Chapter 5: Representation Theory

This chapter discusses when input or output data has a structure that a model can benefit from learning its representation using algebraic theories. Representation learning can be unsupervised or semi-supervised to guide supervised learning algorithms and increase their accuracy. The chapter starts with theoretical discussions of abstract algebra foundations required, then how group representation theory is applied in traditional machine learning and neural networks and then develops to its role in tensor products and their decompositions.

Chapter one introduced machine learning (ML) algorithms using the linear regression example that estimates the weights of an equation that maps the target variable that could be given in a dataset if prelabelled (supervised) or a state or cluster membership (unsupervised) to the other independent variables of the dataset. This equation, f(x), is called mapping or transformation. Chapter two introduced the importance of dimensionality reduction using projections to orthogonal spaces or embedding or manifold learning. Chapters three and four introduced how this can be done for the multi-way structures using tensor decompositions methods. This chapter introduces representation discovery or learning, which is also a dimensionality reduction and regularisation approach achieved by a change of coordinate systems or basis functions. Dimensionality reduction or representation learning are considered data pre-processing steps to perform before using an ML algorithm. Parametric ML algorithms estimate the input/output mapping function using pre-defined parameters by assuming a particular representation or data distribution, such as fitting a linear or non-linear equation of a given polynomial degree. This is usually attempted using trial and error until a good measure of fitness is achieved. Data visualisation can also help to estimate the appropriate representation manually. The non-parametric machine learning algorithms introduced in the SVM Kernel trick in chapter two attempt to learn the representation from the data using kernel methods. This learning will add computational overhead due to permutations having n! elements. However, symmetries that are often found in datasets will reduce the computational overhead. A symmetrisation step can be included as a dimensionality reduction or compression technique with an acceptable loss and work as regularisation. Furthermore, the variables in x can be members of one or more groups in the dataset. These groups can be reducible to subgroups and have properties that make learning this equation's coefficients easier. Algebraic group theory, Lie theory and representation theory (often mean the same thing) and their properties capture symmetries

and are often used in the various machine learning algorithms. This will facilitate our understanding and ability to apply and advance these algorithms as required.

The first section introduces abstract algebra as a recent player in machine learning algorithms to achieve modular programming that benefits from symmetries and group decomposition representation, along with python examples. Computer scientists are already familiar with object-oriented inheritance and polymorphism properties. These concepts enable modular programming such that extra functionality is added or existing functionality is eliminated during inheritance from superstructures to substructures. This enables polymorphic methods that operate differently based on the structure of their parameters. This is how algebraic objects are related in a tree structure of inheritance and polymorphism with different operations and properties.

Then, the second section progresses into Harmonic analysis using Fourier Transforms and Wavelet analysis that discovers the representation basis of a given dataset. The Laplace operator will be further explained for its role in revealing symmetries. Then a section on learning in the Hilbert space using kernel methods will discuss how this can also be performed on group representations of datasets. A section on invariance summarises how all the above methods are actually learning the new invariant subspace basis representing a dataset input or output spaces. A final section on applications of these approaches will discuss various representation learning approaches, in which one approach uses tensor decomposition methods presented in chapters three and four.

The last chapter presented the BSS problem and how it is solved using tensorisation to a higher-order object from which the symmetries and the unique properties of the sinusoidal signals are used to solve the separation. The topics of this chapter will explain another geometric definition of tensors as positive definite symmetric matrices mentioned in chapter one. To understand symmetries, we need to learn abstract algebra. Abstract algebra, as its name suggests, abstracts not only Algebra but also many different tools used in other math topics such as geometry, number theory and topology, such that the same abstract tools can be used in all of them. In algebra, we learned how to solve a system of equations, whether linear, quadratic, cubic or even quartic (4^{th} order) equations. For higher degree equations, $n>4$, we need to understand groups and their role in modular arithmetic and in understanding non-Euclidean geometries. Groups are fundamental objects to many abstract objects that build upon them, such as rings, fields, vector spaces and modules. We covered fields and vector spaces in the previous chapters and will focus on the remaining concepts. The main aim is to understand how machine learning algorithms map datasets such as collections of images, problem states in Reinforcement Learning (RL), or words in natural language processing (NLP), into an implicit vector space and how mappings to output space can be achieved using algebraic methods.

This chapter, again, is full of mathematical definitions and equations. Once again, the aim is not to discuss the derivations, proofs, or all properties. The aim is to familiarise the reader with the wealth of mappings between groups, subgroups, symmetries, reducing groups,

composing groups, rings, and modules. Machine learning algorithms employ these concepts by taking advantage of equivalence between computationally expensive structures and their equivalent or approximate representations reducing the computational overload. The material covered in this chapter is summarised from (Gilmore, 2005), (Risi Kondor, 2008), (Milne, 2021), (Mahadevan, 2008), and many examples from the Socratica YouTube channel. This chapter cannot be a source to study these concepts thoroughly, but it can serve as an appetiser to expand the reader's horizon and create a unified ML context of these topics.

5.1 Group and Representation Theory

Group theory is a branch of abstract algebra that studies groups satisfying certain axioms, such as the representation theory of symmetric groups.

5.1.1 Group Theory Main Structures

5.1.1.1 Sets

A set is a collection of objects that do not necessarily have any additional structure or properties.

5.1.1.2 Group

A group G is defined as a set of elements such as x, y, with an operation $G \times G \to G$ producing results as xy or $x.y$ as the product of the elements. Multiplication $\times$ and addition $+$ are the most used group operations since division can be rewritten as multiplication with reciprocals, and subtraction can be written as addition with negatives. The operation can be generalised as * for abstractness. A finite group is described by its **order** (number of elements or **cardinality**) as $|G|$ and can be continuous or discrete. The axioms to be satisfied include **closure** such that the product $xy \in G$ for a group with multiplication operation, or $x + y \in G$ for a group with addition operation, and generalised/abstracted to $x * y \in G$; **associativity** such that for any $x, y, z \in G$, also $(xy)z = x(yz) \in G$ in multiplication operation or $(x + y) + z = x + (y + z) \in G$ in addition operation, and abstracted to $(x * y) * z = x * (y * z) \in G$**; identity element** such that $ex = xe = x \in G$, which is 1 in multiplication operation and 0 for the addition operation, and abstracted as $e * x = x * e = x \in G$; and **inverse** such that $xx^{-1} = e$ for multiplication operation such that the reciprocal is the inverse or $x + (-x) = e$ for addition operations such that the negative value is the inverse and abstracted as $x * (x^{-1}) = e$. Commutativity as in $(x * y) = (y * x)$ property if existing in a group, then it is an Abelian group. Symmetric groups are not Abelian except when their order n is less than or equal to two. A group can be defined by listing its elements and multiplication table or a table showing other operations called Cayley table.

For example, The Integer numbers group $\mathbb{Z}$ is an infinite group closed under the addition operation to maintain all the above axioms. Given $x, y, z \in \mathbb{Z}$, **closure**: when x=2, y=3, and z=4, then x+y=2+3=5 ∈ $\mathbb{Z}$; **associativity** $(x + y) + z = x + (y + z)$ is $(2 + 3) + 4 = 2 + (3 + 4) = 9 \in \mathbb{Z}$, **identity element** 0, such that $e * x = x * e = x \in G$ is $0 + 2 = 2 + 0 = 2 \in G$; and **inverse** $x + (-x) = e$ is $2 + (-2) = 0$**.** We can solve an equation using these axioms, such as:

x+3=5	Integer Under addition
(x+3)+(-3) = 5+(-3)	Inverse
(x+3)+(-3) = 2	Closed Under addition
x+(3+(-3)) = 2	Associativity
x+0=2	Identity
x=2	

Other examples for different orders are shown in the following table, starting from order one, the trivial group, which would contain the identity element only. For the different orders, the multiplication/Cayley table starting from the identity element is shown in the table:

Order 1

	e
e	e

Order 2 ≅ $\frac{\mathbb{Z}}{2\mathbb{Z}}$(integers mod 2)

	e	a
e	e	a
a	a	e

Order 3, which is the only one ≅ $\frac{\mathbb{Z}}{3\mathbb{Z}}$(integers mod 3), isomorphic groups

	e	a	b
e	e	a	b
a	a	b	e
b	b	e	a

There are only four groups of order four; three of them are equivalent, which means only two order four groups up to isomorphism both are Abelian of prime power order: 1) Cyclic group of order four, 2) Klein Viergruppe four group, the product of the quotient group $\frac{\mathbb{Z}}{2\mathbb{Z}}$ with itself (the group of order two explained above multiplied with itself).

The cyclic group of order 4, using additive notation

	0 (identity)	1 (generator)	2	3 (generator)
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

The cyclic group of order 4, using multiplication notation

	e (identity)	x (generator)	x^2	x^3 (generator)
e	e	x	x^2	x^3
x	x	x^2	x^3	e

Klein Viergruppe V as isomorphic to: $\frac{\mathbb{Z}}{2\mathbb{Z}} \times \frac{\mathbb{Z}}{2\mathbb{Z}}$

	e	a	b	c
e	e	a	b	c
a	a	e	c	b
b	b	c	e	a
c	c	b	a	e

x^2	x^2	x^3	e	x
x^3	x^3	e	x	X^2

Notice that in all the previous Cayley tables, every row and every column should contain the **identity** element once because of the existence of the **inverse** of every element. **No duplicate** element in any row or column because if there is any, then there is an equivalence (redundancy) between the group elements. If the table is symmetric (if you flip it around the diagonal, you get the same table), then the group is Abelian (commutative) such that a*b=b*a.

5.1.1.3 Rings

A ring extends the group by being a set of elements defined with both operations $+/-$ and $\times$, such that the $+$ has inverses, which are the negative numbers such that the subtraction operation is also included, but the $\times$ has no inverses as the reciprocals are not included in the ring. Rings are commutative under $+$, which is Abelian under addition, and associative under $\times$, such that $a \times (b \times c) = (a \times b) \times c$. Rings' distributive property links both addition and multiplication operations: $a \times (b + c) = a \times b + a \times c$. Elements can be generalised to polynomials and matrices. If a ring is commutative under multiplication, it is a commutative ring. If G has identity 1, it is called a "ring with identity" because rings naturally have zero additive identity. A **Unit** in a ring R is the element $x \in R$ that has a multiplicative inverse $x^{-1} \in R$, such that $x.x^{-1} = 1$. All units in a ring form the **group of units** $R^{\times}$ which is a group under multiplication. In $\mathbb{Z}$ ring, the group of Units is $\mathbb{Z}^{\times} = \{1, -1\}$ since all other multiplicative inverses will yield a fraction. In $\frac{\mathbb{Z}}{12\mathbb{Z}}$ ring, the group of Units is $\left(\frac{\mathbb{Z}}{n\mathbb{Z}}\right)^{\times} = \{1, 5, 7, 11\}$; this can be verified from the Cayley table of the ring. Any integer $\mathbb{Z}$ multiplied by the units' set yields associates, such as for $\mathbb{Z}^{\times}$, the associates are {2, -2}, {3, -3} and so forth. Associates are important in identifying equivalent factorisations and ignoring unit factors. The fundamental theorem of Arithmetic states that every integer n except 0, 1, and -1, has a prime factorisation that is unique up to order and associates. For example, a ring $R = \left\{ \begin{bmatrix} a & b \\ c & d \end{bmatrix} \middle| a, b, c, d \in \mathbb{Z} \right\}$ has a unit matrix $\begin{bmatrix} 2 & 7 \\ 1 & 4 \end{bmatrix}$, to which the multiplicative inverse (the matrix inverse) = $\begin{bmatrix} 4 & -7 \\ -1 & 2 \end{bmatrix} \in \mathbb{Z}$, and identity matrix $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Not all matrices in R have multiplicative inverses and the group of units $R^{\times}$= matrices A with determinant (A)= ±1.

An example of infinite rings is the group of Integer numbers $\mathbb{Z}$ under $+, -, \times$, such that addition/subtraction/multiplication of two integers yields an integer, but division yields a fraction, which is not an element of $\mathbb{Z}$. Another example is the infinite group of polynomials $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$ such that addition/subtraction/multiplication of two polynomials yield a polynomial, but division

yields a non-polynomial. The polynomials ring is a commutative ring with identity. Coefficients can be integers, complex numbers, $\frac{\mathbb{Z}}{n\mathbb{Z}}$, matrices, or any ring. Rings can be used in composing rings, such as a ring of polynomials in which the coefficients are rings. A ring of matrices is not commutative because a change in the order in matrix multiplication yield different results.

The quotient group example of $\frac{\mathbb{Z}}{2\mathbb{Z}}$ take the group/ring $\mathbb{Z}$ and divide it by the **normal subgroup/ideal** (both will be defined below) $2\mathbb{Z}$ yielding finite rings. This class of finite rings containing $\frac{\mathbb{Z}}{n\mathbb{Z}}$ groups, except when n is prime p, then $\frac{\mathbb{Z}}{p\mathbb{Z}}$ is a field, which is also a ring. Every field is a ring, but not every ring is a field as shown in Figure 1.

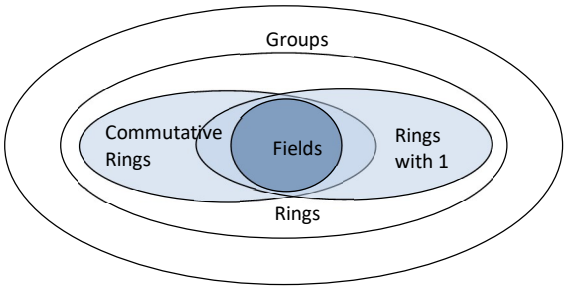


Figure 1: Groups, Rings, Fields:

5.1.1.4 Fields

Fields are defined in chapter one and redefined in chapter three. Fields $\mathbb{F}$ extend the abstract algebra objects as elements on which the four algebraic operations are defined, $+, -, \times, \div$. For abstract algebra generalisation, these operations are redefined as addition, additive inverses, multiplication, and multiplicative inverses. For example, real numbers of fractions $\mathbb{R}$, integer numbers $\mathbb{Z}$, natural numbers $\mathbb{N}$, rational numbers of decimals $\mathbb{Q}$, complex numbers $\mathbb{C}$, and prime fields $\frac{\mathbb{Z}}{p\mathbb{Z}}$ such as $\frac{\mathbb{Z}}{2\mathbb{Z}}$, $\frac{\mathbb{Z}}{5\mathbb{Z}}$, …, etc., are illustrated in Figure 2, such that ($\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{F}$), and ($\mathbb{C} \subset \mathbb{F}$).

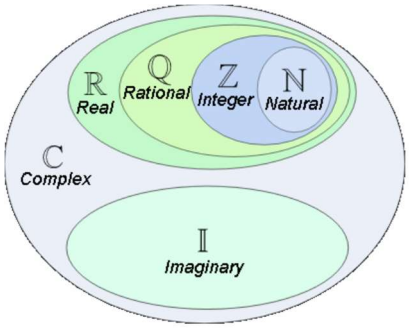


Figure 2:Fields of Numbers as sets/groups/rings, from https://www.mathsisfun.com/sets/number-types.html

These are all infinite fields because they have all the properties of rings with identity, commutative under multiplication, and include multiplicative inverses. Fields are defined as two groups combined: field $\mathbb{F}$ under addition $\langle \mathbb{F}, + \rangle$ as a commutative group, and Field $\mathbb{F}$ under multiplication $\langle \mathbb{F}^{\times}, \cdot \rangle$ as a commutative group, linking addition and multiplication by the distributive property. Infinite fields can be extended, forming other fields. The integer group $\mathbb{Z}$ can be extended with multiplicative inverses including fractions to become the field $\mathbb{Q}$; extend rational numbers field $\mathbb{Q}$ with numbers in the form of $\sqrt{x}$. If we extend $\mathbb{R}$ with an imaginary number to solve equations such as $i^2+1=0$ that needs i^2 to be = -1, and hence $i=\sqrt{-1}$, then this defines the complex numbers field $\mathbb{C}$ as a 2D numbering system. as $\mathbb{C} = \mathbb{R} + i_1\mathbb{R}, {i_1}^2 = -1$, which is the largest field that can not be extended, and it is algebraically closed, such as any polynomial equation in $\mathbb{C}$ can be solved in $\mathbb{C}$. $\mathbb{C}$ can be extended by variables such as $\mathbb{C}(x)$ or multivariable such as $\mathbb{C}(x_1, \ldots, x_n)$. Just as much as any vector in $\mathbb{R}^n$ is a linear combination of n basis vector e_i as $v=v_1e_1+ \ldots + v_ne_n$, also any $\mathbb{C}$ vector is a linear combination of bases, except that each v_i, has 2 components as $v_i=v_{i0}+v_{i1}i$. This means 2N-dimensions are needed to represent a N-dimension $\mathbb{C}$ in $\mathbb{R}$. A change of bases in $\mathbb{R}$ using an invertible transformation matrix A is applied as $e_j = A_j^i e_i$ and the inverse mapping is as $e_i = \left(A_j^i\right)^{-1} e_j = A_i^j e_j$. The same change of basis rules applies to $\mathbb{C}$.

We can construct the quaternions Q from the complex numbers $\mathbb{C}$ as $Q = \mathbb{C} + i_2\mathbb{C}$, such that ${i_2}^2 = -1, i_1 i_2 = -i_2 i_1$, this makes the quaternions a 4D numbering system that can be expressed as $Q = \mathbb{R} + i\mathbb{R} + j\mathbb{R} + k\mathbb{R}$ such that $i^2=j^2=k^2=ijk=-1$, and can be represented as an ordered pair binary form of real and vector parts $Q = [s \in \mathbb{R}, v \in \mathbb{R}^3]$. The multiplication operation is a non-commutative quaternion product with multiplication identity one, defined similar to the cross product of unit cartesian vectors as follows:

	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

A Cayley field can be constructed from quaternions, similarily the 3D rotation groups as explained below. Similar modular construction applies such as quaternions can have a change of basis and be represented as Complex numbers $\mathbb{C}$ using 2N dimensions and real $\mathbb{R}$ using 4N dimensions. Further expansions such as the Octonions are possible, but they are non-associative and hence not a Field. With every expansion, some properties are lost. Convergent sequences Limits can also be used to limit $\mathbb{Q}$ to $\mathbb{R}$, for example. Finite fields are

Galois fields containing a finite number of elements, such as the integer mod p where p is a prime number.

5.1.1.5 Vector Spaces

Vector spaces are defined in chapter one as well. They have the properties of allowing addition and scaling operations and commutative, including the zero vector, identity element, inverses, and associative. These properties generalise vector spaces as commutative groups under addition operation with the additional operation of scaling with its distributive property. Even scalars can be generalised abstractly from any given Field, not necessarily real numbers. This redefines the vector space to a commutative/Abelian group V containing vectors v as elements, under + operation, that has a field of scalars F, if $v \in V, f \in F \Rightarrow f.v \in V$ such that $f.v$ is a scaled vector with distributive property $f.(v_1 + v_2) = f.v_1 + f.v_2$, and $(f_1 + f_2).v = f_1.v + f_2.v$, associative property $f_1.(f_2.v) = (f_1 \times f_2).v$ and scaling identity 1. For example, a polynomial of degree 5 or less is a vector space V such that if adding two polynomials, always keep or reduce the polynomial degree by cancellations, but never increase the degree. Polynomials can be scaled, keeping the degree. Another example of a vector space is the space of continuous functions. Vector spaces are of finite dimensions determined by n, with basis vectors $\{e_1, \dots, e_n\}$, such that Vector components are defined as $V = f_1 e_1 + \ \dots + f_n e_n$such that $f_i \in$ field of scalars F, which makes V isomorphic to F^n: $V \cong F^n$.

