

Research Article

# Linear Algebra – A Powerful Tool for Data Science

Hasheema Ishchi

Lecturer in Mathematics Department, Education Faculty, Jawzjan University, Sheberghan, Afghanistan

Email Address: hasheema.ischi01@gmail.com

Analysis of data is an important task in data managements systems. Many mathematical tools are used in data analysis. A new division of data management has appeared in machine learning, linear algebra, an optimal tool to analyse and manipulate the data. Data science is a multi-disciplinary subject that uses scientific methods to process the structured and unstructured data to extract the knowledge by applying suitable algorithms and systems. The strength of linear algebra is ignored by the researchers due to the poor understanding. It powers major areas of Data Science including the hot fields of Natural Language Processing and Computer Vision. The data science enthusiasts finding the programming languages for data science are easy to analyze the big data rather than using mathematical tools like linear algebra. Linear algebra is a must-know subject in data science. It will open up possibilities of working and manipulating data. In this paper, some applications of Linear Algebra in Data Science are explained.

**Keywords:** Data, Information, Data Science, Linear Algebra

## INTRODUCTION

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio, and more to produce artificial intelligence systems to perform tasks that ordinarily require human intelligence. In turn, these systems generate insights which analysts and business users can translate into tangible business value (Ambrust *et al.*, 2010). Machine learning is the branch of data science used to design algorithms that automatically extract valuable information from data. The focus here is on “automatic”, i.e., machine learning is general-purpose methodologies that can be applied on datasets, while producing something that is meaningful (Kakhani *et al.*, 2015; Philip *et al.*, 2014).

Linear algebra is the branch of mathematics concerning linear equations, linear functions and their representations through matrices and vector spaces. It helps us to understand geometric terms in higher dimensions, and perform mathematical operations on them. By definition, algebra deals primarily with scalars (one-dimensional entities), but Linear Algebra has vectors and matrices (entities which possess two or more dimensional components) to deal with linear equations and functions (Will, 2014).

Linear Algebra is the heart to almost all areas of mathematics like geometry and functional analysis (Hilbert and Lopez, 2011). Its concepts are a crucial prerequisite for understanding the theory behind Data Science. The data scientist doesn't need to understand Linear Algebra before getting started in Data Science, but at some point, it is necessary to understand how the different algorithms really work. Linear algebra in data science is used as follows.

### Scalars, Vectors, Matrices and Tensors

- A scalar is a single number
- A vector is an array of numbers.
- A matrix is a 2-D array
- A tensor is a n-dimensional array with  $n > 2$



**Fig.1.** Representation of data in data science using linear algebra

## APPLICATIONS OF LINEAR ALGEBRA IN DATA SCIENCES

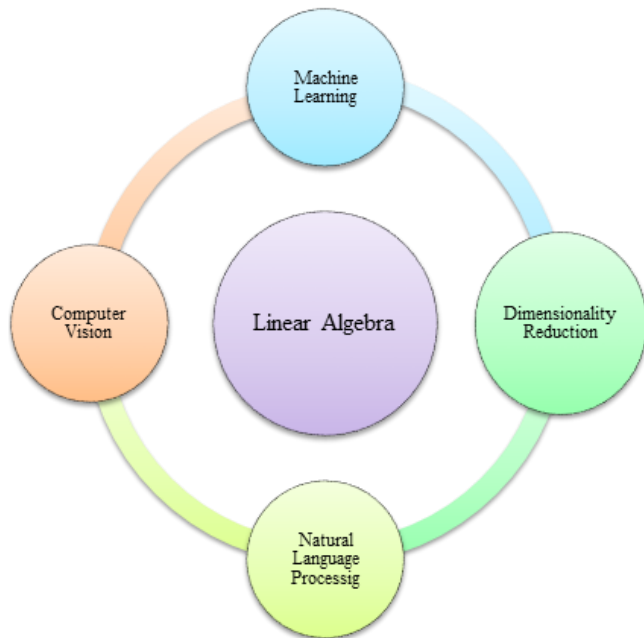


Fig. 2: Applications of linear algebra in data sciences

### Linear Algebra in Machine Learning

The following are the some application areas of linear algebra in machine learning.

