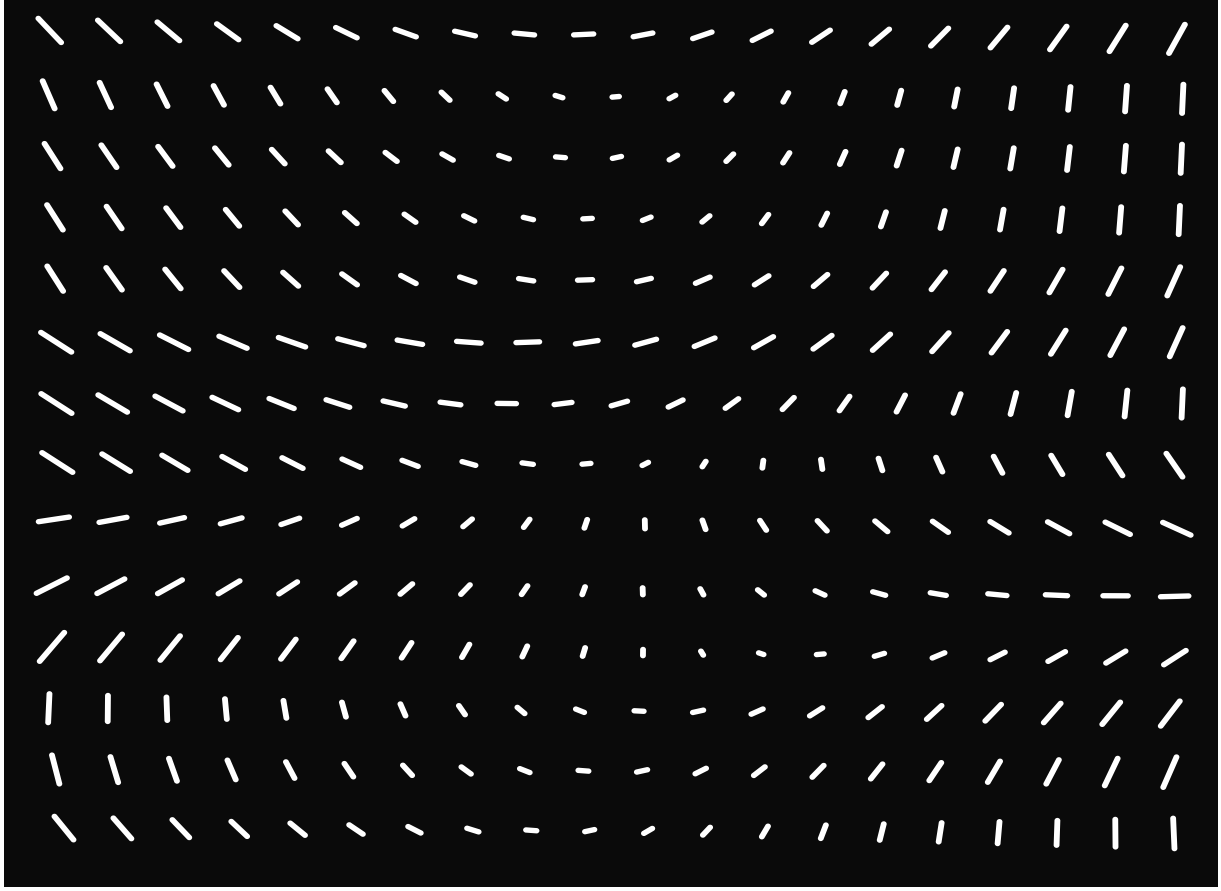
Linear Algebra



Vector Spaces

Span is a set of vectors that can be expressed as a linear combination.

$$\text{Span}(\{x_1, \dots x_n\}) \triangleq \left\{ v : v = \sum_{i=1}^{n} \alpha_i x_i,\ \alpha_i \in R \right\}$$

If vectors $\{x_1, \dots x_n\}$ are set of linearly independent vectors, where each $x_i \in R^n$, then span of the vectors will also be R^n i.e., $\text{span}(\{x_1, \dots x_n\}) = R^n$.

Basis

A basis B is a set of linearly independent vectors that spans the whole space

$$\text{Span}(B) = R^n.$$

Linear Maps

$f : V \to W$ such that $f(v+w) = f(v) + f(w)$

and $f(av) = a f(v)$ $\forall\ v, w \in R$

Here 'a' is some constant. For ex: $f(2x)$

We can compute $y = f(x) \in R^m$ for any $x \in R^n$ as:

$$y = \left(\sum_{j=1}^{n} a_{1j} x_j, \dots, \sum_{j=1}^{n} a_{mj} x_j \right)$$

which is basically computing

$$y = Ax$$

If the function is invertible, then we can write

$$x = A^{-1} y$$

Range and Null space

In equation, $y = Ax$, the range of A is the span of columns in A. Formally, this is written as

$$\text{range}(A) \triangleq \{v \in \mathbb{R}^m : v \in Ax, x \in \mathbb{R}^n\}$$

Think of range as the set of vectors that can be reached or generated by A when multiplied with 'x'

Null space is the set of vectors that get mapped to the null vector when multiplied by A

$$\text{nullspace}(A) \triangleq \{x \in \mathbb{R}^n : Ax = 0\}$$

Linear Projection

Given $y \in \mathbb{R}^m$ and a span of vectors $\{x_1, \dots x_n\} \in \mathbb{R}^m$, the projection is finding the closest vector in the span w.r.t to 'y' in euclidean distance norm $\|v - y\|_2$ where $v \in \text{span}(\{x_1, \dots x_n\})$

$$\text{Proj}(y; \{x_1, \dots x_n\}) = \text{argmin}_{v \in \text{span}(\{x_1, \dots x_n\})} \|y - x\|_2$$

To compute projection with a matrix,

$$\text{Proj}(y; A) = \text{argmin}_{v \in R(A)} \|v - y\|_2 = A(A^TA)^{-1}A^Ty$$

This is an important formula for orthogonal projections in Least squares linear regression. (11.2.2.2 in book)

7.1.3 Norms of a vector and matrix

"ways of measuring the size of a vector and a matrix"

Vector Norms

A norm is a measure of the length of the vector.

Denoted as $\|x\|$

Commonly used norms are:

p-norm $\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$ for $p \geq 1$

if $p = 1$,

1-norm $\|x\|_1 = \sum_{i=1}^{n} |x_i|$

if $p = 2$,

2-norm $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$ a.k.a Euclidean norm

Note: $\|x\|_2^2 = x^T x$

Max-norm $\|x\|_\infty = \max_i |x_i|$

0-norm $\|x\|_0 = \sum_{i=1}^{n} \mathbb{I}(|x_i| > 0)$ a.k.a pseudo norm

It counts the number of non-zero elements in 'x'.

Text book suggests, if we define $0^0 = 0$, we can write the pseudo norm as $\|x\|_0 = \sum_{i=1}^{n} x_i^0$. But how?

Note: we are only defining $0^0 = 0$ not for the rest of the numbers. So, if $x = [1, 0, 0, 1, 0]$

$\|x\|_0 = x_0^0 + x_1^0 + x_2^0 + x_3^0 + x_4^0$

$= 1^0 + 0^0 + 0^0 + 1^0 + 0^0$

$= 2$ ✓

Matrix Norms

Imagine a linear function such as $f(x) = Ax$ where $A \in \mathbb{R}^{m \times n}$

Here the function 'f' lengthens the matrix 'A' to any unit norm.

In matrix norm, we seek to find the maximum amount by which f could lengthen the matrix. We refer this as induced norm of A which is given as

$$\|A\|_p = \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|=1} \|Ax\|_p$$

For $p=2$, the matrix norm would be

$$\|A\|_2 = \sqrt{\lambda_{max}(A^TA)} = \max_i \sigma_i$$

σ_i is the i^{th} singular value.

What is λ_{max}?
It is the maximum eigenvalue

Singular values are non-negative values of a matrix obtained after decomposition. Any matrix could be decomposed into the form $U\Sigma V^*$ where U and V are unitary values. Σ is a diagnol matrix consisting of Singular values. These values in the Σ diagonal matrix are listed in decreasing order. The maximum i.e., $s_0 \in \Sigma$ is the norm of the matrix.

