Master's Thesis

Clustering using Quantum Persistent Homology

R.H.A.J. Meijer

4205197

Submitted in partial fulfillment of the requirements for the Master of Science degree in Computing Science.

Written at Utrecht University & TNO The Hague.

dr. E.J. van Leeuwen
dr. C.P. de Campos

N.M.P. Neumann, MSc
S.R. den Breeijen, MSc

October 1st, 2019

Abstract

Persistent homology is a way to analyse the shape of a dataset rather than its semantics. This is done by assigning the dataset holes in different dimensions, borrowing this theory from the mathematical area of topology. The dataset is treated as being sampled from a topological space, which is in turn approximated at different resolutions. The persistent Betti numbers then count the number of holes in these spaces, registering for how long the holes persist as the resolution increases. Computing such features classically may prove costly, with some steps possibly requiring exponential time. In a 2016 paper, a polynomial time quantum algorithm for the approximation of persistent homology is promised. In this thesis it is shown these methods are generally incapable of uniquely determining the persistent Betti numbers in arbitrary dimensions. The algorithm does succeed in the zeroth dimension. After treating the prerequisites, the methods from this paper are explained, the above statements are proven, and finally the quantum algorithm is related to agglomerative hierarchical clustering and spectral clustering.

Contents

Chapter 1

Introduction

1.1 Topics

This introductory chapter is meant to build an intuitive understanding of the separate fields brought together in this thesis, and therefore mostly serves to draw an outline for the following chapters to fill. Breaking down the title of this thesis, it is revealed that the ideas stem from three different fields. First, the algorithms are to be run on a quantum computer, thus evidently drawing from physics. Second, the treated algorithms are used for cluster analysis, hailing from machine learning and statistics. Lastly, pure mathematics is represented by the use of persistent homology, a technique for data analysis that stems from topology. Persistent homology is used in this thesis as the basic principle for quantum clustering. The topics are discussed in order of their corresponding chapter's appearance, and the dependencies between these chapter's subsections are visualised in the flowchart at the end of this introduction. Following any path in this flowchart leads to the terminal nodes that represent sections of the sixth and last theoretical chapter, where all theory culminates in the discussion of quantum algorithms for hierarchical and spectral clustering.

The remainder of this subsection is used for notes on the general structure of this thesis. As a matter of convention, most definitions are segmented from the main body of text, unless it is best seen as an addendum to previous definitions, or a string of related definitions are presented at once. In these cases the terms that are defined are emphasized. On the topic of typography, important terms that are not given strict definition at their place of occurence are emboldened.

Nearly all recurrent and foundational material is collected in the second chapter. Sometimes the choice was made to not include theory finding use only in specific subsections, instead treating it wherever required. Examples are the treatment of homotopies in aid of the theory of simplicial complexes separate from the section on topology, or that of perturbations for the motivation of spectral clustering. This decision is based on didactical considerations, although the disjunctive modular division of material is usually given higher priority.

The three fields this thesis builds on are now briefly introduced.

1.1.1 Quantum Computing

Quantum computers store their information in quantum bits or **qubits**, the quantum analogue to classical bits. These basic units of information differ from their classical counterparts as they are not strictly binary and discrete, but rather continuous. As with bits, a qubit has two elemental values, commonly denoted $|0\rangle$ and $|1\rangle$, but it differs in that it may also reside in an intermediate state called

a **superposition**. In linear algebraic terms this means that the state of a qubit is determined by a linear combination of the elemental values of the qubit. A curiosity is that the coefficients of this combination, and therefore the full state of the qubit, are not directly accessible. A state is read out by way of quantum **measurement**, obtaining one of the elemental values with probability derived from the aforementioned coefficients. Following this measurement the state collapses, leaving the state of the system in a single elemental value.

In essence, qubits are a way of storing information with a richer structure than their classical binary counterparts. Stored information is manipulated by way of matrix multiplication, and because of this algebraic structure an expansive array of possibilities is offered. As a notable example, one of the consequences of the availability of superpositions is the concept of **quantum parallelism**, the possibility of obtaining several results of an operation in superposition. A different useful property is that of **entanglement**. In a system of multiple qubits, it is possible for a state to occur that can not be described by the individual states of the component qubits. The qubits are then said to have become entangled, and this phenomenon can be exploited for information theoretical gain. Superdense coding, used to communicate two classical bits worth of information by sending only one qubit, exploits the possibility of entanglement. Without entanglement this feat would be impossible (by a result called Holevo's theorem [70]).

This model of computation is not only one of advantages, as it bears disadvantages as well. As was mentioned before, the precise state of a quantum system is inaccessible, and upon measurement the state collapses onto an elemental value. Part of the paradigm is therefore dedicated to **interference**, the process of attempting to manipulate the probability distribution governing these results. Furthermore, the states may suffer from **decoherence**, state decay caused by outside influences. This makes quantum computation without taking additional measures prone to noise and instability. Another difficulty is that classical thinking is strongly ingrained. Truly new algorithms are rarely formulated, and fully exploiting the capabilities of a quantum computer has proven hard.

For reasons now apparent, most quantum algorithms are rooted in a few pioneering routines. A famous example of such a routine is the later to be treated Grover's algorithm that can search through an unstructured database in sublinear time. Another is Shor's algorithm that can accomplish integer factorisation in polynomial time, therefore equipping quantum computers to pose a threat towards modern cryptography that grounds itself in the hardness of the factorisation problem.

1.1.2 Cluster Analysis

Cluster analysis is an example of an unsupervised machine learning task wherein the main objective is the identification of homogeneous groups in a dataset. The problem formulation is pinned on the philosophy that a proper clustering should lead to groupings of datapoints that have a high degree of likeliness within each group, but a high dissimilarity between the separate groups. If the clustering induces a partition of the dataset this corresponds to a hard or **complete** clustering, where each datapoint belongs to exactly one cluster. If not, the clustering is said to be **incomplete**. The incomplete clusterings cover all possibilities that do not require strict memberships. If a clustering is incomplete, datapoints may be said to belong to clusters with a certain probability or likelihood, or datapoints may be labeled outliers that are not assigned to any cluster at all. Regardless the type, clustering is most suited to exploratory data analysis and knowledge discovery, finding application in several fields where it is of importance to be able to identify groups.

The algorithms used are based on distinct ideas on what makes a cluster, leading to several different models. Examples are **connectivity based** models that base their clusters on pairwise distances, **centroid based** models that find identifying points representative for the clusters and **density**

based models. The first two models are treated in this thesis through hierarchical clustering and k-Means clustering. The last is not, but the interested reader is encouraged to read about such models in for instance the form of DBSCAN [79].

Prior to this work, multiple research groups have investigated the possibility of setting a quantum computer to the task of clustering, and more generally machine learning. Highly accessible surveys of the proceeds are given in [4] and [80].

1.1.3 Persistent Homology

Summarily, topology is an area in pure mathematics that studies spaces up to specific continuous deformations. This means that the result of a space that is subject to bending and stretching but not tearing or glueing is considered equivalent to the original space from before these operations were performed. Equivalently, two spaces are seen as equal if one can be transformed into the other by way of a continuous deformation. Seeing as such transformations, known as **homeomorphisms**, are clearly not required to preserve distances between elements, the familiar notion of a metric will not suffice to encode and retain nearness relations. One can instead opt to register this information in certain subsets, leading to the definition of a **topology**, and consequently to that of a **topological space**. The topological space thus arises as a generalisation of the metric space and it is worth noting that every metric indeed induces a topology.

Because a general topological space on itself has no canonical distance measures, its geometry is discussed qualitatively instead of quantitatively. Rather than speak of lengths or angles, topologists converse in terms of paths, loops and holes. It is the latter of these three that is of most importance to this thesis; the theory that captures the holes of a space in each dimension is that of **homology**. A key insight that connects topology to machine learning is that one can not only look at a dataset as representing information, but that it can also be analysed through its innate homological information. The first obstacle is that a dataset is not a topological space, but under a set of mild assumptions this can be remedied. In order to do this, inspiration is drawn from the computation of **simplicial homology**, the homology of a simplicial complex.

Recall that simplices can be considered to be generalisations of the triangle, as they are shapes formed by pairwise connected points. Simplices in turn can be joined together following specific rules to form a simplicial complex which can be realised as a topological space. In order to compute the simplicial homology, all cycles comprised of simplices in such a complex are considered, and each closed cycle that does not bound another simplex is considered to be a hole in the complex. Intuition dictates that if there is nothing to bound, there should be a hole. The simplicial homology is a representation of these holes. In general, a topological space can be approximated by a simplicial complex which is referred to as the act of **triangulation**.

A set of real-valued vectors together with some metric is called a **point cloud**. Given a point cloud, one can assume the points are coordinates sampled from an underlying topological space and attempt to triangulate it based on the sample. A common way to do this is to treat every datapoint as a 0-simplex and to add higher dimensional simplices between them, based on a distance parameter. This would raise the question of what parameter to use, which is a difficult one and is moreover case-dependent. Persistent homology answers the question of what scale parameter to use with another: "why not all of them?" Instead of looking at one fixed scale parameter, persistent homology traces the lifespan of the detected holes as the scale increases. At each timestep the complex is expanded, thereby closing and creating new holes, which means the scale is a factor carrying important information. The associated construction incorporating the homology of all these complexes is called **persistent homology**, as it shows how the holes persist over time. The holes

that live the longest are most likely to be true features of the underlying space, while short-lived holes are likely to be noise in the dataset. Hence we can not only compute the supposed homology, but also obtain reasonable assurance on how accurate an approximation it is.

1.2 Academic Context

As an academic work, this thesis should be read in an academic context. This section should assist in establishing this frame of reference. First, the following literature review is a coarse evaluation of the work related to this thesis. Following this review, a cursory listing of the contributions made in this thesis is given in the second subsection, which will later be revisited in greater detail in the concluding chapter.

1.2.1 Literature Review

Quantum computing and quantum information, both coinciding to a certain extent, are relatively new fields. Although many papers are published on the field's developments, most algorithms and relevant information regarding gate-based quantum computation can be found in [70]. For this reason the review of quantum computation is confined to this modest paragraph.

Although the concept of persistent homology predates [92], this paper both introduces the elements of persistent homology to the reader and proposes algorithms for its computation over both fields and principal ideal domains. The latter are abstract algebraic structures, the use of which grants generality which is often not needed in machine learning applications where the coefficient domain is fixed.

Persistent homology seems suited to clustering in particular because the homology in the zeroth dimension measures the connectivity of a topological space. At the time of writing and to our knowledge, four papers have been published on the possible applications of persistent homology to clustering tasks. These four papers are treated below in no particular order.

In [49], the authors have presented a clustering algorithm using persistent homology. The general idea is to construct neighborhoods around datapoints and to look at their measured overlap based on topological properties at a varying timescale. These overlaps are then used to filter and remove points from each neighborhood. A graph from which the clusters can be extracted is constructed on the resulting sets.

Using examples from time series analysis, the effectiveness of persistent homology for clustering is demonstrated in [74]. Several statistics based on the persistent homology of the data are calculated and the clustering takes place solely in the space of these observations. The found clusters translate back to a clustering of the original dataset. It is worth noting, however, that the paper does not thoroughly analyse the computational complexity.

Of a surveying nature itself, several common clustering algorithms are compared in [90], with the addition of a clustering algorithm rooted in persistent homology. This algorithms identifies clusters by assigning statistical significance to found connected components using the zeroth dimensional homology, using a treshold on the persistence.

Finally, [18] discusses a hill climbing based algorithm where the merging of clusters is guided by persistent homology. The main objective of this paper is to use persistent homology to aid in choosing the clustering parameters, and the theory is formulated in the generality of Riemannian manifolds.

A supposed quantum algorithm for the computation of persistent homology is outlined in [59]. Instead of deterministically constructing simplicial complexes, this algorithm uses a quantum search algorithm to construct an approximation to the complex encoded in a quantum state. This both reduces the computational complexity and compresses the information. Following this, the Betti numbers are estimated using an eigenvalue estimation routine [2], and it is claimed this leads to the computation of persistent homology. This claim is discussed in Section 5.3.

A notable paper [1] that warns against the hidden or overlooked caveats of many quantum algorithms mentions the paper from the above paragraph as a positive example of quantum algorithmic design. Furthermore, there has been a proof of concept realisation that computes the Betti numbers on a small scale photonic quantum processor. This was reported on in [48].

For further context, several quantum counterparts to classical clustering algorithms have been developed [5, 89]. These adaptations depend on replacing certain subroutines of the common algorithms by quantum routines and set the landscape for quantum clustering tasks.

1.2.2 Summary of Contributions

First and foremost, this thesis combines the three fields of quantum computing, machine learning and topology. This conjoinment is cast into one unifying language. The use of this language in particular is to bring seemingly unrelated theoretical results together, and showing they exist in the intersections of their fields. The theoretical gaps left by preceding works are filled, with the result serving as a foundation for the main contributions of this thesis. Reasoning from these foundations it is shown that in dimensions higher than zero, the algorithm in [59] might fail to approximate the persistent Betti numbers. It is also proven the algorithm succeeds in the zeroth dimension. The computation of the zeroth dimensional persistent Betti numbers is related to the process of hierarchical clustering, and the connection between the theory in [59] and spectral clustering is briefly explored.


1.
Introduction
2.1.
Algebra
2.2.
Analysis
2.3.
Topology
4.1.
Hierarchical
Clustering
4.2.
Prototype
Based
Methods
4.3.
Spectral
Clustering
3.1.
Quantum
Mechanics
5.1.
Common
Elements
3.2.
Quantum
Circuits
5.2.
Persistent
Homology
3.3.
Famous
Algorithms
5.3.
Quantum
Persistent
Homology
6.1.
Simplex
Membership
Oracles
6.2.
Quantum
Hierarchical
Clustering
6.3.
Quantum
Spectral
Clustering

Chapter 2

Mathematical Preliminaries

2.1 Algebra

Although linear algebra usually precedes abstract algebra in any mathematically oriented curriculum, this section reverses this order and assumes knowledge from a first course in linear algebra. These prerequisites will be used for minor motivation in the first subsection on abstract algebra, and the concepts from this subsection will be useful to extend the presumed known theory from linear algebra. A good resource to revisit linear algebra is [8], and all abstract algebraic theory discussed in this subsection is but a drop of water in the ocean that is [30].

2.1.1 Abstract Algebra

Abstract algebra is one of the cornerstones of modern mathematics and envelopes the study of abstract algebraic structures. A student's first exposure to abstract algebra is usually in the form of group theory and this subsection dares not deviate from this tradition.

Definition 2.1.1. (Group). A group is a set G together with an operation $\cdot : G \times G \to G$ satisfying the following three axioms for any $x, y, z \in G$:

1. The operation is associative, meaning $x \cdot (y \cdot z) = (x \cdot y) \cdot z$;
2. The set contains an identity element $e \in G$ with the property that $e \cdot x = x \cdot e = x$;
3. Each element x in the set has an inverse $x^{-1} \in G$ so that $x^{-1} \cdot x = x \cdot x^{-1} = e$.

If the group's operation is commutative the group is called an *abelian group*, and if no confusion may arise the composition $x \cdot y$ of two elements may be written as xy. Groups can be thought of as modeling symmetry and can arise in several ways. One example is found in free groups, reminiscent of spanned vector spaces. In linear algebra, the span of a basis contains all linear combinations of the basis elements. The equivalent notion in a free group is a generating set, and the free group is the span of this set in the sense that it contains all products including inverse factors.

Definition 2.1.2. (Free Group). The free group $F(B)$ on a generating set B is the group with as elements all possible finite reduced combinations of the elements in B and their inverses. The cardinality of B is called the rank of $F(B)$.

Vector spaces and groups are in fact directly related, as the vector space axioms ensure that every vector space is in particular an abelian group. Therefore, maps between groups exist, and they carry the same name as their vector space counterparts.

Definition 2.1.3. (Group Homomorphism). A map $f : G \to H$ between groups is called a homomorphism if for all $x, y \in G$: $f(xy) = f(x)f(y)$.

As in linear algebra, if the homomorphism is bijective, it is called an isomorphism.

Definition 2.1.4. (Group Isomorphism). A homomorphism between two groups is called an isomorphism if it is bijective, and two groups G_1 and G_2 are called isomorphic if an isomorphism between them exists. This is denoted $G_1 \cong G_2$.

Group homomorphisms are closely related to subgroups of groups, to be defined momentarily. The reader is invited to explore these connections themselves.

Definition 2.1.5. (Subgroup). A subset H of a group G is called a subgroup, denoted $H < G$, if combined with the restriction of the group operation to H it itself forms a group.

Subgroups will be important to the theory of persistent homology which requires quotient groups in particular for its construction. Quotient groups themselves require normality of the subgroup by which the quotient is taken.

Definition 2.1.6. (Normal Subgroup). A subgroup N of G is called a normal subgroup of G, denoted $N \triangleleft G$, if it is invariant under conjugation. Therefore, N is normal precisely if for all $g \in G$:

$$gNg^{-1} = \left\{gng^{-1} \mid n \in N\right\} \subseteq N. \tag{2.1}$$

The requirement for normality in the definition of a quotient group is related to the group operation that is induced on the resulting quotient.

Definition 2.1.7. (Quotient Group). If G is a group and $N \triangleleft G$ is a normal subgroup of G, the group:

$$G/N = \{gN \mid g \in G\} \text{ with operation } (xN)(yN) = (xy)N \text{ for } x, y \in G$$

is called the quotient group of G by N.

Quotients can be used to construct an abelian group based on a non-abelian one by quotienting out a specific subgroup. This is called the abelianisation of the original group, and the following definition is a prelude to a precise definition.

Definition 2.1.8. (Commutator Subgroup). An element of the form $xyx^{-1}y^{-1}$ for $x, y \in G$ is called a commutator of G. The group $[G, G]$ generated by the commutators of G is called the commutator subgroup of G.

The commutators of an abelian group are per definition all reducible to the identity element so that the commutator subgroup of any abelian group is the trivial group consisting of only the identity element. It can be shown that the commutator subgroup is normal, meaning the following definition is legal. The definition reflects that the abelianisation of an already abelian group should be the group itself.

Definition 2.1.9. (Abelianisation). The abelianisation of a group G is the abelian group $G/[G, G]$.

These constructions lead to the final required construction from group theory.

Definition 2.1.10. (Free Abelian Group). The free abelian group $F'(B)$ on a generating set B is the abelianisation of the free group $F(B)$ on B.

The free abelian group is therefore not just a free group that is abelian, but it is the abelianisation of a freely generated group.

Moving on in abstract algebra, concepts such as rings, modules and fields are met. This subsection will only define fields, which will be used to extend vector spaces past the familiar definition over real or complex scalars.

Definition 2.1.11. (Field). A field is a set $\mathbb{F}$ together with two operations $+ : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ (known as addition) and $\cdot : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ (known as multiplication) so that for every $x, y, z \in \mathbb{F}$:

1. Both operations are associative, meaning $x + (y + z) = (x + y) + z$ and $x \cdot (y \cdot z) = (x \cdot y) \cdot z$;
2. The operations are commutative: $x + y = y + x$ and $x \cdot y = y \cdot x$;
3. The set $\mathbb{F}$ admits distinct identity elements $0_\mathbb{F}$ and $1_\mathbb{F}$ so that $0_\mathbb{F} + x = x$ and $1_\mathbb{F} \cdot x = x$;
4. There exists an additive inverse $-x \in \mathbb{F}$ so that $x + (-x) = x - x = 0_\mathbb{F}$;
5. If $x \neq 0_\mathbb{F}$, there exists a multiplicative inverse $x^{-1} \in \mathbb{F}$ so that $x \cdot x^{-1} = 1_\mathbb{F}$;
6. Multiplication is distributive over addition: $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$.

The *characteristic* $\mathrm{ch}(\mathbb{F})$ of a field $\mathbb{F}$ is defined as the smallest natural number n so that $\Sigma_{i=1}^{n} 1_\mathbb{F} = 0_\mathbb{F}$, or 0 if no such n exists. It is essential to remark that the congruence classes of the integers modulo p for a prime number p form a finite field of characteristic p with the usual addition and multiplication operations:

$$\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0_p, 1_p, ..., (p-1)_p\}. \tag{2.2}$$

This field, among others, will be used to enrich the commonly known theory of linear algebra in the following subsection.

2.1.2 Linear Algebra

Recall that the axioms defining vector spaces make use of scalars. In elementary definitions these scalars are often taken from either $\mathbb{R}$ or $\mathbb{C}$, and it can easily be confirmed that both these objects are fields. In more generality, when the scalars in the axioms are taken from an arbitrary field $\mathbb{F}$, the vector spaces they describe are said to be vector spaces over the field $\mathbb{F}$. Linear maps on these spaces will often be referred to as *linear operators* from this subsection onwards, defined as linear maps given irrespective of basis. In finite dimensions these operators can be given matrix representations with respect to a fixed basis, and the terms operator and matrix will be used interchangeably in the unambiguous presence of such a basis.

Having established this, the underlying fields are immediately used in the next definition.

Definition 2.1.12. (Linear Functional). A linear functional on a vector space is a linear map from the space into its field of scalars.

In order to stimulate the imagination it is remarked that a definite Riemann integral is a linear functional. Linear functionals however need not be complicated: it is claimed without proof that if the vector space is $\mathbb{C}^n$ with its elements represented by column vectors, the linear functionals on the space are precisely the conjugate transposes of the column vectors acting as matrices. In general these functionals are called the *duals* of the vectors and they themselves form a vector space.

Definition 2.1.13. (Dual Space). The dual space V^* of a vector space V is the vector space consisting of all linear functionals on V.

Vector spaces are often combined with their dual spaces in tensor products, leading to interesting correspondences, represented by for instance an evaluation map that applies duals to vectors.

Definition 2.1.14. (Tensor Product). Let V and W be two vector spaces with bases $\{v_1, v_2, ..., v_n\}$ and $\{w_1, w_2, ..., w_m\}$ respectively. The tensor product $V \otimes W$ of V and W is the nm-dimensional vector space generated by the basis $\{v_i \otimes w_j\}_{1 \leq i \leq n, 1 \leq j \leq m}$ of abstract elements.

Objects of a tensor space, and therefore by circular reasoning the basis elements $v_i \otimes w_j$, are called tensors. It is easy to be misled into thinking every tensor can be written in the form $v \otimes w$ for two elements v and w from their respective component spaces. This is not true, as the counterexample $v_1 \otimes w_1 + v_2 \otimes w_2$ illustrates. Further properties are that tensor products are multilinear and the tensor operation is distributive over both components. If both components are inner product spaces the tensor product inherits a well-defined inner product of itself.

The linear operators acting on a tensor product are tensor products of the operators acting on the components. If A and B are two fixed matrices, the map $A \otimes B$ is given a matrix representations in the basis of the tensor space by the *Kronecker product* with submatrices:

$$[A \otimes B]_{ij} = a_{ij} B. \tag{2.3}$$

This product retains and combines several aspects of the component matrices which are mostly verified through direct computation when required.

For the following paragraphs, let V be a finite dimensional space with inner product $\langle \cdot, \cdot \rangle$. If A is an operator on V, there exists a unique operator $A^\dagger$ so that for all vectors $v, w \in V$:

$$\langle v, Aw \rangle = \langle A^\dagger v, w \rangle \tag{2.4}$$

and this operator is called the *adjoint* or *Hermitian conjugate* of A. Note that this definition is easily generalised to the adjoint of an operator between two finite dimensional inner product spaces. On realised matrices, adjoints are the conjugate transposes of their matrices and they have several useful properties which can be verified whenever they are required later in this work. If the operator is its own adjoint, meaning $A = A^\dagger$, the operator is called *Hermitian* or *self-adjoint*. Projectors that map vectors onto subspaces are important examples of Hermitian operators.

