Preliminaries
PART A
Algebra
Unit A-1

Sets

Knowledge and activity graph

1.1. Definition

Definition 1. (Set) A set \( \mathcal{X} = \{x_1, x_2, \ldots \} \) is a well-defined collection of distinct elements, or members of the set, \( x_i, i = 1, 2, \ldots \).

1.2. Maps

Knowledge and activity graph

Requires: k:sets
Results: k:maps

1.3. Definition

We define a function (or map) as a mapping between sets.

Definition 2. (Function) A function \( f : \mathcal{X} \rightarrow \mathcal{Y} \) is a mapping between the sets \( \mathcal{X} \) and \( \mathcal{Y} \). For every input element \( x \in \mathcal{X} \), the mapping will associate an output \( y = f(x) \in \mathcal{Y} \).

1.4. Properties of maps

Maps can be classified by the nature of the relationship between inputs and outputs in: injective, surjective or bijective add-ref.

1) Injective maps

2) Surjective maps

3) Bijective maps
UNIT A-2
Numbers

Knowledge and activity graph

<table>
<thead>
<tr>
<th>Requires: k:sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results: k:naturals, k:integers, k:reals</td>
</tr>
</tbody>
</table>

2.1. Natural numbers

\[ \mathbb{N} = \{0, 1, 2, \cdots\} \]

The natural numbers are the set positive numbers, including zero. Given two natural their addition is always a natural number:

\[ a + b = c \in \mathbb{N}, \forall a, b \in \mathbb{N}. \quad (1) \]

The same does not hold of the subtraction operation:

\[ a - b = c \in \mathbb{N} \iff a \geq b. \]

For this reason set of integer numbers is defined.

2.2. Integers

\[ \mathbb{Z} = \{\cdots, -2, -1, 0, 1, 2, \cdots\} \]

The integers are the set of positive and negative natural numbers, including the zero. By definition, the set of integers includes the naturals: \( \mathbb{Z} \subset \mathbb{N} \).

The sum (i.e., addition and subtraction) of two integers is always an integer:

\[
\begin{align*}
\quad a + b &= c \in \mathbb{Z}, \forall a, b \in \mathbb{N} \\
\quad a - b &= c \in \mathbb{Z}, \forall a, b \in \mathbb{N}.
\end{align*}
\]

The multiplication of two integers is always an integer, but the same does not apply for the division operation:

\[
\frac{a}{b} = c \in \mathbb{Z} \iff a = kb, k \in \mathbb{Z}, b \neq 0.
\]

For this reason the rational numbers are introduced.

2.3. Rationals

The set of rational numbers includes all fractions of integers: \( \mathbb{Q} = \{c|\frac{a}{b} = c, a, b \in \mathbb{Z}, b \neq 0\} \).

The set of rational number is complete under sum and product (i.e., multiplication and division), but not under other operations such as the root. E.g., \( \sqrt{2} \) cannot be expressed as a fraction of two integers. These numbers are not rational, and therefore are defined
as irrationals.

2.4. Irrationals
Irrational numbers are all those numbers that cannot be expressed as a fraction. Notable examples of irrational numbers include the aforementioned $\sqrt{2}$, but even pi ($\pi$) and the Euler number ($e$).
Irrational numbers are not typically referred to as a set by themselves, rather, the union of the rational and irrational numbers defines the set of reals.

2.5. Reals
The real numbers ($\mathbb{R}$) are arguably the most used set of numbers, and are often considered the default set if no specification is provided.
The real numbers are defined as the union of rational and irrational numbers, and therefore by definition include the integers and the naturals.
The reals are still not complete under all “canonical” operations. In fact, there is no solution to the root (of even index) of a negative number.
For this reason, the complex numbers are introduced.

2.6. Complex
Complex numbers are defined as the sum of a real and an imaginary part:

$$z = a + ib, a, b \in \mathbb{R}, i = \sqrt{-1}$$

and can be represented on the plane of Gauss, a Cartesian plane featuring the real part of $z$, $Re(z) = a$, on the x-axis and the imaginary part, $Im(z) = b$, on the y-axis (Figure 2.2).

Figure 2.2. The Gaussian plane is used to represent complex numbers
Complex numbers introduce the concept of phase of a number, which is related to its “orientation”, and are invaluable for describing many natural phenomena such as electricity and applications such as signal decoders.
For more information on the algebra and properties of natural numbers:
- [Unit A-3 - Complex numbers](#).
UNIT A-3
Complex numbers

3.1. Powers of $i$

The powers of $i$

\[ i = \sqrt{-1} \]
\[ i^2 = -1 \]
\[ i^3 = i^2 \cdot i = -i \]
\[ i^4 = i^2 \cdot i^2 = 1 \]
\[ i^5 = i^4 / i = i \]
\[ \vdots \]

3.2. Complex conjugate
Linear algebra provides the set of mathematical tools to (a) study linear relationships and (b) describe linear spaces. It is a field of mathematics with important ramifications. Linearity is an important concept because it is powerful in describing the input-output behavior of many natural phenomena (or systems). As a matter of fact, all those systems that cannot be modeled as linear, still can be approximated as linear to gain an intuition, and sometimes much more, of what is going on.

So, in a way or the other, linear algebra is a starting point for investigating the world around us, and Duckietown is no exception.

Note: This chapter is not intended to be a comprehensive compendium of linear algebra.

4.1. Linearity

In this section we discuss vectors, matrices and linear spaces along with their properties.

Before introducing the these arguments, we need to formally define what we mean by linearity. The word linear comes from the latin linearis, which means pertaining to or resembling a line. You should recall that a line can be represented by an equation like $y = mx + q$, but here we intend linearity as a property of maps, so there is a little more to linearity than lines (although lines are linear maps indeed).

To avoid confusions, let us translate the concept of linearity in mathematical language.

**Definition 3.** (Linearity) A function $f : \mathcal{X} \to \mathcal{Y}$ is linear when, $\forall x_i \in \mathcal{X}, i = \{1, 2\}$, and $\forall a \in \mathbb{R}$:

\[
\begin{align*}
    f(ax_1) &= af(x_1), \quad \text{and:} \\
    f(x_1 + x_2) &= f(x_1) + f(x_2)
\end{align*}
\]

Condition (2) is referred to as the property of homogeneity (of order 1), while condition (3) is referred to as additivity.

**Remark 1.** (Superposition Principle) Conditions (2) and (3) can be merged to express the same meaning through:

\[
    f(ax_1 + bx_2) = af(x_1) + bf(x_2), \forall x_i \in \mathcal{X}, i = \{1, 2\}, \forall a, b \in \mathbb{R}.
\]
This equivalent condition (4) is instead referred to as *superposition principle*, which unveils the bottom line of the concept of linearity: adding up (equivalently, scaling up) inputs results in an added up (equivalently, scaled up) output.

