Mathematics for Deep Learning

Linear Algebra, Calculus and Probability Theory

Kumar Bipin

BE, MS, PhD (MMMTU, IISc, IIIT-Hyderabad)

Robotics, Computer Vision, Deep Learning, Machine Learning, System Software

Contents

Disclaimer

- This is not a math lecture!
- No axioms, no claims, no theorems, no proofs
- The goal is to repeat the necessary minimal math background
  to follow our deep learning, computer vision and self-driving lectures
- Enjoy!

Sets, Scalars, Vectors, Matrices and Tensors

Kumar Bipin

Sets

- A **set** $\mathcal{S}$ is the mathematical model for a collection of different things.
- Examples:
  - $\emptyset$: Empty set
  - $\{\text{red}, \text{green}, \text{blue}\}$: Set of colors
  - $\{0, 1\}$: Set of binary numbers
  - $\mathbb{R}$: Set of real numbers (e.g., $1.234 \in \mathbb{R}$)
  - $\mathbb{N}_0 = \{0, 1, 2, \dots\}$: Set of natural numbers including 0
  - $\{\mathbb{R}, \mathbb{N}_0\}$: Set of two sets
- The **cardinality** of a set $\mathcal{S}$, denoted $|\mathcal{S}|$, is the number of members of $\mathcal{S}$
- We can take the **union** of two sets $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$
- We can take the **intersection** of two sets $\mathcal{S} = \mathcal{S}_1 \cap \mathcal{S}_2$
- We can take the **cartesian product** of two sets $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2$ or $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$

Scalars

- A **scalar** is a single number (in contrast to other objects in linear algebra)
- We write scalars in lower case, non-bold typeface font
- Examples:
  - $x \in \mathbb{R}$
  - $c \in \mathbb{N}_0$
- When introducing them, we specify their type

Vectors

- A **vector** is an array of numbers with specific order
- We write vectors in lower case, bold typeface font
- Example:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

- We can identify each element of the vector (scalar) via its index $(x_1, x_2, \dots)$
- If each $x_i \in \mathbb{R}$, then $\mathbf{x} \in \mathbb{R}^3 = \mathbb{R} \times \mathbb{R} \times \mathbb{R}$ ($\mathbf{x}$ is in the Cartesian product of $\mathbb{R}$)
- We can think of vectors as identifying points in space (element = coordinate)
- We can also index a subset of elements of a vector: $\mathbf{x}_{\mathcal{S}}$ with $\mathcal{S} = \{1, 3\}$

Matrices

- A **matrix** is a 2D array of numbers, each element is identified by two indices
- We write matrices in upper case, bold typeface font
- Example:

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$$

- If a real-valued matrix has M rows and N columns we write $\mathbf{A} \in \mathbb{R}^{M \times N}$
- $M \times N$ is the shape of the matrix
- We can identify each element of a matrix via its two indices $(a_{1,1}, a_{1,2}, \dots)$
  - For example, $a_{i,j}$ is the element of $\mathbf{A}$ in the i'th row and j'th column
  - Sometimes the ',' separating indices is omitted in the notation
- Columns and rows of matrices can be accessed via $\mathbf{a}_j = \mathbf{A}_{:,j}$ and $\mathbf{a}_i^T = \mathbf{A}_{i,:}$

Tensors

- A **tensor** is an array with more than 2 axes (e.g.: RGB image)
- We write tensors in upper case, bold typeface font
- Example for tensor of shape $M \times N \times K$:

$$\mathbf{A} \in \mathbb{R}^{M \times N \times K}$$

- We identify each element of a tensor via its indices $(a_{i,j,k})$
  - Elements of a tensor are scalars and written in lower case non-boldface font

Transpose

- The **transpose** $\mathbf{A}^T$ of a matrix $\mathbf{A}$ mirrors it at its main diagonal:

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{pmatrix} \quad \Rightarrow \quad \mathbf{A}^T = \begin{pmatrix} a_{1,1} & a_{2,1} & a_{3,1} \\ a_{1,2} & a_{2,2} & a_{3,2} \end{pmatrix}$$

- Similarly, a standard column vector can be transposed into a row vector:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix}^T}_{\text{better for inline math}} \quad \Rightarrow \quad \mathbf{x}^T = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix}$$

- For scalars, we have $x^T = x$

Examples with Numbers

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \in \mathbb{R}^{2\times 2} \quad \Rightarrow \quad \mathbf{A}^T = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} \in \mathbb{R}^{2\times 2}$$

$$a_{1,1} = 1 \in \mathbb{R} \qquad a_{1,2} = 2 \in \mathbb{R} \qquad a_{2,1} = 3 \in \mathbb{R} \qquad a_{2,2} = 4 \in \mathbb{R}$$

$$\mathbf{A}_{:,1} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \in \mathbb{R}^{2\times 1} \quad \mathbf{A}_{1,:} = \begin{pmatrix} 1 & 2 \end{pmatrix} \in \mathbb{R}^{1\times 2} \quad \mathbf{A}_{:,2} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} \in \mathbb{R}^{2\times 1} \quad \mathbf{A}^T_{2,:} = \begin{pmatrix} 2 & 4 \end{pmatrix} \in \mathbb{R}^{1\times 2}$$

- The last row can be considered either matrices ($\mathbb{R}^{1\times 2}$ / $\mathbb{R}^{2\times 1}$) or vectors $\mathbb{R}^2$
- If they are considered vectors, we write either $\mathbf{a}$ or $\mathbf{a}^T$ (with $\mathbf{a} \in \mathbb{R}^2$) to distinguish row from column vectors which would otherwise not be clear

Adding and Multiplying Matrices and Vectors

Kumar Bipin

Adding and Subtracting Matrices and Vectors

- We **add/subtract** vectors or matrices by adding/subtracting them elementwise:

$$c_i = a_i + b_i \qquad c_{i,j} = a_{i,j} + b_{i,j}$$

- Example (subtraction analogously):

$$\mathbf{a} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 4 \\ 2 \end{pmatrix} \quad \Rightarrow \quad \mathbf{c} = \mathbf{a} + \mathbf{b} = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix} \quad \Rightarrow \quad \mathbf{C} = \mathbf{A} + \mathbf{B} = \begin{pmatrix} 5 & 5 \\ 5 & 5 \end{pmatrix}$$

- Note that the vectors or matrices must have the same shape

Adding and Subtracting Matrices and Vectors

- We can also **add a scalar to a matrix** or **multiply a matrix by a scalar:**

$$\mathbf{D} = a\,\mathbf{B} + c$$

- This corresponds to performing the operation on each element:

$$d_{i,j} = a\,b_{i,j} + c$$

- In deep learning, we sometimes also allow the addition of a matrix and a vector:

$$\mathbf{C} = \mathbf{A} + \mathbf{b} \quad \text{where} \quad c_{i,j} = a_{i,j} + b_i$$

  - In this example, the vector $\mathbf{b}$ is added to each column of matrix $\mathbf{A}$
  - This implicit copying of $\mathbf{b}$ to many locations is called **broadcasting**

Adding and Subtracting Matrices and Vectors

- Example for **scalar addition and multiplication:**

$$2\begin{pmatrix}1 & 2\\ 2 & 0\end{pmatrix} + 1 = \begin{pmatrix}3 & 5\\ 5 & 1\end{pmatrix}$$

- Example for **broadcasting** (shape of vector determines which type):

$$\begin{pmatrix}1 & 1\\ 2 & 0\end{pmatrix} + \begin{pmatrix}1\\ 3\end{pmatrix} = \begin{pmatrix}2 & 2\\ 5 & 3\end{pmatrix} \qquad \begin{pmatrix}1 & 1\\ 2 & 0\end{pmatrix} + \begin{pmatrix}1 & 3\end{pmatrix} = \begin{pmatrix}2 & 4\\ 3 & 3\end{pmatrix}$$

- NumPy supports broadcasting natively:
  `https://numpy.org/doc/stable/user/basics.broadcasting.html`

Multiplying Matrices and Vectors

- Two vectors or matrices $\mathbf{A}$ and $\mathbf{B}$ can be multiplied if $\mathbf{A}$ has the same number of columns as $\mathbf{B}$ has rows (i.e., $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\mathbf{B} \in \mathbb{R}^{N \times L}$):

$$\mathbf{C} = \mathbf{A}\,\mathbf{B}$$

- The **matrix product** is defined by

$$c_{i,j} = \sum_k a_{i,k}\, b_{k,j}$$

- Note that the matrix product is not a matrix containing the product of the individual elements which is called **Hadamard product:** $\mathbf{A} \odot \mathbf{B}$ $(\mathbf{A}, \mathbf{B} \in \mathbb{R}^{M \times N})$

Multiplying Matrices and Vectors

- Example for a **matrix product:**

$$\mathbf{A} = \begin{pmatrix} 3 & 1 \\ 2 & 1 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix} \quad \Rightarrow \quad \mathbf{A}\,\mathbf{B} = \begin{pmatrix} 6 & 7 \\ 5 & 5 \end{pmatrix}$$

- Example for an **inner product between two vectors:** (= “dot product”)

$$\mathbf{a} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \quad \Rightarrow \quad \mathbf{a}^T\mathbf{b} = 9$$

- Example for an **outer product between two vectors:** (= rank 1 matrix)

$$\mathbf{a} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \quad \Rightarrow \quad \mathbf{a}\,\mathbf{b}^T = \begin{pmatrix} 3 & 9 \\ 2 & 6 \end{pmatrix}$$

Multiplying Matrices and Vectors

- Example for **broadcasting** with the help of the outer product:

$$\begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 3 \end{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 3 & 3 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 5 & 3 \end{pmatrix}$$

Useful Properties

- Matrix multiplication is **distributive:**

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

- Matrix multiplication is **associative:**

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

- However, in general matrix multiplication is **not commutative**:

$$\mathbf{AB} \neq \mathbf{BA}$$

Useful Properties

- The **transpose** of a matrix product has a simple form:

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$$

- Hence, the vector dot product is commutative:

$$\mathbf{x}^T\mathbf{y} = \left(\mathbf{x}^T\mathbf{y}\right)^T = \mathbf{y}^T\mathbf{x} = \mathbf{x}^T\mathbf{y}$$

- Matrix-vector products allow to compactly write **systems of linear equations:**

$$\mathbf{a}_1^T\mathbf{x} = b_1; \quad \mathbf{a}_2^T\mathbf{x} = b_2; \quad \ldots \quad \mathbf{a}_M^T\mathbf{x} = b_M; \quad \text{as} \quad \mathbf{Ax} = \mathbf{b}$$

... where $\mathbf{a}_i^T = \mathbf{A}_{i,:}$ denotes the i'th row of matrix $\mathbf{A}$.