More specialised is the definition of *normal* operators, operators which commute with their adjoint. It follows directly from this definition that every Hermitian operator is normal in particular. Specialising even further, an operator is called *unitary* if its adjoint is also its inverse. The set of all complex unitary $n \times n$ matrices in the standard basis forms a group $\mathrm{U}(n)$ with the operation of matrix multiplication. As a glimpse into the depths of abstract algebra, this group can be equipped with a differentiable structure, turning it into a continuous object called a Lie group.

The final definitions of this subsection involving inner products are those of positive and positive definite operators. An Hermitian operator A on V is called a *positive definite* operator if for all non-zero elements v of the space, $\langle v, Av \rangle > 0$. If equality is allowed to occur, meaning $\langle v, Av \rangle \geq 0$, the operator is simply called *positive*. From this point on the assumptions on V are removed.

Mathematicians familiar with the theory discussed in this subsection will have already noticed the use of the dagger to denote the adjoint, notation that is common in physics but differs from the uses in mathematics. Another notational convention specific to quantum mechanics, and therefore physics, is the use of Dirac's *bra-ket* notation in which a vector is written $|v\rangle$ and referred to as a *ket*, while its dual $\langle v|$ is called a *bra*. This notation is convenient as it offers a fairly natural way to write and manipulate vectors in inner product spaces, for instance demonstrated by writing the

standard complex inner product $\langle v, w\rangle$ as a *bra-ket* $\langle v|w\rangle$. The outer product too can be cast in this form. If $|v\rangle, |w\rangle$ are both elements of inner product spaces V and W respectively, their outer product is defined by the operator that acts as:

$$|w\rangle\langle v| : V \to W,\ (|w\rangle\langle v|)\,|x\rangle = \langle v|x\rangle\,|w\rangle\,. \tag{2.5}$$

As an example, given an orthonormal basis of elements $|i\rangle$ for a vector space, a projection on the subspace spanned by $|j\rangle$ is given by the outer product $|j\rangle\langle j|$. These projections, and outer products in general, can be combined linearly. Moreover, every linear operator can be represented as a sum of outer products with leading coefficients. This immediately ties in to the intuition for the following important theorem [8].

Theorem 2.1.15. *(Spectral Decomposition). A matrix acting on a vector space V can be diagonalised with respect to an orthonormal basis of V if and only if the matrix is normal.*

If A can be diagonalised as:

$$A = \sum_i \lambda_i\,|i\rangle\langle i| \tag{2.6}$$

this representation is called its *spectral decomposition*. The spectral decomposition theorem offers assurance that this is possible if A is normal. As the use of the letter λ might have suggested, this decomposition is comprised of eigenvalues and eigenvectors. This naming convention stems from the spectrum, a generalisation of the set of eigenvalues of an operator as used in functional analysis.

Assuming the underlying field is complex and f is a complex function, f can be extended to a function on the class of normal matrices through spectral decompositions. Assuming A is normal and diagonalised as was done previously, the image of A under f is defined as:

$$f(A) = \sum_i f(\lambda_i)\,|i\rangle\langle i|\,. \tag{2.7}$$

This image is uniquely defined and this extension can be used for a variety of functions.

As a final note on general (not necessarily linear) algebra, the structures in this section can be further abstracted using the language of category theory. Although this claim may be disputed, category theory carries an algebraic flavour. A complete rigorous understanding of persistent homology would require a basic understanding of rings, modules and categories. Interested readers are referred to [30] and [56] to acquire this knowledge.

2.2 Analysis

This section introduces a selection of the basic ideas of mathematical analysis, the area of mathematics that studies theories relating to functions, precision, and limits in spaces with suitable structures. Metric spaces are important to data analysis and will provide an intuitive basis for topologies whereas Hilbert spaces form the domain for all quantum mechanical systems. As such, this section is highly important. Additional resources for these topics are [77, 86] and [55].

2.2.1 Metric Spaces

Metric spaces are those spaces that are equipped with a metric which holds information about the distance between any two elements of the space. The definitions in this subsection are essential to the construction of the theory in this thesis, although they will only be present implicitly once the

foundations have been established. The natural course of action at the start of the study of metric spaces is defining the metric.

Definition 2.2.1. (Metric). Let V be a set. A metric on V is a map $d : V \times V \to \mathbb{R}$ that meets the following three requirements for all $x, y, z \in V$:

1. $d(x, y) \geq 0$, and $d(x, y) = 0$ if and only if $x = y$;
2. $d(x, y) = d(y, x)$;
3. $d(x, y) \leq d(x, z) + d(z, y)$.

A pair (V, d) of a set and a metric on the set is what is known as a *metric space*. The presence of the metric allows for notions of precision and convergence, both most notably present in the definition of the limit of a map.

Definition 2.2.2. (Limit). For $f : (V, d_V) \to (W, d_W)$ a map between metric spaces and v, w elements from V and W respectively, f is said to have the limit w as x approaches v if for every real value $\epsilon > 0$ there exists a $\delta > 0$ so that if x is taken from the domain of f with $d_V(x, v) < \delta$, then $d_W(f(x), w) < \epsilon$.

If the value w meeting these requirements exists the limit is denoted $\lim_{x \to v} f(x) = w$. The thought underlying this definition is that if you can guarantee that for every precision ϵ a corresponding precision δ can be chosen so that if x is δ-close to v in the domain, $f(x)$ will be ϵ-close to w in the codomain, then when taking the limit the function will take the value w.

Limits are not limited to functions, but can also be taken of sequences. This is because a sequence can be defined as a function on the natural numbers equipped with the metric $d(x, y) = |x - y|$, although limits of sequences are subject to some specifics.

Definition 2.2.3. (Sequence). A sequence of elements in a set V is a map $\mathbb{N} \to V$, $n \mapsto a_n$. The usual notation for this sequence is $(a_n)_{n \in \mathbb{N}}$.

Assuming (V, d) is a metric space, the sequence $(a_n)_{n \in \mathbb{N}}$ is said to converge to an element a in V if for every $\epsilon > 0$ there exists a natural number N such that $n > N$ implies $d(a_n, a) < \epsilon$. This is denoted $\lim_{n \to \infty} a_n = a$. If no such a exists the sequence is called divergent.

A specific class of sequences, so called Cauchy sequences, are of direct importance to the following subsection. It can be shown that every convergent sequence must belong to this class.

Definition 2.2.4. (Cauchy Sequence). If $(a_n)_{n \in \mathbb{N}}$ is a sequence in a metric space (V, d), then it is called a Cauchy sequence if for every $\epsilon > 0$ there exists an $N \in \mathbb{N}$ so that for every $m, n \in \mathbb{N}$:

$$m, n \geq N \implies d(a_m, a_n) < \epsilon. \tag{2.8}$$

The converse however need not be true, and this class of sequences leads to a class of spaces.

Definition 2.2.5. (Complete Space). A metric space is called complete if every Cauchy sequence in it converges to an element in the space.

Before leaving the topic of general metric spaces the discussion of two important objects, open and closed balls, is broached. Open and closed sets are important to the study of metric spaces and they should be familiar to the reader in the form of open and closed intervals on the real line. In fact, they were already used throughout this subsection without having properly introduced them. Because more generality is not necessitated these balls are defined without further explaining their open and closed prefixes, leaving the meaning to the imagination of the reader.

Definition 2.2.6. (Metric Balls). A closed ball of radius ϵ centered around a point v of a metric space (V, d) is defined as:

$$\bar{B}_d(v, \epsilon) = \{x \in V \mid d(v, x) \leq \epsilon\}. \tag{2.9}$$

If the inequality in the predicate is taken to be strict, the ball, then denoted $B_d(v, \epsilon)$, is called open.

Following this small detour, the treatment of specialised complete spaces can be started.

2.2.2 Hilbert Spaces

Hilbert spaces are special instances of more general spaces named Banach spaces. For completeness this subsection is opened with the definition of a Banach space, showcasing the first example of an induced metric.

Definition 2.2.7. (Banach Space). A Banach space is a vector space over either $\mathbb{R}$ or $\mathbb{C}$ equipped with a norm $\|\cdot\|$ that is complete with respect to the metric $d(x, y) = \|x - y\|$ induced by the norm.

Hilbert spaces are a strict subset of Banach spaces, and because they are inner product spaces rather than normed spaces, they allow for concepts like orthonormality and unitarity.

Definition 2.2.8. (Hilbert Space). A Hilbert space is an inner product space over either $\mathbb{R}$ or $\mathbb{C}$ that is complete with respect to the metric $d(x, y) = \sqrt{\langle x - y, x - y\rangle}$ induced by its inner product.

Hilbert spaces are often required because they can, among other reasons, offer certain guarantees on convergence, and they are indispensable to functional analysis. Although Hilbert spaces may be infinite dimensional quantum computing does not require this generality, and for this reason the results in this subsection are formulated for finite dimensional Hilbert spaces to avoid further complexity. On the contrary, the following definitions and proposition will greatly reduce the complexity when moving from general quantum mechanics to quantum computing.

Definition 2.2.9. (Isometry). A linear map f between two Hilbert spaces H, K with inner products $\langle\cdot,\cdot\rangle_H$ and $\langle\cdot,\cdot\rangle_K$ is called an isometry if and only if for all $x, y \in H$: $\langle x, y\rangle_H = \langle f(x), f(y)\rangle_K$.

These maps are thus inner product preserving and by extension, as the name implies as well, distance preserving. Isometries can in turn be used to define isomorphisms between Hilbert spaces.

Definition 2.2.10. (Hilbert Space Isomorphism). Let $f : H \to K$ be a map between Hilbert spaces. Then f is called a Hilbert space isomorphism if f is a surjective isometry. If such a map exists the Hilbert spaces H and K are called isomorphic, written $H \cong K$.

A possibly surprising result from functional analysis is that finite dimensional Hilbert spaces can be uniquely characterised by their dimension [55].

Proposition 2.2.11. *(Characterisation of Finite Dimensional Hilbert Spaces). If H and K are both real or complex finite dimensional Hilbert spaces, then $H \cong K$ if and only if they are of equal dimension.*

This proposition is given without proof and is derived from a more general theorem that characterises separable spaces, spaces whose involved definition will be avoided in this work. More information can be found in any text on functional analysis.

2.2.3 Differential Equations

The purpose of this subsection is not to delve into the theory of differential equations, but merely to remind the reader of the existence of, or introduce the reader to, the concept of differential equations. A differential equation containing a univariate function is called an ordinary differential equation and is defined as follows.

Definition 2.2.12. (Ordinary Differential Equation). An n-th order ordinary differential equation for an unknown but at least n times differentiable function $f(x)$ is defined by a function g so that:

$$g\left(x, f, \frac{\mathrm{d}f}{\mathrm{d}x}, \frac{\mathrm{d}^2 f}{\mathrm{d}x^2}, ..., \frac{\mathrm{d}^n f}{\mathrm{d}x^n}\right) = 0. \tag{2.10}$$

A solution of the above differential equation is a function f that satisfies the equation on at least an open ball. The differential equation relates the function to its derivatives and therefore describes a dynamical phenomenon where a function can behave differently in a point depending on its rates of change at that point. Such equations are not restricted to functions of one variable.

Definition 2.2.13. (Partial Differential Equation). A n-th order partial differential equation for an unknown function $f(x_1, x_2, ..., x_k)$ that admits mixed partial derivatives up to at least the n-th order is given by a function g:

$$g\left(x_1, x_2, ..., x_k, f, \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial^2 f}{\partial x_1 x_1}, \frac{\partial^2 f}{\partial x_1 x_2}, ...\right) = 0 \tag{2.11}$$

that may involve all mixed partial derivatives up to order n.

Much is known about the general theory of ordinary differential equations, including good guarantees on the existence and uniqueness of solutions by the Picard-Lindelöf theorem [87]. Partial differential equations do not have this luxury and they are often studied in the context of so-called weak solutions. Good texts on ordinary differential equations and partial differential equations are [87] and [32] respectively.

2.3 Topology

Building on the intuition from real analysis, this section's meaning is to develop the theory of simplicial homology as quickly as possible. This construction requires the treatment of point-set topology, which is the most elementary study of topology in the set-theoretic domain and naturally leads to algebraic topology when the topic shifts to topological invariants. For more elaborate treatments the reader is refered to [68] for point-set topology, [13,45] for algebraic topology and [37] for the younger field of applied topology.

2.3.1 Point-Set Topology

Metric spaces are rigid in the sense that, intuitively, relocating a single point of the space destroys the space's structure. Since the distances between points have changed disproportionally, so too has the metric and the resulting space can no longer be considered equivalent to the original space. This does however not mean that the basic classifying qualities have changed, as the resulting space for instance might still exhibit the same number of holes. A topology is a structure more general than a metric that makes such considerations precise. More accurately, it encodes the relations of a space's elements in a way that is insensitive to transformations that bend or stretch the space.

Definition 2.3.1. (Topology). A topology on a set X is itself a set $\mathcal{T} \subseteq \mathcal{P}(X)$ satisfying the following three axioms:

1. The sets $\varnothing$ and X are elements of $\mathcal{T}$;
2. The intersection of two sets in $\mathcal{T}$ is an element of $\mathcal{T}$;
3. Any union of sets taken from $\mathcal{T}$ is an element of $\mathcal{T}$.

A pair $(X, \mathcal{T})$ of a set X and a topology $\mathcal{T}$ on X is called a *topological space*. The elements of the topology are called the *open sets* of the space, whereas their relative set complements in X are the *closed sets*. Note that because topologies are generalisations of metrics, every metric induces a topology. In general however, there is no canonical topology for an arbitrary set.

Maps that conserve topological properties are required to behave predictably. Before being able to define maps that give topological equivalences, it needs to be specified what it means for a map to be predictable. As in calculus and analysis this is done by way of continuity.

Definition 2.3.2. (Continuity). A map $f : (X, \mathcal{T}_X) \to (Y, \mathcal{T}_Y)$ between topological spaces is called continuous if for all $U \in \mathcal{T}_Y$, $f^{-1}(U) \in \mathcal{T}_X$.

Using the notion of continuity, equivalences between topological spaces can be defined.

Definition 2.3.3. (Homeomorphism). A homeomorphism is a bijective map f between topological spaces such that both f and its inverse f^{-1} are continuous.

If a homeomorphism between two topological spaces exists the spaces are called *homeomorphic* and the spaces will be indistinguishable based on their *invariants*, the qualitative properties of their topologies. Non-algebraic examples of these invariants are not within the scope of this text and will therefore not be treated, although the ideas of connectedness, compactness and separability may sound familiar. The next subsection will see the development of an algebraic invariant. Properties such as these are conserved because a homeomorphism induces a bijection between the topologies, which is the set-theoretical notion of an isomorphism.

2.3.2 Algebraic Topology

One of the most prominent invariants of a topological space is its number of holes. A homeomorphism may bend or stretch a space but by continuity never tear or glue parts of it, meaning it can never create or close holes. This invariant takes the shape of a sequence of groups or vector spaces and is called homology. This subsection will treat the specific theory of simplicial homology, the homology of simplicial complexes, starting from simplices.

Definition 2.3.4. (Simplex). Let $\{v_0, ..., v_k\}$ be an ordered set of affinely independent elements of $\mathbb{R}^n$. The k-simplex σ_k spanned by these elements is defined as:

$$\sigma_k = (v_0, ..., v_k) = \left\{ \sum_{i=0}^{k} \lambda_i v_i \mid \sum_{i=0}^{k} \lambda_i = 1,\ \lambda_i \in [0, 1] \right\}. \tag{2.12}$$

A simplex is therefore defined simply as the convex hull of its vertices. Requiring the vertices to be ordered imposes one of two possible orientations on the simplex, and the convention that two orderings define the same orientation if and only if they differ by an even permutation is maintained. This orientation manifests itself as a positive or negative sign on the simplex.

Simplices are most commonly interpreted as being generalisations of the triangle. Analogously simplices have faces which, along with their sign, are important for the definition of homology.

Definition 2.3.5. (Faces). For $j \leq k$, a j-face of a k-simplex $(v_0, ..., v_k)$ is the j-simplex spanned by a subset of $\{v_0, ..., v_k\}$ with cardinality j.

Faces can be obtained by removing vertices before taking the convex hull. For example, $\sigma_k^{(i)} = (v_0, ..., \hat{v}_i, ..., v_k)$ is the $(k-1)$-simplex spanned by the vertices excluding v_i. The hat indicates that the respective vertex is removed. This can be used to remove any number of vertices.

On the contrary of removal, simplices can be joined together to form larger structures called simplicial complexes. This is done by joining the simplices along their faces.

Definition 2.3.6. (Simplicial Complex). A simplicial complex S is a set of simplices such that:

1. If σ is a simplex in S, every face of σ is a simplex in S;
2. For two simplices σ, η in S, their intersection $\sigma \cap \eta$ is either empty or a face of both σ and η.

The simplicial complex can be realised with a natural topology which is not required to be defined explicitly for the further treatment of homology, and the dimension of this realised simplicial complex as a space is equal to the maximum dimension of its simplices.

Given such a simplicial complex, chains of simplices can be defined, which can be visualised as a partial traversal of the complex following the simplices in this chain. Graphs, being a special instance of a simplicial complex containing only its vertices as 0-simplices and edges as 1-simplices, form a natural illustration. A path or walk in a directed graph is a sequence of edges that connects two vertices in the graph. The simplified idea of homology is then the following: if a walk in a directed planar graph is closed and there exists no smaller closed walk on its vertices, this could be interpreted as having detected a 1-dimensional hole in the graph. The chain is closed but encloses nothing. If this were to be made rigorous and generalised to higher dimensional simplices, the holes in any dimension could be detected. This is precisely what simplicial homology achieves. The first step towards this objective is defining the chains.

Definition 2.3.7. (Chain Group). The k-chain group $\Delta_k(S)$ of a simplicial complex S is the free abelian group generated by the k-simplices of S. An element of this group is called a k-chain.

This definition grants a domain and co-domain for a boundary map that maps a k-simplex σ_k to its boundary $\partial_k(\sigma_k)$, a chain of its $(k-1)$-faces. Formally this boundary is a sum of the oriented faces of σ_k:

$$\partial_k(\sigma_k) = \sum_{i=0}^{k} (-1)^i \, \sigma_k^{(i)}. \tag{2.13}$$

The following definition captures the boundary map, in fact a homomorphism, in its full generality.

Definition 2.3.8. (Boundary Homomorphism). The k-th boundary homomorphism of a simplicial complex S is the map:

$$\partial_k : \Delta_k(S) \to \Delta_{k-1}(S), \; \partial_k \left(\Sigma_\sigma \, c_\sigma \sigma \right) = \Sigma_\sigma \, c_\sigma \partial_k \left(\sigma \right) \tag{2.14}$$

that maps chains of k-simplices to chains of their oriented faces.

It is stated without proof that for any k, $\partial_k \partial_{k+1} = 0$. Under the assumption that 0-simplices have no boundary and therefore $\partial_0 = 0$, the combination of the chain groups and the boundary homomorphisms form a *chain complex*, depicted below.

$$\cdots \xrightarrow{\partial_{k+1}} \Delta_k(S) \xrightarrow{\partial_k} \Delta_{k-1}(S) \xrightarrow{\partial_{k-1}} \cdots \xrightarrow{\partial_1} \Delta_0(S) \xrightarrow{\partial_0} 0$$

Given this chain complex for a simplicial complex, its simplicial homology can be computed.

Definition 2.3.9. (Simplicial Homology). Let S be a simplicial complex with its chain complex given. The k-th simplicial homology group of S is the quotient group:

$$H_k(S) = (\ker \partial_k) \,/\, (\operatorname{im} \partial_{k+1})\,. \tag{2.15}$$

Reiterating the previous discussion in the current context, the homology groups include all closed chains that are not themselves boundaries. Because the simplices are oriented and therefore signed, a chain is mapped to its endpoint minus its starting point, meaning the kernel of ∂_k contains the closed chains. Quotienting out the image of ∂_{k+1} ensures that all boundaries are removed, leaving only those closed chains that are not boundaries. The number of holes are then measured by the Betti numbers.

Definition 2.3.10. (Betti Numbers). If S is a simplicial complex and $H_k(S)$ is finitely generated, its rank β_k is called the k-th Betti number of S.

At this point it is worth noting that the coefficients used in the definition of the chain groups are taken from $\mathbb{Z}$. Alternatively these coefficients can be taken from a field $\mathbb{F}$. If this is done the development of simplicial homology can be recast in the language of vector spaces. Rather than using a generating set for the chain groups a basis is used, quotient spaces are taken instead of quotient groups, and the boundary homomorphisms are given realisations as matrices. The Betti numbers become the dimensions of the resulting vector spaces representing the simplicial homology. A result called the universal coefficient theorem [45] guarantees that the Betti numbers remain unchanged as long as the characteristic of the field is 0.

Chapter 3

Quantum Computing

3.1 Quantum Mechanics

The section at hand, as an amalgation of several sources [41, 43, 70], introduces the elements of quantum mechanics. This theory finds its roots in the irreconcilability of the observation that light exhibits both properties of waves and particles with classical mechanics, which assumes that light should be able to be fully described by either one. This concept is what is known as wave-particle duality. This section however mostly steers away from the physical interpretation of quantum mechanics, offering little more than a short contextual discussion of the common interpretations in the closing subsection, and mostly treats quantum mechanics as a mathematical system based on axioms called the postulates of quantum mechanics.

3.1.1 Postulates of Quantum Mechanics

In essence the theoretical framework of quantum mechanics describes the mathematical laws a general quantum mechanical system should obey, but does not give any information on how these conditions can be met. Therefore the framework is given in terms of postulates. The first postulate forms a suitable example of this dynamic. It speaks of the state space of a system, but does not connect the state space of a particular system to a concrete object.

Postulate 1. (State Space). The state space of an isolated quantum mechanical system is an appropriate Hilbert space $\mathcal{H}$. A pure state of this system is completely described by a state vector $|\psi\rangle$, a unit vector in $\mathcal{H}$. ◇

Physicists refer to normalised linear combinations:

$$\sum_i c_i \, |\psi_i\rangle \tag{3.1}$$

of state vectors as *superpositions*, where an *amplitude* c_i is associated to each state $|\psi_i\rangle$. The possibility of superposition follows from the linearity of the Schrödinger equation which expresses the time evolution of a *closed* system, meaning there is no interaction with any other outside system. In order to describe this evolution, $|\psi\rangle$ is considered as a time-dependent function in the next postulate.

Postulate 2. (Evolution). The continuous time evolution of the state $|\psi\rangle$ of a closed quantum system is given by the Schrödinger equation:

$$i\hbar \frac{\partial \, |\psi\rangle}{\partial t} = H \, |\psi\rangle \, . \tag{3.2}$$

The constants i and $\hbar$ are the imaginary unit and the reduced Planck constant respectively, whereas H is an Hermitian operator known as the Hamiltonian of the system. $\diamond$

Using the theory of differential equations it can be shown that under the assumption of the existence of a solution, discrete time evolution from time t_0 to t_1 corresponds to the application of a unitary operator $U(t_0, t_1)$ to the state.

All of the preceding discussion has been exclusively concerned with closed systems. At some point the system may need to be opened up to measurement, violating the requirement of the system having no outside interactions. Measurement therefore leads to a peculiar behaviour of the system.

Postulate 3. (Measurement). A quantum measurement is described by a collection $\{M_i\}$ of operators on the state space $\mathcal{H}$, with each M_i being called the measurement operator for the possible outcome i. The probability of obtaining the outcome i when measuring the state $|\psi\rangle$ is given by:

$$\mathbb{P}(i) = \langle\psi|M_i^\dagger M_i\,|\psi\rangle \tag{3.3}$$

and the measurement operators satisfy:

$$\sum_i \langle\psi|M_i^\dagger M_i\,|\psi\rangle = \sum_i \mathbb{P}(i) = 1. \tag{3.4}$$

The state of the system directly after having measured the outcome i is given by:

$$\frac{M_i\,|\psi\rangle}{\sqrt{\mathbb{P}(i)}}. \tag{3.5}$$

$\diamond$

The sum condition (3.4) is equivalent to the measurement operators satisfying the much easier to verify *completeness relation*:

$$\sum_i M_i^\dagger M_i\ = I. \tag{3.6}$$

The fourth and final postulate assigns a class of state spaces to a composite quantum system consisting of distinct component systems. Such systems lead to the notion of quantum entanglement, to be defined shortly, which is the idea that when it comes to quantum states the whole may be greater than the sum of its parts.

