

Mathematics for Machine Learning

Adapted from the slides of
Dr. Alex Vakanski at U. of Idaho

Notation

- a, b, c Scalar (integer or real)
- $\mathbf{x}, \mathbf{y}, \mathbf{z}$ Vector (bold-font, lower case)
- $\mathbf{A}, \mathbf{B}, \mathbf{C}$ Matrix (bold-font, upper-case)
- $\mathbf{A}, \mathbf{B}, \mathbf{C}$ Tensor ((bold-font, upper-case)
- X, Y, Z Random variable (normal font, upper-case)
- $a \in \mathcal{A}$ Set membership: a is member of set \mathcal{A}
- $|\mathcal{A}|$ Cardinality: number of items in set \mathcal{A}
- $\|\mathbf{v}\|$ Norm of vector \mathbf{v}
- $\mathbf{u} \cdot \mathbf{v}$ or $\langle \mathbf{u}, \mathbf{v} \rangle$ Dot product of vectors \mathbf{u} and \mathbf{v}
- \mathbb{R} Set of real numbers
- \mathbb{R}^n Real numbers space of dimension n
- $y = f(x)$ or $x \mapsto f(x)$ Function (map): assign a unique value $f(x)$ to each input value x
- $f: \mathbb{R}^n \rightarrow \mathbb{R}$ Function (map): map an n -dimensional vector into a scalar

Notation

- $\mathbf{A} \odot \mathbf{B}$ Element-wise product of matrices \mathbf{A} and \mathbf{B}
- \mathbf{A}^\dagger Pseudo-inverse of matrix \mathbf{A}
- $\frac{d^n f}{dx^n}$ n -th derivative of function f with respect to x
- $\nabla_{\mathbf{x}} f(\mathbf{x})$ Gradient of function f with respect to \mathbf{x}
- \mathbf{H}_f Hessian matrix of function f
- $X \sim P$ Random variable X has distribution P
- $P(X|Y)$ Probability of X given Y
- $\mathcal{N}(\mu, \sigma^2)$ Gaussian distribution with mean μ and variance σ^2
- $\mathbb{E}_{X \sim P}[f(X)]$ Expectation of $f(X)$ with respect to $P(X)$
- $\text{Var}(f(X))$ Variance of $f(X)$
- $\text{Cov}(f(X), g(Y))$ Covariance of $f(X)$ and $g(Y)$
- $\text{corr}(X, Y)$ Correlation coefficient for X and Y
- $D_{KL}(P||Q)$ Kullback-Leibler divergence for distributions P and Q
- $CE(P, Q)$ Cross-entropy for distributions P and Q

Vectors

Vectors

- **Vector** definition
 - **Computer science:** *vector* is a one-dimensional array of ordered real-valued scalars
 - **Mathematics:** *vector* is a quantity possessing both magnitude and direction, represented by an arrow indicating the direction, and the length of which is proportional to the magnitude
- Vectors are written in column form or in row form
 - Denoted by bold-font lower-case letters

$$\mathbf{x} = \begin{bmatrix} 1 \\ 7 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{x} = [1 \quad 7 \quad 0 \quad 1]^T$$

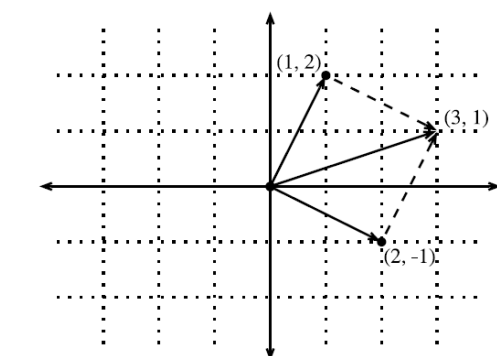
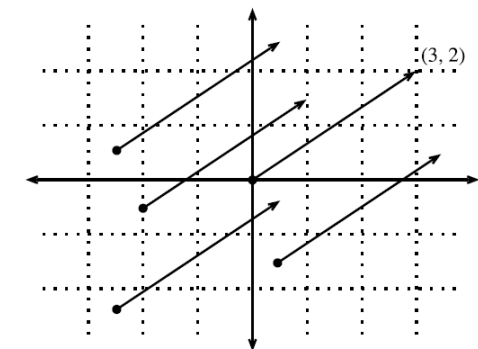
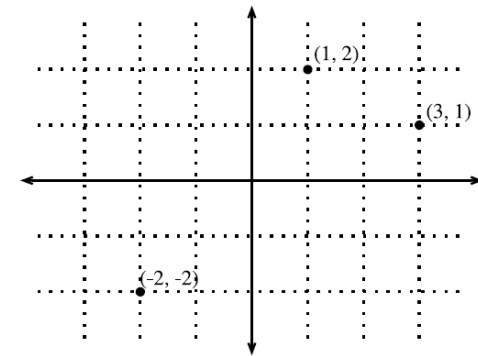
- For a general form vector with n elements, the vector lies in the n -dimensional space $\mathbf{x} \in \mathbb{R}^n$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Geometry of Vectors

Vectors

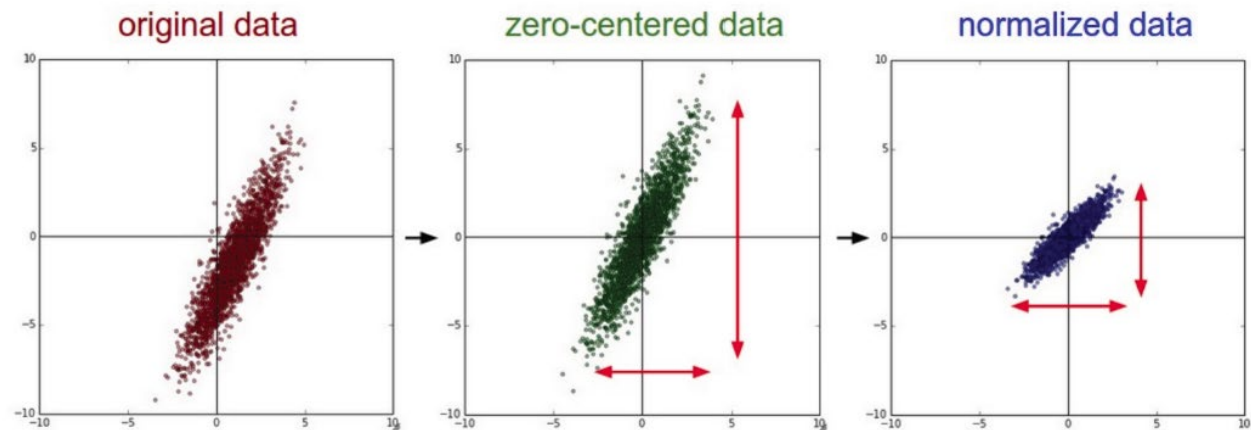
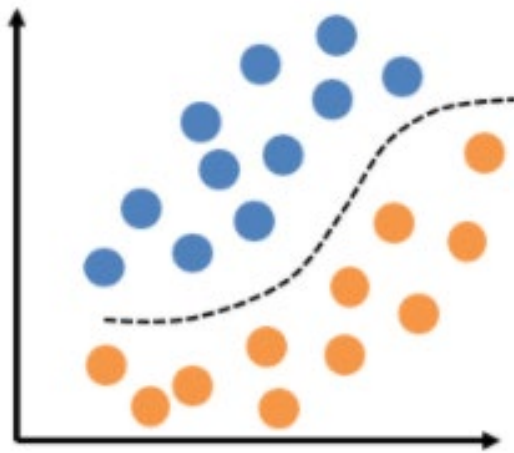
- First interpretation of a vector: **point in space**
 - E.g., in 2D we can visualize the data points with respect to a coordinate origin
- Second interpretation of a vector: **direction in space**
 - E.g., the vector $\vec{v} = [3, 2]^T$ has a direction of 3 steps to the right and 2 steps up
 - The notation \vec{v} is sometimes used to indicate that the vectors have a direction
 - All vectors in the figure have the same direction
- Vector **addition**
 - We add the coordinates, and follow the directions given by the two vectors that are added



Geometry of Vectors

Vectors

- The geometric interpretation of vectors as points in space allow us to consider a training set of input examples in ML as a **collection of points in space**
 - Hence, classification can be viewed as discovering how to separate two clusters of points belonging to different classes (left picture)
 - Rather than distinguishing images containing cars, planes, buildings, for example
 - Or, it can help to visualize zero-centering and normalization of training data (right picture)

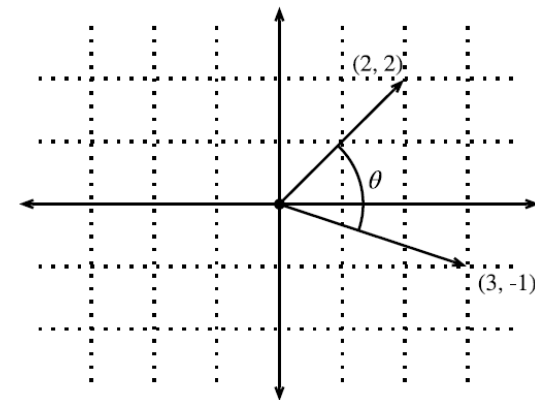


Dot Product and Angles

Vectors

- **Dot product** of vectors, $\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} = \sum_i u_i \cdot v_i$
 - It is also referred to as **inner product**, or **scalar product** of vectors
 - The dot product $\mathbf{u} \cdot \mathbf{v}$ is also often denoted by $\langle \mathbf{u}, \mathbf{v} \rangle$
- The dot product is a symmetric operation, $\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} = \mathbf{v}^T \mathbf{u} = \mathbf{v} \cdot \mathbf{u}$
- Geometric interpretation of a dot product: **angle** between two vectors
 - I.e., dot product $\mathbf{v} \cdot \mathbf{w}$ over the norms of the vectors is $\cos(\theta)$

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos(\theta) \quad \cos\theta = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$$



- If two vectors are orthogonal: $\theta = 90^\circ$, i.e., $\cos(\theta) = 0$, then $\mathbf{u} \cdot \mathbf{v} = 0$
- Also, in ML the term $\cos\theta = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$ is sometimes employed as a measure of closeness of two vectors/data instances, and it is referred to as **cosine similarity**

Norm of a Vector

Vectors

- A vector **norm** is a function that maps a vector to a scalar value
 - The norm is a measure of the size of the vector
- The norm f should satisfy the following properties:
 - Scaling: $f(\alpha \mathbf{x}) = |\alpha|f(\mathbf{x})$
 - Triangle inequality: $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$
 - Must be non-negative: $f(\mathbf{x}) \geq 0$
- The general ℓ_p norm of a vector \mathbf{x} is obtained as: $\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$
 - On next page we will review the most common norms, obtained for $p = 1, 2$, and ∞

Norm of a Vector

Vectors

- For $p = 2$, we have ℓ_2 norm

- Also called **Euclidean norm**
- It is the most often used norm

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2} = \sqrt{\mathbf{x}^T \mathbf{x}}$$

- ℓ_2 norm is often denoted just as $\|\mathbf{x}\|$ with the subscript 2 omitted

- For $p = 1$, we have ℓ_1 norm

- Uses the absolute values of the elements
- Discriminate between zero and non-zero elements

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

- For $p = \infty$, we have ℓ_∞ norm

- Known as **infinity norm**, or **max norm**
- Outputs the absolute value of the largest element

$$\|\mathbf{x}\|_\infty = \max_i |x_i|$$

- ℓ_0 norm outputs the number of non-zero elements

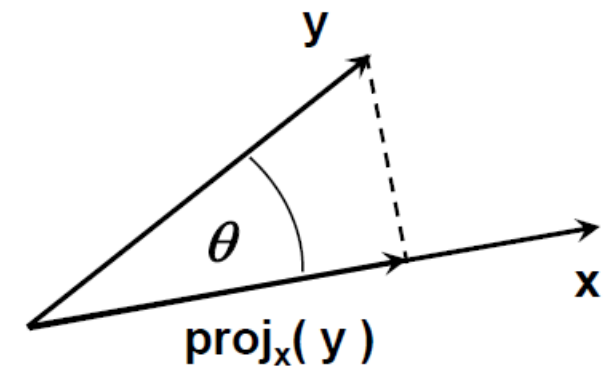
- It is not an ℓ_p norm, and it is not really a norm function either (it is incorrectly called a norm)