5.1.1.6 Modules

The vector spaces are generalised to Modules, such that scalars are not Fields but rings, and the elements of a Module do not need to be vectors. This defines Module M to be an Abelian group of m elements, with Ring R scalars of r elements, such that r.m is a scaled element with distributive property $r.(m_1 + m_2) = r.m_1 + r.m_2$, and $(r_1 + r_2).m = r_1.m + r_2.m$, associative property $r_1.(r_2.m) = (r_1.r_2).m$ and scaling identity 1. For example, M is any Abelian group of elements a, with R = $\mathbb{Z}$ scalars, on which the scalar multiplication is defined r.a ∈ M. Every Abelian group is a $\mathbb{Z}$-module. Another example is a Module M of elements as 2x3 matrices = $\left\{\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} | m_{ij} \in \mathbb{Z}\right\}$, with R = $\mathbb{Z}$ scalars, and scalar multiplication:

$$r.\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} = \begin{bmatrix} r.m_{11} & r.m_{12} & r.m_{13} \\ r.m_{21} & r.m_{22} & r.m_{23} \end{bmatrix}$$

A third example shows that Modules with vector elements can do more than vector spaces, is when the scalars are not defined in $\mathbb{Z}$ but as a Ring of matrix $R = \left\{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} | a_{ij} \in \mathbb{R}\right\}$ which are not sure to be invertible and not commutative but have an identity matrix

3x3. When M= $\mathbb{R}^3 = \{(x, y, z) | x, y, z \in \mathbb{R}\}$ under +, the scalar multiplication r.m = matrix multiplication. Modules can be isomorphic to n copies of Ring scalars R similar to vector spaces, which is called finitely generated. Another finitely generated module that is not n copied of R, is $\mathbb{Z} \times \left(\frac{\mathbb{Z}}{2\mathbb{Z}}\right) \times 4\mathbb{Z}$. In addition, Modules can be infinitely generated, such as the free Modules. This complexity of Modules enables the decomposition of scalar Rings that we can not do with Fields as scalars in Vector spaces.

5.1.1.7 Algebras

A linear algebra A consists of a collection of vectors $v_1, v_2 \ldots, \in V$, a collection of fields $f_1, f_2 \ldots, \in \mathbb{F}$, a field, and three kinds of operations (α) vector addition, (β) scalar multiplication, and (γ) vector multiplication. For the vector space collection, closure, associativity, identity, inverse, and commutativity hold. Other algebras can be defined based on which properties they hold. For example, an algebra of a set of real n × n matrices forms a real n^2-dimensional vector space under matrix addition and scalar multiplication. An associative algebra adds matrix multiplication to the previous algebra. The first algebra example has a subspace of nxn symmetric matrices $(S_{ij})^T = S_{ji} = +S_{ij}$. Symmetric matrices matrix multiplication does not yield symmetric matrices output. Symmetrisation of two symmetric matrices, S, and T, is defined as: $[S, T]_+ = ST + TS$. The anti-commutation relation is $[S, \alpha T_1 + \beta \mathrm{T}_2]_+ = \alpha[S, T_1]_+ + \beta[S, T_2]_+$. A symmetrisation operation and anti-commutation form an algebra from symmetric matrices. An Anti-symmetrisation operation of two antisymmetric matrices A, and B, such that $A^T = -A,\ A_{ij} = -A_{ji},$ is defined as: : $[A, B] = AB - BA$. The commutation property is defined as $[A, \beta \mathrm{B}_1 + \gamma \mathrm{B}_2] = \beta[A, B_1] + \gamma[A, B_2]$. The antisymmetric multiplication and the commutation relations form an algebra called Lie Algebra from antisymmetric matrices, provided that the combinatorial antisymmetrisation operation obeys $\big[A, [B, C]\big] = [\,[A, B], C] - \big[B, [A, C]\big]$, which identifies the Jacobi identity: $\big[A, [B, C]\big] + \big[C, [A, B]\big] + \big[B, [C, A]\big] = 0$. If an Algebra has a norm and a division, it is called a normed division algebra, in which only four exist, the real numbers $\mathbb{R}$, the complex numbers $\mathbb{C}$, the quaternions Q, and the octonions O.

5.1.2 Main Concepts & Definitions

The algebraic structures (group, field, and others mentioned previously) can be mapped into another similar algebraic structure, keeping some or all of its structural properties. A realisation is a mapping into an algebraic structure that can be written down concretely and described analytically. A representation is a mapping into a set of matrices.

The following describes the main concepts and definitions from which many algorithms benefit, such as the conjugate gradient (CG) method (https://scipy-lectures.org/advanced/mathematical_optimization/) used in applying the tensor completion

example presented at the end of chapter three. Many algorithms employ these equivalent decompositions to achieve faster computation.

Let us start by defining conjugacy. Two group elements, x and y, are said to be **conjugate**, which means equivalent if there is a $t \in G$ such that $t^{-1}xt = y$. This can partition group G into **conjugacy classes**. For any two x, and y conjugate vectors, any vector parallel to x and y, is also conjugate. The conjugate gradient algorithm uses conjugacy to reduce the number of iterations required to minimise a quadratic equation. Being linearly independent is being orthogonal in a vector space, but conjugacy means a transformation can make two vectors orthogonal in another space (Haykin, 2009).

Hermitian conjugacy, denoted as A^*, A^H or $A^\dagger$, is complex square matrices that are equal to their conjugate transpose $A = A^{*T}$. The symmetry is captured such as an element, at i,j indices, is equal to its conjugate, at j, i indices: $a_{ij} = a_{ji}^*$. The adjoint of an operator is the infinite-dimensional generalization of conjugate transpose, where you find the transpose of an operator (in matrix form this is done by $A_{ij}^T = A_{ji}$ and then take the complex conjugate of it. This can be done in any order $(A)_{ij}^* = ((A)_{ij}^T)^* = ((A_{ij})^*)^T = A_{ji}^*$.

We need to understand subgroups and simple groups to prepare for group decompositions. **Subgroups** are defined as $H \leq G$, when elements of H are subsets of G, and H has a group invertible operation. Any group G has at least two **normal subgroups**, G is a subgroup of itself, and the **trivial group** containing one identity element $\{e\}$ is a subgroup of any group. A **proper subgroup** is $H < G$, such that both elements are not equal. **Lagrange theorem** states that the order of H divides the order of G: given $H \leq G \Rightarrow |H|$ divides $|G|$. For example, if the number of elements in G, $|G| = 323$, the factors dividing 323 are 1, 17, 19, and 323. The subgroup H containing one element is the identity element, G itself is the subgroup containing all its elements, and there might be other two subgroups containing 17 and 19 elements, respectively.

A **simple group** has no other groups except the identity group and is a building block for other groups, such as prime numbers to the Integers group. There are four classes of simple groups defined as follows:

1. An Abelian group of Integer mod p group under addition: $\frac{\mathbb{Z}}{p\mathbb{Z}}, +$, where p is a prime number, which leads to having only two subgroups, the trivial identity group {0}, and the whole group $\frac{\mathbb{Z}}{p\mathbb{Z}}$, which is the definition of a simple group. Because prime numbers are infinite, there is an infinite number of groups in this class.
2. The alternating non-Abelian group A_n was proven to be simple for n≥5, and will be further explained while discussing normal series and their role in solving polynomial equations.
3. Groups of Lie type will be further explained below.
4. 26 Sporadic Groups do not fit into any category, among which a monster group contains 20 out of the 26 groups.

Conjugate subgroup H^t is isomorphic of subgroup H of group G for any element $x \in G$. H^x is defined as H^x: $\{x^{-1}hx | h \in H, x \in G.\}$. If $H^x = H$, then H is a **normal subgroup or self-conjugate subgroup** $H \trianglelefteq G$. A normal subgroup H of group G is defined such that H: $\{x^{-1}hx | h \in H, x \in G.\}$, where x may or may not be in H. All subgroups of cyclic groups are normal, and one subgroup of Abelian groups is normal. An example normal subgroup is $n\mathbb{Z} \trianglelefteq \mathbb{Z}$, such as $2\mathbb{Z} \trianglelefteq \mathbb{Z}$.

In addition, measures of equivalence and similarity will make group decomposition feasible. Isomorphism, homomorphism, kernels, automorphism and isometry will be defined next. **Isomorphism** is when two groups (having similar structures/equal form) $G \cong G'$ has one-to-one mapping $\emptyset$: $G \to G'$, such that $\emptyset(x)\emptyset(y) = \emptyset(xy)$ for all $x, y \in G$. This leads to considering both groups as the same group. For example, two groups: G the finite integer mod 4 with addition operation: $\frac{\mathbb{Z}}{4\mathbb{Z}}, +$, and the group H containing elements {1, -1, i, -i} with multiplication operation. The Cayley tables for both groups are defined as follows and coloured for similarity, such as the identity element is coloured red, the second element 1 in the first group highlighted in green, needed a swap between the second and third elements in the second group to match the structure. The third element is highlighted in blue, and the last element is highlighted in magenta. The colour map shows that both groups have the same structure and are isomorphic. Abstractly they are the same group, regardless of the different operations and elements.

G: $\frac{\mathbb{Z}}{4\mathbb{Z}}, +$

	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

H: $\{1, -1, \mathrm{i}, -\mathrm{i}\}, \times$

	1	i	-1	-i
1	1	i	-1	-i
i	i	-1	-i	1
-1	-1	-i	1	i
-i	-i	1	i	-1

Homomorphism is when mapping between two groups $\emptyset$: $G \to H$ preserves the same structure, such that $\emptyset(x)\emptyset(y) = \emptyset(xy)$ for all $x, y \in G$, but without one-to-one mapping. A homomorphic group is an **embedding** such as a subgroup or a **quotient map**. For example, the infinite integer group with addition operation: $\mathbb{Z}, +$, and the finite integer mod 2 with addition operation: $\frac{\mathbb{Z}}{2\mathbb{Z}}, +$, containing two elements {0, 1}. The first group: $\mathbb{Z}, +$, can be broken into two sets: $\mathbb{Z} = \{evens\} \cup \{odds\}$, such that the operations in the first columns in the following table are defined, and the operations for the second group $\frac{\mathbb{Z}}{2\mathbb{Z}}$, are defined in the second column. If you replace even with zero and odd with 1, then both columns are the same.

$\mathbb{Z}, +$ divided into $\mathbb{Z} = \{evens\} \cup \{odds\}$	$\frac{\mathbb{Z}}{2\mathbb{Z}}, +$

even + even = even	0+0=0(mod 2),
even + odd = odd	0+1=1(mod 2),
odd + even = odd	1+0=1(mod 2),
odd + odd = even	1+1=0(mod 2)

We can define a mapping function between these two groups: $f: \mathbb{Z} \to \frac{\mathbb{Z}}{2\mathbb{Z}} = \begin{cases} 1 \; if \; \mathbb{Z} \; is \; odd \\ 0 \; if \; \mathbb{Z} \; is \; even \end{cases}$.

To generalise this concept to any two groups $G,*$ and $H,\diamond$ with abstract operations, we attempt defining a mapping function between both groups $f: G \to H$ such that for $x, y \in G$, we have $f(x), f(y) \in H$, and $f(x * y) = f(x) \diamond f(y)$, sending identities to identities, and inverses to inverses. In the isomorphism example, the mapping for identities is $f: 0_G \to 1_H$ and the mapping for inverses is $f: 1_G \to i_H, f: 3_G \to -i_H$, and this is the inverse of the identity $f: 2_G \to -1_H$: This function does not need to be one-to-one/injection nor onto/surjection in homomorphism, but needs to be both 1-1 and onto, which is a bijection for isomorphism. The mapping function does not need to be invertible for both testing for homomorphism and isomorphism. If we can construct such a mapping, these two groups are homomorphic, and if they are of the same cardinality, they are isomorphic. This is very important to identify similar and identical groups and their fundamental building blocks. This is what we did in chapter two to factorise a matrix to decompose into dominant submatrices such as SVD, and we did it in chapters three and four for tensor factorisation/decomposition as well.

A Kernel of a group homomorphism $f: G \to H$ measures how similar these two groups are, i.e. how far the mapping function is far from a one-to-one function. This is done by mapping all elements in G to the identity element in H. Different mapping functions have different kernels. If f is an isomorphism, then the kernel is the identity element. If f is not an isomorphism, then at least two elements in G map to the identity element in H. This makes the definition of the kernel be: $ker(f) = \{x \in G | f(x) = 1_H\}$. Here is at least one mapping of identities both ways in the kernel in the case of isomorphism as a one-to-one mapping, and there will be more than or equal to two mappings in the kernel in the case of homomorphism. This defines the kernel as a subgroup of G.

Automorphism is when a group is mapped to itself, such that the set of automorphisms of G forms a group under composition, denoted Aut(G). For example, all permutations of a graph, such as the connections of its vertices, remain the same, i.e., maintaining the same graph topology, which is called the automorphism group of this graph.

Isometry groups are the group of transformations of a metric space which leave the metric invariant.

5.1.3 Group Types

Infinite but countable groups, such as the set of integers in $\mathbb{Z}$ under addition operation, and the **cyclic group** of order n, denoted $\mathbb{Z}_n$, or $\frac{\mathbb{Z}}{n\mathbb{Z}}$ or C_n. **The order of an element** $x \in G$, is the smallest positive integer n such that $x^n = e$ and is defined as $|x| = n$. When there is no value for n, then the order of the element is infinite, making the group infinite. For multiplication group on non-zero Real numbers $\mathbb{R}$, the order of the identity element $|1| = 1$, because it has no inverse, while $|-1| = 2$ because $(-1)^2 = 1$. Other elements in this group have infinite orders.

Uncountable Topological groups, such as Group G with topology $f(x) = x^{-1}$ as a continuous map $f : G \rightarrow G$; and $g(x, y) = xy$ as a continuous map $g: G \times G \rightarrow G$. A topology is a system of G subsets called open sets that obey certain axioms.

A compact group is a topological group of open sets that can be covered/bounded with a subset of its elements. A locally compact group is a topological group in which any group element has a compact neighbourhood. A space T is compact if every infinite sequence of points $t_1, t_2, \ldots, (t_i \in T)$ contains a subsequence of points that (a) converges to a point and (b) this point is in T. For example, the real line $\mathbb{R}^1$ is not compact because the sequence of points $t_i = i$, i = 1, 2,… does not have a convergent subsequence. The circumference of the unit circle in $\mathbb{R}^2$ is compact. The interior of the unit circle is not compact because the sequence of points $t_n = 1 - 1/n$, n = 1, 2,…, converges to a point on the circumference and, therefore, is not in the original set.

Abelian groups are commutative groups such that $xy = yx$, which are much simpler than the non-commutative groups.

Symmetric groups $\mathbb{S}_n$, is defined as a finite group under function composition ∘ from the n! permutations of a set of degree n, S = {1, 2, …, n}, such that $|\mathbb{S}_n| = n!$. $\sigma_2\sigma_1$ is the permutation that we get by permuting {1, 2, . . . , n} first according to σ_1 and then according to σ_2. It is considered as group G acting on set S, as a subgroup of $\mathbb{S}_n$ denoted $\mathbb{S}_{|S|}$. Symmetric groups are very important in machine learning and computational algebra because, according to the Cayley theorem, every finite group is a subgroup of a Symmetric group. **The two-line notation** of a permutation places the input on the top row and the output mapping in the bottom row. For example, a permutation σ is defined for $\mathbb{S}_4$ as:

$\sigma = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 4 & 2 \end{bmatrix}$, such that $\sigma(1) = 2, \sigma(2) = 3$, and similarly, all remaining element mappings. This enables handling permutations as functions and enables function composition. The relations $\sigma_2\big(\sigma_1(i)\big) = (\sigma_2\sigma_1)(i)$ for all $\sigma_1, \sigma_2 \in \mathbb{S}_n$ and i = 1, 2, . . . , n, define group operations. For example, if $\sigma_1 = \sigma$, and $\sigma_2 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}$ m then $\sigma_2\sigma_1 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{bmatrix}$ by tracing the mapping from the top row to the bottom row in $\sigma_2$first, then

continue from the top row to the bottom row in σ_1. $\sigma_2\sigma_1 \neq \sigma_1\sigma_2$ because $\mathbb{S}_4$is not Abelian. Only $\mathbb{S}_1$and $\mathbb{S}_2$ are Abelian.

Cycle notation allows writing a permutation in one line as an ordered subset $s_1, s_2, \ldots, s_k$ of {1, 2, … , n}, such that $\sigma(s_i) = s_{i+1}$ for i < k and $\sigma(s_k) = s_1$. For example, For example, a permutation σ for $\mathbb{S}_4$ defined above in the cycle notation is (1)(2, 3,4), such that the permutation σ is a product of 2 cycles, the first is one mapping to itself in a 1-cycle, and the second is $2 \to 3, 3 \to 4$, and $4 \to 2$ in a 3-cycle. 1-cycles can be omitted as they do not change anything. A 2-cycle is a transposition. The order of elements in a cycle is not important as they rotate to reach each other, and the order of the cycles does not matter. Therefore, it is advised to start with the smallest number. If a permutation composition has repeated elements in each permutation mapping cycle, then the order of the cycles will matter. For example $\sigma_2{}^{\circ}\sigma_1 = (1,4)(2,3)^{\circ}(2,3,4) \neq \sigma_1{}^{\circ}\sigma_2 = (2,3,4)^{\circ}(1,4)(2,3)$.

The **cycle-type** of σ is a list of the lengths of all the cycles making up σ, encoded as a partition on integer n, denoted $\lambda \vdash n$, such as $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k) such\ that\ \lambda_i \geq \lambda_{i+1}$ for i=1, …, k-1 such that $n = \sum_{i=1}^{k} \lambda_i$, and k is the partition λ length. The partition of the previous example σ is $\lambda = (1,3)$, k=2, and $\lambda \vdash n$ is $(1,3) \vdash 4$.

A cyclic group G with multiplication operation is defined as generated by one element x in a cyclic way. For example, for $x \in G$, the smallest subset H containing x should contain x, its inverse, the identity element 1, and all powers of x and its inverses: {…, $x^{-4}, x^{-3}, x^{-2}, x^{-1}, 1, x, x^2, x^3, x^4, \ldots$ }. This group is the group generated by

$xH = \langle x \rangle$, and when $G = H = \langle x \rangle$, then G is a cyclic group. Another example for the groups with addition operation is the group of the integers $\mathbb{Z}, +$, the smallest subgroup H containing x, is {…, $-4x, -3x, -2x, -x, 1, x, 2x, 3x, 4x, \ldots$ }, and H is cyclic generated by $xH = \langle x \rangle$. We can have finite cyclic groups, such as Integers mod n under addition: $\frac{\mathbb{Z}}{n\mathbb{Z}}, +$. Cyclic groups are fundamental to the theorem of finitely generating Abelian groups, which states that any finitely generated Abelian group can be divided by a finite number of cyclic subgroups.

A symmetric (r, 0) tensor is an (r, 0) tensor whose value is unaffected (invariant) by the interchange (or transposition) of any two of its arguments, $T\ (v_1, \ldots, v_i, \ldots, v_j\ , \ldots, v_r) = T\ (v_1, \ldots, v_j\ , \ldots, v_i, \ldots, v_r)$ for any i and j. For rank two tensors, the symmetry condition implies $T_{ij} = T_{ji}$. Symmetric (0,r) tensors are defined similarly. You can easily check that the symmetric (r, 0) and (0,r) tensors each form vector spaces, denoted $S^r(V^*)$ and $S^r(V)$ respectively. This also means that these matrices are invariant under any rearrangement of the indices (since any rearrangement can be obtained via successive transpositions)

An antisymmetric (or alternating) (r, 0) tensor is one whose value changes sign under transposition of any two of its arguments, i.e. $T\left(v_1, \ldots, v_i, \ldots, v_j\ , \ldots, v_r\right) = -T\left(v_1, \ldots, v_j\ , \ldots, v_i, \ldots, v_r\right)$. Antisymmetric (0,r) tensors are defined similarly, and both

sets form vector spaces, denoted $\bigwedge^r V^*$ and $\bigwedge^r V$ (for r = 1 we define $\bigwedge^1 V^* = V^*$ and $\bigwedge^1 V = V$). Antisymmetry has the following properties:

1. $T(v_1, \ldots, v_r) = 0$ if $v_i = v_j$ for any $i \neq j$

2. $T(v_1, \ldots, v_r) = 0$ if $\{v_1, \ldots, v_r\}$ is linearly dependent

3. If dim (V) = n, then the only tensor in $\bigwedge^r V^*$ and $\bigwedge^r V$ for r>n is the 0 tensor

The symmetrization postulate states that all known particles in nature are either of type fermions such that their motion states are captured in a symmetric tensor or type bosons, such that their states are captured in an antisymmetric tensor. This postulate has many consequences that exclude many states from the high dimensional permutations that could be considered otherwise. It is not clear how valid this postulate is for all datasets in all application domains for machine learning, but it is definitely applied in many applications related to Physical motion and is worth checking.

