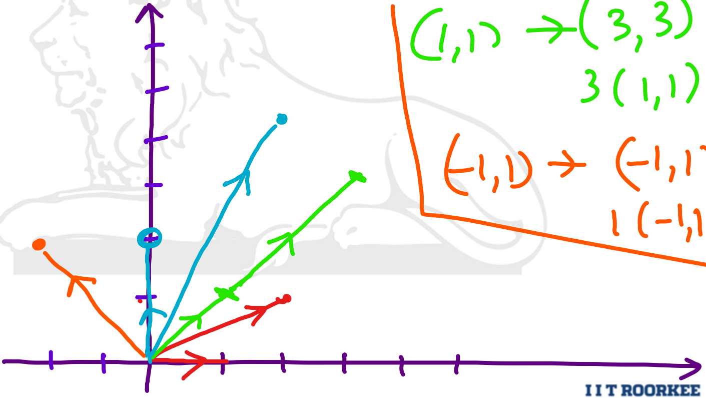
- Addition/Subtraction
- Multiplication of Matrix by a scalar
- Transpose of a Matrix
- Multiplication of two Matrices

$$\begin{bmatrix} \boxed{1 \quad 2 \quad 3} \\ \boxed{4 \quad 5 \quad 6} \end{bmatrix} \begin{bmatrix} \boxed{\begin{matrix} 1 \\ 2 \\ 3 \end{matrix}} & \boxed{\begin{matrix} 4 \\ 5 \\ 6 \end{matrix}} \end{bmatrix} = \begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix}$$

Matrix as Linear Transformation

$$A_{m\times n} B_{n\times p} = C_{m\times p}$$

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2x_1 + x_2 \\ x_1 + 2x_2 \end{bmatrix}$$



$(1,1) \to (3,3)$

$3(1,1)$

$(-1,1) \to (-1,1)$

$1(-1,1)$

Types of Matrices

- Diagonal
- Identity
- Upper Triangular
- Symmetric
- Singular
- Mirror Matrix

Matrices contd.

- Inverse of a Matrix
- Trace of a Matrix
- Relationship of Eigenvalues with trace and Determinant
- Symmetric Matrix
- Orthogonal Matrix

Eigenvalues and Eigenvectors

$M\vec{x} = \lambda\vec{x}$

$\vec{x}$

$-2\vec{x}$

$10\vec{x}$

Let M be a nxn matrix. A non-zero vector $\vec{X}$ is said to be an eigenvector of M corresponding to eigenvalue λ if

$(1, 1)$

$\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$

$\left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)$

Spectral Decomposition

$\lambda_1, \vec{v_1}$
$\lambda_2, \vec{v_2}$
$\lambda_3, \vec{v_3}$
$\vdots$
$\lambda_n, \vec{v_n}$

A n*n matrix can be written in terms of its eigenvalues and eigenvectors as follows -

$$M = \lambda_1 \vec{v_1} \vec{v_1}^T + \lambda_2 \vec{v_2} \vec{v_2}^T + ... + \lambda_n \vec{v_n} \vec{v_n}^T$$

Positive Semi-definite Matrix

$\vec{x}^T A \vec{x} \geq 0$, then A is P.S.D.

All the eigenvalues are greater than zero.

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

(For P.S.D. & symmetric)

PCA - Objectives

We have a Data Matrix(D), whose dimension is $n * f$.

$$\begin{bmatrix} d_{11} & d_{12} & \dots & d_{1f} \\ d_{21} & d_{22} & \dots & d_{2f} \\ \vdots & \vdots & \dots & \vdots \\ d_{(n-1)1} & d_{(n-1)2} & \dots & d_{(n-1)f} \\ d_{n1} & d_{n2} & \dots & d_{nf} \end{bmatrix}$$

We want to reduce the number of features to f_s.

Let us solve a simpler problem first. Let us say we want to reduce the number of dimensions/features to just 1. Which is the best way to go about it?

PCA - Objectives

We have a Data Matrix(D), whose dimension is $n * f$.

$$\begin{bmatrix} d_{11} & d_{12} & \dots & d_{1f} \\ d_{21} & d_{22} & \dots & d_{2f} \\ \vdots & \vdots & \dots & \vdots \\ d_{(n-1)1} & d_{(n-1)2} & \dots & d_{(n-1)f} \\ d_{n1} & d_{n2} & \dots & d_{nf} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_f \end{bmatrix} =$$

We want to reduce the number of features to f_s.