Vector Projection

Vectors

- **Orthogonal projection** of a vector \mathbf{y} onto vector \mathbf{x}
 - The projection can take place in any space of dimensionality ≥ 2
 - The **unit vector** in the direction of \mathbf{x} is $\frac{\mathbf{x}}{\|\mathbf{x}\|}$
 - A unit vector has norm equal to 1
 - The length of the projection of \mathbf{y} onto \mathbf{x} is $\|\mathbf{y}\| \cdot \cos(\theta)$
 - The orthogonal project is the vector $\mathbf{proj}_{\mathbf{x}}(\mathbf{y})$

$$\mathbf{proj}_{\mathbf{x}}(\mathbf{y}) = \frac{\mathbf{x} \cdot \|\mathbf{y}\| \cdot \cos(\theta)}{\|\mathbf{x}\|}$$

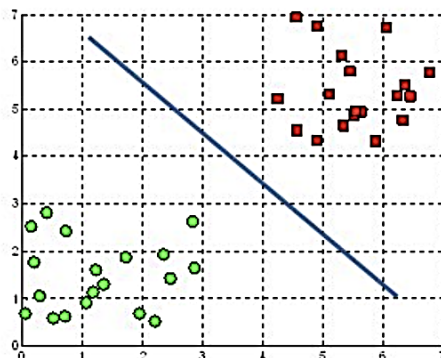


Hyperplanes

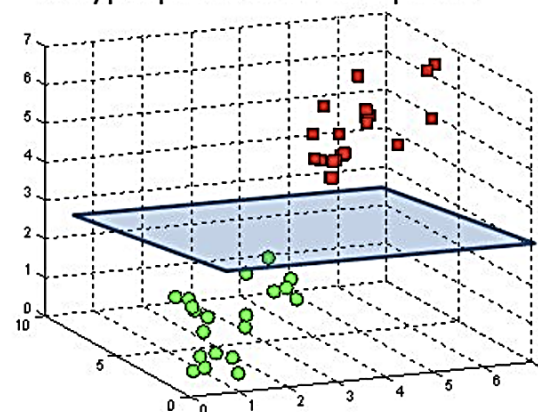
Hyperplanes

- **Hyperplane** is a subspace whose dimension is one less than that of its ambient space
 - In a 2D space, a hyperplane is a straight line (i.e., 1D)
 - In a 3D, a hyperplane is a plane (i.e., 2D)
 - In a d -dimensional vector space, a hyperplane has $d - 1$ dimensions, and divides the space into two half-spaces
- Hyperplane is a generalization of a concept of plane in high-dimensional space
- In ML, hyperplanes are **decision boundaries** used for linear classification
 - Data points falling on either sides of the hyperplane are attributed to different classes

A hyperplane in \mathbb{R}^2 is a line



A hyperplane in \mathbb{R}^3 is a plane



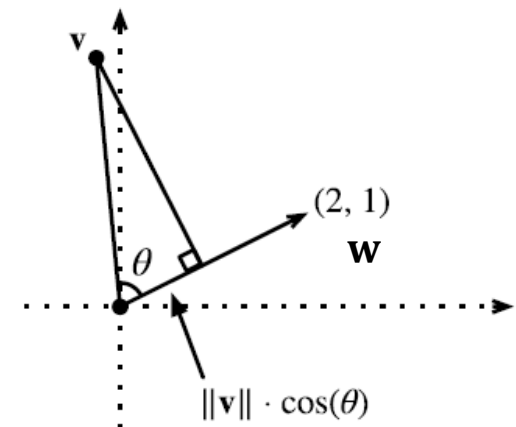
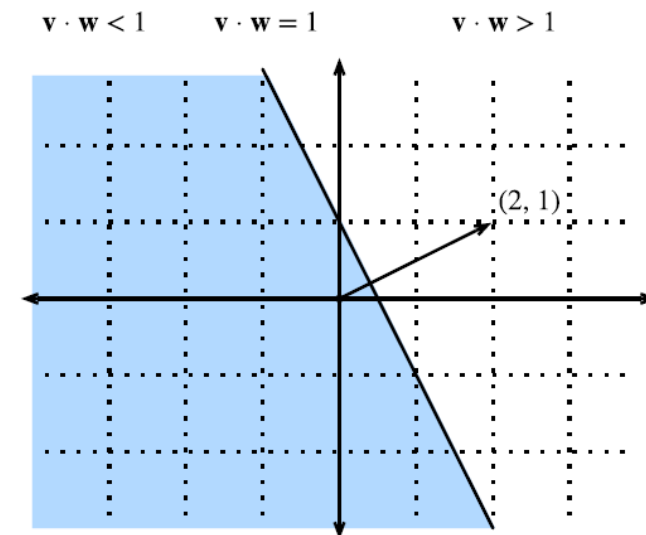
Hyperplanes

Hyperplanes

- For example, for a given data point $\mathbf{w} = [2, 1]^T$, we can use dot-product to find the hyperplane for which $\mathbf{w} \cdot \mathbf{v} = 1$
 - I.e., all vectors with $\mathbf{w} \cdot \mathbf{v} > 1$ can be classified as one class, and all vectors with $\mathbf{w} \cdot \mathbf{v} < 1$ can be classified as another class
- Solving $\mathbf{w} \cdot \mathbf{v} = 1$, we obtain

$$\|\mathbf{v}\| \|\mathbf{w}\| \cos(\theta) = 1 \iff \|\mathbf{v}\| \cos(\theta) = \frac{1}{\|\mathbf{w}\|} = \frac{1}{\sqrt{5}}$$

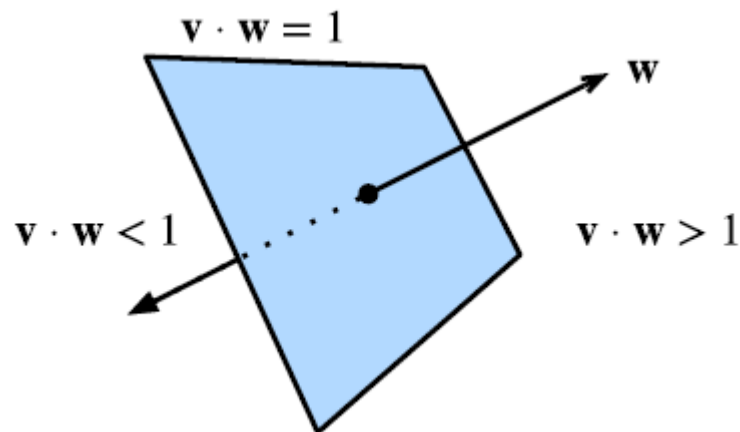
- I.e., the solution is the set of points for which $\mathbf{w} \cdot \mathbf{v} = 1$ meaning the points lay on the line that is orthogonal to the vector \mathbf{w}
 - That is the line $2x + y = 1$
- The orthogonal projection of \mathbf{v} onto \mathbf{w} is $\|\mathbf{v}\| \cos(\theta) = \frac{1}{\sqrt{5}}$



Hyperplanes

Hyperplanes

- In a 3D space, if we have a vector $\mathbf{w} = [1, 2, 3]^T$ and try to find all points that satisfy $\mathbf{w} \cdot \mathbf{v} = 1$, we can obtain a plane that is orthogonal to the vector \mathbf{w}
 - The inequalities $\mathbf{w} \cdot \mathbf{v} > 1$ and $\mathbf{w} \cdot \mathbf{v} < 1$ again define the two subspaces that are created by the plane



- The same concept applies to high-dimensional spaces as well

Matrices

Matrices

- **Matrix** is a rectangular array of real-valued scalars arranged in m horizontal rows and n vertical columns
 - Each element a_{ij} belongs to the i^{th} row and j^{th} column
 - The elements are denoted a_{ij} or \mathbf{A}_{ij} or $[\mathbf{A}]_{ij}$ or $\mathbf{A}(i, j)$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

- For the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, the size (dimension) is $m \times n$ or (m, n)
 - Matrices are denoted by bold-font upper-case letters

Matrices

Matrices

- Addition or subtraction $(\mathbf{A} \pm \mathbf{B})_{i,j} = \mathbf{A}_{i,j} \pm \mathbf{B}_{i,j}$

$$\begin{bmatrix} 1 & 3 & 1 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \end{bmatrix} = \begin{bmatrix} 1+0 & 3+0 & 1+5 \\ 1+7 & 0+5 & 0+0 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 6 \\ 8 & 5 & 0 \end{bmatrix}$$

- Scalar multiplication $(c\mathbf{A})_{i,j} = c \cdot \mathbf{A}_{i,j}$

$$2 \cdot \begin{bmatrix} 1 & 8 & -3 \\ 4 & -2 & 5 \end{bmatrix} = \begin{bmatrix} 2 \cdot 1 & 2 \cdot 8 & 2 \cdot -3 \\ 2 \cdot 4 & 2 \cdot -2 & 2 \cdot 5 \end{bmatrix} = \begin{bmatrix} 2 & 16 & -6 \\ 8 & -4 & 10 \end{bmatrix}$$

- Matrix multiplication $(\mathbf{AB})_{i,j} = \mathbf{A}_{i,1}\mathbf{B}_{1,j} + \mathbf{A}_{i,2}\mathbf{B}_{2,j} + \dots + \mathbf{A}_{i,n}\mathbf{B}_{n,j}$

- Defined only if the number of columns of the left matrix is the same as the number of rows of the right matrix
- Note that $\mathbf{AB} \neq \mathbf{BA}$

$$\begin{bmatrix} \underline{2} & \underline{3} & \underline{4} \\ \underline{1} & \underline{0} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{0} & \underline{1000} \\ \underline{1} & \underline{100} \\ \underline{0} & \underline{10} \end{bmatrix} = \begin{bmatrix} \underline{3} & \underline{2340} \\ \underline{0} & \underline{1000} \end{bmatrix}$$

Matrices

Matrices

- **Transpose** of the matrix: \mathbf{A}^T has the rows and columns exchanged

$$(\mathbf{A}^T)_{i,j} = \mathbf{A}_{j,i} \quad \begin{bmatrix} 1 & 2 & 3 \\ 0 & -6 & 7 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 \\ 2 & -6 \\ 3 & 7 \end{bmatrix}$$

- Some properties

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

- **Square matrix**: has the same number of rows and columns
- **Identity matrix** (\mathbf{I}_n): has ones on the main diagonal, and zeros elsewhere

- E.g.: identity matrix of size 3×3 : $\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Matrices

Matrices

- **Determinant** of a matrix, denoted by $\det(\mathbf{A})$ or $|\mathbf{A}|$, is a real-valued scalar encoding certain properties of the matrix

- E.g., for a matrix of size 2×2 :

$$\det \left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) = ad - bc$$

- For larger-size matrices the determinant of a matrix is calculated as

$$\det(\mathbf{A}) = \sum_j a_{ij} (-1)^{i+j} \det(\mathbf{A}_{(i,j)})$$

- In the above, $\mathbf{A}_{(i,j)}$ is a **minor** of the matrix obtained by removing the row and column associated with the indices i and j

- **Trace** of a matrix is the sum of all diagonal elements

$$\text{Tr}(\mathbf{A}) = \sum_i a_{ii}$$

- A matrix for which $\mathbf{A} = \mathbf{A}^T$ is called a **symmetric matrix**

Matrices

Matrices

- Elementwise multiplication of two matrices **A** and **B** is called the *Hadamard product* or *elementwise product*
 - The math notation is \odot

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \dots & a_{1n}b_{1n} \\ a_{21}b_{21} & a_{22}b_{22} & \dots & a_{2n}b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{m1} & a_{m2}b_{m2} & \dots & a_{mn}b_{mn} \end{bmatrix}$$

Matrix-Vector Products

Matrices

- Consider a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and a vector $\mathbf{x} \in \mathbb{R}^n$
- The matrix can be written in terms of its row vectors (e.g., \mathbf{a}_1^T is the first row)

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_m^T \end{bmatrix}$$

- The **matrix-vector** product is a column vector of length m , whose i^{th} element is the dot product $\mathbf{a}_i^T \mathbf{x}$

$$\mathbf{Ax} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_m^T \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{a}_1^T \mathbf{x} \\ \mathbf{a}_2^T \mathbf{x} \\ \vdots \\ \mathbf{a}_m^T \mathbf{x} \end{bmatrix}$$

- Note the size: $\mathbf{A}(m \times n) \cdot \mathbf{x}(n \times 1) = \mathbf{Ax}(m \times 1)$