Identity and Inverse Matrices

Kumar Bipin

Identity and Inverse Matrices

- A matrix that does not change a vector multiplied with it is called **identity matrix**
- We denote the identity matrix preserving N-dimensional vectors as $\mathbf{I}_N \in \mathbb{R}^{N \times N}$

$$\forall_{\mathbf{x} \in \mathbb{R}^N} : \mathbf{I}_N \, \mathbf{x} = \mathbf{x}$$

- Example:

$$\mathbf{I}_3 = \text{diag}(1, 1, 1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

- $\mathbf{I}_N$ is a square matrix with all diagonal elements equal to one and others zero

Identity and Inverse Matrices

- The **matrix inverse** $\mathbf{A}^{-1}$ of square matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ is defined by

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_N$$

- This allows for **solving linear systems** of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$:

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

- Remark: This is only possible if $\mathbf{A}^{-1}$ exists (has full rank, will discuss later)
- It is not advisable to numerically solve linear systems like this (precision)

Identity and Inverse Matrices

- Example:

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix} \quad \Rightarrow \quad \mathbf{A}^{-1} = \begin{pmatrix} 0 & 0.5 \\ 1 & -0.5 \end{pmatrix}$$

$$\mathbf{A}^{-1}\mathbf{A} = \begin{pmatrix} 0 & 0.5 \\ 1 & -0.5 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathbf{I}_2$$

$$\mathbf{b} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{b} = \begin{pmatrix} 0 & 0.5 \\ 1 & -0.5 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 1.5 \end{pmatrix}$$

Linear Dependence and Span

Kumar Bipin

Linear Dependence and Span

- For $\mathbf{A}^{-1}$ to exist, $\mathbf{Ax} = \mathbf{b}$ must have exactly one solution for every value of $\mathbf{b}$
- It is possible for $\mathbf{Ax} = \mathbf{b}$ to have 0, 1 or infinitely many solutions
- If both $\mathbf{x}$ and $\mathbf{y}$ are solutions then

$$\mathbf{z} = \alpha \mathbf{x} + (1 - \alpha)\mathbf{y}$$

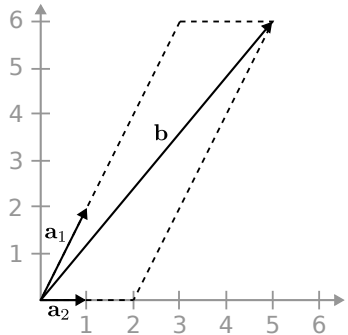
  is also a solution for any $\alpha \in \mathbb{R}$
- We call $\mathbf{Ax} = \sum_i x_i \mathbf{A}_{:,i}$ a linear combination (sum of scalar-vector products)
- Example:

$$\underbrace{\begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} 3 \\ 2 \end{pmatrix}}_{\mathbf{x}} = \underbrace{3}_{x_1} \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \underbrace{2}_{x_2} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 6 \end{pmatrix} + \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

Linear Dependence and Span

- The **span** of a set of vectors is the set of all points obtained by linear combination:

$$\underbrace{\begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} 3 \\ 2 \end{pmatrix}}_{\mathbf{x}} = \underbrace{3}_{x_1} \underbrace{\begin{pmatrix} 1 \\ 2 \end{pmatrix}}_{\mathbf{a}_1} + \underbrace{2}_{x_2} \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{\mathbf{a}_2} = \begin{pmatrix} 3 \\ 6 \end{pmatrix} + \begin{pmatrix} 2 \\ 0 \end{pmatrix} = \underbrace{\begin{pmatrix} 5 \\ 6 \end{pmatrix}}_{\mathbf{b}}$$

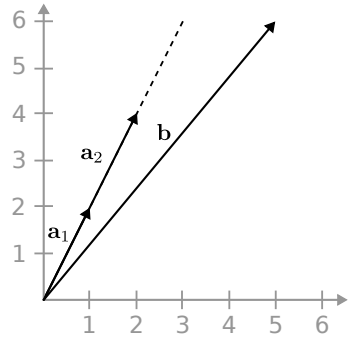


- $\mathbf{Ax} = \mathbf{b}$ has solution $\Leftrightarrow$ $\mathbf{b}$ is in the span of columns of $\mathbf{A}$
- This particular span is known as **column space** or **range**
- When is there no solution?

Linear Dependence and Span

- Consider another example:

$$\underbrace{\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\mathbf{x}} = x_1 \underbrace{\begin{pmatrix} 1 \\ 2 \end{pmatrix}}_{\mathbf{a}_1} + x_2 \underbrace{\begin{pmatrix} 2 \\ 4 \end{pmatrix}}_{\mathbf{a}_2} = \underbrace{\begin{pmatrix} 5 \\ 6 \end{pmatrix}}_{\mathbf{b}}$$

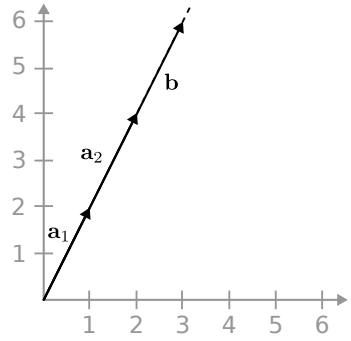


- $\mathbf{a}_2$ is a multiple of $\mathbf{a}_2$
- The columns are **linearly dependent**
- The column space is a line
- $\mathbf{b}$ is **not in the span** of columns of $\mathbf{A}$ $\Rightarrow \mathbf{Ax} = \mathbf{b}$ has no solution
- When are there ∞ many solutions?

Linear Dependence and Span

- Consider another example:

$$\underbrace{\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\mathbf{x}} = x_1 \underbrace{\begin{pmatrix} 1 \\ 2 \end{pmatrix}}_{\mathbf{a}_1} + x_2 \underbrace{\begin{pmatrix} 2 \\ 4 \end{pmatrix}}_{\mathbf{a}_2} = \underbrace{\begin{pmatrix} 3 \\ 6 \end{pmatrix}}_{\mathbf{b}}$$



- $\mathbf{a}_2$ is a multiple of $\mathbf{a}_1$
- The columns are **linearly dependent**
- The column space is a line
- $\mathbf{b}$ is **in the span** of columns of $\mathbf{A}$
  $\Rightarrow \mathbf{Ax} = \mathbf{b}$ has ∞ many solutions

Linear Dependence and Span

- In order for $\mathbf{Ax} = \mathbf{b}$ to have a solution for all values of $\mathbf{b} \in \mathbb{R}^N$, the column space of $\mathbf{A}$ must encompass all of $\mathbb{R}^N$
- For the column space of a matrix with N rows to **encompass** all of $\mathbb{R}^N$, the matrix must contain at least one set of N linearly independent columns
- For the matrix to have an **inverse,** we additionally need to ensure that $\mathbf{Ax} = \mathbf{b}$ has at most one solution $\mathbf{x}$ for each value of $\mathbf{b}$
- Hence, the matrix must have exactly N **columns** (square matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$) and all columns must be **linearly independent**
- The **rank** of a matrix refers to the number of linearly independent rows or columns
- A square matrix with any two linearly dependent columns is called **singular**
- Every matrix with **full rank** (= every **non-singular** matrix) can be inverted

Vector and Matrix Norms

Kumar Bipin

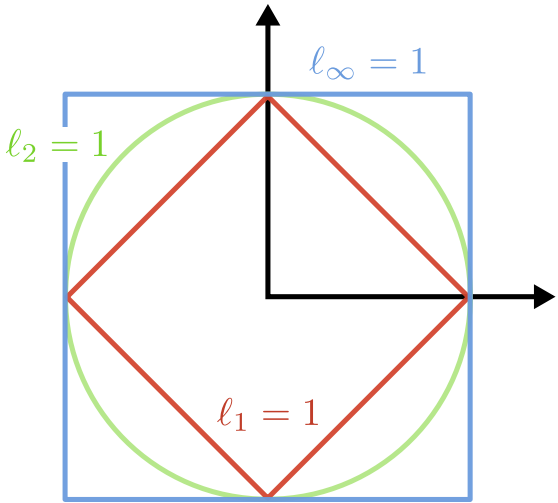
Vector and Matrix Norms

- We can measure the "size" of a vector using a function called **norm**
- The ℓ_p-norm is defined as

$$\|\mathbf{x}\|_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}}$$

- A norm maps vectors to non-negative values (= distance to the origin)
- Mathematically, a norm $f(\cdot)$ satisfies:
  - $f(\mathbf{x}) = 0 \Rightarrow \mathbf{x} = \mathbf{0}$
  - $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$ (triangle inequality)
  - $\forall_\alpha \in \mathbb{R} : f(\alpha \mathbf{x}) = |\alpha| f(\mathbf{x})$

Vector and Matrix Norms



$$\|\mathbf{x}\|_1 = \sum_i |x_i| = |x_1| + |x_2|$$

$$\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2} = \sqrt{\mathbf{x}^T\mathbf{x}} = \sqrt{x_1^2 + x_2^2}$$

$$\|\mathbf{x}\|_\infty = \max_i |x_i| = \max(|x_1|, |x_2|)$$

- ℓ_2 is called the **Euclidean norm**
  (= Euclidean distance to origin)
- ℓ_∞ is called the **max/infinity norm**