Let us solve a simpler problem first. Let us say we want to reduce the number of dimensions/features to just 1. Which is the best way to go about it?

PCA Objective

We are looking for a unit vector $\vec{v} = (v_1, v_2, ..., v_f)$ such that the variance of $D\vec{v}$ is as large as possible.

Variance($D\vec{v}$) can be written as $\vec{v}^T \Sigma \vec{v}$, where Σ is the variance covariance matrix of D. (Try to prove this or atleast verify this)

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1f} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2f} \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_{(f-1)1} & \sigma_{(f-1)2} & \cdots & \sigma_{(f-1)f} \\ \sigma_{1f} & \sigma_{f2} & \cdots & \sigma_f^2 \end{bmatrix}$$

PCA Objective

We are looking for a unit vector $\vec{v} = (v_1, v_2, ..., v_f)$ such that the variance of $D\vec{v}$ is as large as possible.

Variance($D\vec{v}$) can be written as $\vec{v}^T \Sigma \vec{v}$, where Σ is the variance covariance matrix of D. (Try to prove this or atleast verify this)

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1f} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2f} \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_{(f-1)1} & \sigma_{(f-1)2} & \cdots & \sigma_{(f-1)f} \\ \sigma_{1f} & \sigma_{f2} & \cdots & \sigma_f^2 \end{bmatrix}$$

Variance $(D\vec{v}) = \vec{v}^T \Sigma \vec{v}$

PCA Objective

We are looking for a unit vector $\vec{v} = (v_1, v_2, ..., v_f)$ such that the variance of $D\vec{v}$ is as large as possible.

Variance($D\vec{v}$) can be written as $\vec{v}^T\Sigma\vec{v}$, where Σ is the variance covariance matrix of D. (Try to prove this or atleast verify this)

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1f} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2f} \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_{(f-1)1} & \sigma_{(f-1)2} & \cdots & \sigma_{(f-1)f} \\ \sigma_{1f} & \sigma_{f2} & \cdots & \sigma_f^2 \end{bmatrix}$$

PCA Optimization problem

$\Sigma_{f \times f}$

$\lambda_1, \lambda_2, \text{----}, \lambda_f$

$\vec{v_1} \quad \vec{v_2} \quad \vec{v_f}$

Maximize - $\vec{v}^T \Sigma \vec{v}$

such that $||\vec{v}|| = 1$

Σ can be shown to be positive semi-definite. Thus, all the eigenvalues are greater than zero, and all eigenvectors are orthogonal.

$\Sigma = \lambda_1 \vec{v_1} \vec{v_1}^T + \lambda_2 \vec{v_2} \vec{v_2}^T + \ldots + \lambda_f \vec{v_f} \vec{v_f}^T$

$\lambda_1 \geq \lambda_2 \geq \ldots . \geq \lambda_f$

$\lambda_1 \geq \lambda_2 \geq \text{---} \text{--} \geq \lambda_f$

$\Sigma = \lambda_1 \vec{v}_1 \vec{v}_1^T + \lambda_2 \vec{v}_2 \vec{v}_2^T + \text{---} + \lambda_f \vec{v}_f \vec{v}_f^T$

PCA Optimization problem

Maximize - $\vec{v}^T \Sigma \vec{v}$

such that $||\vec{v}|| = 1$

Σ can be shown to be positive semi-definite. Thus, all the eigenvalues are greater than zero, and all eigenvectors are orthogonal.

$$\Sigma = \lambda_1 \vec{v_1} \vec{v_1}^T + \lambda_2 \vec{v_2} \vec{v_2}^T + \ldots + \lambda_f \vec{v_f} \vec{v_f}^T$$

$$\lambda_1 \geq \lambda_2 \geq \ldots. \geq \lambda_f$$

PCA Optimization problem

Maximize - $\vec{v}^T \Sigma \vec{v}$

such that $||\vec{v}|| = 1$

Σ can be shown to be positive semi-definite. Thus, all the eigenvalues are greater than zero, and all eigenvectors are orthogonal.

$$\Sigma = \lambda_1 \vec{v_1} \vec{v_1}^T + \lambda_2 \vec{v_2} \vec{v_2}^T + ... + \lambda_f \vec{v_f} \vec{v_f}^T$$

$$\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_f$$

PCA Optimization problem

Maximize - $\vec{v}^T(\lambda_1 \vec{v_1} \vec{v_1}^T + \lambda_2 \vec{v_2} \vec{v_2}^T + ... + \lambda_f \vec{v_f} \vec{v_f}^T)\vec{v}$

such that $||\vec{v}|| = 1$

All $\vec{v_i}'s$ are unit vectors and orthogonal to each other, so the best choice for $\vec{v}$ is ??