Matrix-Matrix Products

Matrices

- To multiply two matrices $\mathbf{A} \in \mathbb{R}^{n \times k}$ and $\mathbf{B} \in \mathbb{R}^{k \times m}$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nk} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{km} \end{bmatrix}$$

- We can consider the **matrix-matrix product** as dot-products of rows in \mathbf{A} and columns in \mathbf{B}

$$\mathbf{C} = \mathbf{AB} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_n^\top \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_m \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^\top \mathbf{b}_1 & \mathbf{a}_1^\top \mathbf{b}_2 & \cdots & \mathbf{a}_1^\top \mathbf{b}_m \\ \mathbf{a}_2^\top \mathbf{b}_1 & \mathbf{a}_2^\top \mathbf{b}_2 & \cdots & \mathbf{a}_2^\top \mathbf{b}_m \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_n^\top \mathbf{b}_1 & \mathbf{a}_n^\top \mathbf{b}_2 & \cdots & \mathbf{a}_n^\top \mathbf{b}_m \end{bmatrix}$$

- Size: $\mathbf{A}(n \times k) \cdot \mathbf{B}(k \times m) = \mathbf{C}(n \times m)$

Inverse of a Matrix

Matrices

- For a square $n \times n$ matrix \mathbf{A} with rank n , \mathbf{A}^{-1} is its *inverse matrix* if their product is an identity matrix \mathbf{I}

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

- Properties of inverse matrices $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$
 $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$
- If $\det(A) = 0$ (i.e., $\text{rank}(A) < n$), then the inverse does not exist
 - A matrix that is not invertible is called a **singular matrix**
- Note that finding an inverse of a large matrix is computationally expensive
 - In addition, it can lead to numerical instability
- If the inverse of a matrix is equal to its transpose, the matrix is said to be **orthogonal matrix**

$$\mathbf{A}^{-1} = \mathbf{A}^T$$

Pseudo-Inverse of a Matrix

Matrices

- *Pseudo-inverse* of a matrix
 - Also known as **Moore-Penrose pseudo-inverse**
- For matrices that are not square, the inverse does not exist
 - Therefore, a pseudo-inverse is used
- If $m > n$, then the pseudo-inverse is $\mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ and $\mathbf{A}^\dagger \mathbf{A} = \mathbf{I}$
- If $m < n$, then the pseudo-inverse is $\mathbf{A}^\dagger = \mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1}$ and $\mathbf{A} \mathbf{A}^\dagger = \mathbf{I}$
 - E.g., for a matrix with dimension $\mathbf{X}_{2 \times 3}$, a pseudo-inverse can be found of size $\mathbf{X}_{3 \times 2}^\dagger$, so that $\mathbf{X}_{2 \times 3} \mathbf{X}_{3 \times 2}^\dagger = \mathbf{I}_{2 \times 2}$

Tensors

Tensors

- **Tensors** are n -dimensional arrays of scalars
 - Vectors are first-order tensors, $\mathbf{v} \in \mathbb{R}^n$
 - Matrices are second-order tensors, $\mathbf{A} \in \mathbb{R}^{m \times n}$
 - E.g., a fourth-order tensor is $\mathbf{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$
- Tensors are denoted with upper-case letters of a special font face (e.g., **X**, **Y**, **Z**)
- RGB images are third-order tensors, i.e., as they are 3-dimensional arrays
 - The 3 axes correspond to width, height, and channel
 - E.g., $224 \times 224 \times 3$
 - The channel axis corresponds to the color channels (red, green, and blue)

Manifolds

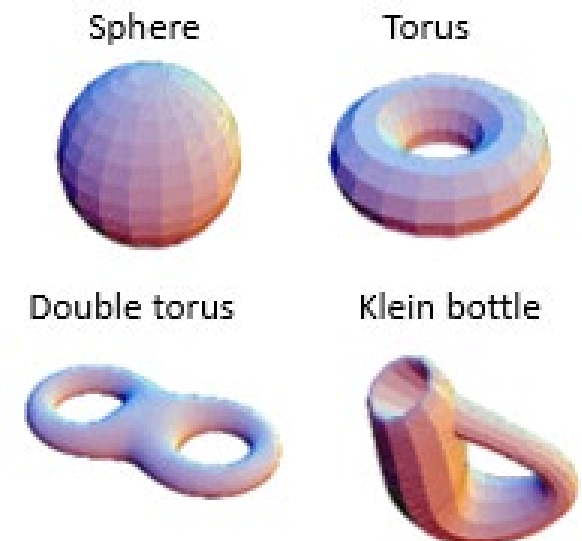
Manifolds

- Earlier we learned that hyperplanes generalize the concept of planes in high-dimensional spaces
 - Similarly, manifolds can be informally imagined as generalization of the concept of surfaces in high-dimensional spaces
- To begin with an intuitive explanation, the surface of the Earth is an example of a two-dimensional manifold embedded in a three-dimensional space
 - This is true because the Earth looks locally flat, so on a small scale it is like a 2-D plane
 - However, if we keep walking on the Earth in one direction, we will eventually end up back where we started
 - This means that Earth is not really flat, it only looks **locally** like a Euclidean plane, but at large scales it **folds up** on itself, and has a different **global** structure than a flat plane

Manifolds

Manifolds

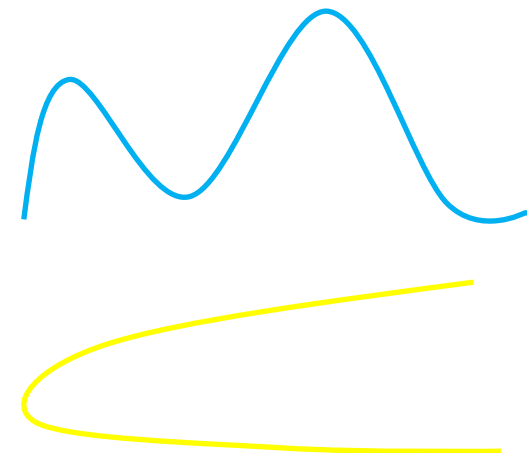
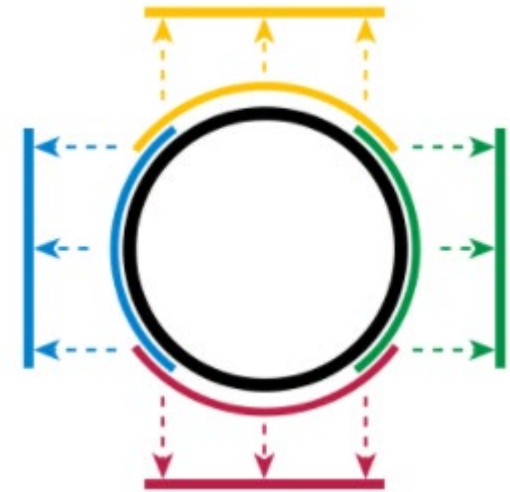
- Manifolds are studied in mathematics under topological spaces
- An n -dimensional *manifold* is defined as a topological space with the property that each point has a neighborhood that is homeomorphic to the Euclidean space of dimension n
 - This means that a manifold locally resembles Euclidean space near each point
 - Informally, a Euclidean space is locally smooth, it does not have holes, edges, or other sudden changes, and it does not have intersecting neighborhoods
 - Although the manifolds can have very complex structure on a large scale, resemblance of the Euclidean space on a small scale allows to apply standard math concepts
- Examples of 2-dimensional manifolds are shown in the figure
 - The surfaces in the figure have been conveniently cut up into little rectangles that were glued together
 - Those small rectangles locally look like flat Euclidean planes



Manifolds

Manifolds

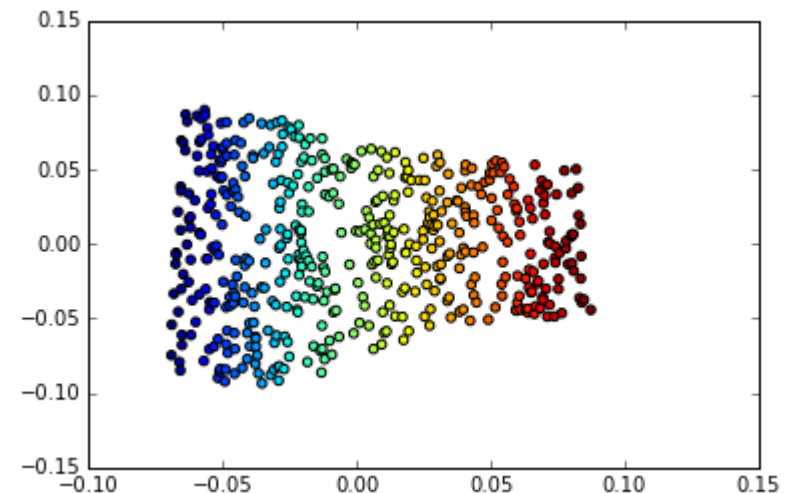
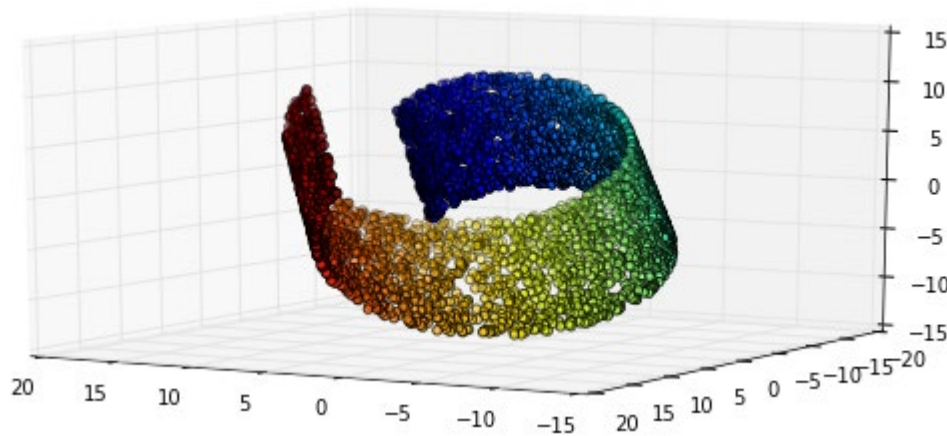
- Examples of one-dimensional manifolds
 - Upper figure: a circle is a 1-D manifold embedded in 2-D, where each arc of the circle locally resembles a line segment
 - Lower figures: other examples of 1-D manifolds
 - Note that a number 8 figure is not a manifold because it has an intersecting point (it is not Euclidean locally)
- It is hypothesized that in the real-world, high-dimensional data (such as images) lie on low-dimensional manifolds embedded in the high-dimensional space
 - E.g., in ML, let's assume we have a training set of images with size $224 \times 224 \times 3$ pixels
 - Learning an arbitrary function in such high-dimensional space would be intractable
 - Despite that, all images of the same class ("cats" for example) might lie on a low-dimensional manifold
 - This allows function learning and image classification



Manifolds

Manifolds

- Example:
 - The data points have 3 dimensions (left figure), i.e., the input space of the data is 3-dimensional
 - The data points lie on a 2-dimensional manifold, shown in the right figure
 - Most ML algorithms extract lower-dimensional data features that enable to distinguish between various classes of high-dimensional input data
 - The low-dimensional representations of the input data are called **embeddings**



Eigen Decomposition

Eigen Decomposition

- ***Eigen decomposition*** is decomposing a matrix into a set of eigenvalues and eigenvectors
- ***Eigenvalues*** of a square matrix \mathbf{A} are scalars λ and ***eigenvectors*** are non-zero vectors \mathbf{v} that satisfy

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

- Eigenvalues are found by solving the following equation

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

- If a matrix \mathbf{A} has n linearly independent eigenvectors $\{\mathbf{v}^1, \dots, \mathbf{v}^n\}$ with corresponding eigenvalues $\{\lambda_1, \dots, \lambda_n\}$, the eigen decomposition of \mathbf{A} is given by

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

- Columns of the matrix \mathbf{V} are the eigenvectors, i.e., $\mathbf{V} = [\mathbf{v}^1, \dots, \mathbf{v}^n]$
- $\mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues, i.e., $\mathbf{\Lambda} = [\lambda_1, \dots, \lambda_n]$
- To find the inverse of the matrix \mathbf{A} , we can use $\mathbf{A}^{-1} = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}^{-1}$
 - This involves simply finding the inverse $\mathbf{\Lambda}^{-1}$ of a diagonal matrix