### 4.2. Vectors

Let \( n \) belong to the set of natural numbers \( \mathbb{N} \), i.e., \( n \in \mathbb{N} \), and let \( a_i \in \mathbb{R} \), \( i = \{1, \ldots, n\} \) be real coefficients. While \( \mathbb{R} \) is the set of real numbers, \( \mathbb{R}^n \) is the set of all \( n \)-tuples of real numbers.

**Definition 4.** (Vector and components) An \( n \)-dimensional vector is an \( n \)-tuple:

\[
\mathbf{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \in \mathbb{R}^{n\times 1} = \mathbb{R}^n,
\]

of components \( v_1, \ldots, v_n \in \mathbb{R} \).

**Remark 2.** (Vector notation) A more general notation for tuples can be used when denoting vectors:

\[
\mathbf{v} = (v_1, \ldots, v_n).
\]

In these preliminaries, we will adopt the (5) “engineering” notation as it arguably simplifies remembering vector-matrix operations (Unit A-7 - Matrices and vectors).

You can imagine a vector Figure 4.2 as a “directional number”, or an arrow that starts a certain point and goes in a certain direction (in \( \mathbb{R}^n \)). In this representation, the *number* is the length of the arrow, or the *magnitude* of the vector (sometimes referred to even as *modulus*), and it can be derived through the vector’s components.

**Definition 5.** (Length of a vector) We define the length, or *modulus*, of a vector \( \mathbf{v} \in \mathbb{R}^n \) as:

\[
\|\mathbf{v}\| = \sqrt{v_1^2 + \cdots + v_n^2} \in \mathbb{R}.
\]

**Remark 3.** (2-norm) Generally speaking, it is not always possible to define the length of a vector (addref). But when it is possible (e.g., in Hilbert spaces), and in Duckietown it always is, there are many ways to define it. The most common and intuitive definition is the *Euclidian- or 2-norm*, which is defined above in (7).

We will discuss norms more in detail in Unit A-11 - Norms.

**Definition 6.** (Unit vector) A unit vector, or *versor*, is a vector \( \mathbf{e} \) of of unit length:

\[
\|\mathbf{e}\| = 1.
\]

Unit vectors are used to define the directions of the components of a vector, allowing for an algebraic rather than vectorial representation. As we will see in Subsection 4.2.1 - Vector algebra, this will make the algebra of vectors more intuitive.
Let \( \mathbf{v} \in \mathbb{R}^3 \) be a vector defined in the Cartesian space. Let, moreover, \((\mathbf{i}, \mathbf{j}, \mathbf{k})^T\) be the versor of the Cartesian axis, i.e.:

\[
\begin{align*}
\mathbf{i} &= [1, 0, 0]^T; \\
\mathbf{j} &= [0, 1, 0]^T; \\
\mathbf{k} &= [0, 0, 1]^T.
\end{align*}
\] (9)

Then, a vector can be written equivalently in vector or algebraic form:
\[ \mathbf{v} = [v_1, v_2, v_3]^T = v_1 \mathbf{i} + v_2 \mathbf{j} + v_3 \mathbf{k}. \]

**Remark 4.** (Normalizing vectors) Every vector can be made into a unit vector, or normalized, by dividing each of its components by the vector’s magnitude:

\[
\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|} = \left[ \frac{v_1}{\|\mathbf{v}\|}, \frac{v_2}{\|\mathbf{v}\|}, \frac{v_3}{\|\mathbf{v}\|} \right]^T. 
\] (10)

1) **Vector algebra**

We here define operations amongst two given vectors defined in the same space:\n\[ \mathbf{u} = [u_1, u_2, u_3]^T, \mathbf{v} = [v_1, v_2, v_3]^T \in \mathbb{R}^3. \]

**Vectorial Sum:**

The sum of two vectors is a vector, and its components are the sum of the two vectors components.

**Definition 7.** (Vectorial sum)

\[
\mathbf{u} + \mathbf{v} = [u_1 + v_1, u_2 + v_2, u_3 + v_3]^T. 
\] (11)

**Remark 5.** (Sum) Mathematical operations come in pairs, which represent the same concept. A sum operation, sometimes more extensively referred to as the algebraic sum, is the concept of summing, i.e., it includes both addition and subtraction. (A subtraction is nothing but an addition between positive and negative numbers.)

The parallelogram law helps visualize the results of the vectorial sum operation **Figure 4.4.**
Figure 4.4. The sum of two vectors can be visualized with the parallelogram law.

**Dot, or scalar, product:**

The dot, or scalar, product of two vectors \((\mathbf{u}, \mathbf{v} \in \mathbb{R}^3)\) is a scalar \((a \in \mathbb{R})\) equal to the sum of the products of the components of the vectors. Equivalently, it can be expressed as the product of the magnitudes of the two vectors times the cosine of the angle between them, \(\phi \in [0, 2\pi)\).

**Definition 8.** (Scalar product)

\[
\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + u_3 v_3 = ||\mathbf{u}|| ||\mathbf{v}|| \cos(\phi) \in \mathbb{R}
\]  

The dot product is a measure of the *projection* of vectors on one another (Figure 4.6).

**Note:** When the two vectors are perpendicular, or orthogonal, the dot product is zero \((\cos(\pi/2) = 0)\). This fact is often used as a test for orthogonality. Orthogonality is an important concept for linear spaces, as the most “efficient” basis are orthogonal.

Figure 4.6. The scalar product between two vectors measures the projection of one on each other.

**Cross, or vector, product:**

While the dot product depends on the metric chosen in the space (the Euclidian norm, in our case), the cross product even requires the definition of an orientation, or hand-
edness.

**Proposition 1.** (Standard Basis) In the Euclidean space $\mathbb{R}^3$, $\hat{i}, \hat{j}, \hat{k}$ are the unit vectors for the standard basis, which is right handed.

In a right handed reference system such as the standard basis, the right hand rule (Figure 4.8) is the handy-est way to identify the direction of the vector resulting from a cross product.

ProTip: There is a valid reason for which it is called the right hand rule. Don’t use your left hand because you are holding a pen with the right one.

Figure 4.8. The right hand rule points in the direction of the resulting vector from a cross product.

The cross, or vector, product between two vectors ($\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$) is a vector that is orthogonal to each of the two vectors, hence is normal, or perpendicular, to the plane containing them. Its magnitude is given by the product of their magnitude times the sine of the angle between them, and its direction is indicated by the normal unit vector ($\hat{n} \in \mathbb{R}^3$), identified by the right hand rule.

**Definition 9.** (Vector product)

$$
\mathbf{u} \times \mathbf{v} = [u_2v_3 - u_3v_2, u_3v_1 - u_1v_3, u_1v_2 - u_2v_1]^T = \|\mathbf{u}\|\|\mathbf{v}\| \sin(\phi)\hat{n}.
$$

**Remark 6.** (Geometric interpretation) A cross product encodes two pieces of information: a direction, which is orthogonal to the plane spanned by the two vectors, and a magnitude, which is equal to the area of the parallelogram having $\mathbf{u}$, and $\mathbf{v}$ as sides.