A Lie group is at the intersection of group theory, differential geometry and linear algebra, named after the Norwegian mathematician Sophus Lie, who studied them in the late 1800s. Lie groups are continuous groups with group elements that are 'infinitely close' to the identity, known as 'infinitesimal transformations' or 'generators' or Lie Algebra. The simplest definition is that Lie groups can be parametrized in terms of a certain number of real variables, which define the dimension of the group, such as the surface of a sphere but should also be studied from the algebra based on an identity element. A lie infinite group G is a topological group with operations multiplication and inversion being smooth maps that is also a smooth differentiable manifold with the same topological maps $f(x) = x^{-1}: G \rightarrow G$ and $g(x, y) = xy: G \times G \rightarrow G$. An example of a Lie group is a group of invertible nxn matrices M of real numbers that is defined as follows: $M = \left\{ A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \right\}$, where det(A) ≠0, i.e. the lie group contains the entries of the matrix, forming a vector of n^2 dimensions: $M \rightarrow (a_{11}, a_{12}, \ldots, a_{1n}, a_{21}, \ldots, a_{2n}, \ldots a_{nn}) \in \mathbb{R}^{n^2}$. Taking the space $\mathbb{R}^{n^2}$ and removing all matrices B with determinants equal to zero, we have a manifold. **A lie type group** is similar to a lie group, but it is a finite group that is defined over a finite field. Many groups belong to this category; that is a simple group.

Lie Algebra is the smallest set of properties necessary for a vector space spanned by infinitesimal generators (defined below) to generate a Lie group maintaining its structure. The $\mathbb{R}^n$ Euclidean space is an example of a non-compact Lie group under the addition of vector, and zero vector as the identity with the differential and topological structures of $\mathbb{R}^n$. Lie algebra g is a vector space that is closed under commutators, in the sense that if X and Y ∈ g, then so is [X,Y] ≡ XY – YX. Expanding the commutators lead to satisfying the Jacobi Identity, such that [[X,Y],Z] + [[Y, Z], X] +[[Z, X],Y] = 0, ∀X,Y,Z ∈ g. A compact Lie group

example is the circle group $\mathbb{T}$ of complex numbers of unit modulus, forming a group under multiplication.

The Lie group connects components on a manifold in terms of **infinitesimal generators**, which means the smallest value of change ϵ that can be applied multiple times on the coordinates unit basis to generate all components in the manifold. For example, in a one-dimensional Manifold M for a group with a multiplication operation, the identity element is 1, and the tangent vector $\sigma = +1$ is taken as a basis for M. An infinitesimal transformation in the neighbourhood of the identity element is g =1+ $\epsilon\sigma$. if $\epsilon = 0.001$, and we want to represent point 2, then g =1+0.001=0.001, and $g^{693} = 1.999013$ is the closest approximation that generates 2. Using calculus, an exact generator can be used to avoid approximation. Let g_θ be a member of the group labelled by a real number θ, then g_θ is expressed in terms of the exponential function $\exp(\theta, \sigma)$. These exponential maps applied on infinitesimal generators reveal much information about the group's structure. Generalising this to a higher dimension, given a Lie Group (differential Manifold) with basis $T_1, T_2, \ldots, T_n$ of Tangent space T_xM, where T_i is the infinitesimal generators, any element x of G in the neighbourhood of the identity can be written in the form: $x(x_1, \ldots, x_n) = \exp(\sum_{i=1}^{n} x_i T_i)$for some $x_1, \ldots, x_n \in \mathbb{R}$.

Lie groups are very close to being entirely determined by the behaviour of their one-parameter subgroups; in particular, every element of a Lie group sufficiently close to the identity is contained in a one-parameter subgroup. This association between finite transformations and their infinitesimal versions is precisely the relationship between a Lie group and its Lie algebra. For example, rotation operators transform the orthogonal matrices into antisymmetric matrices.

The Lie bracket abstracts the Lie algebra from the matrix Lie groups to all other Lie groups. A Lie Group, which is a Manifold with tangent space at point x (T_xM), has a Vector space V of directions (infinitesimal generators) along which this point translates along the manifold. The Lie bracket g operation is the vector space V with bilinear map $[.,.]: V \times V \to V$ turns the Tangent space T_xM into a Lie algebra $\mathcal{L}(V)$, which is the set of all linear operators on V. The exponential mapping of infinitesimal generators tells us that at least locally, $\mathcal{L}$ completely determines the structure of G. This is defined by the bracket [X,Y] for $X, Y \in T_x\mathrm{M}$ generators for the group, which is calculated as [X,Y](f) = X(Y(f)) − Y(X(f)). This means a generator X acting on another generator Y that would give zero first-order term ((I+X)(I+Y) = 1+X+Y), the second-order infinitesimal term is then (XY – YX) independent of parametrization. The Lie bracket g satisfies the Antisymmetry property [X,Y]=- [Y, X] ∀X,Y ∈ g and the Jacobi identity explained earlier. The Lie bracket encodes the entire Taylor series expansion of the multiplication on the Lie group near the identity, capturing the "local" behaviour of the Lie group.

A differentiable Lie group is a differentiable Manifold M, such that a chart at point $x \in M$ is a homeomorphism φ from an open neighbourhood U of p to an open subset of $\mathbb{R}^n$. Every point of a differentiable manifold is covered by at least one chart. The i^{th} component of φ is

a function $[\varphi(x)]_i: U \to \mathbb{R}$, simplified to φ_i. We say that a function $f: M \to R$ is differentiable at point x if $f \circ \varphi^{-1}$ is differentiable at $\varphi(x)$, and use the notation $\frac{\delta}{\delta x^1}$for $\frac{\delta}{\delta x^1}(f \circ \varphi^{-1})$.

5.1.3.1 Classical groups:

These definitions laid down the foundation for the classical groups identified by physicists in an attempt to explain how the universe works. Then further analysed by mathematicians over more than a century in a wealth of literature that can not be explained clearly in a fraction of a chapter. I will focus on the groups used in computational graphics applications summarising some of their properties that are useful in understanding the stack of code in which they are used.

When Lie groups arise as groups of matrices (such Lie groups are called **linear groups**), the generators $T_1, T_2, \ldots, T_n$ are themselves matrices, the exponentiation is just matrix exponentiation, and [A,B] is equal to the commutator AB − BA. In chapter one, we have seen how to use **matrices** to solve a system of equations. We can make groups from matrices in two different ways for addition or multiplication operations. The matrix of real numbers $\mathbb{R}$ under addition has the zero matrices as the identity matrix and forms infinite groups that are Abelian and non-Abelian. The matrices under multiplication require matrices that can be multiplied together, so restricting them to nxn matrices is required, using the nxn identity matrix, and the inverses require the matrices to have non-zero determinants, making them a General Linear Group $GL_n(\mathbb{R})$. When the determinant is equal to one, it is the Special Linear Group $SL_n(\mathbb{R})$.

The general linear group of vector space V denoted GL(V), is a Lie group that is formed by the composition of maps of invertible linear transformations on V. When V is finite n-dimensional space, GL(V) is a group of $n \times n$ invertible (nonsingular) matrices of linearly independent rows and linearly independent columns, under matrix multiplication operation. Sometimes the notation used is GL(N, $\mathbb{R}$), or $GL_N(\mathbb{R})$, which means N-dimensional general linear group of real numbers $\mathbb{R}$. For example, a group containing one element as a 2x2 matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, with $a, b, c, d \in \mathbb{R}$ having determinant $\neq 0$, is a $GL_2(\mathbb{R})$ or GL(2, $\mathbb{R}$), the identity of this group is the 2x2 identity matrix. The subset of these matrices with determinant +1 forms a (sub)group called SL (n). The collection of n× n unitary matrices U(n) also forms a group under matrix multiplication.

When groups are unitary, they form U(N+, N-, $\mathbb{R}$) with parameters as positive dimensions and negative dimensions, along with the Field definition. When linear groups are orthogonal, they form O(N+, N-; $\mathbb{R}$) with the same definitions of the parameters. Orthogonality means that the matrix representing the group has the property $A^{-1} = A^T$, $or\ A^T A = I$. Both U and S groups are denoted for simplicity U(N) and O(N). The Symplectic groups S_p(2N, $\mathbb{R}$) have even dimensions 2N, on $\mathbb{R}$, and $\mathbb{C}$, but not when defined

on Quaternions $S_p(N,Q)$. There are also the unitary symplectic groups $US_p(2N+,2N-;\mathbb{R})$. U and O have special groups, such as the SU(n), a special unitary group of nxn unitary matrices with determinant =1. The Special Orthogonal group denoted as SO generalises rotation transformation matrices invariant of dimension and geometry. They are known as rotation matrices $R(\theta) \in SO(N)$ that are cross-product operators (CPO) capturing similarities with interesting properties. SU(2) is crucial in the theory of angular momentum in quantum mechanics, and SU(3) is fundamental in particle physics.

In chapter one, the rotation 2x2 matrix $R(\theta) \in SO(2)$ was derived manually from trigonometry rules, which is a one-parameter lie abelian and compact group. As a linear Lie group, it is generated from a single parameter, the angle θ, and using the Euler equation; it can be expressed as a complex number as $R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = I_2 \cos\theta + i\sigma_2 \sin\theta = e^{(i\sigma_2\theta)}$, where I_2 is the 2x2 identity matrix. σ_2 is the generator for θ and is obtained by differentiating R(θ), $\sigma_2 = -i\frac{dR}{d\theta}\Big|_{\theta=0} = -i\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$. Figure 3(a) illustrates a point rotation on the complex plane.

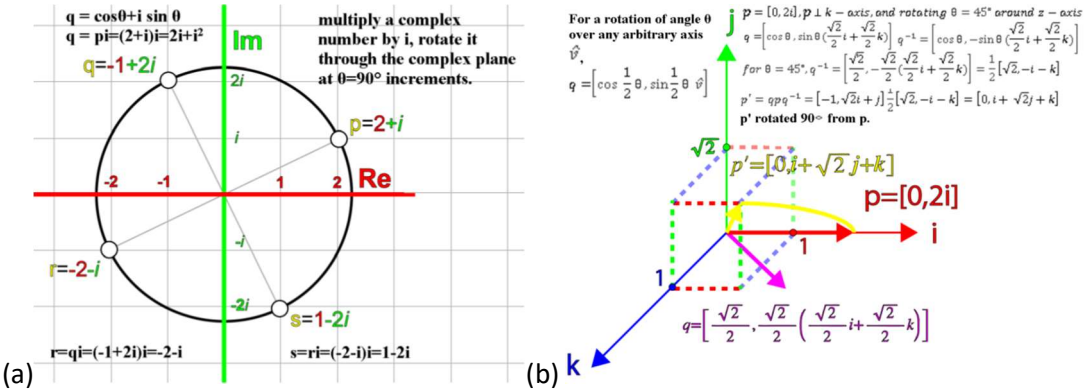


Figure 3: (a) 4 points on a sphere on a complex plane. Starting from point p, each point is generated by multiplying by I, making I a rotation operator for 90-degree rotations. (b) point p rotates 90 degrees around k-axis using quaternion q (Jeremiah, 2012).

Expanding to 3D, we define rotation around each axis independently. A rotation around the z-axis, $R_z(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$, which is generated from $T_z = -i\frac{dR_z(\theta)}{d\theta}\Big|_{\theta=0} = \begin{bmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. A rotation of angle θ can be generated by successive N applications of smaller angle rotations, $\frac{\theta}{N} = \delta\theta$. The infinitesimal angle $\delta\theta$ may be expanded from $R_z(\delta\theta) = I_3 + i\delta\theta T_z$, to N rotations as $R_z(\theta) = [I_3 + i\delta\theta T_z]^N = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + i\delta\theta \begin{bmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & \delta\theta & 0 \\ -\delta\theta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. As N goes to ∞, $R_z(\theta) = \lim_{N\to\infty}\left[1 + i\frac{\theta}{N}T_z\right]^N = e^{(i\theta\ z)}$, which identified T_z

as generator for R_z. This is solved by the series expansion $e^x = \lim_{n\to\infty}(1+\frac{x}{n})^n = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$. Similarly, for the other axes, by differentiating the coordinate rotations: $R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix}$, and $R_y(\varphi) = \begin{bmatrix} \cos\varphi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\varphi \end{bmatrix}$, we get the generators: $T_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{bmatrix}, T_y = \begin{bmatrix} 0 & 0 & i \\ 0 & 0 & 0 \\ -i & 0 & 0 \end{bmatrix}$. The Lie algebra for these rotation groups has a basis $\{I, T_x, T_y, T_z\}$, and is determined by the single relation $[T_a, T_b] = \epsilon_{abc} T_c$, where ϵ_{abc} is +1 if (a, b, c) is a cyclic shift of (x, y, z); ϵ_{abc} = −1 if it is a cyclic shift of the reverse permutation (z, y, x) and ϵ_{abc} = 0 in all other cases. Then SO(3) has commutators of the basis elements as $[T_x, T_y] = T_z$, $[T_y, T_z] = T_x$, $[T_z, T_x] = T_y$. When the generators are labelled with numbers instead of letters, we can extend to the higher dimensions easy as $[T_i, T_j] = \sum_{k=1}^{3} \epsilon_{i,j,k} T_k$. These are the angular momentum commutation relations of quantum mechanics.

The above θ, ψ, φ are axes' angle of rotations used as parameters to generate the rotation matrices. There is always a single-axis rotation that simplifies any combinations of these three rotations using interpolation. Other parameters to generate these matrices exist, such as the Euler angles and the quaternions. Euler rotations apply three consecutive rotations on each axis using Euler angles. Euler sequential rotations cause problems when two axes are lined up and lose a degree of freedom, causing the “gimbal lock” problem. Quaternions solve this problem, performs rotations in any order, and make interpolation easier. A quaternion has four components representing 4 points on the surface of the sphere, representing the rotation in 3D. The first is usually denoted w, as the unrotated point, x represents 180-degree rotation around the x-axis, similarily, y and z components represent 180-degree rotations around y- and z-axis, respectively. A quaternion 3D orientation is expressed as as qpq^{-1} of some 3D point p = xi+yj+xk, and q is the orientation quaternion, for example as (0.5+0.5i+0.5j-0.5k), and its inverse q^{-1}= (0.5-0.5i-0.5j+0.5k) . This uses the quaternion multiplication properties that extend the complex numbers multiplications. Figure 3(b) illustrates a point p 3D rotation using quaternion q. Because it is difficult to choose these points, most animation and graphics software do not expose their values. However, the blender software exposes the quaternion values. There are many visualisation videos to understand quaternions, such as https://eater.net/quaternions.

This example set of groups (rotation and translation) shows how a lie group can be generated from infinitesimal generators, such as the angular momentum operators/vectors generating rotations and the momentum operator generating translations. They also show how two or more successive transformations can be combined using a different transformation and can be reversed using an inverse transformation.

SO(3) is generalised to $\mathbb{R}^n$ as SO(n). SO(n) is a manifold with various charts that allows a change of basis or coordinate systems. This is generally used in computer graphics and

planning the motion of robots or objects in 3D scenes in a series of coordinate system changes between the different coordinate frames of the camera, the robot and the world. (Olguín Díaz, 2018) explains more details about how these matrices are constructed from the degrees of freedom (represented as infinitesimal generators as explained earlier). For example, three parameters are required in $\mathbb{R}^3$ to construct the nine elements of a rotation matrix using exponential mapping of the cross product of the parameters vector. The symmetry of all possible rotations around the intrinsic axis (the current coordinate frame of the robot or object) or the extrinsic axis (the base frame) creates 24 different possibilities. The Euclidean group, denoted E(n) or ISO(n), is a group of Euclidean isometries such that the transformations (all translations, rotations, and reflections) preserve the Euclidean distance between any two points. The Special Euclidean Group denoted $E^+(3)$, $ISO^+(3)$, or SE(3) of order 3 is a mapping transformation that combines rotation R, and a translation d (no reflection) from the parameters of vectors (R, d) on each axis, $SE(3) = \mathbb{R}^3 \times SO(3)$. Their elements represent rigid motion. The Skew Symmetric Group SS(3) of order 3 is a mapping transformation that combines the rotation matrix and the angular velocity.

The projective space P^n, such as those used in image representations, has n+2 basis, the standard basis $e_i = [0, \ldots, 1, \ldots, 0]^T$, where only the i^{th} position element is equal to 1, for $1 \leq i \leq n+1$, and the standard projective basis $e_{n+2}=[1, 1, \ldots, 1]^T$, where all elements are equal to 1. These projective basis are called homogeneous coordinates. For 3D, we need four coordinates represented using the quaternions. P^n is topologically equivalent to the unit sphere S^n of $\mathbb{R}^{n+1}$, in which the antipodal points have been identified. Since any point in S^n is represented by the vector $x=[x_1, \ldots, x_{n+1}]^T$ such that $x = \sum_{i=1}^{n+1} x_i^2 = 1$ is also a point in P^n, with vector -x as the antipodal point of x found in both S^n and P^n. This means all projective spaces are compact spaces, and their differential structures are as simple as the Euclidean ones, such as embedding the sphere units in the higher space, as illustrated in Figure 4. A homography is an isomorphism of projective spaces to enable generalising an image representation invariant of the camera position by projecting a 2D image onto a 3D space, as explained by this OpenCV tutorial on various applications https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html, and applied in data augmentation to achieve more invariance for DL data representation to reduce overfitting on specific image details such as https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator.

The action of the three-dimensional rotation group SO(3) on the unit sphere S_2 is an example of homogeneous space. Taking any point on S_2, the unit vector e_z pointing along the z-axis, and a rotation matrix R, $\{Re_z | R \in SO(3)\}$ sweeps out the entire sphere, so S^2 is a homogeneous space of SO(3). The isotropy group, in this case, is the subgroup of rotations about the z-axis, which is just SO(2). For example, composing two rotations in sequence, the first R_1 is a $\theta_1$rotation from coordinate frame A to B, ${}^A\{R_1\}_B\,(\theta_1)$ and the second R_2 is a

θ_2 rotation from coordinate frame B to C ${}^{B}\{R_1\}_C\ (\theta_2)$, using homogenous coordinate is ${}^{A}[\![\{R\}]\!]_1\}_B\ (\theta_1) \times {}^{B}\{[\![\{R_1\}]\!]_C\ (\theta_2)$:

$$= \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos\theta_2 & \sin\theta_2 & 0 & 0 \\ -\sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\ (\theta_1+\theta_2) & \sin(\theta_1+\theta_2) & 0 & 0 \\ -\sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

All matrices in this subgroup are the same periodic function of one real variable, θ, given by R(θ) $\in SO(2) = R_1(\theta_1) + R_2(\theta_2) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$., i.e. $SO(2)$ is locally isomorphic to $\mathbb{R}^1$. Another $SO(3)$ example combining a rotation with a translation is $\begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$; the set of such matrices is continuously parameterized by these two variables θ and d. Thus, this subgroup is a two-dimensional Lie group, resulting from the Cartesian product $SO(2)\ \times\ T(1)$. These groups are differential; their first-order derivative gives an expression of the velocity of a motion, and the higher-order derivative gives an expression of acceleration.

From an algebraic point of view $S^2 \sim= \frac{SO(3)}{SO(2)}$, and generalises to $S^{n-1} \sim= \frac{SO(n)}{SO(n-1)}$. There is also a surjective relationship from SU(2) to SO(3). In other words, SO(3) has SU(2) as a compact connected covering group, and SU(2) is the double cover of SO(3). Since the higher n SO have a Lie algebra with a 3 × 3 compact regular representation and can be composed of coset decomposition with respect to a compact subgroup.

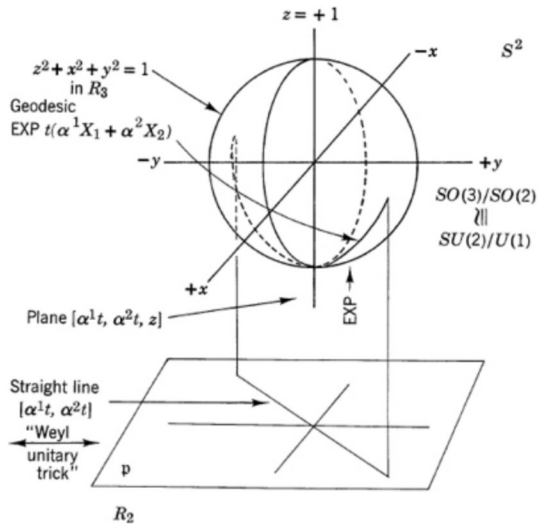


Figure 4: The coset space P=EXPP, originates from SO(3)/SO(2) ≈ SU(2)/U(1), appears as the surface of a sphere S^2 in $\mathbb{R}^3$ (Gilmore, 2005).