1. Loss functions
2. Regularization
3. Covariance Matrix
4. Support Vector Machine Classification

### Loss functions

Consider how good a model is, say a Linear Regression model, and fits a given data:

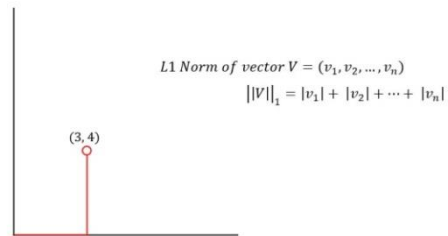
- Some arbitrary prediction function (a linear function for a Linear Regression Model)
- Use it on the independent features of the data to predict the output
- Calculate how far-off the predicted output is from the actual output
- Use these calculated values to optimize prediction function using some strategy like Gradient Descent

It is difficult to calculate how different prediction is from the expected output. This issue can be resolved using loss function. A loss function is an application of the Vector Norm in Linear Algebra. The norm of a vector can simply be its magnitude. There are many types of vector norms. Here, discussed two types.

**L1 Norm:** Also known as the Manhattan Distance or Taxicab Norm. The L1 Norm is the distance travelled from

the origin to the vector if the only permitted directions are parallel to the axes of the space.

### Manhattan Distance or L1 Norm



In this 2D space, consider the vector (3, 4) by traveling 3 units along the x-axis and then 4 units parallel to the y-axis (as shown). Or travelled 4 units along the y-axis first and then 3 units parallel to the x-axis. In either case, travelled a total of 7 units.

**L2 Norm:** Also known as the Euclidean Distance. L2 Norm is the shortest distance of the vector from the origin as shown by the red path in the figure below:

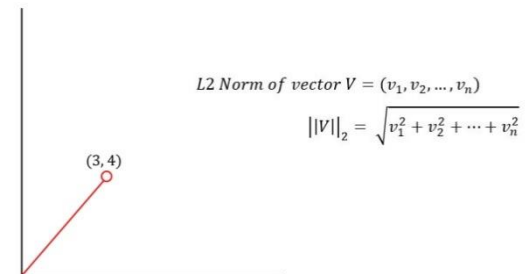


Fig. 3: Euclidean Distance

This distance is calculated using the Pythagoras Theorem. It is the square root of  $(3^2 + 4^2)$ , which is equal to 5. The predicted values are stored in a vector  $P$  and the expected values are stored in a vector  $E$ . Then  $P-E$  is the difference vector. And the norm of  $P-E$  is the total loss for the prediction.

### Regularization

Regularization is a very important concept in data science. It's a technique we use to prevent models from overfitting. Regularization is actually another application of the Norm. A model is said to overfit when it fits the training data too well. Such a model does not perform well with new data because it has learned even the noise in the training data. It will not be able to generalize on data that it has not seen before. The below illustration sums up this idea really well:

Regularization penalizes overly complex models by adding the norm of the weight vector to the cost function. Since we want to minimize the cost function, we will need to minimize this norm. This causes unrequired components of the weight vector to reduce to zero and prevents the prediction function from being overly complex.

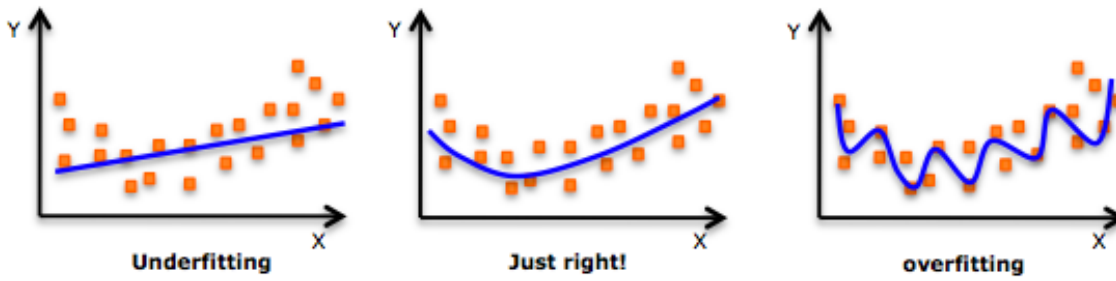


Fig 4: Regularization

The L1 and L2 norms we discussed above are used in two types of regularization:

- L1 regularization used with **Lasso Regression**
- L2 regularization used with **Ridge Regression**

**Covariance Matrix**

Bivariate analysis is an important step in data exploration to study the relationship between pairs of variables. Covariance or Correlation is measures used to study relationships between two continuous variables.

Covariance indicates the direction of the linear relationship between the variables. A positive covariance indicates that an increase or decrease in one variable is accompanied by the same in another. A negative covariance indicates that an increase or decrease in one is accompanied by the opposite in the other.