Vector and Matrix Norms

- The **dot product** of two vectors $\mathbf{x}$ and $\mathbf{y}$ can be written in terms of norms

$$\mathbf{x}^T\mathbf{y} = \|\mathbf{x}\|_2\|\mathbf{y}\|_2 \cos\theta$$

  where θ is the angle between $\mathbf{x}$ and $\mathbf{y}$

- The “size” of a matrix can be measured with the **Frobenius norm:**

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

Special Matrices and Vectors

Kumar Bipin

Special Matrices and Vectors

Some special kinds of matrices and vectors are particularly useful:

- **Diagonal matrices** $\mathbf{D} = \mathrm{diag}(\mathbf{d})$ have non-zero values only at its diagonal
- Multiplying with diagonal matrices is easy: $\mathrm{diag}(\mathbf{d})\mathbf{x} = \mathbf{d} \odot \mathbf{x}$
- Inverting diagonal matrices is easy: $\mathrm{diag}(\mathbf{d})^{-1} = \mathrm{diag}((1/d_1, \ldots, 1/d_N)^T)$
- A **symmetric matrix** is equal to its own transpose: $\mathbf{A} = \mathbf{A}^T$
- Symmetric matrices arise for instance if entries are distances with $a_{i,j} = a_{j,i}$
- A **unit vector** is a vector with unit ℓ_2 norm: $\|\mathbf{x}\|_2 = 1$
- A vector $\mathbf{x}$ is **orthogonal** to vector $\mathbf{y}$ if $\mathbf{x}^T\mathbf{y} = 0$
- Unit vectors that are orthogonal are called **orthonormal**
- A **orthogonal matrix** is a square matrix whose rows/columns are orthonormal:

$$\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I} \quad \Rightarrow \quad \mathbf{A}^{-1} = \mathbf{A}^T \quad \text{(cheap inverse)}$$

Eigenvalue and Singular Value Decomposition

Kumar Bipin

Eigenvalue Decomposition

- Many mathematical objects can be better understood by breaking them into parts (e.g., integers can be decomposed into prime numbers)
- An **eigendecomposition** decomposes a matrix into eigenvectors and eigenvalues
- An **eigenvector** of square matrix $\mathbf{A}$ is a non-zero vector such that multiplication by $\mathbf{A}$ only alters its scale known as the corresponding **eigenvalue**:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

- As any scaled version of $\mathbf{v}$ is an eigenvector we consider unit eigenvectors

Eigenvalue Decomposition

- We concatenate all **eigenvectors** to form a matrix $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_N)$
- We form all **eigenvalues** into a diagonal matrix $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)^T$
  (remark: by convention, we typically sort the eigenvalues in descending order)
- The **eigendecomposition** of $\mathbf{A}$ is given by

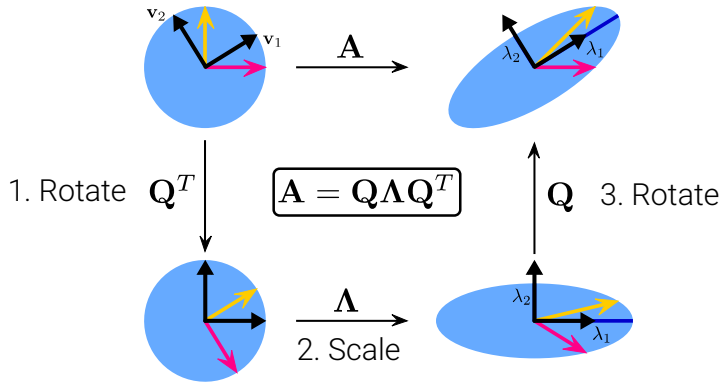
$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$$

- Every **real symmetric** matrix $\mathbf{A}$ (common case) can be decomposed into

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

  where $\mathbf{Q}$ is an orthonormal matrix composed of the eigenvectors $\{\mathbf{v}_i\}_{i=1}^N$ of $\mathbf{A}$

Eigenvalue Decomposition



- Consider a matrix $\mathbf{A}$ with two orthonormal EVs $\mathbf{v}_1$ (with λ_1) and $\mathbf{v}_2$ (with λ_2)
- $\mathbf{A}$ transforms space by 1. rotating, 2. scaling (along CS axes) and 3. rotating back
- Hence, $\mathbf{A}$ distorts the unit circle by scaling space in direction $\mathbf{v}_i$ by λ_i

Eigenvalue Decomposition

- A matrix is **singular** $\Leftrightarrow$ any of its eigenvalues is zero
- The **rank** of a matrix equals the number of non-zero eigenvalues
- Only matrices with **full rank** can be inverted (non-singular matrices)
- A matrix whose eigenvalues are all positive is called **positive definite**
- A matrix whose eigenvalues are all positive or zero is called **positive semi-definite**
- For positive semi-definite matrices we have $\forall_{\mathbf{x}} : \mathbf{x}^T \mathbf{A}\, \mathbf{x} \geq 0$
- Positive definite matrices additionally guarantee that $\mathbf{x}^T \mathbf{A}\, \mathbf{x} = 0 \Rightarrow \mathbf{x} = \mathbf{0}$
- The EVD can be computed easily in NumPy: `numpy.linalg.eig`
- More info: `https://guzintamath.com/textsavvy/2018/05/26/eigenvalues-and-eigenvectors/`

Singular Value Decomposition

- Eigenvalue decomposition can only be applied to square matrices
- For non-square matrices we can use **singular value decomposition**
- The singular value decomposition factorizes a matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ as

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

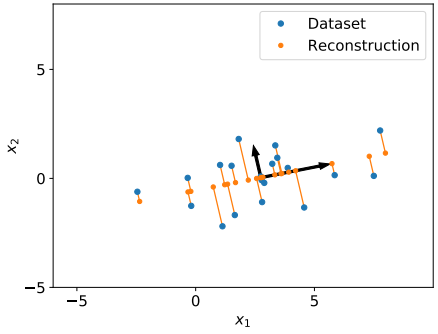
  where $\mathbf{U} \in \mathbb{R}^{M \times M}$, $\mathbf{D} \in \mathbb{R}^{M \times N}$ and $\mathbf{V} \in \mathbb{R}^{N \times N}$
- $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices
- $\mathbf{D}$ is a diagonal (square or non-square) matrix
- The elements along the diagonal of $\mathbf{D}$ are known as **singular values**
- The columns of $\mathbf{U}$ and $\mathbf{V}$ are **left-/right-singular vectors,** respectively

Relationship between EVD and SVD

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$
$$\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{V}\mathbf{D}^2\mathbf{V}^T$$
$$\mathbf{A}\mathbf{A}^T = \mathbf{U}\mathbf{D}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{U}^T = \mathbf{U}\mathbf{D}^2\mathbf{U}^T$$

- The right-singular vectors $\mathbf{V}$ are the eigenvectors of $\mathbf{A}^T\mathbf{A}$
- The left-singular vectors $\mathbf{U}$ are the eigenvectors of $\mathbf{A}\mathbf{A}^T$
- The eigenvalues of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$ are equal to the squared singular values of $\mathbf{A}$

Application: Principal Component Analysis (PCA)



- PCA is a technique for **analyzing** large high-dimensional datasets (see DL lec. 11)
- Eigenvectors of cov. matrix (=principal comp.) face direction of **largest variance**
- In this illustration, the two eigenvectors $\mathbf{v}_i$ shown in black are scaled by $\sqrt{\lambda_i}$

The Trace Operator and Determinant

Kumar Bipin

The Trace Operator

- The **trace operator** returns the sum of all diagonal elements of a matrix:

$$\mathrm{Tr}(\mathbf{A}) = \sum_i a_{i,i}$$

The Trace Operator

- The trace operator allows for writing the **Frobenius norm without summation:**

$$\|A\|_F = \sqrt{\mathrm{Tr}(\mathbf{A}\mathbf{A}^T)}$$

- The trace operator is **invariant to the transpose operator:**

$$\mathrm{Tr}(\mathbf{A}) = \mathrm{Tr}(\mathbf{A}^T)$$

- The trace of the product of square matrices is **invariant to cyclic permutations:**

$$\mathrm{Tr}(\mathbf{ABC}) = \mathrm{Tr}(\mathbf{CAB}) = \mathrm{Tr}(\mathbf{BCA})$$

The Determinant

- The **determinant** of a square matrix is equal to the product of all eigenvalues:

$$\det(\mathbf{A}) = \prod_i \lambda_i$$

- The determinant can be thought of as a **measure** of how much multiplication by the matrix expands or contracts space
- If the determinant is 1, then the transformation is **volume-preserving**
- If the determinant is 0, then space is contracted completely along at least one dimension, causing it to lose all of its volume
- In other words, the matrix is **singular** or **rank-deficient**

Differential Calculus

Kumar Bipin

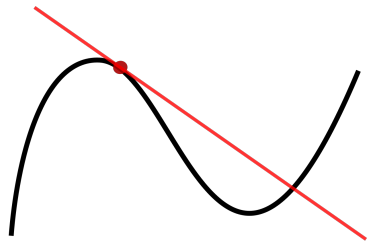
Derivative

- The **derivative** of a function $f(x)$ measures the sensitivity to change of the function value wrt. a change in its argument and is given by

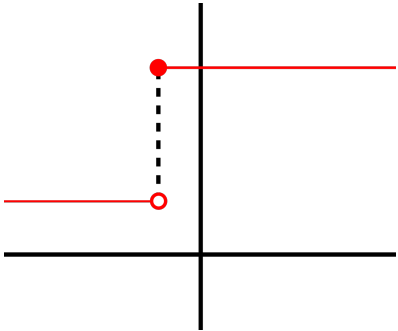
$$f'(a) = \frac{df}{dx}(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$$

