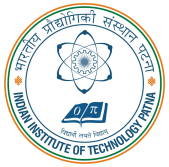
# Introduction to Deep Learning




**Arijit Mondal**
Dept. of Computer Science & Engineering
Indian Institute of Technology Patna
arijit@iitp.ac.in

# Overview of Linear Algebra

# Scalars

- A scalar is a single number
- It can be real, integer, etc.
- Typically it will be denoted using lowercase italics: $a, x, n$
- Example:
  - Let $s \in \mathbb{R}$ be the slope of the line
  - Let $n \in \mathbb{N}$ be the number of units

# Vectors

- It is an array of numbers (eg. scalars) and arranged in order
- Typically it will be denoted using lowercase bold font: $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$
- Need to specify what kind of numbers are stored
- If each element is in $\mathbb{R}$ then the vector lies in $\mathbb{R}^n$ (Cartesian product n times)
- Identify points in space, each element giving the coordinate along different axis
- A set of elements, $x_1, x_3, x_5$ can be specified as $\mathbf{x}_S$ where $S = 1, 3, 5$
- $\mathbf{x}_{-2}$ is a vector containing all elements except $x_2$

# Matrices

- A matrix is a 2D array of numbers $\mathbf{X} = [x_{i,j}] = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m,1} & x_{m,2} & \dots & x_{m,n} \end{bmatrix}$
- Example notation for type and shape $\mathbf{X} \in \mathbb{R}^{m \times n}$
- The $j$th column will be denoted as $\mathbf{x}_j$ or $X_{:,j}$ — $\mathbf{X} = \begin{bmatrix} | & | & \dots & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_n \\ | & | & \dots & | \end{bmatrix}$
- $n$-dimensional vector can be represented as n rows and 1 column $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

# Tensors

- A tensor is an array of numbers that may have
  - Zero dimensions, and be a scalar
  - One dimension, and be a vector
  - Two dimensions, and be a matrix
  - Or, more dimensions.

# Matrix transpose

- Rows and columns are interchanged that is $\mathbf{X}^T = [x_{i,j}]^T = [x_{j,i}]$
- For example, $\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \end{bmatrix}$ $\mathbf{X}^T = \begin{bmatrix} x_{1,1} & x_{2,1} \\ x_{1,2} & x_{2,2} \\ x_{1,3} & x_{2,3} \end{bmatrix}$
- Mirror image of matrix across the main diagonal
- For scalars,

# Matrix transpose

- Rows and columns are interchanged that is $\mathbf{X}^T = [x_{i,j}]^T = [x_{j,i}]$
- For example, $\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \end{bmatrix}$ $\mathbf{X}^T = \begin{bmatrix} x_{1,1} & x_{2,1} \\ x_{1,2} & x_{2,2} \\ x_{1,3} & x_{2,3} \end{bmatrix}$
- Mirror image of matrix across the main diagonal
- For scalars, $a = a^T$

# Matrix transpose

- Rows and columns are interchanged that is $\mathbf{X}^T = [x_{i,j}]^T = [x_{j,i}]$
- For example, $\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \end{bmatrix}$ $\mathbf{X}^T = \begin{bmatrix} x_{1,1} & x_{2,1} \\ x_{1,2} & x_{2,2} \\ x_{1,3} & x_{2,3} \end{bmatrix}$
- Mirror image of matrix across the main diagonal
- For scalars, $a = a^T$
- $(\mathbf{AB})^T$

# Matrix transpose

- Rows and columns are interchanged that is $\mathbf{X}^T = [x_{i,j}]^T = [x_{j,i}]$
- For example, $\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \end{bmatrix}$ $\mathbf{X}^T = \begin{bmatrix} x_{1,1} & x_{2,1} \\ x_{1,2} & x_{2,2} \\ x_{1,3} & x_{2,3} \end{bmatrix}$
- Mirror image of matrix across the main diagonal
- For scalars, $a = a^T$
- $(\mathbf{AB})^T = \mathbf{B}^T \times \mathbf{A}^T$

# Matrix manipulation

- Matrix addition $\mathbf{C} = \mathbf{A} + \mathbf{B} \Rightarrow C_{i,j} = A_{i,j} + B_{i,j}$
- Matrix multiplication $\mathbf{C} = \mathbf{A} \times \mathbf{B} \Rightarrow C_{i,j} = \sum_k A_{i,k} \times B_{k,j}$
- Multiplication and addition are associative:

$$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$$
$$(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$$

- Multiplication is distributive: $\mathbf{A} \times (\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$
- Multiplication is not commutative (in general): $\mathbf{AB} \neq \mathbf{BA}$

# Matrix Dot Product

- Let us assume $Z = X \times Y$, where $X \in \mathbb{R}^{m \times n}$ and $Y \in \mathbb{R}^{n \times p}$
- Number of columns in $X$ should be equal to number of rows in $Y$
- $z_{ij} = \sum_{k=1}^{n} x_{ik} \times y_{kj}$

# Identity matrix

- All elements are 0 except for diagonal elements which are 1
- Example, $\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- $\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{I}_n \mathbf{x} = \mathbf{x}$

# Systems of equations

- Consider following equations

$$\begin{array}{rcrcr} 4x_1 & - & 5x_2 & = & -13 \\ -2x_1 & + & 3x_2 & = & 9 \end{array}$$

- This can be expressed in the form $\mathbf{Ax} = \mathbf{b}$ where

$$A = \begin{bmatrix} 4 & -5 \\ -2 & 3 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -13 \\ 9 \end{bmatrix}$$

- $\mathbf{A}_{1,:}\mathbf{x} = b_1, \mathbf{A}_{2,:}\mathbf{x} = b_2, \ldots$

# Solving system of equations

- A linear system of equations can have:
  - No solution
  - Many solutions
  - Exactly one solution: this means multiplication by the matrix is an invertible function

# Matrix inversion

- $\mathbf{A}^{-1} \times \mathbf{A} = \mathbf{I}_n$
- Solving a system of equations using inverse

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{A}^{-1}\mathbf{A}\mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} \\ \mathbf{I}_n\mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} \\ \mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} \end{aligned}$$

- Numerically unstable, but useful for abstract analysis

# Invertability

- Matrix cannot be inverted if
  - More rows than columns
  - More columns than rows
  - Redundant rows/columns ("linearly dependent","low rank")

# Linear Independence

- Column can be thought of as specifying direction from origin
- Each element of $\mathbf{x}$ specify how far we should move in each of these direction ie, $\mathbf{Ax} = \sum x_i A_{:,i}$
- Formally, this is a linear combination of the set of vectors
- Span of set of vectors is the set of all points obtainable by linear combination of the original vectors
- Solution of $\mathbf{Ax} = \mathbf{b} \Rightarrow$ Testing whether $\mathbf{b}$ is in span of column of $\mathbf{A}$
  - Span is known as column space or range of $\mathbf{A}$