Postulate 4. (Composite Systems). The state space of the composite quantum system with n ordered component state spaces $\{\mathcal{H}_i\}_{i=1}^n$ is given by their tensor product:

$$\mathcal{H} = \bigotimes_{i=1}^{n} \mathcal{H}_i. \tag{3.7}$$

$\diamond$

As is known from the theory of tensor spaces, $\mathcal{H}$ contains states $|\psi\rangle$ for which no sequence of states $|\psi_i\rangle \in \mathcal{H}_i$ exists so that $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$. If the system resides in such a state it is in entanglement and the state is called an *entangled* state. If on the other hand such a sequence does exist the state is called *separable*.

The four postulates given in this subsection describe the abstract fashion in which a quantum mechanical system may be measured and observed. It is a current point of contention how the theory spanned by these postulates corresponds to the physical world. Some common interpretations are discussed later in this section, after having expanded on notable special cases of quantum measurement in the next subsection.

3.1.2 Special Cases of Measurements

Two useful special cases of measurement can be derived from the general third postulate. The first is the projective measurement, useful in particular because their expectations and variances can be calculated directly.

Definition 3.1.1. (Projective Measurement). A projective measurement is given by an Hermitian operator M, called the observable, with spectral decomposition

$$M = \sum_i \lambda_i P_i \tag{3.8}$$

for P_i the projector onto the eigenspace corresponding to the eigenvalue λ_i. When the state $|\psi\rangle$ is measured the probability of obtaining outcome i is given by

$$\mathbb{P}(i) = \langle\psi|P_i|\psi\rangle \tag{3.9}$$

and the state of the system immediately after this measurement is

$$\frac{P_i\,|\psi\rangle}{\sqrt{\mathbb{P}(i)}}. \tag{3.10}$$

Immediately introducing new notation, it can now be verified that the expected value and variance of an observable M for a state $|\psi\rangle$ are given by:

$$\mathbb{E}(M) = \langle M\rangle = \sum_i i\mathbb{P}(i) = \langle\psi|M|\psi\rangle \text{ and } \mathrm{Var}(M) = \langle M^2\rangle - \langle M\rangle^2. \tag{3.11}$$

The use of projective measurement is usually implicit when it is said the measurement is performed in an orthonormal basis of enumerated vectors $|j\rangle$. The projectors are then taken to be $P_i = |i\rangle\langle i|$.

Assume next that a projective measurement has been performed and the result m was obtained. The consideration of a second projective measurement on the resulting state $P_m\,|\psi\rangle\,\mathbb{P}(m)^{-1/2}$ reveals that $\mathbb{P}(m) = 1$ for this measurement, meaning the same result will be observed and the state will remain unchanged as $P_m^2 = P_m$. This is what is known as *repeatability* and illustrates that projective measurement is indeed a special case of the general measurement. A physical measurement of a system may destroy part of it, making a repeat measurement impossible in general.

Alternatively one might be interested only in the probabilities of the outcomes, not in the state the measurement leaves the system in. Under these circumstances it is common to describe the measurement in condensed form as a *Positive Operator-Valued Measure* measurement, or POVM.

Definition 3.1.2. (POVM Measurement). A set $\{E_i\}$ of positive operators is called a POVM measurement if the operators satisfy the completeness relation. Following the third postulate, the probability of measuring i is given by $\mathbb{P}(i) = \langle\psi|E_i|\psi\rangle$. Given a general quantum measurement $\{M_i\}$, the associated POVM measurement is given by the set of positive operators:

$$\left\{M_i^\dagger M_i\right\} \tag{3.12}$$

called the POVM elements of the measurement.

The treatment of measurements is concluded with a short discussion of phase factor invariance. Two states $|\psi\rangle$, $|\varphi\rangle$ are said to be equal up to a *global phase factor* if there exists a real number θ so that $|\varphi\rangle = e^{i\theta}\,|\psi\rangle$. In more generality, two amplitudes are said to differ by a *relative phase factor* if they differ by a factor of this form, and two states are said to differ by a relative phase in some basis if

each of their paired amplitudes differ by the same relative phase.

Adding a global phase factor to a state does not change its measurement statistics, and therefore the statistics for two states that differ only by a global phase factor are identical. One way to see the truth of this statement is by noting that the bra of the ket $e^{i\theta}|\psi\rangle$ introduces a factor $e^{-i\theta}$ which cancels the phase factor:

$$\mathbb{P}(m) = \langle\psi|e^{-i\theta}M_m^\dagger M_m e^{i\theta}|\psi\rangle = \langle\psi|M_m^\dagger M_m|\psi\rangle\,. \tag{3.13}$$

In terms of observations these states can therefore be considered equal, and global phase factors are often omitted. This does not hold for relative phase factors however, as the phases depend on the choice of basis and can therefore not offer the same universal guarantee of invariance.

3.1.3 Interpretations

Up until this point this section has discussed quantum mechanics in abstract terms without regard for physical meaning. This was done for two reasons. The first is to keep the mathematical framework isolated from realisation. The second, most interestingly and in large part motivation for the first, is because there is no universally accepted interpretation. This subsection offers two common possible explanations [41] for the existence of superpositions and the meaning of measurement.

Assume for the next two paragraphs that a system in superposition was measured and a pure state was succesfully identified. The possible interpretations attempt to answer the question of what the dissociation between the superposed state prior to measurement and the concrete state after measurement means physically.

The **Realist interpretation**, most prominently advocated by Einstein, argues that the system was never in superposition and that quantum theory is simply incomplete. Even before measurement the state would have been pure, but the theory itself was uncertain of which of the possible pure states the system was in. According to realists, a completion of the theory would involve the inclusion of additional information on the state that removes indeterminism.

The **Copenhagen interpretation** shifts the blame of uncertainty to the system itself and says the system was not in a fixed pure state, but was compelled to regress to one upon measurement. This supports the possibility of actual physical superposition and at the same time implies the act of measurement truly boasts impact that is unknown.

Although both have been pondered and discussed by physicists extensively, at the time of writing there is only experimentally obtained proof towards the latter interpretation [9, 41].

3.2 Quantum Circuits

Quantum computing is a computational paradigm that depends on the theory of quantum mechanics to obtain speedups over the best known classical algorithms of this time. In order to achieve this the quantum algorithms, expressed as quantum circuits in the gate-based quantum computation scheme, rely on quantum superpositions of the quantum analogues to classical binary values and wave interference to push the information held by a quantum system towards some sought after final result. This result can then be read out through measurement. Although this measurement has a probabilistic nature, the probabilities can be controlled to some extent, and a large part of the design of quantum algorithms concerns itself with precisely this objective.

In this section qubits and quantum circuits are defined and explained in a succint fashion. For a more elaborate treatment of this material the reader is referred to *Quantum Computation and Quantum Information* [70], generally considered to be the bible of the field of quantum computing.

3.2.1 Qubits

The most elementary component of any quantum circuit, and therefore of quantum computing, is the qubit. This quantum counterpart to the classical bit is a mathematical object that allows its base states to be in superposition.

Definition 3.2.1. (Qubit). A qubit corresponds to a quantum mechanical system with a complex state space $\mathcal{H}$ spanned by two orthonormal vectors $|0\rangle$ and $|1\rangle$.

It is known from the theory of functional analysis that because the two-dimensional state space $\mathcal{H}$ admits a countable basis, it can be identified with $\mathbb{C}^2$ through a Hilbert space isomorphism. The custom of using the particular state space $\mathbb{C}^2$ will be upheld from now on, so that every state $|\psi\rangle$ of a qubit can be described as an element of the complex unit sphere:

$$|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle \text{ for } c_0, c_1 \in \mathbb{C} \text{ with } |c_0|^2 + |c_1|^2 = 1. \tag{3.14}$$

Because this norm should be preserved all operations on a qubit should be unitary, and conversely any unitary matrix in U(2) specifies a valid operation on the qubit. This requirement supports the probablistic aspect of the superposition. A projective measurement on a general state with respect to the given basis leads to the following probabilities of observing either state:

$$\mathbb{P}(i) = \langle\psi|i\rangle\langle i|\psi\rangle = \bar{c_i}c_i = |c_i|^2 \text{ for } i \in \{0, 1\}, \tag{3.15}$$

and leaves the system in:

$$\frac{|i\rangle\langle i|\psi\rangle}{\sqrt{\mathbb{P}(i)}} = \frac{c_i}{|c_i|} |i\rangle , \tag{3.16}$$

or simply $|i\rangle$ as $c_i/|c_i|$ has modulus one and global phase factors can safely be ignored.

By the fourth postulate of quantum mechanics, n separate qubit systems can be joined together to form an n-qubit system or register. The state space of this system is the n-tensor product $\mathcal{H}^{\otimes n} \cong \mathbb{C}^{2\otimes n}$ of the individual state spaces. The agreed upon ordered and orthonormal basis for any such state space is the *standard computational basis* in which the k-th element ($0 \leq k \leq 2^n - 1$) is given by:

$$|k\rangle = |k_0 \cdots k_{2^n-1}\rangle := \bigotimes_{j=0}^{2^n-1} |k_j\rangle , \tag{3.17}$$

where the sequence of k_j corresponds to the coefficients in the binary expansion of k.

The operators acting on these systems are tensor products of unitary matrices and a quantum circuit can be defined as a sequence of such operators. These circuits are the subject of the next subsection.

3.2.2 Circuits

Much like classical circuits, quantum circuits consist of wires carrying information and gates acting on that information. The most notable difference is that every gate must act linearly and, due to its required unitarity, reversibly. The following graphic shows the most frequently used single-qubit gates as circuits.

	Circuit	**Matrix**
Hadamard		$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
Pauli X		$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$
Pauli Y		$\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}$
Pauli Z		$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$
Phase		$\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}$
$\pi/8$		$\begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix}$

The Hadamard gate, not to be confused with the Hamiltonian, is important and often used to bring a qubit initialised in the basis state $|0\rangle$ to the uniform superposition of both of its basis states:

$$H|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right). \tag{3.18}$$

Moreover it is easily verified that a second application returns the state to $|0\rangle$, meaning $H^2 = I$. It is a minor exercise to confirm that the Hadamard gate can be written as a linear combination of the Pauli X and Z matrices, gates which swap the amplitudes of a state and flip the sign of the $|1\rangle$ state respectively. The Pauli Y gate represents a rotation of π radians on the unit circle.

Measurements of qubits are part of quantum circuits as well, and if applicable it is customary to both defer measurement to the end of the circuit and assume that wires contain measurements at their endpoints. If these are explicitly included they are represented by a meter symbol:

where the distinction of single wires carrying qubits and double wires carrying classical bits is made.

Once systems of multiple qubits are used, the application of a matrix A to the single k-th qubit in a system of size n is viewed as the application of an n-tensor A_k to the state space of the system with A in its k-th position, and copies of the identity on all others. This naturally extends to the application of different matrices to distinct qubits, and thus a quantum circuit can be defined as follows.

Definition 3.2.2. (Quantum Circuit). A quantum circuit on a system of n qubits is a sequence of n-tensors ended by a measurement unless specified otherwise. Each component matrix is called a gate, and if the qubits in a circuit can be split into groups with different purposes these groups are referred to as registers.

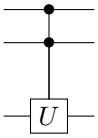
The first elementary example of the use of registers is seen in controlled operations. If an operator acting on some register of qubits is given, the application of this operator may be conditioned on the state of a collection of qubits. This is what is referred to as a controlled operation.

Definition 3.2.3. (Controlled Operation). Let U be a unitary operator on a register of n qubits, and $|\varphi\rangle$ a state of this register. The controlled operation $C_k(U)$ conditioned on k qubits is defined by the unitary operator acting as:

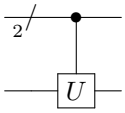
$$C_k(U)\,[\,|x_1 \cdots x_k\rangle \otimes |\varphi\rangle\,] = |x_1 \cdots x_k\rangle \otimes U^{x_1 \cdots x_k}\,|\varphi\rangle \tag{3.19}$$

where the exponent of the operator on the right hand side is the product of the x_i.

The operator is therefore only applied if all of the x_i are set to 1. In a quantum circuit, controlled operations are indicated by connecting wires, as is shown in the circuit below for $n = 1$ and $k = 2$.

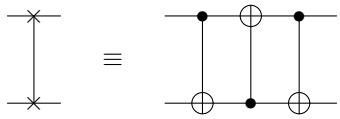


Alternatively, if the wires do not need to be distinguished from eachother in the circuit, the following shorthand representation can be used to bundle them.



This convention is also used to bundle any number of wires outside of controlled operations.

As an important example of controlled operations, the following two circuits show the *swap gate* that swaps the states of two qubits.



Another example is the CNOT gate in itself, used thrice in the above circuit in the right hand representation. It can in fact be shown that the CNOT gate along with the Hadamard, Phase and $\pi/8$ gates form a discrete set of universal operations. This means that any unitary quantum circuit can be approximated to arbitrary accuracy by a quantum circuit consisting only of these universal gates. A proof of this statement is given in [70].

The computational complexity of any quantum circuit, and therefore of any quantum algorithm, is often quantified by counting the total number of component gates. If another quantity is chosen this will explicitly be mentioned in the analysis of a circuit.

3.2.3 Realisation

Several possible physical realisations of a gate-based quantum computer are described in the seventh section of [70]. This short closing subsection discusses in more generality the still ubiquitous "five plus two" requirements for physical implementations introduced by DiVincenzo in [28] and two alternative schemes to gate-based quantum computing following them. For starters, the seven (five plus two) requirements for a physical system to be suitable for quantum computing are the following.

1. It must be scalable and be able to accomodate well-characterised qubits;
2. Said qubits must be initialisable to simple states;
3. Decoherence times of the system must exceed the required gate operation times;
4. The system must admit a universal set of quantum gates;
5. Measurement should be possible per qubit;
6. Both stationary and moving qubits must allow for interconvertation;
7. The transmission of moving qubits from one location to another must be faithful.

The first five are of direct importance to quantum computation. When put together they describe a system that can emulate a Turing machine (1, 2, 4) and retain a state for long enough so that any operation can realistically be applied (3). Moreover, the result of any computation is readable through measurement (5). If a physical system meets these criteria it is a suitable candidate to be used as a quantum computer. The last two requirements are meant for quantum communication.

This section is closed with the brief mention of two alternative schemes for quantum computation. The first of these is **adiabatic quantum computing**, closely related to quantum annealers. This scheme relies on the adiabatic theorem which, in simplified form, states that if the Hamiltonian H_1 of a system moves into another Hamiltonian H_2 gradually enough through a path in a space of matrices, a particle in the j-th eigenstate of H_1 will be carried over into the j-th eigenstate of H_2. This information can then be used to usher a system into a state that represents a solution to an optimisation problem. A reference for this scheme is [20].

The second alternative scheme is **topological quantum computing**. Gate-based quantum computation is prone to errors and therefore requires the use of error correcting codes in practice. The topological quantum computer natively addresses this problem by encoding the information in topological properties rather than in a comparatively rigid quantum state directly. More precisely, topological quantum computation depends on particles called non-abelian anyons that are repeatedly interchanged, which leads to a braiding of their paths through a time dimension. These braids induce unitary operations, meaning a quantum routine can be executed by interchanging the particles in specific ways. References for this scheme include [33] and [76].

3.3 Famous Algorithms

Since the specifics of quantum mechanics deviate significantly from classical intuition, the design of quantum algorithms is considered to be very hard. As of the time of writing there are two major quantum algorithms that provide a significant speedup over the best known classical algorithms. These two algorithms, known as the quantum Fourier transform and Grover's algorithm, both lead to classes of algorithms based on them of which most quantum algorithms that offer significant speedups are a part. As an important example of such an algorithm, this section describes the phase estimation algorithm that is based on the quantum Fourier transform.

3.3.1 The Quantum Fourier Transform

The discrete Fourier transform is a classical data transformation algorithm often used in digital signal processing. Readers unfamiliar with Fourier analysis but who are interested in examples and further theory are refered to sources such as [84]. In brief, the discrete Fourier transform takes in a finite complex sequence $\{x_0, ..., x_{N-1}\}$ and outputs a transformed sequence x' of equal length with

$$x'_k = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} x_l e^{2\pi i \cdot lk/N} \tag{3.20}$$

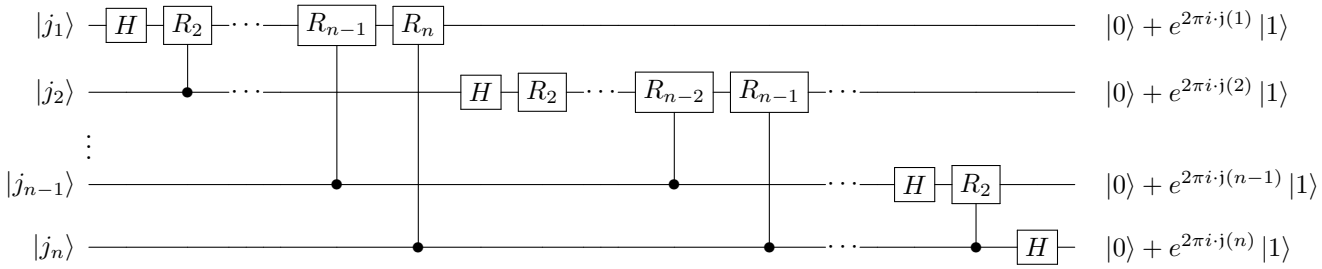
as its elements. The quantum Fourier transform is its quantum counterpart, a unitary mapping the domain of which consists of quantum mechanical amplitudes. Although the domain is shrunk, the advantage is that this transform can be computed in quadratic time whereas the best classical algorithm, the fast Fourier transform, requires exponential time [70]. The mapping applies the discrete Fourier transform to the sequence of amplitudes associated to a quantum state and is therefore given by:

$$\mathrm{QFT}_N : \sum_{j=0}^{N-1} x_j \left|j\right\rangle \mapsto \sum_{j=0}^{N-1} x'_j \left|j\right\rangle \tag{3.21}$$

for an arbitrary input state. Using the general phase shift gates R_m defined as:

$$R_m = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^m} \end{bmatrix} \tag{3.22}$$

and the binary fraction $\mathrm{j}(s) = 0.j_s...j_n$ the following diagram, derived in [70], is an efficient circuit for the quantum Fourier transform acting on a state $|j_1 \cdots j_n\rangle$ given in the standard computational basis.



Note that this representation omits both the swap gates required for the proper ordering of the basis and the normalising factors of the output. Basic tensor algebra then shows that the result of this circuit is equal to the transformation given by Formula 3.21.

The unitarity of the transformation, as well as the computational complexity, can be deduced from this circuit as it is made up wholly of unitary operators. Moreover, the unitarity guarantees the existence of an inverse operator $\mathrm{QFT}_N^\dagger$ given by the adjoint of a matrix representation QFT_N.

In general, the quantum Fourier transform can be used to efficiently solve the hidden subgroup problem (explored in more depth in Chapter 6) of which integer factorisation is a specific instance. Shor's algorithm [81], the famous threat to RSA cryptography, solves the factorisation problem in polynomial time by depending on the transform, thereby making it highly relevant to quantum computing and cryptography.

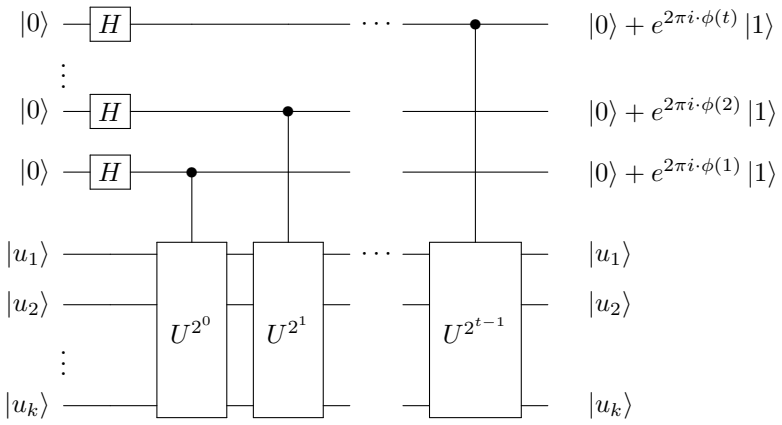
3.3.2 Phase Estimation

Given a unitary operator U with an eigenvector $|u\rangle$ for which the corresponding eigenvalue λ_u is unknown, phase estimation is the process of estimating this eigenvalue. Since U is unitary this eigenvalue must be of the form $e^{2\pi i\varphi}$ for some $\varphi \in [0, 1)$, meaning that λ_u is completely determined by the value of φ. The algorithm outlined in this subsection constructs a quantum state equivalent to the result of a quantum Fourier transform applied to a binary fractional representation of φ and then uses the reversibility of the transform to find the representation of φ.

As in the previous subsection, most algebraic manipulations are delegated to [70], as they are tedious but easily verifiable. The first result of such a manipulation is the *product representation* of the quantum Fourier transform:

$$\mathrm{QFT}_{2^n} : |j_1 \cdots j_n\rangle \mapsto 2^{-n/2} \prod_{l=0}^{n-1} \left(|0\rangle + e^{2\pi i \cdot \mathrm{j}(n-l)} |1\rangle \right), \tag{3.23}$$

which is equal to the statement that the result of the quantum Fourier transform is a product that involves exponentiated partial binary fractions. When φ is represented as a binary fraction $0.\varphi_1 \cdots \varphi_t$ of length t, the following circuit acting on two registers $|0\rangle^{\otimes t}$ and a set of qubits $\otimes_{l=1}^{k} |u_l\rangle$ representing $|u\rangle$ places the first register in the product representation of $\mathrm{QFT}_{2^t} |\varphi_1 \cdots \varphi_t\rangle$.



As before, normalisation factors are not shown. The notation $\phi(s) = 0.\varphi_s ... \varphi_t$ was used analogously to how j was used for j in the previous subsection. This circuit shows how every qubit in the first register is put into uniform superposition by way of Hadamard gates, after which the controlled operator U is applied repeatedly to the second register. After each application the phase factor introduced to $|u\rangle$ can be transfered to a corresponding qubit in the first register by multilinearity, explaining why the second register remains unchanged. Applying the inverse transform to the first register and measuring the result then gives:

$$\mathrm{QFT}_{2^t}^{\dagger} \mathrm{QFT}_{2^t} |\varphi_1 \cdots \varphi_t\rangle = I_{2^t} |\varphi_1 \cdots \varphi_t\rangle = |\varphi_1 \cdots \varphi_t\rangle . \tag{3.24}$$

If φ cannot accurately be represented as a binary fraction of length t, or it needs to be approximated for other reasons, usage of a register of size t leads to an approximation $|\widetilde{\varphi}\rangle$. It is derived in [70] that to approximate φ accurately to t' bits with probability of success at least $1 - \epsilon$, the value of t should be chosen as:

$$t = t' + \lceil \log \left(2 + (2\epsilon)^{-1} \right) \rceil. \tag{3.25}$$

This same reference shows that this algorithm requires $\mathcal{O}(t^2)$ operations excluding the manipulations of the second register which are case-dependent.