Nuclear Norm

aka Trace Norm defined as

$$\|A\|_* = tr(\sqrt{A^TA}) = \sum_i \sigma_i$$

where $\sqrt{A^TA}$ is the matrix square root

Since the singular values are always non-negative, we have

$$\|A\|_* = \sum_i |\sigma_i| = \|\sigma\|_1$$

There are different ways of calculating norms. One among them is Schatten-p norm

Schatten p-norm

$$\|A\|_p = \left(\sum_i \sigma_i^p(A)\right)^{1/p}$$

Frobenius Norm

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2} = \sqrt{tr(A^TA)} = \|vec(A)\|_2$$

In the case of Frobenius norm, we can think the matrix as a vector which is given as $\|A\| = \|vec(A)\|$. Above formula is the case of vector in 2-norm.

Another way of solving Forbenius norm is through singular values i.e.,

$$\|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \cdots + \sigma_n^2}$$

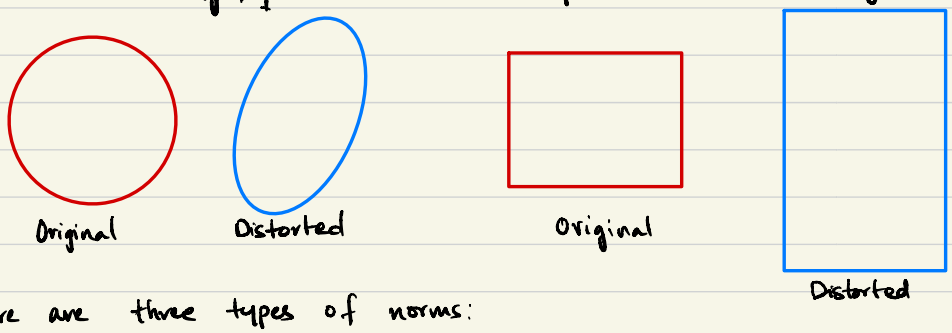
If A is a huge matrix, then computing norm could be expensive. A stochastic approximation to the Frobenius norm could be created by Hutchinson Trace Estimation technique i.e.,

$$\|A\|_F^2 = tr(A^TA) = \mathbb{E}[v^TA^TAv] = \mathbb{E}[\|Av\|_2^2]$$

where $v \sim \mathcal{N}(0, I)$

What is the difference between these norms?

To recap, norms are used to identify how much a function 'f' can lengthen a matrix. When we are provided with a matrix for a shape we could identify by how much the shape could distort using norm.



There are three types of norms:

- Induced norm: which measures what is the maximum of $\frac{\|Ax\|}{\|x\|}$ for any $x \neq 0$ (or, equivalently, the maximum of $\|Ax\|$ for $\|x\|=1$).
- Element-wise norm: which is like unwrapping the matrix A into a a long vector, then calculating its vector norm
- Schatten-norm, which measures the vector norm of singular values of A.

How each norm is different?

Frobenius norm = Element-wise 2-norm = Schatten 2-norm

Induced 2-norm = Schatten ∞-norm. aka Spectral norm.

7.1.4 Properties of a matrix

Trace of a square matrix

Let $A \in \mathbb{R}^{m \times m}$ be a square matrix. The trace of a matrix is the sum of the diagonal elements in the matrix. It is denoted as: $tr(A)$. and defined as:

$$tr(A) \triangleq \sum_{i=1}^{n} A_{ii}.$$

Trace has the following properties:

$$tr(A) = tr(A^T)$$
$$tr(A+B) = tr(A) + tr(B)$$
$$tr(C) = c\,tr(A)$$
$$tr(AB) = tr(BA)$$
$$tr(A) = \sum_{i=1}^{n} \lambda_i \text{ where } \lambda_i \text{ are the eigenvalues of } A.$$
$$tr(ABC) = tr(BCA) = tr(CAB)$$ — Cyclic Permutation Property *

Using cyclic permutation property, we can have the following trace trick with scalars (x, x^T) as follows:

$$x^TAx = tr(x^TAx) = tr(xx^TA)$$

Hutchinson Trace Estimator

For large matrices, it may be expensive to compute trace. But we can compute effecently through matrix-vector products Av, where 'v' is a vector sampled randomly from uniform distribution. Here we could use Monte carlo approximation to $tr(A)$ using the following identity:

$$tr(A) = tr(A\mathbb{E}(vv^T)) = \mathbb{E}[tr(Avv^T)] = \mathbb{E}[tr(v^TAv)]$$

Definite Matrices

A symmetric matrix $M \in \mathbb{R}^{n \times n}$ could be classified into one of the 4 types of definite matrices

Positive-definite matrix: M is positive-definite if $x^TMx > 0 \; \forall \; x \in \mathbb{R}^n \setminus \{0\}$ ($x \neq 0$ but all other numbers)

Positive-semidefinite matrix: M is positive-semidefinite if $x^TMx \geq 0 \; \forall \; x \in \mathbb{R}^n$

Negative-definite matrix: M is negative-definite if $x^TMx < 0 \; \forall \; x \in \mathbb{R}^n \setminus \{0\}$

Negative-semidefinite matrix: M is negative-semidefinite if $x^TMx < 0 \forall x \in \mathbb{R}^n$

* You can also determine definite-ness through eigenvalues.

Positive-definite if all eigenvalues are positive, Positive-semidefinite if all eigenvalues are non-negative, Negative definite if all eigenvalues are negative, Negative-semidefinite if all eigen-values are non-negative, Indefinite if the matrix has mix of positive and negative eigen values.

Determinant of a square matrix

When mathematicians were studying matrices, they were looking for a simple heuristic which capture the characteristics of a matrix. They were looking for a determining function for a matrix which now we refer as determinant.

A determinant of a 2x2 matrix given below is computed as

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad |A| = ad - bc$$

A determinant of a matrix $|A| \neq 0$ suggest that for a given system of linear equations, there exists a unique solution.

If a determinant $|A| = 0$, then the matrix doesn't have an inverse.

A determinant can enable us to measure change in unit volume of a shape through linear transformation. The details of this heuristic are obscure at this point of time. But it just signifies the important of determinant.

Properties

$|A| = |A^T|$

$|cA| = c^n |A|$

$|AB| = |A||B|$

$|A| = 0$ iff A is singular

$|A^{-1}| = 1/|A|$ if A is not singular

$$|A| = \prod_{i=1}^{n} \lambda_i \qquad \lambda_i \rightarrow \text{eigenvalues of } A$$

What does it mean a matrix is singular?
- A square matrix which is not invertible is called Singular Matrix
- Simple check if $|A| = 0$ then A is singular

For a positive definite matrix, we can transform

$A = L L^T \rightarrow$ Cholesky Decomposition.

Lower triangular Cholesky decomp

Upper triangular Cholesky decomp

Because $|A| = |A^T|$ proved

$$|A| = |L||L^T| = |L|^2$$

$$\log|A| = 2\log|A| = 2\log \prod_i L_{ii}$$

$$= 2\,\text{trace}(\log(\text{diag}(L)))$$

Rank of a matrix

- Ranks tells us whether we may have a chance at solving the system of linear equations.
  - Column Rank is the dimension of the space spanned by its columns.
  - Row Rank is the dimension of the space spanned by its rows

What is the meaning of spanned by its row or columns?

- Consider this example matrix

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 6 & 9 \end{bmatrix}$$

Here the row rank is 1
column rank is 1

How?

Row rank: The 2nd row is duplicate of 1st
So only 1 independent row

Column rank: The 3rd column is 3 times the 1st column. The 2nd column is 2 times the 1st column.
So only 1 independent column.

Recall "Span" definition:
Span is the set of linearly independent vectors

Hence, the rank of column is the length of the column span and similarly the rank of rows is the length of row span.

- For any given matrix, the rank of column = rank of row. (As seen in above example)
- Hence, we generally denote simply rank(A) for a matrix.

Properties: (Let $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$)

- $\text{rank}(A) \leq \min(m, n)$. If $\text{rank}(A) = \min(m, n)$, then A is referred as "full rank" or "Rank deficient" (why deficient?)
- $\text{rank}(A) = \text{rank}(A^T) = \text{rank}(A^T A) = \text{rank}(AA^T)$
- $\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))$
- $\text{rank}(A+B) \leq \text{rank}(A) + \text{rank}(B)$

One can show that square matrix is invertible iff it is full rank
A matrix is said to be invertible if $A \cdot A^{-1} = I$ } Invertible matrix suggest that there exist a unique solution.

Condition Numbers

Condition numbers measures how numerically stable any computations involving A (our matrix) will be.

What does it mean numerically stable?