# Linear Independence (contd.)

- $\mathbf{Ax} = \mathbf{b}$ to have solution for all $\mathbf{b} \in \mathbb{R}^m$, column space of $\mathbf{A}$ must be $\mathbb{R}^m$
  - $\mathbf{A}$ must have at least $m$ column ie. $n \geq m$
  - Consider $\mathbf{A}$ has size $3 \times 2$ and $\mathbf{b}$ is 3D point
    - $\mathbf{x}$ will be 2D point
    - It traces out 2D plane within $\mathbb{R}^3$
    - Equation will have solution if $\mathbf{b}$ lies in that plane
- $n \geq m$ is a necessary condition
  - Consider $2 \times 2$ matrix where both columns are the same
  - Column space is just a line in $\mathbb{R}^2$

# Linear Independence (contd.)

- A set of vectors is linearly independent if no vectors in the set is a linear combination of other vectors
  - No new points will be added if linear combination of vectors are added in the set
- Suppose column space is $\mathbb{R}^m$
  - Need to have exactly $m$ linearly independent column
  - No set of $m$ dimensional vectors can have more than $m$ mutually linearly independent column
- A square matrix with linearly dependent columns is known as singular
- A matrix to have inverse, $\mathbf{Ax} = \mathbf{b}$ has at most one solution for each value of $\mathbf{b}$
- $\mathbf{A}$ is not square but singular, it is still possible to solve $\mathbf{Ax} = \mathbf{b}$. However, matrix inversion method cannot be used

# Norms

- Measure the size of vector. It is defined as

$$L^p = \|\mathbf{x}\|_p = \left( \sum_i |x_i|^P \right)^{1/p} \qquad p \in \mathbb{R}, p \geq 1$$

- Intuitive meaning: distance of $\mathbf{x}$ from the origin
- Norm is any function $f$ that satisfies
  - $f(\mathbf{x}) = 0$

# Norms

- Measure the size of vector. It is defined as

$$L^p = \|\mathbf{x}\|_p = \left( \sum_i |x_i|^P \right)^{1/p} \qquad p \in \mathbb{R}, p \geq 1$$

- Intuitive meaning: distance of $\mathbf{x}$ from the origin
- Norm is any function $f$ that satisfies
  - $f(\mathbf{x}) = 0 \Rightarrow \mathbf{x} = 0$

# Norms

- Measure the size of vector. It is defined as

$$L^p = \|\mathbf{x}\|_p = \left( \sum_i |x_i|^P \right)^{1/p} \qquad p \in \mathbb{R}, p \geq 1$$

- Intuitive meaning: distance of $\mathbf{x}$ from the origin
- Norm is any function $f$ that satisfies
  - $f(\mathbf{x}) = 0 \Rightarrow \mathbf{x} = 0$
  - $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$ (triangle inequality)

# Norms

- Measure the size of vector. It is defined as

$$L^p = \|\mathbf{x}\|_p = \left( \sum_i |x_i|^P \right)^{1/p} \qquad p \in \mathbb{R}, p \geq 1$$

- Intuitive meaning: distance of $\mathbf{x}$ from the origin
- Norm is any function $f$ that satisfies
  - $f(\mathbf{x}) = 0 \Rightarrow \mathbf{x} = 0$
  - $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$ (triangle inequality)
  - $\forall \alpha \in \mathbb{R}$, $f(\alpha \mathbf{x})$

# Norms

- Measure the size of vector. It is defined as

$$L^p = \|\mathbf{x}\|_p = \left( \sum_i |x_i|^P \right)^{1/p} \qquad p \in \mathbb{R}, p \geq 1$$

- Intuitive meaning: distance of $\mathbf{x}$ from the origin
- Norm is any function $f$ that satisfies
  - $f(\mathbf{x}) = 0 \Rightarrow \mathbf{x} = 0$
  - $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$ (triangle inequality)
  - $\forall \alpha \in \mathbb{R}$, $f(\alpha \mathbf{x}) = \alpha f(\mathbf{x})$

# Norms (contd)

- $L^2$ norm is known as Euclidean norm
  - It is often denoted as $\|\mathbf{x}\|$ instead of $\|\mathbf{x}\|_2$
  - Squared $L^2$ norm can be determined by $\mathbf{x}^T\mathbf{x}$. This is very often used
    - Derivative of the squared $L^2$ norm depend only on the corresponding element
    - Derivative of $L^2$ depend on entire vector
  - Square $L^2$ norm is undesirable as it increases very slowly at the origin

# Norms (contd)

- Need to identify elements that are zero and elements that are non-zero but small
  - Need a function that grow at the same rate in all locations

$$L^1 = \|\mathbf{x}\| = \sum_i |x_i|$$

  - $L^1$ can be used to differentiate zero and non-zero elements
- $L^\infty$ (max norm) - Absolute value of the elements with the largest magnitude in the vector $\|\mathbf{x}\|_\infty = \max_i |x_i|$
- Frobenius norm

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$$

  - This is analogus to $L^2$ norm of vector

# Special matrices

- Diagonal matrices — Non-zero diagonal elements and rests are zero. Formally $D_{i,j} = 0, i \neq j$
  - Identity matrix
  - diag($\mathbf{v}$) — vectors using diagonal elements
  - diag($\mathbf{v}$)$\mathbf{x}$ — $x_i$ is scaled by $v_i$
  - Inversion is easy diag($\mathbf{v}$)$^{-1}$=diag($[1/\mathbf{v}_1, 1/\mathbf{v}_2, \ldots, 1/\mathbf{v}_n]^T$)
- Not all diagonal matrix be square
  - Rectangular diagonal matrix is possible
  - $\mathbf{D}\mathbf{x}$ — Scaling each element of $\mathbf{x}$
    - Concatenate some zero if $\mathbf{D}$ is taller
    - Discard some last elements if $\mathbf{D}$ is wider
- Symmetric matrix — Arises when the entries are generated by a function of two arguments that does not depend on order
  - Distance matrix $\mathbf{A}_{i,j} = \mathbf{A}_{j,i}$

# Special vectors & matrices

- Unit vector — A vector with unit norm $\|\mathbf{x}\|_2 = 1$
- For vectors $\mathbf{x}$ and $\mathbf{y}$, if $\mathbf{x}^T\mathbf{y} = 0$

# Special vectors & matrices

- Unit vector — A vector with unit norm $\|\mathbf{x}\|_2 = 1$
- For vectors $\mathbf{x}$ and $\mathbf{y}$, if $\mathbf{x}^T\mathbf{y} = 0$
  - Norm of $\mathbf{x}$ or $\mathbf{y}$ is zero
  - $\mathbf{x}$ and $\mathbf{y}$ are at $90^0$

# Special vectors & matrices

- Unit vector — A vector with unit norm $\|\mathbf{x}\|_2 = 1$
- For vectors $\mathbf{x}$ and $\mathbf{y}$, if $\mathbf{x}^T\mathbf{y} = 0$
  - Norm of $\mathbf{x}$ or $\mathbf{y}$ is zero
  - $\mathbf{x}$ and $\mathbf{y}$ are at $90^0$
- In $\mathbf{R}^n$, at most $n$ vectors may be mutually orthogonal with non-zero norm
- Vectors orthogonal and have unit norm is known as orthonormal
- Orthogonal matrix — Square matrix, rows are mutually orthonormal, columns are mutually orthonormal
  - $\mathbf{A}^T\mathbf{A} =$