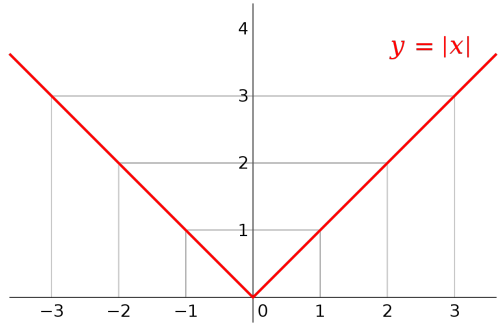
- A function is **differentiable** at a if the limit exists
- The slope of the tangent line is equivalent to the derivative at the tangent point (dark red)
- The **second derivative** is written as

$$f''(a) \quad \text{or} \quad \frac{d^2 f}{dx^2}(a)$$



Examples of Non-Differentiable Functions





- not continuous
- not differentiable

- continuous
- not differentiable

Chain Rule

- The **chain rule** expresses the derivative of the composition of two differentiable functions f and g in terms of the derivatives of f and g
- More precisely, if $h = f \circ g$ is the function such that $h(x) = f(g(x))$ then

$$h'(x) = f'(g(x))g'(x) \qquad \text{Lagrange notation}$$

$$\frac{df}{dx} = \frac{df}{dg}\frac{dg}{dx} \qquad \text{Leibniz notation}$$

- Example:

$$h(x) = (2x^2 + x)^3$$

$$h'(x) = 3(2x^2 + x)^2(4x + 1)$$

Derivatives of Multivariate Functions

Let $f(x, y)$ be a function where $y = y(x)$ depends on x.

The **partial derivative** is defined as:

$$\frac{\partial f(x,y)}{\partial x} = \underbrace{\frac{\partial f}{\partial x}\frac{dx}{dx}}_{\text{Chain Rule}}$$

The **total derivative** is defined as:

$$\frac{df(x,y)}{dx} = \underbrace{\frac{\partial f}{\partial x}\frac{dx}{dx} + \frac{\partial f}{\partial y}\frac{dy}{dx}}_{\text{Multi-Variable Chain Rule}}$$

Example:

$$f(x, y) = xy$$
$$\frac{\partial f(x,y)}{\partial x} = y$$

Example:

$$f(x, y) = xy \ \wedge \ y = x$$
$$\frac{df(x,y)}{dx} = y + x = 2x$$

- Remark: We sometimes write ∂ instead of d, but refer to the total derivative

Implicit Functions and Implicit Differentiation

An **implicit equation** is a relation

$$f(x, y) = 0$$

where $y(x)$ is defined only implicitly.

Implicit differentiation computes the total derivative of both sides wrt. x

$$\frac{\partial f}{\partial x}\frac{dx}{dx} + \frac{\partial f}{\partial y}\frac{dy}{dx} = 0$$

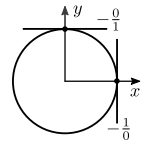
and solves for $\frac{dy}{dx}$.

Example: Let's assume

$$x^2 + y^2 = 1$$

Implicit differentiation yields

$$2x + 2y\frac{dy}{dx} = 0$$
$$\frac{dy}{dx} = -\frac{x}{y}$$



Note the presence of y in this term, i.e., the implicit "function" is a curve.

Vector Calculus

Kumar Bipin

Derivative of a Vector-to-Vector Function

Let $\mathbf{f}\colon \mathbb{R}^N \mapsto \mathbb{R}^M$. Then the (partial) derivative of $\mathbf{f}$ wrt. $\mathbf{x}$ is given by

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}) = \mathbf{J}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_N}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_M}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_M}{\partial x_N}(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^{M \times N}$$

where $\frac{\partial f_i}{\partial x_j}(\mathbf{x})$ is the (partial) derivative of the function $\mathbf{f}$'s i-th output wrt. to the j-th component x_j of the function $\mathbf{f}$'s input vector $\mathbf{x}$. $\mathbf{J}(\mathbf{x})$ is called the **Jacobian** matrix.

Two frequent Special Cases

Scalar-to-scalar function: $N = M = 1$. Then $f\colon \mathbb{R}^1 \mapsto \mathbb{R}^1$.

$$f'(x) = \frac{\partial f}{\partial x}(x) \in \mathbb{R}^{1\times 1} \equiv \mathbb{R}$$

Vector-to-scalar function: $N > 1, M = 1$. Then $f\colon \mathbb{R}^N \mapsto \mathbb{R}^1$.

$$(\nabla_{\mathbf{x}} f)(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f}{\partial x_N}(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^{1\times N}$$

The alternative notation $(\nabla_{\mathbf{x}} f)(\mathbf{x})$ is read as the **gradient** of f at $\mathbf{x}$ and is exclusviely used for vector-to-scalar functions.

Gradient and Hessian

Consider a **vector-to-scalar function** $f : \mathbb{R}^N \mapsto \mathbb{R}^1$.

- The first-order derivative of f is called the **gradient** vector. The components of the gradient vector are the **first-order** partial derivatives of f at $\mathbf{x}$:

$$(\nabla_{\mathbf{x}} f)(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f}{\partial x_N}(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^{1 \times N}$$

- The second-order derivative of f is called the **Hessian** matrix $\mathbf{H}$. The components of the Hessian matrix are the **second-order** partial derivatives of f at $\mathbf{x}$:

$$\mathbf{H}(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(\mathbf{x}) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_N}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_N \partial x_1}(\mathbf{x}) & \cdots & \frac{\partial^2 f}{\partial x_N^2}(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^{N \times N}$$

Chain Rule for Vector-to-Vector Functions

Let $\mathbf{f}\colon \mathbb{R}^N \mapsto \mathbb{R}^N$ with $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}\colon \mathbb{R}^M \mapsto \mathbb{R}^P$ with $\mathbf{g}(\mathbf{y})$. If

$$\mathbf{h}(\mathbf{x}) = \mathbf{g}(\mathbf{f}(\mathbf{x}))$$

then $\mathbf{h}\colon \mathbb{R}^N \mapsto \mathbb{R}^P$ and:

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x}) = \frac{\partial \mathbf{g}}{\partial \mathbf{y}}(\mathbf{f}(\mathbf{x}))\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x})$$

Here, the $P \times M$ matrix $\frac{\partial \mathbf{g}}{\partial \mathbf{y}}(\mathbf{f}(\mathbf{x}))$ gets multiplied by the $M \times N$ matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x})$ to form the resulting $P \times N$ matrix $\frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x})$.

Special Case of Chain Rule

Let $\mathbf{f}\colon \mathbb{R}^1 \mapsto \mathbb{R}^N$ with $\mathbf{f}(x)$ and $g\colon \mathbb{R}^N \mapsto \mathbb{R}^1$ with $g(\mathbf{y})$. If

$$h(x) = g(\mathbf{f}(x))$$

then $h\colon \mathbb{R}^1 \mapsto \mathbb{R}^1$ and:

$$\frac{\partial h}{\partial x}(x) = \frac{\partial g}{\partial \mathbf{y}}(\mathbf{f}(x))\frac{\partial \mathbf{f}}{\partial x}(x) = \sum_{i=1}^{n} \frac{\partial g}{\partial y_i}(\mathbf{f}(x))\frac{\partial f_i}{\partial x}(x)$$

Here the $1 \times N$ matrix $\frac{\partial g}{\partial \mathbf{y}}(\mathbf{f}(x))$ gets multiplied by the $N \times 1$ matrix $\frac{\partial \mathbf{f}}{\partial x}(x)$ to form the resulting 1×1 matrix $\frac{\partial h}{\partial x}(x)$. Note that this is the **total derivative!**

Random Variables and Probability Distributions

Kumar Bipin

Why Probability?

Nearly all activities require some ability to reason in the presence of uncertainty.

There exist 3 sources of uncertainty:

- Inherent stochasticity in the modeled system (e.g., traffic participants)
- Incomplete observability (e.g., 3D reconstruction)
- Incomplete modeling (e.g., discretization of robot location)

Probability Theory:

- Mathematical framework for representing uncertain statements
- Tool to quantify uncertainty and reason in the presence of uncertainty
- Information quantifies the amount of uncertainty in a distribution

Terminology

- A **random variable** is a variable that can take on different values randomly
- Random variables may be either **discrete** or **continuous**
- A **discrete random variable** has a finite or countably infinite number of states
- A **continuous random variable** is associated with a real value
- A **probability distribution** is a description of how likely a random variable or set of random variables is to take on each of its possible states.It is described by:
  - a **probability mass function (PMF)** in the case of discrete variables
  - a **probability density function (PDF)** in the case of continuous variables
- The reader has to infer which function to use based on the identity of the RV

Discrete Probability Distributions

- A **random variable** X can take values from a **discrete set** of outcomes $\mathcal{X}$
- Example: 6-sided dice with equal probability $1/6$ for each of the 6 numbers
- We usually use the short-hand notation

$$p(x) \quad \text{for} \quad p(X = x) \in [0, 1]$$

  for the **probability** that random variable X takes value x
- $p(x)$ is called the **probability mass function**
- In contrast, with $p(X)$ we denote the **probability distribution** over X
- If X follows distribution $p(X)$ we also write $X \sim p(X)$
- The probability $p(x)$ must satisfy the following **conditions:**

$$p(x) \geq 0 \quad \text{and} \quad \sum_{x \in \mathcal{X}} p(x) = 1$$

Joint, Marginal and Conditional Probability

- **Joint probability** (of X and Y)

$$p(x, y) \quad \text{for} \quad p(X = x, Y = y)$$

- **Conditional probability** (of X conditioned on Y)

$$p(x|y) = \frac{p(x, y)}{p(y)} \quad \text{for} \quad p(X = x|Y = y) = \frac{p(X = x, Y = y)}{p(Y = y)}$$

- **Marginal probability** (of Y)

$$p(y) = \sum_{x \in \mathcal{X}} p(x, y) \quad \text{for} \quad p(Y = y) = \sum_{x \in \mathcal{X}} p(X = x, Y = y)$$

