

MACHINE LEARNING AND MATHEMATICS

Project report submitted to

Mahatma Gandhi University

In partial fulfilment of the requirement for the award of the degree of

MASTER OF SCIENCE IN MATHEMATICS

Submitted by

SREELAKSHMI M S

Reg. No. 200011014787

Under the Supervision of

Mrs. Riya Aliyas



DEPARTMENT OF MATHEMATICS

Bharata Mata College, Thrikkakara

2020-2022

ACKNOWLEDGEMENT

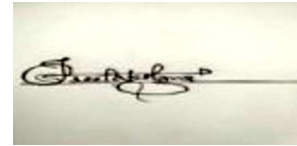
I sincerely thank each and everyone who has helped me for the successful completion of this project. I thank God Almighty for all his blessings showered upon me.

I would like to place on record all my gratitude to Mrs. Riya Aliyas, Department of Mathematics, Bharata Mata College, Thrikkakara, who has provided her valuable guidance throughout this project work.

Also, I thank all other teachers of the department of their encouragement. My friends were also helpful to me through their suggestions for which I am grateful towards them.

Place: Thrikkakara

Date:



SREELAKSHMI M S

CERTIFICATE

This is to certify that the dissertation entitled “MACHINE LEARNING AND MATHEMATICS” submitted by Sreelakshmi M S is a record of work done by the candidate during the period of her study under my supervision and guidance.

Place : Thrikkakara

Date:

Mrs. Riya Aliyas

Department of Mathematics

Bharata Mata College,

Thrikkakara.

DECLARATION

I hereby declare that the project report entitled “MACHINE LEARNING AND MATHEMATICS” submitted for the M.Sc. Degree is my original work done under the supervision of Mrs. Riya Aliyas and the project has not formed the basis for the award of any academic qualification fellowship or other similar title of any other university or board.

Place:

Date:

Sreelakshmi M S

CONTENTS

<u>Introduction</u>	<u>1</u>
<u>Preliminaries</u>	<u>2</u>
<u>1. Least Square Approximation and Minimum Norm Solution</u>	<u>5</u>
<u>1.1 Over determined system of equations</u>	<u>5</u>
<u>1.2 Method of Least Square Approximation</u>	<u>7</u>
<u>1.3 Underdetermined system of Equations</u>	<u>9</u>
<u>1.4 Minimum Norm Solution</u>	<u>9</u>
<u>2. Matrix Decomposition</u>	<u>12</u>
<u>2.1 Matrix Eigen Value Decomposition</u>	<u>13</u>
<u>2.2 Singular Value Decomposition (SVD)</u>	<u>17</u>
<u>2.3 Properties and Applications of SVD –</u>	<u>16</u>
<u>Geometrical Interpretation of SVD</u>	<u>20</u>
<u>2.4 Polar Decomposition</u>	<u>19</u>
<u>3. Low Rank Approximation</u>	<u>23</u>
<u>4. Principle Component Analysis (PCA)</u>	<u>27</u>
<u>4.1 Procedure of performing PCA</u>	<u>29</u>
<u>Conclusion</u>	<u>32</u>
<u>Bibliography</u>	<u>33</u>

ABSTRACT

Machine Learning is powered by four critical concepts and is statistics ,linear algebra, probability and calculus.

In first chapter,we discuss about the Least Square Approximation and Minimum Norm solution.Least Squares method is a mathematical technique that allows the analyst to determine the best way of fitting a curve on top of a chart of data points.It is widely used to make scatter plots easier to interpret and is associated with regression analysis.

In the second chapter we deal with matrix decomposition.There are several matrix decomposition methods while here we discussing about Matrix Eigen Decomposition , properties and application of Singular value decomposition(SVD) and Polar Decomposition.

In third chapter,we study about Low Rank Approximation,which is a way to recover the original low-rank matrix i.e , find the matrix that is more consistentwith the current matrix.

While the last chapter includes Principal Component Analysis ,which is an unsupervised learning algorithm that is used for the dimensionally reduction in machine learning.

INTRODUCTION

Machine learning is an application of artificial intelligence that provides system to the ability to automatically learn and improve from past behaviour. Machine learning is all about maths, which in turn helps in creating an algorithm that can learn from data to make an accurate prediction. Machine learning is primarily built on mathematical prerequisites so as long as you can understand why the maths is used, you will find it more interesting. With this, you will understand why we pick one machine learning algorithm over the other and how it affects the performance of the machine learning model. In this project, we will be discussing exactly the mathematical concepts you need to learn to master the concepts of machine learning. We will also learn why we use mathematics in machine learning with some examples. Machine learning is powered by four critical concepts and is statistics. Linear Algebra, Probability, and Calculus. While statistical concepts are the core part of every model, calculus helps us learn and optimize a model. Linear algebra comes exceptionally handy when we are dealing with a huge dataset and probability helps in predicting the livelihood of events that will be occurring.

PRELIMINARIES

➤ **RANK OF A MATRIX**

$M_{m \times n}(\mathbb{R})$ denotes the set of all $m \times n$ matrices with real entries. Rank of a matrix can be defined as,

The number of linearly independent rows or columns of A

- Order of largest non-singular submatrix of A
- Dimension of row or column space of A
- The number of non-zero rows in row reduced echelon form of A
- The order of identity submatrix in the normal form of A
- The rank of the linear transformation from \mathbb{R}^n to \mathbb{R}^m corresponding A
- Usually denoted by $\rho(A)$

➤ **EIGEN VALUES AND EIGEN VECTORS**

Consider a square matrix $n \times n$. If X is the non-trivial column vector solution of the matrix equation $AX = \lambda X$, where λ is a scalar, then X is the eigen vector of matrix and the corresponding value of λ is the eigen value of matrix A.