# Special vectors & matrices

- Unit vector — A vector with unit norm $\|\mathbf{x}\|_2 = 1$
- For vectors $\mathbf{x}$ and $\mathbf{y}$, if $\mathbf{x}^T\mathbf{y} = 0$
  - Norm of $\mathbf{x}$ or $\mathbf{y}$ is zero
  - $\mathbf{x}$ and $\mathbf{y}$ are at $90^0$
- In $\mathbf{R}^n$, at most $n$ vectors may be mutually orthogonal with non-zero norm
- Vectors orthogonal and have unit norm is known as orthonormal
- Orthogonal matrix — Square matrix, rows are mutually orthonormal, columns are mutually orthonormal
  - $\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I}$

# Special vectors & matrices

- Unit vector — A vector with unit norm $\|\mathbf{x}\|_2 = 1$
- For vectors $\mathbf{x}$ and $\mathbf{y}$, if $\mathbf{x}^T\mathbf{y} = 0$
  - Norm of $\mathbf{x}$ or $\mathbf{y}$ is zero
  - $\mathbf{x}$ and $\mathbf{y}$ are at $90^0$
- In $\mathbf{R}^n$, at most $n$ vectors may be mutually orthogonal with non-zero norm
- Vectors orthogonal and have unit norm is known as orthonormal
- Orthogonal matrix — Square matrix, rows are mutually orthonormal, columns are mutually orthonormal
  - $\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I} \Rightarrow \mathbf{A}^T = \mathbf{A}^{-1}$
  - Orthonormal matrices are of interest as inverse computation is easy

# Eigen decomposition

- Similar to prime factorization of integer
  - $12 = 2 \times 2 \times 3$
  - 12 is not divisible by 5
- Eigen vector of square matrix $\mathbf{A}$ is a non-zero vector such that $\mathbf{Av} = \lambda\mathbf{v}$
  - $\lambda$ is a scalar and known as eigen value
  - Mostly right eigen vector is considered
  - If $\mathbf{v}$ is eigen vector, then so is $s\mathbf{v}$
    - Usually we look for unit eigen vector
- Suppose $\mathbf{A}$ has $n$ linearly independent eigen vector $\{\mathbf{v}^1, \mathbf{v}^2 \ldots, \mathbf{v}^n\}$ with corresponding eigen value $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$
  - Concatenate all eigen vector, one per column $\mathbf{V} = [\mathbf{v}^1, \mathbf{v}^2 \ldots, \mathbf{v}^n]$, similarly for $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]$
    - Eigen decomposition $\mathbf{AV} = \mathbf{V}\boldsymbol{\lambda}$

# Eigen decomposition

- Similar to prime factorization of integer
  - $12 = 2 \times 2 \times 3$
  - 12 is not divisible by 5
- Eigen vector of square matrix $\mathbf{A}$ is a non-zero vector such that $\mathbf{Av} = \lambda\mathbf{v}$
  - $\lambda$ is a scalar and known as eigen value
  - Mostly right eigen vector is considered
  - If $\mathbf{v}$ is eigen vector, then so is $s\mathbf{v}$
    - Usually we look for unit eigen vector
- Suppose $\mathbf{A}$ has $n$ linearly independent eigen vector $\{\mathbf{v}^1, \mathbf{v}^2 \ldots, \mathbf{v}^n\}$ with corresponding eigen value $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$
  - Concatenate all eigen vector, one per column $\mathbf{V} = [\mathbf{v}^1, \mathbf{v}^2 \ldots, \mathbf{v}^n]$, similarly for $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]$
    - Eigen decomposition $\mathbf{AV} = \mathbf{V}\boldsymbol{\lambda} \Rightarrow \mathbf{A} = \mathbf{V}\text{diag}(\boldsymbol{\lambda})\mathbf{V}^{-1}$

# Eigen decomposition (contd)

- Every real symmetric matrix can be decomposed into an expression using only real valued eigen vector and eigen vector

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$$

  - $\mathbf{Q}$ is orthogonal matrix correspond of eigen vector of $\mathbf{A}$
  - $\mathbf{\Lambda}$ - Diagonal matrix
    - $\Lambda_{i,i}$ is associated with eigen vector in column $i$ of $\mathbf{Q}$ ie. $\mathbf{Q}_{:i}$
  - As $\mathbf{Q}$ is orthogonal, $\mathbf{A}$ is scaling space by $\lambda_i$ in $\mathbf{v}^i$
- Real symmetric matrix is guaranteed to have eigen decomposition but not unique
  - Two or more eigen vector can have same eigen value
  - Sort the entries of $\mathbf{\Lambda}$ in descending order

# Eigen decomposition (contd)

- Matrix is said to be singular if any one of the eigen value is 0
- Eigen decomposition can be used for optimization for the expression $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x}$ subject to $\|\mathbf{x}\|_2 = 1$

# Eigen decomposition (contd)

- Matrix is said to be singular if any one of the eigen value is 0
- Eigen decomposition can be used for optimization for the expression $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x}$ subject to $\|\mathbf{x}\|_2 = 1$
  - Whenever $\mathbf{x}$ is equal to an eigen vector of $\mathbf{A}$, $f$ takes on the value of corresponding eigen value

# Eigen decomposition (contd)

- Matrix is said to be singular if any one of the eigen value is 0
- Eigen decomposition can be used for optimization for the expression $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x}$ subject to $\|\mathbf{x}\|_2 = 1$
  - Whenever $\mathbf{x}$ is equal to an eigen vector of $\mathbf{A}$, $f$ takes on the value of corresponding eigen value
- Matrices with
  - All positive eigen value — Positive definite ($\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$)
  - All positive or 0 eigen value — Positive semidefinite ($\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0$)
  - All negative eigen value — Negative definite ($\mathbf{x}^T\mathbf{A}\mathbf{x} < 0$)

# Singular Value Decomposition

- Every real matrix has a singular value decomposition but the same is not true for eigen value decomposition
- EVD — $\mathbf{A} = \mathbf{V}\text{diag}(\boldsymbol{\lambda})\mathbf{V}^{-1}$
- SVD — $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{T}$
  - $\mathbf{A} = m \times n$, $\mathbf{U} = m \times m$, $\mathbf{D} = m \times n$, $\mathbf{V} = n \times n$
  - $\mathbf{U}, \mathbf{V}$ are orthogonal
  - $\mathbf{D}$ - diagonal matrix not necessary square
    - Diagonal elements of $\mathbf{D}$ are known as singular value of $\mathbf{A}$
    - $\mathbf{U}$ is left singular vector
    - $\mathbf{V}$ is right singular vector

# Trace operator & Determinant

- Trace operator
  - $Tr(\mathbf{A}) = \sum_i A_{i,i}$

# Trace operator & Determinant

- Trace operator
  - $Tr(\mathbf{A}) = \sum_i A_{i,i}$
  - $\|\mathbf{A}\|_F =$

# Trace operator & Determinant

- Trace operator
  - $Tr(\mathbf{A}) = \sum_i A_{i,i}$
  - $\|\mathbf{A}\|_F = \sqrt{Tr(\mathbf{A}\mathbf{A}^T)}$