PCA Optimization problem

$\lambda_1(\vec{v}^T\vec{v}_1\vec{v}_1^T\vec{v}) + \lambda_2(\vec{v}^T\vec{v}_2\vec{v}_2^T\vec{v}) + \ldots + \lambda_f(\vec{v}^T\vec{v}_f\vec{v}_f^T\vec{v})$

Maximize - $\vec{v}^T(\lambda_1\vec{v}_1\vec{v}_1^T + \lambda_2\vec{v}_2\vec{v}_2^T + ... + \lambda_f\vec{v}_f\vec{v}_f^T)\vec{v}$

such that $||\vec{v}|| = 1$

Maximize:

All $\vec{v}_i' s$ are unit vectors and orthogonal to each other, so the best choice for $\vec{v}$ is ??

$\left[\lambda_1(\vec{v}\cdot\vec{v}_1)^2 + \lambda_2(\vec{v}\cdot\vec{v}_2)^2 + \ldots + \lambda_f(\vec{v}\cdot\vec{v}_f)^2\right]$

s.t. $||\vec{v}|| = 1 \Rightarrow \sqrt{(\vec{v}\cdot\vec{v}_1)^2 + (\vec{v}\cdot\vec{v}_2)^2 + \ldots (\vec{v}\cdot\vec{v}_f)^2} = 1$

What about reducing to 2 dimensions?

$l_i = (\vec{v} \cdot \vec{v_i})^2$

Variance is no longer a scalar which can be compared across all possible 2 dimensions where data can be projected.

We have an additional constraint that the co-variance terms of the reduced dimensions should be zero.

Max: $\lambda_1 (l_1) + \lambda_2 (l_2) + \dots + \lambda_f l_f$

s.t.

Fortunately, this happens if we keep going in the same sequence of the principal components.

$l_1 + l_2 + \dots + l_f = 1$

We can now try to maximize the sum of the variance terms in the 2 projected directions.

So, it looks like that the most logical thing to do is to pick up the eigenvector corresponding to the second largest eigenvector. This is what we do in PCA.

$\lambda_1 \geq \lambda_2 \geq \lambda_3 \dots \geq \lambda_f$

$\boxed{\begin{array}{l} l_1 = 1 \\ l_2 = 0 \\ \vdots \\ l_f = 0 \end{array}}$

What about reducing to 2 dimensions?

Variance is no longer a scalar which can be compared across all possible 2 dimensions where data can be projected.

We have an additional constraint that the co-variance terms of the reduced dimensions should be zero.

Fortunately, this happens if we keep going in the same sequence of the principal components.

We can now try to maximize the sum of the variance terms in the 2 projected directions.
So, it looks like that the most logical thing to do is to pick up the eigenvector corresponding to the second largest eigenvector. This is what we do in PCA.

What about reducing to 2 dimensions?

Variance is no longer a scalar which can be compared across all possible 2 dimensions where data can be projected.

We have an additional constraint that the co-variance terms of the reduced dimensions should be zero.

Fortunately, this happens if we keep going in the same sequence of the principal components.

We can now try to maximize the sum of the variance terms in the 2 projected directions.
So, it looks like that the most logical thing to do is to pick up the eigenvector corresponding to the second largest eigenvector. This is what we do in PCA.

What about reducing to 2 dimensions?

Variance is no longer a scalar which can be compared across all possible 2 dimensions where data can be projected.

We have an additional constraint that the co-variance terms of the reduced dimensions should be zero.

Fortunately, this happens if we keep going in the same sequence of the principal components.

We can now try to maximize the sum of the variance terms in the 2 projected directions.
So, it looks like that the most logical thing to do is to pick up the eigenvector corresponding to the second largest eigenvector. This is what we do in PCA.

Reconstructing the data

One could start with the objective that we want to minimize the reconstruction error, and we would get the same result as what we have described above.
Basic Ideas - Any k-dimensional vector can be represented in terms of k-orthonormal vectors.
Eg - Consider the vector (2,3) in 2-D space and two orthonormal vectors (1,0) and (0,1)
(2,3) = ((2,3).(1,0))(1,0) + ((2,3).(0,1))(0,1)

Too trivial, isn't it??

Reconstructing the data

One could start with the objective that we want to minimize the reconstruction error, and we would get the same result as what we have described above.