3.3.3 Grover's Algorithm

Grover's algorithm [42] is a quantum search algorithm with sublinear time complexity and one of the main draws of quantum computing. It requires only $\mathcal{O}(\sqrt{N})$ operations on an unstructured database of size N, where a classical algorithm would always scale linearly. It depends heavily on the notion of oracles, functions that can identify solutions to problems even if these solutions were not known prior. As such, this subsection will start with explaining the workings of a general quantum oracle, beginning from classical oracles. Assume that a problem is given and the elements of the search space for this problem can be indexed so that every identifier can be stored in n bits.

Definition 3.3.1. (Classical Oracle). Assuming a problem is given, a classical oracle is a function $f : \{0,1\}^n \to \{0,1\}$ that maps an index to 1 if and only if the index corresponds to a solution for the problem at hand.

Classical oracles can be extended to quantum oracles through the help of an ancillary oracle qubit $|q\rangle$. If this qubit is added to a register fit to contain the indices of the items in the search space, the quantum oracle extending the classical oracle f may be defined as a unitary operator O_f acting as:

$$O_f : |x\rangle \otimes |q\rangle \mapsto |x\rangle \otimes |q + f(x) \pmod 2\rangle \tag{3.26}$$

where the addition is taken modulo 2. The state of the oracle qubit after application of the operator then shows whether or not x corresponds to a solution or not. This is precisely the case if the value of the oracle qubit has changed. A clean definition of the quantum oracle follows from choosing to let $|q\rangle$ represent a specific state:

$$|q\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{3.27}$$

The following definition can then be verified algebraically.

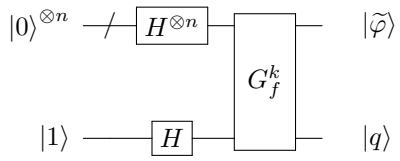
Definition 3.3.2. (Quantum Oracle). Assuming a problem and a classical oracle f for this problem are given, a quantum oracle O_f is defined as the unitary operator that acts as:

$$O_f : |x\rangle \otimes |q\rangle \mapsto (-1)^{f(x)} |x\rangle \otimes |q\rangle . \tag{3.28}$$

Assume for the rest of this paragraph that the search space is a Hilbert space of dimension $N = 2^n$ wherein each basis state corresponds to an item. Moreover, assume that exactly $M \in [1, N]$ of these items are solutions to the search problem, and that $|\psi\rangle$ is the uniform superposition of the aforementioned basis states. The implementation, given an oracle f, is then quite easily explained. Define the Grover diffusion operator G_f as:

$$G_f = (2 |\psi\rangle\langle\psi| - I)O_f. \tag{3.29}$$

For some $k \in \mathbb{N}$, the required value of which will be addressed shortly, the repeated application $G_f^k |\psi\rangle$ then approximates the uniform superposition $|\varphi\rangle$ of the M basis states representing the solutions to the search problem. Schematically this is represented by the following quantum circuit.



The intuition underlying the diffusion operator is that it acts on a two-dimensional subspace of the search space spanned by $|\psi\rangle$ and $|\varphi\rangle$, and with each application the state is rotated closer towards $|\varphi\rangle$. As such, the number of applications k depends on the required precision for the approximation $|\widetilde{\varphi}\rangle$. Besides discussing the geometric interpretation of the algorithm, it is derived in [70] that k scales as $\mathcal{O}(\sqrt{N})$, leading to a complexity of $\mathcal{O}(\sqrt{N})$ in the number of calls to the oracle, and it is shown that this is optimal.

Chapter 4

Clustering

4.1 Hierarchical Clustering

Hierarchical clustering derives its name from the fact that its methods assume there is an hierarchical structure present in the data. This assumption underpins both their strengths and weaknesses. Because of the assumption, identified clusters can easily be represented in visual representations called dendrograms, and the desired final partition of the dataset can be chosen based on the information in this summary. This means that the number of clusters is not required to be chosen beforehand. The downside is that if no hierarchy is present, the methods force it upon the data. An example of why this is generally counterproductive is given later.

4.1.1 The Agglomerative Approach

The agglomerative approach is the most commonly used one, and therefore it is used to introduce hierarchical clustering. As a general description, it is assumed that the clusters admit a hierarchy in the sense that every cluster smaller than the full dataset is part of some larger cluster. At every step these clusters are merged based on intercluster similarity and every merging of the clusters is registered. This partitioning information is kept in a *dendrogram*, an undirected rooted tree together with a vertical axis signifying the number of clusters at a given height in the tree.

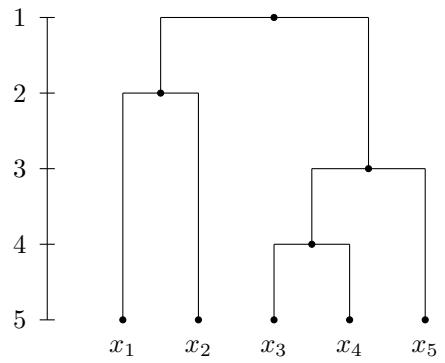


Figure 4.1: An example of a scaled dendrogram, not showing similarities.

Agglomeratively, there are five initial singleton clusters in this example, one for every datapoint. Moving towards the root in the dendrogram, clusters are merged and the lower a merging takes

place, the higher the similarity of the clusters must be. Every node in the tree therefore represents a single cluster. Extracting a clustering from the dendrogram is a straightforward task. Choose the number of clusters k, and draw a horizontal line at height $k-1$ in the dendrogram. The subtrees strictly below this line are then precisely the clusters and the partition of the dataset can be read off by grouping the leaf nodes of each subtree. This description of the dendrogram equates to a sketch of agglomerative hierarchical clustering, leaving only the details regarding the merging to be filled in.

Assume that the dataset X is unlabeled and an arbitrary dissimilarity function $f : X^2 \to \mathbb{R}$ on the paired datapoints is available. In order to merge clusters this dissimilarity needs to be extended to arbitrary subsets of the dataset, and this leads to a *linkage* between clusters. The following graphic specifies several such extensions for two non-empty and disjoint subsets $C_1, C_2 \subset X$ with, if available, centroids c_1^* and c_2^* respectively.

Linkage	**Formula**
Single	$\min_{c_1 \in C_1, c_2 \in C_2} f(c_1, c_2)$
Complete	$\max_{c_1 \in C_1, c_2 \in C_2} f(c_1, c_2)$
Average	$\frac{1}{\lvert C_1 \rvert \lvert C_2 \rvert} \sum_{c_1 \in C_1, c_2 \in C_2} f(c_1, c_2)$
Centroid	$f(c_1^*, c_2^*)$

The reader might wonder why the involved functions are called dissimilarity measures instead of similarity measures. The reason for this is plain to see when a metric is taken as the dissimilarity measure. A higher value is indicative of a higher dissimilarity between two elements, and therefore the metric quantifies the dissimilarity rather than the similarity. This reasoning extends to general dissimilarity measures and therefore justifies the naming convention.

Both the dissimilarity function and linkage determine the resulting dendrogram and are therefore vital to this clustering algorithm. Which of the linkages is suited best to a problem depends on both the structure of the data and the requirements for the sought clusters, meaning the involved reasoning is almost always subjective. By drawing small examples one can clearly see the difference in behaviour for the different linkages.

For completeness the agglomerative hierarchical clustering algorithm is given below. If single linkage is used and this procedure is implemented naively the complexity is $\mathcal{O}(n^3)$ [69], but if the process is optimised the complexity can be reduced to $\mathcal{O}(n^2)$ which is optimal [82].

Algorithm 1 Agglomerative Hierarchical Clustering

Input: A dataset X and a linkage L.
Output: A partition $\{C_i\}_{i=1}^k$ of the dataset.

1: Initialise the dendrogram with $|X|$ nodes representing the singleton clusters;
2: Repeatedly compute the arguments of the minimum linkage L between the current clusters until convergence, merging them in both the partition and the dendrogram;
3: Present the dendrogram to the analyst who chooses the number of clusters k;
4: Let $\{C_i\}_{i=1}^k$ be the collected nodes of the subtrees under level k and return this partition.

The dendrogram used in this description may be replaced by a datastructure that saves the partitions found during the iterations, identified by the minimum linkage. The result corresponds to a unique constructable dendrogram. Using the dendrogram directly is however more intuitive.

The introduction of hierarchical clustering is concluded with the promised example of an unjustly imposed hierarchy taken from [51]. Assume a dataset composed evenly of men and women is given, originating from three different countries. Most likely a partitioning into two clusters would divide the dataset into men and women, but without further information it is equally likely that a division into three clusters would characterise the clusters by nationality. There is however no guarantee that the former clustering resulted from joining two clusters in the latter. It may therefore be possible that some structures are ignored and the results become unrepresentative.

4.1.2 The Divisive Approach

In contrast to joining clusters, clusters may be split as well. As a reconsideration of the example from the previous subsection points out, this principle might even be safer for some datasets. In practice this divisive approach is studied little compared to agglomerative methods because for an arbitrary dataset there are significantly more divisions than agglomerations to consider, leading to a higher number of required computations. Although this problem was addressed to some extent in [62], divisive hierarchical clustering is often still not considered an option.

The adaption to divisions is fairly simple. The dendrograms introduced previously may be read starting from the top, where the first and single cluster is the entirety of the dataset. At every following iteration an allowed set of splits should be considered and as a result one of the clusters should be split in two. The nodes in the dendrogram then represent clusters that are separated instead of merged, and a clustering is extracted from the final dendrogram in identical fashion.

Readers interested in the further specifics of this approach are redirected towards [44].

4.2 Prototype Based Methods

In unsupervised learning, particularly in clustering, a *prototype* is some archetypical point that is treated as being representative for part of the dataset. Finding such points allows for a clustering of the dataset based on proximity to these prototypes, and the prototype based methods therefore aim to identify them. The most popular example is the k-Means algorithm that uses the means of clusters in Euclidean spaces as prototypes, recalibrating at every iteration until the clusters remain unchanged. After introducing this algorithm, the k-Medoids problem that generalises the k-Means problem to arbitrary dissimilarity functions is explained in addition.

4.2.1 k-Means Clustering

The main principles underlying k-Means clustering are the assumptions that the mean of a cluster forms a good prototype, and that recalibration of the clusters based on their means leads to a clustering with small variation within the clusters. In order to motivate the algorithm, the reasoning in this section starts from the ideal but computationally infeasible case.

Let the dataset be given as $X = \{x_1, ..., x_n\} \subseteq \mathbb{R}^m$. Assuming k is fixed and the clusters are given by a partition $\{C_i\}_{i=1}^k$ of the labels, the variation within the p-th cluster as measured through the squared Euclidean metric is given by:

$$\frac{1}{|C_p|} \sum_{i,j \in C_p} d(x_i, x_j)^2. \tag{4.1}$$

The optimal clustering respecting the above principles is therefore found by choosing the partition $\{C_i\}_{i=1}^k$ that minimises the objective function

$$\sum_{p=1}^{k} \left[\frac{1}{|C_p|} \sum_{i,j \in C_p} d(x_i, x_j)^2 \right] \tag{4.2}$$

that simply sums the within-cluster variations. Although easily formulated, the search space for this optimisation problem consists of all partitions of the dataset, and the problem can in fact be shown to be NP-hard even in the planar case [64]. It is therefore unrealistic to solve in practice, and the k-Means algorithm based on Lloyd's algorithm [61], through which it is related to the later to be introduced Voronoi diagrams, is often used instead. The k-Means algorithm shown below is a hill climbing algorithm that converges to a local optimum.

Algorithm 2 The k-Means algorithm

Input: A dataset X and the number of clusters k.
Output: A partition $\{C_i\}_{i=1}^k$ of the labels of the datapoints.

1: Randomly assign each datapoint an initial cluster;
2: Compute the centroids (means) of the clusters, reassigning each datapoint to the cluster represented by the closest centroid and repeat until convergence;
3: Return the partition $\{C_i\}_{i=1}^k$ wherein each datapoint is assigned to the nearest centroid.

Some object to calling this algorithm the k-Means algorithm, stating "k-Means" is only the name of the clustering problem as their reason. Due to its effectiveness however, the above algorithm has come to be synonymous with solving the k-Means problem.

An interesting aspect of this algorithm is the random initialisation step. Instead of randomly assigning the datapoints to clusters, several other initialisation schemes have been proposed [7, 71]. Regardless, random initialisation is still the most popular one at the time of writing. The algorithm is usually ran multiple times, yielding different results each time due to the involved randomness, and the result giving the lowest value for the objective function given in Equation 4.2 is kept. If this is repeated q times the time complexity is given by $\mathcal{O}(qkmn)$ [58], where n is the number of datapoints and m is the dimension of the datapoints.

As a conceptual preparation for the next subsection, although it will not be treated, an alike problem called the k-Medians problem can be obtained from the k-Means problem with slight modifications.

4.2.2 k-Medoids Clustering

Before being able to think about k medoids, the single medoid needs to be defined: a *medoid* is a prototype that behaves much like a mean or median but is required to be an element of the dataset rather than of an ambient space. Given a dissimilarity function f and a subset $C \subseteq X$ of the data, the medoid c' of C can be defined formulaically as:

$$c' = \arg\min_{c \in C} \sum_{x \in C} f(c, x). \tag{4.3}$$

By now defining an objective function:

$$\sum_{p=1}^{k} \left[\sum_{x \in C_p} f(c'_p, x) \right], \tag{4.4}$$

a problem similar in nature to the k-Means problem is obtained. Instead of measuring the variation within a cluster by the averages of the Euclidean squared distances, the dissimilarities between the elements in the cluster and the medoid of the cluster are summed. The k-Medoids algorithm, also called the partitioning around medoids algorithm, heuristically solves this minimisation problem by repeatedly adjusting the clustering based on the medoids. This algorithm is given below.

Algorithm 3 The k-Medoids algorithm

Input: A dataset X, a dissimilarity function f and the number of clusters k.
Output: A partition $\{C_i\}_{i=1}^{k}$ of the labels of the datapoints.

1: Randomly select k elements of X to serve as the first medoids;
2: Assign each element to its most similar medoid, and therefore to a cluster;
3: As long as the value of the objective function decreases, check per cluster if replacing the medoid by another element in the cluster reduces variation. If so, select this element as the new prototype and reassign all elements;
4: Return the stabilised partition $\{C_i\}_{i=1}^{k}$.

Because of the use of dissimilaries instead of the Euclidean metric, this algorithm is more generally applicable. Furthermore, because the medoids are part of the dataset, the dissimilarities do not need to be recomputed after the first iteration. In the case of k-Means the prototypes may shift to arbitrary real vectors, and this means that the distances need to be recomputed during every iteration. Regardless, the k-Medoids algorithm is often more costly to run in practice with a time complexity of $\mathcal{O}(k(n-k)^2)$ [58].

4.3 Spectral Clustering

Spectral clustering derives its name from the importance of the spectra of matrices associated to graphs for its methods. In more detail, these spectra can be related to the number of connected components of the graph, and this knowledge can be exploited to find a clustering of a suitable dataset into groups. The entirety of this section is based on [88], a comprehensive tutorial on graph Laplacians and spectral clustering. The proofs of the propositions in this section can be found in this paper, as well as both algorithms discussed.

4.3.1 Graph Laplacians

Spectral graph theory studies graphs through the spectra of the matrices associated to them, for example the adjacency matrix, or more strongly pertaining to this section, the graph Laplacians. A graph Laplacian is a matrix that in some sense encodes a discrete analogue to the gradient known from calculus, measured when moving between the vertices. This subsection collects the different definitions used for the graph Laplacian in practice along with some elementary results.

Before beginning to build towards the definitions of the graph Laplacians, a note is made of conventions regarding the spectrum of a matrix. It will be assumed that eigenvectors are not normalised and therefore have no unique representation unless noted otherwise, and the set of eigenvalues will be assumed to be ordered increasingly. This allows for an ordering on the eigenvectors corresponding

to these eigenvalues.

Let $G = (V, E)$ be a non-negatively weighted and undirected graph with vertices $V = \{v_1, ..., v_n\}$ and weighted adjacency matrix W with entries w_{ij}. If two vertices v_i and v_j are not connected by an edge in the graph this is reflected in W by $w_{ij} = 0$. The *degree* d_i of any vertex v_i in the graph is defined as:

$$d_i = \sum_{j=1}^{n} w_{ij}, \tag{4.5}$$

and the *degree matrix* D of the graph is defined as the diagonal matrix $\text{diag}(d_1, ..., d_n)$. The graph Laplacian is then defined as the difference of the degree matrix and the weighted adjacency matrix.

Definition 4.3.1. (Graph Laplacian). The graph Laplacian $\mathcal{L}$ of G is defined as $\mathcal{L} = D - W$.

Alternatively the following definition can be used, which is called the normalised graph Laplacian for historical reasons.

Definition 4.3.2. (Normalised Graph Laplacian). The normalised graph Laplacian $\mathcal{L}^*$ is defined from the graph Laplacian $\mathcal{L}$ as the matrix $\mathcal{L}^* = D^{-1/2}\mathcal{L}D^{-1/2}$.

The first proposition concerning the graph Laplacians depends on specialised indicator vectors for subsets of the vertices of the graph, defined as follows.

Definition 4.3.3. (Indicator Vectors for Graphs). Let $A \subseteq V$ be a subset of the vertices of G. The indicator vector $1_A = (b_1, ..., b_n)$ is the binary vector that indicates for each vertex v_i if it is contained in A, meaning $b_i = 1$ if and only if $v_i \in A$.

The proposition itself relates the spectra of the graph Laplacians to the connected components of their graph, making it crucial to spectral clustering. As a preview to the succeeding proposition it is mentioned that 0 is always an eigenvalue of any graph Laplacian defined in this subsection.

Proposition 4.3.4. *The multiplicity k of the eigenvalue 0 of two graph Laplacians $\mathcal{L}$ and $\mathcal{L}^*$ equals the number of connected components $A_1, ..., A_k$ of the graph they are based on. The eigenvectors corresponding to the eigenvalue 0 of the unnormalised graph Laplacian $\mathcal{L}$ are the indicator vectors $1_{A_1}, ..., 1_{A_k}$ whereas these eigenvectors of the normalised graph Laplacian $\mathcal{L}^*$ are given by:*

$$D^{-1/2} \cdot 1_{A_1}, ..., D^{-1/2} \cdot 1_{A_k}. \tag{4.6}$$

The final result from this subsection is a collection of properties of the graph Laplacians.

Proposition 4.3.5. *(Properties of the graph Laplacians). If $\mathcal{L}$ is any graph Laplacian and $\mathcal{L}^*$ is any normalised graph Laplacian, the following statements are true:*

1. *Both $\mathcal{L}$ and $\mathcal{L}^*$ are positive semi-definite;*
2. *The smallest eigenvalues of $\mathcal{L}$ and $\mathcal{L}^*$ are 0;*
3. *All eigenvalues of $\mathcal{L}$ and $\mathcal{L}^*$ are real.*

Datasets with measures of similarity, or less generally distances between points, can be written as weighted graphs, and can therefore be subjected to a cluster analysis using the propositions from this subsection. This type of clustering algorithm is the topic of discussion for the next subsection.

4.3.2 Spectral Clustering Algorithms

The spectral clustering algorithms treated in this subsection take *similarity matrices* that contain the similarity of the points in a dataset as their input. As an example, such a matrix can canonically be constructed when the dataset comes equipped with a pre-defined metric d. Under this assumption the similarity matrix would be defined as the matrix S with entries $s_{ij} = d(x_i, x_j)$ for all points x_i, x_j in the dataset. Based on the information the similarity matrix provides, different weighted graphs with the datapoints as their vertices can be realised, and these graphs are called *similarity graphs*. Each similarity graph recipe has different conditions for the insertion of an edge into the graph, but all model the local relationships of the datapoints through their edges.

In abstract formulation the following spectral clustering algorithms take in a similarity matrix, construct a similarity graph and extract the clusters after changing the representation of the data.

Algorithm 4 Unnormalised Spectral Clustering

Input: A similarity matrix $S \in \mathbb{R}^{n \times n}$ and the number of clusters k.
Output: A partition $\{C_i\}_{i=1}^{k}$ of the datapoints identified by their label.

1: Construct a similarity graph based on S;
2: Determine the graph Laplacian $\mathcal{L}$ of the similarity graph;
3: Compute the first k eigenvectors $v_1, ..., v_k$ of $\mathcal{L}$;
4: Let $A \in \mathbb{R}^{n \times k}$ denote the matrix with the eigenvectors $v_1, ..., v_k$ of $\mathcal{L}$ as its columns;
5: Use the k-Means algorithm to find a clustering $\{C_i\}_{i=1}^{k}$ of the row vectors A and return it.

Like the graph Laplacian has a normalised counterpart, so too does the unnormalised spectral clustering algorithm. It varies slightly from the above algorithm, most notably in that it has an added normalisation step for the eigenvectors. Both have a time complexity of $\mathcal{O}(n^3)$ [91].

Algorithm 5 Normalised Spectral Clustering

Input: A similarity matrix $S \in \mathbb{R}^{n \times n}$ and the number of clusters k.
Output: A partition $\{C_i\}_{i=1}^{k}$ of the datapoints identified by their label.

1: Construct a similarity graph based on S;
2: Determine the normalised graph Laplacian $\mathcal{L}^*$ of the similarity graph;
3: Compute the first k eigenvectors $v_1, ..., v_k$ of $\mathcal{L}^*$ and normalise them to unit length;
4: Let $A \in \mathbb{R}^{n \times k}$ denote the matrix with the normalised eigenvectors of $\mathcal{L}^*$ as its columns;
5: Use the k-Means algorithm to find a clustering $\{C_i\}_{i=1}^{k}$ of the row vectors A and return it.

Both algorithms are very similar, but it should be noted that the normalised spectral clustering algorithm assumes a defined norm or metric. If this structure is given the choice of algorithm is situational. At a glance, the change in representation of the data in both algorithms may seem arbitrary. The following subsection offers intuition for when and why these algorithms are expected to result in a satisfying clustering of the dataset, and addresses the need for a normalisation step in the second algorithm.

4.3.3 Motivation using Perturbations

The motivation in this subsection is one of multiple offered in [88], where the reader can find alternative explanations in terms of graph cuts and random walks. The explanation in this subsection bases itself on perturbation theory.

Perturbation theory is the theory that studies small disturbances and is often used in numerical analysis, where otherwise unsolvable real life phenomena are considered as small perturbations of a solvable problem. The use of such disturbances is then to solve a simpler or exact problem and derive an approximate solution to the assumedly perturbed and more complex instance of the same problem. The general idea for the use of this theory to motivate spectral clustering is then as follows.

The ideal case where the similarity graph consists of the input number k of connected components is considered, and the actual dataset's similarity graph is considered to be a perturbed instance of this graph. Following from Proposition 4.3.4, the set of row vectors in the matrix A found in the unnormalised algorithm for the ideal case are precisely the indicator vectors for the k connected components, and the k-Means algorithm trivially detects k clusters with these indicator vectors as their centroids. If the perturbation is small, one would hope that this translates to a small change in the connected components, and therefore a reliable clustering of the actual dataset. This explanation can analogously be given in more generality by considering the similarity matrix used to construct the similarity graph, which gives a more intuitive basis for perturbations in general.

The remainder of this subsection is dedicated to a theoretical basis for the above explanation in terms of matrix perturbation theory. The first definition is therefore that of a perturbation of a matrix.

Definition 4.3.6. (Matrix Perturbation). Let M be any matrix, and H a matrix of the same dimensions. If H is considered small compared to M with respect to any measure of size, the matrix $\tilde{M} = M + H$ is said to be a perturbation of M.

The meaning of a measure of size used in this definition was kept vague on purpose, so that many such measures remain eligible. A norm can be used, and as the Frobenius norm is required later in this subsection it will be given as an example.

Definition 4.3.7. (Frobenius Norm). The Frobenius norm $\|M\|_F$ of a matrix M is defined by:

$$\|M\|_F = \sqrt{\operatorname{trace}(M^{\mathrm{T}}M)}. \tag{4.7}$$

Other required definitions concern angles between subspaces of a fixed Euclidean space. These definitions depend on singular values of matrices and extend the familiar notion of angles between vectors.