# Trace operator & Determinant

- Trace operator
  - $Tr(\mathbf{A}) = \sum_i A_{i,i}$
  - $\|\mathbf{A}\|_F = \sqrt{Tr(\mathbf{A}\mathbf{A}^T)}$
  - $Tr(\mathbf{A}) = Tr(\mathbf{A}^T)$

# Trace operator & Determinant

- Trace operator
  - $Tr(\mathbf{A}) = \sum_i A_{i,i}$
  - $\|\mathbf{A}\|_F = \sqrt{Tr(\mathbf{A}\mathbf{A}^T)}$
  - $Tr(\mathbf{A}) = Tr(\mathbf{A}^T)$
  - $Tr(a) = a$

# Trace operator & Determinant

- Trace operator
  - $Tr(\mathbf{A}) = \sum_i A_{i,i}$
  - $\|\mathbf{A}\|_F = \sqrt{Tr(\mathbf{A}\mathbf{A}^T)}$
  - $Tr(\mathbf{A}) = Tr(\mathbf{A}^T)$
  - $Tr(a) = a$
  - $Tr(\mathbf{ABC}) = Tr(\mathbf{CAB}) = Tr(\mathbf{BCA})$

# Trace operator & Determinant

- Trace operator
  - $Tr(\mathbf{A}) = \sum_i A_{i,i}$
  - $\|\mathbf{A}\|_F = \sqrt{Tr(\mathbf{A}\mathbf{A}^T)}$
  - $Tr(\mathbf{A}) = Tr(\mathbf{A}^T)$
  - $Tr(a) = a$
  - $Tr(\mathbf{ABC}) = Tr(\mathbf{CAB}) = Tr(\mathbf{BCA})$
    - $Tr(\prod_{i=1}^{n} \mathbf{F}^i) = Tr(\mathbf{F}^n \prod_{i=1}^{n-1} \mathbf{F}^i)$

# Trace operator & Determinant

- Trace operator
  - $Tr(\mathbf{A}) = \sum_i A_{i,i}$
  - $\|\mathbf{A}\|_F = \sqrt{Tr(\mathbf{A}\mathbf{A}^T)}$
  - $Tr(\mathbf{A}) = Tr(\mathbf{A}^T)$
  - $Tr(a) = a$
  - $Tr(\mathbf{ABC}) = Tr(\mathbf{CAB}) = Tr(\mathbf{BCA})$
    - $Tr(\prod_{i=1}^{n} \mathbf{F}^i) = Tr(\mathbf{F}^n \prod_{i=1}^{n-1} \mathbf{F}^i)$
  - $Tr(\mathbf{AB}) = Tr(\mathbf{BA})$ where $\mathbf{A} = m \times n$ and $\mathbf{B} = n \times m$

# Trace operator & Determinant

- Trace operator
  - $Tr(\mathbf{A}) = \sum_i A_{i,i}$
  - $\|\mathbf{A}\|_F = \sqrt{Tr(\mathbf{A}\mathbf{A}^T)}$
  - $Tr(\mathbf{A}) = Tr(\mathbf{A}^T)$
  - $Tr(a) = a$
  - $Tr(\mathbf{ABC}) = Tr(\mathbf{CAB}) = Tr(\mathbf{BCA})$
    - $Tr(\prod_{i=1}^{n} \mathbf{F}^i) = Tr(\mathbf{F}^n \prod_{i=1}^{n-1} \mathbf{F}^i)$
  - $Tr(\mathbf{AB}) = Tr(\mathbf{BA})$ where $\mathbf{A} = m \times n$ and $\mathbf{B} = n \times m$
- Determinant of $\mathbf{A}$ is denoted as $\det(\mathbf{A})$
  - Defined only for square matrix
  - Product of all eigen value of the matrix

# Principal Component Analysis

- We have $m$ points $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m\}$ in $\mathbb{R}^n$
- Representing these points using a lossy compression

# Principal Component Analysis

- We have $m$ points $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m\}$ in $\mathbb{R}^n$
- Representing these points using a lossy compression
  - For each point choose a lower dimension ie. $\mathbf{x}^i \in \mathbb{R}^n \rightarrow \mathbf{c}^i \in \mathbb{R}^l$

# Principal Component Analysis

- We have $m$ points $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m\}$ in $\mathbb{R}^n$
- Representing these points using a lossy compression
  - For each point choose a lower dimension ie. $\mathbf{x}^i \in \mathbb{R}^n \rightarrow \mathbf{c}^i \in \mathbb{R}^l$
  - Target is to find out $f$ such that $f(\mathbf{x}) = c$ and a decode function $g$ such that $\mathbf{x} \approx g(f(\mathbf{x}))$

# Principal Component Analysis

- We have $m$ points $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m\}$ in $\mathbb{R}^n$
- Representing these points using a lossy compression
  - For each point choose a lower dimension ie. $\mathbf{x}^i \in \mathbb{R}^n \rightarrow \mathbf{c}^i \in \mathbb{R}^l$
  - Target is to find out $f$ such that $f(\mathbf{x}) = c$ and a decode function $g$ such that $\mathbf{x} \approx g(f(\mathbf{x}))$
- Let $g(\mathbf{c}) = \mathbf{D}\mathbf{c}$ where $\mathbf{D} \in \mathbb{R}^{n \times l}$
  - PCA constraints the column of $\mathbf{D}$ to be orthogonal

# Principal Component Analysis

- We have $m$ points $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m\}$ in $\mathbb{R}^n$
- Representing these points using a lossy compression
  - For each point choose a lower dimension ie. $\mathbf{x}^i \in \mathbb{R}^n \rightarrow \mathbf{c}^i \in \mathbb{R}^l$
  - Target is to find out $f$ such that $f(\mathbf{x}) = c$ and a decode function $g$ such that $\mathbf{x} \approx g(f(\mathbf{x}))$
- Let $g(\mathbf{c}) = \mathbf{D}\mathbf{c}$ where $\mathbf{D} \in \mathbb{R}^{n \times l}$
  - PCA constraints the column of $\mathbf{D}$ to be orthogonal
    - $\mathbf{D}$ is not orthogonal matrix

# Principal Component Analysis

- We have $m$ points $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m\}$ in $\mathbb{R}^n$
- Representing these points using a lossy compression
  - For each point choose a lower dimension ie. $\mathbf{x}^i \in \mathbb{R}^n \rightarrow \mathbf{c}^i \in \mathbb{R}^l$
  - Target is to find out $f$ such that $f(\mathbf{x}) = c$ and a decode function $g$ such that $\mathbf{x} \approx g(f(\mathbf{x}))$
- Let $g(\mathbf{c}) = \mathbf{D}\mathbf{c}$ where $\mathbf{D} \in \mathbb{R}^{n \times l}$
  - PCA constraints the column of $\mathbf{D}$ to be orthogonal
    - $\mathbf{D}$ is not orthogonal matrix
    - For unique solution column of $\mathbf{D}$ have unit norm

