Mathematics for Machine Learning
Linear Algebra

Graph Steps
① Fit curve given parameter
② Goodness of fit ↑
Parameter Space - Cost
③ ↓ Cost wrt parameter

Fitting equation - Data / Solving equation
- Optimization → optimal value of parameter

Linear Algebra
- System of linear equation
  - vector - obj → moves around space
  - vector space - mapping in between vector space

Operations - Vector
- Geometric Object
- List of attributes

Represent
- Polar Coordinate - Geometric Object
- In terms of basis vector - Co-ordinate System

Properties
- Addition - Associative
- Scaler Multiplication - Distributive

Attributes - Size/Length
- Geometric Obj - length
- Co-ordinate System - Pythagoras theorem

Dot Product / Inner Product $r \cdot s = r_i s_i + r_j s_j$ $\qquad r \cdot r = |r|^2$
- Commutative
- Distributive

Cosine Rule ① $c^2 = a^2 + b^2 + 2 a \cdot b \cdot \cos\theta$ $\qquad \cos\theta = \frac{\text{adjacent}}{\text{hypothesis}}$

② $|r - s|^2 = |r|^2 + |s|^2 - 2 r \cdot s$

①+② → $r \cdot s = ab \cos\theta$

$= |r|\ |s| \cos\theta$ → adjacent → shadow / Projection $= \frac{r \cdot s}{|r|}$
- Scaler Projection
- Vector Projection $\frac{r \cdot s}{|r|} \cdot \frac{r}{|r|} = \frac{r \cdot s}{r \cdot r} \cdot r$

Reference Frame
Vector - describe space → Co-ordinate System
basis vector
- angle, length } different
  - independent from co-ordinate system
  - Transform ← dot product
  - Projection → regular & equally spaced grid
  - map → wrap / fold space
- Set of n vectors
  - linearly independent → n dimension space
  - span space

Matrix - object that operates on vector
Simultaneous equation → matrix

Linear Algebra
value, notation - mathematical object → vector
- system of manipulating

Easy - n vectors
- Orthogonal
- Unit length

Matrix transformation Properties
$Ar = r'$
$A(nr) = nr'$
$A(r+s) = Ar + As$

$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} r_i \\ r_j \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \left[r_i \begin{bmatrix} 1 \\ 0 \end{bmatrix} + r_j \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right] = r_i \begin{bmatrix} a \\ c \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + r_j \begin{bmatrix} b \\ d \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

matrix, vector; vector → basis vector; vector transformed to new space defined by matrix

Type
- Identity Matrix $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- Reflection mirror: x-axis $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, $x=y$ $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, y-axis $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$
- Rotation $\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$
- Stretching $\begin{bmatrix} k & 0 \\ 0 & 1 \end{bmatrix}$
- Shearing $\begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$
- Squeezing $\begin{bmatrix} k & 0 \\ 0 & 1/k \end{bmatrix}$

Composition/Combination of Matrix transformation $A_1(A_2 r) = (A_1 A_2) r$ order dependence

Matrix Inverse $A^{-1}A = I$
- Gaussian Elimination → Row elimination, Back substitution → Nonsingular Matrix → Inverse
- Use Determinant - A matrix → stretch original basis vector → to new basis vector space
  area increase by factor → determinant

det = 0
- linearly dependent
- singular matrix
can't be inversed

Matrix makes linear mapping
Matrix Product represent Einstein Convention $(ab)_{ik} = \sum_j a_{ij} b_{jk} = a_{ij} b_{jk}$
- coding
- non square matrix
matrix product

Dot Product - column matrix $u \cdot v = u^T v$
$u \cdot v = v \cdot u$
Geometrically similar Δ

changes in basis world
- Bear
- My

Bear's basis vector in my world * Bear's vector = My vector
$(B) * (B_v) = (M_v)$ → $B^{-1} * M_v = B_v$

Bear's Vector $\xrightarrow{\text{Transform}}$ My Vector
My Vector $\xrightarrow{\text{Transform}}$ Bear Vector

Basis Vectors → Orthogonal → Dot product → Projection

My Vector · Bear's axis in my world = Bear's Vector
- all the axis one by one
- orthogonal vector
- unit vector → normalized by length

