Introduction to Tensor Computing in Python

By Manal Helal

Lecturer in Computer Science

School of Engineering, Physics, and Computer Science

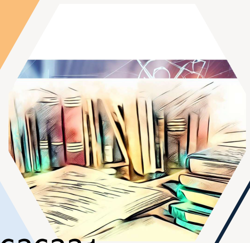
Hertfordshire University

Presented to SAP Innovation Center Network

21/9/2023

Agenda

- Introduction
- Linear Algebra
- Linear Subspace Learning
- Multi-linear Algebra & Tensors
- Multilinear Subspace Learning
- Tensors Networks and Python Libraries
- Representation Learning Approaches
- Tensor Computation Applications – From Direct Analysis to DL
- Challenges and Future Trends



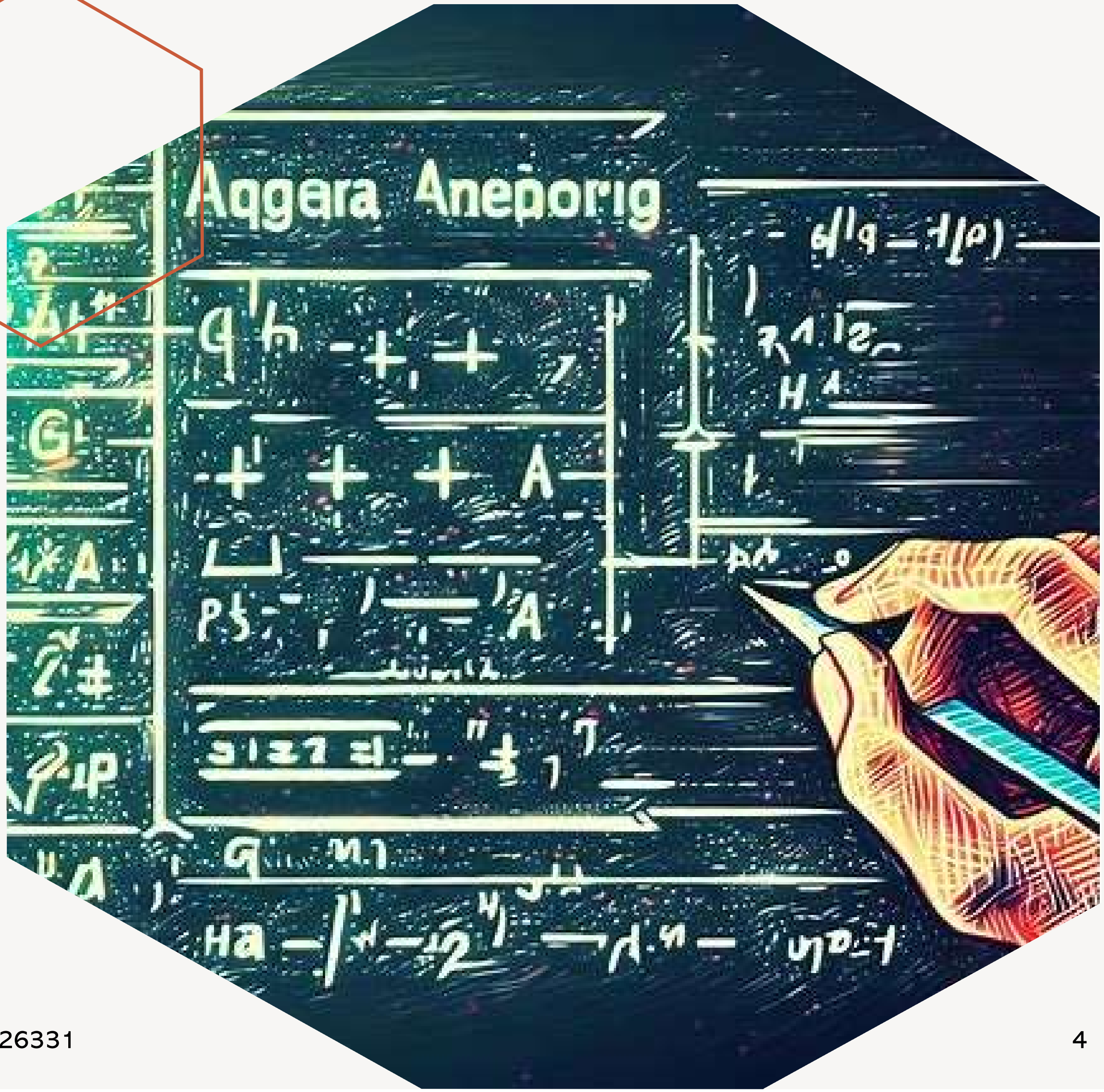
What you can achieve

You can solve one of your current problems using these approaches.