It is difficult to explain concisely how these relationships have been identified and how many of them are used in many computational applications. Generally, some groups are

composed of simpler groups, simpler groups approximate complex groups, and others can be generated from parameters. For example, there is a Lie algebra isomorphism stemming from the group homomorphism between lower n SU and higher n SO, when SU act on a given vector field such as $SU(2) \cong SO(3)$ and $SU(4) \cong SO(6)$ because of the double cover in the representation. A theory states that nonsemisimple Lie groups can be constructed from semisimple Lie groups by a limiting procedure, and group expansion performs the inverse process. Applications of Lie groups and their algebra leads to a complete listing of all the globally symmetric pseudo-Riemannian symmetric spaces, such as the sphere and the hyperboloid, are the Riemannian symmetric spaces associated with the group SO(3). If this very concise summary of this huge topic motivates you to learn how these are derived and the many properties and applications of expansions and contractions of these algebraic structures, please read full books on the subject such as (Gilmore, 2005) (Vince, 2021), (Altmann, 1986), (Jeevanjee, 2011) and (Dixon, 2002).

The SO and SE matrices are implemented in OpenCV Python, such as affine transformations using "getAffineTransform", "getRotationMatrix2D", and the perspective transformations using "getPerspectiveTransform". Peter Corke implemented them in Matlab and partially implemented them in the Python package Robopy (Corke, 2017). Also, Python package PyGeometry implements many of these matrices. RobotPy is also a Robot simulator in Python that its path planning module builds on top of these matrices.

5.1.3.2 Geometric groups:

The group of symmetries is defined as a group of groups formed from the symmetries of a geometric shape, such that the shape looks the same before and after flipping and rotation. For example, **the dihedral group** is the group of symmetries of a regular n-polygon shape. This group contains 2n symmetries groups, the identity transformation that does nothing e, the θ rotation symmetries such that $\theta = 360°$ for complete rotation, or $\theta = \frac{360^{\circ}}{n} = \frac{2\pi}{n}$ radians for r single symmetric rotation that move one point to the next point position. Repeated application of r gives n different symmetries to rotate back to the original position, which is the identity: $r^n = e$. The reflection/flip around an axis is another symmetric transformation. For the n-polygon, we have an axis passing through each point that can be used for symmetric reflections. If you flip twice, you return to the identity position: $f^2 = e$. Other symmetries are formed by composing rotations and reflections. This makes the finite dihedral group of symmetries contains 2n symmetries defined as $\{e, r, r^2, \dots, r^{n-1}, f, r.f, r^2.f, \dots, r^{n-1}.f\}$. This is non-Abelian such that $r.f \neq f.r$. Some denote this group as $D_n$for number of elements in the group, and others denote it as $D_{2n}$for the number of symmetries in the group. For example, the simplest n-polygon is the equilateral triangle where n=3, and the dihedral group of symmetries for it contains six symmetries = $\{e, r, r^2, f, r.f, r^2.f\}$ such that the element order of r is 3, $r^3 = e$, and the element order of f is 2, $f^2 = e$ as shown in Figure 5.

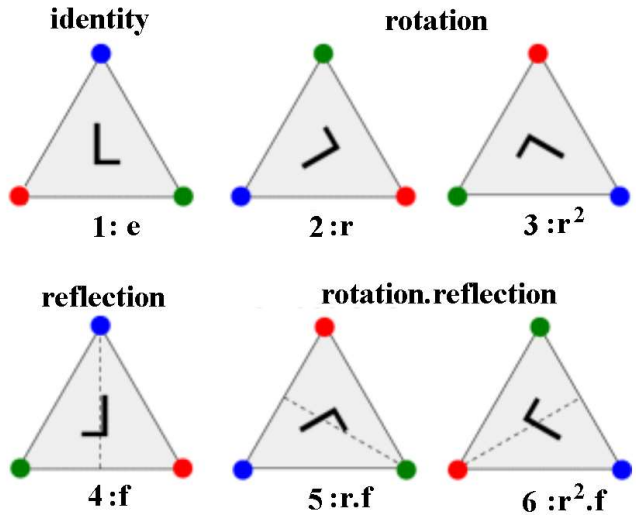


Figure 5: The dihedral group of symmetries for an equilateral triangle.

This goes on for polygons of higher degrees and other shapes. To summarise the group types discussed above, we have:

	Finite Groups	**Infinite Groups**
Abelian Groups	Integers mod n: $\mathbb{Z}\backslash n\mathbb{Z}$	$\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ under +
Non-Abelian Groups	Symmetric Groups: $\mathbb{S}_n$ for n > 2	Matrices: GL, SO, SU, …

5.1.4 Group Decomposition

Group decomposition into simpler ones is achieved using various methods. This section will explain some of these methods. Cosets are not subgroups, but they provide a traversal method of group elements, and their elements can be treated as subgroups. The following will explain the difference between cosets and subgroups, how they are used in group decompositions, and how similar methods can be applied to rings and modules.

Left and right cosets of $H < G$ and $x \in G$ are defined as $xH = \{xh | h \in H\}$ as the left coset of x and the set $Hx = \{hx | h \in H\}$ as the right coset of x, both cosets have the same cardinality as H. cosets are not groups and do not have the axioms stated earlier. Any two cosets are either identical or disjoint, and the set of left cosets provides a partition of G, i.e. G of cardinality $|G| = n$ is split into k non-overlapping (disjoint) left cosets of the same cardinality $|H| = d$: $H,\ x_1H,\ x_2H, \dots,\ x_nH$, such that $d.k = n$, i.e. d divides n: $d|n$. The same can be done using the right cosets. Abelian groups have left cosets the same as the right cosets. If H is a **normal subgroup**, then $xH = Hx$ for any $x \in G$, hence the systems of left and right cosets are the same. This leads to $\frac{G}{H}$ forming a **quotient group** under the

operation $(xH)\ (yH)\ = (xy)H$, which is not a subgroup of G. Example quotient group $\frac{\mathbb{Z}}{5\mathbb{Z}}$ Is explained below.

A quotient space is defined for the left and right cosets. G/H is the quotient space of left cosets. Similarly, H\G is the quotient space of the right cosets. If H is a normal subgroup of G, then the group of cosets $\frac{G}{H}$ forms a group that is called **factor group/quotient group** and denoted $H \trianglelefteq G$.

Module Arithmetic, founded by Gauss, takes a group of integers, partitions them into a finite number of sets, and handles each set as a new type of number. This is done using **normal subgroups and Quotient groups** in group theory. For example, the Integer mod 5 group is the sets of r= Integer mod 5, as follows:

r=0	{… , -10, -5, 0, 5, 10, …}
r=1	{… , -9, -4, 1, 6, 11, …}
r=2	{… , -8, -3, 2, 7, 12, …}
r=3	{… , -7, -2, 3, 8, 13, …}
r=4	{… , -6, -1, 4, 9, 14, …}

Applying the group operation on an element from one specific subgroup to an element in another specific subgroup will produce an element in the same third subgroup. These are called congruence classes. For example, in integer mod n group: $\mathbb{Z}/n\mathbb{Z}$, the group with elements a, b, a is congruent to b, if they give the same remainder when divided by n. Another example is the Integers $\mathbb{Z}$ group with addition operation has an infinite number of subgroups: $\mathbb{Z}, 2\mathbb{Z}, 3\mathbb{Z}, 4\mathbb{Z}, 5\mathbb{Z}$, … Given one subgroup $5\mathbb{Z}$, and all possible remainders to it, we represent $\mathbb{Z}$ group with one disjoint subgroup and four cosets as shown in the table below. Adding an element from the first coset to an element of the third coset yield an element in the fourth coset: $(1 + 5\mathbb{Z}) + (3 + 5\mathbb{Z}\ ) = (4 + 5\mathbb{Z}\ )$. These sets are congruence classes.

$5\mathbb{Z}$ subgroup	r=0	{… , -10, -5, 0, 5, 10, …}
$1 + 5\mathbb{Z}$ coset	r=1	{… , -9, -4, 1, 6, 11, …}
$2 + 5\mathbb{Z}$ coset	r=2	{… , -8, -3, 2, 7, 12, …}
$3 + 5\mathbb{Z}$ coset	r=3	{… , -7, -2, 3, 8, 13, …}
$4 + 5\mathbb{Z}$ coset	r=4	{… , -6, -1, 4, 9, 14, …}

A left transversal A is a set of coset representatives with two distinctive elements $x, y \in A$, $xH \neq yH$ and $\cup_{x\in A}\ xH = G$. **Right transversals** are defined analogously.

Other types of cosets are two-sided and double cosets. **Two-sided cosets** are sets of the form $x_1Hx_2 = \{x_1hx_2|h \in H\}$ (with $x_1, x_2 \in G$). **Double cosets** are defined for H_1 and H_2 are both subgroups of G such that $H_1xH_2 = \{h_1xh_2|\ h_1 \in H_1, h_2 \in H_2\}$ (with $x \in G$). This forms the **double quotient space** $\frac{H_1\backslash G}{H_2}$.

Algebraic structures are often constructed from the direct sums or direct products of two or more simpler structures. **The direct product** of two groups G, H, denoted $G \times H$, is the group of pairs $(g, h), g \in G, h \in H$ resulting from the cartesian product defined as $(g_1, h_1)(g_2, h_2) = (g_1g_2, h_1h_2)$. The identity of $G \times H$ is (e_G, e_H). The group operations are defined on the pairs as given $(a, b), (x, y) \in G \times H$, then $(a, b).(x, y) = (a.x, b.y)$. For example, given G is $\mathbb{Z}$ under addition and H is {1, -1, i, -i} under × , then $G \times H = \{(x, y)|x \in \mathbb{Z}, y = \pm 1\ or\ y = \pm i\}$ having identity element (0, 1). A group operation on the pairs (5, -i), (0, 1) = (5+0, -i.1)=(5, -i). This is because the first group operation is the addition, and the second group operation is the multiplication. The direct product can be for any number of groups, forming tuples containing elements from each group, such as $G_1 \times G_2 \times G_3 = \{(x, y, z)|x \in G_1, y \in G_2, z \in G_3\}$ with identity element $\left(e_{G_1}, e_{G_2}, e_{G_3}\right)$, and abstract group operation as defined for each group $(a, b, c) * (x, y, z) = \left(a *_{G_1} x, b *_{G_2} y, c *_{G_3} z\right)$. If any one of the k groups in the direct product is infinite, then the direct product is infinite. If all groups are finite with orders $n_1, n_2, \dots, n_k$, then the order of the direct product is $n_1 \times\ n_2 \times\ \dots \times\ n_k$. The order of the direct product is infinite for the product of an infinite number of finite groups. If any one of the groups is non-Abelian, then the direct product is non-Abelian. Another example is $\mathbb{R}^n$, as a group, is the direct product of n copies of $\mathbb{R}$. Similarly, the n-dimensional unit torus $\mathbb{T}^n$ is a direct product of n copies of the circle group. Not all groups are composed of the direct product of simpler groups. Other decomposition methods will be explained below.

A Normal subgroup N of group G and $x \in G$, partitions group G into several cosets. The elements of these cosets form a **factor group** in which the identity element is the normal subgroup, such that the multiplication of an element in one coset with an element in another coset will produce an element in a third coset: $x_1N.x_2N = \{x.y|x \in x_1N, y \in x_2N\} = x_jN$. A factor group example in which N is the identity element and the rest are the cosets partitioned from G is as follows:

$$\begin{pmatrix} x_6N & x_7N & x_8N \\ x_5N & x_4N & x_3N \\ N & x_1N & x_2N \end{pmatrix}$$

An ideal $I \subseteq R$ to a ring R is like a normal subgroup is to a group, with concepts of partitioning into cosets and kernels of homomorphism. Ideal partition rings into cosets; a collection of cosets are called factor groups or quotient groups. Ideal I must be an additive subgroup of ring R: (I, +)≤ (R, +), should also be closed under multiplication, and for any elements r in R, and i in I, then ir, ri should also be in I, such as cosets R/I form a factor/quotient ring.

A sub-module can be similarly defined to decompose modules, the way normal subgroups and ideals decompose groups and rings, respectively. For example, $3\mathbb{Z}$ is a $\mathbb{Z}$-module, with infinite sub-modules as $6\mathbb{Z}$, $9\mathbb{Z}$, $12\mathbb{Z}$, …, etc.

The Normal Series is defined as decomposing finite group G and choosing a proper, normal and maximal subgroup such that it is the biggest normal subgroup in G, denoting it N1, then the next maximal normal proper subgroup N2, and so on until the identity subgroup. This forms the Normal Series: $G \rhd N_1 \rhd N_2 \rhd \cdots \rhd 1$ such that $N_i < G$, and no duplicate subgroups. **The composition series** is a normal series that is as long as possible, similar to the prime factorisation of an Integer. A finite group G can have more than one composition series that are equivalent by having the same lengths and identical factor groups that might need to be rearranged to be isomorphic according to Jordan-Hölder Theorem. The composition series is used in solving equations of polynomial n such as $a_nx^n+a_{n-1}x^{n-1}+….+a_1x+a_0 = 0$ formed as a group S_n. The solution is to decompose S_n into its composition series $S_n \rhd N_1 \rhd N_2 \rhd \cdots \rhd 1$ with abelian factor groups of simple quotient groups $\left(\frac{N_1}{1}, \frac{N_2}{N_1}, \frac{N_3}{N_2}, \dots\right)$, but this fails when the polynomial degree n is ≥5. When n≥5, the group S_n is decomposed into three groups only, the full group itself, the alternating group and the trivial identity group: $S_n \rhd A_n \rhd 1$. This leads to having the factor groups be $\left(\frac{S_n}{A_n}, \frac{A_n}{1}\right)$, such that $\frac{S_n}{A_n} \cong \frac{\mathbb{Z}}{2\mathbb{Z}'}$ and $\frac{A_n}{1} \cong A_n$, and A_n was proven to be simple for n≥5 and can not be decomposed.

Semi-direct product of a group H and a subgroup H, denoted $G \rtimes H$, is the group of all ordered pairs $(x, \Lambda)(x \in G, \Lambda \in H)$ with group multiplication defined $(x', \Lambda')(x, \Lambda) = (x'\Lambda'(x), \Lambda'\Lambda)$. The unit element of $G \rtimes H$ is (e_G, e_H) and the inverse of (x, Λ) is $(\Lambda^{-1}(x^{-1}), \Lambda^{-1})$. For example, the isometry group $ISO^+(3)$ of $\mathbb{R}^3$ (excluding reflections), composed of transformations $x \to Rx + b$, where R is a rotation matrix, and b is a translation vector. This group and its generalisations to n dimensions are called the **rigid body motions groups**.

Group mappings are identified by different properties as follows:

- **A function on a group** G is a function $f: G \to S$ mapping each group element to a member of some set S, which is primarily a vector field $\mathbb{F}$. If G is a finite group and V is a vector space over $\mathbb{C}$ of dimension |G|, taking any basis of V and labelling the basis vectors with the group elements $\{e_x\}_{x\in G}$, we identify the linear mapping of G elements L(G) with V by mapping each $f \in L(G)$ to the vector $\sum_{x\in G} f(x)e_x$.
- A **G-module** of a group G is a vector space V over a field $\mathbb{F}$ with operation $G \times V \to V$ satisfying: $xv \in V$; $x(\alpha v + \beta w) = \alpha(xv) + \beta(xw)$; $(xy)v = x(yv)$; $ev = v$. A G-module admits a homomorphism $\varphi: G \to GL(V)$.
- **Translating a function on a group,** $f: G \to S$ by $t \in G$, will yield $f^t: G \to S$, given by $f^t(x) = f(t^{-1}x)$, is the left-translate. If G is non-Abelian, we must distinguish the left-translate from the right-translate $f^{(t)}(x) = f(xt^{-1})$.

- **Haar measures** μ is a left-translation and right-translation invariant measure on G by any t ∈ G. Its left-translate is $\mu^t(X) = \mu(\{ x \in G | tx \in X\})$, where X is the left-translate. It is unique up to scaling that is usually set so that $\int_G \ \mu(x) = 1$.
- **A closed set of transformations** turns a bijective mapping into a group by two operations, compositions, and inverse produce elements in the set. Given a set of bijective mappings $T \ = \ \{T_i : S \to S\}$ if for any $T_1, T_2 \ \in \ T$, their composition $T_2 \ \circ \ T_1$ is an element of T, and if for any $T_i \in \ T$, the inverse map T_i^{-1} is also $\in \ T$, and it reverses the transformation T_i.
- **A group G acts on a set S** if, to every group element x, we can associate a function $T_x : S \to S$ (also $T_x(s), x(s), or\ xs$) in such a way that $T_e(s) = s$ for all $s \in S$ and $T_{xy} \ = \ T_x \left(T_y(s)\right)$ for all $x, y \in S$ and s ∈ S.
- An **orbit** is what we get when we apply every element of G to s as a subset of S $\{x(s)\}_{s \in S, x \in G}$. The orbits partition S into disjoint subsets. If there is only one orbit, we say that G acts transitively on S.
- **A homogeneous space of group** G is a set S when there is only one orbit, and we fix any $s_0 \in S$, the map $\varphi : G \to S$ given by $x \to xs_0$ is surjective (i.e., sweeps out the entire set S).
- **The isotropy subgroup of group G** is formed from group elements fixing s_0 , which are subgroup $H = \{h \in G | hs_0 = s_0\}$. The Left quotient spaces and homogeneous spaces are just two aspects of the same concept since $xhs_0 \ = \ xs_0$ for any $h \in H$ is a one-to-one correspondence between S and the left cosets G/H.

The Groups, Algorithms, and Programming (GAP) is an interpreted language written in C that has Pascal-like syntax and can be compiled. The core GAP system contains built-in capabilities for group theory representation, algorithms, data sets, and many user-contributed packages. They have GAP 4 now, online documentation, and examples, and it can be downloaded from https://www.gap-system.org/Releases/index.html (GAP4, 2022). A Python wrapper to GAP can be found at https://github.com/embray/gappy/. Some elementary abstract algebra routines are implemented in Python, such as sack: https://github.com/johnkerl/sack. Ch5.ipynb uses some examples from sack to illustrate some of the concepts introduced in this section.

5.1.5 Representation Theory

Representation theory models the abstract groups using concrete matrices. A matrix representation ρ of a compact group G over Field $\mathbb{F}$ associated with each element of $x \in G$, a matrix $\rho(x) \in \ \mathbb{F}^{d_\rho \times d_\rho}$ such that $\rho(xy) = \rho(x)\rho(y)$, for all $x, y \in G$, $\rho(e) = I$, $\rho(x^{-1}) = (\rho(x))^{-1}$, d_ρ is the order of the representation. This section explains the representation types and their properties.

Trivial representation is a one-dimensional irreducible constant representation $\rho_{tr}(x) = I \; \forall x \in G$. From one-dimensional to |G|-dimensional are possible based on the complete reducibility theorem.

Equivalent representations are defined for ρ_1, ρ_2, when there is an invertible square matrix T such that $\rho_1(x) = T^{-1}\rho_2(x)T, \forall x \in G$.

The fundamental representation of a Lie group can be thought of as the generators of the group. For example, vector representation is used for G = O(3) or SO(3) acting on vector space V = $\mathbb{R}^3$, and the spinor representation are used for G = SU(2) acting on a vector space V = $\mathbb{C}^2$.

The adjoint representation is based on the Adjoint homomorphism as a map from G to GL(g), where the operator Ad_A for A ∈ G is defined as $Ad_A(X) = AXA^{-1}, X \in g$. This is the Lie algebra of G. The following table summarises this and the previous representation type for some known groups (Jeevanjee, 2011):

Group	Fundamental Representation	Adjoint Representation
SO(3)	vector	vector
O(3)	vector	pseudovector
SU(2)	spinor	vector

Regular representation is the |G|-dimensional representation based on the action of G on itself, such that the representation matrices' rows and columns are labelled by the group elements, the matrix entries of the regular representations are:

$$\left[\rho_{reg}(x)\right]_{xy} = \begin{cases} 1, \; if \; xy = z \\ 0, \; otherwise \end{cases}.$$

A **character** is assigned to each representation χ: $G \rightarrow C$, such that it is the trace of the representation matrices, $\chi(x) = tr\big(\rho(x)\big)$, such that equivalent representations will share the same character.