Transformation in Bear's world
$B^{-1} * \text{Transformation} * \underbrace{B * \begin{pmatrix} x \\ y \end{pmatrix}}_{\text{vector in my world}}$
- Transformed vector in my world
- Transformed vector in bear's world

Orthogonal Matrix $A = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}$ basis set
- unit length + orthogonal to each other } orthonormal matrix
$a_i \cdot a_j = 0 \quad i \neq j$
$a_i \cdot a_j = 1 \quad i = j$

Transpose $A^T_{ij} = A_{ji}$ → $A^T A = I$
$A^T = A^{-1}$ $\quad AA^T = I$ → rows are also orthonormal matrix

factor area $|A| = \pm 1$, -1 → flip space

Projection of vector → Dot product

Convert to orthonormal basis vector
linearly independent → Determinant
Gram Schmidth process

orthogonal unit basis set
orthonormal basis set

$v = \{v_1, v_2, \dots, v_n\}$
$v_1 \to u_1 = e_1 = v_1 / \|v_1\|$ → unit vector
$v_2 \to u_2 \quad u_2 = v_2 - (v_2 \cdot e_1) e_1 \qquad e_2 = \frac{u_2}{\|u_2\|}$
$v_3 \to u_3 \quad u_3 = v_3 - (v_3 \cdot e_1) e_1 - (v_3 \cdot e_2) e_2 \qquad e_3 = \frac{u_3}{\|u_3\|}$

Transformation in another world

$$\begin{tikzcd} \text{another world} & r \arrow[r, "\text{transform}", "\text{hard}"'] \arrow[d, "E^{-1}"'] & r' \\ \text{orthonormal basis vector word} & r_E \arrow[r, "\text{transform}"] & r'_E \arrow[u, "E"'] \end{tikzcd}$$

$$r' = E T_E E^{-1} r$$

Eigen Vector & Eigen Value
↳ German – characteristic

Geometry
Linear transformation – vector
- every vector in space → shape distort
  - After tranformation, some vector may lie in same line
    - characteristic vector / eigen vector } impact – length → λ / direction – $1, -1$ } eigen value
- 2D
  - Uniform scale – any vector
  - Rotation – $\theta \neq 180°$ – no vector / $\theta = 180$ – any vector $\lambda = 1$
  - Horizontal shear & vertical scalling – 2
  - } eigen vector
- 3D
  - Rotation axis of rotation = eigen vector

Linear Algebra definition – $Ax = \lambda x$

x – vector
A – tranformation
λ = eigen value

$(A - \lambda I)x = 0$

$\Leftrightarrow A - \lambda I = 0 \quad |A - \lambda I| = 0$

Changing eigen basis → efficient matrix manipulation throug diagonilization

$V_n = T(T(T(\ldots(T v_0))))$
$= T^n v_0$
- computationally expensive if T is not diagonal matrix
  - change basis where $T \to$ diagonal matrix
    - eigen basis

matrix of eigen vectors, using eigen value

$T = CDC^{-1}$
$T^2 = CDC^{-1}CDC^{-1}$
$T^n = CD^nC^{-1}$

- Undiagonizable matrix
- Complex Eigenvector

PageRank Algorithm

$$\text{Link Matrix} = \begin{array}{c|c} & A \;\; B \; \ldots \; Z \text{ (from web pages)} \\ \hline \begin{matrix} A \\ B \\ \text{to } \vdots \\ Z \end{matrix} & \begin{bmatrix} (A) & (B) & \cdots & (Z) \end{bmatrix} \end{array}$$

to → row – inward link normalize with page of origin

↓ from
Outward link from given page
Normalized → Probability

Rank of A
- rank of all pages
- link to page A ?
- outgoing link from page A

$$r_A = \sum_{j=1}^{n \to \text{all pages}} \underbrace{L_{Aj}}_{\text{weight from link matrix}} r_j$$

$r = Lr$
- Initial guess for r
- iterate untill convergance
} Power Method
- only interested eigenvalue = 1
- sparse Matrix

Dampening factor
$$r^{i+1} = d(Lr^{(i)}) + \frac{1-d}{n}$$

$d \in \{0, 1\}$
$d = 1 - p \to$ random
Stability vs speed
d – probability to follow link