Basic Ideas - Any k-dimensional vector can be represented in terms of k-orthonormal vectors.

Eg - Consider the vector (2,3) in 2-D space and two orthonormal vectors (1,0) and (0,1)

(2,3) = ((2,3).(1,0))(1,0) + ((2,3).(0,1))(0,1)

Too trivial, isn't it??

Reconstructing the data

One could start with the objective that we want to minimize the reconstruction error, and we would get the same result as what we have described above.
Basic Ideas - Any k-dimensional vector can be represented in terms of k-orthonormal vectors.
Eg - Consider the vector (2,3) in 2-D space and two orthonormal vectors (1,0) and (0,1)
$(2,3) = ((2,3).(1,0))(1,0) + ((2,3).(0,1))(0,1)$

Too trivial, isn't it??

Reconstructing the data

Eg - Consider the vector (2,3) in 2-D space and two orthonormal vectors (0.6,0.8) and (-0.8,0.6)

(2,3) = ((2,3).(0.6,0.8))(0.6,0.8) + ((2,3).(-0.8,0.6))(-0.8,0.6)

Thus, in general for a k-dimensional vector $\vec{d}$, and k-orthonormal vectors $\vec{v_1}, \vec{v_2}, \vec{v_3}, ..., \vec{v_k}$

$$\vec{d} = (\vec{d}.\vec{v_1})\vec{v_1} + (\vec{d}.\vec{v_2})\vec{v_2} + (\vec{d}.\vec{v_3})\vec{v_3} + ... + (\vec{d}.\vec{v_k})\vec{v_k}$$

Reconstructing the data

Eg - Consider the vector (2,3) in 2-D space and two orthonormal vectors (0.6,0.8) and (-0.8,0.6)

(2,3) = ((2,3).(0.6,0.8))(0.6,0.8) + ((2,3).(-0.8,0.6))(-0.8,0.6)

Thus, in general for a k-dimensional vector $\vec{d}$, and k-orthonormal vectors $\vec{v_1}, \vec{v_2}, \vec{v_3}, ..., \vec{v_k}$

$$\vec{d} = (\vec{d}.\vec{v_1})\vec{v_1} + (\vec{d}.\vec{v_2})\vec{v_2} + (\vec{d}.\vec{v_3})\vec{v_3} + ... + (\vec{d}.\vec{v_k})\vec{v_k}$$

PCA reconstruction

In PCA, once we have found out the eigenvalues and eigenvectors and projected the data in the lower dimensional space(f_s), the way to reconstruct for a data point back to the original dimensions (f) is -

$$\vec{d_{inv}} = (\vec{d}.\vec{v_1})\vec{v_1} + (\vec{d}.\vec{v_2})\vec{v_2} + (\vec{d}.\vec{v_3})\vec{v_3} + ... + (\vec{d}.\vec{v_{f_s}})\vec{v_{f_s}}$$

This is essentially what we get after the matrix manipulations we saw in the previous session.

PCA reconstruction

In PCA, once we have found out the eigenvalues and eigenvectors and projected the data in the lower dimensional space(f_s), the way to reconstruct for a data point back to the original dimensions (f) is -

$$\vec{d_{inv}} = (\vec{d}.\vec{v_1})\vec{v_1} + (\vec{d}.\vec{v_2})\vec{v_2} + (\vec{d}.\vec{v_3})\vec{v_3} + ... + (\vec{d}.\vec{v_{f_s}})\vec{v_{f_s}}$$

This is essentially what we get after the matrix manipulations we saw in the previous session.

PCA reconstruction

In PCA, once we have found out the eigenvalues and eigenvectors and projected the data in the lower dimensional space(f_s), the way to reconstruct for a data point back to the original dimensions (f) is -

$$\vec{d_{inv}} = (\vec{d}.\vec{v_1})\vec{v_1} + (\vec{d}.\vec{v_2})\vec{v_2} + (\vec{d}.\vec{v_3})\vec{v_3} + ... + (\vec{d}.\vec{v_{f_s}})\vec{v_{f_s}}$$

This is essentially what we get after the matrix manipulations we saw in the previous session.

Percentage of Variance Captured in PCA

Trace of Variance-Covariance Matrix = Sum of Eigenvalues

Think of trace as total variance in the data

SVD

Singular Value Decomposition - A way to approximate a general matrix of dimension m*n.

Can we use PCA to approximate the matrix of size n*n? Can we do it always??