**Note:** Keeping (13) and Remark 6 - Geometric interpretation in mind, it should be intuitive to understand that:

$$
\mathbf{v} \times \mathbf{v} = 0, \forall \mathbf{v} \in \mathbb{R}^n, \quad \mathbf{v} \times 0 = 0, \forall \mathbf{v} \in \mathbb{R}^n.
$$

**Note:** The zero vector (0) is a vector with zero magnitude, not the same as the number 0.
zero (0).

**Note:** Each component of \( \mathbf{w} \) is the difference of the products of the two other components of \( \mathbf{u} \), and \( \mathbf{v} \), in the order given by the chosen handedness of the basis. This combination resembles a *cross* (Figure 4.10), from which the name of *cross product*.

![Image](image_url)

**Figure 4.10.** Each component of the resulting vector is the product of the alternated other components, forming a cross.

**Note:** The components of a cross product can be computed through the Sarrus rule (see Section 5.6 - Determinant).

As consequence of the vectorial product’s definition and right handedness of the basis, the following hold true in the Cartesian space:

\[
\begin{align*}
\hat{i} \times \hat{j} &= \hat{k} \\
\hat{j} \times \hat{k} &= \hat{i} \\
\hat{k} \times \hat{i} &= \hat{j}.
\end{align*}
\]  

(15)

2) Properties of vectors

In this section we highlight the properties of vector operations, that derive from their definitions.

**Sum:**

The vector sum obeys the following:

- \( \mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u} \),
- \( (\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w}) \),
- \( a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v} \),
- \( (a + b)\mathbf{u} = a\mathbf{u} + b\mathbf{u} \),
- \( \mathbf{u} + \mathbf{0} = \mathbf{u} \), therefore \( \mathbf{u} + (\mathbf{-u}) = \mathbf{0} \).

**Dot product:**
Letting $\phi \in [0, 2\pi)$ be the angle between two vectors $\mathbf{u}, \mathbf{v}$, the dot product obeys the following:

- $\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\|\|\mathbf{v}\| \cos(\phi)$,
- $\mathbf{u} \cdot \mathbf{u} = \|\mathbf{u}\|^2$,
- $\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}$,
- $\mathbf{u} \cdot (\mathbf{v} + \mathbf{w}) = \mathbf{u} \cdot \mathbf{v} + \mathbf{u} \cdot \mathbf{w}$,
- $a(\mathbf{u} \cdot \mathbf{v}) = (a\mathbf{u}) \cdot \mathbf{v}$,
- $0 \cdot \mathbf{u} = 0$
- $\mathbf{u} \cdot \mathbf{v} = 0 \iff \mathbf{u} = \mathbf{0}, \mathbf{v} = \mathbf{0}$, or $\mathbf{u} \perp \mathbf{v}$.

**Cross product:**

Letting $\phi \in [0, 2\pi)$ be the angle between two vectors $\mathbf{u}, \mathbf{v}$, the cross product obeys the following:

- $\mathbf{u} \times \mathbf{v} = \|\mathbf{u}\|\|\mathbf{v}\| \sin(\phi) \mathbf{n}$,
- $\mathbf{u} \times \mathbf{v} = -\mathbf{v} \times \mathbf{u}$,
- $(a\mathbf{u}) \times \mathbf{v} = \mathbf{u} \times (a\mathbf{v}) = a(\mathbf{u} \times \mathbf{v})$,
- $\mathbf{u} \times (\mathbf{v} + \mathbf{w}) = \mathbf{u} \times \mathbf{v} + \mathbf{u} \times \mathbf{w}$,
- $\mathbf{u} \cdot (\mathbf{v} \times \mathbf{w}) = (\mathbf{u} \times \mathbf{v}) \cdot \mathbf{w}$,
- $\mathbf{u} \times (\mathbf{v} + \mathbf{w}) = (\mathbf{w} \cdot \mathbf{u})\mathbf{v} - (\mathbf{v} \cdot \mathbf{u})\mathbf{w} \neq (\mathbf{u} \times \mathbf{v}) + \mathbf{w}$,
- $\mathbf{u} \times \mathbf{v} = 0 \iff \mathbf{u} = \mathbf{0}, \mathbf{v} = \mathbf{0}$, or $\mathbf{u} \parallel \mathbf{v}$.

### 4.3. Linear dependance

**Definition 10.** (Linear dependance) Two or more vectors $\{\mathbf{v}_1, \cdots, \mathbf{v}_n\}$ are **linearly dependent** if there exists a set of scalars $\{a_1, \cdots, a_k\}$, $k \leq n$, that are not all zero, such that:

$$a_1\mathbf{v}_1 + \cdots + a_k\mathbf{v}_k = \mathbf{0}. \quad (1)$$

**Note:** When (1) is true, it is possible to write at least one vector as a linear combination of the others.

**Definition 11.** (Linear independance) Two or more vectors $\mathbf{v}_1, \cdots, \mathbf{v}_n$ are **linearly independant** if (1) can be satisfied only by $k = n$ and $a_i = 0, \forall i = 1, \cdots, n$.

### 4.4. Pointers to Exercises

### 4.5. Conclusions

In this section we have defined the fundamental concept of linearity and linear dependance. Moreover, we have introduced vectors, with their operations and algebraic properties.

Vectors and linearity are the base for understanding linear spaces, which are useful because they introduce some fundamental concepts related to the foundation of modeling of natural phenomena. Modeling will be invaluable in understanding the behavior of systems, and a powerful tool to predict future behaviors of the system, and control
them when needed.

**Knowledge and Activity Graph**

* [1]
* Matrix cookbook

**Author:** Jacopo Tani

**Point of contact:** Jacopo Tani
A matrix:

\[
A = \begin{bmatrix}
    a_{11} & \cdots & a_{1n} \\
    \vdots & \ddots & \vdots \\
    a_{m1} & \cdots & a_{mn}
\end{bmatrix} \in \mathbb{R}^{m \times n}
\] (1)

is a table ordered by \((m)\) horizontal rows and \((n)\) vertical columns. Its elements are typically denoted with lower case latin letters, with subscripts indicating their row and column respectively. For example, \(a_{ij}\) is the element of \(A\) at the \(i\)-th row and \(j\)-th column.

5.1. Matrix dimensions

The number of rows and columns of a matrix are referred to as the matrix dimensions. \(A \in \mathbb{R}^{m \times n}\) has dimensions \(m\) and \(n\).

**Definition 12.** (Fat matrix) When \(n > m\), i.e., the matrix has more columns than rows, \(A\) is called **fat** matrix.

**Definition 13.** (Tall matrix) When \(n < m\), i.e., the matrix has more rows than columns, \(A\) is called **tall** matrix.