➤ **SOME SPECIAL MATRICES**

Symmetric Matrix

In linear algebra, a symmetric matrix is a square matrix is equal to its transpose.

Skew Symmetric Matrix

A skew symmetric matrix is a square matrix that is equal to negative to its transpose.

Positive Definite Matrix

A positive definite matrix is a symmetric matrix where every eigen value is positive

Negative Definite Matrix

A negative definite matrix is a symmetric matrix all of whose eigen values are negative.

➤ SYSTEMS OF LINEAR EQUATIONS

Linear Systems

A linear equation in variables x_1, x_2, \dots, x_n is an equation of the form

$$b = a_1x_1 + a_2x_2 + \dots + a_nx_n,$$

where a_1, a_2, \dots, a_n and b are constant real or complex numbers. The constant a_i is called the coefficient of x_i and b is called the constant term of the equation.

A system of linear equations (linear system) is a finite collection of linear equation in same variables. For instance, a linear system of m equations in n variables x_1, x_2, \dots, x_n can be written as;

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m$$

A solution of a linear system above is a tuple (s_1, s_2, \dots, s_n) of numbers that makes each equation a true statement when the values s_1, s_2, \dots, s_n are substituted for x_1, x_2, \dots, x_n respectively. The set of all solutions of a linear system is called the solution of the system.

Any system of linear equations has one of the following exclusive conclusions.

- (a) No solution.
- (b) Unique solution.
- (c) Infinitely many solutions.

A linear system is said to be consistent if it has at least one solution; and is said to be inconsistent if it has no solution.

CHAPTER 1

Least Square Approximation and Minimum Norm Solution

1.1 Over determined System of Equations

Definition: A Linear system is overdetermined if it has more equations than variables.

i.e.,

Consider a linear system of equations $A X = b$, where A is m by n coefficient matrix, X is the unknown vector belongs to R^n and b belongs to R^m is the right hand side vector, which is given to us. So, if $m > n$, (that is when the number of observations greater than number of variables), then the system is said to be over-determined.

Example 1.1.1: The following system is an overdetermined system:

$$a + b = 0$$

$$a - b = 1$$

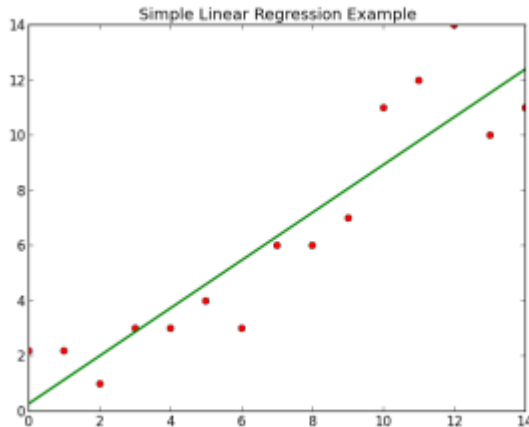
$$a - 2b = 0$$

Clearly No solution to the above system.

An overdetermined system is almost always inconsistent (it has no solution) when constructed with random coefficients. However, an overdetermined system will have solutions in some cases, for example if some equation occurs several times in the system, or if some equations are linear combinations of the others. But, in general, exact solution will come very rarely. So, what we have to look? We have to look for **n approximate solution**. Such an approximate solution is called **least square approximation of over determined system**.

one of the very basic example of over determined system is linear regression, that is the line fitting. So, generally we often run into the problem that we have more than two points and try to represent our points with one straight line. So, suppose we are given 10 points and we have to fit a line, which is the best fit line from these 10 points. However, these 10 data points which I am talking do not lie on a straight line. So, we can try infinitely many straight line to fit all the data points, under this situation the problem of least square is to find the line that fits the data the best. Here best means which is having the minimum residual error, this is called **linear regression**. The best fitting line is often called the **least square line** or the **regression line** also. And based on that we say for over determined system the solution is least square approximation solution. The best means which is having the minimum residual error. Now **Residual** is nothing but the distances between the observed data points and the corresponding points on the model line.

Fig 1.1:



To obtain the best fitting line we need to minimize the sum of the square of the residuals as we are doing here. Residual is obtained by taking the perpendicular distance from line to all points.

And then sum of the square of all those residual is called the residual error.

1.2 METHOD OF LEAST SQUARE APPROXIMATION

Given $AX = b$, where A belongs to $M_{m \times n}(R)$

And m is quite bigger than n that we are having an over determined system.

Here in least square approximation, we solve the **optimization problem** that is, we minimize the Euclidean norm between AX and b .

i.e., $\min \|AX - b\|_2$

Example 1.2.1:

Suppose we have a 3 by 2 system.

Say,

$$a_{11} x_1 + a_{12} x_2 = b_1$$

$$a_{21} x_1 + a_{22} x_2 = b_2$$

$$a_{31} x_1 + a_{32} x_2 = b_3$$

Then, Let $E = \|AX - b\|_2$

$$\begin{aligned} &= (a_{11} x_1 + a_{12} x_2 - b_1)^2 + \\ &\quad (a_{21} x_1 + a_{22} x_2 - b_2)^2 + \\ &\quad (a_{31} x_1 + a_{32} x_2 - b_3)^2 \end{aligned}$$

for minimizing this, we have to put the necessary condition of the minima that

is ,

$$\frac{\partial E}{\partial x_1} = 0 \text{ and } \frac{\partial E}{\partial x_2} = 0$$

From this we will get two linear equations in x_1 and x_2 , then by solving those two linear equations we will get the value of x_1 and x_2 , which minimize the sum of the squares of the residual errors.