- Let's considered a matrix 'A'. If we multiply it with a vector 'x', then we have a new matrix B

$$Ax = B$$

- Let's change the x by a small amount. Normally, what we expect is the B also changes by a small amount
- However, instead if the B changes by a large value then A is deemed to be numerically unstable
- A numerically stable matrix have outputs proportional to the change in perturbations.

Conditional number is denoted as $\kappa(A)$

$$\kappa(A) \triangleq \|A\| \cdot \|A^{-1}\|$$

where $\|A\|$ is the norm of matrix.

If we recall, there are different types of norms. Condition number changes based on the type of norm

If $\kappa(A) \leq 1$, then it is well-conditioned

If $\kappa(A)$ is large, then it is ill-conditioned

A larger $\kappa(A)$ indicates it is nearly singular, whereas matrix with $\kappa(A)$ close to 1 is far from singular.

In real world, singular matrices are rare — meaning have low probability. But singular matrices acts as a boundary for positive & negative determinant matrices

For example:

Let $A = 0.1\, I_{100 \times 100}$

the determinant

$|A| = 10^{-100}$ (which means close to being singular)

but $\kappa(A) = 1$ (because scaling by 10 would lead to non-singular matrix)

If norm is euclidean-norm (aka L_2-norm), then $\kappa(A)$ could also be computed as the ratio of large & small singular values

$$\kappa(A) = \frac{\sigma_{max}}{\sigma_{min}} = \sqrt{\frac{\lambda_{max}}{\lambda_{min}}}$$

λ is eigenvalues

A well-conditioned matrix A of an elliptical shape preserves the shape. Whereas an ill-conditioned matrix skews the unit sphere into a long thin cigar or needle shape.

Note: the quadratic objective function for the above ellipsoid shape is

$$f(x) = x^T A x$$

If $K(A) = 1$, the shape will stay circular.

If $K(A) \approx 10^3$, the shape could be a line or skewed ellipse

1.1.5 Special Type of Matrices

Diagonal Matrix

A matrix where all non-diagonal elements are 0.

Denoted as $D = (d_1, \dots d_n)$

$$D = \begin{bmatrix} d_1 & & \\ & d_2 & \\ & & \ddots \\ & & d_n \end{bmatrix} \quad \text{Ex: } \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \text{Identity matrix}$$

Identity matrix:

$I \in R^{n \times n}$ — Square matrix

$I = diag(1, 1, 1 \dots)$

$AI = IA = A$

Block Diagonal

Contains matrices on its main diagonal

Eg: $\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$

Band – Diagonal matrix

Contains entries only along the diagonal and k sides (width) of diagonal. For a tridiagonal 6×6 matrix (k=3)

$$\begin{bmatrix} A_{11} & A_{12} & 0 & 0 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & A_{34} & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & A_{45} & 0 \\ 0 & 0 & 0 & A_{45} & A_{55} & A_{56} \\ 0 & 0 & 0 & 0 & A_{65} & A_{66} \end{bmatrix} \quad \text{k=3}$$

Triangular Matrices

An upper triangular matrix has non-zero entries above diagonal & vice-verse for lower triangular matrix.

In triangular matrices, the diagonal elements are the eigen values of A. Therefore, the $|A| = \Pi_i A_{ii}$

Positive Definite Matrices

Given $A \in R^{n \times n}$ and $x \in R^n$ (vector), the scalar obtained through the product $x^T A x$ is called Quadratic Form

$$x^T A x = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i x_j$$

Property:

$$x^T A x = (x^T A x)^T = x^T A^T x = x^T \left(\frac{1}{2} A + \frac{1}{2} A^T\right) x$$

Assumption

Matrices appearing often in quadratic form are symmetric

Positive Definite: $x^T A x > 0 \quad \forall x \in R^n$

Negative Definite: $x^T A x < 0 \quad \forall x \in R^n$

Indefinite: $x^T A x$ neither +ve nor -ve for varying x values i.e., $x_1^T A x_1 < 0$ and $x_2^T A x_2 > 0$

Positive Semi definite: $x^T A x \geq 0 \quad \forall x \in R^n$

Negative Semi definite: $x^T A x \leq 0 \quad \forall x \in R^n$

- If A is positive definite, then $-A$ is negative definite } Vice versa
- Similarly A is semi positive definite, then $-A$ is negative semi definite }
- If all eigen values are +ve, then A is +ve definite.
- If all elements are +ve, it does not mean it is +ve definite. A +ve definite matrix can have -ve entries.
- Positive definite matrix diagonal values are larger than sum of the elements of row of diagonal element. +ve Definite are diagonally dominant

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad \forall i$$

- In 2d, any symmetric matrix is +ve definite iff $a>0$, $d>0$, and $ad>b^2$ for the matrix $\begin{pmatrix} a & b \\ b & d \end{pmatrix}$

Gram Matrix

Any matrix $A \in \mathbb{R}^{m \times n}$, the Gram matrix

$G = A^T A$ is always positive semidefinite

if $m \geq n$ and A is full rank ($\text{rank}(A) = \min(m,n)$ → completely independent vectors in $m \geq n$)

then $G = A^T A$ is +ve definite

Orthogonal Matrices

Two vectors $x, y \in \mathbb{R}^n$ are orthogonal if

$$x^T y = 0$$

A vector is normalized if

$$\|x\|_2 = 1 \quad \text{(normalized } x\text{)}$$

A set of vectors that is pairwise orthogonal and normalized is called orthonormal

A square matrix U is orthogonal if all its columns are orthonormal

Unitary: If the entries of U are complex then we call U as Unitary

U is orthogonal iff

$$U^T U = U U^T = I$$

If U is not square and $U \in \mathbb{R}^{m \times n}$, where $n < m$ and columns are orthonormal then $U^T U = I$, but $U U^T \neq I$

We call orthogonal only for square matrix

Ex: $R = \begin{pmatrix} \cos(\alpha) & -\sin\alpha & 0 \\ \sin(\alpha) & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$

$R(-\alpha) = R(\alpha)^{-1} = R(\alpha)^T$

Try: $\alpha = 45°$

Euclidean Norm:

$$\|Ux\|_2 = \|x\|_2 \qquad x \in \mathbb{R}^n \setminus \{0\}$$

Angle between two vectors are always preserved after transformed by an orthogonal matrix.

$$\cos(\alpha(x,y)) = \frac{x^T y}{\|x\| \|y\|}$$

so, $\cos(\alpha(Ux, Uy)) = \frac{(Ux)^T (Uy)}{\|Ux\| \|Uy\|} = \frac{x^T y}{\|x\| \|y\|} = \cos(\alpha(x,y))$

Transformations by orthogonal matrix are generalizations

Rotations if $|U| = 1$

Reflections if $|U| = -1$

Tip: Any square matrix can be transformed into orthogonal matrix using a technique called Gram Schmidt Orthogonalization.

7.2 Matrix Multiplication

Let $A \in R^{m \times n}$, $B \in R^{n \times p}$ and the product of the two matrices is given as: $C = AB \in R^{m \times p}$

where, $C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}$

- Time complexity: $O(mnp)$
- Matrix multiplication can be parallelized.
  - GPUs and TPUs perform faster.

Properties:

Associative $(AB)C = A(BC)$

Distributive $A(B+C) = AB + AC$

Not commutative $AB \neq BA$

7.2.1 Vector-vector products

Two vectors $x, y \in R^n$, the quantity is called inner product, dot product or scalar product

$$\langle x, y \rangle \triangleq x^T y = \sum_{i=1}^{n} x_i y_i$$

Dot products are commutative $x^T y = y^T x$

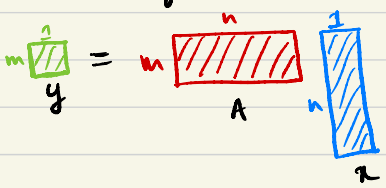
Given vectors $x \in R^m$, $y \in R^n$, xy^T is called outer product

It is a matrix whose entries are given by $(xy^T)_{ij} = x_i y_j$

$$xy^T \in R^{m \times n} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_n \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_m y_1 & x_m y_2 & \cdots & x_m y_n \end{bmatrix}$$

Matrix – vector products

Given a matrix $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$, the product between the two will be: $y = Ax \in \mathbb{R}^m$



y can be viewed as the linear combination of the columns of A where the coefficients of this combination are 'x'

$$y = Ax = \begin{bmatrix} | \\ a_1 \\ | \end{bmatrix} x_1 + \begin{bmatrix} | \\ a_2 \\ | \end{bmatrix} x_2 + \cdots + \begin{bmatrix} | \\ a_n \\ | \end{bmatrix} x_n$$

These columns can be viewed as a set of basis vectors defining a linear subspace.

