Essentials of Mathematics in Data Science

R. C. Mittal
Jaypee Institute of Information Technology

Essentials of Mathematics in Data Science

R.C. Mittal

(Formerly Prof. IIT Roorkee)

Department of Mathematics

Jaypee Institute of Information Technology

Sector 62 NOIDA (U.P.)

Outlines

- Mathematics Needed in Data Science
- Linear Regression
- Principal Component Analysis
- Singular Value Decomposition
- Power Method
- Linear Classification

Need of Mathematics

- Differential Calculus
- Statistics and Sampling Theory
- Optimization
- Linear Algebra
  - Dimension
  - Eigen Values and Eigen Vectors
  - Inner Product and Norms
- Soft Technologies
  - Neural Network
  - Support Vector Machine

Linear Regression

Suppose we have a table with two columns

X	Y
x_1	y_1
x_2	y_2
$\vdots$	
$\vdots$	y_n
x_m	

From this data we wish to predict the value y from given x.

We use a linear regression

$$y = ax + b \qquad (1)$$

where a and b are unknown. We assume all given points in the table lie near the line (1). So approximately

$$y_i = ax_i + b \quad \ldots (2)$$

for $i = 1 \cdots m$

So $\sum_{i=1}^{m} y_i = a \sum_{i=1}^{m} x_i + mb$

or $\bar{y} = a\bar{x} + b$

where $\bar{x} = \frac{\sum x_i}{m}$, $\bar{y} = \frac{\sum y_i}{m}$

So means $(\bar{x}, \bar{y})$ lie on the line.

So $\bar{y} = a\bar{x} + b$

or $y - \bar{y} = a(x - \bar{x})$ -- (3)

Therefore each set of data in the table, we have

$$y_i - \bar{y} = a(x_i - \bar{x}) \quad \text{-- (4)}$$

We define the vectors

$$X = (x_1 - \bar{x}, x_2 - \bar{x}, \dots, x_m - \bar{x})$$

$$Y = (y_1 - \bar{y}, y_2 - \bar{y}, \dots, y_m - \bar{y})$$

Using (4) for each component

$$Y = aX$$

So $a = \dfrac{X \cdot Y}{X \cdot X} = \dfrac{X \cdot Y}{\|X\|^2}$ -- (5)

Also $\bar{y} = a\bar{x} + b$ -- (6)

So $b = \bar{y} - a\bar{x}$

Now from (5) and (6) the line (1) is known

So $y = ax + b$ can be used to predict y for given x.

n-dimension linear regression

Consider a table

X_1	X_2	$---$	X_m
x_{11}	x_{12}		x_{1m}
x_{21}	x_{22}		x_{2m}
$\vdots$			
x_{n1}	x_{n2}	$--$	x_{nm}

We want to use a linear predictor of the form

$$y = b + w_1 x_1 + w_2 x_2 + \cdots + w_m x_m$$

where $x_1, x_2 - x_m$ are independent variables. Then approximately

$$y_1 = b + w_1 x_{11} + w_2 x_{12} + \cdots + w_m x_{1m}$$

$$y_2 = b + w_1 x_{21} + w_2 x_{22} + \cdots + w_m x_{2m}$$

$$\vdots$$

$$y_n = b + w_1 x_{n1} + w_2 x_{n2} + \cdots + w_m x_{nm}$$

In matrix form, we can write this as

$$Y = X \begin{bmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} \quad \cdots (7)$$

where

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1m} \\ 1 & x_{21} & x_{22} & \cdots & x_{2m} \\ 1 & & & & \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

Here X is $n \times (m+1)$ matrix

Now multiplying $\boxed{7}$ by X^T both sides we have

$$X^T Y = X^T X \begin{bmatrix} b \\ w \end{bmatrix} \quad \cdot - \boxed{8}$$

The system $\boxed{8}$ can easily be solved to get unknown vector b and $w_1, w_2 - w_m$.

Then the predictor

$$y = b + w_1 x_1 + - + w_m x_m$$

now can be used to predict the values.

Singular Value Decomposition

Suppose A is a real mxn matrix, then singular values of A are nothing but square roots of non-negative eigen values of A^TA . Here A^TA is a self adjoint matrix.

Example- Consider the matrix

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 5 & 4 \end{bmatrix} \qquad (1)$$

$$\text{So } A^T = \begin{bmatrix} 1 & 2 & 5 \\ 2 & 3 & 4 \end{bmatrix} \quad \text{and } A^TA = \begin{bmatrix} 30 & 28 \\ 28 & 29 \end{bmatrix} \qquad (2)$$

Eigen values of A^TA are (59 +/- 56.009)/2 i.e. 57.504 and 1. 495.

Therefore singular values of A are = 7.583 and 1.223 .