Joint, Marginal and Conditional Probability

- **Joint probability**

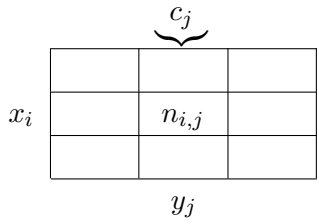
$$p(x_i, y_j) = \frac{n_{i,j}}{N} = \frac{n_{i,j}}{\sum_{i,j} n_{i,j}}$$

- **Conditional probability**

$$p(x_i \mid y_j) = \frac{n_{i,j}}{c_j} = \frac{n_{i,j}}{\sum_i n_{i,j}}$$

- **Marginal probability**

$$p(y_j) = \frac{c_j}{N} = \frac{\sum_i n_{i,j}}{\sum_{i,j} n_{i,j}}$$



$$c_j = \sum_i n_{i,j}$$

$$N = \sum_{i,j} n_{i,j}$$

Joint, Marginal and Conditional Probability

Example:

	y_1	y_2	y_3
x_1	1	1	1
x_2	1	1	2
x_3	1	1	1

Counts

	y_1	y_2	y_3
x_1	0.1	0.1	0.1
x_2	0.1	0.1	0.2
x_3	0.1	0.1	0.1

Joint Probabilities

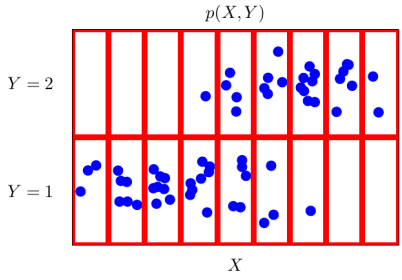
0.3	0.1	0.1	0.1
0.4	0.1	0.1	0.2
0.3	0.1	0.1	0.1
	0.3	0.3	0.4

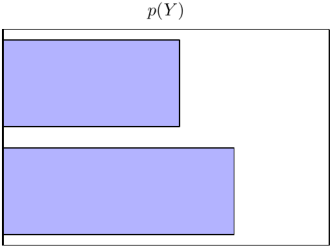
Marginals $p(x)$ and $p(y)$

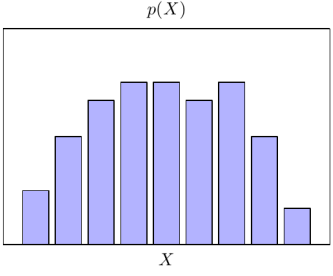
0.25	0.1	0.1	**0.1**
0.50	**0.1**	**0.1**	**0.2**
0.25	0.1	0.1	**0.1**
	0.25	0.25	0.50

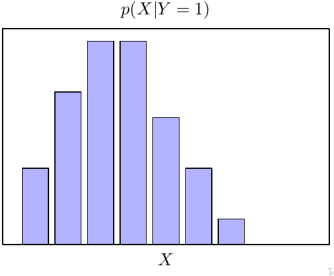
Conditionals $p(x|y_3)$ and $p(y|x_2)$

Joint, Marginal and Conditional Probability









The Rules of Probability

- **Sum rule**

$$p(x) = \sum_{y \in \mathcal{Y}} p(x, y) \quad \text{or} \quad p(y) = \sum_{x \in \mathcal{X}} p(x, y)$$

  We say that we "marginalize" x / y $\Rightarrow$ $p(x)$ / $p(y)$ are called marginal probability.

- **Product rule**

$$p(x, y) = p(y|x)p(x) = p(x|y)p(y)$$

- And as a consequence we obtain **Bayes Theorem:**

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

Continuous Probability Distributions

- Now X is a **continuous** random variable, e.g., taking values in $\mathbb{R}$
- The probability that X takes a value in the interval (a, b) is

$$p(X \in (a, b)) = \int_a^b p(x)\, dx$$

and we call $p(x)$ the **probability density function (PDF)**

Probability Densities

- $p(x)$ must satisfy the following **conditions:**

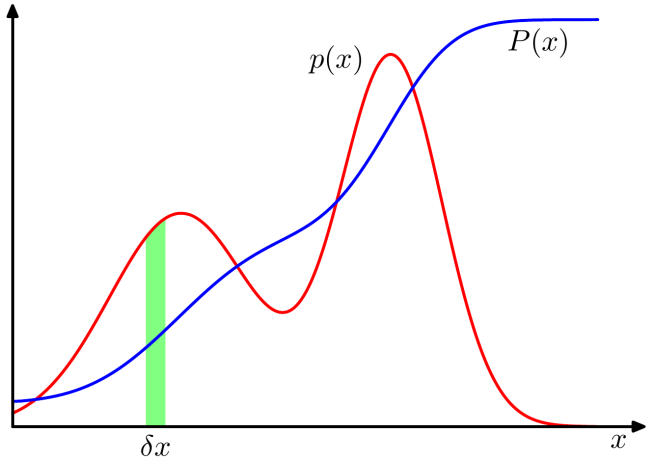
$$\begin{aligned} p(x) &\geq 0 \\ \int_{-\infty}^{\infty} p(x) &= 1 \end{aligned}$$

- The probability of $x \in (-\infty, z)$ is given by the **cumulative distribution function:**

$$P(z) = \int_{-\infty}^{z} p(x)dx$$

- Note that $p(X = x)$ is often 0 while $p(x)$ is often greater than 0. For continuous distributions we consider $p(x)$ as the PDF and not as short-hand for $p(X = x)$.

Probability Densities



Probability density of a continuous variable

Joint, Marginal and Conditional Probability

- **Joint density** (of X and Y)

$$p(x, y)$$

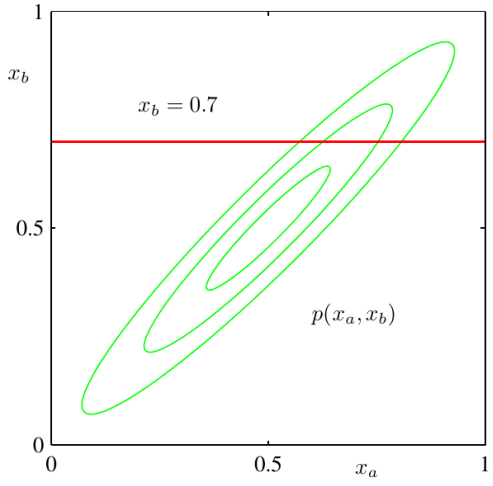
- **Conditional density** (of X conditioned on Y)

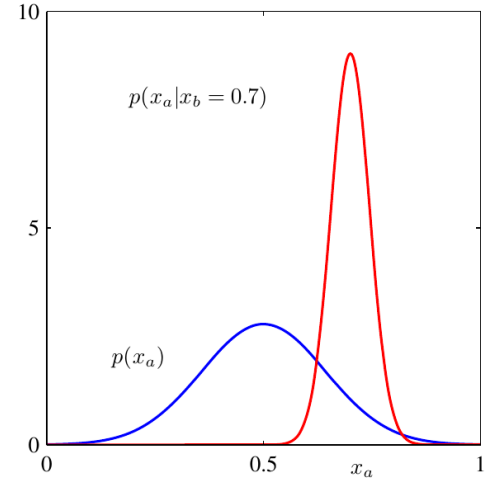
$$p(x|y) = \frac{p(x, y)}{p(y)}$$

- **Marginal density** (of Y)

$$p(y) = \int_{x \in \mathcal{X}} p(x, y) dx$$

Joint, Marginal and Conditional Probability





joint, marginal, conditional probability

The Rules of Probability

- **Sum rule**

$$p(x) = \int_{y \in \mathcal{Y}} p(x, y) dy \quad \text{or} \quad p(y) = \int_{x \in \mathcal{X}} p(x, y) dx$$

  We say that we “marginalize” x / y $\Rightarrow$ $p(x)$ / $p(y)$ are called marginal density.

- **Product rule**

$$p(x, y) = p(y|x)p(x) = p(x|y)p(y)$$

- And as a consequence we obtain **Bayes Theorem:**

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

Independence and Conditional Independence

- Two random variables X and Y are **independent** if their probability distribution can be expressed as a product of two factors:

$$p(x, y) = p(x)\, p(y)$$

- Two random variables X and X are **conditionally independent** given a random variable Z if the conditional probability distribution over X and Y factorizes as follows:

$$p(x, y|z) = p(x|z)\, p(y|z)$$

- We denote these two statements compactly as $X \perp\!\!\!\perp Y$ and $X \perp\!\!\!\perp Y|Z$

Common Probability Distributions

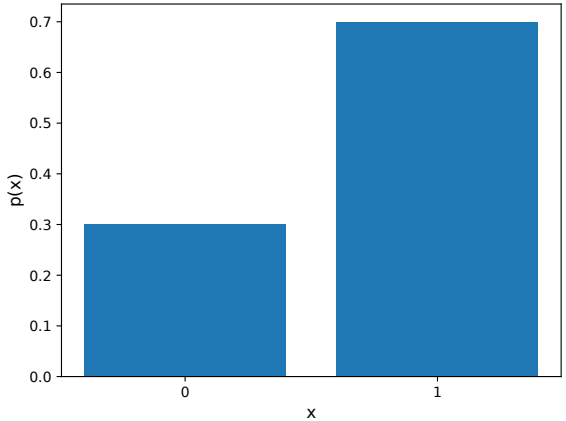
Kumar Bipin

Bernoulli Distribution

Bernoulli distribution:

$$p(x) = \mu^x (1-\mu)^{(1-x)}$$

- μ: probability for $x = 1$
- Handles two classes
  e.g. (“cats” vs. “dogs”)