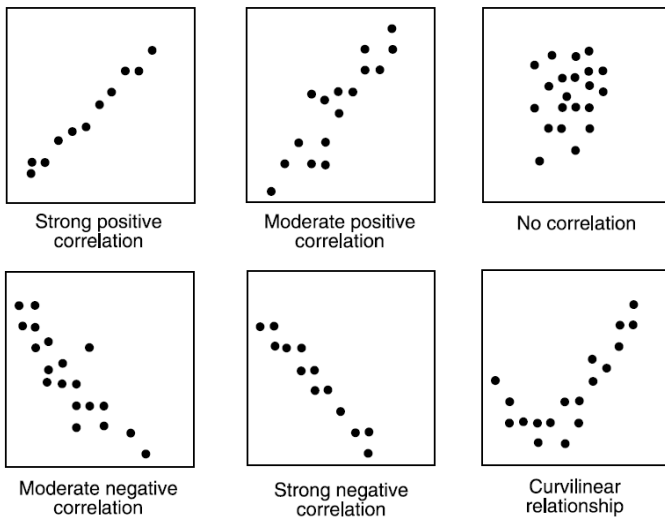


Fig 5: Co-variance

On the other hand, correlation is the standardized value of Covariance. A correlation value tells us both the strength and direction of the linear relationship and has the range from -1 to 1. Using the concepts of transpose and matrix multiplication in Linear Algebra, there is another expression for the covariance matrix:

$$cov = X^T X$$

Here, X is the standardized data matrix containing all numerical features.

**Support Vector Machine Classification**

Support vector machine is the most common classification algorithms that regularly produces remarkable results. It is an application of the concept of Vector Spaces in Linear Algebra. Support Vector Machine, or SVM, is a discriminative classifier that works by finding a decision surface. It is a supervised machine learning algorithm. In this algorithm, we plot each data item as a point in an n-dimensional space (where n is the number of features) with the value of each feature being the value of a particular coordinate. Then, perform classification by finding the hyperplane that differentiates the two classes very well i.e. with the maximum margin, which is C in this case.

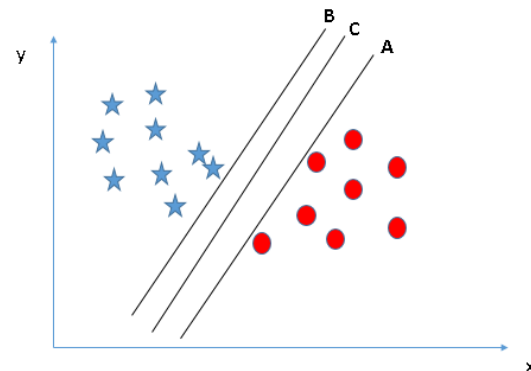


Fig 6: Support Vector Machine

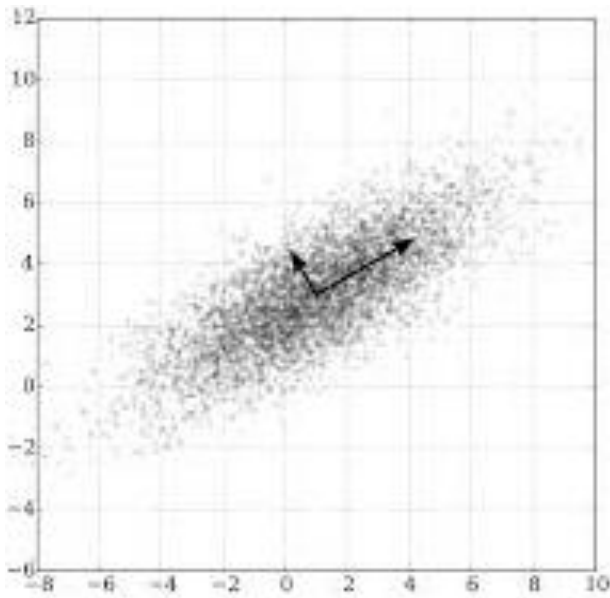
A hyperplane is a subspace whose dimensions are one less than its corresponding vector space, so it would be a straight line for a 2D vector space, a 2D plane for a 3D vector space and so on. Again, Vector Norm is used to calculate the margin.