**Definition 14.** (Fat matrix) When \(n = m\), \(A\) is called **square** matrix.

**Note:** Square matrices are particularly important.

5.2. Matrix diagonals
• Main diagonal
• Secondary diagonal

5.3. Diagonal matrix

Definition 15. (Diagonal matrix) A diagonal matrix has non zero elements only on its main diagonal.

\[ A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix} \]

5.4. Identity matrix

Definition 16. (Identity matrix) An identity matrix is a diagonal square matrix with all elements equal to one.

\[ I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \]

5.5. Null matrix

Definition 17. (Null matrix) The null, or Zero, matrix is a matrix whose elements are all zeros.

\[ 0 = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \]

5.6. Determinant
• 2x2
• 3x3
• nxn

5.7. Rank of a matrix

5.8. Trace of a matrix
6.1. Adjugate matrix
\[ \text{Adj}(A) = \det(A)I \]

6.2. Matrix Inverse
Square matrix:
\[ AA^{-1} = I \]

6.3. Nonsingularity of a matrix
Exercise: Calculate a (square) Matrix Inverse
Exercise: Inverting a well-conditioned matrix (practice)
Exercise: Inverting an ill-conditioned matrix (practice)
UNIT A-7
Matrices and vectors

- matrix-vector product

7.1. Matrix as representation of linear (vector) spaces
- linear system to matrix representation
- linearly dependent and independent spaces

1) Fundamental spaces
- Null space
- Range/image

2) Eigenvalues and Eigenvectors
- for square matrices
- for rectangular matrices (topic for advanced-linear-algebra?)
- condition number of a matrix (?)
UNIT A-8
Matrix operations (basic)

- sum of matrices
- product of matrices
- matrix transpose – Symmetric matrix
- matrix concatenation

1) Properties
UNIT A-9

Matrix operations (complex)

- matrix scalar product
- matrix Hadamart product
- matrix power
- matrix exponential

1) Properties
UNIT A-10

Matrix diagonalization

• singular value decomposition SVD (topic for advanced-linear-algebra?)

1) Preferred spaces (matrix diagonalization)
• show how to diagonalize matrices and why it is relevant (it will come in handy for state space representation chapter)

10.1. Left and Right Inverse (topic for advanced-linear-algebra?)
• what if the matrix is not square? (topic for advanced-linear-algebra?)
• Moore-Penrose pseudo-inverse

*Example: Find eigenvalues and eigenvectors:
Example: Find range and null spaces of a matrix:
Other metrics can be defined to measure the “length” of a vector. Here, we report some commonly used norms. For a more in depth discussion of what constitutes a norm, and their properties, see: 

- [4].

1) $p$-norm

Let $p \geq 1 \in \mathbb{R}$. The $p$-norm is defined as:

**Definition 18.** ($p$-norm)

$$
\|v\|_p = \left( \sum_{i=1}^{n} |v_i|^p \right)^{\frac{1}{p}}.
$$

(1)

The $p$-norm is a generalization of the 2-norm ($p = 2$ in (??)) introduced above (**Definition 5 - Length of a vector**). The following 1-norm and $\infty$-norm can as well be obtained from (1) with $p = 1$ and $p \to \infty$ respectively.

2) One norm

The 1-norm is the sum of the absolute values of a vector’s components. It is sometimes referred to as the *Taxicab norm*, or *Manhattan distance* as it well describes the distance a cab has to travel to get from a zero starting point to a final destination $v_i$ on a grid.

**Definition 19.** (1-norm) Given a vector $v \in \mathbb{R}^n$, the 1-norm is defined as:

$$
\|v\| = \sum_{i=1}^{n} |v_i|.
$$

(2)

3) $\infty$-norm

The infinity norm measures the maximum component, in absolute value, of a vector.

**Definition 20.** ($\infty$-norm)

$$
\|v\| = \max(|v_1|, \ldots, |v_n|).
$$

(3)

11.1. Definition

11.2. Properties
Convex optimization is a minimization technique where one optimizes a convex function over convex sets.
In the following few lines we'll talk about the standard form of convex optimization.

12.1. Motivation
Often there are problems which should be minimized but with respect to some constraint. One can think of a nutrition problem where people want to spend the least amount of money possible on food while still having all the important levels of nutrition. So one would minimize the cost with respect to inequality constraints like the amount of calcium has to be bigger than 1000 milligrams per day or the amount of vitamin C has to be bigger than 90 milligrams a day.

12.2. Mathematical description of the problem
Mathematically speaking the problem would be formulated as follows. One defines the function \( f(x) \) to be minimized. Now a point \( x^* \) has to be found such that \( f(x^*) = \min f(x) : x \in X \) which means \( f(x^*) \) is the minimal function value of all values \( x \) in some feasible set \( X \).
The function \( f(x) \) can be minimized now subject to several constraints. There are basically two possibilities: 1. inequality constraints \( h(x) \leq 0 \) or \( h(x) \geq 0 \) 2. equality constraints \( h(x) = 0 \) which can be replaces by a pair of inequality constraints \( h(x) \leq 0 \) and \( -h(x) \leq 0 \) The standard form would be formulated as follows:

\[
\min_{x} f(x) \\
\text{subject to } g_i(x) \leq 0, i = 1, \ldots, m \\
\text{and } h_i(x) = 0, i = 1, \ldots, p
\]

12.3. Limits
The functions \( f(x) \) and \( g_i(x) \) have to be convex and \( h_i(x) \) has to be affine such that the replacement of the equality constraint by two inequality constraints is valid.
PART B

Geometry
UNIT B-1
From ODEs to LTI systems

1.1. ODEs

1.2. LTI Systems

1) Properties of LTI systems
UNIT B-2
Linearization

2.1. Taylor Series Expansion

2.2. Linearization of a nonlinear system
Kinematics is the study of position, velocity and acceleration of geometric points.

**Note:** A point has no dimensions. For example, the center of mass of a body is a point.
UNIT B-4

Dynamics
5.1. Motivation

In order to uniquely specify a position in some space, we use some numbers, coordinates, in a coordinate system. For example, in daily life, we would use the intersection of two streets, each with a unique name in the city, to specify the location of some cafe. If you find yourself in the ocean, you might communicate your location to someone by telling them the latitude and longitude readout on your GPS device. Generally speaking, a coordinate system provides us a way to denote any position in an unambiguous way. So you can communicate any position to another person, and by using the given coordinates, that person can arrive at the same exact position in space. Note, however, different coordinates can correspond to the same point.

5.2. Definitions

Definition 21. (Coordinate system) A coordinate system is a surjective function mapping from a tuple of reals to some space \( S \), respecting the local topology. (The local topology, which is beyond the scope of this chapter, roughly means that “nearby” points have coordinates “close” together.)

5.3. Examples

Because the ability of naming or specifying a point in space is so fundamentally important, we often take that coordinates given by some coordinate system, often a Cartesian coordinate system, as the name of the point.

1) 1D

Consider the real number line \( \mathbb{R} \) as the space \( S \). We can name an arbitrary point \( x \in \mathbb{R} \), which happens to be a real number, by itself. So to check with our definition, any two distinct points on the real line would have two distinct coordinates. Furthermore any point has a coordinate. Let’s call this coordinate system \( A \).