You can also collaborate with me at various scales.

Introduction

Linear Algebra as foundation for most Machine Learning Algorithms



- Vectors & Matrices Operations
- Linear Dependence
- Calculus
- Statistics and Probability

$$y = \epsilon + \sum_{i=0}^{N} w_i x_i$$

Linear Subspace Learning

Projective Methods losing non-linear structure:

- Principal Component Analysis (PCA),
- Singular Value Decomposition (SVD),
- Independent Component Analysis (ICA),
- Linear Discriminant Analysis (LDA),
- Canonical Correlation Analysis (CCA),
- Partial Least Squares (PLS),
- Factor Analysis (FA),
- Non-Negative Matrix Factorisation (NMF),
- and the generalised Nystrőm method

Manifold Modelling Methods:

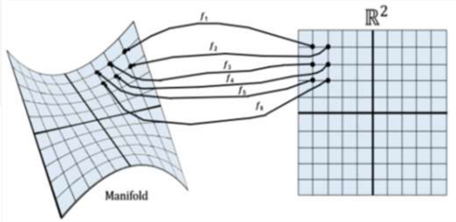
- Mapping the data without learning the manifold:
    - Multidimensional Scaling (MDS)
- Learning the Manifold
    - Isometric Feature Map (Isomap)
    - t-distributed Stochastic Neighbour Embeddings (t-SNE)
    - Locally Linear Embedding
    - Spectral Clustering

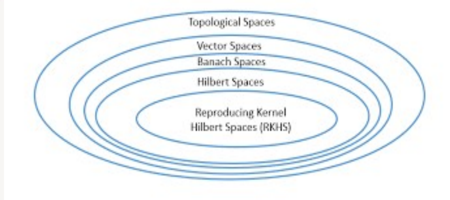
Mapping to higher dimension/Kernel Trick:

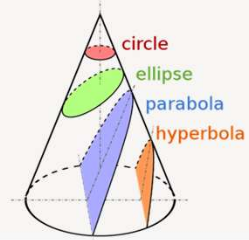
- SVM

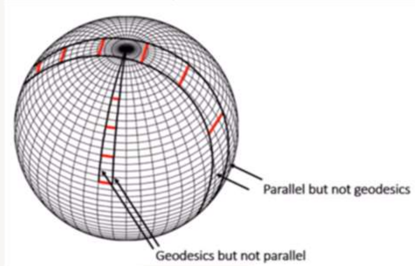
Multi-linear Algebra & Tensors

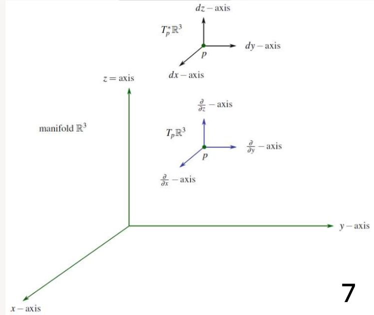
Manifolds	Hilbert Space	Curves	Riemannian Geometry	Differential geometry on Manifolds
Collection of points and not vectors, on a local Euclidean space	Hilbert Space $\mathcal{H}$ is a generalisation of Euclidian space in the infinite dimension.	Non-Euclidean spaces using hyperbolic and elliptic geometry	Riemannian Manifolds describe curvatures in higher dimensions and provide geometric properties to facilitate the partial differential equations used in many Machine Learning (ML) algorithms.	Change of basis using the Jacobian matrix determinant. The tangent vector space with basis $\frac{\delta}{\delta x^i}$ for every dimension i, and the dual (cotangent) space with basis $e^i(v)$ / dx^i as the coordinate function (projection on the coordinates)
Euclidean distance measures or Root Mean Square Error (RMSE)	A Kernel function $K(x_i , x_j) = (\Phi(x_i). \Phi(x_j))$ produces a similarity metric between the data points without explicitly mapping every vector in the dataset. This reduces searching the large space $\mathcal{H}$ to just finding the optimal values of the m coefficients $\alpha_1, \dots , \alpha_m$ of the features $x_1, \dots , x_m$. Example Kernel functions are: Gaussian radial basis function (RBF) equation. 2-layer sigmoidal neural network.	Solve Higher Polynomial Equations, and can be done in Tensor Form, using tensor metric, which is also a dot product for tensors.	Geodiscs, Riemannian metric such as the Fisher metric	Differential Forms: (Tensors) Zero forms (takes a scalar and produce a scalar), 1-form takes a vector and produce a scalar (vector length). 2-forms takes two vectors and produce the area scalar of the parallelogram formed by the two vectos, 3-form takes 3 vectors and produce the volume scalar ....