Eigen Decomposition

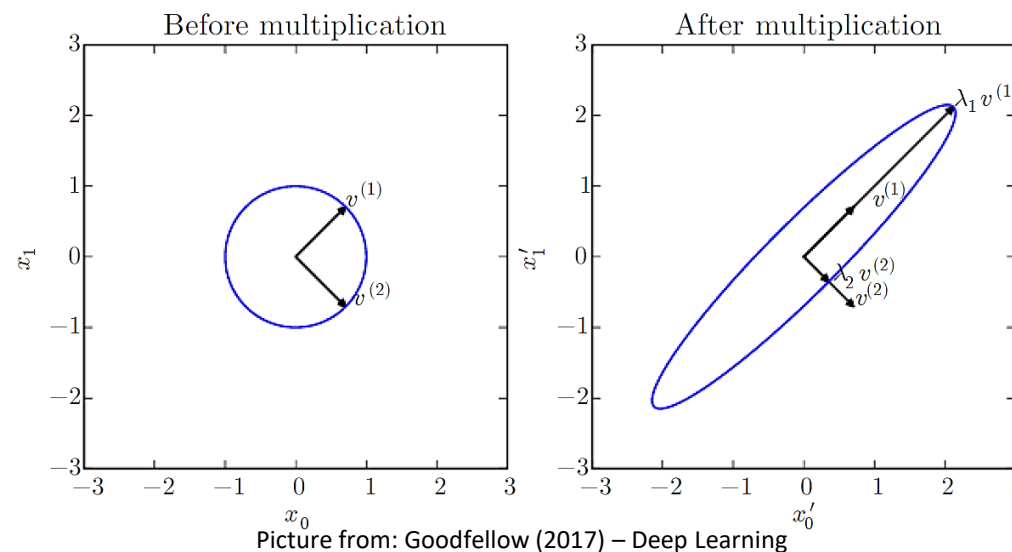
Eigen Decomposition

- Decomposing a matrix into eigenvalues and eigenvectors allows to analyze certain properties of the matrix
 - If all eigenvalues are positive, the matrix is **positive definite**
 - If all eigenvalues are positive or zero-valued, the matrix is **positive semidefinite**
 - If all eigenvalues are negative or zero-values, the matrix is **negative semidefinite**
 - Positive semidefinite matrices are interesting because they guarantee that $\forall \mathbf{x}, \mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$
- Eigen decomposition can also simplify many linear-algebraic computations
 - The determinant of \mathbf{A} can be calculated as
$$\det(\mathbf{A}) = \lambda_1 \cdot \lambda_2 \cdots \lambda_n$$
 - If any of the eigenvalues are zero, the matrix is singular (it does not have an inverse)
- However, not every matrix can be decomposed into eigenvalues and eigenvectors
 - Also, in some cases the decomposition may involve complex numbers
 - Still, every real symmetric matrix is guaranteed to have an eigen decomposition according to $\mathbf{A} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^{-1}$, where \mathbf{V} is an orthogonal matrix

Eigen Decomposition

Eigen Decomposition

- Geometric interpretation of the eigenvalues and eigenvectors is that they allow to stretch the space in specific directions
 - Left figure: the two eigenvectors \mathbf{v}^1 and \mathbf{v}^2 are shown for a matrix, where the two vectors are unit vectors (i.e., they have a length of 1)
 - Right figure: the vectors \mathbf{v}^1 and \mathbf{v}^2 are multiplied with the eigenvalues λ_1 and λ_2
 - We can see how the space is scaled in the direction of the larger eigenvalue λ_1
- E.g., this is used for dimensionality reduction with PCA (principal component analysis) where the eigenvectors corresponding to the largest eigenvalues are used for extracting the most important data dimensions



Differential Calculus

Differential Calculus

- For a function $f: \mathbb{R} \rightarrow \mathbb{R}$, the **derivative** of f is defined as

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

- If $f'(a)$ exists, f is said to be **differentiable** at a
- If $f'(c)$ is differentiable for $\forall c \in [a, b]$, then f is differentiable on this interval
 - We can also interpret the derivative $f'(x)$ as the **instantaneous rate of change** of $f(x)$ with respect to x
 - I.e., for a small change in x , what is the rate of change of $f(x)$
- Given $y = f(x)$, where x is an independent variable and y is a dependent variable, the following expressions are equivalent:

$$f'(x) = f' = \frac{dy}{dx} = \frac{df}{dx} = \frac{d}{dx} f(x) = Df(x) = D_x f(x)$$

- The symbols $\frac{d}{dx}$, D , and D_x are **differentiation operators** that indicate operation of **differentiation**

Differential Calculus

Differential Calculus

- The following rules are used for computing the derivatives of explicit functions
 - **Derivative of constants.** $\frac{d}{dx}c = 0.$
 - **Derivative of linear functions.** $\frac{d}{dx}(ax) = a.$
 - **Power rule.** $\frac{d}{dx}x^n = nx^{n-1}.$
 - **Derivative of exponentials.** $\frac{d}{dx}e^x = e^x.$
 - **Derivative of the logarithm.** $\frac{d}{dx}\log(x) = \frac{1}{x}.$
 - **Sum rule.** $\frac{d}{dx}(g(x) + h(x)) = \frac{dg}{dx}(x) + \frac{dh}{dx}(x).$
 - **Product rule.** $\frac{d}{dx}(g(x) \cdot h(x)) = g(x)\frac{dh}{dx}(x) + \frac{dg}{dx}(x)h(x).$
 - **Chain rule.** $\frac{d}{dx}g(h(x)) = \frac{dg}{dh}(h(x)) \cdot \frac{dh}{dx}(x).$

Higher Order Derivatives

Differential Calculus

- The derivative of the first derivative of a function $f(x)$ is the *second derivative* of $f(x)$

$$\frac{d^2 f}{dx^2} = \frac{d}{dx} \left(\frac{df}{dx} \right)$$

- The second derivative quantifies how the rate of change of $f(x)$ is changing
 - E.g., in physics, if the function describes the displacement of an object, the first derivative gives the velocity of the object (i.e., the rate of change of the position)
 - The second derivative gives the acceleration of the object (i.e., the rate of change of the velocity)
- If we apply the differentiation operation any number of times, we obtain the *n -th derivative* of $f(x)$

$$f^{(n)}(x) = \frac{d^n f}{dx^n} = \left(\frac{d}{dx} \right)^n f(x)$$

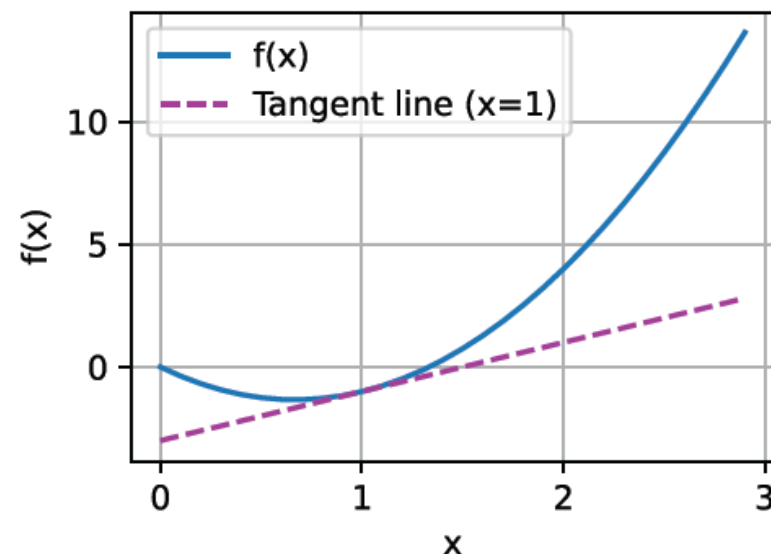
Geometric Interpretation

Differential Calculus

- To provide a geometric interpretation of the derivatives, let's consider a first-order Taylor series approximation of $f(x)$ at $x = x_0$

$$f(x) \approx f(x_0) + \left. \frac{df}{dx} \right|_{x_0} (x - x_0)$$

- The expression approximates the function $f(x)$ by a line which passes through the point $(x_0, f(x_0))$ and has slope $\left. \frac{df}{dx} \right|_{x_0}$ (i.e., the value of $\frac{df}{dx}$ at the point x_0)
- Therefore, the first derivative of a function is also the **slope of the tangent line** to the curve of the function



Partial Derivatives

Differential Calculus

- So far, we looked at functions of a single variable, where $f: \mathbb{R} \rightarrow \mathbb{R}$
- Functions that depend on many variables are called **multivariate functions**
- Let $y = f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$ be a multivariate function with n variables
 - The input is an n -dimensional vector $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ and the output is a scalar y
 - The mapping is $f: \mathbb{R}^n \rightarrow \mathbb{R}$
- The **partial derivative** of y with respect to its i^{th} parameter x_i is

$$\frac{\partial y}{\partial x_i} = \lim_{h \rightarrow 0} \frac{f(x_1, x_2, \dots, x_i + h, \dots, x_n) - f(x_1, x_2, \dots, x_i, \dots, x_n)}{h}$$

- To calculate $\frac{\partial y}{\partial x_i}$ (∂ pronounced “del” or we can just say “partial derivative”), we can treat $x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ as constants and calculate the derivative of y only with respect to x_i
- For notation of partial derivatives, the following are equivalent:

$$\frac{\partial y}{\partial x_i} = \frac{\partial f}{\partial x_i} = \frac{\partial}{\partial x_i} f(\mathbf{x}) = f_{x_i} = f_i = D_i f = D_{x_i} f$$

Gradient

Differential Calculus

- We can concatenate partial derivatives of a multivariate function with respect to all its input variables to obtain the *gradient* vector of the function
- The gradient of the multivariate function $f(\mathbf{x})$ with respect to the n -dimensional input vector $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, is a vector of n partial derivatives

$$\nabla_{\mathbf{x}}f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right]^T$$

- When there is no ambiguity, the notations $\nabla f(\mathbf{x})$ or $\nabla_{\mathbf{x}}f$ are often used for the gradient instead of $\nabla_{\mathbf{x}}f(\mathbf{x})$
 - The symbol for the gradient is the Greek letter ∇ (pronounced “nabla”), although $\nabla_{\mathbf{x}}f(\mathbf{x})$ is more often it is pronounced “gradient of f with respect to \mathbf{x} ”
- In ML, the gradient descent algorithm relies on the opposite direction of the gradient of the loss function \mathcal{L} with respect to the model parameters θ ($\nabla_{\theta}\mathcal{L}$) for minimizing the loss function
 - Adversarial examples can be created by adding perturbation in the direction of the gradient of the loss \mathcal{L} with respect to input examples x ($\nabla_x\mathcal{L}$) for maximizing the loss function

Optimization

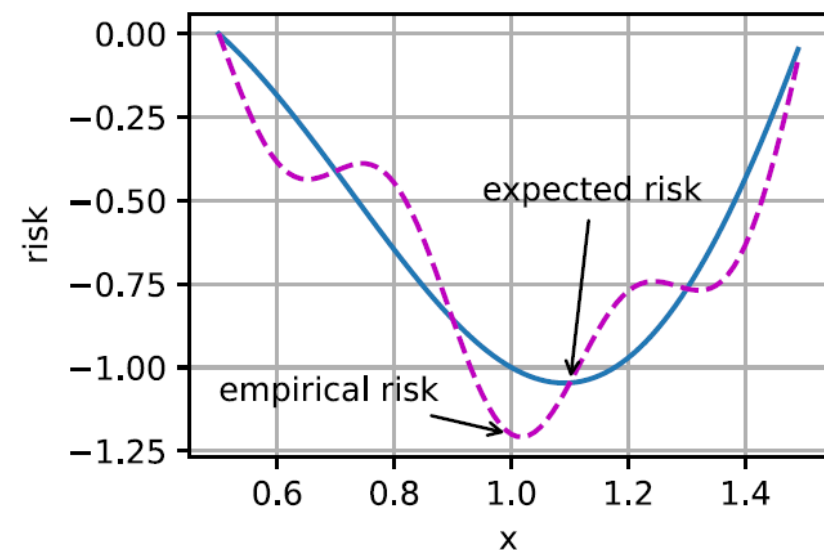
Optimization

- **Optimization** is concerned with optimizing an **objective function** — finding the value of an argument that minimizes or maximizes the function
 - Most optimization algorithms are formulated in terms of minimizing a function $f(x)$
 - Maximization is accomplished via minimizing the negative of an objective function (e.g., minimize $-f(x)$)
 - In minimization problems, the objective function is often referred to as a **cost function** or **loss function** or **error function**
- Optimization is very important for machine learning
 - The performance of optimization algorithms affect the model's training efficiency
- Most optimization problems in machine learning are **nonconvex**
 - Meaning that the loss function is not a convex function
 - Nonetheless, the design and analysis of algorithms for solving convex problems has been very instructive for advancing the field of machine learning