SVD

$\lambda_1 \geq \lambda_2 \geq \lambda_3 \dots \geq \lambda_n$

$$\underset{n \times n}{M} = \underbrace{\lambda_1 \vec{v}_1 \vec{v}_1^T + \lambda_2 \vec{v}_2 \vec{v}_2^T + \dots}\quad + \lambda_n \vec{v}_n \vec{v}_n^T$$

Singular Value Decomposition - A way to approximate a general matrix of dimension m*n.

Can we use PCA to approximate the matrix of size n*n? Can we do it always??

SVD

Singular Value Decomposition - A way to approximate a general matrix of dimension m*n.
Can we use PCA to approximate the matrix of size n*n? Can we do it always??

SVD

$A_{m \times n}$

$(AA^T)_{m \times m} \rightarrow$ symmetric

Let A be a general matrix of size m*n.
What can be said about AA^T and A^TA

$(AA^T)^T = (A^T)^T A^T = AA^T$

Is AA^T positive semi-definite

$[x^T A A^T x] \rightarrow (A^T x)^T (A^T x) \geq 0$

A^TA

symm.

positive

semi-

definite

$$\boxed{A\vec{v}_1 = \sqrt{\lambda_1}\,\vec{u}_1}$$

$$\boxed{m \geq n}$$

$$(AA^T)_{m\times m} \rightarrow \lambda_1, \lambda_2, \ldots, \lambda_m$$

$$\vec{u}_1 \quad \vec{u}_2 \quad \vec{u}_m$$

$$\boxed{A\vec{v}_2 = \sqrt{\lambda_2}\,\vec{u}_2}$$

Let A be a general matrix of size m*n.
What can be said about AA^T and A^TA

$$\boxed{A^T\vec{u}_1 = \sqrt{\lambda_1}\,\vec{v}_1}$$

$$(A^TA)_{n\times n} \rightarrow \gamma_1, \gamma_2, \ldots, \gamma_n$$

$$\vec{v}_1 \quad \vec{v}_2 \quad \vec{v}_n$$

$$\boxed{\lambda_1 = \gamma_1, \lambda_2 = \gamma_2, \ldots \lambda_n = \gamma_n, \boxed{\lambda_{n+1}, \ldots \lambda_m = 0}}$$

SVD

$m \geq n$

$$\boxed{A = \sqrt{\lambda_1}\, \vec{u}_1 \vec{v}_1^T + \sqrt{\lambda_2}\, \vec{u}_2 \vec{v}_2^T + \dots + \sqrt{\lambda_n}\, \vec{u}_n \vec{v}_n^T}$$

r

Which *u* and *v* to choose?

$$\boxed{A = U \Sigma V^T}$$

$(m \times r)\ (r \times r)\ (r \times n)$

Limitations of PCA

1.) No clear interpretation of the new features

2.) Doesn't preserve the separability

3.) LDA tries to overcome this limitation

PCA - Objectives

We have a Data Matrix(D), whose dimension is $n * f$.

$$\begin{bmatrix} d_{11} & d_{12} & \dots & d_{1f} \\ d_{21} & d_{22} & \dots & d_{2f} \\ \vdots & \vdots & \dots & \vdots \\ d_{(n-1)1} & d_{(n-1)2} & \dots & d_{(n-1)f} \\ d_{n1} & d_{n2} & \dots & d_{nf} \end{bmatrix}$$

We want to reduce the number of features to f_s.

Find the eigenvalues and eigenvectors of Σ matrix. Project along the eigenvectors corresponding to top f_s eigenvalues

PCA - Objectives

We have a Data Matrix(D), whose dimension is $n * f$.

$$\begin{bmatrix} d_{11} & d_{12} & \dots & d_{1f} \\ d_{21} & d_{22} & \dots & d_{2f} \\ \vdots & \vdots & \dots & \vdots \\ d_{(n-1)1} & d_{(n-1)2} & \dots & d_{(n-1)f} \\ d_{n1} & d_{n2} & \dots & d_{nf} \end{bmatrix}$$

We want to reduce the number of features to f_s.

Find the eigenvalues and eigenvectors of Σ matrix. Project along the eigenvectors corresponding to top f_s eigenvalues

Eigenfaces example

- Several photos(4324) of people stored as 62*47 image.

$$\begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots \end{bmatrix}$$

- View this as a vector of size 2914 (=62*47)

$$\begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

- Now, we have a matrix of size 4324*2914
- Apply PCA of this matrix, get the eigenvectors (of 2914 dimensiona)
- View this eigenvector as 62*47, convert this into a image