Definition 4.3.8. (Singular Values). The singular values of a matrix M of finite dimensions are the square roots of the eigenvalues of the product $M^{\dagger}M$.

Readers with more experience in data analysis or statistics may recognise this definition as it is used in the singular value decomposition required for principal component analysis. Here, the singular values are stepping stones towards the definition of canonical angles between Euclidean subspaces.

Definition 4.3.9. (Canonical Angles). Let M_1 and M_2 be two matrices whose respective columns form orthonormal bases for two m-dimensional subspaces of $\mathbb{R}^n$, and let σ_i denote the singular values of $M_1^{\dagger}M_2$ for $i = 1, ..., m$. The canonical angles of these spaces are then given by $\arccos(\sigma_i)$ for $i = 1, ..., m$.

For $m = 1$ this definition coincides with the definition of an angle between two vectors, and it therefore generalises this angle to a set of angles which is in fact invariant under a wide class of transformations of the ambient Euclidean space. This definition can be even further generalised to subspaces of differing dimensions. In practice these angles are used to measure the distance between the two column spaces of M_1 and M_2 through the norm of the diagonal matrix:

$$S(M_1, M_2) = \text{diag}(\sin\sigma_1, ..., \sin\sigma_m) \tag{4.8}$$

that contains the sines of the canonical angles between the column spaces of M_1 and M_2.

The main result required from matrix perturbation theory can now be stated. The statement of this theorem uses the conventions that for a matrix M and an interval $I \subset \mathbb{R}$, the set of all eigenvalues of M contained in I is given by $\lambda_I(M)$, the space $V_I(M)$ is the eigenspace spanned by the eigenvectors corresponding to these eigenvalues, and

$$\delta_I(M) = \inf\{|\lambda - x| \text{ for } \lambda \in \lambda_{\bar{I}}(M) \text{ and } x \in I\} \tag{4.9}$$

denotes the distance from I to the closest eigenvalue of M in the complement $\bar{I}$ of I in $\mathbb{R}$ defined using the infimum.

Theorem 4.3.10. *(Davis-Kahan). Let M be a square symmetric matrix and let $\tilde{M}$ be a perturbation of M by a second symmetric matrix. If $I \subset \mathbb{R}$ is any interval, then the distance between the eigenspaces $V_I(M)$ and $V_I(\tilde{M})$ is bounded above by:*

$$\left\| S(V_I(M),\, V_I(\tilde{M})) \right\|_F \leq \frac{\left\| \tilde{M} - M \right\|_F}{\delta_I(M)}. \tag{4.10}$$

This theorem is both stated in more generality and proven in [11]. Readers interested in matrix perturbation theory are also referred to this source, and this opportunity is taken to redirect readers interested in a rich theory of linear subspaces to the theory of Grassmannians, as can be found in for instance [57].

This theorem can now be directly related to the explanation given earlier in this subsection. If a graph Laplacian is substituted for M, a small perturbation should lead to a small change in the eigenspaces, and therefore conserve the desired properties from the ideal case to a certain degree.

Finally, the need for a normalisation step in the normalised spectral clustering algorithm can be addressed. In the above explanations for the unnormalised spectral clustering algorithm, the rows taken from the matrix A formed in the algorithm can be shown to be indicator vectors and much of the theory in this section is founded on this fact. For the normalised graph Laplacian however, it follows from Proposition 4.3.4 that these rows each contain exactly one non-binary entry in the ideal case. The row normalisation step would counter this problem and thus make the above theory applicable. Once again this step relies on the assumption that the change in properties is somehow proportional to the size of the perturbation.

Chapter 5

Persistent Homology

5.1 Common Elements

This section serves as a precursory section to the following sections on persistent homology and quantum persistent homology, introducing point clouds as well as several different ways in which simplicial complexes can be constructed on a point cloud. The concept of persistent homology depends heavily on these constructions, as it assumes a point cloud is a set of points sampled from some topological space and attempts to approximate its homology. Simplicial complexes are essential to this process as they are used to bridge the gap between the discrete sample and the underlying space. On its own, a finite point cloud is discrete and therefore its homology is of no interest. Although point clouds and simplicial complexes are common to both algorithms they are treated differently, which is the reason their introduction is abstracted to this additional section.

5.1.1 Point Clouds

All qualitative datasets without further structure can be seen as subsets of an Euclidean space, so that the dataset inherits the Euclidean metric. If another metric is defined, or the datapoints are not qualitative, this may not necessarily be true. In the generalised case any dataset representable by real vectors accompanied by some metric forms a point cloud. This subsection describes point clouds, datasets eligible as input for persistent homology, and addresses under what circumstances a dataset may be translated to a point cloud. As the following definition reveals, point clouds are easily described.

Definition 5.1.1. (Point Cloud). An n-dimensional point cloud X is a subset $X \subseteq \mathbb{R}^n$ together with a metric on $\mathbb{R}^n$. Any dataset that can be represented as such is called point cloud data.

Note that this definition implies that the data does not need to be given as a subset of some real space, but only that it can be represented as one. In practice a map that effectuates this translation is required to be given. As a minor example, consider the group $\mathrm{U}(n)$ of unitary matrices. There is a natural map from the matrices in this group to the n^2-dimensional Euclidean space, meaning a set of unitary matrices may be referred to as point cloud data.

When translating between metrics spaces there may be stronger requirements involved, for instance that the mappings preserve the exact distances between elements. To this end isometric embeddings are defined.

Definition 5.1.2. (Isometric Embedding). Let (X, d_X) and (Y, d_Y) be two metric spaces. An isometric embedding of (X, d_X) into (Y, d_Y) is a map $f : (X, d_X) \to (Y, d_Y)$ so that for all $x_1, x_2 \in X$:

$$d_X(x_1, x_2) = d_Y(f(x_1), f(x_2)). \tag{5.1}$$

In full generality an embedding is a map that shows an instance of a structure as a substructure of another structure, and it is isometric if it preserves distances. Henceforth any representation of a dataset as a point cloud will be referred to as an embedding of the dataset into a real space.

As a characterisation of when a metric space can be isometrically embedded in an Euclidean space the following theorem is given. It is proven in [12].

Theorem 5.1.3. *Let X be a metric space. There exists an isometric embedding of X into $(\mathbb{R}^n, d_{Eucl})$ if and only if every finite subset with at most $n+3$ elements can be isometrically embedded in this same space.*

In the next subsection processes for the construction of simplicial complexes on point clouds are discussed. It will also see to a more general definition of simplicial complexes that is required when the complex is built on an intermediate structure rather than the point cloud directly.

5.1.2 Construction of Simplicial Complexes

The types of simplicial complexes treated in this subsection were specifically chosen because they have seen widespread use, were implemented classically in common packages and offer a wide variety of options. They are therefore well studied and established. A source that offers directions towards other simplicial complexes is the survey paper [72].

Throughout this subsection an arbitrary point cloud X with associated metric d will be considered. As an opening note on notation, the convention of using $S_\epsilon(X)$ to denote an arbitrary simplicial complex at scale ϵ on X will be used, and this may be shortened to S_ϵ if no confusion may arise. This is useful if a statement regarding a simplicial complex is made that does not discriminate based on how the simplicial complex was constructed, as this subsection introduces several such recipes.

The first two constructions, the Čech complex and the Vietoris-Rips complex [16], are constructed in similar ways and will moreover be shown to be related. For their construction the spaces:

$$X(\epsilon) := \bigcup_{x \in X} \bar{B}_d(x, \epsilon/2) \subset (\mathbb{R}^n, d) \tag{5.2}$$

at different scale values ϵ are considered. These spaces are unions of closed balls with diameter ϵ centered around the elements of the point cloud, as the following figure shows for a small two-dimensional example.

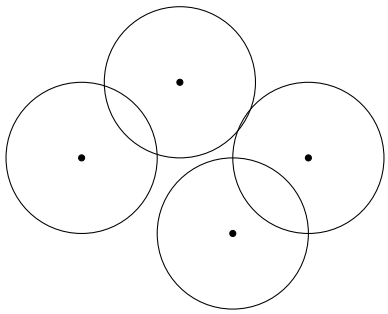


Figure 5.1: A set $X(\epsilon)$.

The mentioned complexes base themselves on these spaces by adding simplices conditioned on the overlappings of the closed balls, specifically as follows.

Definition 5.1.4. (Čech Complex). The Čech complex $C_\epsilon(X)$ on X is the simplicial complex whose k-simplices are determined by unordered finite sequences of points $\{x_i\}_{i=0}^k$ in X for which the closed balls $\bar{B}(x_i, \frac{\epsilon}{2})$ in $X(\epsilon)$ have a point of common intersection.

A k-simplex $\sigma = (x_0, ..., x_k)$ can therefore be a part of a Čech complex if and only if the balls surrounding all $k+1$ of its vertices have a non-empty shared intersection. This equals the following statement:

$$\sigma \in C_\epsilon(X) \Leftrightarrow \bigcap_{i=0}^{k} \bar{B}_d(x_i, \epsilon/2) \neq \varnothing. \tag{5.3}$$

The Čech complex is often seen as the ideal simplicial complex based on the $X(\epsilon)$ spaces. The explanation requires a few general topological definitions, starting from homotopies.

Definition 5.1.5. (Homotopy). A homotopy is a family of maps $h_t : X \to Y$ between topological spaces, indexed over the unit interval I, so that the associated map $H : I \times X \to Y$ defined by $H(t, x) = h_t(x)$ is continuous. Two continuous maps $f, g : X \to Y$ are called homotopic, denoted $f \simeq g$, if there exists a homotopy h_t with $h_0 = f$ and $h_1 = g$ connecting them.

A homotopy is a continuous deformation of one map into another, and as an example, two homotopic paths are intuitively best pictured as two curves with shared start and end points with either able to be continuously slid onto the other. In turn, homotopies lead to a notion of equivalence between topological spaces that is broader than the homeomorphism.

Definition 5.1.6. (Homotopy Equivalence). A continuous map $f : X \to Y$ between topological spaces is called a homotopy equivalence if there exists a continuous map $g : Y \to X$ such that $g \circ f \simeq 1_X$ and $f \circ g \simeq 1_Y$. If such maps exist the spaces X and Y are to be said homotopy equivalent, or of the same homotopy type. This is denoted $X \simeq Y$.

Every homeomorphism in particular can be shown to be a homotopy equivalence although the converse does not hold, and the most interesting aspect of this new idea is that topological properties such as homology are still invariant under homotopy equivalence [45]. This fact, combined with the following variant of the Nerve theorem [16], illuminates why the Čech complex is so important.

Theorem 5.1.7. *The Čech complex $C_\epsilon(X)$ is homotopy equivalent to $X(\epsilon)$.*

As per the theory discussed so far, the homology of $X(\epsilon)$ can not directly be computed. The above theorem however guarantees that the simplicial homology of $C_\epsilon(X)$ corresponds to that of $X(\epsilon)$, so that knowing one automatically grants you the other. In a way the Čech complex can therefore be thought of as accurately representing the space $X(\epsilon)$, at least in terms of homology.

The downside to the Čech complex is that its construction, as per the above definition, has a large asymptotic complexity as every possible simplex needs to be checked. This problem is neutralised to some degree by relying on the Vietoris-Rips complex.

Definition 5.1.8. (Vietoris-Rips Complex). The Vietoris-Rips complex $VR_\epsilon(X)$ on X is the simplicial complex whose k-simplices are determined by unordered finite sequences of points $\{x_i\}_{i=0}^k$ in X for which the $\bar{B}(x_i, \frac{\epsilon}{2})$ in $X(\epsilon)$ have pairwise non-empty intersections.

This complex admits a k-simplex if and only if all of its 1-simplices are already in the complex, meaning the complex is completely determined by its 1-simplices. In order to more accurately discuss substructures as these, the following definition introduces skelata of simplicial complexes.

Definition 5.1.9. (n-Skeleton). The n-skeleton S^n of a simplicial complex S is the simplicial complex consisting of all simplices in S of dimension n or lower.

Note that this notation should not cause the n-skeleta to be confused with n-spheres, as this notation is usually reserved for the latter. This choice was made under the consideration that n-spheres will not be explicitly used in this thesis.

The full Vietoris-Rips complex VR is the clique complex of VR^1, which can be treated as a graph since the 0-simplices can be equated to nodes, and the 1-simplices can be equated to edges. It can therefore be stored in graph datastructures, which opens this complex to the already existing and significant number of results on graph processing. More important however, speaking from a theoretical point of view, is its following relation to the Čech complex.

Proposition 5.1.10. *For any positive scale parameter ϵ,*

$$C_\epsilon(X) \subseteq VR_{2\epsilon}(X) \subseteq C_{2\epsilon}(X). \tag{5.4}$$

The implications of this proposition will become apparent when sequences of simplicial complexes are considered, and the reader is encouraged to revisit this proposition after filtrations of simplicial complexes are defined in the section following this one. For now, it is noted that this proposition can be found in [16], and that although the construction of the Vietoris-Rips complex is significantly faster than it is for the Čech complex, the problem can be reduced to the clique problem and is therefore NP-complete [54]. Regardless, the Vietoris-Rips complex has been shown to work well in practice [17, 23, 83].

Before moving on to describe other complexes, the difference between the Vietoris-Rips complex and the Čech complex is illustrated by the following figure.

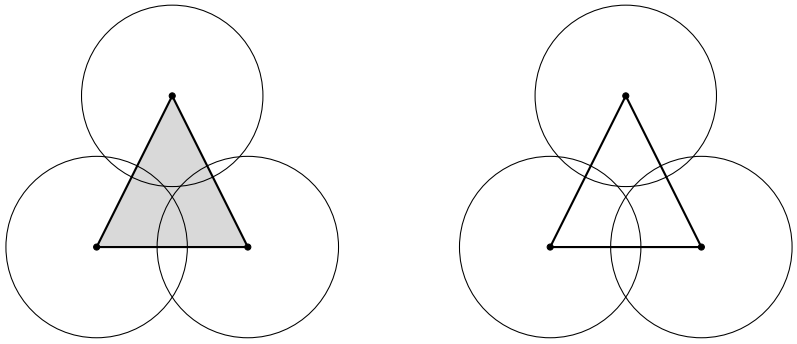


Figure 5.2: The Vietoris-Rips and Čech complexes compared.

This figure shows that the Vietoris-Rips complex on the left contains a 2-simplex at this scale, as its 1-faces are part of the complex. On the right, the Čech complex does not admit a 2-simplex as even though the 1-simplices that would form its faces are present, the closed balls have trivial intersection.

As an alternative, the Witness complex was first introduced in [22] specifically to combat the inherent high simplicial dimensions of the Vietoris-Rips and Čech complexes. The mechanism that reduces these dimensions is simultaneously the reason the Witness complex is easier to compute. It is shown in the following definition that this complex does not depend on the full point cloud, but is instead based on a subset $L \subseteq X$ called the *landmarks* of the complex.

Definition 5.1.11. (Witness Complex). For every $x \in X$, let $m_x = \min_{l \in L} d(x, l)$. The Witness complex $W_\epsilon(X, L)$ on X is the simplicial complex with the landmarks L as its 0-simplices and that admits a k-simplex $(l_{i_0}, ..., l_{i_k})$ as a member if and only if there exists a witness $x \in X$ to the simplex so that $d(x, l_{i_j}) \leq m_x + \epsilon$ for all landmarks $l_{i_0}, ..., l_{i_k}$.

The Witness complex is best thought of as simply a complex with a lot of degrees of freedom, as the choice for the landmarks is free. Before expanding on the choices involved, the following figure should give some geometric intuition to accompany the abstract definition.

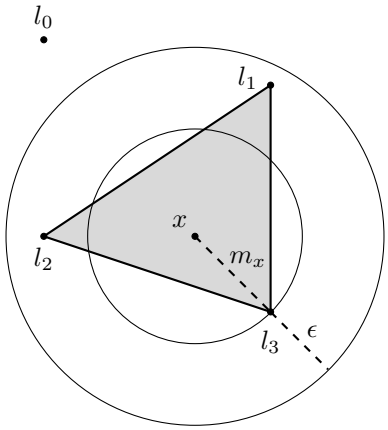


Figure 5.3: An example of a 2-simplex with a point x as its witness.

This figure shows a 2-simplex within the context of a landmark set $\{l_0, l_1, l_2, l_3\}$. For this choice of scale ϵ, the point x is a witness to the simplex (l_1, l_2, l_3), because all three of its elements lie within the ball $B_d(x, m_x + \epsilon)$. The last landmark l_0 is excluded because it is not within the specified range.

The choice for the number of landmarks is case-dependent, but the original paper suggests that a ratio of the point cloud size to the number of landmarks of at least 20 works well for two-dimensional surfaces. A precise rule in any number of dimensions is unknown. With this number chosen, the paper offers two ways of selecting the landmarks. The first is a random initialisation, and the second is an inductive procedure executed as follows. First, choose an arbitrary landmark l_0. In any following step, assume that i landmarks

$$L_i = \{l_0, ..., l_{i-1}\} \tag{5.5}$$

have already been chosen. The next landmark l_i is then chosen as the element of $X \setminus L_i$ that maximises the objective function

$$x \mapsto \min_{l \in L_i} d(x, l), \tag{5.6}$$

and this is repeated until the desired number of landmarks were chosen. This procedure should result in a set of landmarks with high variability, but one that also tends to favour extreme points. For this reason careful consideration is required when Witness complexes are used.

As per [22], the idea for the Witness complex arose from the theory of Delaunay triangulations from computational geometry [21], which in turn are closely related to Voronoi diagrams. Based on these notions one more type of parametrised simplicial complex, the Alpha complex [6], is defined.

Definition 5.1.12. (Voronoi Diagram). Let $\{P_i\}_{i\in I}$ be a sequence of non-empty subsets of $\mathbb{R}^n$ indexed over $I \subseteq \mathbb{N}$. Then every P_k is called a site and the Voronoi cell associated to P_k is given by:

$$V_k = \{x \in \mathbb{R}^n \mid \forall i \in I \setminus \{k\} : d(x, P_k) \leq d(x, P_i)\}. \tag{5.7}$$

The sequence $V = \{V_i\}_{i\in I}$ of Voronoi cells covers $\mathbb{R}^n$ and is called a Voronoi diagram of $\mathbb{R}^n$.

This definition depends on a measure of distance between points x and subsets A of $\mathbb{R}^n$ similar to the one used in the statement of the Davis-Kahan theorem in the previous chapter, here specifically defined as:

$$d(x, A) = \inf\{d(x, a) \mid a \in A\}. \tag{5.8}$$

Voronoi diagrams can be efficiently computed in two dimensions by Fortune's algorithm [34]. In higher dimensions however, this process becomes more complex [39].

Delaunay triangulations can be characterised as combinatorially described simplicial complexes based on Voronoi cells. Because of this their definition requires more generality than the definition of simplicial complexes used thus far can provide, and the first course of action should be to extend the latter's definition.

Definition 5.1.13. (Abstract Simplicial Complex). An abstract simplicial complex $S = (V, \Sigma)$ is a set of vertices V and a collection Σ of finite non-empty subsets of V called abstract simplices so that the membership $\sigma \in \Sigma$ implies that every non-empty subset of σ is also an element of Σ.

Intuition dictates that a face of an abstract simplex σ should simply be a subset of σ, and the dimension of σ should be defined as its cardinality. All other terms and prior intuition should transfer to this new definition.

Definition 5.1.14. (Delaunay Triangulation). The Delaunay Triangulation $D(X, V)$ defined over a given Voronoi diagram V of $\mathbb{R}^n$ is the abstract simplicial complex (X, Σ) where:

$$\Sigma = \{\sigma \subseteq X \mid \bigcap_{x\in\sigma} V_x \neq \varnothing\}. \tag{5.9}$$

The final simplicial complex to be defined in this subsection can be extracted as a subcomplex from a Delaunay triangulation by adding scale constraints. For its definition, let $m(x)$ denote the smallest index of a Voronoi cell the element $x \in X$ is a member of and define:

$$V(x, \epsilon) = V_{m(x)} \cap \bar{B}_d(x, \epsilon) \tag{5.10}$$

to be the intersection of this cell with the closed ball of radius ϵ centered around x. The Alpha complex is then defined as follows.

Definition 5.1.15. (Alpha Complex). The Alpha complex $A_\epsilon(X)$ on X is the abstract simplicial complex (X, Σ) whose abstract simplices are given by:

$$\Sigma = \{\sigma \subseteq X \mid \bigcap_{x\in\sigma} V(x, \epsilon) \neq \varnothing\}. \tag{5.11}$$

The Alpha complex has several interesting properties. First, it is homotopy equivalent to the Čech complex at the same scale ϵ, meaning the Alpha complex too has the same homology as $X(\epsilon)$ per transitivity [22]. However, in contrast to the Čech complex, the dimension of the Alpha complex is upper bounded by the dimension of the point cloud [72].

5.2 Persistent Homology

It follows from the previous section that if a point cloud dataset is given and the simplicial homology of the space assumed to underly the dataset is to be estimated, a simplicial complex needs to be constructed. This construction itself however requires an arbitrary choice of the scale parameter, which completely determines the topology of the complex. It is therefore preferable to investigate the complexes at different scale parameters and register how the homology of the approximating complexes change as the scale changes, resulting in the idea of persistent homology. The theory developed in this section makes this idea precise and offers two summaries of this information in the form of barcodes and persistence diagrams.

5.2.1 Persistent Betti Numbers

When wanting to base an analysis of any kind on multiple scale parameters, it is useful to bundle these different values together. A sequence is the natural choice of object, and the result is named a parameter sequence.

Definition 5.2.1. (Parameter Sequence). A parameter sequence $\mathbf{s} = \{\epsilon_i\}_{i=1}^n$ of length n is a finite and strictly increasing sequence of positive real numbers.

Defining this object as a finite sequence is a conscious decision. As any real world point cloud dataset X is necessarily finite, the set of scale parameters that define a unique complex on this cloud is finite itself, for the Čech and Vietoris-Rips complexes bounded above by the diameter

$$\max_{x_1,x_2 \in X} d(x_1, x_2) \tag{5.12}$$

of the dataset. Therefore, given any infinite parameter sequence, there exists a finite subsequence that represents exactly the same information and the complexity of the sequence can be severely reduced by discarding the redundant scale parameters. From this point onwards the finiteness of point clouds will be assumed.

Given any parameter sequence, a simplicial complex $S_\epsilon(X)$ can be constructed on the point cloud for any element ϵ of the sequence. By the same reasoning these complexes too can be taken together to form one singular object.

Definition 5.2.2. (Filtration on a Point Cloud). Let X be a point cloud, and $\mathbf{s} = \{\epsilon_i\}_{i=1}^n$ a parameter sequence. The ordered sequence

$$\mathcal{F}_\mathbf{s} = \{S_{\epsilon_i}(X)\}_{i=1}^n \tag{5.13}$$

of simplicial complexes is called a filtration of simplicial complexes on X.

Theoretical papers on persistent homology may opt to define these filtrations by way of Morse functions [67] from differential topology. Although this definition removes the need for an explicit parameter sequence, their use is infeasible in practice. For this reason the theory required for Morse functions and this consequent alternative definition will not be treated in this thesis.

Instead of computing the simplicial homology of the individual complexes, the homology can be computed within the context of the different complexes in the filtration. The direct underlying object for persistent homology is therefore a filtration instead of a single simplicial complex. Let ∂_*^i denote the boundary operators associated to the chain complex of the i-th simplicial complex in the filtration for the next definition.