Optimization

Optimization

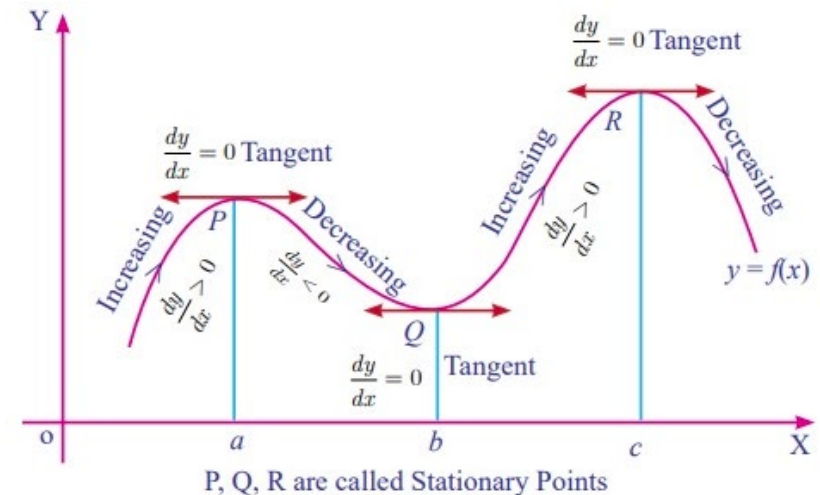
- Optimization and machine learning have related, but somewhat different goals
 - Goal in optimization: minimize an objective function
 - For a set of training examples, reduce the **training error**
 - Goal in ML: find a suitable model, to predict on data examples
 - For a set of testing examples, reduce the **generalization error**
- For a given empirical function g (dashed purple curve), optimization algorithms attempt to find the point of minimum **empirical risk**
- The expected function f (blue curve) is obtained given a limited amount of training data examples
- ML algorithms attempt to find the point of minimum **expected risk**, based on minimizing the error on a set of testing examples
 - Which may be at a different location than the minimum of the training examples
 - And which may not be minimal in a formal sense



Stationary Points

Optimization

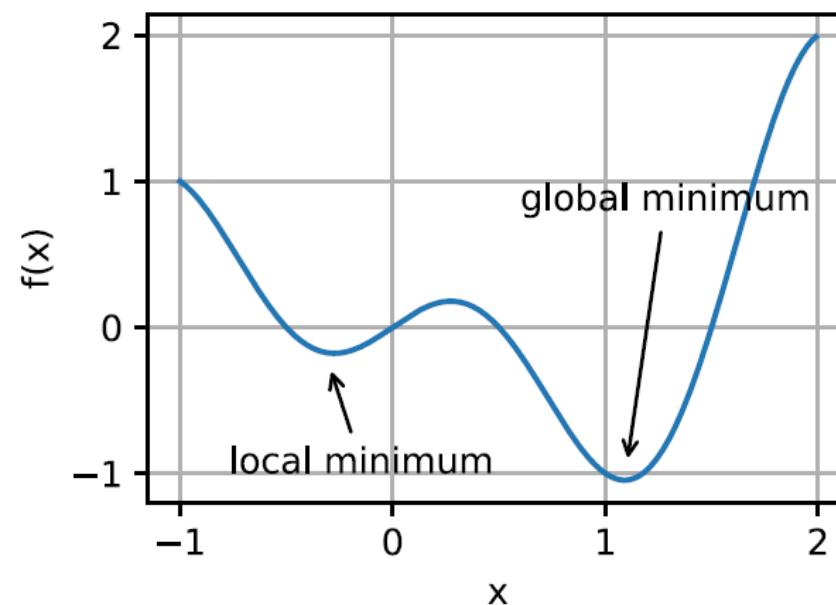
- **Stationary points** (or **critical points**) of a differentiable function $f(x)$ of one variable are the points where the derivative of the function is zero, i.e., $f'(x) = 0$
- The stationary points can be:
 - **Minimum**, a point where the derivative changes from negative to positive
 - **Maximum**, a point where the derivative changes from positive to negative
 - **Saddle point**, derivative is either positive or negative on both sides of the point
- The minimum and maximum points are collectively known as **extremum points**
- The nature of stationary points can be determined based on the second derivative of $f(x)$ at the point
 - If $f''(x) > 0$, the point is a minimum
 - If $f''(x) < 0$, the point is a maximum
 - If $f''(x) = 0$, inconclusive, the point can be a saddle point, but it may not
- The same concept also applies to gradients of multivariate functions



Local Minima

Optimization

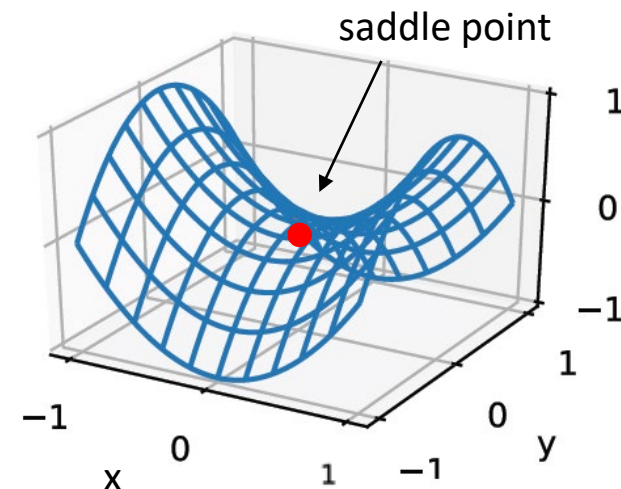
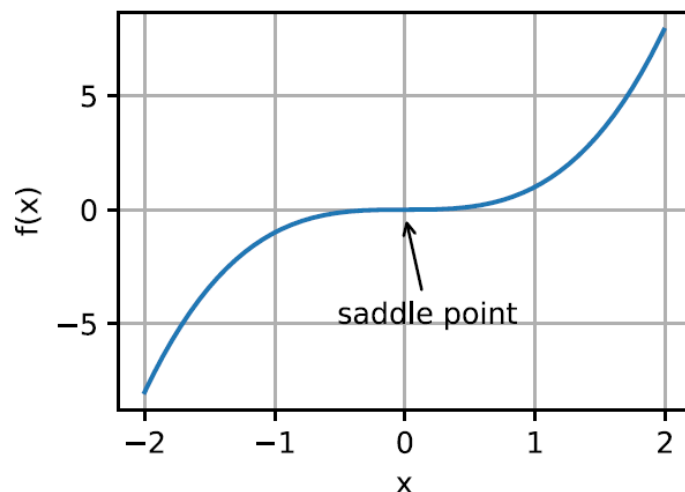
- Among the challenges in optimization of model's parameters in ML involve local minima, saddle points, vanishing gradients
- For an objective function $f(x)$, if the value at a point x is the minimum of the objective function **over the entire domain** of x , then it is the *global minimum*
- If the value of $f(x)$ at x is smaller than the values of the objective function at any other points in **the vicinity** of x , then it is the *local minimum*
- The objective functions in ML usually have many local minima
 - When the solution of the optimization algorithm is near the local minimum, the gradient of the loss function approaches or becomes zero (vanishing gradients)
 - Therefore, the obtained solution in the final iteration can be a local minimum, rather than the global minimum



Saddle Points

Optimization

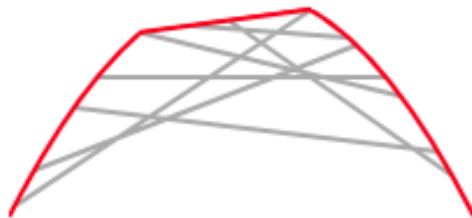
- The gradient of a function $f(x)$ at a **saddle point** is 0, but the point is not a minimum or maximum point
 - The optimization algorithms may stall at saddle points, without reaching a minima
- Note also that the point of a function at which the sign of the curvature changes is called an **inflection point**
 - An inflection point ($f''(x) = 0$) can also be a saddle point, but it does not have to be
- For the 2D function (right figure), the saddle point is at $(0,0)$
 - The point looks like a saddle, and gives the minimum with respect to x , and the maximum with respect to y



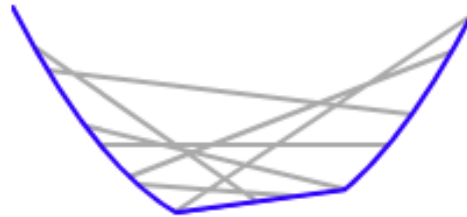
Convex Optimization

Optimization

- A function of a single variable is *concave* if every line segment joining two points on its graph does not lie above the graph at any point
- Symmetrically, a function of a single variable is *convex* if every line segment joining two points on its graph does not lie below the graph at any point



A concave function:
no line segment joining
two points on the graph
lies above the graph
at any point



A convex function:
no line segment joining
two points on the graph
lies below the graph
at any point



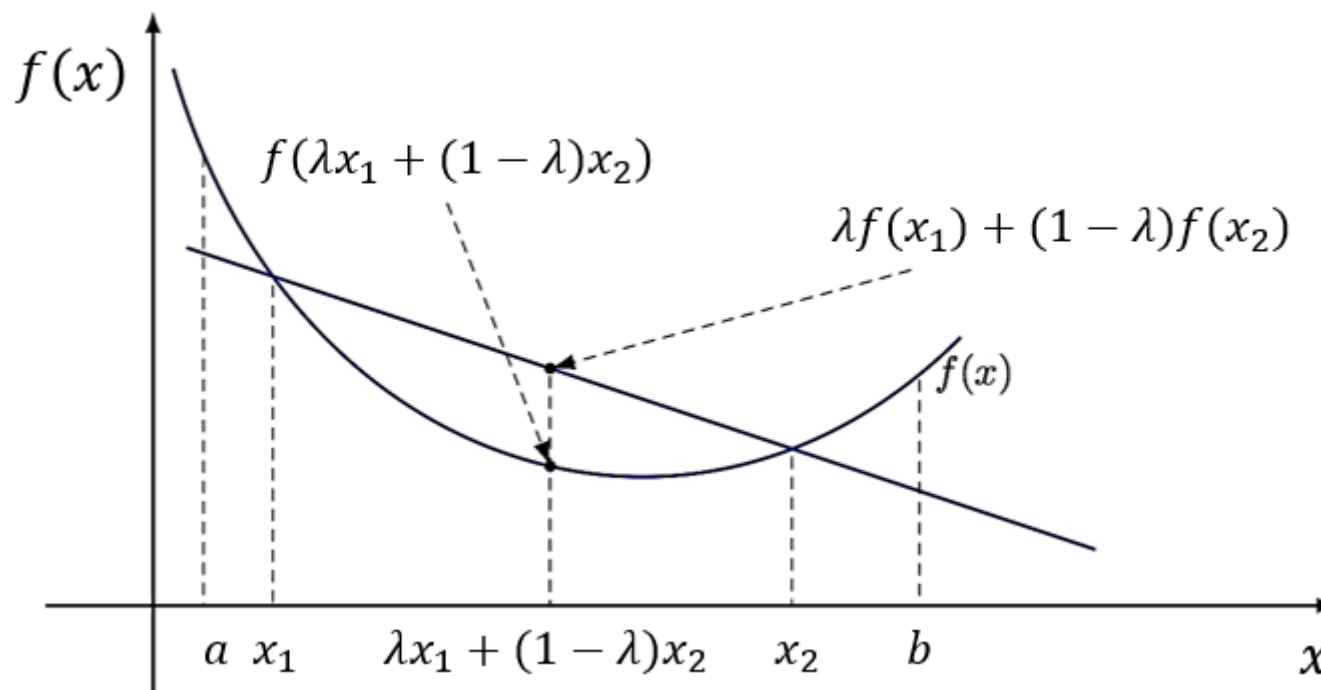
A function that is neither
concave nor convex:
the line segment shown lies
above the graph at some
points and below it at others