One should note that the coordinate system given above is not the only possible way to name the points on a real line. We can assign coordinate \(-x\) to the point \( x \in \mathbb{R} \) and obtain an equally valid coordinate system \( B \). To be more specific, given a point \( p \) with the coordinate \( a \) in the first coordinate system \( A \), we know that in the second coordinate system \( B \), \( p \) would have the coordinate \(-a\). Therefore, there is a way to translate between coordinates in the two systems.

2) 2D plane

Consider the real plane \( \mathbb{R}^2 \) as the space \( S \). This is an important space within robotics and beyond. It can be used to represent images, with each pixel having its own position, or the location of your Duckiebot in Duckietown.

2D Cartesian coordinate systems:

We can draw two perpendicular lines on the plane \( S \). We call these two lines the \( x \)-axis
and the y-axis. We assign (0,0) to the point of intersection, which we call the origin. Then we decide on the positive directions and units for each axis and the unit. We will use coordinates (x, y) to specify a point located x-many units in the positive x-direction and y-many units in the positive y-direction away from the origin, respectively. If we draw the axis-parallel lines with integral x coordinate and lines with integral y coordinate, we obtain a visualization similar to that in Figure 5.2.

When representing the location of your Duckiebot in Duckietown, you might decide to choose a corner of the map as the origin and take east as the positive x-direction and north as the positive y-direction, and 1 meter as the unit length. In this case, a Duckiebot with location (1,−2) would sit at 1 meter east and 2 meters south of the designated corner of the map.

**Figure 5.2.** A Cartesian coordinate system in the 2D plane

**Remark 7.** (Image space) It is customary to put the origin of an image at the top-left corner with the x-axis being horizontal and increasing to the right and the y-axis vertical and increasing downwards. In this way, the x and y coordinates index the column and row, respectively, of a particular pixel in the image. Such a convention is observed in OpenCV and other software libraries.

**Polar coordinate systems:**

An alternative coordinate system for the plane is the polar coordinate system, where we specify a point by its direction and distance from a fixed reference point. To set up a polar coordinate system, you first decide on the pole, the reference point, then the polar axis, the reference direction. We will call the distance from the pole, the radial coordinate or radius, commonly denoted by r or ρ, and the angle from the polar axis, the angular coordinate or polar angle, commonly denoted by φ, ϕ or θ. See an example in Figure 5.4.
Figure 5.4. A polar coordinate system with pole O and polar axis L.
Note that in a polar coordinate system, a point has many equally valid names.

**Check before you continue**
Exercise to readers: provide two such points and a few of their coordinates each.

Now, consider a Cartesian coordinate system $C$ whose origin is at the pole and its positive $x$-direction coincides with the polar axis. It is not hard to convert polar coordinates to Cartesian coordinates in $C$.

**Check before you continue**
Exercise to readers: consult Figure 5.6 and write out the conversion formulae.)

Figure 5.6. Converting from polar coordinates to Cartesian coordinates.
Given the many options, you might wonder which coordinate system to use in any given situation. The answer is a practical one. Choose the one that helps simplify the problem at hand. As a trivial example, consider the equation for a unit circle. In a Cartesian coordinate system, it would be $x^2 + y^2 = 1$ whereas in a polar coordinate system, it would be much simpler: $r = 1$.

**Check before you continue**
Exercise to readers: how about the equation for a straight line in polar coordinates?

3) 3D space

This is an important space since we live in a three-dimensional world. Since many of our robots operate in this same world, many robots similarly represent coordinates in three dimensions, including unmanned aerial vehicles (UAVs) and autonomous underwater vehicles (AUVs).

**3D Cartesian coordinate systems:**

Suppose the plane is the page or screen you are reading from, which is just a slice through the 3D space around it, we can extend a 2D Cartesian coordinate system on the plane to 3D by adding a z-axis that is perpendicular to the page, i.e., the z-, x-, and y-axes are mutually perpendicular. As done before, we need to choose a positive z-direction and there are two choices: coming out of the page or going into the page. They form the right-handed coordinate system or the left-handed coordinate system, respectively. We shall use right-handed coordinate systems unless otherwise noted. For more on handedness, see Wikipedia. Now the resulting coordinates become \((x, y, z)\), see Figure 5.8.

![Figure 5.8. A 3D Cartesian coordinate system.](image)

**Spherical coordinate systems:**

Similarly, we can extend the polar coordinate systems on the page or screen to 3D by defining a zenith direction (upwards) perpendicular to the plane, the polar angle to be the angle away from zenith, and the azimuth angle to be the orthogonal projection of a point’s angle away from the polar axis on the plane. Together, a point has coordinates \((r, \theta, \phi)\) where \(\theta\) denotes the polar angle and \(\phi\), the azimuth angle. See Figure 5.10.

**Check before you continue**

Exercise to readers: how to convert spherical coordinates to 3D Cartesian coordinates? and back?
Figure 5.10. A spherical coordinate system.

5.4. Further reading
If you find this topic interesting, there are many more coordinate systems than the ones covered here, such as the:

* cylindrical coordinate system in 3D space and

* parabolic coordinate system in 2D space.

Author: Falcon Dai
UNIT B-6
Reference frames

Check before you continue Required Reading: The following assumes a working familiarity with 2D and 3D Cartesian coordinate systems. If you are not familiar with Cartesian coordinate systems, please read the chapter on coordinate systems.

I will ignore this because it is an external link.

> I do not know what is indicated by the link '#preliminaries/coordinate_systems'.

Location not known more precisely.

Created by function check_if_any_href_is_invalid in module mcdp_docs.check_missing_links.

6.1. Motivation
Does Earth move around the sun or does the sun move around Earth? It turns out that this is an ill-posed question until we specify a frame of reference for measurement. For us, observers on Earth, it appears that the sun is moving. But for an observer on the moon, both Earth and the sun are moving. In general, motions are relative and we need a reference when measuring motion.

6.2. Definition
A reference frame, or just frame, is an abstract coordinate system and the set of physical reference points that uniquely specify the coordinate system (location of the origin and orientation). As a way of specifying the reference frame, we often say object $A$ is moving relative to object $B$, instead of specifying explicitly a reference frame $F_B$ that is attached to object $B$.

6.3. Example
Suppose there are two cars A and B both moving at 60 miles per hour eastward. When saying this, we implicitly assume the reference frame of the ground. In the ground’s reference frame, the ground itself is at rest, car A and car B are both moving at 60 miles per hour eastward. In car A’s reference frame, however, car B is at rest and the ground is moving 60 miles per hour westward!
6.4. Translating motions in one frame to another

To simplify the following discussion, we additionally assume that we choose reference frames such that their axes are parallel. In order to translate motions between reference frames, we assume two rules.