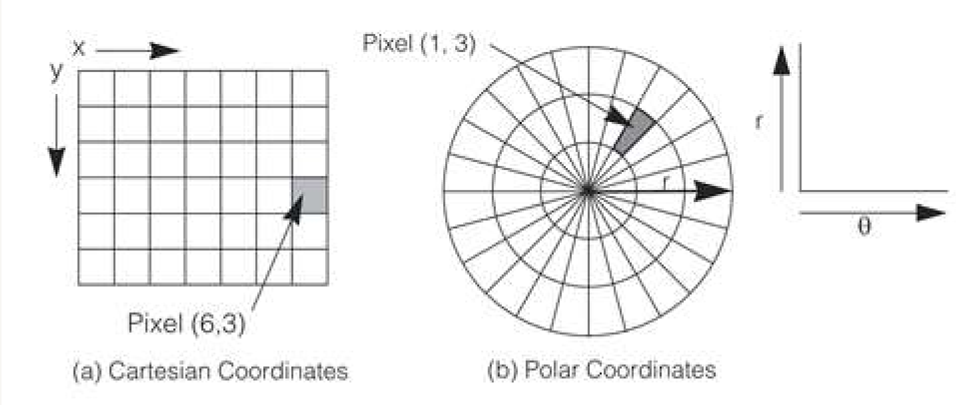






Introduction to Tensor Computing in Python - Manal Helal – ISBN: 9781916626331

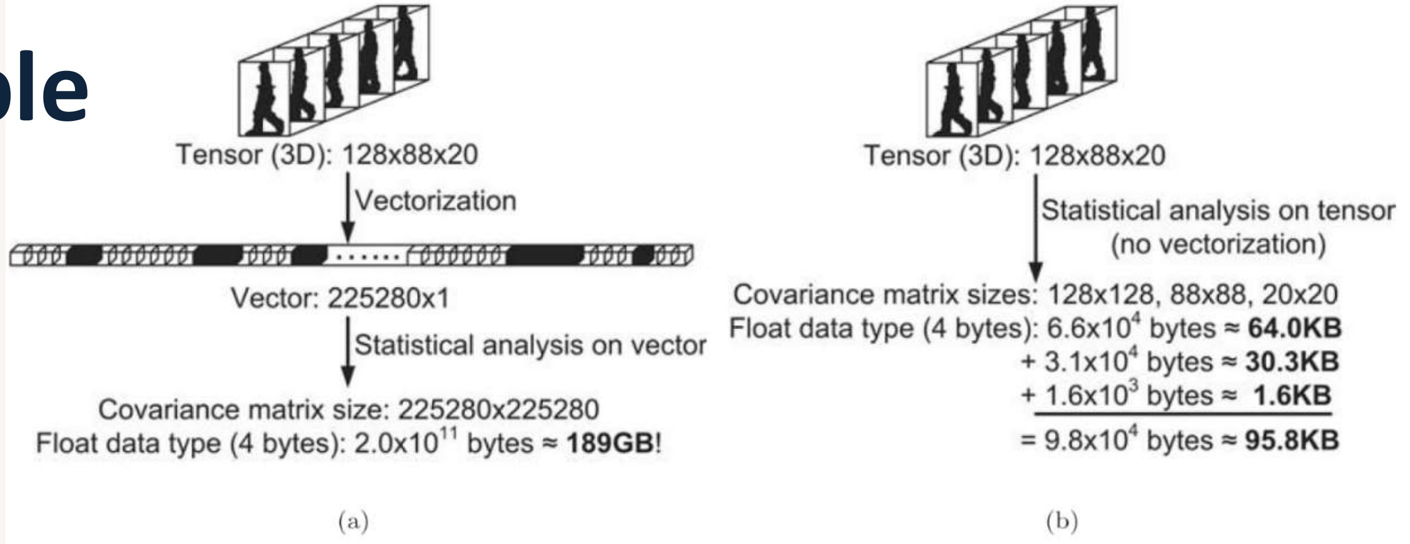
Simple Example



- For example, given a point p = (6, 3), the cartesian coordinate functions $\mathbb{R}^2 \rightarrow \mathbb{R}$ are x(p) = 6, and y(p) = 3. A point exists in any other coordinate system, such as polar, spherical, or cylindrical. The same p point is mapped to the polar coordinate system using mapping functions $r = \sqrt{x^2 + y^2}$, and $\theta = arctan\left(\frac{x}{y}\right)$, and inverse mapping functions are: x= r cos(θ) and y = r sin(θ).
- A transformation of components is achieved for point p in two charts, x and y. Then their coordinate vectors transform with the Jacobian of the coordinate transformation x↦y $\left(\Lambda_{j'}^{i}\right)$: $\frac{\delta}{\delta y^i} = \sum_{j=1}^{n} \frac{\delta x^i}{\delta y^i} \frac{\delta}{\delta x^j}$, and inverse mapping $\Lambda_i^{j'} = \left(\Lambda_{j'}^{i}\right)^{-1}$.
- The cartesian to polar change of basis can be achieved using the Jacobian matrix determinant as:

$$\begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{vmatrix} = \begin{vmatrix} \cos\theta & -\mathrm{r}\sin\theta \\ \sin\theta & r\cos\theta \end{vmatrix} = r\ \cos^2\theta + r\sin^2\theta = r;$$ therefore, $dx\, dy = r\, dr\, d\theta$.

Application Example



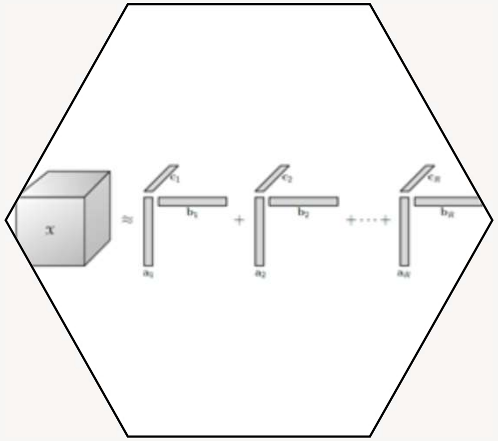
- Given a 3-dimensional video dataset, the first two dimensions being spatial rows and columns of 128 x 88 dimensionality and a time third dimension of 20 frames.
- A Linear Subspace Learning (LSL) vectorisation in (a) performed by the product of the number of dimensions in each mode, results in a large covariance matrix of 189 GB memory fingerprint and the resulting processing time.
- A Multi-linear Subspace Learning (MSL) tensor-based analysis performing the sum of three smaller covariance matrices, results in 95.8KB of memory fingerprint and reduced processing time

Lu, H., Plataniotis, K.N. and Venetsanopoulos, A.N. (2011) 'A survey of multilinear subspace learning for tensor data', *Pattern Recognition*, 44(7), pp. 1540–1551. Available at: https://doi.org/10.1016/j.patcog.2011.01.004.