Easiest Approach

The easiest way is, we are having $AX = b$.

Multiply both side by A^T .

So, $A^T A X = A^T b$. So, here it A is m by n matrix then $A^T A$ will become n by n matrix.

$$X = (A^T A)^{-1} A^T b$$

$= A^\dagger b$, where $A^\dagger = (A^T A)^{-1} A^T$ and is called the right pseudo inverse of A.

And $X = A^\dagger b$, is the least square solution of $AX = b$.

x

Example 1.2.2:

Consider the overdetermined system,

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix}$$

Here, $A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}$

Then, $A^T A = \begin{pmatrix} 3 & 3 \\ 3 & 5 \end{pmatrix}$

$$(A^T A)^{-1} = \frac{1}{6} \begin{bmatrix} 5 & -3 \\ -3 & 3 \end{bmatrix}$$

$$X = A^\dagger b$$

$$= (A^T A)^{-1} A^T b$$

$$= \frac{1}{6} \begin{bmatrix} 5 & -3 \\ -3 & 3 \end{bmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 5 \\ -3 \end{pmatrix}$$

$$x_1 = 5 \text{ and } x_2 = -3$$

This is how we find least square solution or approximation of an over determined system.

1.3 UNDER DETERMINED SYSTEM OF EQUATIONS

Definition: A Linear system is underdetermined if it has more variables than equations.

i.e.,

Consider a linear system of equations $AX = b$, where A is m by n coefficient matrix, X is the unknown vector belongs to R^n and b belongs to R^m in the right hand side vector, which is given to us. So, if $m < n$, (that is when the number of observations is considerably lesser than number of variables), then the system is said to be underdetermined. In this case, we are having $n-m$ free variables, assigning any arbitrary values to these variables lead to a solution of $AX = b$. Therefore, we can have infinitely many solutions of the system $AX = b$.

Example 1.3.1:

The following system is an underdetermined system:

$$\begin{cases} a + b = 0 \\ a - c = 1 \end{cases}$$

Clearly this system has infinite number of solutions.

1.4 MINIMUM NORM SOLUTION

A **minimum norm** solution is that which minimize $\|X\|$ among these infinite solutions. i.e., Out of those infinite numbers of solutions we are looking for a solution which is having the minimum norm and such a solution is called **minimum normed solution**.

Mathematically how can we pose this problem? So, we have to find out X , which Minimize the norm of X , subject to $AX = b$.

$$\text{i.e., } \min \{ \|AX - b\|_2 + \|X\|_2 \}$$

If we just compare with the earlier one least square approximation case, there we were having only this objective function but here

we are having this

minimum norm condition extra. So, how to solve such a system? Again we will use the concept of pseudo inverse.

The minimization problem can be solved as,

$$X^* = A^T(AA^T)^{-1} b$$

Here, $A^{\dagger} = A^T(AA^T)^{-1}$ is called the left pseudo-inverse of the matrix A.

Example 1.4.1:

Consider the underdetermined system

$$\begin{pmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\text{Here } A = \begin{pmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \end{pmatrix}$$

$$\begin{aligned} AA^T &= \begin{pmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix} \end{aligned}$$

$$(AA^T)^{-1} = \frac{1}{8} \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$$

$$\begin{aligned} X &= A^{\dagger} b \\ &= A^T(AA^T)^{-1} b \\ &= \begin{pmatrix} 1 & -1 \\ 1 & -1 \\ 1 & 1 \end{pmatrix} \frac{1}{8} \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{2} \end{pmatrix} \end{aligned}$$

$$x_1 = 1/4$$

$$x_2 = 1/4$$

$$x_3 = 1/2$$

This is the minimum normed solution of the given underdetermined system.

CHAPTER 2

Matrix Decomposition

What is going to be the benefit of decomposing a matrix? What does that mean? When we decompose anything, we break it into its constituent elements. Assume we are going to disintegrate a tool (a car or a watch!). Such action helps us to understand the core particles and their tasks. Furthermore, it helps to have a better understanding of how that specific tool works and its characteristics! Assume that the tool is a matrix which we would like to decompose. There are different approaches to decompose a matrix. However, perhaps the most commonly used one is Matrix Eigen Decomposition which is decomposing a matrix using its Eigenvectors and Eigenvalues.

Definition : Assuming we have the square matrix of $\mathbf{A} \in \mathbb{R}^{N \times N}$. The nonzero vector $\mathbf{V} \in \mathbb{R}^{N \times 1}$ is an eigen vector and scalar λ is its associated eigenvalue if we have:

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

2.1 Matrix Eigen decomposition:

Assuming we have the square matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ of which has N linear independent eigenvectors $\mathbf{V}^i, i \in 1, \dots, N$. Then, we can factorize matrix \mathbf{A} as below:

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$$

Where $\mathbf{V} \in \mathbb{R}^{N \times N}$ is the square matrix whose j^{th} column is the eigenvector \mathbf{V}^j of \mathbf{A} , and $\mathbf{\Lambda}$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues of \mathbf{A} .

Singular Values:

Let A be a m by n matrix having real entries. Consider the matrix $A^T A$. So, $A^T A$ will be a n by n matrix which is symmetric and positive semi definite, that is, all the eigenvalues of $A^T A$ are non-negative.

Now suppose Eigen values of A are $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$ such that ,

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_n \geq 0$$

Let $\sigma_i = \sqrt{\lambda_i}$

$$\text{i.e., } \sigma_1 \geq \sigma_2 \geq \sigma_3, \dots, \sigma_n \geq 0$$

The numbers $\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_n$ are called singular values of A .

