Introduction

1. What is LA ?

Linear algebra simply means, converting real world problem to linear equations. The equation simply means $ax + by + cz + ... = 0$. In LA we study these equations and solve them, we find the value of coefficients (a, b, c). These are called linear because the maximum power of equation will be 1.

2. Application in ML

LA is most fundamental in ML. ML is data driven, means we need the data to train machine learning models. Now this data can be tabular, textual, images, videos or even audio. We need to represent data into numerical form, to do this we need to represent data as a vector. You must be familiar with vector. A vector is an array or a row matrix. We use vector everywhere in ML to represent data, to understand and manipulate vector we need to study linear algebra.

Second big reason is that, in ML we get higher dimensional data. Human brain can imagine unto 3 dimensions or 4 if you add time. But in high dimension (lets say 50) we can not imagine. At that time linear algebra becomes very useful, by linear algebra we can generalise problems to higher dimension. I.e., If we have equation in two-three dimension, we can generalise to n dimension.

Apart from these two reasons, there are many to study LA.

3. Vector

A vector is an array to store and represent data. It can be 1D means the array has only one element, 2D means array has 2 elements or nD. Here don't confuse n dimensional vector with n dimensional array, the n dimensional vector has shape of n x 1. In ML vector plays a key role, you can think of vector as data structure which can be manipulated for best outcomes. However vector in physics and maths is somehow different than in ML.

Vector in physics is a quantity that has magnitude and direction, like velocity. Vector in maths is the arrow from origin. Vector in ML somehow links to vector in maths. Like geometric intuition of vector in ML differs from dimension to dimension. Like in 1D it is point, 2D it is arrow, 3D it is plane and in nD it is hyperplane. A vector can be represented by arrow above its name like $\vec{a}$ or **a**.

Think of a ML regression problem to predict house price, here inputs are soft. Area, # of bedrooms and # of balconies. We can represent a datapoint as a vector like this : [1000, 3, 2]. So for n datapoints we can have n vectors of size 1 x 3. In ML the vectors will be column vector by default, unless specified.

4. Distance between vectors

The distance between two vectors tells us how similar / dissimilar vectors are. Like in k nearest neighbour algorithm the distance between vectors is used. Mainly two types of distances are there : 1. Euclidian distance and 2. Manhattan distance. The euclidian distance is the shortest

distance between two points, where as Manhattan distance is the distance to be travelled through each dimension. ED is computationally difficult than MH distance and ED is default distance in ML algorithms.

Let's say there are two vectors, A (a_x, a_y, a_z) and B (b_x, b_y, b_z). Now the euclidian distance will be :

$$Euclian\ Distance = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2}$$

The Manhattan distance is given by :

$$Manhattan\ Distance = |a_x - b_x| + |a_y - b_y| + |a_z - b_z|$$

5. Product of vector

The product of a vector can be with vector and scaler. The product of vector with scaler is called scaling of vector. It stretches or compresses the vector by keeping the direction same.

The product of vector with vector is interesting, it is divided into two categories, dot product and cross product. In ML DP is widely used. The DP between two vectors A and B (mentioned above) is given by

$$\mathrm{A} \cdot \mathrm{B} = a_x b_x + a_y b_y + a_z b_z$$

The dot product gives similarity between two vectors, more the vector be similar more will be the dot product. The DP is also defined by,

$$\mathrm{A} \cdot \mathrm{B} = |A|\,|B|\,cos\theta$$

Here θ is the angle between two vectors A and B. Notice that, as the angle between two vector decreases, their dot product increases, means they are more similar. Orthogonal vectors has no similarity and vector opposite to each other have maximum dissimilarity. Finally, vectors in same direction have most similarity.

This concept can be used in ML, think of a movie recommendation engine, you represent each movie by vector of 5 dimension (5 x 1 vector) now how you find similar movies.

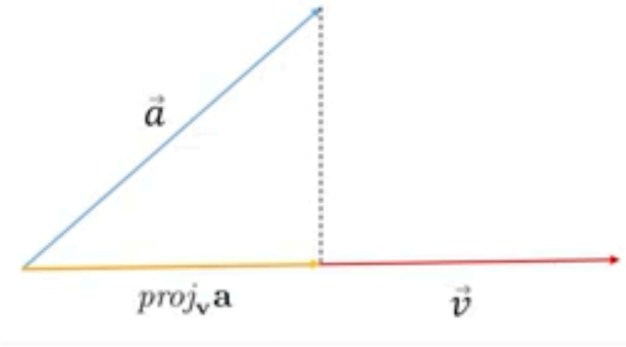
The second product is cross product. It is less widely used in ML. The CP of two vectors is the third vector perpendicular to area created by two vectors. The magnitude of CP is the value of area of two vectors, the direction is given by thumb rule. Below is the formula for cross product.

$$\mathrm{A} \times \mathrm{B} = \begin{vmatrix} \hat{i} & a_x & b_x \\ \hat{j} & a_y & b_y \\ \hat{k} & a_z & b_z \end{vmatrix}$$

$$|\mathrm{A} \times \mathrm{B}| = |A|\,|B|\,sin\theta$$

6. Projection of vector

Projection of a vector A on vector B, shows the magnitude and direction of vector A in the direction of B. The direction of PV is the direction of the vector on which the another vector is being projected. See the figure.



$$proj_{\mathbf{v}}\mathbf{a} = \frac{\mathbf{a} \cdot \mathbf{v}}{\|\mathbf{v}\|^2}\mathbf{v}$$

$proj_{\mathbf{v}}\mathbf{a}$ represents projection of **a** on **v**.

7. Vector, matrix & tensor

The 1D array is called vector, 2D array is called matrix and array of higher dimension is call tensor. Here don't get confused by 1D vector, here 1D is mentioned in terms of vector dimension, i.e., n x 1. Where as in ML high dimensional vector means n x 1 vector having n points, means n dimensions.