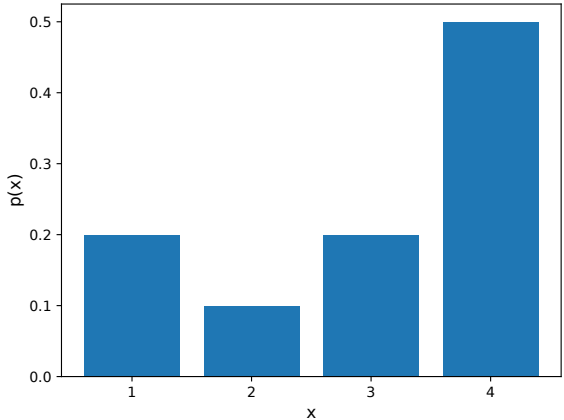


Categorical Distribution

Categorical distribution:

$$p(x) = \mu_x$$

- μ_x: probability for class x
- Handles multiple classes

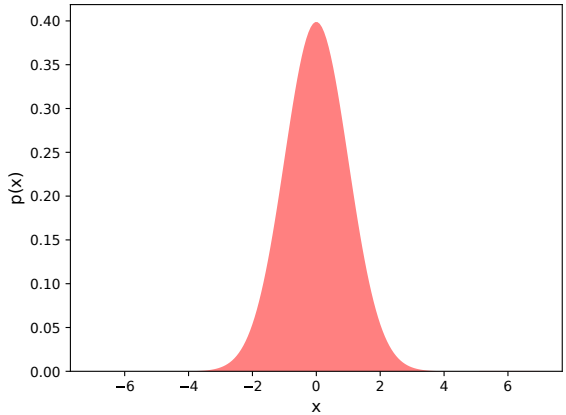


Gaussian Distribution

Gaussian distribution:

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

- μ : mean
- σ : standard deviation
- The distribution has thin “tails”: $p(x) \to 0$ quickly as $x \to \infty$

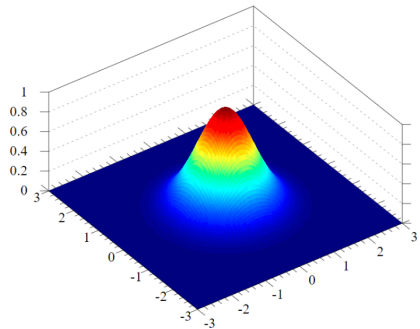


Multivariate Gaussian Distribution

Multivariate Gaussian distribution:

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

- $\boldsymbol{\mu} \in \mathbb{R}^N$: mean vector
- $\boldsymbol{\Sigma} \in \mathbb{R}^{N \times N}$: covariance matrix
- The distribution has thin “tails”: $p(\mathbf{x}) \to 0$ quickly as $\mathbf{x} \to \infty$

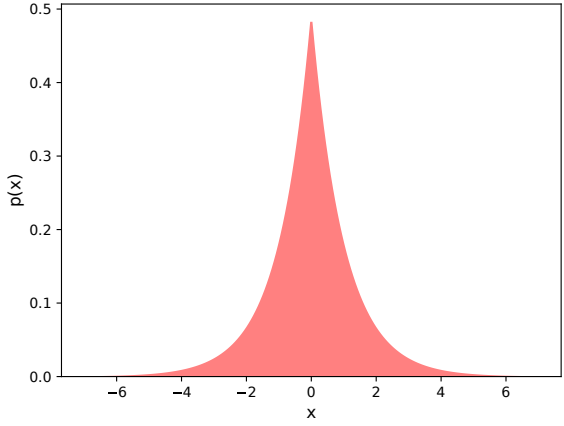


Laplace Distribution

Laplace distribution:

$$p(x) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right)$$

- μ : location
- b : scale
- The distribution has heavy “tails”: $p(x) \to 0$ more slowly as $x \to \infty$



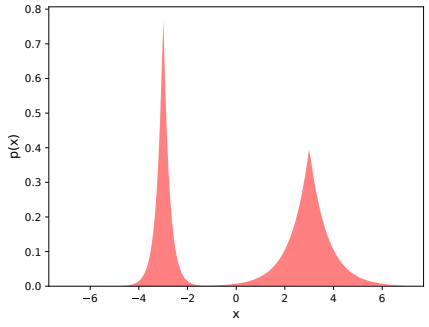
Mixture Distributions

We can also model **mixture densities:**

$$p(x) = \sum_{m=1}^{M} \pi_m \frac{1}{2\,b_m} \exp\left(-\frac{|x-\mu_m|}{b_m}\right)$$

Example:

- Mixture of Laplace distribution
- π_m: weight of mode m
- Constraint $\sum_m \pi_m = 1$



Mixture Distributions

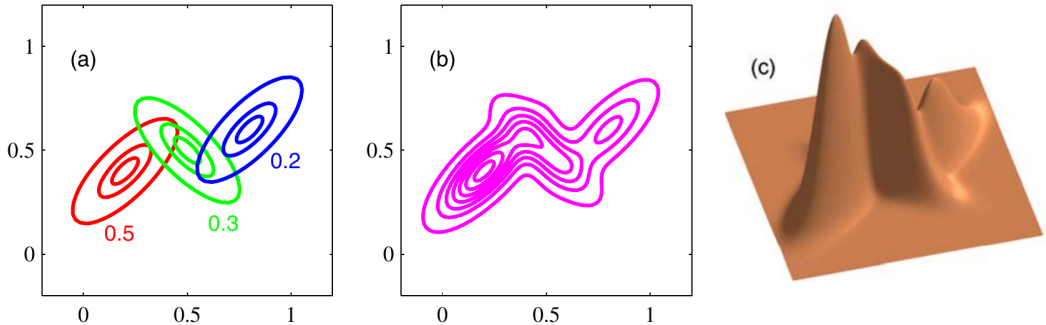


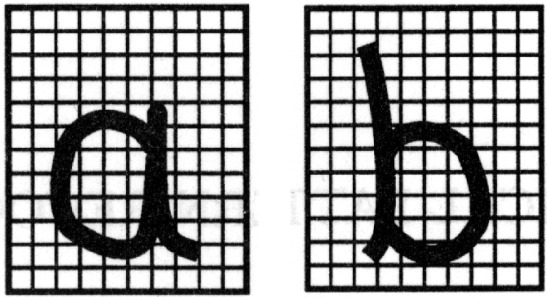
Figure 2.23 Illustration of a mixture of 3 Gaussians in a two-dimensional space. (a) Contours of constant density for each of the mixture components, in which the 3 components are denoted red, blue and green, and the values of the mixing coefficients are shown below each component. (b) Contours of the marginal probability density $p(\mathbf{x})$ of the mixture distribution. (c) A surface plot of the distribution $p(\mathbf{x})$.

Bayesian Decision Theory

Kumar Bipin

Digit Classification

- Classify digits “a” versus “b”



- **Goal:** classify new digits such that probability of error is minimized

Digit classification: Prior

Prior Distribution:

- How often do the letters “a” and “b” occur ?
- Let us assume

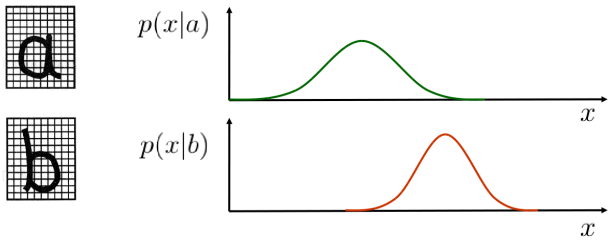
$$c_1 = a \qquad p(c_1) = 0.75$$
$$c_2 = b \qquad p(c_2) = 0.25$$

- Note that the **prior** has to be a distribution, in particular:

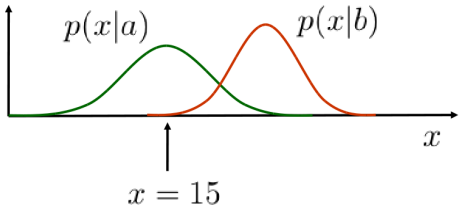
$$\sum_{k=1,2} p(c_k) = 1$$

Digit Classification: Class Conditionals

- We describe every digit using some **feature vector** $\mathbf{x}$, e.g.:
  - the number of black pixels in each box
  - relation between width and height
- **Likelihood:** How likely has $\mathbf{x}$ been generated from $p(\mathbf{x} \mid a)$ or $p(\mathbf{x} \mid b)$?

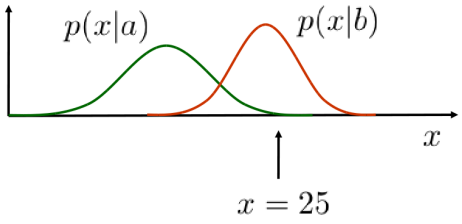


Digit Classification



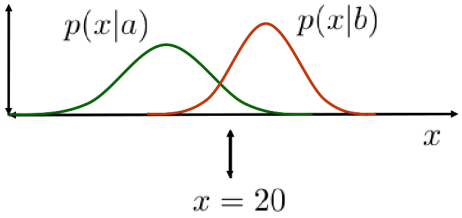
- Which class should we assign $\mathbf{x}$ to?
- Class a

Digit Classification



- Which class should we assign $\mathbf{x}$ to ?
- Class b

Digit Classification



- Which class should we assign $\mathbf{x}$ to ?
- Class a, since p(a)=0.75

Bayes Theorem

- How do we formalize this?
- We already mentioned **Bayes Theorem:**

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

- Let us now apply it:

$$p(c_k|\mathbf{x}) = \frac{p(\mathbf{x}|c_k)p(c_k)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|c_k)p(c_k)}{\sum_i p(\mathbf{x}|c_i)p(c_i)}$$

Bayes Theorem

- Repeated from last slide:

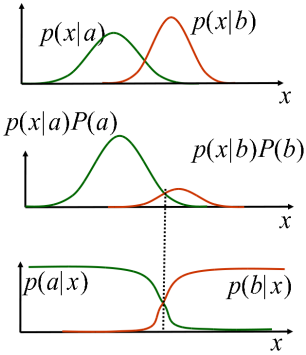
$$p(c_k|\mathbf{x}) = \frac{p(\mathbf{x}|c_k)p(c_k)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|c_k)p(c_k)}{\sum_i p(\mathbf{x}|c_i)p(c_i)}$$