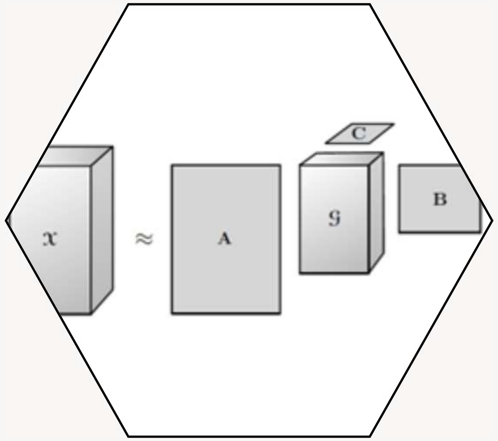
Multilinear Subspace Learning

Multi-way PCA, Multi-way SVD

Tensors Decomposition

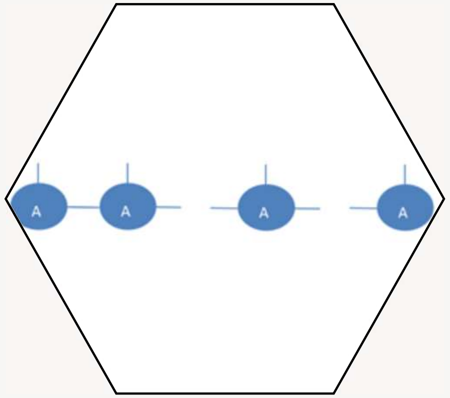


CP Decomposition

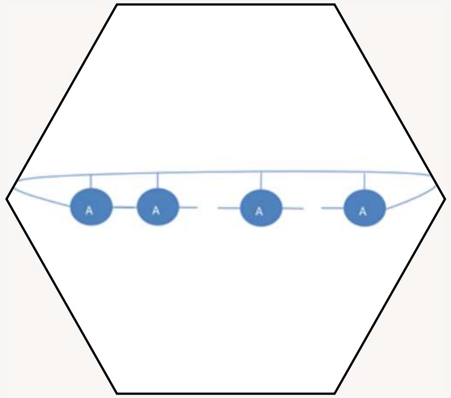


Tucker Decomposition

Tensor Networks

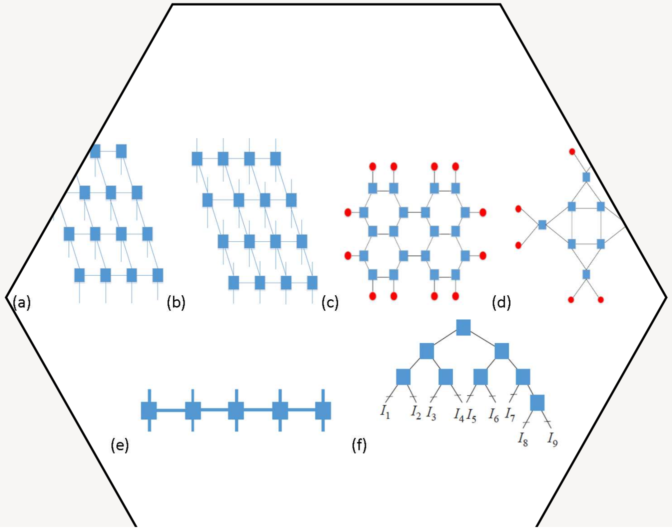


Tensor Trains

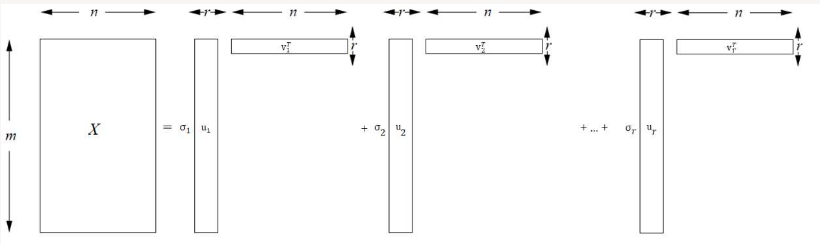


Tensor Rings

Tensors Nesting



2D SVD:



Tensors Networks and Python Libraries

Tensorly
Python

T3F
Python

scikit-tt
Python

TensorD
Python

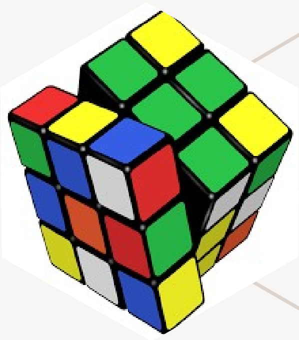
HOTTBOX
Python

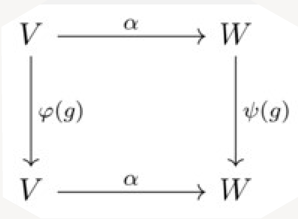
tednet
Python

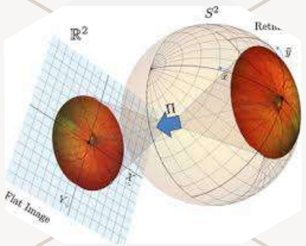
ttrecipes
Python

TensorLab
Matlab

Representation Learning Approaches







Group Theory & Abstract Algebra

Unified Code for all structures using OO concepts like Polymorphism

Representation Theory

Can be manually encoded, and can be learned by initial DL Layers

Change of Coordinates

FFT as the classical change of coordinate from time to frequency, and other examples