2.2 Singular Value Decomposition:

Let A be a m by n matrix with rank r . Here in machine learning usually we play with real data. So, here we are assuming that entries are real numbers. The singular value decomposition of A ,

$$A = U \Sigma V^T ;$$

where, U is a m by m matrix or better to write m by m orthogonal matrix such that $U^T = U^{-1}$ or columns of U are pair wise orthonormal. V is a n by n orthogonal matrix.

Σ is a m by n matrix, where diagonal elements of first r rows are singular values of A and rest of the entries are 0.

Matrices U and V

The columns of V are orthonormal Eigen vectors $v_1, v_2, v_3, \dots, v_n$ of the n by n positive semi definite matrix, $A^T A$. So for $i = 1, 2, 3, \dots, n$ we have,

$$A^T A v_i = \sigma_i^2 v_i$$

Similarly columns of U are orthonormal Eigen vectors $u_1, u_2, u_3, \dots, u_m$ of the m by m positive semi definite matrix AA^T , So for $j = 1, 2, 3, \dots, m$ we have,

$$AA^T u_j = \sigma_j^2 u_j$$

Example 2.1 :

Find the SVD of $A = \begin{pmatrix} 0 & 1 & 1 \\ \sqrt{2} & 2 & 0 \\ 0 & 1 & 1 \end{pmatrix}$

Step 1 (Finding the matrix U)

Here $AA^T = \begin{pmatrix} 2 & 2 & 2 \\ 2 & 6 & 2 \\ 2 & 2 & 2 \end{pmatrix}$

Now find the Eigen values of AA^T and arrange them in descending order,

i.e., $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$

On computing, we get the Eigen values of AA^T as 8,2,0

i.e., $\lambda_1 = 8, \lambda_2 = 2, \lambda_3 = 0$

Hence $\sigma_1 = 2\sqrt{2}, \sigma_2 = \sqrt{2}, \sigma_3 = 0$

Now the Eigen vectors corresponding to the Eigen values of AA^T are :

$$\lambda_1 = 8: \left(\frac{1}{\sqrt{6}} \quad \frac{2}{\sqrt{6}} \quad \frac{1}{\sqrt{6}} \right)^T$$

$$\lambda_2 = 2: \left(\frac{1}{\sqrt{3}} \quad \frac{1}{\sqrt{3}} \quad -\frac{1}{\sqrt{3}} \right)^T$$

$$\lambda_3 = 0: \left(\frac{1}{\sqrt{2}} \quad 0 \quad -\frac{1}{\sqrt{2}} \right)^T$$

$$\text{So, } U = \begin{pmatrix} \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Step 2 (Finding the matrix V)

Here we consider the matrix $A^T A$,

$$A^T A = \begin{pmatrix} 2 & 2\sqrt{2} & 0 \\ \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

On simple computation we get Eigen values of $A^T A$ are : 8,2,0

Now the Eigen vectors corresponding to the Eigen values of $A A^T$ are :

$$\mu_1 = 8 : (\sqrt{2} \ 3 \ 1)^T$$

$$\mu_2 = 2 : \left(-\frac{1}{\sqrt{2}} \ 0 \ 1\right)^T$$

$$\mu_3 = 0 : (\sqrt{2} \ -1 \ 1)^T$$

$$\text{Therefore } V = \begin{pmatrix} \sqrt{2} & -\frac{1}{\sqrt{2}} & \sqrt{2} \\ 3 & 0 & -1 \\ 1 & 1 & 1 \end{pmatrix}$$

Step 3 (Finding the Matrix Σ)

We know Σ is a m by n matrix, where diagonal elements of first r rows are singular values of A and rest of the entries are 0, where r is the rank of the matrix A.

Here rank of A = 2 , so we need only consider the singular values $\sigma_1 = 2\sqrt{2}$ and $\sigma_2 = \sqrt{2}$

$$\text{Hence the matrix } \Sigma = \begin{pmatrix} 2\sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

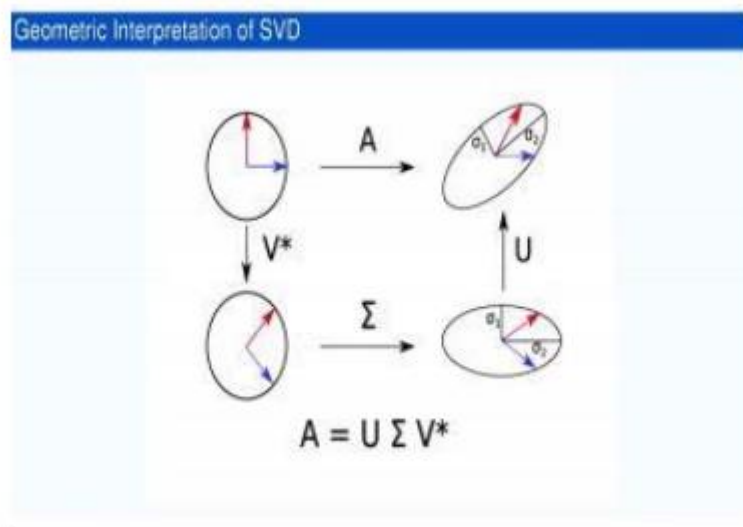
Thus we get the singular value decomposition of the given matrix

$$\text{i.e., } \begin{pmatrix} 0 & 1 & 1 \\ \sqrt{2} & 2 & 0 \\ 0 & 1 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{pmatrix} \begin{pmatrix} 2\sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \sqrt{2} & 3 & 1 \\ -\frac{1}{\sqrt{2}} & 0 & 1 \\ \sqrt{2} & -1 & 1 \end{pmatrix}$$