Convex Functions

Optimization

- In mathematical terms, the function f is a **convex function** if for all points x_1, x_2 and for all $\lambda \in [0,1]$

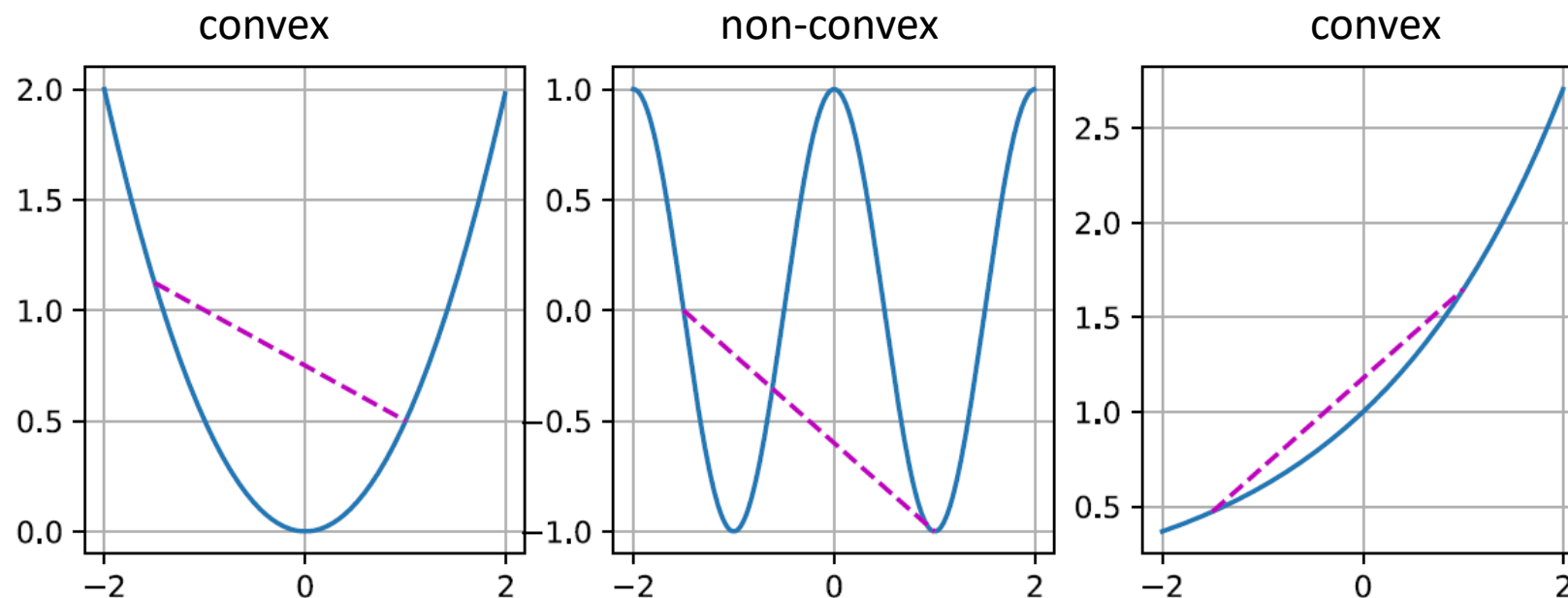
$$\lambda f(x_1) + (1 - \lambda)f(x_2) \geq f(\lambda x_1 + (1 - \lambda)x_2)$$



Convex Functions

Optimization

- One important property of convex functions is that they **do not have local minima**
 - Every local minimum of a convex function is a global minimum
 - I.e., every point at which the gradient of a convex function = 0 is the global minimum
 - The figure below illustrates two convex functions, and one nonconvex function



Probability

Probability

- Intuition:
 - In a process, several outcomes are possible
 - When the process is repeated a large number of times, each outcome occurs with a *relative frequency*, or *probability*
 - If a particular outcome occurs more often, we say it is more probable
- Probability arises in two contexts
 - In actual repeated experiments
 - Example: You record the color of 1,000 cars driving by. 57 of them are green. You **estimate** the probability of a car being green as $57/1,000 = 0.057$.
 - In idealized conceptions of a repeated process
 - Example: You consider the behavior of an unbiased six-sided die. The **expected** probability of rolling a 5 is $1/6 = 0.1667$.
 - Example: You need a model for how people's heights are distributed. You choose a normal distribution to represent the **expected** relative probabilities.

Probability

Probability

- Solving machine learning problems requires to deal with uncertain quantities, as well as with stochastic (non-deterministic) quantities
 - Probability theory provides a mathematical framework for representing and quantifying uncertain quantities
- There are different sources of uncertainty:
 - Inherent stochasticity in the system being modeled
 - For example, most interpretations of quantum mechanics describe the dynamics of subatomic particles as being probabilistic
 - Incomplete observability
 - Even deterministic systems can appear stochastic when we cannot observe all of the variables that drive the behavior of the system
 - Incomplete modeling
 - When we use a model that must discard some of the information we have observed, the discarded information results in uncertainty in the model's predictions
 - E.g., discretization of real-numbered values, dimensionality reduction, etc.

Random variables

Probability

- A **random variable** X is a variable that can take on different values
 - Example: $X =$ rolling a die
 - Possible values of X comprise the **sample space**, or **outcome space**, $\mathcal{S} = \{1, 2, 3, 4, 5, 6\}$
 - We denote the event of “seeing a 5” as $\{X = 5\}$ or $X = 5$
 - The probability of the event is $P(\{X = 5\})$ or $P(X = 5)$
 - Also, $P(5)$ can be used to denote the probability that X takes the value of 5
- A **probability distribution** is a description of how likely a random variable is to take on each of its possible states
 - A compact notation is common, where $P(X)$ is the probability distribution over the random variable X
 - Also, the notation $X \sim P(X)$ can be used to denote that the random variable X has probability distribution $P(X)$
- Random variables can be discrete or continuous
 - **Discrete random variables** have finite number of states: e.g., the sides of a die
 - **Continuous random variables** have infinite number of states: e.g., the height of a person

Axioms of probability

Probability

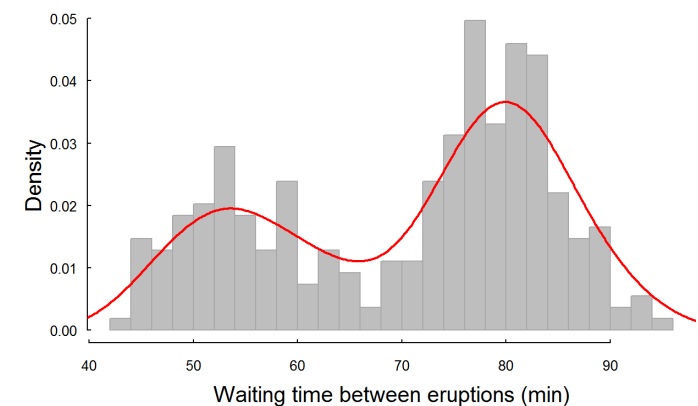
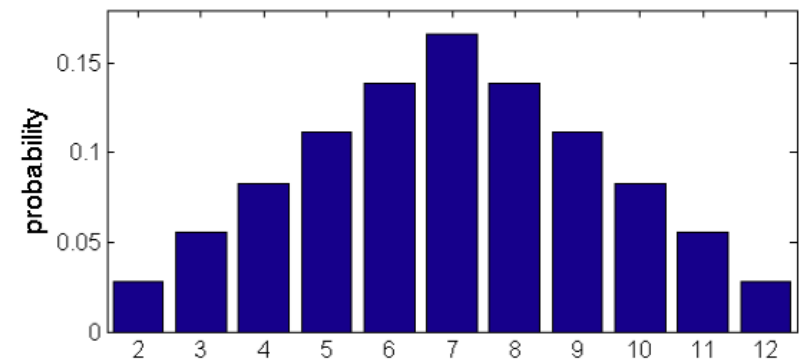
- The probability of an event \mathcal{A} in the given sample space \mathcal{S} , denoted as $P(\mathcal{A})$, must satisfy the following properties:
 - Non-negativity
 - For any event $\mathcal{A} \in \mathcal{S}$, $P(\mathcal{A}) \geq 0$
 - All possible outcomes
 - Probability of the entire sample space is 1, $P(\mathcal{S}) = 1$
 - Additivity of disjoint events
 - For all events $\mathcal{A}_1, \mathcal{A}_2 \in \mathcal{S}$ that are mutually exclusive ($\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$), the probability that both events happen is equal to the sum of their individual probabilities, $P(\mathcal{A}_1 \cup \mathcal{A}_2) = P(\mathcal{A}_1) + P(\mathcal{A}_2)$
- The probability of a random variable $P(X)$ must obey the axioms of probability over the possible values in the sample space \mathcal{S}

Discrete Variables

Probability

- A probability distribution over **discrete variables** may be described using a *probability mass function* (PMF)
 - E.g., sum of two dice
- A probability distribution over **continuous variables** may be described using a *probability density function* (PDF)
 - E.g., waiting time between eruptions of Old Faithful
 - A PDF gives the probability of an infinitesimal region with volume δX
 - To find the probability over an interval $[a, b]$, we can integrate the PDF as follows:

$$P(X \in [a, b]) = \int_a^b P(X)dX$$



Multivariate Random Variables

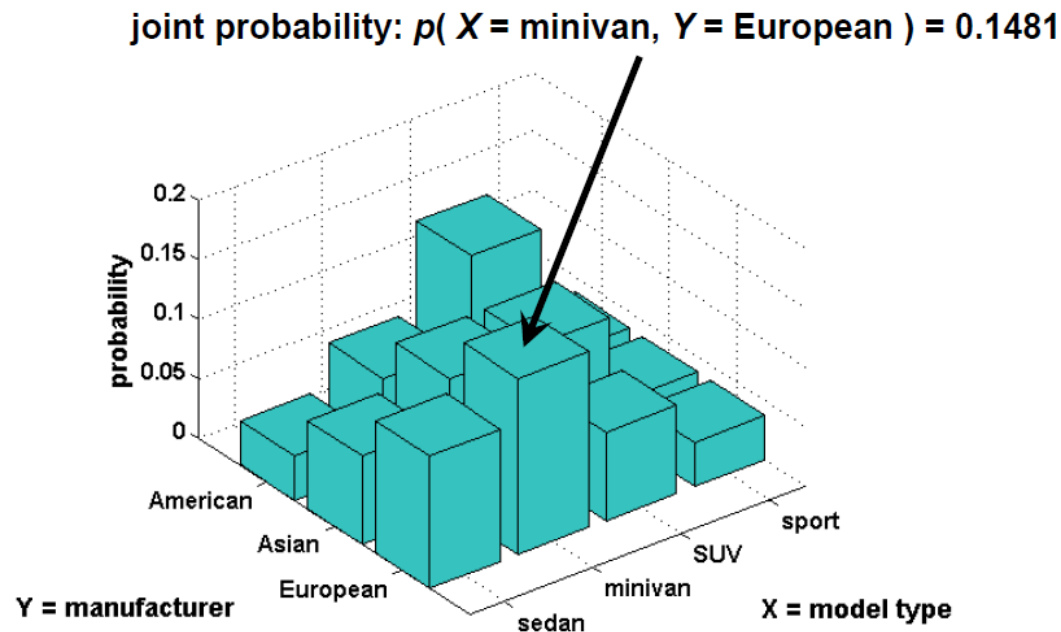
Probability

- We may need to consider several random variables at a time
 - If several random processes occur in parallel or in sequence
 - E.g., to model the relationship between several diseases and symptoms
 - E.g., to process images with millions of pixels (each pixel is one random variable)
- Next, we will study probability distributions defined over multiple random variables
 - These include joint, conditional, and marginal probability distributions
- The individual random variables can also be grouped together into a random vector, because they represent different properties of an individual statistical unit
- A *multivariate random variable* is a vector of multiple random variables $\mathbf{X} = (X_1, X_2, \dots, X_n)^T$

Joint Probability Distribution

Probability

- Probability distribution that acts on many variables at the same time is known as a *joint probability distribution*
- Given any values x and y of two random variables X and Y , what is the probability that $X = x$ and $Y = y$ simultaneously?
 - $P(X = x, Y = y)$ denotes the joint probability
 - We may also write $P(x, y)$ for brevity