# Principal Component Analysis

- We have $m$ points $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m\}$ in $\mathbb{R}^n$
- Representing these points using a lossy compression
  - For each point choose a lower dimension ie. $\mathbf{x}^i \in \mathbb{R}^n \rightarrow \mathbf{c}^i \in \mathbb{R}^l$
  - Target is to find out $f$ such that $f(\mathbf{x}) = c$ and a decode function $g$ such that $\mathbf{x} \approx g(f(\mathbf{x}))$
- Let $g(\mathbf{c}) = \mathbf{D}\mathbf{c}$ where $\mathbf{D} \in \mathbb{R}^{n \times l}$
  - PCA constraints the column of $\mathbf{D}$ to be orthogonal
    - $\mathbf{D}$ is not orthogonal matrix
    - For unique solution column of $\mathbf{D}$ have unit norm
- Generate optimal code point $\mathbf{c}^*$ for each $\mathbf{x}$

# Principal Component Analysis

- We have $m$ points $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m\}$ in $\mathbb{R}^n$
- Representing these points using a lossy compression
  - For each point choose a lower dimension ie. $\mathbf{x}^i \in \mathbb{R}^n \rightarrow \mathbf{c}^i \in \mathbb{R}^l$
  - Target is to find out $f$ such that $f(\mathbf{x}) = c$ and a decode function $g$ such that $\mathbf{x} \approx g(f(\mathbf{x}))$
- Let $g(\mathbf{c}) = \mathbf{D}\mathbf{c}$ where $\mathbf{D} \in \mathbb{R}^{n \times l}$
  - PCA constraints the column of $\mathbf{D}$ to be orthogonal
    - $\mathbf{D}$ is not orthogonal matrix
    - For unique solution column of $\mathbf{D}$ have unit norm
- Generate optimal code point $\mathbf{c}^*$ for each $\mathbf{x}$
  - Minimize distance between $\mathbf{x}$ and $g(\mathbf{c}^*)$
  - We use $L^2$ norm ie. $\mathbf{c}^* = arg_{\mathbf{c}} \min \|\mathbf{x} - g(\mathbf{c})\|_2$
  - We can switch to squared $L^2$ norm $\mathbf{c}^* = arg_{\mathbf{c}} \min \|\mathbf{x} - g(\mathbf{c})\|_2^2$

# Principal Component Analysis (contd.)

- We need to minimize

$$(\mathbf{x} - g(\mathbf{c}))^T(\mathbf{x} - g(\mathbf{c}))$$

# Principal Component Analysis (contd.)

- We need to minimize

$$
\begin{aligned}
& (\mathbf{x} - g(\mathbf{c}))^T (\mathbf{x} - g(\mathbf{c})) \\
\Rightarrow \quad & \mathbf{x}^T\mathbf{x} - \mathbf{x}^T g(\mathbf{c}) - g(\mathbf{c})\mathbf{x}^T + g(\mathbf{c})^T g(\mathbf{c})
\end{aligned}
$$

# Principal Component Analysis (contd.)

- We need to minimize

$$
\begin{aligned}
& (\mathbf{x} - g(\mathbf{c}))^T(\mathbf{x} - g(\mathbf{c})) \\
\Rightarrow \quad & \mathbf{x}^T\mathbf{x} - \mathbf{x}^T g(\mathbf{c}) - g(\mathbf{c})\mathbf{x}^T + g(\mathbf{c})^T g(\mathbf{c}) \\
\Rightarrow \quad & \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c})
\end{aligned}
$$

# Principal Component Analysis (contd.)

- We need to minimize

$$\begin{aligned} & (\mathbf{x} - g(\mathbf{c}))^T(\mathbf{x} - g(\mathbf{c})) \\ \Rightarrow \quad & \mathbf{x}^T\mathbf{x} - \mathbf{x}^T g(\mathbf{c}) - g(\mathbf{c})\mathbf{x}^T + g(\mathbf{c})^T g(\mathbf{c}) \\ \Rightarrow \quad & \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c}) \end{aligned}$$

- Therefore we have,

$$\mathbf{c}^* \quad = \quad arg_{\mathbf{c}} \min(\mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c}))$$

# Principal Component Analysis (contd.)

- We need to minimize

$$\begin{aligned} & (\mathbf{x} - g(\mathbf{c}))^T(\mathbf{x} - g(\mathbf{c})) \\ \Rightarrow \quad & \mathbf{x}^T\mathbf{x} - \mathbf{x}^T g(\mathbf{c}) - g(\mathbf{c})\mathbf{x}^T + g(\mathbf{c})^T g(\mathbf{c}) \\ \Rightarrow \quad & \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c}) \end{aligned}$$

- Therefore we have,

$$\begin{aligned} \mathbf{c}^* \quad = \quad & arg_{\mathbf{c}} \min(\mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c})) \\ \Rightarrow \quad & arg_{\mathbf{c}} \min(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{D}^T\mathbf{D}\mathbf{c}) \end{aligned}$$

# Principal Component Analysis (contd.)

- We need to minimize

$$\begin{aligned} & (\mathbf{x} - g(\mathbf{c}))^T(\mathbf{x} - g(\mathbf{c})) \\ \Rightarrow \quad & \mathbf{x}^T\mathbf{x} - \mathbf{x}^T g(\mathbf{c}) - g(\mathbf{c})\mathbf{x}^T + g(\mathbf{c})^T g(\mathbf{c}) \\ \Rightarrow \quad & \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c}) \end{aligned}$$

- Therefore we have,

$$\begin{aligned} \mathbf{c}^* \quad = \quad & arg_{\mathbf{c}} \min(\mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c})) \\ \Rightarrow \quad & arg_{\mathbf{c}} \min(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{D}^T\mathbf{D}\mathbf{c}) \\ \Rightarrow \quad & arg_{\mathbf{c}} \min(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{c}) \end{aligned}$$

# Principal Component Analysis (contd.)

- We need to minimize

$$\begin{aligned} & (\mathbf{x} - g(\mathbf{c}))^T(\mathbf{x} - g(\mathbf{c})) \\ \Rightarrow \quad & \mathbf{x}^T\mathbf{x} - \mathbf{x}^T g(\mathbf{c}) - g(\mathbf{c})\mathbf{x}^T + g(\mathbf{c})^T g(\mathbf{c}) \\ \Rightarrow \quad & \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c}) \end{aligned}$$

- Therefore we have,

$$\begin{aligned} \mathbf{c}^* \quad = \quad & arg_{\mathbf{c}} \min(\mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c})) \\ \Rightarrow \quad & arg_{\mathbf{c}} \min(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{D}^T\mathbf{D}\mathbf{c}) \\ \Rightarrow \quad & arg_{\mathbf{c}} \min(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{c}) \end{aligned}$$

- Optimization problem can be solved by differentiating

$$\nabla_c(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{c}) \quad = \quad 0$$

# Principal Component Analysis (contd.)

- We need to minimize

$$\begin{aligned} & (\mathbf{x} - g(\mathbf{c}))^T(\mathbf{x} - g(\mathbf{c})) \\ \Rightarrow \quad & \mathbf{x}^T\mathbf{x} - \mathbf{x}^T g(\mathbf{c}) - g(\mathbf{c})\mathbf{x}^T + g(\mathbf{c})^T g(\mathbf{c}) \\ \Rightarrow \quad & \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c}) \end{aligned}$$