2.3 Properties and Applications of SVD

Geometric Interpretation of SVD



Suppose we have a unit circle and these two vectors; one is blue and another one is red. If we apply a transformation A which may be a rectangular matrix also, then this particular circle transformed into this ellipse, having this orientation. As we know that singular value decomposition is

$$A = U \Sigma V$$

First we will apply V^T on it. So, once we apply V^T on it, which is an orthogonal matrix because V is an orthogonal matrix. It will rotate this object by some angle based on the value of V transpose. Now we are having the same circle,

but orientation is different. Now, we will apply Σ on it. So, Σ is having scaling factors and though these scaling factors are proportional to singular values, so it will scale this circle and it will deform into an ellipse. If both singular values are equal i.e., $\sigma_1 = \sigma_2$, it will remain as a circle otherwise it will become an ellipse. Now apply U on it. U is again an orthogonal matrix and it will rotate this ellipse as show in the above figure. In this way, we can interpret singular value decomposition, that it is a sequence of transformation; first rotation, then scaling and then rotation. So, this is the geometrical interpretation of singular value decomposition.

Properties

A be a m by n real matrix means having real entries and rank of A equals to r .

Let us consider that the singular value decomposition of A is $U \Sigma V^T$. Now, if the rank A equals to r , then the first r singular values of A will be nonzero. That is rank of matrix A equals to number of nonzero singular values. **The range space of A** is given by the first r columns of the matrix U . **The null space of A** is given by the last $n - r$ columns of V that is the solution space of AX equals to 0 .

Now consider A^T ,

$$A^T = (U \Sigma V^T)^T = V \Sigma^T U^T$$

now what I want to say that the in the similar way that the range space of A^T

is given by the first r columns of V and the null space of A^T is given by last $m-r$ columns of U .

Another application is the relation between SVD and pseudo inverse

Let A is m by n matrix. So, here U will be m by m orthogonal matrix and V will be n by n orthogonal matrix and Σ is a m by n matrix. Let rank of A equals to r , then

$\sigma_1 \geq \sigma_2 \geq \sigma_3, \dots, \sigma_r > 0$ and rest of the singular values = 0.

Because A is a rectangular matrix.

$$A^\dagger = (U \Sigma V^T)^\dagger$$

$$= (V^T)^{-1} \Sigma^\dagger U^{-1}$$

$$= (V^{-1})^{-1} \Sigma^\dagger U^{-1} \text{ [since V is orthogonal, } V^T = V^{-1} \text{]}$$

$$= V \Sigma^\dagger U^T \text{ [since U is orthogonal, } U^T = U^{-1} \text{]}$$

So, if U and V transpose are given you can easily find out V and U transpose.

Calculating Σ^\dagger

Suppose A is 3 by 3 square matrix, with singular values $\sigma_1, \sigma_2, \sigma_3$. But ($\sigma_3 = 0$)

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ then } \Sigma^\dagger = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 \\ 0 & \frac{1}{\sigma_2} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Suppose A is 3 by 5 matrix, with singular values $\sigma_1, \sigma_2, \sigma_3$. But ($\sigma_3 = 0$)

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ then } \Sigma^\dagger = \begin{bmatrix} 1/\sigma_1 & 0 & 0 & 0 & 0 \\ 0 & 1/\sigma_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Suppose A is 5 by 3 matrix, with singular values $\sigma_1, \sigma_2, \sigma_3$. But ($\sigma_3 = 0$)

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ then } \Sigma^\dagger = \begin{bmatrix} 1/\sigma_1 & 0 & 0 \\ 0 & 1/\sigma_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Example : Find the pseudo-inverse of $A = \begin{pmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{pmatrix}$

The SVD of A =

$$\begin{pmatrix} -3/\sqrt{10} & 1/\sqrt{10} \\ -1/\sqrt{10} & -3/\sqrt{10} \end{pmatrix} \begin{pmatrix} 6\sqrt{10} & 0 & 0 \\ 0 & 3\sqrt{10} & 0 \end{pmatrix} \begin{pmatrix} -1/3 & -2/3 & -2/3 \\ -2/3 & -1/3 & 2/3 \\ -2/3 & 2/3 & -1/3 \end{pmatrix}^T$$

$$A^\dagger = \begin{pmatrix} -1/3 & -2/3 & -2/3 \\ -2/3 & -1/3 & 2/3 \\ -2/3 & 2/3 & -1/3 \end{pmatrix} \begin{pmatrix} 1/6\sqrt{10} & 0 \\ 0 & 1/3\sqrt{10} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} -3/\sqrt{10} & 1/\sqrt{10} \\ -1/\sqrt{10} & -3/\sqrt{10} \end{pmatrix}^T$$

$$A^\dagger = \begin{pmatrix} -0.0056 & 0.0722 \\ 0.0222 & 0.0444 \\ 0.0556 & -0.0556 \end{pmatrix}$$

$$AA^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

if we calculate here A times pseudo inverse of A, we will get the identity matrix of order 2, that is why it is the right pseudo inverse here in this case. So, this is another application of singular value decomposition, i.e., calculating pseudo inverse.

Other Applications of Singular Value Decomposition (SVD)

- o Image Compression
- o Image Recovery
- o Eigen faces
- o Spectral Clustering
- o Background Removal from Videos

2.4 POLAR DECOMPOSITION

Any complex number z in polar form, can be written as $z = r e^{i\theta}$, $r = \sqrt{x^2 + y^2}$, $\theta = \tan^{-1}(y/x)$. Now, if we see here r is always non negative and $e^{i\theta}$ gives an orientation. In the same manner we want some factorization of a matrix

somewhat similar to the what we have seen for the complex number z . Now this is the motivation for **Polar Decomposition** of a matrix. We are having many applications of this polar decomposition like in machine learning , in computer graphics and so on. Because it is a matrix factorization something like in terms of an orthogonal matrix and a positive semi definite matrix. And that you can utilize in many of the learning algorithm.