- Relative motions are equal in magnitude and opposite in direction. If frame $R$ is moving relative to frame $S$ at $u$, then frame $S$ is moving at $-u$ relative to $R$.
- Motions are additive. If a frame $T$ is moving at $u$ relative to frame $S$, and frame $S$ is moving at $v$ relative to frame $R$, then frame $T$ is moving at $u + v$ relative to frame $R$.

Check before you continue
Exercise to readers: derive these rules from kinematics.

As a corollary to the first rule, we immediately derive that frame $R$ is at rest relative to itself, since $0$ is the only vector is also its own opposite.

Check before you continue
Exercise to readers: translate car B’s motion relative to the ground to relative to car A.

Author: Falcon Dai
UNIT B-7
Transformations

Check before you continue Required Reading: The following assumes a working familiarity with 2D and 3D Cartesian reference frames. If you are not familiar with Cartesian reference frames, please read the chapter on reference frames (unknown ref preliminaries/reference_frames)

Previous warning next (2 of 3) index
warning
I will ignore this because it is an external link.

> I do not know what is indicated by the link '
#preliminaries/reference_frames'.

Location not known more precisely.
Created by function check_if_any_href_is_invalid in module mcdp_docs.check_missing_links.

Some familiarity with linear algebra (unknown ref preliminaries/linear_algebra)

Previous warning (3 of 3) index
warning
I will ignore this because it is an external link.

> I do not know what is indicated by the link '
#preliminaries/linear_algebra'.

Location not known more precisely.
Created by function check_if_any_href_is_invalid in module mcdp_docs.check_missing_links.

is also helpful.

7.1. Introduction
Transformations are functions that map points to other points in space, i.e. \( f : X \rightarrow X \). These maps are useful for describing motions over time. A particularly important class of transformations are linear transformations. These transformations can be represented by square matrices as they are linear and has the same domain and image.

7.2. Definitions and important examples
Please refer to the Robotics Handbook section 1.2, and in the context of this course, the reader’s goal is to attain a conceptual understanding, not necessarily knowing the exact formulae.
7.3. Applications
PART C
Probability
In this chapter we give a brief review of some basic probabilistic concepts. For a more in-depth treatment of the subject we refer the interested reader to a textbook such as [5].

1.1. Random Variables

The key underlying concept in probabilistic theory is that of an *event*, which is the output of a random trial. Examples of an event include the result of a coin flip turning up HEADS or the result of rolling a die turning up the number “4”.

**Definition 22.** (Random Variable) A (either discrete or continuous) variable that can take on any value that corresponds to the feasible output of a random trial.

For example, we could model the event of flipping a fair coin with the random variable $X$. We write the probability that $X$ takes HEADS as $p(X = \text{HEADS})$. The set of all possible values for the variable $X$ is its *domain*, $\mathcal{X}$.

In this case,

$$\mathcal{X} = \{\text{HEADS}, \text{TAILS}\}.$$  

Since $X$ can only take one of two values, it is a *binary* random variable. In the case of a die roll,

$$\mathcal{X} = \{1, 2, 3, 4, 5, 6\},$$  

and we refer to this as a *discrete* random variable. If the output is real value or a subset of the real numbers, e.g., $\mathcal{X} = \mathbb{R}$, then we refer to $X$ as a *continuous* random variable. Consider once again the coin tossing event. If the coin is fair, we have

$$p(X = \text{HEADS}) = p(X = \text{TAILS}) = 0.5.$$  

Here, the function $p(x)$ is called the probability mass function or pmf. The pmf is shown in Figure 1.2.
Here are some very important properties of $p(x)$: $0 \leq p(x) \leq 1$, $\sum_{x \in \mathcal{X}} p(x) = 1$

In the case of a continuous random variable, we will call this function $f(x)$ and call it a probability density function, or pdf.

In the case of continuous RVs, technically the $p(X = x)$ for any value $x$ is zero since $\mathcal{X}$ is infinite. To deal with this, we also define another important function, the cumulative density function, which is given by $F(x) \triangleq p(X \leq x)$, and now we can define $f(x) \triangleq \frac{d}{dx} F(x)$.

A pdf and corresponding cdf are shown in Figure 1.4. This happens to be a Gaussian distribution, defined more precisely in Subsection 1.1.8 - The Gaussian Distribution.

1) Joint Probabilities

If we have two different RVs representing two different events $X$ and $Y$, then we represent the probability of two distinct events $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ both happening, which we will denote as follows:

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

The function $p(x, y)$ is called joint distribution.
2) Conditional Probabilities

Again, considering that we have to RVs, $X$ and $Y$, imagine these two events are linked in some way. For example, $X$ is the numerical output of a die roll and $Y$ is the binary even-odd output of the same die roll. Clearly these two events are linked since they are both uniquely determined by the same underlying event (the rolling of the die). In this case, we say that the RVs are dependent on one another.

In the event that we know one of events, this gives us some information about the other. We denote this using the following conditional distribution

$$p(X = x \text{ GIVEN } Y = y) \equiv p(x|y).$$

Check before you continue

Write down the conditional pmf for the scenario just described assuming an oracle tells you that the die roll is even. In other words, what is $p(x|\text{EVEN})$?

If you think this is very easy that’s good, but don’t get over-confident.

The joint and conditional distributions are related by the following (which could be considered a definition of the joint distribution):

$$p(x, y) = p(x|y)p(y) \quad (1)$$

and, similarly, the following could be considered a definition of the conditional distribution:

$$p(x|y) = \frac{p(x, y)}{p(y)} \text{ if } p(y) > 0 \quad (2)$$

In other words, the conditional and joint distributions are inextricably linked: you can’t really talk about one without the other.

If two variables are independent, then the following relation holds: $p(x, y) = p(x)p(y)$.

3) Bayes’ Rule

Upon closer inspection of (1), we can see that the choice of which variable to condition upon is completely arbitrary. We can write:

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

and then after rearranging things we arrive at one of the most important formulas for mobile robotics, Bayes’ rule:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad (3)$$

Exactly why this formula is so important will be covered in more detail in later sections (TODO), but we will give an initial intuition here.

Consider that the variable $X$ represents something that we are trying to estimate but cannot observe directly, and that the variable $Y$ represents a physical measurement that relates to $X$. We want to estimate the distribution over $X$ given the measurement $Y$,
\( p(x|y) \), which is called the posterior distribution. Bayes’ rule lets us do this.

For every possible state, you take the probability that this measurement could have been generated, \( p(y|x) \), which is called the measurement likelihood, you multiply it by the probability of that state being the true state, \( p(x) \), which is called the prior, and you normalize over the probability of obtaining that measurement from any state, \( p(y) \), which is called the evidence.

**Check before you continue**

From Wikipedia: Suppose a drug test has a 99% true positive rate and a 99% true negative rate, and that we know that exactly 0.5% of people are using the drug. Given that a person’s test gives a positive result, what is the probability that this person is actually a user of the drug.

Answer: \( \approx 33.2\% \). This answer should surprise you. It highlights the power of the prior.