Definition 5.2.3. (Persistent Homology). Let $\mathcal{F}_{\mathbf{s}}$ be a filtration based on a parameter sequence of length n. For any $p \in \mathbb{N}$, the k-th p-persistent homology group of the i-th simplicial complex in the filtration is defined as:

$$\mathcal{H}_k^{i,p} = \left(\ker \partial_k^i\right) / (\operatorname{im} \partial_{k+1}^{i+p} \cap \ker \partial_k^i). \tag{5.14}$$

In order to unpack this definition, first notice the likeliness to the definition of simplicial homology. Here too the quotient of all closed cycles of k-simplices by the boundaries is taken, with the difference being that the boundaries are now taken from the larger $(i+p)$-th complex in the filtration. Intersecting this image with the kernel ensures that the quotient is well defined. The simplices of the i-th complex must be strictly contained within any successive complex in the filtration, but the holes that were present in the i-th complex may have closed since. Therefore the elements of the p-persistent homology groups correspond to all holes that have persisted in the p complexes in the filtration following the i-th complex.

For further remarks, notice that the persistent homology groups stabilise past a certain point where the chain groups become trivial by the non-existence of generating simplices. The number of persistent homology groups is therefore bounded, as there can not exist simplices on the point cloud with more vertices than the point cloud has elements. Notice too that this definition can be formulated in terms of vector spaces as well by choosing suitable field coefficients. For a deeper treatment of persistent homology the reader is refered to [66].

From a practical point of view, the next definition should be an analogue to the Betti numbers that summarise the simplicial homology of a single complex. Since persistent homology is defined over a finite point cloud the following definition has no finiteness requirements like the definition of the Betti numbers did have.

Definition 5.2.4. (Persistent Betti Numbers). The k-th p-persistent Betti Number of the i-th simplicial complex in a filtration is the rank of $\mathcal{H}_k^{i,p}$, denoted by $\beta_k^{i,p}$.

As a digression, the persistent analogue to the Betti numbers could be defined as sequences

$$\boldsymbol{\beta}_k^i = \{\beta_k^{i,j}\}_{j=i}^{|\mathbf{s}|-i} \tag{5.15}$$

rather than single numbers in order to stay consistent with the dependence of persistent homology on sequences. The choice of definition is not particularly important, as the analysis of persistent homology relies mostly on visual summaries in the form of barcodes and persistence diagrams.

5.2.2 Barcodes and Persistence Diagrams

Once the persistent homology groups are known they can be represented visually in two different ways. In order to not let the connection to the abstract theory waver, holes will be referred to as homology generators for the remainder of this subsection, which is the nomenclature from algebraic topology. This is especially important because the full category theoretical justification for these summaries is not given. Instead, the theorem that lies at the heart of their definitions is used as a springboard, and the full foundations are discussed in [66].

Theorem 5.2.5. *(Structure of Persistent Homology Modules). Given a filtration on a point cloud X, the persistent homology modules over a field $\mathbb{F}$ admit decompositions of the following form:*

$$\mathcal{H}_*(X;\mathbb{F}) \cong \left(\bigoplus_{i=1}^{m} \Gamma^{x_i}\mathbb{F}[t]\right) \oplus \left(\bigoplus_{i=1}^{n} \Gamma^{y_i}\mathbb{F}[t]/(t^{n_i})\right). \tag{5.16}$$

The object on the left hand side of this equation is a persistent homology module, encoding the k-th persistent homology in full. The decomposition on the right hand side is split into two components. The first corresponds to all homology generators that appear at some scale x_i and remain in all successive complexes. The second component captures all homology generators that appear at scale y_i and disappear from the filtration at scale $y_i + n_i$. This theorem thus grants a method to extract the *birth* and *death* times of the homology generators, and with them the *lifespan* of said generator.

Definition 5.2.6. (Bar). A bar is an ordered pair $(b, d) \in \mathbb{Z} \times \mathbb{Z} \cup \{\infty\}$ with $b < d$.

Every homology generator has an associated bar (b, d) and the lifespan $d-b$ can be directly computed from it. Because a mapping from a persistent homology module to a set of bars can be defined, and the bars are ordered pairs of numbers, they can be represented in a visual summary called the barcode.

Definition 5.2.7. (Barcode). The barcode of a filtration on a point cloud is the ordered set of all bars corresponding to the persistent homology of the filtration, ordered on the dimension of the homology generators and birth scale in that order.

The following diagram may be considered as a stylised example of a barcode.

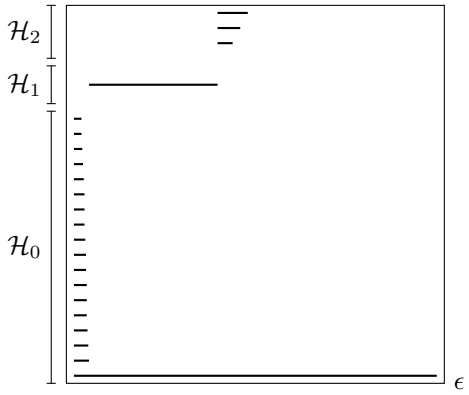


Figure 5.4: An example of a barcode based on a point cloud sampled from a circle.

Every homology generator is literally represented as a bar, with its birth and death visible along the horizontal axis. This summary clearly shows which of the generators persist the longest and are therefore most likely to represent true homological properties of the underlying space from which the point cloud is assumed to be sampled. This same explanation holds for the stylised persistence diagram shown on the next page, which is a different but equivalent visualisation of the bars.

Every bar is now shown as a point plotted above the diagonal line, its value along the horizontal axis signifying its birth, and the value along the vertical axis its death. Because the death of a generator cannot occur before its birth, no points can be plotted on or below the diagonal. The points are often styled by colour or shape based on their dimension for clarity.

Other summaries of persistent homology exist. The most prominent of these is the persistence landscape [15]. In contrast to barcodes and persistence diagrams, the persistence landscapes can be treated as elements of a normed vector space and are suited to calculus, meaning they are receptive to statistical methods. This means that these summaries can for instance be averaged for repeated experiments. Their name derives from the fact that when plotted, the results look likes a landscape of hills. These landscapes are not treated here because they require knowledge of measure theory.

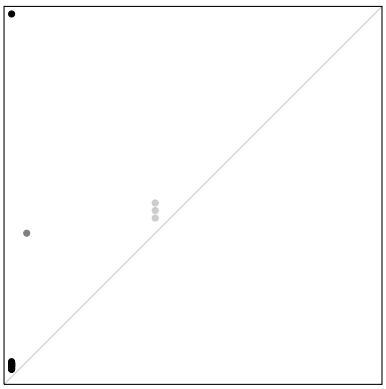


Figure 5.5: A persistence diagram based on the same point cloud sample from a circle.

5.2.3 Computation

This subsection offers a method for the algorithmic computation of homology, which can be extended to directly accomodate the computation of persistent homology as was done in [92]. Because of its effectiveness this algorithm is often referred to as the standard algorithm, even though it is officially named the reduction algorithm. The algorithm revolves around the reduction of matrices to their Smith normal form, introduced in the next definition.

Definition 5.2.8. (Smith Normal Form). A matrix M with entries in a field $\mathbb{F}$ is said to be in Smith normal form if it is represented as a diagonal matrix

$$M = \begin{pmatrix} m_1 & & & & & \\ & \ddots & & & & \\ & & m_i & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix} \tag{5.17}$$

for non-zero $m_1, ..., m_i \in \mathbb{F}$ so that m_j is a divisor of m_{j+1} for all $j < i$.

It is stated without proof that every matrix over a field can be written in Smith normal form. As the usage of a field suggests, this subsection will treat the homology of a topological space as a sequence of vector spaces. With this being the case, the k-th vector space in the sequence has a basis of k-simplices, and the boundary operator ∂_k can be realised as a matrix called the *standard matrix representation* of the boundary operator with respect to these bases. It follows directly from the definition of the boundary operator that this matrix has entries taken from $\{-1, 0, 1\}$ corresponding to the orientations of the simplices in the boundary, and these matrix representations completely determine the homology. Explaining the reduction algorithm equates to explaining why this is the case.

Assume the vector spaces Δ_i of i-chains are given, and their dimensions are denoted by d_i. For any i the standard matrix representation of $\partial_i : \Delta_i \to \Delta_{i-1}$ is given by a $d_{i-1} \times d_i$ matrix with null space $\ker \partial_i$ and range space $\operatorname{im} \partial_i$. Reducing this matrix to Smith normal form by elementary row and column operations therefore reveals essential information: modified bases $\{e_{j,i}\}$ for the spaces Δ_i, and through them the dimensions of the null spaces and range spaces. These bases follow from the changes in the initial bases of simplices induced by the row and column operations. If the Smith normal form of ∂_i is determined by ordered entries $m_1, ..., m_{l_i}$ the following information is known.

1. The set $\{e_{k,j} \mid l_k + 1 \leq j \leq d_k\}$ is a basis for $\ker \partial_k$ and therefore $\dim \ker \partial_k = d_k - l_k$;
2. The elements of $\{m_j e_{k-1,j} \mid 1 \leq j \leq l_k\}$ form a basis for $\operatorname{im} \partial_k$ and $\dim \partial_k = l_k$;
3. The k-th Betti number can be computed as $\beta_k = d_k - l_k - l_{k+1}$.

The formula for the k-th Betti number is derived as the codimension of $\operatorname{im} \partial_{k+1}$ in $\ker \partial_k$:

$$\begin{aligned} \beta_k &= \dim H_k \\ &= \dim \ker \partial_k / \operatorname{im} \partial_{k+1} \\ &= \dim \ker \partial_k - \dim \operatorname{im} \partial_{k+1} \\ &= d_k - l_k - l_{k+1}. \end{aligned} \tag{5.18}$$

The complexity of the reduction algorithm is cubic in the number of simplices in the complex [92].

5.3 Quantum Persistent Homology

This section is based on the description of an incomplete quantum algorithm for the computation of persistent homology given in [59]. The general idea is that one can forego the expensive construction of a simplicial complex if one is given access to a quantum computer, instead approximating it in a quantum state through the use of Grover's search algorithm. The paper uses the Vietoris-Rips complex in particular. Following this, the combinatorial Laplacians, generalisations of the unnormalised graph Laplacian that are tied to the homology of the complex, can be derived. It is then attempted to approximate the persistent homology through this construction. Although this method may fail to compute the persistent Betti numbers of a filtration, it can be used to approximate the Betti numbers of a simplicial complex. This routine was implemented on a quantum computer as a proof of concept, and the results were published in [48].

5.3.1 Simplicial Complexes in Quantum States

For a simplicial complex S of choice, ensure that an ordering is fixed on the elements of its 0-skeleton $S^0 = \{s_1, ..., s_n\}$. This subsection will concern itself with the construction of a quantum state encoding the complex, and in order to do this a simplex σ in the complex will be represented by a bitstring of length n where the value on the i-th position is 1 if and only if the i-th element of S^0 is a vertex of σ. This allows for the identification of the sets of k-simplices $S^k \setminus S^{k-1}$ in the complex with subspaces $\mathcal{S}_k$ of Hilbert spaces $\mathcal{H}_k$ spanned by all possible k-simplices represented in the standard computational basis. Extending this further, a relation:

$$S \sim \mathcal{H} = \bigoplus_{i=1}^{n} \mathcal{H}_i \tag{5.19}$$

between the full simplicial complex S and a subspace $\mathcal{S}$ of the Hilbert space $\mathcal{H}$ is found. Note that boundary operators on this space can be formulated through the use of projections:

$$P_k : \mathcal{H} \to \mathcal{H}_k, \tag{5.20}$$

and simplex boundaries can therefore be identified as elements of this space.

The above identifications grant a domain for the simplicial complex, as the component spaces of $\mathcal{H}$ are spanned by all possible simplices. If the simplicial complex is not yet explicitly determined, the last remaining step is its approximation, and the complex is manifested specifically as the uniform

superposition $|s\rangle \in \mathcal{H}$ of the member simplices of S. Once a membership oracle for the simplices in the complex is formulated, this state is determined through a straightforward application of Grover's algorithm.

5.3.2 Extraction of the Betti Numbers

In order to find the Betti numbers of a simplicial complex in a quantum state, the Abrams-Lloyd algorithm introduced in [2] and expanded on in [50] is used, an algorithm for the estimation of eigenvalues which, compared to the phase estimation algorithm, only requires an approximation to the eigenvector corresponding to the eigenvalue that is sought. This algorithm is described in generality before it is applied.

Let U be a unitary operator and $\tilde{v}$ an approximation to the eigenvector v of U with eigenvalue λ_v. If $\tilde{v}$ is represented in a quantum state by $|\tilde{v}\rangle$, the initial state to be constructed is $|0\rangle^{\otimes t} \otimes |\tilde{v}\rangle$ with the first register of t qubits serving the same goal as the first register in the phase estimation algorithm. This initial state can then be coaxed into the state:

$$|\psi\rangle = \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle \otimes U^j |\tilde{v}\rangle \tag{5.21}$$

by first applying a Hadamard transform to the first register and then consecutively applying U the required number of times to the second register. Now note that because U is unitary it must be normal as well, and therefore the spectral decomposition theorem (Theorem 2.1.15) applies. As a consequence the state $|\tilde{v}\rangle$ can be written as a linear combination:

$$|\tilde{v}\rangle = \sum_k c_k |v_k\rangle \tag{5.22}$$

for suitable complex coefficients c_k and where the v_k form a basis of eigenvectors of U. Because the eigenvalues corresponding to the eigenvectors v_k can be written in the form $e^{2\pi i \varphi_k}$ for values $\varphi_k \in [0, 1)$, the state $|\psi\rangle$ can then be rewritten as:

$$\begin{aligned}
|\psi\rangle &= \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle \otimes U^j \left(\sum_k c_k |v_k\rangle \right) \\
&= \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle \otimes \left(\sum_k c_k e^{2\pi i \varphi_k j} |v_k\rangle \right) \\
&= \frac{1}{\sqrt{2^t}} \sum_k c_k \sum_{j=0}^{2^t-1} |j\rangle \otimes e^{2\pi i \varphi_k j} |v_k\rangle \\
&= \frac{1}{\sqrt{2^t}} \sum_k c_k |v_k\rangle \otimes \left(\sum_{j=0}^{2^t-1} e^{2\pi i \varphi_k j} |j\rangle \right).
\end{aligned} \tag{5.23}$$

The reader should note that the registers are swapped in the last step. After applying an inverse quantum Fourier transform, a subsequent measurement yields one of the eigenvalues. This feature of the algorithm is what is used to approximate the Betti numbers of the simplicial complex S.

As a first step, the boundary operator ∂_k is embedded into a different operator:

$$B_k = \begin{pmatrix} 0 & \partial_k \\ \partial_k^\dagger & 0 \end{pmatrix} \tag{5.24}$$

acting on $\mathcal{S}_{k-1} \oplus \mathcal{S}_k$ that can be verified as being Hermitian as well as a Hamiltonian following the Dirac formalism [27]. The consequence of this embedding is that the Abrams-Lloyd algorithm can now be used to estimate the dimension of the kernel of ∂_k, as it corresponds to the kernel of B_k, and this in turn can be used to compute the estimated Betti numbers. In order to do this, an important map called the matrix exponential needs to be introduced.

Without concern for the domain, codomain and the matter of convergence, the matrix exponential is an extension of the exponential function e^x to matrices. Recall that the exponential function can be expanded as:

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} \tag{5.25}$$

for x a scalar. Since this expansion involves only products and sums, which are also allowable operations for matrices, the exponential e^X of a matrix X is defined identically.

It is important to note that the matrix exponential e^{-iHt} of any Hamiltonian H is unitary [70]. Moreover, all eigenvalues of an Hermitian operator must be real [8]. Lastly, it can be verified from the definition that every eigenvalue of a matrix exponential must be of the form e^λ where λ is an eigenvalue of the exponent. Put together, these observations show that when the Abrams-Lloyd algorithm is applied to e^{-iB_k}, the fraction of times the eigenvalue $e^0 = 1$ is measured is an indication for how often the eigenvalue 0 occurs in the spectrum of ∂_k. More precisely, the algorithm is applied using the uniform mixture of the $(k-1)$-simplices and k-simplices in S as $|\tilde{v}\rangle$, with [59] representing this mixture as a density operator [70]. This representation is not treated here.

Let p_k be the observed probability of measuring the eigenvalue 1. As per the deriviation of the k-th Betti number as a codimension in Subsection 5.3.2:

$$\beta_k = \dim \ker \partial_k - \dim \operatorname{im} \partial_{k+1}. \tag{5.26}$$

Since in this subsection ∂_{k+1} is an operator acting on $\mathcal{S}_{k+1}$, it is also a given that

$$\dim \operatorname{im} \partial_{k+1} = \dim \mathcal{S}_{k+1} - \dim \ker \partial_{k+1} \tag{5.27}$$

as a consequence of the rank-nullity theorem [8]. Using $p_k \cdot \dim \mathcal{S}_k \approx \dim \ker \partial_k$ the following estimation $\tilde{\beta}_k$ of the k-th Betti number is obtained:

$$\begin{aligned} \tilde{\beta}_k &= \dim \ker \partial_k + \dim \ker \partial_{k+1} - \dim \mathcal{S}_{k+1} \\ &\approx p_k \cdot \dim \mathcal{S}_k + p_{k+1} \cdot \dim \mathcal{S}_{k+1} - \dim \mathcal{S}_{k+1} \\ &\approx p_k \cdot \dim \mathcal{S}_k + (p_{k+1} - 1) \cdot \dim \mathcal{S}_{k+1}. \end{aligned} \tag{5.28}$$

The time complexity of this process is derived in [59] as being $\mathcal{O}(\xi n^5)$ for ξ a multiplicative factor depending on the desired accuracy, simplices in the complex and the dimensions of the eigenspaces. We, however, are of the opinion that this analysis is lacking, as for instance the assumptions made during the complexity analysis of the Abrams-Lloyd algorithm are not properly addressed. For one, [2] formulates a requirement on the value of $|\langle \tilde{v}, v \rangle|^2$ which is not shown to be met in [59].

Unfortunately this procedure can not be used to compute the persistent Betti numbers of a filtration. The authors of [59] have erroneously assumed the persistent Betti numbers can be calculated from the Betti numbers, which is not the case. The obstacles are discussed after the next subsection that introduces the combinatorial Laplacians, which are required to continue the discussion of the ideas put forward in [59].

5.3.3 Combinatorial Laplacians

Like simplicial complexes subsume graphs (as the latter are one-dimensional simplicial complexes), so too do the combinatorial Laplacians defined in this subsection subsume the graph Laplacian. A combinatorial Laplacian is much like a graph Laplacian except that it's defined more generally on a simplicial complex. It holds the same type of information and serves the same general purpose.

The definition of the combinatorial Laplacians for a simplicial complex requires knowledge of an associated chain complex of vector spaces and an inner product to be defined on these spaces. Using this structure an adjoint operator $\partial_k^\dagger : \Delta_{k-1} \to \Delta_k$ to each boundary map $\partial_k : \Delta_k \to \Delta_{k-1}$ can be found. It is worth noting that these adjoints are strongly associated to coboundary operators used for the definition of cohomology through cochain complexes [45]. Cohomology is another major construction used heavily in algebraic topology and differential geometry, and there exists a parallel to persistent homology called persistent cohomology [24]. Returning from this small detour, the following definition is in order.

Definition 5.3.1. (Combinatorial Laplacians). Let S be a simplicial complex, and let ∂_i denote the boundary operators of the chain complex associated to S. The k-th combinatorial Laplacian $\mathcal{L}_k^c : \Delta_k \to \Delta_k$ of S is defined as the linear operator:

$$\mathcal{L}_k^c = \partial_{k+1} \circ \partial_{k+1}^\dagger + \partial_k^\dagger \circ \partial_k. \tag{5.29}$$

This definition takes a drastically differing form from the earlier definition of the graph Laplacian, and this is due to the more complex structures present in simplicial complexes. Note that if the boundary maps are realised as their standard matrix representations, the resulting combinatorial Laplacians will be realised as matrices as well. This can be demonstrated while simultaneously sketching how the graph Laplacian fits into this definition using straightforward binary weights.

Let G be any undirected graph with degree matrix D and binary weight matrix W. Starting with the graph G as a template, construct a new graph G' with any orientation on its edges so that G' becomes a directed graph. The graph G' will be viewed as both a graph and a simplicial complex, and the imposition of the orientations guarantees the orderings required for the boundary maps. As a graph, G' admits an incidence matrix B, and as a simplicial complex it theoretically grants two boundary maps ∂_1 and ∂_0, the latter of which is trivial per definition. It follows from the definition of the incidence matrix that B^{T} coincides with the standard matrix representation of ∂_1 [40], and that its adjoint equals its transpose because it can not contain complex entries. Finally, since ∂_0 is trivial, its adjoint must be trivial too. Therefore the 0-th combinatorial Laplacian of G' is given by:

$$\mathcal{L}_0^c = \partial_1 \circ \partial_1^\dagger + \partial_0^\dagger \circ \partial_0 = B^{\mathrm{T}} \circ (B^{\mathrm{T}})^{\mathrm{T}} + 0 \circ 0 = B^{\mathrm{T}} B. \tag{5.30}$$

For any suitable graph G, this computation reveals that $\mathcal{L}_0^c$ so obtained is a matrix with the degrees of the nodes of G on its diagonal, and that the values outside of the diagonal correspond to the negated entries of the weight matrix. Therefore, $\mathcal{L}_0^c = D - W = \mathcal{L}$ for such a graph G. A proof of the proposition that $B^{\mathrm{T}}B$ equals $D - W$ is found in [36].

Because the combinatorial Laplacians are defined in terms of boundary maps and their adjoints, there is an a priori connection with homology and cohomology. It follows from Hodge theory that this connection is actually fairly strong. Hodge theory concerns itself with the representation of de Rham cohomology [57] on certain manifolds by harmonic forms, solutions to specific partial differential equations. Mathematically educated readers might realise at this point that the Laplacians can be seen as discretizations of the differential Laplace operator, the fact of which was already hinted at during the introduction of the graph Laplacians, and that the discrete harmonic forms are then harmonic cycles. The combinatorial treatment of these discrete objects through the lens of Hodge theory was simply dubbed combinatorial Hodge theory, and the following is one of this area's main results.

Theorem 5.3.2. *(Combinatorial Hodge Theorem). Let S be a finite simplicial complex, and let $\mathcal{L}_k^c : \Delta_k \to \Delta_k$ denote the k-th combinatorial Laplacian of the chain complex associated to S for a field $\mathbb{F}$ of characteristic zero. The vector space Δ_k then decomposes as:*

$$\Delta_k \cong \operatorname{im} \partial_{k+1} \oplus \ker \mathcal{L}_k^c \oplus \operatorname{im} \partial_k^\dagger. \tag{5.31}$$

This theorem along with references for historical accounts can be found in [31], while [65] offers a longer and physically motivated explanation of why the discretization of the Laplace operator might be useful. Of the highest conceptual relevance to this thesis is [35], that uses this theorem to construct an algorithm that computes the Betti numbers from the combinatorial Laplacians. The next paragraph sketches how this theorem is related to homology, and therefore the Betti numbers, completing the relation between homology and the combinatorial Laplacians.

First, assuming V, W to be finite dimensional inner product spaces and $T : V \to W$ a linear map between them, recall the following two facts from linear algebra.

1. The image $\operatorname{im} T^\dagger$ of the adjoint of T equals the orthogonal complement $(\ker T)^\perp$ of its kernel;
2. For any linear subspace U of V, $V \cong U \oplus (V/U)$.

It can then be derived from the above theorem that under the given conditions, the kernel of the k-th combinatorial Laplacian and the k-th homology space $H_k(S; \mathbb{F})$ are isomorphic as vector spaces. In order to see this, compare the result of the following derivation that uses the above facts to the decomposition given in Theorem 5.3.2:

$$\begin{aligned} \Delta_k &= \ker \partial_k \oplus (\ker \partial_k)^\perp \\ &= \ker \partial_k \oplus \operatorname{im} \partial_k^\dagger \\ &\cong \operatorname{im} \partial_{k+1} \oplus (\ker \partial_k / \operatorname{im} \partial_{k+1}) \oplus \operatorname{im} \partial_k^\dagger \\ &= \operatorname{im} \partial_{k+1} \oplus H_k(S; \mathbb{F}) \oplus \operatorname{im} \partial_k^\dagger, \end{aligned} \tag{5.32}$$

so that it must follow that $\ker \mathcal{L}_k^c \cong H_k(S; \mathbb{F})$.