- We use the following names:

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Normalization Factor}}$$

- The normalization factor is also called the **Partition Function** or **Evidence** and commonly denoted with the symbol Z

Bayes Theorem



Likelihood

Likelihood $\times$ Prior

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Normalization Factor}}$$

Bayesian Decision Theory:

- Model prior and likelihood; decide for class with highest posterior
- Decision theory typically additionally considers loss function/risk

Expectation, Variance and Covariance

Kumar Bipin

Expectation of a Function

- The **expectation** or **expected value of a function** $f(x)$ with respect to a probability distribution $p(X)$ is the average of $f(x)$ for $X \sim p(X)$
- For discrete variables this can be computed with a summation:

$$\mathbb{E}_{X \sim p}[f(x)] = \sum_{x \in \mathcal{X}} p(x) f(x)$$

- For continuous variables, it is computed with an integral:

$$\mathbb{E}_{X \sim p}[f(x)] = \int_{x \in \mathcal{X}} p(x) f(x)\, dx$$

Expectation of a Random Variable

- An important special case is the one where $f(x) = x$
  in which case we obtain the **expectation of the random variable** X
- For discrete variables we have:

$$\mathbb{E}_{X \sim p}[x] = \sum_{x \in \mathcal{X}} p(x)x$$

- For continuous variables we obtain:

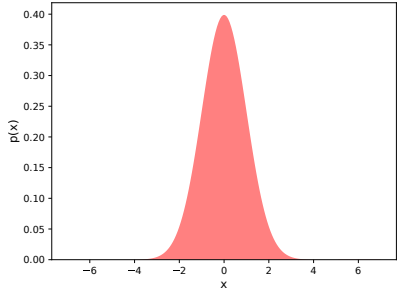
$$\mathbb{E}_{X \sim p}[x] = \int_{x \in \mathcal{X}} p(x)x \, dx$$

Properties of Expectations

- Expectations are **linear:**

$$\mathbb{E}[\alpha\, f(x) + \beta\, g(x)] = \alpha\, \mathbb{E}[f(x)] + \beta\, \mathbb{E}[g(x)]$$

- The expected value of a Gaussian random variable is the mean of its distribution:



Variance of a Function

- The **variance** measures how much the values of a function of a random variable X vary as we sample different values of x from its probability distribution:

$$\text{Var}[f(x)] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right]$$

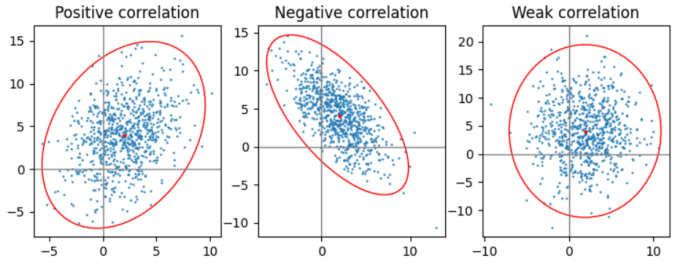
- When the variance is low, the values of $f(x)$ cluster near their expected value
- The square root of the variance is known as the **standard deviation**
- The variance/standard deviation is a parameter of the Gaussian distribution

Covariance of Functions

- The **covariance** measures how much two values are linearly related:

$$\text{Cov}[f(x), g(y)] = \mathbb{E}\left[\left(f(x) - \mathbb{E}\left[f(x)\right]\right)\left(g(y) - \mathbb{E}\left[g(y)\right]\right)\right]$$

- The covariance matrix of a random vector $\mathbf{x} \in \mathbb{R}^N$ is a $N \times N$ matrix
- The diagonal elements of the covariance matrix are the individual variances
- The sign of the covariance determines if variables are pos./neg. correlated



Information and Entropy

Kumar Bipin

Information Theory

- Branch of applied mathematics
- Goal: quantify how much information is in a signal
- Father of information theory: Claude E. Shannon
- Founding work of the field of information theory:
  Claude E. Shannon: “A Mathematical Theory of Communication”

A Mathematical Theory of Communication

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist[1] and Hartley[2] on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

Shannon: A mathematical theory of communication. Bell System Technical Journal, 1948.

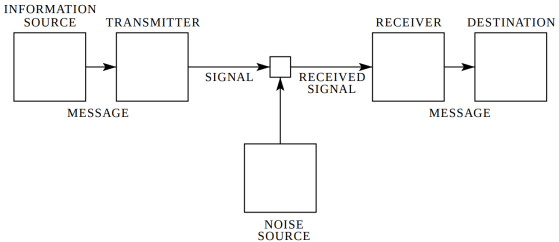
A Mathematical Theory of Communication

Claude Shannon
1916–2001

After graduating from Michigan and MIT, Shannon joined the AT&T Bell Telephone laboratories in 1941. His paper 'A Mathematical Theory of Communication' published in the *Bell System Technical Journal* in 1948 laid the foundations for modern information theory. This paper introduced the word 'bit', and his concept that information could be sent as a stream of 1s and 0s paved the way for the communications revolution. It is said that von Neumann recommended to Shannon that he use the term entropy, not only because of its similarity to the quantity used in physics, but also because "nobody knows what entropy really is, so in any discussion you will always have an advantage".

Shannon: A mathematical theory of communication. Bell System Technical Journal, 1948.

Information Theory



- Information theory was originally invented to study **sending messages** from discrete alphabets over noisy channels (e.g., communication via radio waves)
- Information theory tells how to **design optimal codes** and calculate the expected length of messages sampled using various coding schemes
- Here, we mostly use a few key ideas from information theory to **characterize probability distributions** or quantify similarity between probability distributions

Information Theory

Basic Intuition:

- The basic intuition behind information theory is that observing an **unlikely event** is **more informative** than observing a likely event

Example:

- The message “the sun rose this morning” is **uninformative** (not worth sending)
- In contrast, the message “there was a solar eclipse this morning” is **informative**
- This is a rare event - in other words, receiving this message surprises us
- We will quantify information content as the **“level of surprise”**

Self-Information

We would like to **quantify information** in a way that formalizes this intuition:

- **Likely events** should have **low information content** & certain events none
- **Less likely events** should have **higher information content**
- Independent events should have **additive information**
  (Finding out that a tossed coin has come up as heads twice should convey twice as much information as finding out that it has come up as heads once.)

To satisfy all requirements, we define the **self-information** of an event $X = x$ as

$$I(x) = -\log p(x) \quad \text{[nats]}$$

where $\log$ refers to the natural logarithm with units of "nats" and $I(x) \geq 0$.

Self-Information

To satisfy all requirements, we define the **self-information** of an event $X = x$ as

$$I(x) = -\log p(x) \quad \text{[nats]}$$

where $\log$ refers to the natural logarithm with units of "nats" and $I(x) \geq 0$.

Remarks:

- Self-information is also interpreted as **quantifying the level of "surprise"**
- One nat is the amount of information gained by observing an event of probability $\frac{1}{e}$
- When using **base-2 logarithms,** units are called **"bits"** or "shannons"
- Information measured in bits is just a rescaling of information measured in nats

Self-Information

- Single fair coin toss: $I(x) = -\log_2 p(x) = -\log_2 \frac{1}{2} = 1$ bit
- Two fair coin tosses: $I(x) = -\log_2 p(x) = -\log_2 \frac{1}{4} = 2$ bits
- Single unfair coin toss: $I(x) = -\log_2 p(x) = -\log_2 1 = 0$ bits

Shannon Entropy

- Self-information deals only with a single outcome
- We can quantify the amount of uncertainty in an entire probability distribution $p(X)$ using the **Shannon Entropy:**

$$H(p) = \mathbb{E}_{X \sim p}[I(x)] = -\mathbb{E}_{X \sim p}[\log p(x)]$$

- Entropy $H(p)$ = **expected amount of information** when drawing from $p(X)$
- As the self-information is positive $I(x) \geq 0$, the entropy is also positive $H(p) \geq 0$
- When using base-2 logarithm, the entropy specifies the lower bound on **number of bits required** on average to encode symbols drawn from a distribution $p(X)$
- When x is continuous, the Shannon entropy is known as **differential entropy**

Shannon Entropy

Shannon Entropy:

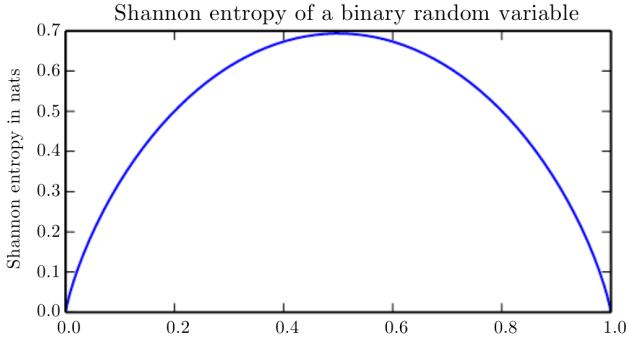
$$H(p) = -\mathbb{E}_{X\sim p}[\log p(x)] = -\sum_{x\in\mathcal{X}} p(x)\log p(x)$$

Differential Entropy:

$$H(p) = -\mathbb{E}_{X\sim p}[\log p(x)] = -\int_{x\in\mathcal{X}} p(x)\log p(x)\,dx$$