**Linear Algebra in Dimensionality Reduction**

1. Principal Component Analysis
2. Singular Value Decomposition

**1. Principal Component Analysis**

Principal Component Analysis, or PCA, is an unsupervised dimensionality reduction technique. PCA finds the directions of maximum variance and projects the data along them to reduce the dimensions. Without going into the math, these directions are the eigenvectors of the covariance matrix of the data (Gupta *et al.*, 2010; Slavkovic and Jevtic, 2012)



Eigenvectors for a square matrix are special non-zero vectors whose direction does not change even after applying linear transformation (which means multiplying) with the matrix. They are shown as the red-colored vectors in the figure below:

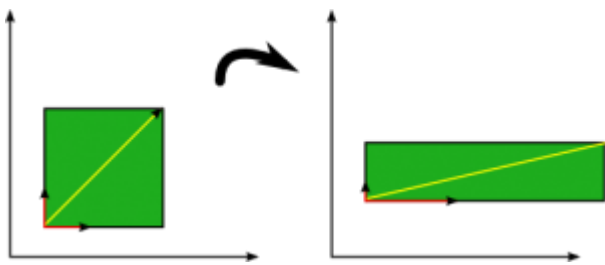


Fig 7: Eigen Vectors

## 2. Singular Value Decomposition

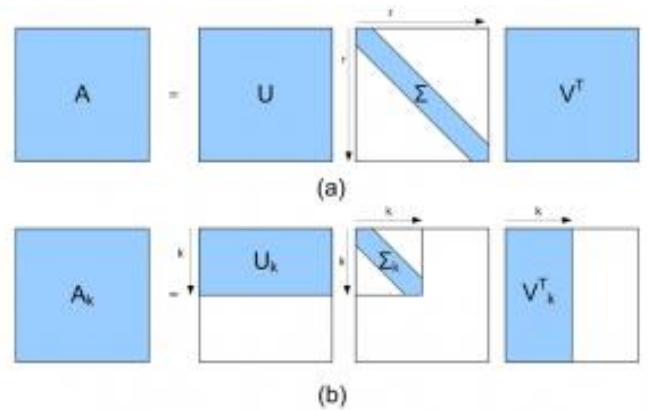
Singular Value Decomposition (SVD) is underrated and not discussed enough. It is an amazing technique of matrix decomposition with diverse applications. Here focused about SVD in Dimensionality Reduction. Specifically, this is known as Truncated SVD.

- Start with the large  $m \times n$  numerical data matrix  $A$ , where  $m$  is the number of rows and  $n$  is the number of features

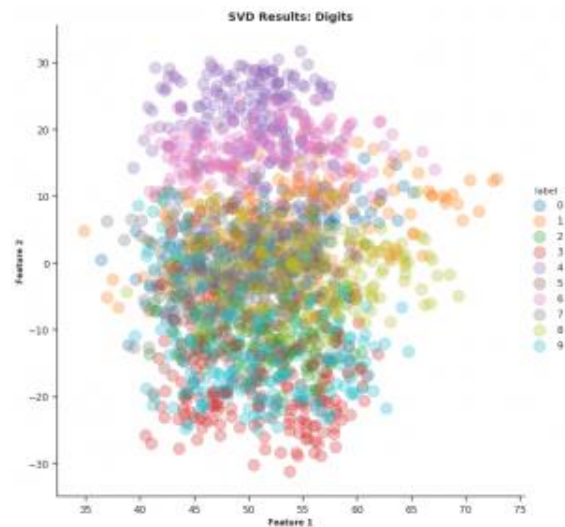
$$A = U D V^T$$

Left singular vectors  $U$       $D$  Singular values      $V^T$  Right singular vectors

Decompose it into 3 matrices as shown here:  
 Choose  $k$  singular values based on the diagonal matrix and truncate (trim) the 3 matrices accordingly:



Finally, multiply the truncated matrices to obtain the transformed matrix  $A_k$ . It has the dimensions  $m \times k$ . So, it has  $k$  features with  $k < n$ . On applying truncated SVD to the Digits data, the below plot was obtained.