Applications

All ML applications need RL as a preprocessing step

Python Libraries

OpenNE learns Network Embedding, and GRLL is a Graph Representation Learning Library

Tensor Computation Applications – From Direct Analysis to DL

Scientific Computing	Knowledge Graphs	Images / Video Object Detection	Text NLP	Multi-modal Applications
Bioinformatics Psychometrics Chemometrics Computational Physics	Social Networks Semantic Web Ontology Building	EigenFaces Tensor Faces Motion Detection CNNs	Symbolic: verb connecting to its subject/object using ANN Sub-symbolic: using embeddings and Deep Learning Models: RNNs, LSTMs, & Transformers	Visual Question Answering (VQA)Multi-Modal Sentiment Analysis (Audio, Visual, & Text)

Other Applications

Graph Neural Networks

The input layer takes graph or network data structure. The NN layers build the computational graph as a multi-partite graph, using various graph and network theory algorithms

Generative Neural Networks

Restricted Boltzmann Machines (RBM) modelled as the Tensor Networks States (TNS)

General Framework

1. Tensorization

If a dataset is not organized in the higher dimension, tensorization can be intuitively applied using reshaping. Avoiding the dimensionality curse might require using an aggregation function, factorization, or other techniques.

2. Representation

Either learned or manually modelled.

Either full dataset or using a compressed/factorised form.

3. Analysis Model

Use Tensor Decomposition Analysis or Tensorize some steps in the NN model.

4. Compression

Reducing the number of parameters

5. Evaluation of Results

Traditional Evaluation of results with the extra benefit of expressiveness.

Challenges and Future Trends

Parallelization

- Various Parallelisation libraries are available in many programming languages, such as Python.
- Various Parallel Architectures such as Multi-core, GPUs, and TPUs are available.
- Various Numeric libraries and ML algorithms high-level packages are already implemented to execute in parallel in the various hardware platforms.

Challenges

- Most datasets are in matrix form, and the tensorization step might need some skills to master.
- Existing high-level libraries for many well-tested ML and DL packages are fixing the dimensions for any given application.
- Hardware is yet to be developed to process data in the higher dimensions, although many are proposed but not yet tested.

Future Work

- Test the state-of-the-art packages and hardware for the tensorized models.
- Continue advancing the applications of the existing tensorized ML packages.

Why These Concepts Matter for SAP Professionals

Enable enhanced problem-solving and innovation within the company by joint higher-degree research project supervision and/or collaboration on funded research projects.

SAP Related Future Projects

Data Cubes Analysis

So far, I think direct methods provide cube materialisation. Employing Tenor operations is hypothesized to enhance performance.

Business Intelligence

Prediction from high dimensional data is known to be enhanced when tensorized methods are employed.

Digital Twins for Business

Digital Twins relay on connecting Various IoT devices creating network of messages

Tensorized DL / Transformer

Analysis of Time Series, or any sequential data such as NLP, High dimensional Data.

Various Products or Research Outcomes

Postgraduate Supervision, R&D Collaboration, Writing Reports, books, or papers

Peiris, P., 2017. BigDataCube: Distributed Multidimensional Data Cube Over Apache Spark (Master of Science). KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY, Stockholm, Sweden.

Spelta, A., 2017. Financial market predictability with tensor decomposition and links forecast. Appl Netw Sci 2, 7. https://doi.org/10.1007/s41109-017-0028-1

Summary

This book summarised a journey into advancing our knowledge from machine learning algorithms based on linear algebra to tensor/multiway machine learning algorithms based on multi-linear algebra. Various fundamental building blocks are explained, such as the mathematical foundations, the algorithmic steps, and wide application domains. Many project ideas are proposed for the various levels, from graduation, MSc, PhD to R&D, using large organisations' resources.

Thank you

Manal Helal

m.helal@herts.ac,uk

www.manalhelal.com