In singular value decomposition of the matrix A, we decompose the matrix A into UDV^T , U is a mxm orthonormal matrix, V is a nxn orthonormal matrix and D is a mxn matrix with rxr diagonal matrix containing singular values of A.

Here r is the rank of the matrix A.

Principal Component Analysis

Today a lot of data is generated daily in different social sites. In fact 90% of today data is generated in the last 3-4 years. This data is to be properly analyzed to gather important and relevant information.

PCA is a dimension reduction technique which give most dominant features from the data. These newly extracted features are known as "Principal Components" .

Key Points of PCA

- A principal component is a linear combination of the original variables
- Principal components are extracted in such a way that the first principal component explains maximum variance in the dataset
- Second principal component tries to explain the remaining variance in the dataset and is uncorrelated to the first principal component
- Third principal component tries to explain the variance which is not explained by the first two principal components and so on

It may be noted that each additional dimension we add to the PCA technique captures less and less of the variance in the model. The first component is the most important one, followed by the second, then the third, and so on.

How to find Principal Components?

Following steps are used in PCA

(1) Normalization of Data – Compute the mean μ and standard deviation σ .

Now normalize the data in [0,1] by using formula

$$z = \frac{x - \mu}{\sigma}$$

2. Computation Covariance Matrix- The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information.

So, in order to identify these correlations, we compute the covariance matrix

$$\begin{pmatrix} \operatorname{cov}(x,x) & \operatorname{cov}(x,y) & \operatorname{cov}(x,z) \\ \operatorname{cov}(x,y) & \operatorname{cov}(y,y) & \operatorname{cov}(y,z) \\ \operatorname{cov}(x,z) & \operatorname{cov}(y,z) & \operatorname{cov}(z,z) \end{pmatrix} \qquad (4)$$

This is a symmetric matrix.

Cov(x, x) = var(x) etc.

3. Computation of Eigen values and Eigen Vectors-

Now compute the eigen values and corresponding eigen vectors of the covariance matrix (4). You will find some eigen values are very small in magnitude . Precisely these eigen values can be neglected. Thus reducing the dimension.

Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components.

So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on.

Power Method for dominant eigen-value:

To get the most prominant feature of a data set we apply the Power Method to get the largest in magnitude eigen-value and corresponding eigen-vector.

Suppose we have a matrix of order $n \times n$ obtained from the n features from the data set.

The dominating eigen-value is λ_1 (say) i.e.

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \cdots \geq |\lambda_n|$$

Let the corresponding eigen-vector be $\bar{v}_1, \bar{v}_2, \ldots \bar{v}_n$.

Now any vector $\bar{v}$ can be represented as a linear combination of $\bar{v}_1, \bar{v}_2, \bar{v}_3, \ldots \bar{v}_n$ so

$$\bar{v} = c_1 \bar{v}_1 + c_2 \bar{v}_2 + \cdots + c_n \bar{v}_n$$

$$A\bar{v} = c_1 A \bar{v}_1 + c_2 A \bar{v}_2 + \cdots + c_n A \bar{v}_n$$

or $A\bar{v} = c_1\lambda_1\bar{v}_1 + c_2\lambda_2\bar{v}_2 + \dots + c_n\lambda_n\bar{v}_n$

$$= \lambda_1\left(c_1\bar{v}_1 + c_2\frac{\lambda_2}{\lambda_1}\bar{v}_2 + \dots + c_n\frac{\lambda_n}{\lambda_1}\bar{v}_n\right)$$

Then $A^2\bar{v} = \lambda_1^2\left(c_1 v_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^2 \dots + \dots + c_n\left(\frac{\lambda_n}{\lambda_1}\right)^2\bar{v}_n\right.$

$\vdots$

$$A^r\bar{v} = \lambda_1^r\left(c_1 v_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^r\bar{v}_2 + \dots + c_n\left(\frac{\lambda_2}{\lambda_1}\right)^r\bar{v}_n\right)$$

Since $\frac{\lambda_i}{\lambda_1} < 1$ for $i=2, \dots n$

As $r\to\infty$, $\frac{\lambda_i}{\lambda_1}\to 0$ $i=2 \dots n$

Therefore $A^r\bar{v}\to c_1\lambda_1\bar{v}_1$ as $r\to\infty$

This gives us an idea to get λ_1 easily.

Select an ~~a~~ non-zero vector $\bar{v}_0$ and find $\bar{v}_1 = A\bar{v}_0$, take the maximum ~~o~~ magnitude component of $\bar{v}_1$ to normalize it.

So we get normalized value of v_1

My iterative Scheme will be $v^{(n+1)} = A v^{(n)}$

The maximum component of v^{n+1} → the eigen value and v_{n+1} is corresponding eigen-vector.