Theorem 2.4.1

For any square matrix A , let us say A belongs to the vector space of n by n real matrices, there exists an orthogonal matrix W and a positive semi definite matrix P such that $A = W P$. So, here W is an orthogonal matrix means; the columns of W are orthonormal and P is a positive semi definite matrix.

Furthermore, if A is an invertible matrix, then the factorization is unique; means this kind of decomposition is unique. Such a decomposition of any matrix $A = W P$ is called **Polar Decomposition**.

Proof

From the SVD of A , we have

$$\begin{aligned} A &= USV^T \\ &= UV^T V S V^T \text{ (as } V \text{ is orthogonal, } V^T = I_n \text{)} \\ &= W P \end{aligned}$$

Here $W = UV^T$ and is an orthogonal matrix, because product of two orthogonal matrices is orthogonal.

and $P = V S V^T$, which is a Positive Semi Definite Matrix, because

$P = V S V^T$, implies that the matrices P and S are similar. Thus Eigen values of both the matrices will be the same. All the Eigen values of S are non-negative,

Example 2.4.1:

Find Polar Decomposition of the matrix $A = \begin{pmatrix} 11 & -5 \\ -2 & 10 \end{pmatrix}$

The SVD of $A = USV^T$

$$= \begin{pmatrix} 4/5 & 3/5 \\ -3/5 & 4/5 \end{pmatrix} \begin{pmatrix} 10\sqrt{2} & 0 \\ 0 & 5\sqrt{2} \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}^T$$

Polar Decomposition of $A = WP$

$$W = UV^T = \begin{pmatrix} 4/5 & 3/5 \\ -3/5 & 4/5 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}^T = \frac{1}{5\sqrt{2}} \begin{pmatrix} 7 & -1 \\ 1 & 7 \end{pmatrix}$$

$$P = VSV^T = \frac{5}{\sqrt{2}} \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}$$

$$\text{So, } A = WP = \frac{1}{5\sqrt{2}} \begin{pmatrix} 7 & -1 \\ 1 & 7 \end{pmatrix} \cdot \frac{5}{\sqrt{2}} \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}$$

Right Polar Decomposition

Let A be a m by n matrix, where $m \geq n$. Then, the right polar decomposition of A is $A = UP$; where U is a m by n matrix with orthonormal columns and P is a n by n positive semi definite matrix.

Left Polar Decomposition

The another decomposition is left polar decomposition. So, let A be a m by n matrix where $m \leq n$, then the left polar decomposition of A is $A = HU$; where H is a m by m positive semi definite matrix and U is a m by n matrix having orthonormal columns.

Example 1

Find polar decomposition of $A = \begin{pmatrix} 3 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$

Here we will be having right polar decomposition. i.e., $A = UP$

Note: In right decomposition, the procedure to find Matrix P is as follows:

$$P = \sqrt{A^T A}$$

$A^T A$ will be a $n \times n$ square matrix, hence it will have **Matrix Eigen**

decomposition.

$A^T A = T Q^{1/2} T^{-1}$, where T is the square matrix whose j^{th} column is the eigenvector V^j of $A^T A$ and Q is the diagonal matrix whose diagonal elements are the corresponding eigenvalues of $A^T A$.

$$\text{So, } \sqrt{A^T A} = T Q^{1/2} T^{-1}$$

$$\text{i.e, } P = T Q^{1/2} T^{-1}$$

$$\text{Finally, } U = AP^{-1}$$

So, in this problem, $A^T A = \begin{pmatrix} 10 & 3 \\ 3 & 2 \end{pmatrix}$. Now use Matrix Eigen Decomposition and

decompose the matrix $\begin{pmatrix} 10 & 3 \\ 3 & 2 \end{pmatrix}$ into $T Q T^{-1}$

On computing, we get

$$\begin{pmatrix} 10 & 3 \\ 3 & 2 \end{pmatrix} = \begin{pmatrix} 3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & -3/\sqrt{10} \end{pmatrix} \begin{pmatrix} 11 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & -3/\sqrt{10} \end{pmatrix}^T$$

$$\text{Hence, } P = \sqrt{A^T A} = T Q^{1/2} T^{-1}$$

$$\text{Now } U = AP^{-1}$$

CHAPTER 3

LOW RANK APPROXIMATION

In mathematics, low-rank approximation is a minimization problem, in which the cost function measures the fit between a given matrix (the data) and an approximating matrix (the optimization variable), subject to a constraint that the approximating matrix has reduced rank. The problem is used for mathematical modelling and data compression. The rank constraint is related to a constraint on the complexity of a model that fits the data. In applications, often there are other constraints on the approximating matrix apart from the rank constraint, e.g., non-negativity and Hankel structure.

Definition

$M_{m \times n}(R)$ denotes the set of all $m \times n$ matrices with real entries. Rank of a matrix can be defined as,

The number of linearly independent rows or columns of A

- Order of largest non-singular submatrix of A
- Dimension of row or column space of A
- The number of non-zero rows in row reduced echelon form of A
- The order of identity submatrix in the normal form of A
- The rank of the linear transformation from R^n to R^m corresponding A
- Usually denoted by $\rho(A)$

SUPPOSE r be the rank of A then, that suggests that dimension of column space of A is r , implies there exist a basis consisting of r linearly independent vectors that spans the column space of A .

Let $B = \{b_1, b_2, b_3, \dots, b_r\}$ be such a basis for the column space of A .