Basis vectors are a set of linearly independent vectors spanning whole space.

Matrix – Matrix Products

$C = AB$ where $A \in \mathbb{R}^{m \times n}$ $B \in \mathbb{R}^{n \times p}$

vector-vector inner product:

Let $A = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$ $B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

$$C = AB = \begin{bmatrix} — & a_1^T & — \\ — & a_2^T & — \end{bmatrix} \begin{bmatrix} | & | \\ b_1 & b_2 \\ | & | \end{bmatrix} = \begin{bmatrix} a_1^T b_1 & a_1^T b_2 \\ a_2^T b_1 & a_2^T b_2 \end{bmatrix}$$

$$a_1^T b_1 = \begin{bmatrix} 1 & 3 \end{bmatrix} \begin{bmatrix} 5 \\ 7 \end{bmatrix} = 5 + 21 = 26$$

$$a_2^T b_1 = \begin{bmatrix} 2 & 4 \end{bmatrix} \begin{bmatrix} 5 \\ 7 \end{bmatrix} = 10 + 28 = 38$$

$$a_1^T b_2 = \begin{bmatrix} 1 & 3 \end{bmatrix} \begin{bmatrix} 6 \\ 8 \end{bmatrix} = 6 + 24 = 30$$

$$a_2^T b_2 = \begin{bmatrix} 2 & 4 \end{bmatrix} \begin{bmatrix} 6 \\ 8 \end{bmatrix} = 12 + 32 = 44$$

$$\begin{bmatrix} 26 & 30 \\ 38 & 44 \end{bmatrix}$$

Vector-vector outer product:

$$C = AB = \begin{bmatrix} | & | & \\ a_1 & a_2 & \cdots \\ | & | & \end{bmatrix} \begin{bmatrix} - b_1^T - \\ - b_2^T - \\ \vdots \end{bmatrix} = \sum_{i=1}^{n} a_i b_i^T$$

Matrix-vector product:

$$C = AB = A \begin{bmatrix} | & | & \\ b_1 & b_2 & \cdots \\ | & | & \end{bmatrix} = \begin{bmatrix} | & | & \\ Ab_1 & Ab_2 & \cdots \\ | & | & \end{bmatrix}$$

$$C = Ab_i$$

Similarly,

$$C = AB = \begin{bmatrix} - a_1^T - \\ - a_2^T - \\ \vdots \end{bmatrix} B = \begin{bmatrix} - a_1^T B - \\ - a_2^T B - \\ \vdots \end{bmatrix}$$

$$c_i^T = a_i^T B$$

$A^2 = A \cdot A$

$A^{\odot 2} = [A_{ij}^2] \rightarrow$ elementwise square

If A is a diagonal matrix, then

$$A^2 = A^{\odot 2}$$

Application: Manipulating data matrices

Given $X \in N \times D$ matrix

Sum across rows:

$$1_N^T X = (\Sigma_n x_{n1} \cdots \Sigma_n x_{nD})$$

Sum across columns:

$$X 1_D = \begin{pmatrix} \Sigma_d x_{1d} \\ \vdots \\ \Sigma_d x_{Nd} \end{pmatrix}$$

Sum all entries:

$$1_N^T X 1_D = \Sigma_{ij} x_{ij}$$

Scaling Rows and Columns

Scaling Rows using diag(s) matrix where 's' is a scaling factor given by diag(s)X

Scaling Columns using diag(s) as X diag(s)

Remember, scaling is generally used for standardizing a matrix (or a.k.a normalization). To standardize a matrix:

$$\text{standardize}(X) = (X - 1_N \mu^T)\,\text{diag}(\sigma)^{-1}$$

→ Why inverse?

→ Are these row or column vectors

where $\mu = \bar{x}$ (mean), σ is standard deviation of vectors.

Sum of squares

Used in 4.59, 7.236, 11.6

The sum of squares matrix for $X \in \mathbb{R}^{D \times D}$

$$S_0 \triangleq X^T X = \sum_{n=1}^{N} x_n x_n^T$$

Scatter matrix

$$S_{\bar{x}} \triangleq \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T = \left(\sum_n x_n x_n^T\right) - N \bar{x}\bar{x}^T$$

Here we are computing mean centered data ($\tilde{X}$) from X

Mean $\bar{x} = \frac{1}{N} X^T 1_N$

We can compute the centered data matrix using

$$\hat{X} = X - 1_N \bar{x}^T = X - \frac{1}{N} 1_N 1_N^T X = C_N X$$

where

$$C_N \triangleq I_N - \frac{1}{N} 1_N 1_N^T$$

C_N is the centering matrix

Scatter matrix can now be computed as:

$$S_{\bar{x}} = \tilde{X}^T \tilde{X} = X^T C_N^T C_N X = X^T C_N X$$

What is the use of Scatter matrix?

- $S_{\bar{x}}$ is the unnormalized covariance matrix
- Covariance matrix

$$C = \frac{1}{N-1} S_{\bar{x}}$$

Example:

$$C_0 = [0]$$

$$C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2/3 & -1/3 & -1/3 \\ -1/3 & 2/3 & -1/3 \\ -1/3 & -1/3 & 2/3 \end{bmatrix}$$

Gram Matrix

A matrix of inner products for x is called Gram Matrix (K)

$$K \triangleq XX^T$$

Inner products of mean-centered data vectors

$$\tilde{K} = \tilde{X}\tilde{X}^T$$

- 'K' is the feature similarity matrix. If we're only given K and we would like to model the data, then first we need to normalize the data.
- In cases, where X is not provided and only 'K' provided, we can normalize the data without $\tilde{x}$ through centering matrix 'C'
- Double centering is a trick which involves multiplying centering matrix on both sides to achieve zero mean in the data. We double center $K \rightarrow \tilde{K}$ as follows

$$\tilde{K} = \tilde{X}\tilde{X}^T = C_N K C_N$$

Distance Matrix

Let $x \in R^{N \times P}$ $Y \in R^{N \times P}$, the distance between the elements is computed as

$$D_{ij} = (x_i - y_j)^T (x_i - y_j) = \|x_i\|^2 - 2x_i^T y_j + \|y_j\|^2$$

In matrix, $\|x_i\|^2$ can be computed by $\text{diag}(XX^T) = \hat{x}$

Similarly, $\|y_j\|^2 = \text{diag}(YY^T) = \hat{y}$

Finally, the distance matrix can be computed as follows:

$$D = \hat{x} 1_{Ny}^T - 2XY^T + 1_{N_x} \hat{y}^T$$

How 1_{N_x} and 1_{Ny} is interpreted?

If $X = Y$, then

$$D = \hat{x} 1_N^T - 2xx^T + 1_N \hat{x}^T$$

7.2.5 Kronecker Products

If $A \in R^{m \times n}$, $B \in R^{p \times q}$, the kronecker product is given as

$$A \otimes B \in R^{mp \times nq}$$

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}$$

Check textbook for example

Properties:

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

$$(A \otimes B)\,\mathrm{vec}(C) = \mathrm{vec}(BCA^T)$$

7.2.6 Einstein Summation

a.k.a einsum is a short-hand notation for specifying operations on matrix, vector, and tensors.