Almost all interesting representations result from tensor products of other, simpler representations. Tensors themselves are elements of tensor product spaces, enabling transformations and reducible representation. **Tensor representation** is defined as $(\rho_1 \otimes \rho_2)(g) \equiv \rho_1(g) \otimes \rho_2(g) \in \mathcal{L}(V_1 \otimes V_2)$. The tensor representation provides a huge analysis of equivalent representations of known groups that can be studied from (Jeevanjee, 2011).

Reducible representation ρ if each matrix $\rho(x)$ has the block structure: $\rho(x) = \begin{bmatrix} A(x) & B(x) \\ 0 & C(x) \end{bmatrix}$, then there is always a similarity transformation $\rho \rightarrow T^{-1}\rho T$ that reduces ρ

to a direct sum: $T^{-1}\rho(x)T = \begin{bmatrix}\rho_1(x) & 0\\ 0 & \rho_2(x)\end{bmatrix}, \forall x \in G$ of smaller representations, $\rho_1(x), \rho_2(x)$. This decomposition is unique, up to equivalence of representations and changing the order of the terms in the direct sum. The different ρ_i's are not necessarily distinct.

Identifying the inequivalent representations of a given group or Lie algebra is important. This is difficult but can be simplified using the fact that for each group or algebra, there exists a denumerable set of inequivalent representations (known as the "irreducible" representations), out of which all other representations can be built by a kind of summation.

Irreducible representation ρ is when there is no invertible square matrix T that can simultaneously block diagonalise all $\rho(x)$ matrices in the above way. According to Wedderburn's or Maschke's theorem of complete reducibility, the blocks ρ_i used in the decomposition above are called irreducible blocks.

A complete set of inequivalent irreducible representations of G, denoted as $\mathcal{R}_G$ or $\mathcal{R}$ finding is essential for decomposition. Any ρ ∈ $\mathcal{R}$ can be replaced by an equivalent representation $\rho^1(x) = T^{-1}\rho(x)T$. When, $\rho(x^{-1}) = (\rho(x))^{-1}\ \forall x \in G$, then these are **unitary** irreducible representations. Some essential properties are as follows:

- If G is finite, then $\mathcal{R}$ is a finite set.
- If G is compact but not finite, then $\mathcal{R}$ is a countable set.

An **irreducible character** is assigned an irreducible representation $\chi: G \rightarrow C$, such that it is the trace of the irreducible representation matrices, $\chi(x) = tr\big(\rho(x)\big)$.

A complete set of irreducible characters form an orthogonal basis for the space of class functions, and with conjugacy classes, both can unambiguously label the irreducible representations of a finite group, such that $\tilde{\rho}(x) = \rho(x),\ if\ x \in H$; otherwise, it is equal to zero.

The dual space $\hat{G}$ of group G is the space of characters. The Abelian groups corresponding characters form a group under the operation $(\chi_1\chi_2)(x) = \chi_1(x)\chi_2(x)$. Pontryagin theorem states that the double dual $\hat{\hat{G}}$ is isomorphic to G, and the isomorphism is canonical in the sense that there is a unique $\xi \in \hat{\hat{G}}$ satisfying $\xi(\chi) = \chi(x)$ for any $\chi \in \hat{G}$. **Harmonic analysis** is founded on this theorem on Locally Compact Abelian (LCA) groups.

G-Module V used in $\varphi: \mathrm{G} \rightarrow \mathrm{GL(V)}$, which is equivalent to $\rho: \mathrm{G} \rightarrow \mathrm{GL(V)}$, identifies an **equivalence class** of equivalent representations ρ that are equivalent to mapping φ. V is defined with the basis $e_1, e_2, \ldots, e_{d_\rho}$, from which the matrix entries of the representation matrix ρ can be recovered as $[\rho(x)]_{i,j} = \big[\rho(x)e_j\big]_i$.

The permutation representation $\mathbb{C}S$ **naturally constructs group representations** of group G associated with set S, forming a G-module with basis vectors labelled by elements of S on which G acts by $g(e_s) = e_g(s)$.

The restricted representation $\rho\downarrow_H^G, \rho\downarrow_H$ is defined as given group G, its representation ρ, and a subgroup H of G, then $\rho\downarrow_H^G = \rho(x), \forall x \in H$, which is just a subset of the matrices of the representation of G.

The induced representation $\rho\uparrow_H^G, \rho\uparrow^G$ is defined as given a subgroup H, $t_1, t_2, \ldots, t_l$ be a transversal for the left cosets of H in G; we can induce a representation ρ of G, by the block diagonal matrix:

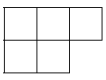
$\rho\downarrow_H^G = \begin{bmatrix} \tilde{\rho}(t_1^{-1}xt_1) & \cdots & \tilde{\rho}(t_1^{-1}xt_l) \\ \vdots & \ddots & \vdots \\ \tilde{\rho}(t_l^{-1}xt_1) & \cdots & \tilde{\rho}(t_l^{-1}xt_l) \end{bmatrix}$, where $\tilde{\rho}(x) = \rho(x)$ if $x \in H$ and otherwise, it is zero.

For Symmetric group $\mathbb{S}_n$ **the conjugacy classes** correspond exactly to the collection of elements of a given cycle type as defined earlier. Hence, the irreducible representations of $\mathbb{S}_n$ can be labelled by the integer partitions $\lambda \vdash n$.

The conjugacy class of transpositions is defined as the permutations of the form (i, j). A cycle $(s_1, s_2, \ldots, s_k)$ can be written as the product $(s_1, s_2) \cdot (s_2, s_3) \cdot \ldots \cdot (s_{k-1}, s_k)$, the set of transpositions generates the entire symmetric group.

The sign of σ, denoted $sgn(\sigma) \equiv (-1)^m$, defines the **even permutations** for which sgn(σ) = 1, and the **odd permutations** for which sgn(σ) = −1. This creates an **alternating representation** of $\mathbb{S}_n$ of degree n, denoted A_n. Alternating groups and symmetric groups form a family of non-Abelian finite groups. **The trivial representation** ρ(σ) = 1 and the alternating representation ρ(σ) = sgn(σ) are both irreducible representations and the only two one-dimensional ones.

The defining representation describes $\mathbb{S}_n$ in terms of its action on (1, 2, …, n) with the n-dimensional representation $\left[\rho_{def}(\sigma)\right]_{i,j} = \begin{cases} 1, & if\ \ \sigma(\mathrm{i}) = \mathrm{j} \\ 0, & otherwise \end{cases}$, such that the $\rho_{def}(\sigma)$ matrices are often called permutation matrices. The defining representation is reducible by the direct sum of trivial representations and some n-1 dimensional representation.

A Ferres diagram is a graphical representation for integer partitions consisting of simply laying down $\lambda_1, \lambda_2, \ldots, \lambda_k$ empty boxes in k consecutive rows, such as for the shape of the (3, 2) partition of 5: 

A Young tableau is a Ferres diagram bijectively populated by the numbers 1, 2, . . . , n, such as: $t = \begin{array}{|c|c|c|}\hline 3 & 2 & 5 \\ \hline 1 & 4 \\ \cline{1-2}\end{array}$

A tabloid is an equivalence class of tableau under permutations of the numerals in each of the rows. For example, the set of tableaux of shape (3, 2), denoted as:

$$\{t\} = \begin{array}{|c|c|c|}\hline 3 & 2 & 5 \\ \hline 1 & 4 \\ \cline{1-2}\end{array} =$$

$$\left\{ \begin{array}{|c|c|c|}\hline 3 & 2 & 5 \\ \hline 1 & 4 \\ \cline{1-2}\end{array}, \begin{array}{|c|c|c|}\hline 3 & 5 & 2 \\ \hline 1 & 4 \\ \cline{1-2}\end{array}, \begin{array}{|c|c|c|}\hline 2 & 3 & 5 \\ \hline 1 & 4 \\ \cline{1-2}\end{array}, \begin{array}{|c|c|c|}\hline 2 & 5 & 3 \\ \hline 1 & 4 \\ \cline{1-2}\end{array}, \begin{array}{|c|c|c|}\hline 5 & 2 & 3 \\ \hline 1 & 4 \\ \cline{1-2}\end{array}, \begin{array}{|c|c|c|}\hline 5 & 3 & 2 \\ \hline 1 & 4 \\ \cline{1-2}\end{array}, \right.$$

$$\left. \begin{array}{|c|c|c|}\hline 3 & 2 & 5 \\ \hline 4 & 1 \\ \cline{1-2}\end{array}, \begin{array}{|c|c|c|}\hline 3 & 5 & 2 \\ \hline 4 & 1 \\ \cline{1-2}\end{array}, \begin{array}{|c|c|c|}\hline 2 & 3 & 5 \\ \hline 4 & 1 \\ \cline{1-2}\end{array}, \begin{array}{|c|c|c|}\hline 2 & 5 & 3 \\ \hline 4 & 1 \\ \cline{1-2}\end{array}, \begin{array}{|c|c|c|}\hline 5 & 2 & 3 \\ \hline 4 & 1 \\ \cline{1-2}\end{array}, \begin{array}{|c|c|c|}\hline 5 & 3 & 2 \\ \hline 4 & 1 \\ \cline{1-2}\end{array} \right\}$$

The permutation representation is a real-valued representation of shape λ that is defined such that if t and t' are two members of some tabloid {t}, then their images σ(t) and σ(t') for some $\sigma \in \mathbb{S}_n$ will also be members of a common tabloid σ ({t}). The set of tabloids of shape λ form **a homogeneous space of the symmetric group** $\mathbb{S}_n$.

The row stabiliser R_t of a tableau t of shape λ is defined as the subgroup of $\mathbb{S}_n$ which leaves the rows of t invariant, i.e., only permutes numerals within each row. **The column stabiliser** C_t is analogous to the row stabiliser.

The permutation module denoted M^λ is the submodule of $\mathbb{C}[\mathbb{S}_n]$, formed from the elements corresponding to tabloids of shape λ. Permutation modules M^λ are reducible, and they are used to construct irreducible representations of $\mathbb{S}_n$.

The Specht module S^λ is an irreducible submodule of each permutation module M^λ.

A complete set of irreducible representations is formed from the G-modules of G from the collection of Specht modules $\{S^\lambda\}^\lambda$.

A polytabloid is a linear combination of tabloids that is antisymmetric with respect to this subgroup: $e_t = \sum_{\pi \in Ct} sgn(\pi)\pi(\{t\})$. For example, the polytabloid corresponding to the previous example is:

$$e_t = \begin{array}{|c|c|c|}\hline 3 & 2 & 5 \\ \hline 1 & 4 \\ \cline{1-2}\end{array} - \begin{array}{|c|c|c|}\hline 1 & 2 & 5 \\ \hline 3 & 4 \\ \cline{1-2}\end{array} - \begin{array}{|c|c|c|}\hline 3 & 4 & 5 \\ \hline 1 & 2 \\ \cline{1-2}\end{array} + \begin{array}{|c|c|c|}\hline 1 & 2 & 5 \\ \hline 3 & 4 \\ \cline{1-2}\end{array}$$

Some examples are: for shape λ = n, $M^{(n)}$ corresponds to the trivial one-dimensional representation and is irreducible such that $M^{(n)} = S^{(n)}$. for shape λ = n-1, $M^{(n-1,1)}$ contains a copy of trivial Specht module $S^{(n)}$, such that $M^{(n-1,1)} = S^{(n)} \oplus S^{(n-1,1)}$. This continues until the opposite case of $M^{(1,1,\ldots,1)}$ as the regular representation decomposes into the direct sum of S^λs where λ runs over all partitions of n.

Standard tableaux is defined such that numbers in each row and each column are increasing. **The standard irreducible representation** of $\mathbb{S}_n$ is defined from the linearly independent polytabloids corresponding to the standard tableaux of a given shape.

Young's orthogonal representation (YOR) for $\mathbb{S}_n$ is simpler than the standard irreducible representation, such that the rows of its representation matrices are orthogonal. It is based on labelling the dimensions of ρ_λ by standard tableaux of shape λ, without invoking polytabloids or the group algebra. It is restricted to the adjacent transpositions $\{\tau_1, \tau_2, \dots, \tau_{n-1}\}$, we can specify the matrix entries of $\rho_\lambda(\tau_k)$ explicitly. These matrices were proven to be very sparse such that the only non-zero entries in any row, indexed by the standard tableau t, are the diagonal entry $[\rho_\lambda(\tau_k)]_{t,t} = \frac{1}{d_t(k,k+1)}$, $d_t(k, k+1)$ is a special signed distance defined on Young tableaux.

The irreducible representations of GL_n are labeled by partitions λ, similar to $\mathbb{S}_n$. **Young symmetrizer** acts on basis vectors to form equivalence classes corresponding to the analogues of polytabloids. This yields the **Weyl module** that is considered the irreducible GL_n-module. It has the following structure that shows a subgroup of GL_n to be isomorphic to GL_{n-1} as a form of embedding:

$$M = \begin{bmatrix} m_{1,1} & \cdots & m_{1,n-1} & 0 \\ \vdots & \ddots & \vdots & 0 \\ m_{n-1,1} & \cdots & m_{n-1,n-1} & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}$$

The irreducible representation of GL_n takes the form D_λ can be constructed by a Gelfand-Tsetlin basis adapted to the tower of subgroups: $GL_n > GL_{n-1} \times \mathbb{C}^* > GL_{n-2} \times (\mathbb{C}^*)^2 > \cdots > (\mathbb{C}^*)^n$.

5.2 Harmonic analysis

Intuitively, Harmonic Analysis is the study of symmetry originally introduced from music theory. According to the Nöether theorem, behind every conservation principle in physics, from conservation of energy to linear and angular momentum, lies a symmetry problem. Mahadevan book (Mahadevan, 2008) explains how machine learning algorithms perform representation discovery using harmonic analysis utilizing the Fourier Transforms and wavelet analysis. The harmonic analysis maps a phenomenon that occurs over space and time into a frequency-oriented coordinate system, a change of basis or coordinate systems as discussed previously. Representation discovery is enabled by learning from the data a (very small) orthogonal basis functions Φ that span a set of invariant subspaces that can uniquely generate/reconstruct all the data. These basis functions extract regularities from data and summarise them by projecting them into invariant subspaces.

An example of an invariant subspace is the one-dimensional space spanned by an eigenvector associated with a specific eigenvalue of a matrix. The choice of the right representative basis affects the efficiency of the machine learning, optimization, or search algorithms in terms of accuracy and storage requirements. The Fourier analysis provides one form of a small set of bases in the frequency domain. The spectral analysis provided by the

Laplacian operator analyses data in terms of its projection to orthogonal subspaces to identify structures and clusters. The wavelet analysis provides another multiscale basis discovery suitable to functions on data with discontinuities. In the context of data on groups, the natural choice of subspaces is called the isotypals, which are the irreducible subalgebras in the corresponding decomposition, as explained earlier in this chapter. Approximation methods can be used to scale basis construction to large graphs, including exploiting symmetries in graphs matrix sparsification, low-rank approximation, graph partitioning, and Kronecker product approximation.

The group theoretic definition of Harmonic Analysis is that it is a projection of functions on an orthonormal basis. The convolution theorem, in particular, tells us that a mapping or function F is special because it corresponds to a decomposition of L(G) (the class of all complex-valued functions on G into a sum of spaces closed under convolution (Risi Kondor, 2008).

A dataset in the form of a signal can be of various types. The following are some examples:

- Audio (sound waves that are generated from the vocal tract, musical instruments or another audio recording and sensed by the ear),
- Video (continuous stream of images that are electromagnetic radiation that the eye can sense),
- Physical or mechanical interaction can be viewed as processes where a quantity: air pressure, electromagnetic field, physical bodies or their positions are changing as a function of time, in which the harmonic analysis tests the response of these structures to loads that vary sinusoidally with time to predict if resonance would occur.
- Time-series, or any structures that experience vibrations or cyclic loadings, such as bridges, engines and traffic flow problems.

Starting from the number systems, a number representation such as 3 (decimal), III (Roman), and 011 (Binary) all represent that same object. The decimal representation uses the place-value notation uses basis functions of 1, 10, 100, … and so on. So, number 232 is expanded/**analysed** as $2*(10)^2+3*(10)^1+2*(10)^0 = 232$. Each coefficient is calculated/**synthesized** by dividing by the basis function relevant to its place/position, such as $232/ (10)^2=2$ for the third position number. This analysis/synthesis is what learning the correct basis can do to a dataset by summarising it and identifying symmetries such that storage requirements can be reduced. For example, as we studied in chapter one, all vectors are synthesized from their coordinate basis e_i. Another example, a vector v describing an object such as a function on a graph, is synthesized from basis functions $\Phi = \{\phi_1, \dots\}$, as a linear expansion $v = \sum_i \alpha_i \phi_i$, where each coefficient α_i can be viewed as a "measurement" of the object. This makes v **analysed** as linear functionals $v = \{\langle v, \phi_1\rangle, \dots, \langle v, \phi_n\rangle\}: V \times V \rightarrow \mathbb{R}$. This is expressed as linear functions of inner product form: $\alpha_i = \langle v, \psi_i\rangle$, where ψ_i are the synthesis features, which are the dual basis to ϕ_i. This makes the object **synthesized** as $\hat{v} = \sum_i c_i \psi_i = \sum_i \langle v, \phi_i\rangle \psi_i$, where c_i are the coefficients of the

measurements in the dual space. This equation is defined as a vector's abstract Fourier series expansion. It is a full reconstruction when all synthesis coefficients are used in the finite dimension. If fewer c_is are selected, then this is an approximate reconstruction that can be applied using various constraints such as orthogonality, compactness, sparsity, and complexity, and these may be in conflict. One basis selection strategy is to use the bases with the largest inner product. Another approach is the usual iterative approach to minimise the error ϵ in reconstruction such that the choice of bases in the summation index i will produce a reconstruction $\hat{v} = \sum_i c_i \psi_i$, with $\|v - \sum_i c_i \psi_i\| \leq \epsilon$.

Fourier transform performs a change of basis to global basis from the space or time domains to the frequency domain. Due to its inability to handle smooth functions with local discontinuities and cannot reveal multi-scale regularities, wavelet analysis transforms the data from space or time to combined multi-scale space-frequency or time-frequency scales using a graph-based approach called diffusion wavelets. Fourier and wavelet analysis can be generalised from Euclidean spaces to non-Euclidean spaces defined by graphs, groups, and manifolds, enabling new basis discovery techniques to be developed in discrete data and search spaces. Abstract harmonic analysis on finite Abelian and non-Abelian groups and its applications are further studied in (Mahadevan, 2008) and (Stanković et al., 2005).

5.2.1 Fourier transforms

Fourier transforms real-valued functions and decomposes them as linear combinations of highly symmetric trigonometric functions. The change of basis of space or time in x (based on the nature of the dataset) to the frequency domain k using sines and cosines as basis functions is defined as the mapping $\mathcal{F}: f \rightarrow \hat{f}$. For example, a function f in the space or time domain is considered to be composed by summing several sinusoids of different frequencies, as shown in Figure 6 for two sine waves. This means any periodicity/symmetries in the data are captured as frequency amplitude, and the period is the frequency width, providing a compressed representation from which the original space or time data can be reconstructed by inverse transform. Example applications are the PCA, SVD, as explained in chapter two, and time-series and image-compression using FFT, manifold and graph-based methods such as diffusion maps, ISOMAP, LLE, and Laplacian eigenmaps (Spectral embedding), among many more in various disciplines. Chapter two, accompanying source code, showed a number of algorithms in the 2-way matrix form that belong to this class. Ch5.ipynb shows more examples using an audio dataset, applying harmonic analysis features extraction methods such as STFT (short-time Fourier transform) and Continous and Discrete Wavelet transform and their reconstruction errors and their effects on classification models learning time and accuracy.