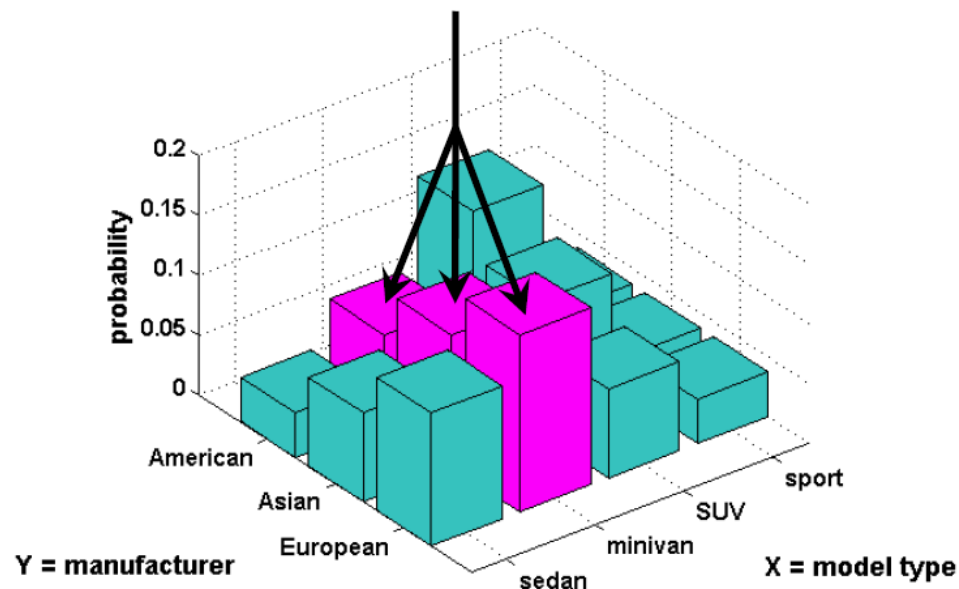


Marginal Probability Distribution

Probability

- **Marginal probability distribution** is the probability distribution of a single variable
 - It is calculated based on the joint probability distribution $P(X, Y)$
 - I.e., using the **sum rule**: $P(X = x) = \sum_y P(X = x, Y = y)$
 - For continuous random variables, the summation is replaced with integration, $P(X = x) = \int P(X = x, Y = y) dy$
 - This process is called **marginalization**

$$\text{marginal probability: } p(X = \text{minivan}) = 0.0741 + 0.1111 + 0.1481 = 0.3333$$

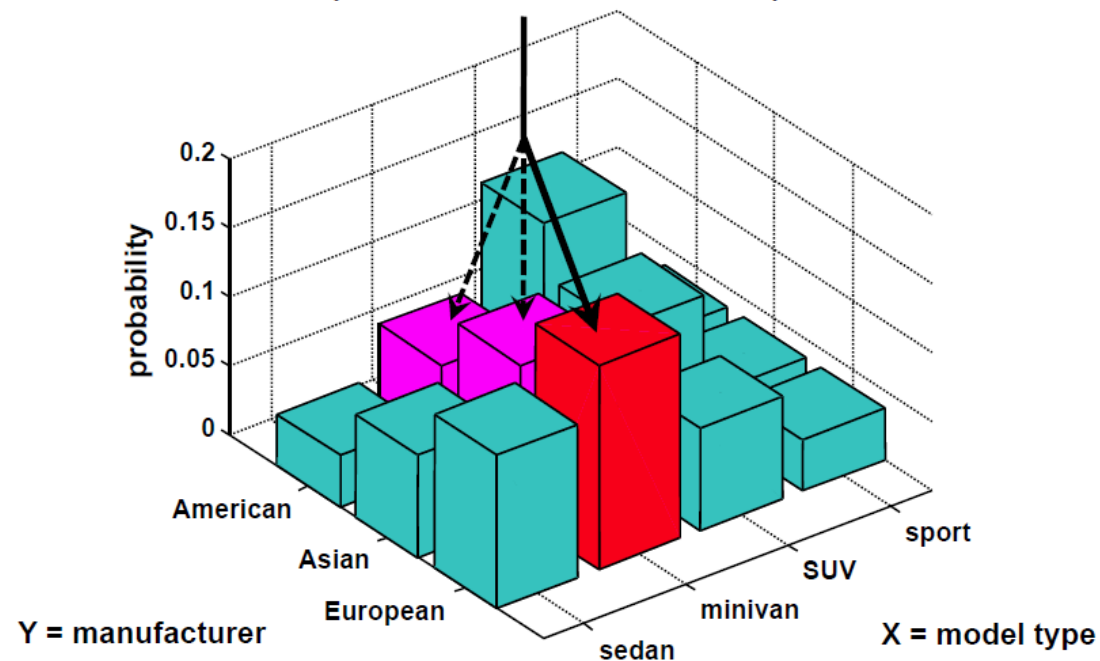


Conditional Probability Distribution

Probability

- **Conditional probability distribution** is the probability distribution of one variable provided that another variable has taken a certain value
 - Denoted $P(X = x | Y = y)$
- Note that: $P(X = x | Y = y) = \frac{P(X=x, Y=y)}{P(Y=y)}$

conditional probability: $p(Y = \text{European} | X = \text{minivan}) = 0.1481 / (0.0741 + 0.1111 + 0.1481) = 0.4433$



Bayes' Theorem

Probability

- **Bayes' theorem** – allows to calculate conditional probabilities for one variable when conditional probabilities for another variable are known

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

- Also known as Bayes' rule
- **Multiplication rule** for the joint distribution is used: $P(X, Y) = P(Y|X)P(X)$
- By symmetry, we also have: $P(Y, X) = P(X|Y)P(Y)$
- The terms are referred to as:
 - $P(X)$, the **prior probability**, the initial degree of belief for X
 - $P(X|Y)$, the **posterior probability**, the degree of belief after incorporating the knowledge of Y
 - $P(Y|X)$, the **likelihood** of Y given X
 - $P(Y)$, the **evidence**
 - Bayes' theorem: **posterior probability** = $\frac{\text{likelihood} \times \text{prior probability}}{\text{evidence}}$

Independence

Probability

- Two random variables X and Y are *independent* if the occurrence of Y does not reveal any information about the occurrence of X
 - E.g., two successive rolls of a die are independent
- Therefore, we can write: $P(X|Y) = P(X)$
 - The following notation is used: $X \perp Y$
 - Also note that for independent random variables: $P(X, Y) = P(X)P(Y)$
- In all other cases, the random variables are *dependent*
 - E.g., duration of successive eruptions of Old Faithful
 - Getting a king on successive draws from a deck (the drawn card is not replaced)
- Two random variables X and Y are *conditionally independent* given another random variable Z if and only if $P(X, Y|Z) = P(X|Z)P(Y|Z)$
 - This is denoted as $X \perp Y|Z$

Expected Value

Probability

- The *expected value* or *expectation* of a function $f(X)$ with respect to a probability distribution $P(X)$ is the average (mean) when X is drawn from $P(X)$
- For a discrete random variable X , it is calculated as

$$\mathbb{E}_{X \sim P}[f(X)] = \sum_X P(X)f(X)$$

- For a continuous random variable X , it is calculated as

$$\mathbb{E}_{X \sim P}[f(X)] = \int P(X)f(X) dX$$

- When the identity of the distribution is clear from the context, we can write $\mathbb{E}_X[f(X)]$
- If it is clear which random variable is used, we can write just $\mathbb{E}[f(X)]$
- Mean is the most common measure of central tendency of a distribution
 - For a random variable: $f(X_i) = X_i \Rightarrow \mu = \mathbb{E}[X_i] = \sum_i P(X_i) \cdot X_i$
 - This is similar to the mean of a sample of observations: $\mu = \frac{1}{N} \sum_i X_i$
 - Other measures of central tendency: median, mode

Variance

Probability

- **Variance** gives the measure of how much the values of the function $f(X)$ deviate from the expected value as we sample values of X from $P(X)$

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

- When the variance is low, the values of $f(X)$ cluster near the expected value
- Variance is commonly denoted with σ^2
 - The above equation is similar to a function $f(X_i) = X_i - \mu$
 - We have $\sigma^2 = \sum_i P(X_i) \cdot (X_i - \mu)^2$
 - This is similar to the formula for calculating the variance of a sample of observations:
$$\sigma^2 = \frac{1}{N-1} \sum_i (X_i - \mu)^2$$
- The square root of the variance is the **standard deviation**
 - Denoted $\sigma = \sqrt{\text{Var}(X)}$

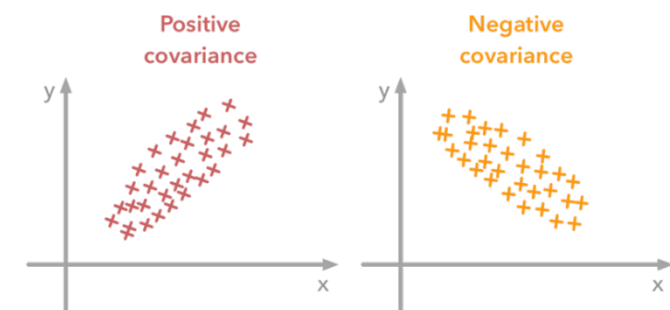
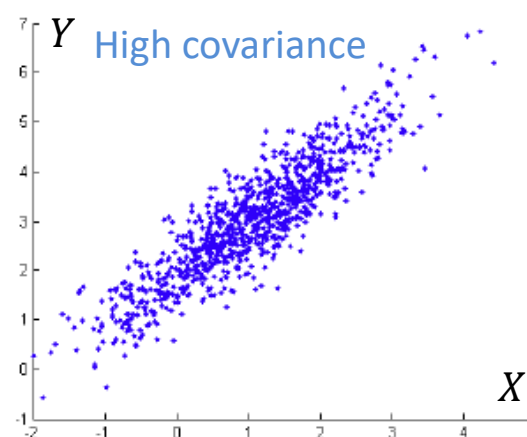
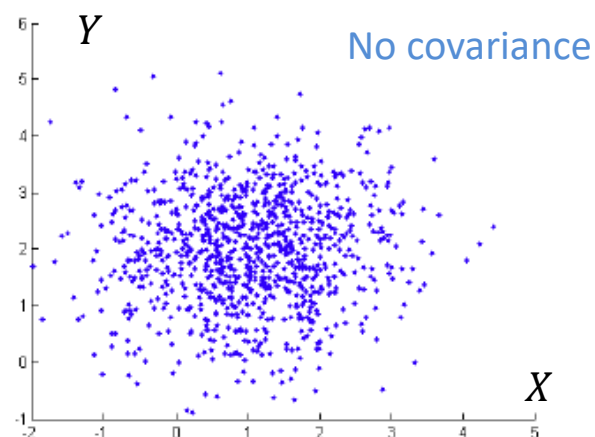
Covariance

Probability

- **Covariance** gives the measure of how much two random variables are linearly related to each other

$$\text{Cov}(f(X), g(Y)) = \mathbb{E}[(f(X) - \mathbb{E}[f(X)])(g(Y) - \mathbb{E}[g(Y)])]$$

- If $f(X_i) = X_i - \mu_X$ and $g(Y_i) = Y_i - \mu_Y$
 - Then, the covariance is: $\text{Cov}(X, Y) = \sum_i P(X_i, Y_i) \cdot (X_i - \mu_X) \cdot (Y_i - \mu_Y)$
 - Compare to covariance of actual samples: $\text{Cov}(X, Y) = \frac{1}{N-1} \sum_i (Y_i - \mu_X)(Y_i - \mu_Y)$
- The covariance measures the tendency for X and Y to deviate from their means in same (or opposite) directions at same time