5.3.4 Towards Persistent Betti Numbers

First, a proposition is presented that shows that the persistent Betti numbers can not generally be constructed from the Betti numbers. Following this, the observations involving the combinatorial Laplacians and persistent homology made in [59] are treated.

Proposition 5.3.3. *The Betti numbers of the simplicial complexes in a filtration do not necessarily provide enough information to uniquely compute the persistent Betti numbers of the filtration.*

Proof. This statement is proven constructively by way of an example. Consider the following point cloud with the Euclidean metric on $\mathbb{R}^2$:

$$X = \{(0,0),(0,1),(1,0),(1,1),(1+\sqrt{2},0),(1+\sqrt{2},1)\}, \tag{5.33}$$

visualised below. This point cloud will be used to illustrate that the persistent Betti numbers can not be computed from the Betti numbers without additional information.

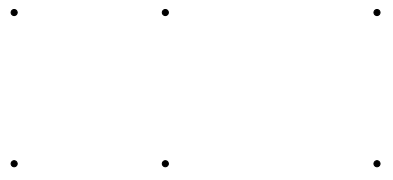


Figure 5.6: The point cloud X.

Using the Vietoris-Rips complex, as this is the complex [59] uses, together with the parameter sequence $\mathbf{s} = \{0, 1, \sqrt{2}\}$, the following behaviour is observed.

Figure 5.7: The complexes VR_1 (left) and $VR_{\sqrt{2}}$ (right) on X.

The first Betti number of VR_1 equals 1, but so too does the first Betti number of $VR_{\sqrt{2}}$ while these values do not stem from the same homology generator. Because for general complexes the Betti numbers in and of themselves contain no additional information on the generators, their persistence can not be derived from just the Betti numbers. There is no way to truthfully distinguish between the two possible values $\{0, 1\}$ for $\beta_1^{1,1}$ while its true value is 0. □

Because [59] made use of the Vietoris-Rips complex it was used as the complex of choice in the proof, but one should note that the Čech complex is identical to the Vietoris-Rips complex if only simplices of dimension 1 or lower are considered. This can be verified by comparing Definitions 5.1.4 and 5.1.8: if only 1-simplices are considered, the common intersection in the definition of the Čech complex is the intersection of a pair of closed balls, the same intersection used in the definition of the Vietoris-Rips complex. This means this result is not limited to just the Vietoris-Rips complex.

Of course the recurrent mention of the homology generators in the proof betrays that if they could be found, the persistent Betti numbers, and more generally the persistent homology, could be computed in full. The following interesting theory was described in [59]. If the direct sum of the operators B_k (5.24):

$$B = \bigoplus_k B_k \tag{5.34}$$

is taken and the result is squared, the resulting operator is the direct sum:

$$B^2 = \bigoplus_k \mathcal{L}_k^c \tag{5.35}$$

of the combinatorial Laplacians belonging to the simplicial complex. Note that for the explicit construction of these operators, the overlap between the domains of the B_k needs to be accounted for, and that terms of the form $\partial_k \partial_{k+1}$ are trivial and therefore vanish. Because $\mathcal{S}$ is a complex Hilbert space and $\mathbb{C}$ is a field of characteristic 0, the combinatorial Hodge theorem applies and for all k the following holds:

$$\ker \mathcal{L}_k^c \cong H_k(\mathcal{S}; \mathbb{C}), \tag{5.36}$$

where $\mathcal{S}$ is treated as a simplicial complex. Therefore, if a way is found to characterise these kernels, the full persistent homology can be computed based on comparison of the generators.

Lloyd et al. do attempt to use (5.34) and (5.36) in [59] to compute all Betti numbers at once by applying the Abrams-Lloyd algorithm to B in the same way as was done for the individual B_k in Subsection 5.3.2. Note that the Abrams-Lloyd algorithm can not be directly applied to B^2, because of the requirement of Hermiticity. There are two problems with this approach. First, the dimension of the kernel of B does not necessarily equal the dimension of the kernel of B^2, so that knowing the former does not grant knowledge of the latter. The following nilpotent matrix is an illustrating example:

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}. \tag{5.37}$$

While the dimension of the kernel of A is 1, the dimension of the kernel of A^2 is 2 as $A^2 = 0$. Second, assuming that through this method $\dim\ker B^2 = \dim\ker \oplus_k \mathcal{L}_k^c$ would be found, this yields the dimension of the kernel of the direct sum of the combinatorial Laplacians. Because knowing the value of a sum does not equate to knowing the value of its components, this dimension does not lead to the individual dimensions of the $\ker \mathcal{L}_k^c$.

Chapter 6

Quantum Clustering

In this chapter the theoretical work on persistent homology treated thus far is related to cluster analysis. This relation will be established by first formulating novel membership oracles required for the construction of the different simplicial complexes. Following this, it is argued that the process of constructing 1-simplices in the Vietoris-Rips and Čech complexes parallels single linkage clustering, and it is shown that part of the barcode can be constructed from just the Betti numbers. This partial barcode is related to the dendrogram. At the tail end, the possibilities for a quantum spectral clustering algorithm within the current framework are briefly discussed.

Over the course of this chapter it will be assumed that distances among datapoints are computed classically and available in a quantum random access memory [38], the quantum counterpart to the random access memory. This memory structure maps address input states to output states as:

$$\sum_k c_k \left|k\right\rangle_a \otimes \left|0\right\rangle_d \;\mapsto\; \sum_k c_k \left|k\right\rangle_a \otimes \left|D_k\right\rangle_d , \tag{6.1}$$

where the image is a superposition of the information D_k in the queried memory cells. The subscripts a and d indicate elements of the address and data registers respectively.

6.1 Simplex Membership Oracles

As the methods discussed previously use Grover's algorithm to construct simplicial complexes in quantum states, they require access to a quantum oracle. In this section several classical oracles that directly amount to quantum oracles (per Definition 3.3.2) for the complexes treated in Section 5.1.2 are given as a proof of concept, with all being based on the inexpensive direct comparison of distances. Simplices are again encoded in bit strings and all given functions have binary output, meaning they qualify as classical oracles.

6.1.1 The Čech Complex

Because the Čech complex depends on the intersection of a number of closed balls it is difficult to construct a membership function for arbitrary metrics. Without explicit knowledge of the metric, a general expression for the classical membership oracle would be of the form:

$$M_\epsilon^{\check{C}}((x_0, ..., x_k)) = \mathbb{1}_{\bigcap_{i=0}^{k} \bar{B}_d(x_i, \epsilon/2) \neq \varnothing}, \tag{6.2}$$

where $(x_0, ..., x_k)$ is a simplex with vertices in the point cloud, and $\mathbb{1}$ is the indicator function. This ensures that the binary function $M_\epsilon^{\check{C}}$ equals 1 if and only if the argument simplex is a member of

the Čech complex at scale ϵ, which in turn is only the case if the intersection in the predicate in Equation 6.2 is non-empty per the definition (5.1.4) of the Čech complex.

As an example, consider the Euclidean metric. An implementation scheme could be a function that computes the volume of a polytope contained in the intersection. The argument simplex would then be a member of the complex if and only if the determined volume is non-zero.

6.1.2 The Vietoris-Rips Complex

Since the Vietoris-Rips complex is a clique complex, the membership oracle is comparatively easier to implement. Recall that for a clique complex, a simplex is a member of the complex if and only if its 1-faces are members of the complex, leading to the following membership oracle:

$$M_\epsilon^{VR}((x_0, ..., x_k)) = \prod_{(i,j), i \neq j} \mathbb{1}_{d(x_i, x_j) \leq \epsilon}, \tag{6.3}$$

where the product ranges over the unordered pairs (i, j) of vertex indices. The product equals 1 exclusively if all vertices are pairwise within range, meaning the intersection of the closed balls at scale ϵ surrounding them is non-empty, and the computation may be terminated as soon as this condition is not met for a single pair to increase efficiency.

6.1.3 The Witness Complex

A supporting function for the Witness complex's membership oracle is given before defining the oracle itself. This function verifies an element x in the point cloud as a witness to a k-simplex:

$$g_\epsilon^W(x, l_{i_0}, ..., l_{i_k}) = \mathbb{1}_{\left[\sum_j \mathbb{1}_{d(x, l_{i_j}) \leq m_x + \epsilon} = k+1\right]}, \tag{6.4}$$

assuming the vertices were taken from the landmark set and this does not require checking. This function determines the number of landmarks l_{i_j} for which $d(x, l_{i_j}) \leq m_x + \epsilon$, and x is a witness to the k-simplex precisely if this number equals $k + 1$, meaning the inequalities hold for all $k + 1$ landmarks making up the simplex. An extension to a membership oracle then takes the form:

$$M_\epsilon^W(l_{i_0}, ..., l_{i_k}) = \mathbb{1}_{\left[\sum_{x \in X} \mathbb{1}_{g_\epsilon^W(x, l_{i_0}, ..., l_{i_k}) > 0}\right]}. \tag{6.5}$$

A simplex is a member of the Witness complex if there exists at least one witness to the simplex, meaning the sum in the predicate must be larger than 0. It is possible that the elements of the point cloud can be sorted by their likeliness to be witnesses, thus decreasing the number of computations, as the routine may be terminated as soon as a single witness is found. Such optimisations are not considered here.

6.1.4 The Alpha Complex

The Alpha complex per definition has a high similarity to the Čech complex, and can even be seen as a substructure of the latter. A simplex can only be a member of the Alpha complex if it would also be a member of the Čech complex. For this reason it suffers from the same drawbacks as the Čech complex does: without an expression for the metric, a membership oracle is difficult to construct.

Assume the Voronoi diagram is available and it is denoted V. Computing V for each separate call to the oracle would produce significant overhead. Given this diagram, the membership oracle for the Alpha complex is then an extension of the oracle for the Čech complex:

$$M_\epsilon^A((x_0,...,x_k)) = \mathbb{1}_{\bigcap_{i=0}^k [V_{m(x_i)} \cap \bar{B}_d(x_i,\epsilon/2)] \neq \varnothing}. \tag{6.6}$$

It is unclear to what extent the dependency on the Voronoi diagram complicates the problem of finding an expression for this oracle. Considering the process naively, some measure of complexity is added, but the Voronoi cells may also serve as reference points to aid in the construction of the aforementioned polytopes.

6.2 Quantum Hierarchical Clustering

Since the preceding section has seen the formulation of membership oracles, Grover's algorithm can be used for the construction of simplicial complexes per the theory in Subsection 5.3.1. In this section it is assumed that a filtration over the point cloud for some parameter sequence $\mathbf{s}$ is available, with each simplicial complex being represented in a separate quantum state. Using these elements the possibility of a quantum algorithm for the task of agglomerative hierachical clustering is explored, starting from linkages.

6.2.1 Simplices as Linkage

Linkages were introduced in Subsection 4.1.1 as constructions to measure the connectivity of two clusters based on a given dissimilarity function. If the dataset is a point cloud and the dissimilarity function is taken to be the metric, the 1-simplices of a Vietoris-Rips or Čech complex on the point cloud parallel single linkage. In order to see this connection, assume that the scale parameter for the construction of the simplicial complexes is allowed to vary continuously over a closed real interval. As the scale increases, 1-simplices are added between the datapoints that are closest according to the metric, meaning that the minimising connected components C_1 and C_2 for:

$$\min_{c_1 \in C_1, c_2 \in C_2} d(c_1, c_2) \tag{6.7}$$

are merged first. This is precisely the formula for single linkage clustering, using the metric as the dissimilarity function, and the connection is completed by considering the parameter sequence $\mathbf{s}$ as a discretization of the interval. This observation is the inspiration for considering the persistent homology of the 1-skelata in a filtration on the point cloud for hierarchical clustering. In doing so, the complexity of the process of constructing quantum states for these complexes should be further reduced, as for general complexes no calls to a quantum oracle needs to be made for the higher dimensional simplices.

Higher dimensional simplices may still serve a purpose within this context. The presence of high dimensional simplices in a cluster may indicate a high confidence in its elements being truly related, as may the number of higher dimensional simplices present. Considering only the zeroth dimensional persistent homology over the filtration has an enticing advantage however, as the persistent Betti numbers in this dimension can be found by the quantum algorithm in [59]. This statement is proven in the next subsection.

6.2.2 Partial Persistence

In particular, the algorithm in [59] can find the persistent homology in the zeroth dimension because the persistent Betti numbers in this dimension equal specific Betti numbers. The following lemma details why this is the case.

Lemma 6.2.1. *In the zeroth dimension, the persistent Betti numbers of a filtration can be computed from the Betti numbers of the individual complexes in the filtration.*

Proof. First, let $\beta_0(\epsilon_i)$ denote the function that maps a scale parameter ϵ_i in $\mathbf{s}$ to the zeroth Betti number of the simplicial complex at that scale. This function is monotonically decreasing on $\mathbf{s}$. For all i, j with $i < j$, the complex at scale ϵ_i is a subset of the complex at scale ϵ_j, meaning that as the scale increases simplices can only be added but never removed. Recall now that the zeroth Betti number of a simplicial complex equals the number of connected components in the complex. When a simplex of any dimension is added in the step from ϵ_i to ϵ_j, it is either added to a connected component, or a connected component disappears because the simplex connects two connected components that were previously disconnected. In either case, the number of connected components does not increase, thus

$$\beta_0(\epsilon_i) \geq \beta_0(\epsilon_j), \tag{6.8}$$

and it may be concluded that $\beta_0(\cdot)$ is monotonically decreasing. Moreover, per the preceding explanation, an increase in scale only ever brings about the death of generators. Therefore, at any scale, $\beta_0(\cdot)$ gives the number of generators that have survived since scale parameter 0, or more generally since the last complex in the filtration. Given that the Betti numbers are known for all scales in the parameter sequence, the zeroth p-persistent Betti number of the i-th simplicial complex in the filtration is then calculated as:

$$\beta_0^{i,p} = \beta_0(\epsilon_{i+p}). \tag{6.9}$$

□

The relevance of this result to hierarchical clustering is that the barcodes and persistence diagrams in the zeroth dimension may be read much in the same way as the dendrogram. Consider the following example of a barcode for the zeroth dimensional persistent homology.

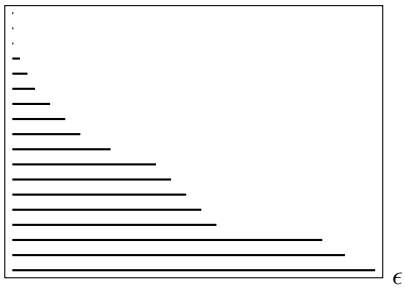


Figure 6.1: An example of a partial barcode.

Note that this barcode can be constructed without requiring specific knowledge of the homology generators. For the trivial scale parameter, the number of connected components equals the number of datapoints, and therefore $\beta_0(0) = |X|$. A bar is started for each of these generators in the barcode. As general procedure, starting from any scale parameter ϵ, the bars for the currently living generators all persist up until the first value ϵ' in the parameter sequence for which $\beta_0(\epsilon')$ differs from $\beta_0(\epsilon)$. At this point the extension of the highest $\beta_0(\epsilon') - \beta_0(\epsilon)$ still mutating bars is stopped as their corresponding generators have died.

Similarly to how the dendrogram is read in order to choose the number of clusters, the partial barcode can be used to choose a number of clusters based on the longetivity of the generators. For instance, assuming for this paragraph that the parameter sequence used for the construction of the above partial barcode is uniformly spaced, the third generator from the bottom seems to have the longest solitary lifespan. This could be the basis for a choice of the scale parameter at the end

point of this bar. The main difference with the dendrogram is then that the complete clustering, or equivalently the partition, can not be read directly from the barcode. Instead, the scale parameter points towards a single simplicial complex in a quantum state from which the clusters need to be extracted. This point is addressed in the following subsection.

6.2.3 Obtaining the Partition

Once a scale parameter ϵ is chosen, either automatically through some pre-defined selection rule or through visual inspection of the barcode by the analyst, the partitioning of the point cloud needs to be extracted from the corresponding simplicial complex. Several possible routines are discussed in this subsection. Since the premise is finding the connected components of a simplicial complex S, it will be useful to consider its 1-skeleton S^1 as an undirected graph $(S^0, S^1 \setminus S^0)$ so that graph theoretical methods become available.

Classically, one would use the depth-first search algorithm to find the connected components of an undirected graph. At the time of writing and to our knowledge however, this algorithm has no quantum counterpart. The first option offered is therefore to reconstruct the state containing the simplicial complex through quantum state tomography [70, 73], after which classical methods like the depth-first search become available.

Staying within the confines of quantum computing, it is likely this task can be reformulated based on the undirected *st*-connectivity problem. This problem is, given an undirected graph and two vertices s and t in it, to determine if there is a path connecting the vertices s and t. A quantum algorithm to solve this problem is given in [10], along with the following theorem on its complexity.

Theorem 6.2.2. *Assume the adjacency matrix of an undirected graph with n vertices is given, and that if two vertices s and t are connected by a path, this path is of length at most m. The st-connectivity problem can then be decided in $\mathcal{O}(n\sqrt{m})$ quantum queries.*

More work in this direction was done in [52], in the form of an extension of the complexity analysis and the introduction of new quantum algorithms for the determination of graph connectivity.

Another option could be the use of quantum walks [75], the quantum analogue to the random walks from stochastics. Random walks are stochastic processes that model paths, usually studied in the setting of an integer lattice, where each step to a different position in the lattice is taken with some probability. Interestingly, these quantum walks have been shown to be possible primitives for universal computation [19], but more importantly the possibility of using them for searching problems has been studied [63,78]. They become explicitly connected to the problem at hand through [3], in which quantum walks on graphs are considered.

The hidden subgroup problem was mentioned previously in the chapter on the quantum Fourier transform, as the latter is instrumental to solving the hidden subgroup problem for abelian groups on a quantum computer [70]. The problem is revisited here as a possible basis to reformulate the problem of finding the connected components on. The following definition is needed for a precise definition of the general hidden subgroup problem.

Definition 6.2.3. (Subgroup Hiding Function). Let G be a group and Y a finite set. A subgroup $H \leq G$ is said to be hidden by a function $f : G \to Y$ if for all $g_1, g_2 \in G$ the following holds:

$$f(g_1) = f(g_2) \Leftrightarrow g_1 H = g_2 H. \tag{6.10}$$

The problem is then as follows: if G is a finitely generated group, Y is a finite set and f is a function hiding a subgroup H of G, find a generating set for the subgroup H. This problem is the general

case of many problems of interest, such as the problem of order finding, computation of discrete logarithms, and even bears relation to the graph isomorphism problem [53]. The mention of the graph isomorphism problem should indicate that it is not unprecedented to involve the hidden subgroup problem in the analysis of graphs.

How the hidden subgroup problem can be utilised to find the connected components is an open problem. In order to recast the latter problem as an instance of the former, the group G, the set Y and a function f would need to be chosen. A lead towards this structure may be found in the idea of the fundamental group from topology [45], that studies the closed paths in a pointed topological space up to homotopy. Pointed, in this case, refers to a singled out point in the space that serves as a basepoint for the paths. However, it should be added that the fundamental group, and the homotopy groups in general, are notoriously hard to compute. Moreover the fundamental group is generally not abelian. As an addition, one that is extremely likely to be in excess, inspiration may be drawn from the fundamental groupoid [14] as well, which removes the need for a choice of basepoint.

Finally, possibly in supplement to the ideas from the previous paragraph, the linear algebraic study of graphs may prove to help connect the problem of finding the connected components to the hidden subgroup problem. In particular the theory of vertex spaces, edge spaces and cut spaces treated in [25] may be of help. In this formalism, vector spaces are formed from the vertices and edges of a graph $G = (V, E)$. The vertex space $\mathcal{V}$ is constructed as the vector space of all functions $V \to \mathbb{F}_2$ over $\mathbb{F}_2$, which grants a natural correspondence between subsets of the vertices and (characteristic) functions in $\mathcal{V}$ that are constant equal to 1 on these subsets. Under this translation the sum $x + y$ of two elements x, y from this space is defined by the symmetric difference:

$$x + y := x \,\Delta\, y = (x \setminus y) \cup (y \setminus x)\,, \tag{6.11}$$

and the empty set $\varnothing$ is taken to be the zero element. The edge space $\mathcal{E}$ is constructed following the same formula, and from this starting point several familiar graph theoretical notions such as cycles and cuts can be transported to a linear algebraic setting. Because there is a natural inner product:

$$\langle \cdot, \cdot \rangle : \mathcal{E} \times \mathcal{E} \to \mathbb{F}_2,\ (F, F') \mapsto \sum_{e \in E} F(e)F'(e) \tag{6.12}$$

on the edge space one can even speak of orthogonality of the edges, leading to several interesting general results [25].

6.2.4 Complexity

If the complexity analysis of Lloyd et al. in [59] holds, the computational complexity of finding the Betti numbers should take $\mathcal{O}(\xi n^5)$ time, where n is the size of the point cloud. It is highly likely that this complexity can be further reduced if only the zeroth Betti numbers of the complexes are computed. Redoing this analysis however would be difficult, since it is unclear how it should be confirmed that the requirements for the Abrams-Lloyd algorithm are met, and how the issues of convergence should be addressed. With the zeroth Betti numbers found, the construction of the partial barcode takes linear time, as it only requires one iteration over the Betti numbers. This leaves only the complexity of finding the partition from the chosen simplicial complex.

The complexity of the first option, quantum state tomography, is hard to determine, as it depends both on what is considered to constitute a measurement and how the computational complexity of a physical measurement operation is determined. Even if these problems are addressed, the problem of dependency on the probability distribution corresponding to the quantum state remains. Finding the connected components by solving the *st*-connectivity problem or hidden subgroup problem

however does lead to explicit time complexities.

Assume for both that the problem reduction does not exceed the time required to solve the problem. In the case of the *st*-connectivity problem, Theorem 6.2.2 then guarantees that the connected components can be found in at most $\mathcal{O}(n^{7/2})$ time by setting $m := n$ and checking every unordered pair of vertices. The hidden subgroup problem, in turn, can be solved by an algorithm with running time polynomial in the logarithm of the group size [53]. If the group can be constructed with the vertices as it elements, the resulting complexity would be $\mathcal{O}(\text{poly} \log n)$.

Since single linkage hierarchical clustering can be implemented classically with a running time of $\mathcal{O}(n^2)$, formulating a quantum algorithm as is done in this chapter may at first seem like a fruitless endeavor. The use of simplices, however, may offer additional information useful for the qualitative evaluation of the clusterings. This idea is touched upon later in the discussion of future directions.

6.3 Quantum Spectral Clustering

In this closing section some interesting connections between spectral clustering and the material in this thesis are mentioned. Specifically, having to compute the spectrum of the graph Laplacian is one of the major drawbacks of spectral clustering. Having access to the combinatorial Laplacians, the theory in [59] and [2] may ease this, as an approximation to the most important part of the spectrum is obtained in the routine for the computation of the Betti numbers.

6.3.1 Finding the Eigenvectors

One of the interesting aspects of the Abrams-Lloyd algorithm is that once an eigenvalue is measured, part of the state collapses onto one of the eigenvectors corresponding to this eigenvalue. If this or any other routine can be adapted to find the eigenvalues and eigenvectors of the zeroth combinatorial Laplacian, this process covers the first four out of five steps for the unnormalised spectral clustering algorithm. Note that this is a direct result of the zeroth combinatorial Laplacian equating to the unnormalised graph Laplacian, and that the obstruction in applying the Abrams-Lloyd is that this operator is in general neither unitary nor Hermitian. The next result will show that the normalised graph Laplacian of an undirected graph without isolated vertices is in fact Hermitian and, although more difficult to construct, is therefore also a candidate for use in a quantum spectral clustering algorithm.