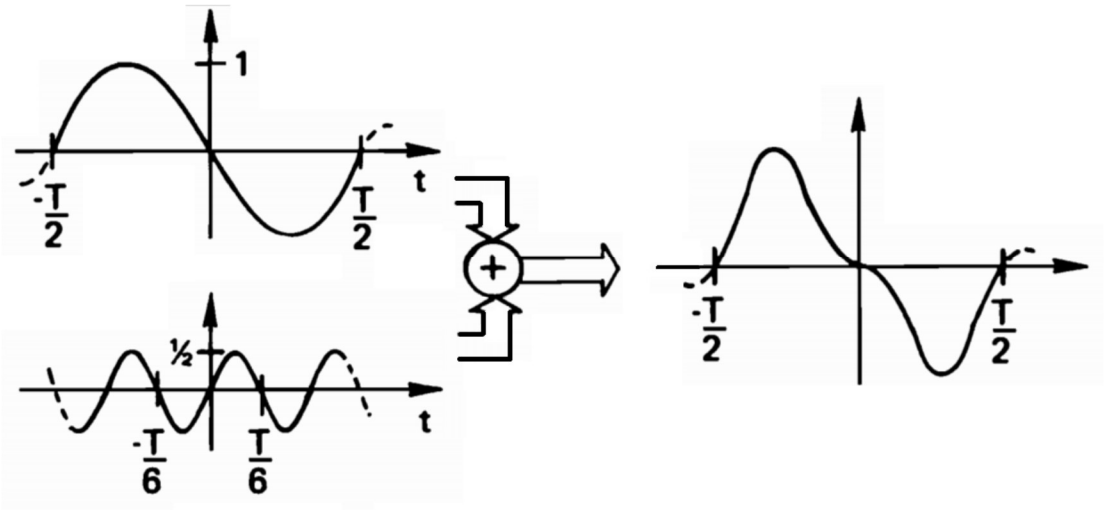


Figure 6: Two sine waves on the right-hand side are summed together to form aA function in the time domain on the left-hand side adapted from (Brigham, 1988)

The Fourier transform is defined as:

$$\hat{f}(k) = \int e^{-i2\pi kx} f(x) dx$$

The Fourier series is a special case of FT, calculated as follows:

$$\hat{f}(k) = \frac{1}{2\pi} \int_0^{2\pi} e^{ikx} f(x) dx$$

The inverse FT is defined as the inverse mapping $\mathcal{F}^{-1}: \hat{f} \to f$ as follows:

$$f(x) = \sum_{k=-\infty}^{\infty} \hat{f}(k) e^{ikx} = \int e^{2\pi ikx} \hat{f}(k) dk$$

This requires that the input function f(x) has some integrability properties for the forward transform and continuity properties for the inverse. Working in the new frequency domain is equivalent but more compact than in the measurements domain; this includes the inner product norm, translation property and convolution property. Parseval's theorem or Plancherel's theorem proves that the inner products to measure similarities between two functions before and after the transform are unitary, such that $\langle f . g \rangle = \langle \hat{f} . \hat{g} \rangle$. The translation property means if f translates by t, $f^t(x) = f([x - t]_{2\pi})$, where $[.]_z$ means modulo z, then $\hat{f}^t(k) = e^{2\pi itkx} \hat{f}(k)$. A shift by α in the space or time domain is multiplication by exp $(-i\alpha k)$ in the frequency domain $f(x - \alpha) = \hat{f}(k) e^{-i\alpha k}$. A scaling by α in the space or time domain is scaling by the reciprocal of α in the frequency domain $f(\alpha x) = \frac{1}{|\alpha|} \hat{f}\left(\frac{k}{\alpha}\right)$. This is extended to the convolution theorem such that the convolution of f(x) by g(x) in the space or time domain is just a multiplication of their frequency domain

transforms: $\mathcal{F}(f * g)(k) = \hat{f}(k) \cdot \hat{g}(k)$. Also, the derivative of a function in the time domain is equivalent to multiplying the frequency domain transform by the frequency value raised to the power of the order of the derivative: $\left(\frac{d}{dx}\right)^m f(x) \Longrightarrow (ik)^m \hat{f}(k)$.

The FT is essential to the study of partial differential equations because when the underlying space has a natural differential structure, the FT relates to that in a canonical way: on the unit circle (parametrised by angle) we have $\widehat{\partial f}(k) = ik\ \hat{f}(k)$, while on the real line $\widehat{\partial f}(k) = 2\pi ik\, \hat{f}(k)$.

The group theoretic FT is based on the fact that the domain of f is a group G, such that $\mathbb{T} = \mathbb{R}/\mathbb{Z}, \mathbb{R}$ and $\mathbb{Z}_n$, these factors are the irreducible characters of G. This redefines FT as: $\hat{f}(\chi) = \mathcal{F}(f)(\chi) = \int_G \chi(x) f(x) d\mu(x)$, where χ ranges over the characters of G, and μ is the Haar measure on G. Since the irreducible characters of an LCA group form a dual group $\hat{G}$, the inverse transform is given by a similar integral over the dual group: $f(\chi) = \mathcal{F}^{-1}(\hat{f})(\chi) = \int_{\hat{G}} \chi(x^{-1}) \hat{f}(x) d\hat{\mu}(x)$. This extends the harmonic analysis to LCA groups while maintaining all its properties by taking advantage of the fact that the irreducible characters of an Abelian group form an orthonormal basis for the group algebra. Using group and representation theory properties, a representation of f(x) can be restricted to $\rho(x)$ such that the FT can be defined as $\hat{f}(\rho) = \sum_{x \in G} f(x)\rho(x), \rho \in \mathcal{R}$, and the inverse is $f(x) = \frac{1}{|G|} \sum_{\rho \in \mathcal{R}} d_\rho tr\left[\hat{f}(\rho)\rho(x^{-1})\right]$,.The derivation is provided in (Risi Kondor, 2008). The convolution of a group is defined as $(f * g)(x) = \sum_{y \in G} f(xy^{-1}) g(y)$, satisfying the convolution theorem $\widehat{f * g}(\rho) = \hat{f}(\rho)\hat{g}(\rho)$, and consequently also the left- and right-translation properties $\hat{f}^t(\rho) = \rho(t)\hat{f}(\rho)$ and $\hat{f}^{(t)}(\rho) = \hat{f}(\rho)\rho(t)$.

An example application of Fourier analysis on groups is defined as studying the effect of sunlight(s), weed killer(w) and fertilizer(r) on the yield of wheat measured by $f(s, w, r)$. These three variables are assumed binary (high/low) or (+,-) for simplification and to reduce to a 2^k factorial design class of problems. For 3 variables, there are 2^3 possible permutations f_{+++}, f_{++-}, f_{+-+}, … etc. The different linear combinations of these variables contain the zeroth order that computes the mean wheat yield independent of the effects of the variables, producing a grand mean as:

$$\mu = \frac{1}{8}(f_{+++} + f_{++-} + f_{+-+} + f_{+--} + f_{-++} + f_{-+-} + f_{--+} + f_{---})$$

First-order effects consider each factor in isolation and compute its mean yield. For example, the mean for the first factor (sunlight/s) is defined as:

$$\mu_s = \frac{1}{4}(f_{+++} + f_{++-} + f_{+-+} + f_{+--}) - \frac{1}{4}(f_{-++} + f_{-+-} + f_{--+} + f_{---})$$

Similar first-order effects are computed for the other two variables. A second-order effects measure the different combination of two variables' effects on the wheat yield producing a mean relative to the two variables specified. For example, the mean for the combined effect of the first and second variables is defined as follows:

$$\mu_{sw} = \frac{1}{2}\left(f_{+++} + f_{++-}\right) - \frac{1}{2}\left(f_{+-+} + f_{+--}\right) - \frac{1}{2}\left(f_{-++} + f_{-+-}\right) + \frac{1}{2}\left(f_{--+} + f_{---}\right)$$

Notice that we add the partial means when both variables are positive or negative and subtract when they are different. The remaining two combinations of 2 variables can be computed similarly. The final third-order measures the effects of all variables combined on the wheat yield, producing the mean:

$$\mu_{swr} = \frac{1}{8}\left(f_{+++} - f_{++-} + f_{+--} - f_{+-+} + f_{-+-} - f_{-++} + f_{--+} - f_{---}\right)$$

Notice the rearrangement of the (+/-)combinations and subtractions when two variables are negatives (low). These three orders of effects and their variable combinations give a complete representation of the data, from which the original data can be reconstructed from them. This representation can be defined in a transform matrix form denoted $\mathbb{Z}_2^3$ as follows:

$$\begin{bmatrix} \mu \\ \mu_s \\ \mu_w \\ \mu_r \\ \mu_{sw} \\ \mu_{sr} \\ \mu_{wr} \\ \mu_{swr} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} f_{+++} \\ f_{++-} \\ f_{+-+} \\ f_{+--} \\ f_{-++} \\ f_{-+-} \\ f_{--+} \\ f_{---} \end{bmatrix}$$

The vector of μ's is the Fourier transform of f over this group. The specific subindices can be transformed to index k as follows: $\mu_k = \sum_{x \in \mathbb{Z}_2^3} \chi_k(x) f(x)$, such that the order of k establishes an isomorphism $x \to \chi_k$ from $\mathbb{Z}_2^3$ to its dual, with the identity being (+,+,+), the norm on $\mathbb{Z}_2^3$ is the count of negative components mapped to the dual. The zeroth-order effect will have norm 0; the first-order effects will have norm 1, …, etc. This is similar to analyzing functions on $\mathbb{R}^n$ in terms of a hierarchy of Fourier components of increasing frequency. Check another Symmetric group example in (Risi Kondor, 2008).

Fourier transform connects continuous mathematics, such as linear differential equations, to concepts in discrete mathematics, such as linear algebra and matrix theory, using the principle of diagonalisation. Diagonalising continuous spaces using a discrete version of the "Laplacian" operator yields a discrete FT. This operator is commonly called the graph Laplacian and is explained below.

Other FT algorithms are defined for various applications and for computational efficiencies, such as the Fast Fourier Transform (FFT), which reduces the computation from $O(n^2)$ to $O(n \log_2 n)$, where n is the number of data samples in f(x). A C++ implementation of FFT on the symmetric group using the Clausen FFT algorithm is provided by the author of (Risi Kondor, 2008) and is available in https://people.cs.uchicago.edu/~risi/SnOB/.

5.2.2 Laplace on functions and graphs

The Laplace operator is another tool for identifying symmetries. A Riemannian manifold is represented using an undirected graph of the points in the manifold. The graph Laplacian (as defined below) is of interest in several areas, including dimensionality reduction, Markov processes, spectral graph theory, and web page ranking.

In chapter three, the continuous Laplace operator was provided using the divergence operator of the gradient of a function and the trace of the Hessian matrix of the function, which is for a function of two variables is $\frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2} = tr(H(f))$, which is invariant to coordinate/basis change and measures function curvature. For a given graph G = (V, E, W), where |V| = n nodes, E is the edges list, an edge between nodes (u, v) ∈ E, edges' weights are given in the W matrix. Another form is to merge E & W and use adjacency matrix A, which includes either 1 or 0 in unweighted graphs or the weight value in the case of weighted graphs. Also, for directed graphs, the weight or the adjacency can be non-symmetric as the existence of (u, v) edge does not imply the existence of a (v, u) edge. For u ~ v, meaning an (undirected) edge between u and v, the degree of u is defined to be $d(u) = \sum_{u \sim v} w(u, v)$. D will denote the diagonal matrix defined by D_{uu} = d(u), as the row sums of W or A. Whether the graph is defined with an adjacency list or adjacent matrix, the discrete version of the Laplace operator can be defined as L = D-W or L = D-A. The normalized Laplacian $\mathcal{L} = D^{-1/2}(D - W)D^{-1/2}$. The Laplacian is an operator on the space of functions $F: V \to \mathbb{R}$ on a graph, such that $Lf(u) = \sum_{u \sim v}(f(u) - f(v))\ w(u, v)$, replacing the continuous derivatives with finite differences.

The fundamental property of the graph Laplacian is that projections of functions on the eigenspace of the Laplacian produce the smoothest global approximation respecting the underlying graph topology. This constructs the basis functions that best capture the graphs' structure due to the close Laplacian connection with classical Fourier analysis, the continuous Laplacian on manifolds, and random walks. Whether the normalised Laplacian is used or not, both have non-negative eigenvalues, with corresponding eigenfunctions that can be used as an orthonormal basis of smooth global functions to approximate any function on a graph. The projection of a function f on the smallest set S of the indices to produce the smallest acceptable error onto the top k eigenvectors corresponding to the smallest eigenvalues of the Laplacian is the smoothest approximation to f. The second eigenvector is the Fiedler eigenvector, which is the eigenvector associated with the smallest nonzero eigenvalue of the Laplacian matrix and is generally used to partition the graph. It

characterises the sensitivity or a bottleneck in a graph that can be used in clustering or partitioning the graph. The basis functions are then computed by diagonalizing a Laplacian/ diffusion operator on the space of functions on the graph.

The Laplace operator can be calculated from the Fourier analysis by taking the Laplacian and finding its eigenvectors. It can also be calculated by doing a multi-scale wavelet analysis, which takes the powers of the random walk Laplacian and builds multi-scale representations at different spatial and temporal levels.

Computing the Laplacian on a group, we turn the group into an undirected graph and defines the neighbourliness of x∼y, denoting the fact that x, y ∈ G are neighbours, and letting d_x denote the number of edges incident on x, then the graph Laplacian as the |G|×|G| matrix:

$$\Delta_{x,y} = \begin{cases} 1, & if\ \mathrm{x} \sim \mathrm{y} \\ -dx, & if\ \mathrm{x} = \mathrm{y} \\ 0, & otherwise \end{cases}$$

The diffusion kernels generalise the Gaussian kernels induced by captured local similarities by the Laplacian while regularising the function. The diffusion kernel of the above Laplacian is:

$$k(x,y) = \left[e^{\beta\Delta}\right]_{x,y} = \left[\lim_{n\to\infty}\left(1+\frac{\beta\Delta}{n}\right)^n\right]_{x,y}.$$

The Laplacian on functions on G is defined by $(\Delta f)(x) = \sum_{y\in G}\Delta_{x,y}f(y)$, the kernel is defined as: $\langle f, \Delta f \rangle = -\frac{1}{2}\sum_{\mathrm{x}\sim\mathrm{y}}(f(x)-f(y))^1$, such that $-\Delta$ measures how f violates the graph structure. Computing the Laplacian eigenvalues $\lambda_1, \ldots, \lambda_{|G|}$ and eigenvectors $v_1, \ldots, v_{|G|}$, we can express the kernel as:

$$K = \sum_{i=1}^{|G|} e^{\beta\lambda_i} v_1 v_i^T$$

And the discrete regularisation operator as:

$$\Upsilon = \sum_{i=1}^{|G|} e^{\beta\lambda_i/2} v_1 v_i^T$$

This diffusion kernel penalizes functions according to how much energy they have in the "high frequency" modes on the graph, i.e., the ones that violate many edges. For more examples and discussion of various applications, check (Risi Kondor, 2008).

The Hodge theorem states that the eigenfunctions of the Laplacian on a manifold provide a complete discrete basis for all square-integrable functions on a continuous manifold. The key problem in extending eigenfunctions to continuous spaces is how to extend sample values to new unobserved values: out-of-sample extension. Nyström interpolation provides a solution to the out-of-sample extension.

Spectral data embedding, as shown in chapter two's source code examples (Maniflods.ipynb), clustering, and edge detections, are general applications of the Laplacian. Scipy python implements the graph laplacian in their compressed sparse graph

routines package, such as “scipy.sparse.csgraph.laplacian” function. OpenCV also implements the function. Both examples are demonstrated in ch5.ipynb accompanying source code examples.

5.2.3 Wavelet Analysis

Fourier Analysis use global basis functions of sines and cosines. Hence, they can transform stationary signals only. Stationary signals' amplitude and frequency do not change over time. Short Short-Time Fourier Transform (STFT) divide the time series into windows and apply FT on each window to identify its frequency amplitude. This produces uncertainties on the overlapping window sizes and variable frequencies within. Consequently, they can not adequately handle local discontinuous functions and multi-scale regularities. Instead of mapping space-time x to the frequency domain k, the wavelet analysis combines time and space into multi-scale/multi-resolution time-frequency or space-frequency of varying granularities.

Furthermore, the literature proposes variable basis functions called wavelets instead of using trigonometry basis functions. Also, a concept of dilation equations using a diffusion operator on the graph is used instead of diagonalisation and differential equations used in FT. The random walk method is used to construct the basis elements at multiple levels of spatial and temporal abstractions by constructing each level by dilating/scaling the ones at the previous level. This constructs a hierarchy of vector spaces and two sets of basis functions called scaling functions representing coarser views and wavelets representing the finer detailed view. The computation cost is reduced using matrix compression, sampling, and domain knowledge concepts.

In convolution, a filter (matrix) is multiplied by a region in a larger data matrix to identify the presence of the object of the filter content in this region, then slides over the remaining temporal or spatial regions to provide translation invariance. Similarly, different scales of the different wavelets are convoluted over the time or space data series to identify the location of the wavelet presence is found. Then all are summed up to identify the multi-scale/multi-resolution combined frequency-time or frequency-space domain.

The Haar basis is the earliest, simplest, and most adopted example of a wavelet basis. For a one-dimensional x variable, a decimal value of 7 can be decomposed using the Haar basis for 1D as:

$$7 \; = \; 4\phi(x) + 3\psi(x)$$

Such that $\phi(x) = \begin{cases} 1\,, & for\ 0 \le x < 1/2 \\ -1, & for\ 1/2 \le x < 1 \\ 0, & otherwise \end{cases}$, $\psi(x) = \begin{cases} 1, & 0 \le x < 1 \\ 0, & otherwise \end{cases}$, are the mother scale function and mother wavelet function, respectively. Other basis functions could be

derived by scaling and translating them. The process is to repeat this average & difference process to obtain one scale coefficient and N-1 detail coefficients for N signals.

We can do this using the linear mapping transformation matrix from Vector space V to vector space W, as explained in chapter one $T: V \to W$, also called operator. To make the mapping apply basis changes from $B_1 = \{u_1, \dots, u_n\}$to $B_2 = \{v_1, \dots v_n\}$, it is denoted $[T]_{B_1}^{B_2}$. Given a vector, a in V expressed as $a = a_1u_1 + \cdots + a_nu_n$, a transformation to vector b in W would include the basis transformation as $b = b_1T(u_1) + \cdots + b_nT(u_n)$, such that $T(u_i) = u_i'$, and $u_i' = \sum_{i=1}^{m} \alpha_1 v_1$ is the direct sum of vectors, in case the dimensionality of the input space n is different from the dimensionality of the output space m, α_i are the transformation coefficients. For example, Haar basis for $\mathbb{R}^4$ is defined as:

$$e_{h_1} = \begin{bmatrix}1\\1\\1\\1\end{bmatrix}, e_{h_2} = \begin{bmatrix}1\\1\\-1\\-1\end{bmatrix}, e_{h_3} = \begin{bmatrix}1\\-1\\0\\0\end{bmatrix}, e_{h_4} = \begin{bmatrix}0\\0\\1\\-1\end{bmatrix}$$

The transformation matrix from the Haar unit basis H to the identity unit basis U in $\mathbb{R}^4$,

$[I]_H^U \begin{bmatrix}1 & 1 & 1 & 0\\1 & 1 & -1 & 0\\1 & -1 & 0 & 1\\1 & -1 & 0 & -1\end{bmatrix}$ and the inverse $([I]_H^U)^{-1} = [I]_U^H =$
$\begin{bmatrix}0.25 & 0.25 & 0.25 & 0.25\\0.25 & 0.25 & -0.25 & -0.25\\0.5 & -0.5 & 0 & 0\\0 & 0 & 0.5 & -0.5\end{bmatrix}$

The rows of the inverse of the Haar basis change matrix are the dual basis of the Haar basis, which are the scaled (÷0.25) of the original Haar basis because they are orthogonal. These bases are more efficient than the unit vector basis. For example, given a vector v in $\mathbb{R}^4$ as $[5, 4.5, -4, -5]$, can be represented using a linear combination of the Haar basis as:

$$v_H = ([I]_H^U)^{-1} v_U \begin{bmatrix}0.25 & 0.25 & 0.25 & 0.25\\0.25 & 0.25 & -0.25 & -0.25\\0.5 & -0.5 & 0 & 0\\0 & 0 & 0.5 & -0.5\end{bmatrix} [5 \quad 4.5 \quad -4 \quad -5] = \begin{bmatrix}0.125\\4.625\\0.25\\0.5\end{bmatrix}$$

One basis vector, the second of 4.625, dominates the rest, and we can construct a reasonable approximation of this vector to get the approximation $v = \begin{bmatrix}5\\4.5\\-4\\-5\end{bmatrix} \cong$
$4.625 \begin{bmatrix}1\\1\\-1\\-1\end{bmatrix} = \begin{bmatrix}4.625\\4.625\\-4.625\\-4.625\end{bmatrix}$, where the error in approximation is around 0.8. Also, this basis

transformation allows decomposition into the product of smaller matrices, and its inverse is its transpose because it is orthogonal (Mahadevan, 2008).