4) Marginal Distribution

If we already have a joint distribution \( p(x, y) \) and we wish to recover the single variable distribution \( p(x) \), we must marginalize over the variable \( Y \). The involves summing (for discrete RVs) or integrating (for continuous RVs) over all values of the variable we wish to marginalize:

\[
\begin{align*}
p(x) &= \sum_y p(x, y) \\
f(x) &= \int p(x, y)dy
\end{align*}
\]

This can be thought of as projecting a higher dimensional distribution onto a lower dimensional subspace. For example, consider Figure 1.6, which shows some data plotted on a 2D scatter plot, and then the marginal histogram plots along each dimension of the data.
Marginalization is an important operation since it allows us to reduce the size of our state space in a principled way.

5) Conditional Independence

Two RVs, $X$ and $Y$ may be correlated, we may be able to encapsulate the dependence through a third random variable $Z$. Therefore, if we know $Z$ then $X$ and $Y$ may be conditionally independent.

6) Moments

The $n$th moment of an RV, $X$, is given by $E[X^n]$ where $E[]$ is the expectation operator with:

\[ E[X^n] = \sum_{x} x^n f(x) \]
in the discrete case, and

$$E[f(X)] = \sum_x x f(x)$$

in the continuous case.

The 1st moment is the mean, $\mu_X = E[X]$.

The $n$-th central moment of an RV, $X$, is given by $E[(X - \mu_X)^n]$.

The second central moment is called the covariance, $\sigma_X^2 = E[(X - \mu_X)^2]$.

Another equation:

$$p(x | y) = \mathcal{X}$$

7) Entropy

**Definition 23.** The *entropy* of an RV is a scalar measure of the uncertainty about the value the RV.

A common measure of entropy is the *Shannon entropy*, whose value is given by

$$H(X) = -E[\log_2 p(x)]$$

(4)

This measure originates from communication theory and literally represents how many bits are required to transmit a distribution through a communication channel. For many more details related to information theory, we recommend [6].

As an example, we can easily write out the Shannon entropy associated with a binary RV (e.g. flipping a coin) as a function of the probability that the coin turns up heads (call this $p$):

$$H(X) = -p \log_2 p - (1-p) \log_2 (1-p).$$

(5)
Notice that our highest entropy (uncertainty) about the outcome of the coin flip is when it is a fair coin (equal probability of heads and tails). The entropy decays to 0 as we approach $p = 0$ and $p = 1$ since in these two cases we have no uncertainty about the outcome of the flip. It should also be clear why the function is symmetrical around the $p = 0.5$ value.

8) The Gaussian Distribution

In mobile robotics we use the Gaussian, or normal, distribution a lot. The 1-D Gaussian distribution pdf is given by:

$$
\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}
$$

where $\mu$ is called the mean of the distribution, and $\sigma$ is called the standard deviation. A plot of the 1D Gaussian was previously shown in Figure 1.4. We will rarely deal with the univariate case and much more often deal with the multivariate Gaussian:

$$
\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right]
$$

The value from the exponent: $(x - \mu)^T \Sigma^{-1} (x - \mu)$ is sometimes written $||x - \mu||_{\Sigma}$ and is referred to as the Mahalanobis distance or energy norm. Mathematically, the Gaussian distribution has some nice properties as we will see. But is this the only reason to use this as a distribution. In other words, is the assumption of
Gaussianicity a good one?

There are two very good reasons to think that the Gaussian distribution is the “right” one to use in a given situation.

1. The central limit theorem says that, in the limit, if we sum an increasing number of independent random variables, the distribution approaches Gaussian.

2. It can be proven (TODO:ref) that the Gaussian distribution has the maximum entropy subject to a given value for the first and second moments. In other words, for a given mean and variance, it makes the least assumptions about the other moments.

Exercise: derive the formula for Gaussian entropy.

9) Banana distributions

comment
The banana distribution is the official distribution in robotics! - AC

comment
The banana distribution is Gaussian! http://www.roboticsproceedings.org/rss08/p34.pdf - LP
PART D
Signal processing
**UNIT D-1**

**Time series**

**KNOWLEDGE AND ACTIVITY GRAPH**

- **Requires:** Knowledge of k:tuple
- **Results:** Knowledge of time series.
- **Results:** Knowledge of time series operations, such as upsampling and downsampling.

1.1. **Definition of time series**

**Definition 24.** A *time series* with domain $\mathcal{X}$ and time domain $\mathcal{T}$ is a sequence of tuples $\langle t_k, x_k \rangle \in \mathcal{T} \times \mathcal{X}$.

1.2. **Operations on time series**

1) **Up-sampling**
2) **Down-sampling**

1.3. **Further reading**

Super-dense time
Clustering is the process of grouping some objects such that similar objects belong to the same group. In the sense of colors it could be that similar colors are grouped together e.g. bright red, ruby and pink belong to the group of red colors whereas azure blue, Copenhagen blue and dark blue would be grouped to the blue colors. Various algorithm can solve such a task. They differ in how they define what is a cluster (e.g. the members are within a certain distance) and how efficiently these algorithms can find these clusters. Following some algorithms for determining clusters are presented.

### 2.1. k-Means clustering

1) **Description of the algorithm**

Let’s assume we have \( d \) data points \( x_1, \ldots, x_d \) and \( k \) cluster centers \( m_1, \ldots, m_k \). Now the algorithm tries to put the centers such that optimally all the clusters existing in the data are found. A data point \( x_i \) belongs to the cluster \( j \) if the cluster center \( m_j \) is the nearest among all the clusters \( m_1, \ldots, k \).

So the algorithm can be described as follows:

1. Define a number \( k \) (number of cluster centers), initialize \( m_1, \ldots, k \), for example randomly.
2. Determine for every data point \( x_i \) the nearest cluster center \( m_j \). Assign this data point to the cluster \( j \).
3. Recompute the cluster centers \( m_1, \ldots, k \) by finding the mean of all data points assigned to the cluster \( j \in 1, \ldots, k \).
4. Repeat step 2 and 3 until convergence. Often a certain tolerance is used such that the algorithm stops until the difference between the iterations is smaller than the tolerance. Another way to stop the algorithm is to set a maximum number of iterations beforehand.
2) Advantages and Limits

**Advantages:**
- Simple clustering method compared to other clustering methods.
- Unsupervised learning is possible. One doesn’t need any labeled data to perform a cluster analysis. Back to the color example from the introduction: The k-Means algorithm doesn’t know about the colors. The algorithm only knows that the data points are near each other in RGB-space for example and therefore assigns them to the same cluster.