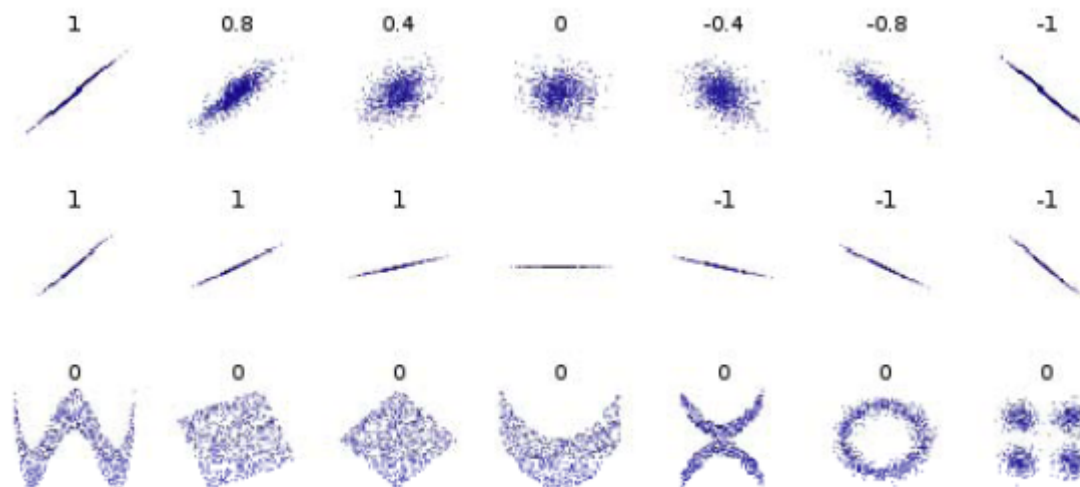
Correlation

Probability

- **Correlation coefficient** is the covariance normalized by the standard deviations of the two variables

$$\text{corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \cdot \sigma_Y}$$

- It is also called **Pearson's correlation coefficient** and it is denoted $\rho(X, Y)$
- The values are in the interval $[-1, 1]$
- It only reflects linear dependence between variables, and it does not measure non-linear dependencies between the variables



Linear dependence
with noise

Linear dependence
without noise

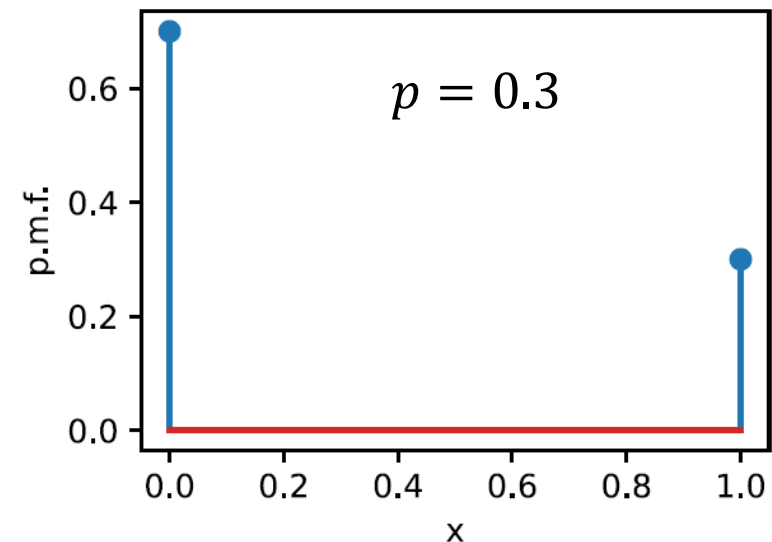
Various nonlinear
dependencies

Probability Distributions

Probability

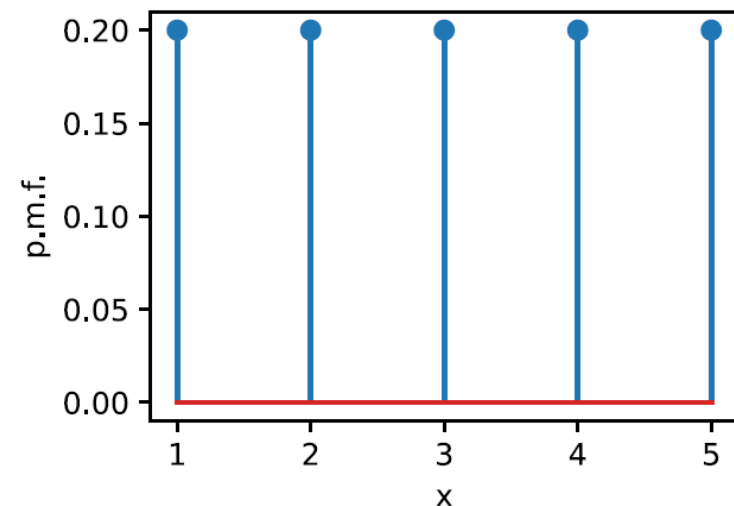
- **Bernoulli distribution**

- Binary random variable X with states $\{0, 1\}$
- The random variable can encode a coin flip which comes up 1 with probability p and 0 with probability $1 - p$
- Notation: $X \sim \text{Bernoulli}(p)$



- **Uniform distribution**

- The probability of each value $i \in \{1, 2, \dots, n\}$ is $p_i = \frac{1}{n}$
- Notation: $X \sim U(n)$
- Figure: $n = 5$, $p = 0.2$

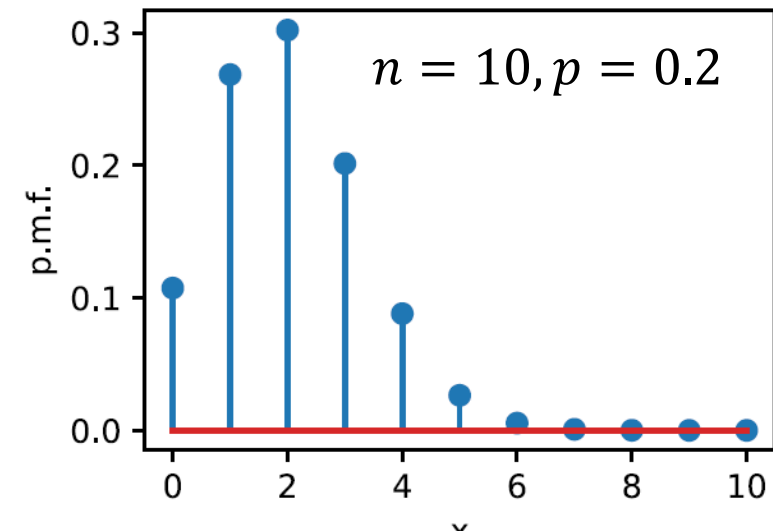


Probability Distributions

Probability

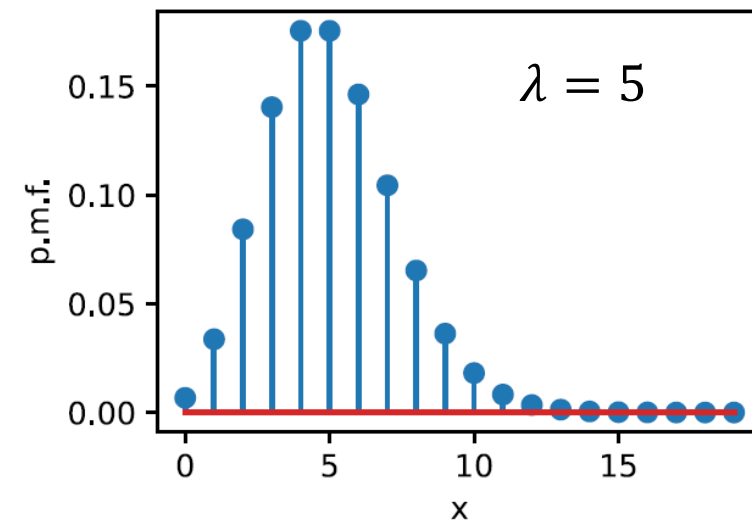
- **Binomial distribution**

- Performing a sequence of n independent experiments, each of which has probability p of succeeding, where $p \in \{0, 1\}$
- The probability of getting k successes in n trials is $P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$
- Notation: $X \sim \text{Binomial}(n, p)$



- **Poisson distribution**

- A number of events occurring independently in a fixed interval of time with a known rate λ
- A discrete random variable X with states $k \in \{0, 1, 2, \dots\}$ has probability $P(X = k) = \frac{\lambda^k \cdot e^{-\lambda}}{k!}$
- The rate λ is the average number of occurrences of the event
- Notation: $X \sim \text{Poisson}(\lambda)$



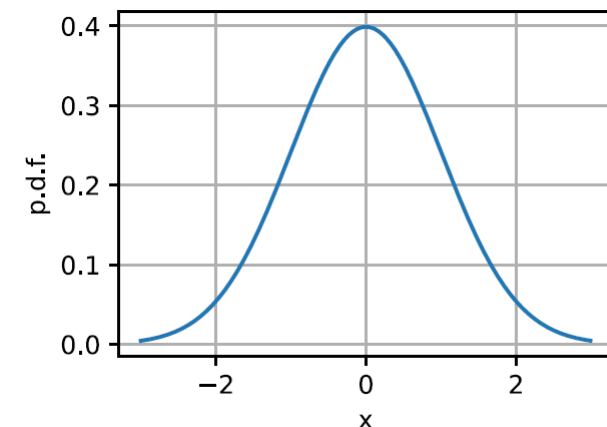
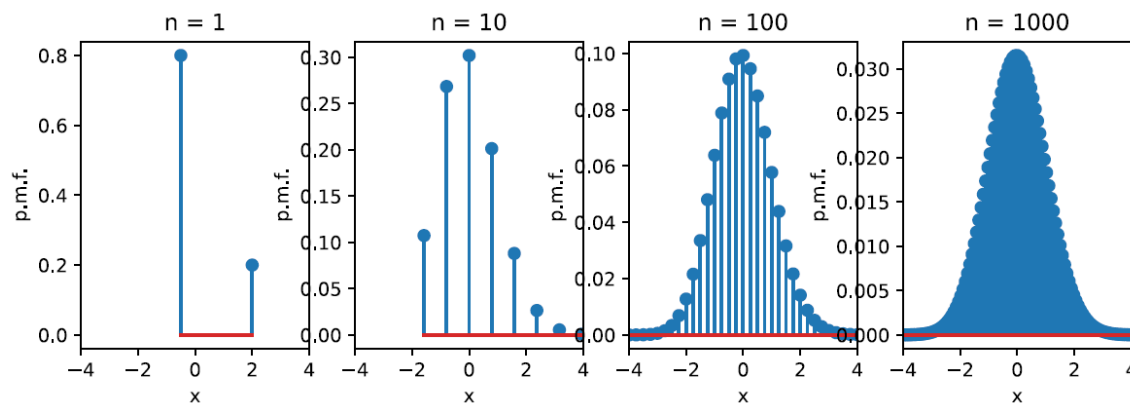
Probability Distributions

Probability

- **Gaussian distribution**

- The most well-studied distribution
 - Referred to as **normal distribution** or informally **bell-shaped distribution**
- Defined with the mean μ and variance σ^2
- Notation: $X \sim \mathcal{N}(\mu, \sigma^2)$
- For a random variable X with n independent measurements, the density is

$$P_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Cross-entropy

Information Theory

- **Cross-entropy** is closely related to the KL divergence, and it is defined as the summation of the entropy $H(P)$ and KL divergence $D_{KL}(P||Q)$

$$CE(P, Q) = H(P) + D_{KL}(P||Q)$$

- Alternatively, the cross-entropy can be written as

$$CE(P, Q) = -\mathbb{E}_{X \sim P} [\log Q(X)]$$

- In machine learning, let's assume a classification problem based on a set of data examples $\{x_1, x_2, \dots, x_n\}$, that need to be classified into k classes
 - For each data example x_i we have a class label y_i
 - The true labels \mathbf{y} follow the true distribution P
 - The goal is to train a classifier (e.g., a NN) parameterized by θ , that outputs a predicted class label \hat{y}_i for each data example x_i
 - The predicted labels $\hat{\mathbf{y}}$ follow the estimated distribution Q
 - The cross-entropy loss between the true distribution P and the estimated distribution Q is calculated as: $CE(\mathbf{y}, \hat{\mathbf{y}}) = -\mathbb{E}_{X \sim P} [\log Q(X)] = -\sum_X P(X) \log Q(X) = -\sum_i y_i \log \hat{y}_i$
 - The further away the true and estimated distributions are, the greater the cross-entropy loss is

References

1. A. Zhang, Z. C. Lipton, M. Li, A. J. Smola, *Dive into Deep Learning*, <https://d2l.ai>, 2020.
2. I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2017.
3. M. P. Deisenroth, A. A. Faisal, C. S. Ong, *Mathematics for Machine Learning*, Cambridge University Press, 2020.
4. Jeff Howbert – Machine Learning Math Essentials presentation
5. Brian Keng – Manifolds: A Gentle Introduction [blog](#)
6. Martin J. Osborne – Mathematical Methods for Economic Theory ([link](#))