Other wavelet basis functions are introduced in the literature to choose from, such as db4, db16, coif1, sym4, sym8, bior1.3, and bior3.1. Scipy.signal Python package has continuous wave transform (cwt) using wavelets Morlet and Ricker. PyWavelets is another Python package that implements more wavelets as rbio, dmey, gaus, mexh, morl, cgau, shan, fbsp, cmor. Also, diffusion wavelet bases are adapted to the geometry of a graph and can be learned adaptively from sampling a data set or a state space. A reader can research how each one is designed and what useful properties they offer to be suitable for a particular application. Ch5.ipynb accompanying source code examples include some of these wavelet functions to compare to stft feature representation method and their effects on reconstruction error, classification learning time, and accuracy. The models presented are very simple to introduce the idea, but each model can be enhanced in many other ways to achieve higher accuracy and less convergence time.

5.3 Learning on groups

As explained in chapter one, estimating a mapping function $f: \chi \to \Upsilon$ between the input space χ, and the output space Υ is the aim of machine learning (ML) algorithms. Chapter two introduced various input vector space mapping $f: \chi \to \tilde{\chi}$ learning to estimate the transformation matrix between both spaces aiming at reducing the dimensionality. ML algorithms generalise the mapping from input to output for prediction, regression, and other objectives. Most ML algorithms are based on a probabilistic framework assuming a single probability distribution D on $\chi \times \Upsilon$ in which both training and testing examples are drawn, each sample entity being statistically independent of the others. In predictive models such as the frequentist probability methods and supervised machine learning, the distribution is not important to be understood. The Bayesian methods, on the other hand, expect the distribution to be pre-defined and parameterised to the algorithm, to build a generative model that estimates the spaces of $\chi \times \Upsilon$. In both cases, an ML algorithm would learn a mapping function $f: \chi \to \Upsilon$ to predict a y for a given test x, from a set of functions known as a hypothesis set. The training phase, given a loss function L defined over the space: $L: \Upsilon \times \Upsilon \to \mathbb{R}$, aims to minimise the expected value over the distribution D of the input and output spaces $\min E_{(x,y)\sim D}[L(\mathrm{f}(x), \mathrm{y}]$. The various machine learning algorithms parametrise the loss function to use, the distribution of the input and output spaces if required by the algorithm, and other detailed requirements to accommodate different classes of problems.

If the model overfits the coefficients of the mapping function, then a regularisation parameter λ will attempt to reduce the values of the coefficients, and the regularizer $\Omega[f]$ measures the smoothness of the function. On the other side, Kernel methods are based on

reproducing kernel Hilbert spaces. This class of algorithms belong to the set $\Omega[f] = \langle f, f \rangle$ for some appropriately defined inner product between hypotheses, and trade-off the smoothness of the function and performance of the model.

5.3.1 Hilbert space learning algorithms

Chapter two introduced the kernel methods for SVM and stated that the data need to be mapped to the Hilbert space of infinite dimensions such that a non-linearly separable dataset can become linearly separable in the higher dimension. An exhaustive search on the best higher dimension would grow to the largest dimensional space is $\mathbb{R}^{|\chi|}$, where $|\chi|$is the cardinality of input space χ. This can be further reduced to $\mathbb{R}^{|w|}$, where $|w|$ is the cardinality of the parameters to learn w, for a reduced hypothesis set. Kernel functions even reduce this space using inner products between input space samples.

The previous definition is good enough to use these methods properly. Since this book deep dives, to an extent, into the new layer of abstract algebra and how it has advanced computer science algorithms in the past three decades, this section will explain further how to define an inner product on Hilbert spaces.
Remember from chapter one that a scalar product on $\mathbb{R}^n$ is defined as $(v|w) = \sum_{i=1}^{n} v^i w^i$. A Hermitian scalar product on a function such as L^2([−a,a]) on $\mathbb{C}^n$, extends the definition of the scalar product on $\mathbb{R}^n$, to be defined as $(v|w) = \sum_{i=1}^{n} \bar{v}^i w^i$, where the bar on v denotes complex conjugation (Hermiticity). L^2([−a,a]) is a square-integrable complex-valued functions on an interval [-a, a] with expansion $f = \sum_{n=-\infty}^{\infty} c_n \, e^{i\frac{n\pi x}{a}}$. This is known as the Fourier series of f, and c_n, are the Fourier coefficients, which are the components of the vector f in the basis $\left\{e^{i\frac{n\pi x}{a}}\right\}_{n\in\mathbb{Z}}$. L^2([−a,a]) structure is an infinite-dimensional Hilbert space because the basis allows for infinite linear combinations.

Given f, g ∈ L^2([−a,a]), $(f\ |g) \equiv \frac{1}{2a}\int_{-a}^{a} \bar{f} g dx$, where a bar on f denotes hermiticity again, defines the inner product (.|.) on L^2([−a,a]). This inner product turns L^2([−a,a]) to a Hilbert space. A Hilbert space H has the set $\{e_i\} \subset H$ as orthonormal basis if $(e_i|f)=0$, $\forall i \Rightarrow f = 0$, such as $\left\{e^{i\frac{n\pi x}{a}}\right\}_{n\in\mathbb{Z}}$. L^2([−a,a]) is the set of all expressions of the expansion form defined above, and when the following condition is satisfied:

$$\frac{1}{2a}\int_{-a}^{a} |f|^2 dx = \sum_{n=-\infty}^{\infty} |c_n|^2 < \infty$$

L^2([−a,a]) then defines the infinite-dimensional Hilbert spaces. a basis for a Hilbert space is an infinite set whose infinite linear combinations, together with some suitable convergence condition, form the entire vector space. Dealing with spatial degrees of freedom, as opposed to 'internal' degrees of freedom like spin, Hilbert spaces like L^2([−a,a]) and $L^2(\mathbb{R})$

are often encountered, which are most conveniently described by 'basis' vectors which are eigenvectors of either the position operator $\hat{x}$ or the momentum operator $\hat{p}$.

There is an oft-unwritten rule that one should take the tensor product of the corresponding Hilbert spaces to add degrees of freedom. For example, by adding an additional translational degree of freedom for a spinless particle constrained to move in one x dimension; the quantum mechanical Hilbert space for this system is $L^2(\mathbb{R})$ with basis $\{|x\rangle\}_{x\in\mathbb{R}}$ (this is the ket notation explained in chapter one, which means x is a vector in vector space X) adding a second y dimension with its own Hilbert space $L^2(\mathbb{R})$ with basis $\{|y\}_{y\in\mathbb{R}}$, then the particle can move in these two dimensions in a Hilbert space defined by the dot product of each individual Hilbert space, $L^2(\mathbb{R}) \otimes L^2(\mathbb{R})$ with basis $\{|x\rangle \otimes |y\rangle\}_{x,y\in\mathbb{R}}$ and so forth for higher dimensions. Adding spinning requires adding basis in the $\mathbb{C}^3$ space (Jeevanjee, 2011).

Hilbert space is formally defined as a complete linear inner product space. Starting from the inner product on a vector space V over $\mathbb{R}$, it is defined as a function $V \times V \to \mathbb{R}$, denoted $\langle v, w\rangle$, satisfying:
$\langle u + v, w\rangle = \langle u, w\rangle + \langle v, w\rangle$,
$\langle \alpha u, v\rangle = \alpha\langle u, v\rangle$,
$\langle u, v\rangle = \langle v, u\rangle$,
$\langle u, u\rangle \geq 0$ with equality only when $u = 0_V$, for all $u, v, w \in V$ and $\alpha \in \mathbb{F}$. Such an inner product gives rise to the norm $\|u\| = \sqrt{\langle u, u\rangle}$, and this provides V with a topology and a distance metric $d(x, x') = \|x - x'\|$.

A sequence $a_1, a_2, \ldots$ in a metric space is said to **converge** to $a \in M$ if $\lim_{i\to\infty} d(a_i, a) = 0$ and is called a Cauchy sequence if $\lim_{\min(I,j)\to\infty} d(a_i, a_j) = 0$. The space M is said to be a complete metric space if every Cauchy sequence in M converges to some $a \in M$.

A Hilbert space is a vector space $\mathcal{H}$ that is a complete metric space because of the inner product-induced norm. The Hilbert space formalises the presence of three different but closely related structures: the vector space, the inner product, and the topology. A Hilbert space is defined over any field, not only $\mathbb{R}$, and can be finite or infinite dimensional. Any finite Hilbert space is isomorphic to the Euclidean space $\mathbb{R}^n$. In the infinite-dimensional, we can define a basis for $\mathcal{H}$, and if we label the basis vectors with elements of some countably or uncountably infinite set S, we can expand any $v \in \mathcal{H}$ as:

$$v = \sum_{s\in S} \alpha_s e_s \,, \alpha_s \in \mathbb{R}$$

This makes the Hilbert space defines a space of functions $v(s) = \alpha_s$, in which all linear algebra carries over to the infinite dimension. The Kernel functions defined on them are computationally efficient. As defined in chapter two, a kernel function is $k(x, x') = \Phi(x), \Phi(x')$. In chapter three, a pull-back is defined as moving a differential form from one manifold to another manifold. This is clearly what the kernel function k is doing, moving the metric from the Hilbert space inner product to the χ space. This makes the kernel function

$k: \chi \times \chi \to \mathbb{R}$ symmetric, $k(x, x') = k(x', x)$, with any linear combination $f = \sum_{i=1}^{m} \alpha_i \Phi(x_i)$, $\alpha_1, \dots, \alpha_m \in \mathbb{R}, x_1, \dots, x_m \in \chi$, must satisfy $\langle f, f \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j \langle \Phi(x_i), \Phi(x_j) \rangle \geq 0$, equivalent to $\sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j \langle x_i, x_j \rangle \geq 0$.

This adds the definition of positive as well and semi-definite because of accepting equal or greater than. Adding also the condition $\sum_{i=1}^{n} \alpha_i = 0$, makes this a symmetric conditionally positive semi-definite. These conditions are sufficient to have a Hilbert space corresponding to the kernel function k used to characterise the input space.

The steps below do the construction of **reproducing kernel Hilbert space (RKHS)** to characterise the function space corresponding to a given k as stated in (Risi Kondor, 2008):

1. Define the functions $k_x(\cdot) = k(x, \cdot)$, and form a vector space V with basis labelled by $\{k_x\}_{x \in \chi}$.
2. Define an inner product between basis vectors by $\langle k_x, k_{x'} \rangle = k(x, x')$ and extend this by linearity to the rest of V. This set of functions taken as basis will span the linear combinations of $k_x's$.
3. Finally, complete V to $\mathcal{H}$ by adjoining to it the limits of all Cauchy sequences and extending the inner product by continuity. Remember, a Cauchy sequence is an infinite sequence in which successive terms tend to get closer together and converge to a limit. This can reach the largest dimension $\mathbb{R}^{|\chi|}$ but on a linearly independent set.

The representer theorem of Kimeldorf and Wahba (1971) states that given χ be an input space, Υ an output space, $T = \{(x_i, y_i)\}_{i=1}^{m}$ a training set, $L: \Upsilon \times \Upsilon \to \mathbb{R}$ a loss function, and $\mathcal{H}$ a reproducing kernel Hilbert space induced by some positive definite kernel $k: \chi \times \chi \to \mathbb{R}$. Then the minimizer f_T of any regularized empirical risk (loss on the training data) functional of the from $R_{reg}[f] = \frac{1}{m} \sum_{i=1}^{m} L(f(x_i), y_i)) + \lambda \|f\|^2$
Such that the regularizer $\Omega[f] = \langle f, f \rangle$, identify $\Omega[f]$ with the squared Hilbert space norm $\|f\|^2 = \langle f, f \rangle$, with regularisation parameter λ>0 is expressible as a linear combination $f_T(x) = \sum_{i=1}^{m} \alpha_i k_{x_i}(x)$, $\alpha_i \in \mathbb{R}$. Some machine learning algorithms raises $\|f\|$ to a different power than 2, but 2 is sufficient according to the representer theorem.

This theorem reduces searching the large space $\mathcal{H}$ to just finding the optimal values of the m coefficients $\alpha_1, \dots, \alpha_m$. This is a non-parametric model because it does not attempt to fit a fixed model with a finite number of pre-defined parameters. The literature has various kernel functions, and continuous research will present more in the future. A number of them are introduced in chapter two, and more are discussed in (Risi Kondor, 2008). Each one of them applies a different regularisation scheme. The most popular kernel on $\mathbb{R}^n$, the Gaussian (RBF) kernel discussed in chapter two, contains a length scale parameter (or variance parameter) σ. Letting $\hat{f}$ denote the Fourier transform of f and $\hat{\Upsilon}$ the frequency space regularization operator $\widehat{\Upsilon f} = \hat{\Upsilon} \hat{f}$, that was shown that $(\hat{\Upsilon} \hat{f})(\omega) = e^{|\omega|^2 \sigma^2} \hat{f}(\omega)$ penalises high-frequency components in f by a factor exponential in $|\omega|^2$, providing natural

regularisation. An alternative description of Υ in terms of derivatives of f is $\|\Upsilon f\|_F^2 = \int_\chi \sum_{i=0}^{\infty} \frac{\sigma^{2i}}{i!2^n} \left\|\left(\mathcal{O}^i f\right)(x)\right\|_{L_2}^2 dx$, where for i even $\mathcal{O}^i = \Delta^{\frac{i}{2}}$, and for i odd $\mathcal{O}^i = \nabla\Delta^{\frac{i-1}{2}}$, and Δ is the Laplacian operator explained previously. This again penalises functions which have a large amount of energy in their high order derivatives, i.e. they are not smooth. For more discussions on how the choice of kernels and loss function define various machine learning algorithms such as Gaussian Processes, SVM, and Kernels for different learning problems, such as learning symmetries in data and translation invariance as in convolution neural networks (CNN) expressed using group theory, and learning permutations check (Risi Kondor, 2008).

These kernel machines, aka neural networks, build up the hypothesis set by going through the dataset in iterations to minimise the empirical loss. Their ability to generalise to unseen data as well is justified by the kernel function choice.

5.4 Invariance

In machine learning, data, for example, images must be represented individually in translation- rotation- and scale-invariant forms. Learning the representation invariant subspace requires a change of bases from the measurement basis to the transformation invariant basis. A vector is covariant if its components change proportional to the change of the base after a change of basis. It is a contravariant vector when its components change inversely proportional to the change in the basis. Representation discovery is the process of identifying an invariant subspace under some mapping operator T. A subspace V is invariant under T, when every vector v in V, when the operator is applied on it yields a vector w, it $Tv -> w$, such that w also is in V. Invariant subspaces are useful since they enable irreducible representations of linear mappings. As explained in the wavelet section, a transformation operator T in a given subspace χ is denoted $[T]_\chi = T|_\chi$. In previous chapters, we studied matrix and tensor decompositions that reduce a matrix/tensor into a linear sum of simpler rank-one matrices/tensors, which are essentially just the outer product of these simpler structures. For example, combining the Haar basis with its dual basis, construct a set of invariant subspaces that results in a direct sum decomposition of the original vector space V. Given the same vector used before $[v]_U = [5, 4.5, -4, -5]^T$, we can decompose this vector as the sum of four vectors, each produced from the outer product of the Haar basis vectors with their corresponding dual basis vectors multiplied by the invariant rank-one Haar representation $[v]_H$:

$$[v]_U = \begin{bmatrix} 5 \\ 4.5 \\ -4 \\ -5 \end{bmatrix} = \begin{bmatrix} 0.125 \\ 0.125 \\ 0.125 \\ 0.125 \end{bmatrix} + \begin{bmatrix} 4.625 \\ 4.625 \\ -4.625 \\ -4.625 \end{bmatrix} + \begin{bmatrix} 0.25 \\ -0.25 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.5 \\ -0.5 \end{bmatrix}$$

There are many methods to construct invariant subspaces, such as eigenspace decomposition in which the space spanned by the eigenvector is an invariant space to the input matrix, QR Decomposition and Gram–Schmidt Orthogonalization, and SVD. The

mapping to the Hilbert space, using the inner product norm, is a transformation to a coordinate-free or infinite dimensional invariant space. The abstract Fourier expansion $\hat{v} = = \sum_i \langle v, \phi_i \rangle \psi_i$, where ϕ_i represents the analysis basis space and ψ_i represents the synthesis basis space, generalises and projects to general Hilbert space, and constructs unknown vectors in the invariant space from known measurements in the input space. The Reproducing Kernel Hilbert Space (RKHS) is also a special class of the Hilbert space that uses a reproducing kernel function. The kernels induce Gram symmetric matrices G on a given set of samples,

$$G(\phi_1, \ldots, \phi_n) = \begin{bmatrix} \langle \phi_1, \phi_1 \rangle & \cdots & \langle \phi_n, \phi_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle \phi_1, \phi_n \rangle & \cdots & \langle \phi_n, \phi_n \rangle \end{bmatrix}$$

The Gram matrices enable extending basis functions computed on a set of samples in α basis to new points β basis, such as: $G(\phi_1, \ldots, \phi_n)\ \alpha = \beta$(Mahadevan, 2008).

As discussed earlier in this chapter, invariance can be captured in groups since group captures symmetries. The translation property of the Fourier Transform is explained earlier on groups and their representations as the left- and right-translation properties: $\hat{f}^t(\rho) = \rho(t)\hat{f}(\rho)$ and $\hat{f}^{(t)}(\rho) = \hat{f}(\rho)\rho(t)$ respectively, and was extended to convolution invariant as $\widehat{f * g}(\rho) = \hat{f}(\rho)\hat{g}(\rho)$. To generalise, a data represented as a group G acting transitively on a set χ, we may find invariants of functions $f: \chi \to \mathbb{C}$ with respect to the induced (translation or other) action $f \to f^g$ defined $f^g(x) = f(g^{-1}(x)), g \in G$. The simplest example is $G = \chi$ acting on itself by left-multiplication, in which case f^g is the left-translate of f.

This leads to defining any matrix-valued functional s: $L(G) \to \mathbb{C}^{d_\rho \times d_\rho}$, which obeys ρ-covariant, $s(f^t) = \rho(t)s(f)$, while ρ-contravariant functionals are defined to transform according to $s'(f^t) = s'(f)\rho(t)^\dagger$. For example, a ρ-contravariant function is s': $f \to \hat{f}(\rho)^\dagger$.

The power spectrum of f is defined as a system of translation invariant matrices; Given unitary irreducible representations $\rho \in \mathcal{R}$, the product of a ρ-contravariant and a ρ-covariant function is invariant to translation: $s'(f^t)s(f^t) = s'(f)\rho(t)^\dagger\rho(t)s(f) = s'(f)\ s(f)$. This defines the power spectrum of f as $\hat{a}f(\rho) = \hat{f}(\rho)^\dagger\hat{f}(\rho), \rho \in R$. For more details on the invariance on groups in different settings, read (Risi Kondor, 2008) and (Stanković et al., 2005).

5.5 Applications:

The thesis in (Risi Kondor, 2008) presented two learning applications on groups. We will review one application that uses the invariant features required for representing images. Any image is better transformed from the spatial x-y grid pixel values to the spectral/Fourier terms/frequency domains. This achieves the desirable properties of this transformation such as scale, shift, translate, rotation invariance and compression for applications such as edge

or motion energy detection, filtering, directional derivative, textural signature, statistical structure identification such as objects and Optical character recognition. An image is represented by a linear combination of basis functions: $f(x,y) = \sum_k a_k \Psi_k(x,y)$, as a 2D Fourier analysis, $\Psi_k(x,y) = \exp(i(u_k x + v_k y))$, where $\exp(i\theta) = \cos(\theta) + i\sin(\theta)$.