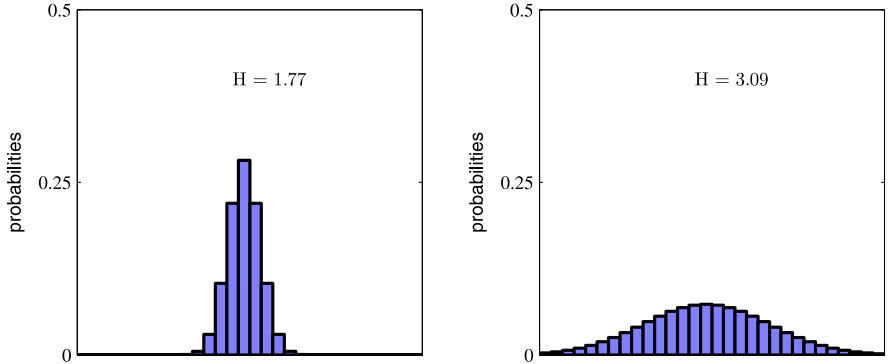
- By convention, we treat $0\log 0$ as $\lim_{x\to 0} x\log x = 0$

Shannon Entropy Example



- X-Axis: Probability p of a binary random variable (e.g., coin toss) being equal to 1
- Y-Axis: Entropy of corresponding distribution $H(p) = -(1-p)\log(1-p) - p\log p$
- Distributions that are close to deterministic have **low entropy**
  while distributions that are close to uniform have **high entropy**

Shannon Entropy Example



- Histograms of two **discrete probability distributions** over 30 bins
- The entropy of the broader distribution (on the right) is higher
- A uniform distribution would yield the largest entropy $H = -\log(1/30) = 3.4$ nats

Relation to Shortest Coding Length

- Consider a RV X with 8 possible states $\{a, b, c, d, e, f, g, h\}$ and a distribution $p(X)$
- If each state is equally likely, the entropy is: $H(p) = -8 \cdot \frac{1}{8} \log_2 \frac{1}{8} = 3$ bits
- Now let the probabilities be: $\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\right\}$
- Then we have: $H(p) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{4} \log_2 \frac{1}{4} \ \ldots \ - \frac{1}{64} \log_2 \frac{1}{64} = 2$ bits
- The nonuniform distribution has a smaller entropy than the uniform one
- Let's consider coding a message to be sent from a sender to a receiver
- We could do this using a 3-bit number, but this would be suboptimal
- Instead, we can use shorter codes for the more probable events:
  $a = 0, b = 10, c = 110, d = 1110, e = 111100, f = 111101, g = 111110, h = 111111$
- The average code length equals the entropy: $\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \cdots + \frac{1}{64} \cdot 6 = 2$ bits
- Note that shorter codes cannot be used while uniquely disambiguating symbols
  e.g., 11001110 decodes uniquely into the sequence c, a, d

Kullback-Leibler Divergence

Kumar Bipin

Kullback-Leibler Divergence

If we have two probability distributions $p(X)$ and $q(X)$ over the same random variable X, we can use the **Kullback-Leibler (KL) divergence** as a "measure of distance":

$$D_{KL}(p \,||\, q) = \mathbb{E}_{X \sim P}\left[\log \frac{p(x)}{q(x)}\right] = \mathbb{E}_{X \sim p}\left[\log p(x) - \log q(x)\right]$$

- In the case of discrete variables, it measures the extra amount of information required to send a message containing symbols drawn from $p(X)$, when we use a code that was designed to minimize the length of messages drawn from $q(X)$
- The KL divergence is non-negative, and zero only iff $p(X) = q(X)$
- The KL divergence is not a true distance measure: $D_{KL}(p \,||\, q) \neq D_{KL}(q \,||\, p)$
- The KL divergence is a special case of a larger class of so-called **f-divergences**

Kullback-Leibler Divergence

Discrete Distributions:

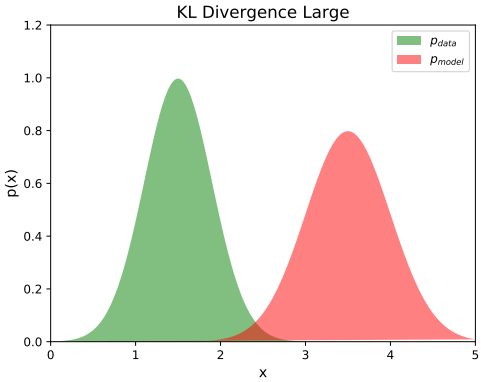
$$D_{KL}(p \,||\, q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$

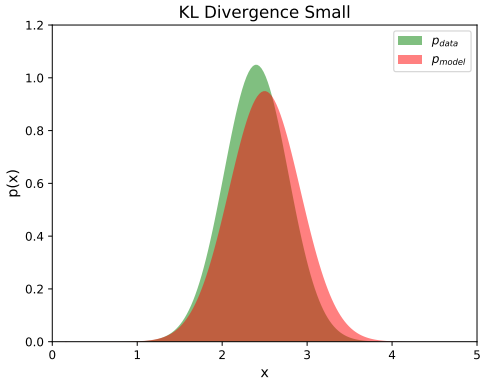
Continuous Distributions:

$$D_{KL}(p \,||\, q) = \int_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \, dx$$

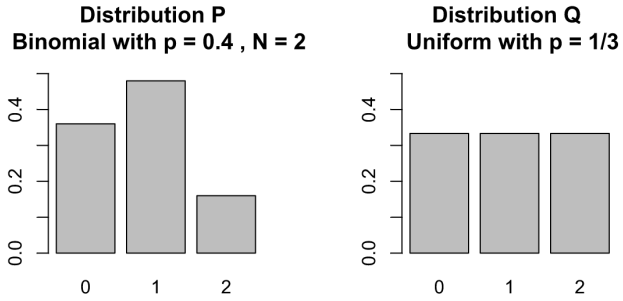
- By convention, we treat $0 \log 0$ as $\lim_{x \to 0} x \log x = 0$

Kullback-Leibler Divergence Example





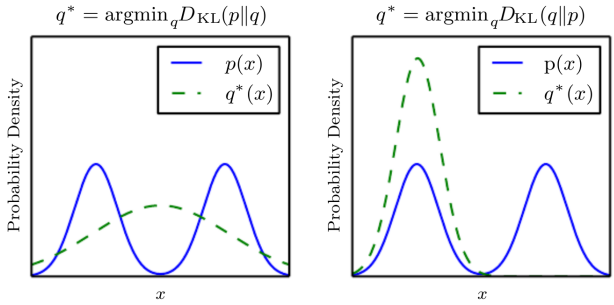
Kullback-Leibler Divergence Example



$$D_{KL}(p \,||\, q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = 0.36 \log \frac{0.36}{0.33} + 0.48 \log \frac{0.48}{0.33} + 0.16 \log \frac{0.16}{0.33} \approx 0.085$$

$$D_{KL}(q \,||\, p) = \sum_{x \in \mathcal{X}} q(x) \log \frac{q(x)}{p(x)} \approx 0.097$$

Kullback-Leibler Divergence Asymmetry



- Minimizing $D_{KL}(p\,||\,q)$ and $D_{KL}(q\,||\,p)$ wrt. q does not lead to the same result
- Left: Select a q that has **high probability** where p has high probability
  $\Rightarrow$ leads to blurring multiple modes to put high probability to all of them
- Right: Select a q that has **low probability** where p has low probability
  $\Rightarrow$ picks one of the modes to not put probability mass in low-prob. region

Relationship to Cross-Entropy

The **cross-entropy** is closely related to the KL divergence:

$$
\begin{aligned}
H(p, q) &= -\mathbb{E}_{X\sim p}[\log q(x)] \\
&= -\mathbb{E}_{X\sim p}[\log p(x)] + \mathbb{E}_{X\sim p}\left[\log p(x)\right] - \mathbb{E}_{X\sim p}\left[\log q(x)\right] \\
&= -\mathbb{E}_{X\sim p}[\log p(x)] + \underbrace{\mathbb{E}_{X\sim p}\left[\log p(x) - \log q(x)\right]}_{\text{Kullback-Leibler Divergence}} \\
&= H(p) + D_{KL}(p \,||\, q)
\end{aligned}
$$

- Minimizing the cross-entropy wrt. q is equivalent to minimizing the KL divergence wrt. q, because $H(p)$ does not depend on q

The Argmin and Argmax Operators

Kumar Bipin

The Argmin and Argmax Operators

Let $\mathcal{X}$ denote a set. We define the **argmin** and **argmax** operators as follows:

$$\operatorname*{argmin}_{x \in \mathcal{X}} \ f(x) = \left\{ x \middle| f(x) = \min_{x' \in \mathcal{X}} f(x') \right\}$$

$$\operatorname*{argmax}_{x \in \mathcal{X}} \ f(x) = \left\{ x \middle| f(x) = \max_{x' \in \mathcal{X}} f(x') \right\}$$

Examples:

- $\operatorname*{argmin}_{x \in \mathcal{R}} \ x^2 = 0$
- $\operatorname*{argmax}_{x \in [0, 4\pi]} \ \cos(x) = \{0, 2\pi, 4\pi\}$
- $\operatorname*{argmin}_{x \in [-1, 1]} \ x = -1$
- $\operatorname*{argmin}_{x \in \mathbb{R}} \ 2 = \mathbb{R}$

Example: Maximum Likelihood Estimation

- Let $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be a dataset with samples drawn i.i.d. from p_{data}
- Let the model $p_{model}(y|\mathbf{x}, \mathbf{w})$ be a parametric family of probability distributions
- The conditional **maximum likelihood estimator** for $\mathbf{w}$ is given by

$$\begin{aligned}\hat{\mathbf{w}}_{ML} &= \underset{\mathbf{w}}{\operatorname{argmax}}\ p_{model}(\mathbf{y}|\mathbf{X}, \mathbf{w}) \\ &\overset{\text{iid}}{=} \underset{\mathbf{w}}{\operatorname{argmax}}\ \prod_{i=1}^{N} p_{model}(y_i|\mathbf{x}_i, \mathbf{w}) \\ &= \underset{\mathbf{w}}{\operatorname{argmax}}\ \underbrace{\sum_{i=1}^{N} \log p_{model}(y_i|\mathbf{x}_i, \mathbf{w})}_{\text{Log-Likelihood}}\end{aligned}$$