- Therefore we have,

$$\begin{aligned} \mathbf{c}^* &= arg_{\mathbf{c}} \min(\mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c})) \\ &\Rightarrow arg_{\mathbf{c}} \min(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{D}^T\mathbf{D}\mathbf{c}) \\ &\Rightarrow arg_{\mathbf{c}} \min(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{c}) \end{aligned}$$

- Optimization problem can be solved by differentiating

$$\begin{aligned} \nabla_c(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{c}) &= 0 \\ -2\mathbf{D}^T\mathbf{x} + 2\mathbf{c} &= 0 \end{aligned}$$

# Principal Component Analysis (contd.)

- We need to minimize

$$\begin{aligned} & (\mathbf{x} - g(\mathbf{c}))^T(\mathbf{x} - g(\mathbf{c})) \\ \Rightarrow\ & \mathbf{x}^T\mathbf{x} - \mathbf{x}^T g(\mathbf{c}) - g(\mathbf{c})\mathbf{x}^T + g(\mathbf{c})^T g(\mathbf{c}) \\ \Rightarrow\ & \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c}) \end{aligned}$$

- Therefore we have,

$$\begin{aligned} \mathbf{c}^* &= arg_{\mathbf{c}} \min(\mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T g(\mathbf{c}) + g(\mathbf{c})^T g(\mathbf{c})) \\ &\Rightarrow arg_{\mathbf{c}} \min(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{D}^T\mathbf{D}\mathbf{c}) \\ &\Rightarrow arg_{\mathbf{c}} \min(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{c}) \end{aligned}$$

- Optimization problem can be solved by differentiating

$$\begin{aligned} \nabla_c(-2\mathbf{x}^T\mathbf{D}\mathbf{c} + \mathbf{c}^T\mathbf{c}) &= 0 \\ -2\mathbf{D}^T\mathbf{x} + 2\mathbf{c} &= 0 \\ \mathbf{c} &= \mathbf{D}^T\mathbf{x} \end{aligned}$$

# Principal Component Analysis (contd.)

- Optimal encoding can be done using matrix-vector multiplication $f(\mathbf{x}) = \mathbf{D}^T\mathbf{x}$

# Principal Component Analysis (contd.)

- Optimal encoding can be done using matrix-vector multiplication $f(\mathbf{x}) = \mathbf{D}^T\mathbf{x}$
- PCA reconstruction $r(\mathbf{x}) = g(f(\mathbf{x})) = \mathbf{D}\mathbf{D}^T\mathbf{x}$

# Principal Component Analysis (contd.)

- Optimal encoding can be done using matrix-vector multiplication $f(\mathbf{x}) = \mathbf{D}^T\mathbf{x}$
- PCA reconstruction $r(\mathbf{x}) = g(f(\mathbf{x})) = \mathbf{D}\mathbf{D}^T\mathbf{x}$
- $\mathbf{D}$ can be determined by minimizing distance between inputs and reconstruction ie.

$$\mathbf{D}^* = \arg_{\mathbf{D}} \min \sqrt{\sum_{i,j} \left( x_j^{(i)} - r(x^{(i)})_j \right)^2} \text{ subject to } \mathbf{D}^T\mathbf{D} = \mathbf{I}_l$$

# Principal Component Analysis (contd.)

- Optimal encoding can be done using matrix-vector multiplication $f(\mathbf{x}) = \mathbf{D}^T\mathbf{x}$
- PCA reconstruction $r(\mathbf{x}) = g(f(\mathbf{x})) = \mathbf{D}\mathbf{D}^T\mathbf{x}$
- $\mathbf{D}$ can be determined by minimizing distance between inputs and reconstruction ie.

$$\mathbf{D}^* = \arg_{\mathbf{D}} \min \sqrt{\sum_{i,j} \left( x_j^{(i)} - r(x^{(i)})_j \right)^2} \text{ subject to } \mathbf{D}^T\mathbf{D} = \mathbf{I}_l$$

- To derive $\mathbf{D}^*$, we start by considering $l = 1$
  - $\mathbf{D}$ becomes $\mathbf{d}$

# Principal Component Analysis (contd.)

- Optimal encoding can be done using matrix-vector multiplication $f(\mathbf{x}) = \mathbf{D}^T\mathbf{x}$
- PCA reconstruction $r(\mathbf{x}) = g(f(\mathbf{x})) = \mathbf{D}\mathbf{D}^T\mathbf{x}$
- $\mathbf{D}$ can be determined by minimizing distance between inputs and reconstruction ie.

$$\mathbf{D}^* = \arg_{\mathbf{D}} \min \sqrt{\sum_{i,j} \left( x_j^{(i)} - r(x^{(i)})_j \right)^2} \text{ subject to } \mathbf{D}^T\mathbf{D} = \mathbf{I}_l$$

- To derive $\mathbf{D}^*$, we start by considering $l = 1$
  - $\mathbf{D}$ becomes $\mathbf{d}$
- Simplifying based on the assumptions

$$\mathbf{d}^* = \arg_{\mathbf{d}} \min \sum_i \|\mathbf{x}^{(i)} - \mathbf{d}\mathbf{d}^T\mathbf{x}^{(i)}\|_2^2 \text{ subject to } \|\mathbf{d}\|_2 = 1$$

# Principal Component Analysis (contd.)

- Optimal encoding can be done using matrix-vector multiplication $f(\mathbf{x}) = \mathbf{D}^T\mathbf{x}$
- PCA reconstruction $r(\mathbf{x}) = g(f(\mathbf{x})) = \mathbf{D}\mathbf{D}^T\mathbf{x}$
- $\mathbf{D}$ can be determined by minimizing distance between inputs and reconstruction ie.

$$\mathbf{D}^* = \arg_{\mathbf{D}} \min \sqrt{\sum_{i,j} \left( x_j^{(i)} - r(x^{(i)})_j \right)^2} \text{ subject to } \mathbf{D}^T\mathbf{D} = \mathbf{I}_l$$

- To derive $\mathbf{D}^*$, we start by considering $l = 1$
  - $\mathbf{D}$ becomes $\mathbf{d}$
- Simplifying based on the assumptions

$$\mathbf{d}^* = \arg_{\mathbf{d}} \min \sum_i \|\mathbf{x}^{(i)} - \mathbf{d}\mathbf{d}^T\mathbf{x}^{(i)}\|_2^2 \text{ subject to } \|\mathbf{d}\|_2 = 1$$

  - $\mathbf{d}^T\mathbf{x}^{(i)}$ is scalar, hence $\mathbf{d}\mathbf{d}^T\mathbf{x}^{(i)} = \mathbf{d}^T\mathbf{x}^{(i)}\mathbf{d} = \mathbf{x}^{(i)T}\mathbf{d}\mathbf{d}$

# Principal Component Analysis (contd.)

- We get

$$\mathbf{d}^* = \arg_{\mathbf{d}} \min \sum_i \|\mathbf{x}^{(i)} - \mathbf{x}^{(i)T}\mathbf{d}\mathbf{d}\|_2^2 \text{ subject to } \|\mathbf{d}\|_2 = 1$$

# Principal Component Analysis (contd.)

- We get

$$\mathbf{d}^* = \arg_{\mathbf{d}} \min \sum_i \|\mathbf{x}^{(i)} - \mathbf{x}^{(i)T}\mathbf{d}\mathbf{d}\|_2^2 \text{ subject to } \|\mathbf{d}\|_2 = 1$$