For example:

$$C_{ij} = \sum_k A_{ik} B_{kj}$$

can be written as

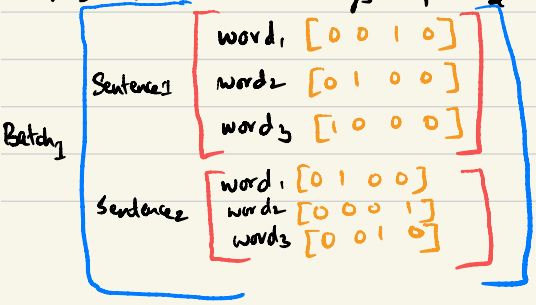
$$C_{ij} = A_{ik} B_{kj}$$

We just dropped $\sum_k$

In programming, for example in numpy

```
C = np.einsum('ik,kj->ij', A, B)
```

Similar operations can be performed on tensor. Consider a tensor which has a batch (B) of sentences (M) where each sentence has word embeddings of size (D). To illustrate



Let $W \in R^{D \times P}$ is a word embedding matrix. We can multiply as:

$$E_{bmp} = S_{bmd} W_{dp}$$

https://obilaniu6266h16.wordpress.com/2016/02/04/einstein-summation-in-numpy/

7.3 Matrix Inversion

For matrix $A \in R^{n \times n}$, the inverse is A^{-1}

$$A^{-1}A = AA^{-1} = I$$

A^{-1} exists iff $\det(A) \neq 0$. If $\det(A) = 0$, then A is singular matrix

Properties

$$(A^{-1})^{-1} = A$$

$$(AB)^{-1} = B^{-1}A^{-1}$$

$$(A^{-1})^{T} = (A^{T})^{-1} \triangleq A^{-T}$$

For $A \in R^{2 \times 2}$, the A^{-1} will be

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad A^{-1} = \frac{1}{\|A\|}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

For block diagonal

$$\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} & 0 \\ 0 & B^{-1} \end{pmatrix}$$

7.4 Eigenvalue Decomposition (EVD)

Let $A \in R^{n \times n}$, then the eigenvalue of A is denoted as $\lambda \in R$ and eigenvector of A will be $u \in R^{n}$, only if

$$Au = \lambda u, \quad u \neq 0$$

Intuitively what this means is for a given vector 'u' if we multiply with a matrix we will get a new vector as the same size of u but only scaled by a factor of λ.

If A is a rotation matrix, u is the axis of rotation, then multiplying the two matrices will yield a new scale for axis of rotation. ($\lambda = 1$)

The scalar component $c \in R$ for eigenvector is:

$$A(cu) = cAu = c\lambda u = \lambda cu$$

We usually assume eigenvector is normalized to have length 1.

(λ, u) is an eigenvalue – eigenvector pair of A only if

$$(\lambda I - A)u = 0 \quad (u \neq 0)$$

⇝ How? $\lambda Iu - Au = 0$, $Au = \lambda Iu$

$(\lambda I - A)u = 0$ will have non-zero solution to u iff $(\lambda I - A)$ has a non-empty null space, which will be the case if $(\lambda I - A)$ is singular.

i.e., $\det(\lambda I - A) = 0$

Remember: we're saying singular only for $\lambda I - A$. However, A still needs to be non-singular i.e $|A| \neq 0$

- This called characteristic equation of A.
- The n solutions of this eqn^n are n eigenvalues λ_i and u_i are corresponding eigenvectors
- It is standard to sort the eigenvalues by their magnitude.

Somewhat closely reminds of why we consider only the first k principal components of a matrix to represent the whole matrix

Properties:

$$tr(A) = \sum_i^n \lambda_{ii}$$

$$\det(A) = \prod_{i=1}^n \lambda_i$$

$$rank(A) = nnz(\lambda)$$

eigenvalue of $A^{-1} = 1/\lambda_i$ and eigenvector $A^{-1}u_i = 1/\lambda_i u_i$

In triangle matrix or diagonal matrix, eigenvalues are the diagonal values.

Diagonalization

We can write all the eigenvector equations as

$$AU = U\Lambda$$

⇝ How? Note: previously we have realized there could be many equations from the matrix.

where $U \in \mathbb{R}^{n \times n}$

$$U = \begin{bmatrix} | & | & | & & | \\ u_1 & u_2 & u_3 & \cdots & u_n \\ | & | & | & & | \end{bmatrix} \quad \Lambda = diag(\lambda_1, \cdots \lambda_n)$$

If the eigenvectors of A are linearly independent, then matrix U will be invertible (which is basically going other side as inverse form)

$$A = U\Lambda U^{-1}$$

⇝ Any matrix in this form is called Diagonalizable.

7.4.3 Eigenvalues and Eigenvectors of Symmetric matrices

If A is a symmetric matrix, then the eigenvectors can be shown as orthonormal i.e.,

$$u_i^T u_j = 0 \quad \text{iff} \quad i \neq j$$

$$u_i^T u_i = u_i u_j = 1 \quad \text{iff} \quad i = j \text{ (diagonal elements)}$$

Therefore, $U U^T = U^T U = I$ ⇝ Hence, U is orthogonal matrix

Note previously we represented $A = U \Lambda U^{-1}$

Now we are saying A can also be represented as

$$A = U \Lambda U^T$$

$$= \begin{pmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_n \\ | & | & & | \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix} \begin{pmatrix} - & u_1^T & - \\ - & u_2^T & - \\ & \vdots & \\ - & u_n^T & - \end{pmatrix}$$

$$= \sum_{i=1}^{n} \lambda_i u_i u_i^T$$

Here the assumption is $U^T = U^{-1}$.

With that assumption, the A^{-1} will be

$$A^{-1} = U \Lambda^{-1} U^{-1} = \sum_{i=1}^{n} 1/\lambda_i \, u_i u_i^T$$

Checking For Positive Definiteness

Recall positive definiteness is calculated by

$$x^T A x > 0$$

Similarly we can substitute A with eigenvalue & eigenvector

$$x^T A x = x^T U \Lambda U^T x \qquad \text{Let } y = U^T x \quad \because y^T = x^T U$$

$$= y^T \Lambda y$$

$$= \sum_{i=1}^{n} \lambda_i y_i^2$$

$\because y_i^2$ will always be positive, the sign of λ_i determines the definiteness of matrix. i.e., if all (λ_i)

$\lambda_i > 0$ +ve definite $\quad \lambda_i \geq 0$ +ve semidefinite.

$\lambda_i < 0$ −ve definite $\quad \lambda_i \leq 0$ −ve semidefinite

if λ's contains +ve & −ve, then indefinite matrix.

7.4.4 Geometry of Quadratic Forms

Quadratic form is a function written for a matrix A as

$$f(x) = x^T A x$$

We know $A = UAU^T$, so

$$f(x) = x^T A x = x^T UAU^T x = y^T \Lambda y = \sum_{i=1}^{n} \lambda_i y_i^2$$

The level sets of $f(x)$ define hyper-epsilloids

For example: $\lambda_1 y_1^2 + \lambda_2 y_2^2 = r$ (eqn for a 2d ellipse)

Eigenvectors determine orientation of the ellipse. and eigenvalues determine how elongated it is.

7.4.5 Standardizing and whitening data

- Standardizing is a preprocessing technique applied to the data so that each column has zero mean and unit variance.
- Standardizing data will have unit variance but it doesnot remove the correlation between the columns.
- To remove correlation, we approach a process called "Whitening".
- Let's say, if we are training on images, the raw input is redundant, since adjacent pixel values are highly correlated.
- The goal of whitening is to make the input less redundant. i.e., features are less correlated and have same variance.

Let the covariance matrix of data be

$$\Sigma = \frac{1}{N} X^T X$$

Now Σ can be diagonalized as

$$\Sigma = EDE^T$$

→ Which is basically decomposing any matrix into eigenvector & eigenvalues

Parallely, we decompose X using SVD

$$X = USV^T$$

When we compare the decomposed forms,

$$\boxed{E = V \quad \text{and} \quad D = S^2}$$ ⇝ How?

We define PCA whitening matrix as:

$$W_{pca} = D^{-1/2} E^T$$ ⇝ Pulled out of thin air

Now we construct a transformed vector

$y = W_{pca} x$ which will be a decorrelated version of x

To check the covariance of the whitening:

$$Cov[y] = W E[xx^T] W^T = W \Sigma W^T$$

$$= (D^{-1/2} E^T)(EDE^T)(ED^{-1/2}) = I$$

We know $W_{pca} = D^{-1/2} E^T$

$\Sigma = EDE^T$

This is W_{pca}^{-1}

Rotating the whitening matrix will still maintain the whitening property

If R is rotation, then $W = RW_{pca}$ is the new rotated matrix. The whitening property $W^T W = \Sigma^{-1}$ is preserved.

We know $W_{pca} = D^{-1/2} E^T$

Let's assume we rotated $W_{zca} = R W_{pca} = R D^{-1/2} E^T$

Let's also assume $R = E$

$$W_{zca} = E D^{-1/2} E^T = \Sigma^{-1/2} = V S^{-1} V^T$$

$\Sigma = EDE^T$

This is called Mahalanbois whitening or ZCA (Zero phase component Analysis)

- When applied to images the ZCA transformed vectors will still look like images.

7.4.6 Power Method

Power method is a simple iterative method to compute the largest eigenvalue and the corresponding vector for a matrix.

Let A be a matrix with orthonormal eigenvectors and eigenvalues. We know $|\lambda_1| > |\lambda_2| > \dots |\lambda_m| \geq 0$.