Proposition 6.3.1. *The normalised graph Laplacian $\mathcal{L}^*$ of an undirected graph without isolated vertices is Hermitian.*

Proof. Let D denote the degree matrix of the graph, W its weighted adjacency matrix, and $\mathcal{L}$ the graph's unnormalised graph Laplacian. Recall that in order to show that $\mathcal{L}^*$ is Hermitian, it needs to be demonstrated that it is self-adjoint:

$$(\mathcal{L}^*)^\dagger = \left(D^{-1/2}\mathcal{L}D^{-1/2}\right)^\dagger = \mathcal{L}^*. \tag{6.13}$$

This will be done by first showing that $D^{-1/2}$ and $\mathcal{L}$ are symmetric, from which the above can be derived. First, note that the degree matrix D is normal following its definition in Subsection 4.3.1. As such, exponents of D can be taken through Equation 2.7, and if D is written as $\text{diag}(d_1, ..., d_n)$ then:

$$D^{-1/2} = \text{diag}(d_1^{-1/2}, ..., d_n^{-1/2}). \tag{6.14}$$

Because all entries in D are necessarily real, non-zero and positive, $D^{-1/2}$ is again a symmetric real matrix. This step is the cause for the requirement of the graph not having isolated vertices: if a vertex i without any incident edges can exist in the graph, this vertex would lead to an entry $d_i = 0$.

Next, recall that a weight matrix of a non-negatively weighted graph too can only have positive real entries, and that they represent the weights of the edges in the graph. Since the weight matrix W is based on an undirected graph, it must follow that W is symmetric. The graph Laplacian $\mathcal{L}$ can then be shown to be symmetric as:

$$\begin{aligned} \mathcal{L}^{\mathrm{T}} &= (D - W)^{\mathrm{T}} \\ &= D^{\mathrm{T}} - W^{\mathrm{T}} \\ &= D - W \\ &= \mathcal{L}, \end{aligned} \tag{6.15}$$

and recollecting the above facts it can be deduced that $\mathcal{L}^*$ is Hermitian:

$$\begin{aligned} (\mathcal{L}^*)^\dagger &= \left(D^{-1/2}\mathcal{L}D^{-1/2}\right)^\dagger \\ &= \left(D^{-1/2}\mathcal{L}D^{-1/2}\right)^{\mathrm{T}} \\ &= \left(D^{-1/2}\right)^{\mathrm{T}} \mathcal{L}^{\mathrm{T}} \left(D^{-1/2}\right)^{\mathrm{T}} \\ &= D^{-1/2}\mathcal{L}D^{-1/2} \\ &= \mathcal{L}^*. \end{aligned} \tag{6.16}$$

As a product of real matrices, $\mathcal{L}^*$ must be real as well, and therefore its adjoint must be its transpose. Next, the distributive property of the transpose is used to distribute the transpose over the (reversed) components, and finally it is used that all three components are symmetric, completing the derivation. □

Relying on the normalised graph Laplacian, and therefore making use of the normalised spectral clustering algorithm, removes the problem of not being Hermitian that the unnormalised graph Laplacian did face. The result is that the Abrams-Lloyd algorithm can be applied to approximate the spectrum of this matrix, but although the algorithm then obtains eigenvectors of its matrix exponential, it is unclear whether or not these can be uniquely related to its eigenvectors. In addition this approach does not allow the presence of isolated points.

Other ideas for obtaining the information required for spectral clustering may follow from distantly related work, like the quantum algorithm for principal component analysis [60]. At this junction it suffices to conclude that if the spectrum and the eigenvectors can be found, the spectral clustering algorithms are completed by a quantum variant of the k-Means algorithm.

6.3.2 Quantum k-Means Algorithms

This subsection offers ideas for the final step of a quantum spectral clustering algorithm that were not explored further during the research for this thesis. Since the classical spectral clustering algorithms in Subsection 4.3.2 require use of the k-Means algorithm, a full quantum variant is likely to require a quantum counterpart to k-Means clustering. Therefore, these ideas are presented in the form of possible connections between the k-Means algorithm and known quantum algorithms.

An interesting quantum routine that was published in [89] computes the Euclidean distances of a vector to a set of cluster centroids, and may be used to accelerate the k-Means algorithm hybridly. This use was described in the same paper. Previously mentioned, the quantum principal component analysis algorithm [60] may indirectly be related to the k-Means algorithm, specifically through the work of [26] that explores the classical relation between principal component analysis and the k-Means problem by viewing the latter as a sparse variation of the former. It is worthy of note that this paper was met with criticism based on supposedly unclear phrasing, potentially placing the results in contention. Lastly, [29] considers a continuous relaxation of the normally discrete k-Means problem and shows this instance of the problem can be solved through a singular value decomposition, again offering a possible new site for future research.

If any of these ideas can be united, the result will likely prove to be a large step towards a full quantum k-Means algorithm, and therefore towards a quantum spectral clustering algorithm as well.

Chapter 7

Discussion

7.1 General Discussion

The general discussion consists first and foremost of a reflection on the work put forward in this thesis. The contributions made are summarised in the first subsection, while also discussing them in relation to each other by shedding light on their commonalities. These contributions themselves stem from, and add to, a selection of open problems. A selection is given in the second subsection.

7.1.1 Contributions

As a first contribution, it was proven by way of counterexample that the method presented in [59] is not able to compute the persistent homology of a general filtration. This counterexample is based on the fact that the birth and death of homology generators are not exclusive events, meaning that moving from one complex to a subsequent complex in a filtration, a birth and death may coincide. If this occurs, the Betti numbers of the complexes alone are insufficient to discern the persistence, as the persistence of the generators is obfuscated. Since the method offers no insight into the generators bringing forth the Betti numbers, this deficiency is irremediable.

It was later shown as reconciliation that this statement excludes persistence in the zeroth dimension. As the complexes in a filtration are increasing, each being contained within the next, births of homology generators are of no concern in this dimension. By formulating a function on the parameter sequence defining the filtration, it was shown that the zeroth Betti numbers of the complexes can at most decrease when varying the scale parameter upwards. In turn this means that the persistent Betti numbers in the zeroth dimension equal specific Betti numbers, and can thus be found by the proposed quantum algorithm in [59]. The implementation in [48] can be seen as a partial proof of concept for this theory along with the theory in Chapter 6.

Because persistent homology is a tool used most commonly in data analysis, it is inherently connected to machine learning. Since homology in the zeroth dimension is concerned with connectivity, the most natural point of entry to this domain is through cluster analysis. Specifically, the observation that the barcodes and dendrograms used in hierarchical clustering bear resemblance leads to a connection with agglomerative hierarchical clustering. This thread is followed, offering quantum routines to each classical step in the classical algorithm, until the selection of a complete clustering. Once the simplicial complex containing these clusters as its connected components is chosen, several options and ideas were offered for their extraction.

Further relations to clustering were explored by pairing the zeroth combinatorial Laplacian of a simplicial complex, previously investigated in relation to homology [31], with the graph Laplacians used in spectral clustering. By considering the applicability of the Abrams-Lloyd algorithm [2] to both the normalised and unnormalised graph Laplacians, showing the former to be Hermitian, two starting points for quantum spectral clustering algorithms were mooted. The capstone to these ideas would be a quantum algorithm for k-Means clustering, for which several possible venues of future research were offered.

The research done in the process of writing this thesis has consequently lead to the identification of several open problems. Some were mentioned in the earlier chapters, and these will be revisited along with a new addition in the next subsection.

7.1.2 Open Problems

By far the largest open problem is the completion of a quantum algorithm for the computation of persistent homology and its complexity analysis. Although the methods discussed so far were very ingeniously found by the original authors, it is not clear if the problems they still face can be circumnavigated. If not, a different approach, possibly more algebraic in flavour, may need to be explored.

A different problem that remains open is the extraction of the clustering from the quantum state resulting from the quantum routines paralleling the agglomerative hierarchical clustering algorithm (Subsection 6.2.3). The use of quantum state tomography is possible, but it would be preferable to have a full quantum algorithm that returns the clusters in some form. It is to this end that the st-connectivity problem, quantum walks, and the hidden subgroup problem were explored.

Of a different nature is the practical consideration of chance. As most quantum algorithms are probabilistic, the effects of approximating a full filtration of linked simplicial complexes should be analysed statistically. Simulations would constitute a large part of this analysis. No implementations were realised as part of this thesis in favor of a conceptual approach.

7.2 Future Directions

Having listed the contributions made and their effect on the open problems left, this following section offers ideas and directions for future research. These ideas are segmented into two classes. One, corresponding to the first subsection, is a collection of possible extensions of the work presented. The second, corresponding to the second subsection by process of elimination, treats problems that do not exhibit such a tight relationship to the preceding chapters.

7.2.1 Additions and Extensions

Proposition 4.3.4, found in the section on spectral clustering, shows that the graph Laplacians have several properties that are intimately tied to the connectivity of the graphs they are based on. Although the higher combinatorial Laplacians serve different purposes, inspired by the combinatorial Hodge theorem one might entertain the possibility that they too conceal geometric information much like the graph Laplacians do. As discretisations of the Laplace operator they almost surely do, but to what extent this information relates to clustering is unknown. On the topic of dimensions, the higher dimensional simplices were mostly neglected in clustering applications. It might be interesting to study how these simplices fit into the picture of cluster analysis, by for instance looking at the percentage of higher dimensional simplices in a connected component as a measure of subjective accuracy.

7.2.2 Related Topics

The inspiration for naming the following related topics originates mostly from considerations on the theory's practical applications. For one example, simplicial complexes, and therefore persistent homology based clustering algorithms, could potentially be used for geometric anomaly detection. If a vertex remains unconnected to any other vertex for a long stretch in the parameter sequence, or a higher dimensional simplex with this vertex as a face takes a long time to form, this may indicate the vertex in question is an outlier. Similarly, the articulation points in the skelata can be used to add a measure of statistical belief to the strength of a clustering.

On a more fundamental level, related to treating the lower dimensional simplices as vertices and edges, the theory regarding graph states [46] might offer results adjustable to simplicial complexes. Such states are normally employed for different applications, but it is worth noting every graph has an associated graph state regardless.

As an example of a topic that is more involved with recent developments in machine learning, [47] is offered. This paper treats the topological signatures found by persistent homology algorithms as intermediate stops rather than objectives, and besets deep neural networks on these signatures to learn representations suitable to specific learning tasks. Future research should point out what benefits having access to these representations might bring.

Up until this point, all quantum algorithmic theory presented has been based on the translation of classical algorithms to quantum algorithms. This work is therefore concluded by returning to the root of this problem, following this quantization in reverse. Dequantization is the informal name given to the process of translating quantum algorithms with a proven speedup over the best known classical algorithms back to classical algorithms preserving this speed advantage, and the results are classified as quantum-inspired classical algorithms. The most prominent example of such an algorithm is the quantum-inspired algorithm solving the recommendation problem [85], which stood for a long time as a strong example of exponential speedup on a quantum computer. Attempting to dequantify any quantum algorithm, including the ones in this thesis, could illuminate both the classical and quantum playing fields.

References

[1] S. Aaronson. Quantum Machine Learning Algorithms: Read the Fine Print. *Nature Physics*, 11(4):291–296, 2015.

[2] D.S. Abrams and S. Lloyd. Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors. *Physical Review Letters*, 83(24):5162–5165, 1999.

[3] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani. Quantum Walks on Graphs. *Proceedings of the 33th Annual ACM Symposium on Theory of Computing*, pages 50–59, 2001.

[4] E. Aïmeur, G. Brassard, and S. Gambs. Machine Learning in a Quantum World. *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 431–442, 2006.

[5] E. Aïmeur, G. Brassard, and S. Gambs. Quantum Speed-Up for Unsupervised Learning. *Machine Learning*, 90(2):261–287, 2013.

[6] N. Akkiraju, H. Edelsbrunner, M. Facello, P. Fu, E.P. Mücke, and C. Varela. Alpha Shapes: Definition and Software. *Proceedings of the 1st International Computational Geometry Workshop*, 1995.

[7] D. Arthur and S. Vassilvitskii. k-Means++: The Advantages of Careful Seeding. *Proceedings of the 18th Annual ACM SIAM Symposium on Discrete Algorithms*, pages 1027–1035, 2007.

[8] S. Axler. *Linear Algebra Done Right*. Undergraduate Texts in Mathematics. Springer, 1997.

[9] J.S. Bell. On the Einstein Podolsky Rosen Paradox. *American Physical Society Physics*, 1(3):195–200, 1964.

[10] A. Belovs and B.W. Reichardt. Span Programs and Quantum Algorithms for st-Connectivity and Claw Detection. *European Symposium on Algorithms*, pages 193–204, 2012.

[11] R. Bhatia. *Matrix Analysis*. Graduate Texts in Mathematics. Springer, 1997.

[12] J.C. Bowers and P.L. Bowers. A Menger Redux: Embedding Metric Spaces Isometrically in Euclidean Space. *The American Mathematical Monthly*, 124(7):621–636, 2017.

[13] G.E. Bredon. *Topology and Geometry*. Graduate Texts in Mathematics. Springer, 1993.

[14] R. Brown. *Topology and Groupoids*. Createspace, 2006.

[15] P. Bubenik. Statistical Topological Data Analysis using Persistence Landscapes. *Journal of Machine Learning Research*, 16(1):77–102, 2015.

[16] G. Carlsson. Topology and Data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[17] G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian. On the Local Behavior of Spaces of Natural Images. *International Journal of Computer Vision*, 76(1):1–12, 2008.

[18] F. Chazal, L.J. Guibas, S.Y. Oudot, and P. Skraba. Persistence-Based Clustering in Riemannian Manifolds. *Journal of the Association for Computing Machinery*, 60(6):41, 2013.

[19] A.M. Childs. Universal Computation by Quantum Walk. *Physical Review Letters*, 102(18):180501, 2009.

[20] A. Das and B.K. Chakrabarti. Colloquium: Quantum Annealing and Analog Quantum Computation. *Reviews of Modern Physics*, 80:1061–1081, 2008.

[21] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer, 2000.

[22] V. de Silva and G. Carlsson. Topological Estimation Using Witness Complexes. *Proceedings of the 1st Eurographics Conference on Point-Based Graphics*, pages 157–166, 2004.

[23] V. de Silva and R. Ghrist. Coverage in Sensor Networks via Persistent Homology. *Algebraic and Geometric Topology*, 7(1):339–358, 2007.

[24] V. de Silva, D. Morozov, and M. Vejdemo-Johansson. Persistent Cohomology and Circular Coordinates. *Discrete and Computational Geometry*, 45(4):737–759, 2011.

[25] R. Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer-Verlag Berlin Heidelberg, 2017.

[26] C. Ding and X. He. k-Means Clustering via Principal Component Analysis. *Proceedings of the 21st International Conference on Machine Learning*, page 29, 2004.

[27] P.A.M. Dirac. *The Principles of Quantum Mechanics*. Oxford University Press, 1981.

[28] D.P. DiVincenzo. The Physical Implementation of Quantum Computation. *Progress of Physics*, 48:771–783, 2000.

[29] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering Large Graphs via the Singular Value Decomposition. *Machine Learning*, 56:9–33, 2004.

[30] D.S. Dummit and R.M. Foote. *Abstract Algebra*. Wiley, 2003.

[31] A.M. Duval and V. Reiner. Shifted Simplicial Complexes are Laplacian Integral. *Transactions of the American Mathematical Society*, 354(11):4313–4344, 2002.

[32] L.C. Evans. *Partial Differential Equations*. Graduate Studies in Mathematics. American Mathematical Society, 2010.

[33] B. Field and T. Simula. Introduction to Topological Quantum Computation with Non-Abelian Anyons. *Quantum Science and Technology*, 3(4):045004, 2018.

[34] S. Fortune. A Sweepline Algorithm for Voronoi Diagrams. *Algorithmica*, 2(1):153, 1987.

[35] J. Friedman. Computing Betti Numbers via Combinatorial Laplacians. *Algorithmica*, 21(4):331–346, 1998.

[36] J. Gallier. *Discrete Mathematics*. Universitext. Springer-Verlag New York, 2011.

[37] R. Ghrist. *Elementary Applied Topology*. Createspace, 2014.

[38] V. Giovannetti, S. Lloyd, and L. Maccone. Quantum Random Access Memory. *Physical Review Letters*, 100(16):160501, 2008.

[39] J.E. Goodman, J. O'Rourke, and C.D. Tóth. *Handbook of Discrete and Computational Geometry*. Discrete Mathematics and its Applications. Chapman & Hall, 2018.

[40] L.J. Grady and J.R. Polimeni. *Discrete Calculus*. Springer, 2010.

[41] D.J. Griffiths. *Introduction to Quantum Mechanics*. Prentice Hall, 1995.

[42] L.K. Grover. A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 212–219, 1996.

[43] B.C. Hall. *Quantum Theory for Mathematicians*. Graduate Texts in Mathematics. Springer, 2013.

[44] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, 2001.

[45] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2001.

[46] M. Hein, J. Eisert, and H.J. Briegel. Multiparty Entanglement in Graph States. *Physical Review A*, 69(6):062311, 2001.

[47] C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl. Deep Learning with Topological Signatures. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1633–1643, 2017.

[48] H. Huang, X. Wang, P.P. Rohde, Y. Luo, Y. Zhao, C. Liu, L. Li, N. Liu, C. Lu, and J. Pan. Demonstration of Topological Data Analysis on a Quantum Processor. *Optica*, 5(2):193–198, 2018.

[49] U. Islambekov and Y.R. Gel. Unsupervised Space-Time Clustering Using Persistent Homology. *Environmetrics*, page e2539, 2018.

[50] P. Jaksch and A. Papageorgiou. Eigenvector Approximation Leading to Exponential Speedup of Quantum Eigenvalue Calculation. *Physical Review Letters*, 91(25):257902, 2003.

[51] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Texts in Statistics. Springer, 2013.

[52] M. Jarret, S. Jeffery, S. Kimmel, and A. Piedrafita. Quantum Algorithms for Connectivity and Related Problems. *The 26th Annual European Symposium on Algorithms*, (49):1–13, 2018.

[53] R. Jozsa. Quantum Factoring, Discrete Logarithms, and the Hidden Subgroup Problem. *Computing in Science & Engineering*, 3(2):34–43, 2001.

[54] R.M. Karp. Reducibility Among Combinatorial Problems. *Complexity of Computer Computations*, pages 85–103, 1972.

[55] E. Kreyszig. *Introductory Functional Analysis with Applications*. Wiley, 1978.

[56] S. Mac Lane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer, 1978.

[57] J.M. Lee. *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. Springer, 2013.

[58] X. Li. k-Means and k-Medoids. *Encyclopedia of Database Systems*, pages 1588–1589, 2009.

[59] S. Lloyd, S. Garnerone, and P. Zanardi. Quantum Algorithms for Topological and Geometric Analysis of Data. *Nature Communications*, 7:10138, 2016.

[60] S. Lloyd, M. Mohseni, and P. Rebentrost. Quantum Principal Component Analysis. *Nature Physics*, 10(9):631–633, 2014.

[61] S.P. Lloyd. Least Squares Quantization in PCM. *Transactions on Information Theory*, 28(2):129–137, 1982.

[62] P. MacNaughton-Smith, W.T. Williams, M.B. Dale, and L.G. Mockett. Dissimilarity Analysis: A New Technique of Hierarchical Sub-Division. *Nature*, 202(4936):1034, 1964.

[63] F. Magniez, A. Nayak, J. Roland, and M. Santha. Search via Quantum Walk. *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 575–584, 2007.

[64] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The Planar k-Means Problem is NP-Hard. *International Workshop on Algorithms and Computation*, pages 274–285, 2009.

[65] S. Maletić and M. Rajković. Combinatorial Laplacian and Entropy of Simplicial Complexes Associated with Complex Networks. *The European Physics Journal Special Topics*, 212(1):77–97, 2012.

[66] R.H.A.J. Meijer. Topological Data Analysis. https://github.com/QuirijnMeijer/TDA, 2018.

[67] J. Milnor. *Morse Theory*. Annals of Mathematics Studies. Princeton University Press, 1969.

[68] J.R. Munkres. *Topology*. Pearson, 2000.

[69] F. Murtagh and P. Contreras. Algorithms for Hierarchical Clustering: An Overview. *Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[70] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information - 10th Anniversary Edition*. Cambridge University Press, 2010.

[71] R. Ostrovsky, Y. Rabani, L.J. Schulman, and C. Swamy. The Effectiveness of Lloyd-Type Methods for the k-Means Problem. *Journal of the Association for Computing Machinery*, 59(6):28, 2012.

[72] N. Otter, M.A. Porter, U. Tillmann, P. Grindrod, and H.A. Harrington. A Roadmap for the Computation of Persistent Homology. *The European Physics Journal Data Science*, 6(1):17, 2017.

[73] M. Paris and J. Řeháček. *Quantum State Estimation*. Lecture Notes in Physics. Springer-Verlag Berlin Heidelberg, 2004.

[74] C.M.M. Pereira and R.F. de Mello. Persistent Homology for Time Series and Spatial Data Clustering. *Expert Systems with Applications*, 42:6026–6038, 2015.

[75] R. Portugal. *Quantum Walks and Search Algorithms*. Quantum Science and Technology. Springer-Verlag New York, 2013.

[76] E.C. Rowell and Z. Wang. Mathematics of Topological Quantum Computing. *Bulletin of the American Mathematical Society*, 55(2):183–238, 2018.

[77] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 1964.

[78] M. Santha. Quantum Walk Based Search Algorithms. *Theory and Applications of Models of Computation*, pages 31–46, 2008.

[79] E. Schubert, J. Sander, M. Ester, H. Kriegel, and X. Xu. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *Transactions on Database Systems*, 42(3):19, 2017.

[80] M. Schuld, I. Sinayskiy, and F. Petruccione. An Introduction to Quantum Machine Learning. *Contemporary Physics*, 56(2):172–185, 2015.

[81] P.W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[82] R. Sibson. SLINK: An Optimally Efficient Algorithm for the Single-Link Cluster Method. *The Computer Journal*, 16(1):30–34, 1973.

[83] G. Singh, F. Memoli, T. Ishkhanov, G. Sapiro, G. Carlsson, and D.L. Ringach. Topological Analysis of Population Activity in Visual Cortex. *Journal of Vision*, 8(8):1–18, 2008.

[84] D. Sundararajan. *The Discrete Fourier Transform.* World Scientific, 2001.

[85] E. Tang. A Quantum-Inspired Classical Algorithm for Recommendation Systems. *Proceedings of the 51st Annual ACM SIGACT on Theory of Computing*, pages 217–228, 2019.

[86] T. Tao. *Analysis I.* Texts and Readings in Mathematics. Springer, 2006.

[87] G. Teschl. *Ordinary Differential Equations and Dynamical Systems.* Graduate Studies in Mathematics. American Mathematical Society, 2012.

[88] U. von Luxburg. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[89] N. Wiebe, A. Kapoor, and K.M. Svore. Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning. *Quantum Information and Computation*, 15(3):316–356, 2015.

[90] B.A. Wubie, A. Andres, R. Greiner, B. Hoehn, A. Montano-Loza, N. Kneteman, and G. Heo. Cluster Identification via Persistent Homology and Other Clustering Techniques, with Application to Liver Transplant Data. *Research in Computational Topology*, pages 145–177, 2018.

[91] D. Yan, L. Huang, and M.I. Jordan. Fast Approximate Spectral Clustering. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 907–915, 2009.

[92] A. Zomorodian and G. Carlsson. Computing Persistent Homology. *Discrete Computational Geometry*, 33(2):249–274, 2005.