- Rewriting in terms of single design matrix

$$\mathbf{d}^* = \arg_{\mathbf{d}} \min \|\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T\|_2^2 \text{ subject to } \|\mathbf{d}\|_2 = 1$$

# Principal Component Analysis (contd.)

- We get

$$\mathbf{d}^* = \arg_{\mathbf{d}} \min \sum_i \|\mathbf{x}^{(i)} - \mathbf{x}^{(i)T}\mathbf{d}\mathbf{d}\|_2^2 \text{ subject to } \|\mathbf{d}\|_2 = 1$$

- Rewriting in terms of single design matrix

$$\mathbf{d}^* = \arg_{\mathbf{d}} \min \|\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T\|_2^2 \text{ subject to } \|\mathbf{d}\|_2 = 1$$

- Now we have,

$$\arg_{\mathbf{d}} \min \|\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T\|_F^2 \text{ subject to} \|\mathbf{d}\mathbf{d}^T\| = 1$$

# Principal Component Analysis (contd.)

- Simplifying,

$$\arg_{\mathbf{d}} \min \|\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T\|_F^2$$

# Principal Component Analysis (contd.)

- Simplifying,

$$\begin{aligned} & \arg_{\mathbf{d}} \min \|\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T\|_F^2 \\ = \; & \arg_{\mathbf{d}} \min Tr\left((\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)\right) \end{aligned}$$

# Principal Component Analysis (contd.)

- Simplifying,

$$
\begin{aligned}
& \arg_{\mathbf{d}} \min \|\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T\|_F^2 \\
= \; & \arg_{\mathbf{d}} \min Tr\left((\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)\right) \\
= \; & \arg_{\mathbf{d}} \min Tr(\mathbf{X}^T\mathbf{X} - \mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T - \mathbf{d}\mathbf{d}^T\mathbf{X}^T\mathbf{X} + \mathbf{d}\mathbf{d}^T\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T)
\end{aligned}
$$

# Principal Component Analysis (contd.)

- Simplifying,

$$
\begin{aligned}
& \arg_{\mathbf{d}} \min \|\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T\|_F^2 \\
= \; & \arg_{\mathbf{d}} \min Tr\left((\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)\right) \\
= \; & \arg_{\mathbf{d}} \min Tr(\mathbf{X}^T\mathbf{X} - \mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T - \mathbf{d}\mathbf{d}^T\mathbf{X}^T\mathbf{X} + \mathbf{d}\mathbf{d}^T\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T) \\
= \; & \arg_{\mathbf{d}} \min -2\, Tr(\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T) + Tr(\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T\mathbf{d}\mathbf{d}^T)
\end{aligned}
$$

# Principal Component Analysis (contd.)

- Simplifying,

$$
\begin{aligned}
& \arg_{\mathbf{d}} \min \|\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T\|_F^2 \\
= \; & \arg_{\mathbf{d}} \min Tr\left((\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)\right) \\
= \; & \arg_{\mathbf{d}} \min Tr(\mathbf{X}^T\mathbf{X} - \mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T - \mathbf{d}\mathbf{d}^T\mathbf{X}^T\mathbf{X} + \mathbf{d}\mathbf{d}^T\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T) \\
= \; & \arg_{\mathbf{d}} \min -2\, Tr(\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T) + Tr(\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T\mathbf{d}\mathbf{d}^T) \\
= \; & \arg_{\mathbf{d}} \min -Tr(\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T)
\end{aligned}
$$

# Principal Component Analysis (contd.)

- Simplifying,

$$
\begin{aligned}
& \arg_{\mathbf{d}} \min \|\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T\|_F^2 \\
= \; & \arg_{\mathbf{d}} \min Tr\left((\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)\right) \\
= \; & \arg_{\mathbf{d}} \min Tr(\mathbf{X}^T\mathbf{X} - \mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T - \mathbf{d}\mathbf{d}^T\mathbf{X}^T\mathbf{X} + \mathbf{d}\mathbf{d}^T\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T) \\
= \; & \arg_{\mathbf{d}} \min -2\, Tr(\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T) + Tr(\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T\mathbf{d}\mathbf{d}^T) \\
= \; & \arg_{\mathbf{d}} \min -Tr(\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T) \\
= \; & \arg_{\mathbf{d}} \min -Tr(\mathbf{d}^T\mathbf{X}^T\mathbf{X}\mathbf{d})
\end{aligned}
$$

# Principal Component Analysis (contd.)

- Simplifying,

$$
\begin{aligned}
& \arg_{\mathbf{d}} \min \|\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T\|_F^2 \\
= \; & \arg_{\mathbf{d}} \min Tr\left((\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{d}\mathbf{d}^T)\right) \\
= \; & \arg_{\mathbf{d}} \min Tr(\mathbf{X}^T\mathbf{X} - \mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T - \mathbf{d}\mathbf{d}^T\mathbf{X}^T\mathbf{X} + \mathbf{d}\mathbf{d}^T\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T) \\
= \; & \arg_{\mathbf{d}} \min -2\, Tr(\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T) + Tr(\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T\mathbf{d}\mathbf{d}^T) \\
= \; & \arg_{\mathbf{d}} \min -Tr(\mathbf{X}^T\mathbf{X}\mathbf{d}\mathbf{d}^T) \\
= \; & \arg_{\mathbf{d}} \min -Tr(\mathbf{d}^T\mathbf{X}^T\mathbf{X}\mathbf{d})
\end{aligned}
$$

- Optimization problem can be solved by eigen decomposition

# Overview of Probability

# Probability

- Mathematical framework for representing uncertain statements
- Possible source of uncertainty
  - Inherent stochasticity
  - Incomplete observability
  - Incomplete model
- Two broad categories
  - Frequentist
    - Getting a card
  - Bayesian
    - Chance of having flu

# Random Variable

- Variable that can take different values randomly
- Example
  - $X$ is random variable that can take value $x_1$ or $x_2$

# Probability Distribution

- How likely a random variable is to take on each of its possible states

# Probability Distribution

- How likely a random variable is to take on each of its possible states
- Probability Mass Function
  - Discrete
  - Maps state to probability of taking that state

# Probability Distribution

- How likely a random variable is to take on each of its possible states
- Probability Mass Function
  - Discrete
  - Maps state to probability of taking that state
- Joint probability distribution
  - $P(X = x, Y = y)$ — probability of $X$ taking value $x$ and $Y$ taking value $y$

# Probability Distribution

- How likely a random variable is to take on each of its possible states
- Probability Mass Function
  - Discrete
  - Maps state to probability of taking that state
- Joint probability distribution
  - $P(X = x, Y = y)$ — probability of $X$ taking value $x$ and $Y$ taking value $y$
- Probability function $P$ on $X$
  - The domain of $P$ is set of all possible state of $X$
  - $\forall x \in X \; 0 \leq P(x) \leq 1$
  - $\sum_{x \in X} P(x) = 1$
- Example
  - Uniform distribution with $k$ different states $1/k$

# Probability Density Function

- Continuous variable ($p$)