So $A = U \Lambda U^T$. Let v_0 be a random vector within the same range as A. We can assume that $Ax = v_0$ for some suitable x. Hence we can write v_0 in the form of A as:

$$v_0 = U(\Lambda U^T x) = a_1 u_1 + \dots + a_m u_m \quad \text{(Here 'a}_i\text{' are scalars} \in \mathbb{R}^n)$$

$$a \in A \in \mathbb{R}^{m \times m}$$

Now here comes the trick. If we repeatedly multiply v with A at certain time 't', we would have $v_{t+1} \simeq v_t$ i.e., $|v_{t+1} - v_t| < \epsilon$ where ϵ is often known as 'tolerance' is a very small value eg: $\epsilon = 1e^{-5}$. Formally

$$v_t \propto A v_{t-1}$$

(Note: We normalize v at each time step after multiplying with A $v_t = v_t / |v_t|$)

$$v_0 \propto a_1 \lambda_1^0 u_1 + a_2 \lambda_2^0 u_2 + \dots + a_m \lambda_m^0 u_m$$

$$v_1 \propto a_1 \lambda_1^1 u_1 + a_2 \lambda_2^1 u_2 + \dots + a_m \lambda_m^1 u_m$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$v_t \propto a_1 \lambda_1^t u_1 + a_2 \lambda_2^t u_2 + \dots + a_m \lambda_m^t u_m$$

$$= \lambda_1^t \left(a_1 u_1 + a_2 (\lambda_2/\lambda_1)^t u_2 + \dots + a_m (\lambda_m/\lambda_1)^t u_m \right)$$

$$= \lambda_1^t a_1 u_1 \quad \rightsquigarrow \text{because } \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots$$

$$\text{and } \lambda_1 \leq \frac{\lambda_2}{\lambda_1} \leq \frac{\lambda_3}{\lambda_1} \leq \dots < 1$$

Which suggests that a random vector v_0 will eventually converge to v_1 (the largest eigen vector of A) at some timestep 't'. The only requirement being $v_0^T v_1 \neq 0$, which will be true for random v_0 with high probability.

What does it mean when a random vector v_0 to U is zero?
Well, if you look interms of cosine similarity $= \frac{a^T a}{|a^T a|}$
If the $v_0^T v_1 = 0$, then the vector doesn't belong in the matrix space.

Previously we stated that $Ax = V_{(0)}$ for some value of x. Let's say we do several iterations of v_t i.e., $v_t \propto Av_{t-1}$. For some timestep t we have v_t such that

$$v_t \approx k \cdot x \quad \text{and remember } v_t \propto \lambda_1^t a_1 u_1$$

where x is our approximation of a dominant eigenvector of A.

We now use Rayleigh quotient to find the eigen value. The Rayleigh Quotient is defined as

$$R(A, x) \triangleq \frac{x^T A x}{x^T x}$$

Hence for u_i as x

$$R(A, u_i) = \frac{u_i^T A u_i}{u_i^T u_i} = \frac{u_i^T \lambda_i u_i}{u_i^T u_i} = \frac{\lambda_i u_i^T u_i}{u_i^T u_i} = \lambda_i$$

→ We know $Ax = \lambda x$ so $Au_i = \lambda u_i$

7.4.7 **Deflation**

Using power method, we obtain first eigenvector and eigenvalue. In order to obtain the following values we approach the deflation trick.

For $(\lambda_2, v_2) \Rightarrow B = (I - u_1 u_1^T)A$

$$= A - u_1 u_1^T A$$

$$= A - \lambda_1 u_1 u_1^T$$

Note: $u_1 u_1^T$ is an outer product which will yield a $m \times m$ matrix of the same size(A).

Using the matrix 'B', we apply power method to compute λ_2 and v_2.

Similarly for (λ_3, v_3), (λ_4, v_4), ...

For $(\lambda_3, v_3) \Rightarrow C = B - \lambda_2 v_2 v_2^T$

For $(\lambda_4, v_4) \Rightarrow D = C - \lambda_3 v_3 v_3^T$

& so on.

Deflation technique is used to implement PCA and Sparse PCA.

7.4.8 Eigenvectors optimize quadratic forms

This is where we dive into matrix calculus to solve an optimization problem.

Here is the following constrained optimization problem:

$$\max_{x \in \mathbb{R}^n} \; x^T A x \qquad \text{subject to } \|x\|_2^2 = 1$$

Why $\|x\|_2^2$?

One way to solve such optimization problem is through Langrangian Optimization.

To maximize / minimize a function $f(x)$ which is subject to $g(x)$, the Langrange is defined as

$$\mathcal{L}(x, y) \triangleq f(x) + \lambda g(x)$$

This is called as Langrange multiplier

$\|x\|_2$ is the L2-norm. $\|x\|_2 = 1$ indicates unit sphere. Although squaring on both sides does not change the fact. We'll have $\|x\|_2^2 = 1^2 \Rightarrow x^T x = 1$ (we know $x^T x$ is matrix squared)

What happens if choose $\|x\|_2^2 > 1$ (like 100)? We could and still be able to solve but I guess it just complicates I guess.

To solve the quadratic form, the Langrangian is given as:

$$\mathcal{L}(x, y) = \underbrace{x^T A x}_{f(x)} + \lambda \underbrace{(1 - x^T x)}_{g(x)}$$

Thereby, we can establish that x^* to be an optimal point to the problem. That is by finding the point where the gradient of Langrangian is at zero. Formally given as

$$\nabla_x \mathcal{L}(x, \lambda) = 2A^T x - 2\lambda x = 0$$

The above equation is familiar

$$2A^T x - 2\lambda x = 0$$

$$2A^T x = 2\lambda x$$

$$A^T x = \lambda x$$

Remember that, Eigenvectors (A) = eigenvectors (A^T)

The only points which could maximize or minimize A are its eigenvectors.

Hence, we are essentially finding the eigenvectors that maximize or minimize through Langrangian optimization. We don't need to perform Langrangian optimization to find the vectors responsible for maximizing or minimizing the quadratic form. This is more of a theoretical confirmation.

But also what other forms we might encounter which are other than ellipsoid shapes? For such forms Langrangian is one way to find the vectors responsible.

7.5 Singular Value Decomposition (SVD)

So far we have discussed Eigen Value Decomposition (EVD) for saluare matrices. Now we will look into generalizing this decomposition for rectangular matrices.

7.5.1 Basics

A matrix $A \in \mathbb{R}^{m \times n}$ can be decomposed as

$$A = USV^T$$

$$= \sigma_1 \begin{pmatrix} | \\ u_1 \\ | \end{pmatrix} \begin{pmatrix} - & v_1^T & - \end{pmatrix} + \sigma_2 \begin{pmatrix} | \\ u_2 \\ | \end{pmatrix} \begin{pmatrix} - & v_2^T & - \end{pmatrix} + \cdots + \sigma_r \begin{pmatrix} | \\ u_r \\ | \end{pmatrix} \begin{pmatrix} - & v_r^T & - \end{pmatrix}$$

- Here $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ and $S \in \mathbb{R}^{m \times n}$
- The matrix S contains $r = \min(m,n)$ singular values which are $\sigma_i \geq 0$ ($\sigma_i \in S$). The rest all are 0s.
- Columns of U are left singular vectors and columns of V are right singular vectors
- This is called Singular Value Decomposition.

Why don't we call these eigen values and what's the difference between them?

If a matrix consists of $m > n$ dimensions, then only n singular values are computed and the rest $m-n$ are ignored i.e., zero-ed. This leads to a simple economy sized SVD known as Thin SVD.

Time complexity for SVD: $O(\min(mn^2, m^2n))$

7.5.2 Connection between SVD and EVD

If A is a real, symmetric, and positive definite matrix, then

Singular values are equal to the eigenvalues

Singular vectors are equal to the eigenvectors

$$A = \underbrace{USV^T}_{\text{SVD}} = \underbrace{USU^T}_{\text{EVD}} = \underbrace{USU^{-1}}_{\text{EVD}}$$

For A^TA and AA^T, we can make following deductions:

$$A^TA = VS^TU^TUSV^T \qquad AA^T = USV^TVS^TU^T$$

$$A^TA = VS^TSV^T = VS^TSV^{-1} \qquad AA^T = USS^TU^T = USS^TU^{-1}$$

$$(A^TA)V = VS^TS = VD_n \quad (V^T = V^{-1}) \qquad (AA^T)U = U(SS^T) = UD_m$$

A: n, n, λ

So eigenvectors of AA^T is right singular vector and eigenvalues are D_n ($n \times n$ diagonal matrix)

Similarly eigenvectors of A^TA is left singular vector and eigenvalues are D_m ($m \times m$ diagonal matrix)

In summary,

U = eigenvector (AA^T)

V = eigenvector (A^TA)

D_m = eigenval (AA^T)

D_n = eigenval (A^TA)

For an economy sized SVD, we can define

$$D = S^2 = S^TS = SS^T$$

Note: EVD might not exist for some square matrices, however, SVD does exist for them.