**Limits:**
- There is no guarantee that k-Means converges to a global optimum. It depends on the initialization of the cluster centers in the beginning. Converging to local minima can lead to counter intuitive/”wrong” results.
- Computationally difficult: K-Means is in general NP-hard. The complexity scales with the number of data points \(n\), centers \(k\), iterations \(i\), and attributes \(a\) of each data point. (In the color example we have only one attribute. But it could be that there exist data points which have several attributes e.g. temperature, humidity, ...). So the complexity is in order of \(n \cdot k \cdot i \cdot a\). But in the worst case the complexity can reach super polynomial behavior.
- Clusters are expected to have similar size and spherical shape since k-Means decides whether a data point lies within a cluster based on the Euclidean distance.

2.2. Gaussian mixture models

We’ve seen above several drawbacks of k-Means clustering. That’s why there is motivation to search for something better.

1) Idea

Simply speaking Gaussian mixture models are a more general version of k-Means assuming we have $k$ components. This means we fit to each cluster a Gaussian distribution. The underlying assumption is that every data point is generated through a mixture of Gaussian distributions. Their parameters are unknown and are determined through the algorithm.

2) Algorithm
1. Initialize $n$ Gaussian mixture components with zero mean and identity covariance. (Often a diagonal covariance matrix is used.)
2. E-step: for each point, estimate the probability that each Gaussian generated it.
3. M-step: modify the parameters according to the hidden variable to maximize the likelihood of the data.
4. Repeat step 2 and 3 (expectation-maximization [EM]) until average gain in log-likelihood is below this threshold.

3) Advantages and Disadvantages

Positives:
- Does not have any underlying assumption of structure in the data. Compared to k-Means this is a clear advantage. Gaussian mixture models can handle elliptical data for example. The same counts for the size of the cluster.

Negatives:
- Difficult to interpret since a data point can belong to a certain degree to several clusters since affiliation is probabilistic.
- Complexity: The theoretical complexity of the GMM approach is similar as k-Means. So it results in $n \cdot k \cdot i$ for the number of data points $n$, centers $k$ and iterations $i$.
- Difficulties for high dimensional data.

Figure 2.6. K-Means in comparison with EM algorithm

[source]
This text will talk about histograms used for image analysis.

3.1. Motivation

Histogram Equalization can be used for contrast adjustment. For example when a gray scale picture has all values around one value the contrast mathematical difference between different pixels is in general really low. But with histogram equalization the colors are kind of normalized such that the difference and the therefore the contrast of a gray scale picture is increased.

3.2. Histogram

We'll assume in the following 8-bit values for every data point in a picture. One can create histogram by counting the number of values for each appearance of intensity in a channel. Such a histogram has bin size 1. So for a one channel picture (gray scale) we have different gray types between 0 and 255 (8-bit image). A RGB picture would have for every channel such a histogram.

3.3. Description of the Equalization

1) Problem description

So let’s assume we'll take a picture of a scene lightened with a perfect red light source. If we have a look at the red channel of the picture we would see a peak somewhere (potentially towards the higher values) in the histogram. Since through the red light every pixel is distorted by a certain added red light and the camera most probably overexposes for the red channel we'll end up with a peak in the red channel.

Figure 3.2. Distorted histogram
So if we want to fix that problem we could equalize this histogram.

2) Algorithm

The optimal idea is illustrated in the following picture:

![Figure 3.4. Basic idea](source)

To get to this flat shaped histogram one can follow the algorithm described here: 1. Create a normalized histogram consisting probabilities $p_i$ where $i = 1, \ldots, 255$ for a 8-bit image. The histogram describes how probable a certain value in the channel is: $p_i = \frac{\text{number of pixels with intensity } n}{\text{total number of pixels}}$. 2. Using that histogram one can equalize a picture by converting each pixel value $g_{i,j}$ of the picture with the following formula $g_{i,j} = \text{floor}((L - 1) \cdot \sum_{n=0}^{L} p_n)$. With this procedure the histogram will be flattened as seen in the following picture.

![Figure 3.6. Equalized histogram](source)

3.4. Limits of the algorithm

Histogram equalization is rather unproblematic in single channel images like gray scale. But if one has three color channels the problem is that histogram equalization changes the relative distributions of the channels. This can lead to unexpected and unwanted behavior.
A good idea to solve this problem is to use for example HSL instead of RGB space since you could equalize only luminance there. This would not change any hue or saturation value and provide a good white balance. HSV or LAB could be considered as well.

[source]
PART E

Misc computer science
Types

1.1. Types vs sets
Types are similar to sets in that they are collections that include some objects and exclude others. On a first glance, you can interpret $0 : \text{nat}$ as $0 \in \mathbb{N}$, reading the colon as membership in sets without any ill effect. But types and sets are the basic concepts belonging to two different formal systems, type theory and set theory, respectively. It is not essential to appreciate the difference in the scope of this course but for the curious readers, this section on Wikipedia can serve as a brief summary.

1.2. Example: product types
Given two types $A$ and $B$, we can construct the type $A \times B$, which we call their cartesian product. (Compare this with the similar concept of cartesian product of sets which is defined in terms of its elements and not a primitive.)

1.3. In Duckietown
As we will be using ROS, which models a robotic system as a network of communicating programs. In order to understand each other, all the communicating programs talk to each other in well-defined message types. Message types in ROS are product types composed of primitive types and other message types.

Check before you continue
Please read section 2 on ROS/msg page and answer: what are some primitive types in ROS? what are the fields and their types in message type Header?

Additionally, ROS/common_msgs page provides a list of pre-defined message types commonly used in robotics, such as Image (note how Header, a non-primitive type, is included in the definition) and Pose2D. As you have likely guessed, an RGB camera publishes Image messages, and a routing planning program might subscribe to the duckiebot's current position in duckietown, as represented in a Pose2D message and calculates the appropriate wheel actions.

1.4. Historical notes
Historically, the flexibility of naive set theory allows for some paradoxical sets such as a set that contains all sets that does not contain itself. Does this set contains itself? This
is known as Russell’s paradox which demonstrated that naive set theory is inconsistent. In response, Russell and colleagues developed type theory which demands all terms to be typed, i.e., to have a type, and used a hierarchy of types to avoid Russell’s paradox. Later, a subclass of type theories known as intuitionistic type theories internalized many key ideas in constructive mathematics and became a foundation for programming languages where computability is a major concern.

On a side note, this is not to say sets cannot serve as a formal foundation of mathematics. Russell’s paradox only shows that naive set theory is inconsistent. In fact, most working mathematicians today believe that the axiomatized Zermelo-Fraenkel set theory (together with the axiom of choice, usually abbreviated as ZFC) can serve as a “consistent” foundation of all mathematics.

1.5. Further reading

Type theory is a fascinating subject in itself and recently, Homotopy Type Theory (HoTT) captured a lot of research interest. For more on the subject consult HoTT website. The first chapter of the HoTT book also provides a reasonable introduction to type theory. For more practical applications of these abstract ideas, you may be intrigued by the field of formal verification, where software is verified by mathematical proofs against the formal specification, automatically.

Author: Falcon Dai
UNIT E-2

Computer science concepts

This unit will contain brief explanations / pointers to basic concepts of computer science.

2.1. Real-time Operating system