# Probability Density Function

- Continuous variable ($p$)
  - The domain of $p$ is set of all possible state of $X$
  - $\forall x \in X \quad p(x) \geq 0$
  - $\int_{x \in X} p(x)dx = 1$

# Probability Density Function

- Continuous variable ($p$)
  - The domain of $p$ is set of all possible state of $X$
  - $\forall x \in X \quad p(x) \geq 0$
  - $\int_{x \in X} p(x)dx = 1$
- Example
  - Uniform distribution in $[a, b]$ is represented as $X \sim U(a, b)$

# Marginal Probability

- Probability distribution over the subset
- Let $X, Y$ be random variables and $P(x, y)$ is known
  - $P(x) = \sum_{y \in Y} P(x, y)$
  - $p(x) = \int_{y \in Y} p(x, y) dy$

# Conditional Probability

- Probability of some event given that some other event has happened

$$P(Y = y|X = x) = \frac{P(Y = y, X = x)}{P(X = x)}$$

# Conditional Probability

- Probability of some event given that some other event has happened

$$P(Y = y|X = x) = \frac{P(Y = y, X = x)}{P(X = x)}$$

- Chain rule

$$P(x_1 x_2 \ldots x_n) = P(x_1) \prod_{i=2}^{n} P(x_i|x_1 \ldots x_{i-1})$$

  - $P(a, b, c) = P(a|b, c)P(b|c)P(c)$

# Conditional Probability

- Probability of some event given that some other event has happened

$$P(Y = y|X = x) = \frac{P(Y = y, X = x)}{P(X = x)}$$

- Chain rule

$$P(x_1 x_2 \dots x_n) = P(x_1) \prod_{i=2}^{n} P(x_i|x_1 \dots x_{i-1})$$

  - $P(a, b, c) = P(a|b, c)P(b|c)P(c)$
- Independence of random variable

# Conditional Probability

- Probability of some event given that some other event has happened

$$P(Y = y|X = x) = \frac{P(Y = y, X = x)}{P(X = x)}$$

- Chain rule

$$P(x_1 x_2 \ldots x_n) = P(x_1) \prod_{i=2}^{n} P(x_i|x_1 \ldots x_{i-1})$$

  - $P(a, b, c) = P(a|b, c)P(b|c)P(c)$
- Independence of random variable
  - $\forall x \in X, y \in Y \quad p(X = x, Y = y) = p(X = x)p(Y = y)$
  - Conditional independence
    - $p(x, y|z) = p(x|z)p(y|z)$

# Expectation

- Expected value of some function with respect to probability distribution $P(x)$

# Expectation

- Expected value of some function with respect to probability distribution $P(x)$
  - $\mathbb{E}_{X \sim P}[f(x)] = \sum_x P(x) f(x)$

# Expectation

- Expected value of some function with respect to probability distribution $P(x)$
  - $\mathbb{E}_{X \sim P}[f(x)] = \sum_x P(x) f(x)$
  - $\mathbb{E}_{X \sim p}[f(x)] = \int_x P(x) f(x) dx$

# Expectation

- Expected value of some function with respect to probability distribution $P(x)$
  - $\mathbb{E}_{X\sim P}[f(x)] = \sum_x P(x)f(x)$
  - $\mathbb{E}_{X\sim p}[f(x)] = \int_x P(x)f(x)dx$
  - $\mathbb{E}_X[\alpha f(x) + \beta g(x)] = \alpha\mathbb{E}_X[f(x)] + \beta\mathbb{E}_X[g(x)]$

# Variance & Covariance

- How much the values of a given function vary as we sample different values of $x$ from its probability distribution

# Variance & Covariance

- How much the values of a given function vary as we sample different values of $x$ from its probability distribution
  - $\text{Var}(f(x)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$

# Variance & Covariance

- How much the values of a given function vary as we sample different values of $x$ from its probability distribution
  - $\text{Var}(f(x)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$
- How much two values are linearly related

# Variance & Covariance

- How much the values of a given function vary as we sample different values of $x$ from its probability distribution
  - $\text{Var}(f(x)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$
- How much two values are linearly related
  - $Cov(f(x), g(y)) = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])(g(y) - \mathbb{E}[g(y)])\right]$

# Variance & Covariance

- How much the values of a given function vary as we sample different values of $x$ from its probability distribution
  - $\text{Var}(f(x)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$
- How much two values are linearly related
  - $Cov(f(x), g(y)) = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])(g(y) - \mathbb{E}[g(y)])\right]$
    - It can be positive or negative

# Bayes' rule

- Suppose $P(y|x)$ and $P(x)$ known and need to find out $P(x|y)$

$$P(x|y) = \frac{P(x)P(y|x)}{P(y)}$$

# Bayes' rule

- Suppose $P(y|x)$ and $P(x)$ known and need to find out $P(x|y)$

$$P(x|y) = \frac{P(x)P(y|x)}{P(y)}$$

- Now $P(y)$ can be found out from

$$P(y) = \sum_{x} P(y|x)P(x)$$

# Problems

- Suppose that we have two bags each containing black and white balls. One bag contains three times as many white balls as blacks. The other bag contains three times as many black balls as white. Suppose we choose one of these bags at random. For this bag we select five balls at random, replacing each ball after it has been selected. The result is that we find 4 white balls and one black. What is the probability that we were using the bag with mainly white balls?

- Given the following statistics, what is the probability that a person has cancer if he has a positive pathological result?
  - One percent of people over 50 have this cancer.
  - Ninety percent of people who have this cancer test positive on pathological report.
  - Eight percent of people will have false positives.

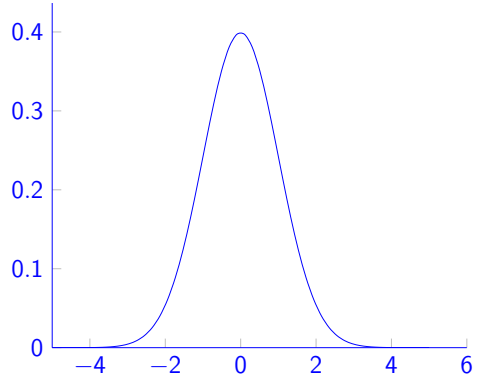
# Information theory

- Quantifying how much information is present in a signal
  - The sun rises in the east. — uninformative
  - There was a solar eclipse this morning. — informative
- Therefore, we would like quantify
  - Likely event should have low information content
    - Events guaranteed to happen should have no information
  - Less likely event should have higher information content
  - Independent event should have additive information
- Information of an event $X = x$ be $I(x) = -\log P(x)$
  - Natural logarithm, with base $e$
  - Unit of $I(x)$ is *nat*

# Gaussian distribution

- Also, known as Normal Distribution

$$\mathcal{N}(x;\mu,\sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$



# Multivariate normal distribution

- Defined as

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sqrt{\frac{1}{(2\pi)^n \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \mu)\right)$$

- $\boldsymbol{\mu}$ — mean of the distribution
- $\boldsymbol{\Sigma}$ — Covariance matrix of the distribution

# References


- "Introduction to Linear Algebra" by Gilbert Strang
- "Probability Theory: The Logic of Science" by Jaynes, E. T. (2003). Cambridge University Press.