7.5.3 Pseudo Inverse

Remember for a matrix to be invertible (A^{-1}), the matrix has to be a square matrix. Finding inverse of a matrix suggests that there exists a unique solution for the matrix. Similarly, to find inverse for a non-square matrix, pseudo-inverse is computed for a matrix. It is denoted as A^+.

Properties

$$AA^+A = A$$
$$A^+AA^+ = A$$
$$(AA^+)^T = AA^+$$
$$(A^+A)^T = A^+A$$

For a matrix $A \in R^{m \times n}$,

if $m == n$ (i.e., square matrix) then,

$$A^+ = A^{-1}$$

if $m > n$,

$$A^+ = (A^TA)^{-1}A^T$$

This A^+ is called Left inverse because when you multiply A on right $A^+A = I$

if $m < n$

$$A^+ = A^T(AA^T)^{-1}$$

This is called right inverse when you multiply A on left $AA^+ = I$

Here A is full-rank matrix
i.e rank $(A) = \min(m, n)$.
Full rank ~ linearly independent vectors

Computing A^+ using SVD:

$$A = USV^T \Rightarrow A^+ = VS^TU^T$$
$$A^+ = A^{-1} = (USV^T)^{-1} = VS^{-1}U^T$$

So computing SVD decomposition and inverting would yield A^+. Here $S^{-1} = \text{diag}(\sigma_1^{-1}, \dots \sigma_r^{-1}, 0, \dots 0)$

Computes upto rank ↳ rank

7.5.4 **SVD and the range and null space of a matrix**

Here we show that left and right vectors form an orthonormal basis for the range and null space.

Let's unpack this statement for a bit.

What does it mean vectors form an orthonormal basis?

- Two vectors are othogonal if their dot products are equal to zero. Basis is the set of linearly independent vectors. Orthonormal basis is set of linearly independent vectors which are orthogonal and their unit norm is 1. Remember, unit norm, usually for columns of a matrix, is the total length of vectors in euclidean space. We can also say such matrix is normalized or standardized.

Now what does it mean orthonormal basis for range and null space?

- Recall that Range is the set of columns that can be written as a linear combination to generate a vector.
- Nullspace is the set of columns that map to zero to null space

We know

$$A = USV^T$$

$$Ax = (USV^T)x = \sum_{j:\sigma_j>0} \sigma_j (v_j^T x) u_j = \sum_{j=1}^{r} \sigma_j (v_j^T x) u_j$$

r is the rank of A

Dimension : r

Therefore, from this we can say Ax is the linear combination of Left singular vectors $u_1 \dots u_r$ i.e., range(A) = Span$(\{u_j : \sigma_j > 0\})$

To find null space, let's consider vector $y \in \mathbb{R}^n$ that is a linear combination solely of right singular vectors for zero singular values

$$y = \sum_{j:\sigma_j=0} c_j v_j = \sum_{j=r+1}^{n} c_j v_j$$

Dimension : $n-r$

Therefore nullspace(A) = span$(\{v_j : \sigma_j = 0\})$

Dimension(range(A)) + Dimension(nullspace(A)) = $r + n - r = n$

i.e., rank + nullity = n

This is called rank-nullity theorem.

7.5.5 Truncated SVD

We know

$$A = USV^T$$

If we pick the top k singular values then,

$$A_k = U_k S_k V_k^T$$

The error between A and A_k

$$\|A - A_k\|_2 \sim 0$$

If the error is close to zero, then it is a good approximation.

Usually when the $k = r$ i.e., Rank(A), then error will be close to zero

If $k < r$ and error is negligible, which means certain non-zero singular values are dieing towards zero. (Common in Natural data)

We refer $k < r$ decomposed matrix as Truncated SVD.

Total parameters needed to represent a $M \times N$ matrix using a rank k approximation is

k vectors of M + k vectors of N + k Singular values

$$Mk + Nk + k = k(M+N+1) \text{ parameters}$$

For ex: 100×50 matrix has SVD. Let $k = 10$, then

$10(100+50+1) = 1510$ parameters. Which is $\frac{1510}{3000} = 36\%$ of original matrix size.

7.6.1 LU Factorization

An matrix can be factorized into upper and lower triangular matrices

$$A = LU$$

In L the upper diagonal values will be zero

In U the lower diagonal values will be zero

However, before applying LU Factorization we may need to apply permute the entries. The reason being, let's assume the following matrix

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

If $a_{11} = 0$, then either l_{11} or $u_{11} = 0$ which indicates either L or U is singular which means L or U doesn't have unique solution.

To avoid this, the matrix needs to be permuted so that the a_{11} is non-zero. This is repeated for subsequent steps. We denote this process as

$$PA = LU$$

where P is a permutation matrix where $P_{ij} = 1$ if row j gets permuted to row i. This is called partial pivoting.

7.6.2 QR Decomposition

Another way to decompose a matrix $A \in \mathbb{R}^{m \times n}$ is by finding a series of orthonormal vectors $q_1, q_2, \ldots$ that span successive subspaces of A columns i.e.,

$$\begin{pmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{pmatrix} = \underbrace{\begin{pmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_n \\ | & | & & | \end{pmatrix}}_{\text{orthonormal vectors}} \underbrace{\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \\ & & & r_{mn} \end{pmatrix}}_{\text{coefficients.}}$$

Here we can write

$$a_1 = r_{11} q_1$$
$$a_2 = r_{12} q_1 + r_{22} q_2$$
$$\vdots$$
$$a_n = r_{1n} q_1 + \cdots + r_{nn} q_n$$

Lower triangular entries are zero

We notate as

$A = \hat{Q}\hat{R}$ where $\hat{Q} \in \mathbb{R}^{m \times n}$, $\hat{R} \in \mathbb{R}^{n \times n}$ → Reduced QR or economy sized QR

$A = QR$ where $Q \in \mathbb{R}^{m \times (n+m-n)}$ i.e., $Q \in \mathbb{R}^{m \times m}$, $\hat{R} \in \mathbb{R}^{n \times n}$ → Full QR

This is a square matrix which satisfies $QQ^T = Q^TQ = I$

7.6.3 Cholesky Decomposition

In this decomposition, any matrix can be factorized into

$A = R^T R$ ~ Also can be written as $A = LL^T$ $(L = R^T)$

where R is upper triangle matrix

This looks similar to LU factorization. ie., $A = LU$. But remember in LU factorization, the $L \neq U$.

Cholesky Factorization is also known as Matrix Square Root
↳ why?

Time Complexity: $O(V^3)$

Application: Sampling from MVN

Given a matrix $A \in R^{m \times m}$ and $\Sigma = AA^T \in R^{m \times m}$.

Here Σ is the covariance of A. We can then decompose this covariance matrix Σ using cholesky decomposition. Let $L \in R^{m \times m}$ be the decomposed matrix.

So we would like to get a random multivariate normal distribution $y \sim \mathcal{N}(\mu, \Sigma)$ where μ is the random mean $\in R^m$.

We could totally acheive this by simply random sampling

$x \sim \mathcal{N}(0, I)$ and then set

$$y = Lx + \mu$$

↓

This yields random multivariate normal distribution which has the column mean ~ random sample mean and you can change the dimension of x and μ while keeping the 'L' constant to randomly sample multivariate distributions from this technique.

Cholesky decomposition can capture the desired correlations faithfully and can be used for generating correlated data.