Then by definition, columns of A can be uniquely expressed as linear combination of members in B .

Say, Let a_1, a_2, \dots, a_n be n columns of A

Then,

$$a_1 = c_{11} b_1 + c_{21} b_2 + \dots + c_{r1} b_r \text{ Type equation here.}$$

$$a_2 = c_{12} b_1 + c_{22} b_2 + \dots + c_{r2} b_r$$

.....

$$a_n = c_{1n} b_1 + c_{2n} b_2 + \dots + c_{rn} b_r$$

$$[\mathbf{a}_1 , \mathbf{a}_2 , \dots , \mathbf{a}_n] = [\mathbf{b}_1 , \mathbf{b}_2 , \dots , \mathbf{b}_r] \begin{bmatrix} \mathbf{C}_{11} & \dots & \mathbf{C}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{r1} & \dots & \mathbf{C}_{rn} \end{bmatrix}$$

So, we are saying that if A is a m by n matrix, rank of A equals to r then we can have a factorization of A as , $A = B \times C^T$, means we are writing the columns of A as the linear combination of product of the basis of the range space of A and then a matrix C; and here B is m by r matrix because, you are having r vectors each one of m dimension, and C is a n by R matrix. Then this particular product will be well defined.

IMPORTANT POINTS TO REMEMBER

Number of elements in A = mn

Number elements in $B * C^T = m.r + n.r = (m+n) . r$, where $r = \rho(A)$

Now, consider a situation where, where A is a 50 by 100 matrix with rank of A equals to let us say 20. So, how many elements you have to store here. For writing this matrix means you have to store 50 into 100 that is 5000 elements; however, in this case how many elements you require to save? 50 into 20 that is 1000 plus 100 into 20 that is 2000, so only 3000. This is the motivation of low rank approximation. So, in original case you have to save 5000 elements, but in this case you have to save only 3000 elements, and you are saving the same thing that is the matrix A. So, in that **way you can save the memory as well as**

the computational cost because, if you want to operate something on this matrix A, you have to process all the 5000 elements means, if it is an digital image you will be having 5000 pixels in A, but in case of if you are writing it in this way B into C transpose then, you have to play with only 3000 pixels. This is a motivation for low rank approximation. Given a matrix or a tensor you find out another matrix or tensor which is having the lower rank to the original one and the same time it is quite close to the original one.

Let $A \in M_{m \times n}$ (often large), having $\text{rank}(A) \leq \min\{m, n\}$. The low rank approximation of A is to find another matrix $A_k \in M_{m \times n}$ which is having rank $k \leq r = \text{rank}(A)$ and approximate A.

To find the “best” A_k , we must define how closely A_k approximates A. The simplest metric is the Frobenius norm of $A - A_k$. This criterion leads to the following low rank approximation problem.

$$\|A - A_k\| = \min \{ \|A - A_k\|_F : A_k \in M_{m \times n} \text{ and } \rho(A_k) \leq k \}$$

Now, best-k approximation to A is given as $A^k = \sum_{i=1}^k \sigma_i u_i v_i^T$, where u_i and v_i denote the i th column of U and V respectively in the SVD of A.

In the sense that,

$$\|A - A_k\| \leq \|A - \tilde{A}\|$$

For any \tilde{A} in $R^{m \times n}$ with $\text{rank}(\tilde{A}) \leq k$

Measure of Quality of Approximation

The Measure of Quality of Approximation is given by,

$$\|A_k\|_F^2 / \|A\|_F^2 = \frac{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \dots + \sigma_k^2}{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \dots + \sigma_r^2}$$

Example:

Find the rank 2 approximation of $A = \begin{pmatrix} 3 & 2 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

Solution : The SVD of A is given by , $A = USV^T$

$$\begin{pmatrix} 0.91 & 0.42 & 0.02 \\ 0.42 & -0.87 & -0.26 \\ 0.09 & -0.24 & 0.97 \end{pmatrix} \begin{pmatrix} 4.04 & 0 & 0 \\ 0 & 1.70 & 0 \\ 0 & 0 & 0.87 \end{pmatrix} \begin{pmatrix} 0.67 & 0.73 & 0.08 \\ 0.65 & -0.54 & -0.53 \\ 0.35 & -0.41 & 0.84 \end{pmatrix}^T$$

The rank 2 approximation of A is given by A_2

$$= \begin{pmatrix} 0.91 & 0.42 \\ 0.42 & -0.87 \\ 0.09 & -0.24 \end{pmatrix} \begin{pmatrix} 4.04 & 0 \\ 0 & 1.70 \end{pmatrix} \begin{pmatrix} 0.67 & 0.73 \\ 0.65 & -0.54 \\ 0.35 & -0.41 \end{pmatrix}^T$$

$$= \begin{pmatrix} 2.99 & 2.01 & 0.98 \\ 0.02 & 1.88 & 1.1 \\ -0.07 & 0.45 & 0.29 \end{pmatrix}$$

CHAPTER 4

PRINCIPLE COMPONENT ANALYSIS

It is a very popular and very important concept in machine learning for reducing the dimension of the data that is called Principal Component Analysis.

So, it is a manifold learning technique and really very applicable among machine learning research.

what is this PCA? So, take a very simple example.

Suppose, we are having some data of five cities. So, cities are C_1, C_2, C_3, C_4 and C_5 and 4 attributes or parameters and they are Education, Transport, Entertainment and finally Safety, now based on these 4 parameters we are going to judge the best city. Suppose we have graded all those parameters education, transport facility, entertainment opportunity and safety on a 10 pointy scale.