Classification
Linear Classifier

Unlike in linear regression where we predict a numerical value, here we predict a class such as winner or loser; rich or poor etc.

This is one of the important problem in data analysis. We may predict

(i) A team will win or not ?

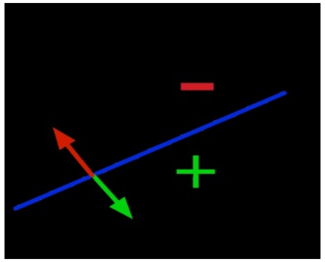
(ii) An individual will like a movie or not ?

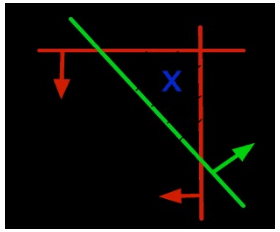
Our input here is a point $X \subseteq R^n$, where each element x in X also has an associated label y having value -1 or 1.

Suppose $X = (x_1 , x_2 , \ldots\ldots, x_n)$ is a point set and

$W = (w_1 , w_2 , \ldots\ldots, w_n)$ is a weight vector. We wish to design a linear predictor as follows

$$y = b + w_1x_1 + w_2x_2 + \ldots\ldots.. + w_nx_n$$

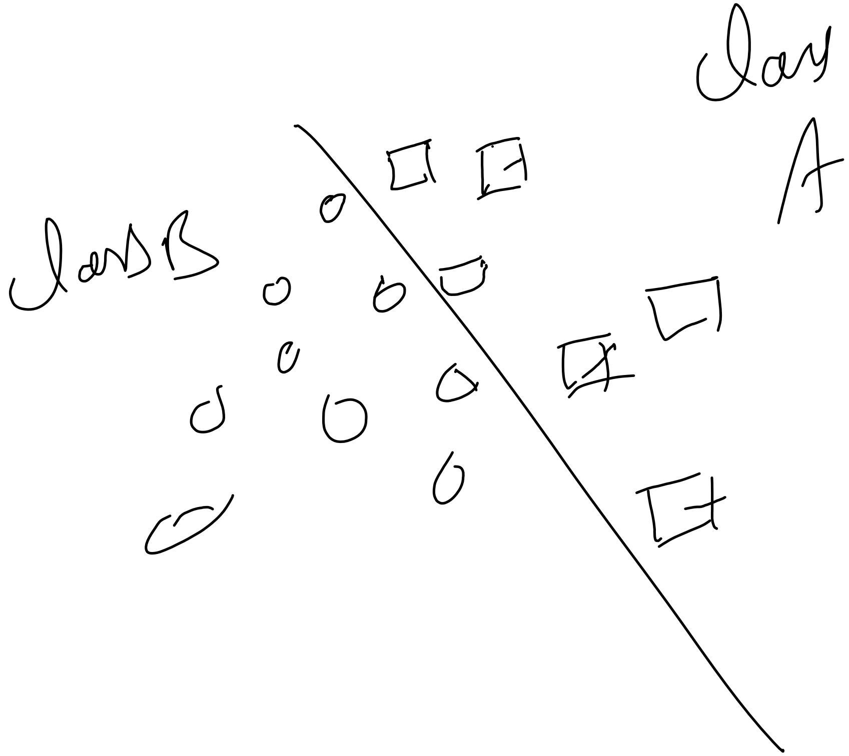




My linear predictor is

$$y = b + w_1x_1 + w_2x_2 + \dots + w_nx_n \quad (A)$$

Suppose we have two classes A and B such that $A \cap B = \phi$



The equation (A) represents a hyperplane. Geometrically b gives the length of the normal from the origin and $w_1, w_2, w_3 — w_n$ are some sort directions of the normal w.r.t. to axes

Now a point can lie after the line or on the line or before the line.

That is if we substitute the value of a new object in (A), y can be >0, or 0 or <0

Therefore we know that zoo positive value belongs to class A and negative value belongs to class B-

In order to design a linear predictor we choose some values of b and weight(s) $w_1, w_2 - w_n$.

Now for given set of data for 70% data we train the data as follows

(1) Find

$$z = b + \sum_{i=1}^{n} w_i x_i$$

If sign(z) and sign of target value (already known) are same then no changes in selected values as it identifies the right class.

Otherwise, modify

$$w_i = w_i - 1$$

Re-train the data.

Once you set the idea which weights must be more which should be less, the hyper plane can be obtained.

References

1. J.M. Phillips “ Mathematical Foundations for Data Analysis “ (2018)
2. R. C. Mittal “Some Applications of Linear Algebra in Computer Science” https://www.researchgate.net/publication/352180145_Some_ApplicAtionS_of_lineAr_AlgebrA_in_computer_Science
3. G. Strang , “Introduction to Linear Algebra” MIT Publication (2016)