- x is random matrix
- L is obtained from covariance matrix decomposition of random matrix.
- The L make sure the features responsible for constructing Σ ie (A) are also responsible for generating new matrices with a random vector.
- Which is why covariance of Σ and y are same.
- Now covariance will remain same for any size of y wrt to Σ

7.1 Solving systems of linear equations

Let's consider the following equations

$$3x_1 + 2x_2 - x_3 = 1$$
$$2x_1 - 2x_2 + 4x_3 = -2$$
$$-x_1 + \tfrac{1}{2}x_2 - x_3 = 0$$

which can be represented as

$$Ax = b$$

$$A = \begin{pmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ -1 & \tfrac{1}{2} & -1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}$$

Solving for x yield $x = [1, -2, -2]$

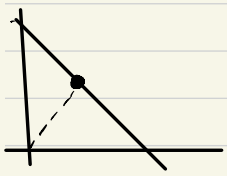
In general, a matrix $A \in R^{m \times n}$

if $m = n$, then its full Rank and have atleast 1 unique sol^n.

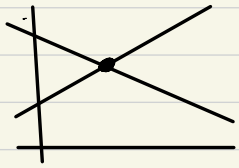
if $m < n$, then system is undetermined, so no unique sol^n.

if $m > n$, then system is over determined as there are more constraints than unknown. So no lines will intersect

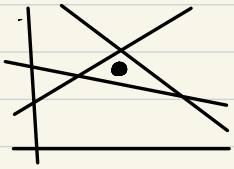
Let's assume $n = 2$



$m = 1$ (undetermined)



$m = 2$ (unique solution)



$m = 3$ (no unique solution. as over determined)

7.1.1 Solving square systems

When $m = n$, we can solve the system of linear equations using LU decomposition i.e.,

$$Ax = b$$
$$LUx = b \quad (\because A = LU)$$
$$Ux = L^{-1}b$$
Let $y \triangleq L^{-1}b$
$$Ux = y \Rightarrow x = U^{-1}y$$

therefore, solving y should be sufficient enough. $y = L^{-1}b$ or $Ly = b$

evaluating $L_{11}y_1 = b_1$ will help recursively solve the rest.

7.7.3 Solving overconstrained systems (least square estimation)

When $m > n$, we have an overdetermined solution. So we'll typically not have an exact solution. But we may have an approximate solution. For a given system $Ax = b$, we can approximate a solution by reducing the cost function. ie., least squares objective

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2$$

↝ What happens to the rest of the derivations if cross entropy is chosen?

The gradient for this cost function will be

$$g(x) = \frac{\partial}{\partial x} f(x) = A^T(Ax - b)$$

An optimum can be found by solving $g(x) = 0$

$$A^T(Ax - b) = 0$$

$$A^T Ax = A^T b$$

$$\hat{x} = (A^T A)^{-1} A^T b$$

We know that, $(A^T A)^{-1} A^T = A^+$ (left pseudo inverse)

↘ This is the ordinary least squares solution

In order to verify whether the solution is unique, we can differentiate $g(x)$, which is basically known as "Hessian" is given as:

$$H(x) = \frac{\partial}{\partial x} g(x) = \frac{\partial^2}{\partial x^2} f(x) = A^T A$$

- If $H(x)$ is positive definite, then we'll have a unique solution.
- If A is full rank, then H will be positive definite, since for any random vector > 0

$$v^T(A^T A)v = (Av)^T(Av) = \|Av\|^2 > 0$$

- Therefore in the case full rank, the least squares objective has unique global minimum.
- So we can show that the gradient plots for full rank matrices will be a ∪ curve with no bumps or intermediate valleys like this ∿

7.8 Matrix Calculus

Calculus deals with computing "rates of changes" of functions as we vary their inputs.

$f(x_1) = \hat{y}_1 \qquad f(x_2) = \hat{y}_2$

what changed to cause the new observation

7.8.1 Derivatives

Derivatives are all about change in the output with respect to change in the input.

You are making coffee. You added 1 tsp of coffee. Your coffee is level 2 strong on a scale of 1–5 (based on your own theory coffee strength).

So here $x_1 = 1$ (tsp)

$y_1 = 2$ (level)

Now you add one more tsp and the coffee is level 4 strong now.

So here the new observations $x_2 = 2$ (tsp)

$y_2 = 4$ (level)

By how much the coffee strength changed w.r.to number of teaspoons.

$$\frac{\text{Change in coffee strength}}{\text{Change in number of tsps}} = \frac{4-2}{2-1} = 2$$

This is also the slope

$$\text{slope} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x}$$

Formally, we can write as

$$\frac{\Delta y}{\Delta x} = \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

Δx → aka "step size" (h)

We can write derivative of a function as

$$f'(x) \triangleq \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Remember h and Δx are same
Just trying to stick to text book notations

We can consider,

$$f(x+h) \approx f(x) + f'(x)\, h$$

New output $\approx$ old output + change in output * perturbation

Finite Difference Approximation

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} = \lim_{h \to 0} \frac{f(x+h/2) - f(x-h/2)}{h} = \lim_{h \to 0} \frac{f(x) - f(x-h)}{h}$$

forward difference → When we extra tsp of coffee $t_0 \to t_1$

central difference → When we added 3 tsps and look what happened at $t_1 \leftarrow t_2$

backward difference → When we look back at what we messed up $t_0 \leftarrow t_1$

Smaller the step size, the better the estimate

If h is too small ~ 0, then numerical errors may occur.

f' notation is called Lagrange notation.

$\frac{dy}{dx}$ notation is called Leibnitz notation

7.8.2 **Gradients**

Before look into gradients, let's understand what are partial derivatives are.

If we have more than one variable we would like to observe rate of change w.r.to each variable

eg: $x^2 + y^3 = z$

Partial derivative w.r.to x would be

$$\frac{\partial}{\partial x}(x^2 + y^3) = 2x$$

Similarly,

$$\frac{\partial}{\partial y}(x^2 + y^3) = 3x^2$$

In a matrix $A \in \mathbb{R}^{m \times n}$, we have n different features. So when there is a function output, we would like to understand rate of change w.r.to each feature as

$$\frac{\partial f}{\partial x_i} = \lim_{h \to 0} \frac{f(x + he_i) - f(x)}{h}$$

e_i is the i^{th} unit vector

The gradient of a function at a point x is the vector of its partial derivatives

$$g = \frac{\partial f}{\partial x} = \nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

For a specific point, the gradient can be written as

$$g(x^*) \triangleq \left. \frac{\partial f}{\partial x} \right|_{x^*}$$

7.8.3 Directional Derivative

Change of gradient along a direction v is given as:

$$\underbrace{D_v f(x)}_{\text{Directional derivative}} = \nabla f(x) \cdot v$$

7.8.5 Jacobian

If there is a function $f: \mathbb{R}^n \to \mathbb{R}^d$, the first-order partial derivatives of this multivariate function can be written as

$$J_f(x) = \frac{\partial f}{\partial x^T} \triangleq \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & & \\ \frac{\partial f_d}{\partial x_1} & \frac{\partial f_d}{\partial x_2} & \cdots & \frac{\partial f_d}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_d(x)^T \end{pmatrix}$$

7.8.5.1 **Multiplying Jacobians and vectors**

Jacobian vector product is multiplying vector on the right side

$$J_f(x)v = \begin{pmatrix} \nabla f_1(x) \\ \vdots \\ \nabla f_j(x)^T \end{pmatrix} v$$

where J_f is the Jacobian for $f: R^n \to R^d$

We can approximate $J_f(x)v$ to f with just two calls.

↳ $f(x) \in R^{n \times d}$

$J_f \in R^{n \times d}$, $v \in R^d$ → Same dim

$J_f(x)v \in R^{n \times d}$

So what is v & how is it approximating to f?

Vector Jacobian Product is basically left multiplying the vector ie.,

$$u^T J_f(x) = u^T \begin{pmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_d} \end{pmatrix}$$

Here $J \in R^{n \times d}$ and $u \in R^n$

If $n \geq d$, JVP is efficient

If $n \leq d$, VJP is efficient.

Jacobian of a composition

If we have composition of functions, for instance

$$h(x) = g(f(x))$$

Jacobian of $h(x)$ would be:

$$J_h(x) = J_g(f(x)) J_f(x)$$ ⇝ Chain Rule.

7.8.6 **Hessian**

Any function which is twice differentiable, we call the second partial derivative matrix of the function $f: R^d \to R$ as "Hessian"

$$H_f = \frac{\partial^2 f}{\partial x^2} = \nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

Hessian is the Jacobian of gradient.

But what does Hessian reveal?

7.8.7 Gradients of commonly used functions

Refer textbook pg. 265 - 266 for all the formulas related to mapping differential of a function into simple vector or matrix transformations These are refered as identities.

When implementing the differentials of a function, you don't need to use explicit differential library to compute the output. One can rely on the identities.