The transform finds a set of complex coefficients a_k for every spatial frequency and orientation in the 2D Fourier domain spanned by the 2D frequency variables (u_k, v_k). a_k are computed as the orthonormal projection of the function f(x) onto one complex exponential $\exp(-i(u_k x + v_k y))$. The Fourier Transform may compute these coefficients: $a_k = \int_X \int_Y \exp(-i(u_k x + v_k y))\ f(x,y)dxdy$, and $\hat{f}(u,v) = \int_X \int_Y \exp(-i(u_k x + v_k y))\ f(x,y)dxdy$, such that each $\hat{f}(u,v)$ is a complex coefficient which defines the magnitude and phase of a sinusoid basis function/coordinates called vector spatial frequencies. An array of them must span the (u,v) Fourier plane in a uniform Cartesian lattice. OpenCV Python package has a discrete FT function "dft" and its invserse "idft", and "numpy.fft" module contains discrete FT functions for 1D, 2D, ND arrays and their inverses.

These computed Fourier coefficients a_k are complex-valued. If the function f(x) is real-valued, then its frequency domain representation has two-fold redundancy. The real parts of the a_k have even-symmetry: $a_k = a_{-k}$, and their imaginary parts have odd symmetry: $a_k = -a_{-k}$. Given this "Hermitian" symmetry, all coefficients are obtained by computing only half of them.

Complex exponentials $\exp(i\theta)$, with real part as a cosine wave, and imaginary part as a sine wave, are the Eigenfunctions of linear systems $\exp(iu_k t) \to h(t) \to A\exp(iu_k t)$. Therefore, Fourier transform becomes a linear operation: $\mathcal{F}\big(\alpha f(x) + \beta g(x)\big) = : \alpha\mathcal{F}(f(x)) + \beta\mathcal{F}(g(x))$. The Euler relation $e^{i\pi} + 1 = 0$, connects the five most important mathematical constants and harmonic analysis of four branches of mathematics: 1) {0, 1} represents arithmetic, 2) π≈ 3.14 represents geometry, 3) $i = \sqrt{-1}$ represents algebra, and 4) e≈2.718 represents the analysis. FT can then be computed by taking the limit of $\left(1 + \frac{1}{n}\right)^n$ as $n \to \infty$, that can be computed using the power-series definitions for the transcendental functions: $exp\ (\theta) = 1 + \frac{\theta}{1!} + \frac{\theta^2}{2!} + \frac{\theta^3}{3!} + \cdots$, $cos\ (\theta) = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \cdots$, $sin\ (\theta) = 0 - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \cdots$.

The properties identified earlier for 1D FT are extended to 2D FT as follows:

- A shift by α, β in the spatial domain is multiplication by $\exp(-i(\alpha u + \beta v))$ in the frequency domain $f(x-\alpha, y-\beta) = \hat{f}(u,v)e^{-i(\alpha u + \beta v)}$,
- A scaling by α, β in the spatial domain is scaling by the reciprocal of α, β in the frequency domain $f(\alpha x, \beta y) = \frac{1}{|\alpha\beta|}\hat{f}\left(\frac{u}{\alpha}, \frac{v}{\beta}\right)$.
- A convolution of f(x,y) by g(x,y) in the spatial domain is just a multiplication of their frequency domain transforms: $\mathcal{F}(f * g)(u,v) = \hat{f}(u,v) \cdot \hat{g}(u,v)$.

- The derivative of a function in the spatial domain is equivalent to multiplying the frequency domain transform by the frequency value raised to the power of the order of the derivative: $\left(\frac{d}{dx}\right)^m \left(\frac{d}{dy}\right)^n f(x,y) \Longrightarrow (iu)^m (iv)^n \hat{f}(u,v)$. This enables the filtering of an image using different filter kernels g(x, y). This leads to isotropic differentiation, which treats all directions equally (for which the lowest possible order of differentiation is the 2nd-order, which is defined earlier as the Laplacian operator Δ^2) is equivalent simply to multiplying the 2DFT of the image by a paraboloid: $\Delta^2 f(x,y) = \left(\frac{d^2}{d^2x} + \frac{d^2}{d^2y}\right) f(x,y) = -(u^2+v^2)\hat{f}(u,v)$.
- Also, in 2D, a rotation by angle θ in the spatial domain is a rotation by the same angle in the frequency domain: $f(x\cos(\theta) + y\sin(\theta), -x\sin(\theta) + y\cos(\theta)) = \hat{f}(u\cos(\theta) + v\sin(\theta), -u\sin(\theta) + v\cos(\theta))$.
- A change to the log-polar coordinates (r, θ), where $r = \log(\sqrt{(u^2+v^2)})$, and $\theta = \tan^{-1}(\frac{v}{u})$, enables dilation/size change in the original pattern to become simply a translation along the r-coordinate, and any rotation of the original pattern becomes simply a translation along the orthogonal θ-coordinate. This adds size and orientation invariance in the Fourier domain.
- The power spectrum of an image is its 2DFT multiplied by its conjugate complex: $\hat{f}(u,v)\hat{f}^*(u,v)$. The conjugate complex is computed by reversing the sign of i in the imaginary part of the 2DFT: $\hat{f}^*(u,v) = \int_X \int_Y \exp(i(u_k x + v_k y))\ f(x,y)dxdy$. The translation by α, β is invariant in the power spectrum as: $\hat{f}(u,v)\hat{f}^*(u,v) = e^{-i(\alpha u+\beta v)}\hat{f}(u,v)\ e^{i(\alpha u+\beta v)}\hat{f}^*(u,v)$

An energy signal (almost all non-periodic signals) has finite energy and no power, such as the pulse signal has finite energy, though power is zero. A power signal (almost all periodic signals) has infinite and finite energy, such as the sinusoid signal. The power spectrum of a signal in the Fourier domain enables translation- size- and orientation-invariant pattern representations. In computer vision, this enables patterns' representation in a manner that is independent of the patterns' position in the image, their orientation, and their size (i.e. the Poincaré group of transformations). The power spectrum is the Fourier transform of the autocorrelation, which is the convolution of the signal with itself (not flipped), where delay/lag is the parameter that identifies how much a signal overlaps with itself over different lag values over time. In the classical cases $\hat{f}(u,v)$, we have the energy in each frequency mode and its phase information. The translation-invariant power spectrum loses the overall phase of the signal, while the relative phases of the different components contain important information.

Bispectrum addresses the lost phase problem of the power spectrum while maintaining the translation-invariance property. It is defined over triple correlation by coupling different Fourier components. It has the form $\hat{a}_{3,f}(\mathrm{k}_1, \mathrm{k}_2) = \hat{f}(\mathrm{k}_1) * \hat{f}(\mathrm{k}_2) * \hat{f}(\mathrm{k}_1 + \mathrm{k}_2)$, taking two arguments and is a highly redundant representation of f. It is the Fourier Transform of the

triple correlation: $a_{3,f}(\mathrm{x}_1, \mathrm{x}_2) = \sum_{i=0}^{n-1} f(i - \mathrm{x}_1) * f(i - \mathrm{x}_2) * f(i)$, showing that it is shift-invariant, and translation invariant by t is proven by $\hat{a}_{3,f^t} = (e^{-i2\pi \mathrm{k}_1 x} \hat{f}(\mathrm{k}_1)) * (e^{-i2\pi \mathrm{k}_2 x} \hat{f}(\mathrm{k}_2)) * (e^{-i2\pi(\mathrm{k}_1+\mathrm{k}_2)x} \hat{f}(\mathrm{k}_1 + \mathrm{k}_2) = \hat{a}_{3,f}$. $\left|\hat{f}(k)\right| > 0$ for all k, i,e, it is complete and uniquely determines f, up to translation. It is computed by recurrence from base cases: $\hat{f}(0) = (\hat{a}_{3,f}(0,0))^{1/3}$, $\hat{f}(1) = (\hat{a}_{3,f}(0,1)/\hat{f}(0))^{1/2}$ or $\hat{f}(1) = e^{i\varphi}(\hat{a}_{3,f}(0,1)/\hat{f}(0))^{1/2}$ for any phase factor $\varphi \in [0, 2\pi)$. Then the recurrence is $\hat{f}(k) = \frac{\hat{a}_{3,f}(1,\mathrm{k}-1)}{\left(\hat{f}(k-1)\right)^2}$. Because of possible dividing by zero and hard to compute higher dimensional generalisations using tensors, other methods are proposed, such as the skew spectrum on compact groups reducing computation steps using the inherent symmetries. For complete formulation using group and tensor decomposition and applications to homogenous space, refer to (Risi Kondor, 2008).

The bispectrum serves as a complete source of invariants for homogeneous spaces of compact groups, including such important domains as the sphere S^2, which is helpful for computer vision by providing a compact set of rotation invariants that improves discrimination and detects bilateral reflection symmetry.

To apply the bispectrum on images to be translate- and rotation-invariant, $ISO^+(2)$ need to be compactified by exploiting a local isomorphism between the action of $ISO^+(2)$ of rigid body motions R^2 and the action of SO(3) on the two-sphere S^2. The author derived the bispectrum on the SO(3) and showed that representing the MNIST digits images dataset using bispectral representation outperforms the baseline representation of the images. This was confirmed by using linear and Gaussian RBF SVMs as 2-class SVMs for all ten class combinations. The preprocessing included rotations by a random angle between 0 and 2π, clipping, and embedding at a random position in a 30 × 30 patch for each of the 1000 original images of 28 × 28 pixels in size, but most of them only occupy a fraction of the image patch. The Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm is used as the iterative method for solving unconstrained nonlinear optimization problems to find the bispectrum images closest to a given bispectrum because the bispectrum consists of a collection of cubic polynomials, and inverting the mapping.

Another application to representation learning is in the Transformer DNN model. The Transformer Deep Learning model was first proposed in NLP applications as an embedding learning approach that implements a unique encoding function that connects a word to its predecessor and successor words in the complete sentence. It was soon applied to various non-textual datasets, such as images. It was concisely introduced in chapter four and will be discussed in chapter six. Here, the context of this chapter requires a mention of data representation learning applications of the Transformer model as introduced in (Merrill and Althoff, 2021) and (Zhang et al., 2022). The Spatial Transformer Network (T-net) is a layer that can be added to CNN models (that has translation invariance by design) to learn a representation that is translation-, scale-, and rotation-invariant (Jaderberg et al., 2016). Their code is published at https://github.com/kevinzakka/spatial-transformer-network.

Furthermore, Point clouds are a set of points identified by their 3D coordinate based on the image acquisition method and might belong to multiple objects based on the view characteristics, such as the view angle, occlusion, light condition and source. Point Cloud analysis is the process of aligning the point and registering it, or in CAD models, use it in surface reconstruction. Four popular DNNs for point cloud analysis are PointNet, PointNet++, SpiderCNN, and DGCNN. Various tensor decomposition methods have been applied to this problem, such as Tucker (Li et al., 2019), Tensor block-wise singular value decomposition (Tamilmathi and Chithra, 2022), and Tensor regression (Yan et al., 2019).

The 3D-Rotation-Equivariant Quaternion Neural Networks (REQNN) proposed in (Shen et al., 2020) identified the conditions in which using quaternion learns a permutation-invariant and rotation-equivariant representations of 3D point cloud analysis. Their code is published in https://github.com/ada-shen/REQNN.

The Rotation Transformation Network (RTN) proposed in (Deng et al., 2021) utilizes an Euler-angle-based rotation discretization manner to learn the pose of input 3D objects and then transforms them into a view-invariant pose by reducing their Rotation Degree of Freedom (RDF) to zero. The authors used the Princeton ModelNet dataset, which is a collection of 3D CAD models for objects used for object classification. Objects are also identified as belonging to SO(0) with zero RDF, i.e. their pose does not change in 3D space, SO(1) with one RDF, i.e. they are on a plane in the 3D space, or SO(3) with three RDF, i.e. they have an arbitrary pose in a centralized 3D space. The authors noticed that T-net performance degrades with the increasing RDF. They proposed the RTN layers added to the existing architectures and identified that RTN+DGCNN has the highest instance and average per-class accuracy. Their code is published in https://github.com/ds0529/RTN.

(Mahadevan, 2008) representation discovery to a stochastic state-space planning problem, such as Markov Decision Processes (MDP), using Reinforcement Learning (RL) framework. Sequential decisions approximate a utility function that is often approximated using pre-defined basis functions. Harmonic analysis was used to synthesise basis functions that were shown to outperform the best manually pre-defined basis. The unknown environment and utility functions were modelled by an agent using the random walk to build a graph with nodes identifying states, and the adjacency is built based on the succession of states temporally or based on achieved rewards in the exploratory phase. Then, the method applies the Laplacian operator on the graph, diagonalising it, and finding the smoothest eigenvectors corresponding to the smallest eigenvalues to approximate a policy representation capturing the underlying manifold of the samples collected. This means the manifold's topology representing a particular control task's state (action) space non-parametrically constructs the new representation basis dynamically using the most concise (non-uniform local density regions). The identified smooth eigenvectors form the columns of the basis function $|S| \times k$ matrix φ. These Laplacian basis functions can be used in conjunction with a standard “black box” parameter estimation method, such as Q-learning or least-squares policy iteration (LSPI), to learn the optimal policy that maximises the action-

value function; the encoding $\varphi(s): S \rightarrow \mathbb{R}^k$ of a state s is computed as the value of the k proto-value functions on that state. Then this learned policy can be added to a list of policies, and the process can be repeated iteratively to learn more policies. More details on the algorithm, its application in various control tasks, and a comparison to pre-defined representation bases are presented (Mahadevan, 2008).

Another application of representation learning presented in (Mahadevan, 2008) was in the 3D object compression. The classic 2D image compression approach JPEG relies on the discrete cosine transform, a type of Fourier analysis on 2D arrays, and JPEG-2000 relies on the wavelet transform. Both approaches do not scale well to 3D object compression, which relies on identifying the 3D vertices in an object and defining their topology graph. Hundreds of megabytes of vertices can be defined in the input unit bases. A highly sparse representation basis can be identified in run-time using harmonic analysis methods presented in this chapter, whether by Fourier analysis or Wavelet analysis. More details about the performance of each approach as compared to the classic approaches are presented in the book. The third application in information extraction and retrieval (IR) from text datasets was presented in the book. The classical approach of latent semantic indexing (LSI) uses SVD, which is a form of Fourier analysis, to construct the term-document matrix. The author showed that employing diffusion wavelets instead reveals multiscale regularities across documents.

The work of (Armenta and Jodoin, 2021) explains how the quiver representation as a directed graph that allows multiple arrows and loops is used in various concepts of neural networks such as fully-connected layers, convolution operations, residual connections, batch normalisation, pooling operations and even randomly wired neural networks. Data are also represented as quiver representation and mapped to a geometrical space called the moduli space.

In previous chapters, various applications of tensor factorisation were discussed. The work of (Yang and Hospedales, 2017) presents a multi-task learning (MTL) representation learning using tensor factorisation (Tucker and TT) as a generalisation of the matrix factorisation (such as PCA) to share knowledge across tasks in fully connected and convolutional DNN layers. They compare their method to Single Task Learning (STL) vs MTL, using user-defined representation vs the learned representation on shallow and deep layer networks. The increased accuracy of learning the representation using tensor factorisation on deep layers is due to the end-to-end training of both the classifier and feature extractor. They published their code in https://github.com/wOOL/DMTRL/blob/master/demo.ipynb.

There is a wealth of contributions in representation learning algorithms. For example, given a network, the representation can be learned using various methods as surveyed by (Zhang et al., 2018). OpenNE is a Python package that implements many network embedding (NE) and network representation learning (NRL), such as DeepWalk, LINE, node2vec, GraRep, TADW, and GCN. Their published code is in https://github.com/thunlp/OpenNE. Similarly, given a graph structure, various methods exist to learn their representation (Chen et al.,

2020). The authors contributed GRLL (Graph Representation Learning Library) as a python package that implements many of these algorithms and evaluates their performance. Their code is published at https://github.com/yunchengwang/graph-representation-learning.

For the latest papers and code on this topic, the Paper-with-code task https://paperswithcode.com/task/representation-learning keeps benchmarks, datasets, and papers contributing to the field.

References

Altmann, S.L., 1986. Rotations, quaternions, and double groups, Oxford science publications. Clarendon Press ; Oxford University Press, Oxford [Oxfordshire] : New York.

Armenta, M.A., Jodoin, P.-M., 2021. The Representation Theory of Neural Networks.

Brigham, E.O., 1988. The fast Fourier transform and its applications, Prentice-Hall signal processing series. Prentice Hall, Englewood Cliffs, N.J.

Chen, F., Wang, Y., Wang, B., Kuo, C.-C.J., 2020. Graph Representation Learning: A Survey. SIP 9. https://doi.org/10.1017/ATSIP.2020.13

Corke, P., 2017. Robotics, Vision and Control, Springer Tracts in Advanced Robotics. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-54413-7

Deng, S., Liu, B., Dong, Q., Hu, Z., 2021. Rotation Transformation Network: Learning View-Invariant Point Cloud for Classification and Segmentation, in: 2021 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. https://doi.org/10.1109/ICME51207.2021.9428265

Dixon, G.M., 2002. Division algebras: octonions, quaternions, complex numbers and the algebraic design of physics, 2. print. ed, Mathematics and its applications <Dordrecht>. Kluwer Academic, Dordrecht.

GAP4, 2022. The GAP Group, GAP -- Groups, Algorithms, and Programming.

Gilmore, R., 2005. Lie groups, Lie algebras, and some of their applications. Dover Publications, Mineola, N.Y.

Haykin, S.S., 2009. Neural networks and learning machines, 3rd ed. ed. Prentice Hall, New York.

Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K., 2016. Spatial Transformer Networks.

Jeevanjee, N., 2011. An introduction to tensors and group theory for physicists. Birkhäuser, New York.

Jeremiah, 2012. Understanding Quaternions. 3D Game Engine Programming. URL https://www.3dgep.com/understanding-quaternions/ (accessed 8.8.22).

Li, J., Zhang, X.-P., Tran, T., 2019. Point Cloud Denoising Based on Tensor Tucker Decomposition, in: 2019 IEEE International Conference on Image Processing (ICIP). Presented at the 2019 IEEE International Conference on Image Processing (ICIP), IEEE, Taipei, Taiwan, pp. 4375–4379. https://doi.org/10.1109/ICIP.2019.8803602

Mahadevan, S., 2008. Representation discovery using harmonic analysis, 1. ed. ed, Synthesis lectures on artificial intelligence and machine learning. Morgan & Claypool, San Rafael, Calif.

Merrill, M.A., Althoff, T., 2021. Transformer-Based Behavioral Representation Learning Enables Transfer Learning for Mobile Sensing in Small Datasets.

Milne, J.S., 2021. Group Theory. Available at www.jmilne.org/math/.

Olguín Díaz, E., 2018. 3D Motion of Rigid Bodies, Studies in Systems, Decision and Control, (SSDC). Springer Berlin Heidelberg, New York, NY.

Risi Kondor, 2008. Group theoretical methods in machine learning (Ph.D. thesis). COLUMBIA UNIVERSITY.

Shen, W., Zhang, B., Huang, S., Wei, Z., Zhang, Q., 2020. 3D-Rotation-Equivariant Quaternion Neural Networks.

Stanković, R.S., Moraga, C., Astola, J., 2005. Fourier analysis on finite groups with applications in signal processing and system design. IEEE Press ; Wiley-Interscience, Piscataway, NJ : Hoboken, N.J.

Tamilmathi, A.C., Chithra, P.L., 2022. Tensor block-wise singular value decomposition for 3D point cloud compression. Multimed Tools Appl. https://doi.org/10.1007/s11042-021-11738-7

Vince, J., 2021. Quaternions for computer graphics, Second edition. ed. Springer, London.

Yan, H., Paynabar, K., Pacella, M., 2019. Structured Point Cloud Data Analysis Via Regularized Tensor Regression for Process Modeling and Optimization. Technometrics 61, 385–395. https://doi.org/10.1080/00401706.2018.1529628

Yang, Y., Hospedales, T., 2017. Deep Multi-task Representation Learning: A Tensor Factorisation Approach. Presented at the International Conference on Learning Representations (ICLR), arXiv.

Zhang, D., Yin, J., Zhu, X., Zhang, C., 2018. Network Representation Learning: A Survey.

Zhang, L., Zheng, S., Lu, J., Zhao, X., Zhu, X., Fu, Y., Xiang, T., Feng, J., 2022. Visual Representation Learning with Transformer: A Sequence-to-Sequence Perspective.