Consider following table;

	$X_1 = \text{Education}$	$X_2 = \text{Transportation}$	$X_3 = \text{Entertainment}$	$X_4 = \text{Safety}$
C_1	8	6	9	7
C_2	5	7	8	10
C_3	4	7	6	5
C_4	6	7	6	6
C_5	10	7	4	10

Now, we have to classify all those cities based on these parameters. But what is happening here, suppose we do not want to classify with all four features. So, instead of these four features, four attributes, education, transport, entertainment and safety, we want three features so that we can plot a threedimensional plot, we can have for these all five city data and then, I can classify them by using some hyper plane in R^3 . Essentially, what we need is to transform a 4 dimensional space into a 3 dimensional one. i.e. to transform R^4 to R^3 .

Any vector in R^4 will be of the form $\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix}$ and that of R^3 will be $\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix}$, so the

required transformation can be achieved if we can find a 3×4 matrix A such that,

$$A \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix}, \text{ where } A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}$$

This way we can transform four-dimensional data set to a three-dimensional data

set or suppose, we want to go to two-dimensional data set it will become a 2 by 4 matrix. In short, our aim is to reduce the dimension of data without compensating on its content or information. For example, here if we say for classification which of the feature vector is not having much information. The vector or the column in which I am having minimum variation and that is the column corresponding to the parameter transportation. So, if you remove this column, then it will not make any because all the cities are having almost same value. So, it will not make any difference in the classification. **PCA principal component analysis** is a tool for doing this kind of dimension reduction.

Definitions

Mean: Given a data set $X = \{x_1, x_2, \dots, x_n\}$. The arithmetic mean is the most commonly used and readily understood measure of central tendency in a data set. In statistics, the term average refers to any of the measures of central tendency. The arithmetic mean of a set of observed data is defined as being equal to the sum of the numerical values of each and every observation, divided by the total number of observations.

Standard Deviation: It is a measure of the amount of variation or dispersion of a set of values. A low standard deviation indicates that the values tend to be close to the mean of the set, while a high standard deviation indicates that the values are spread out over a wider range.

Covariance: Covariance is a measure of the relationship between two random variables and to what extent, they change together.

4.1 Procedure for Performing Principal Component

Analysis

Step 1: Data Set

<u>Features or attributes</u>	<u>Example 1</u>	<u>Example 2</u>	<u>.....</u>	<u>Example m</u>
X_1	X_{11}	X_{12}	<u>.....</u>	X_{1m}
X_2	X_{21}	X_{22}	<u>.....</u>	X_{2m}
.
.
.
.
.
.
X_n	X_{n1}	X_{n2}	<u>.....</u>	X_{nm}

Step 2: Compute the means of the variables

Mean of $X_i = \bar{X}_i = 1/m \sum_{k=1}^m X_{ik}$

Step 3: Calculate Covariance Matrix

Find covariance of all ordered pairs (X_i, X_j)

$$\text{Cov}(X_i, X_j) = 1/m-1 \sum_{k=1}^m (X_{ik} - \bar{X}_i)(X_{jk} - \bar{X}_j)$$

Construct an $n \times n$ matrix S and this matrix is called **Covariance matrix**.

$$S = \begin{pmatrix} \text{Cov}(X_1, X_1) & \dots & \text{Cov}(X_1, X_n) \\ \vdots & \ddots & \vdots \\ \text{Cov}(X_n, X_1) & \dots & \text{Cov}(X_n, X_n) \end{pmatrix}$$

this particular matrix tells you, if you are having n -dimensional data in which direction it is having maximum variation. It is symmetric matrix. And if it is symmetric matrix, it will be having real eigenvalues and you will be having always orthogonal decomposition, means you can have orthogonal eigenvectors of this matrix.

Step 4: Calculate Eigen values and normalized Eigen vectors of the Covariance matrix.

Definition: The **Principal Components** are the eigenvectors of the covariance matrix of the data. **First Principal Component** is the eigenvector corresponding to largest eigenvalue of the covariance matrix. The meaning of first principal component is that, in the direction of that vector the dataset will be having the **maximum variability**. So, using this fact, if we want to transform a n dimensional dataset to a k -dimensional dataset, ($k < n$) then we will select first k principal components. We will get k eigenvectors; those are orthogonal because the covariance matrix is a symmetric matrix. Those k orthogonal eigenvectors will expand a k -dimensional space and then your data will become k -dimensional.

So, this is the overall idea of principal component analysis.

Step 5: Derive New Data Set

In the new data set with reduced dimension, the new entries will be P_{ij}

corresponding to each X_{ij} .

And $P_{ij} = e_i^T$ $\begin{bmatrix} X_{1j} - \bar{X}_1 \\ X_{2j} - \bar{X}_2 \\ \cdot \\ \cdot \\ X_{nj} - \bar{X}_n \end{bmatrix}$ and i represents the i th Principal component

CONCLUSION

Machine Learning is a division of Artificial Intelligence that focuses on building applications by processing available data accurately. The primary aim of machine learning is to help computers process calculations without human intervention. Machine learning is one of the most popular topics of nowadays research. This particular topic is having applications in all the areas of engineering and sciences. Various tools of machine learning are having a rich mathematical theory. Therefore, in order to develop new algorithms of machine/deep learning, it is necessary to have knowledge of all such mathematical concepts.

BIBLIOGRAPHY

1. Gilbert Strang, Introduction to Linear Algebra Fourth Edition, CENGAGE learning, 2005
2. G.W. Thomas Mathematics for Machine Learning, University of California, 2018
3. Ward Cheney, Analysis for Applied Mathematics, New York: Springer Science and Business Medias, 2001.
4. S. Axler, Linear Algebra Done Right (Third Edition): Springer International Publishing, 2015