## Linear Algebra in Natural Language Processing

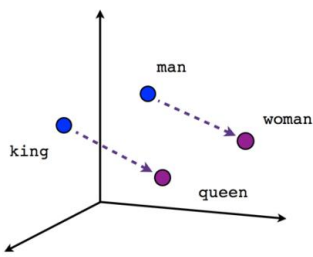
- Word Embeddings
- Latent Semantic Analysis

### Word Embeddings

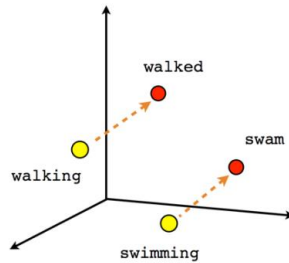
Machine learning algorithms cannot work with raw textual data. The raw data needs to be converted to some numerical and statistical features to create model inputs. There are many ways for extracting features from text data, such as:

- Meta attributes of a text, like word count, special character count, etc.
- NLP attributes of text using Parts-of-Speech tags and Grammar Relations like the number of proper nouns
- Word Vector Notations or Word Embeddings

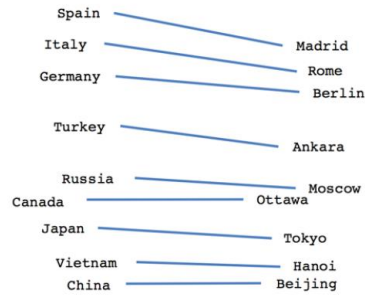
Word Embeddings is a way of representing words as low dimensional vectors of numbers while preserving their context in the document. These representations are obtained by training different neural networks on a large amount of text which is called a corpus. They also help in analyzing syntactic similarity among words:



Male-Female



Verb tense



Country-Capital

Word2Vec and GloVe are two popular models to create Word Embeddings.

**Latent Semantic Analysis**

Latent Semantic Analysis (LSA), or Latent Semantic Indexing, is one of the techniques of Topic Modeling. It is another application of Singular Value Decomposition. Latent means 'hidden'. True to its name, LSA attempts to capture the hidden themes or topics from the documents by leveraging the context around the words.

- First, generate the Document-Term matrix for the data

Use SVD to decompose the matrix into 3 matrices:

- Document-Topic matrix
- Topic Importance Diagonal Matrix
- Topic-term matrix
- Truncate the matrices based on the importance of topics

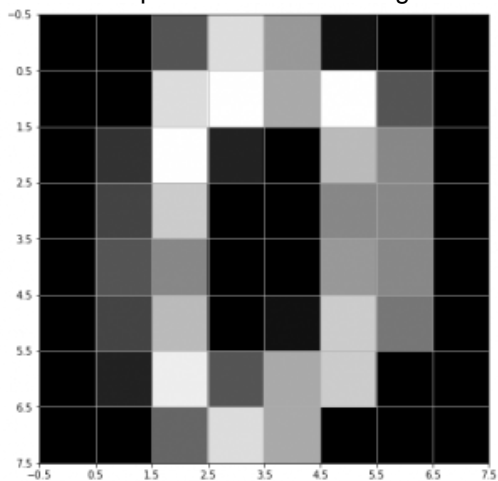
**Linear Algebra in Computer Vision**

Deep learning methods can achieve state – of – the – art results on challenging computer vision problems such as image classification, object detection and face recognition.

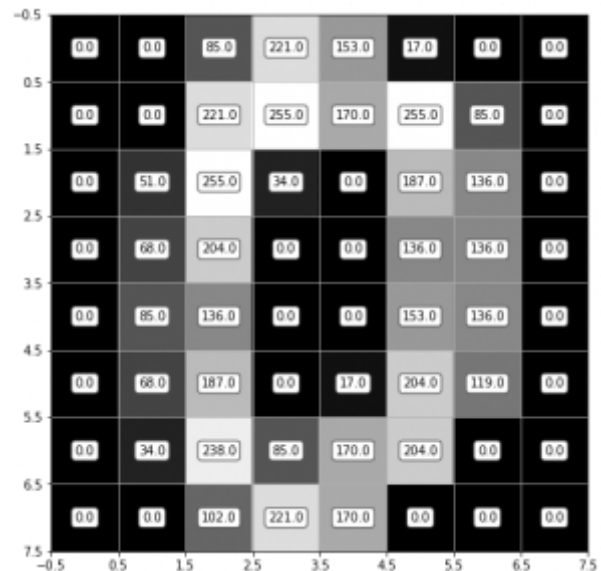
- Image representation as Tensors
- Convolution and Image Processing

**Image representation as Tensors**

A computer does not process images as humans do. Machine learning algorithms need numerical features to work with. A digital image is made up of small indivisible units called pixels. Consider the figure below:



This grayscale image of the digit zero is made of  $8 \times 8 = 64$  pixels. Each pixel has a value in the range 0 to 255. A value of 0 represents a black pixel and 255 represent a white pixel. Conveniently, an  $m \times n$  grayscale image can be represented as a 2D matrix with  $m$  rows and  $n$  columns with the cells containing the respective pixel values:



A colored image is generally stored in the RGB system. Each image can be thought of as being represented by three 2D matrices, one for each R, G and B channel. A pixel value of 0 in the R channel represents zero intensity of the Red color and of 255 represents the full intensity of the Red color. Each pixel value is then a combination of the corresponding values in the three channels: In reality, instead of using 3 matrices to represent an image, a tensor is used. A tensor is a generalized n-dimensional matrix. For an RGB image, a 3rd ordered tensor is used. Imagine it as three 2D matrices stacked one behind another:

**Convolution and Image Processing**

2D Convolution is a very important operation in image processing. It consists of the below steps:

- Start with a small matrix of weights, called a kernel or a filter

- Slide this kernel on the 2D input data, performing element-wise multiplication
- Add the obtained values and put the sum in a single output pixel

INPUT IMAGE				
18	54	51	239	244
55	121	75	78	95
35	24	204	113	109
3	154	104	235	25
15	253	225	159	78

WEIGHT		
1	0	1
0	1	0
1	0	1

429

The function can seem a bit complex but it's widely used for performing various image processing operations like sharpening and blurring the images and edge detection.

## CONCLUSION

Linear algebra has vast uses in real world. Linear algebra methods are applied on the data science to improve the efficiency of the algorithms to attain the more accurate results. In this paper, compiled the applications of linear algebra in data sciences and given an insight of each method. The data scientists can be used linear algebra as tool analyze the data sets. Machine learning approaches are of particular interest considering steadily increasing search outputs and accessibility of the existing evidence is a particular challenge of the research field quality improvement.

## REFERENCES

- Armbrust A, Fox R, Griffith R, Joseph AD, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. (2010). A view of cloud computing. *Commun. ACM.* 53: 50-58.
- Bollen J, Van H, de Sompel, Hagberg A, Chute R, Rodriguez MA, Balakireva L (2009), Click stream Data Yields High-Resolution Maps of Science. *PLoS ONE* 4: 1-11.
- Chapman P, Clinton J, Kerber R, Shearer C, Wirth R. (2000), In: *CRISP-DM 1.0: Step-by-step data mining guide*. The CRISP-DM Consortium.
- Davenport TH, Harris JG (2007). *Competing on Analytics: The New Science of Winning*. 1<sup>st</sup> Ed. Harvard Business School Press. ISBN: 9781422103326
- Dean J, Ghemawat S (2008). Map Reduce: Simplified data processing on large clusters. *Commun. ACM.* 51: 107-113.
- Gupta S, Sahoo OP, Goel A, Gupta R. (2010). A new optimized approach to face recognition using eigen faces. *Global Journal of Computer Science and Technology*, 10: 15–17.

Hilbert M, Lopez P (2011). The world's technological capacity to store, communicate and compute information. *Science.* 332: 60-65.

Jones BF, Wuchty S, Uzzi B. (2008). Multi-University Research Teams: Shifting Impact Geography and Stratification in Science. *Science* 322: 1259-1262.

Kakhani MK, Kakhani S, Biradar SR (2015). Research issues in big data analytics, *International Journal of Application or Innovation in Engineering and Management.* 2: 228-232.

Philip CL, Chen Q, Zhang CY (2014). Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences.* 275: 314-347.

Slavkovic M, Jevtic D (2012). Face Recognition Using Eigenface Approach. *Serbian Journal of Electrical Engineering.* 9: 121-130.

Sturm P, Ramalingam S, Tardif JP, Gasparini S, Barreto J, (2010) Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends in Computer Graphics and Vision.* 6: 1–183.

Szeliski R (2010) *Computer vision algorithms and applications*. London: Springer.

Waldrop MM (1992). *Complexity: The Emerging Science at the Edge of Order and Chaos*. Simon & Schuster. ISBN: 978-0671872342

Will H (2014) *Linear Algebra for Computer Vision*. Attribution-Share Alike 4.0 International (CC BY-SA 4.0). 1-14.

Wuchty S, Jones BF, Uzzi B. (2007). The increasing dominance of teams in production of knowledge. *Science* 316: 1038-1039.

**Accepted 18 November 2019**

**Citation:** Ishchi H. (2019). Linear Algebra – A Powerful Tool for Data Science. *International Journal of Statistics and Mathematics*, 6(3): 137-142.



**Copyright:** © 2019 Ishchi H. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are cited.