Application of Graph Learning to inverse problems

Master Thesis Preperation

Natural Science Faculty of the University of Basel
Department of Mathematics and Computer Science
Data-Analytics

Examiner: Prof. Dr. Ivan Dokmanić
Supervisor: Dr. Valentin Debarnot

Cédric Mendelin
cedric.mendelin@stud.unibas.ch
2014-469-274

29.11.2021

Table of Contents

1
Introduction

Inverse problems aims at estimating a signal that went through a system, based on the output observation. Machine learning (ML) is a tool to model and solve IP. They are widely used throughout different science directions, such as ML, signal processing, computer vision, natural language processing and many more.
In recent years, Graphs got a lot of attention in ML and are one of the most promising research areas. Graphs are a well suited data structure, simple but with high expressiveness. For some specific scenarios ordinary ML algorithm fail but Graph ML approaches have great success, e.g dimensionality reduction for high-dimensional data. Data can be in a graph structure already, like social networks, or they can be constructed for arbitrary datasets.
Cryo-electron microscopy (cryo-EM), where molecules are imaged in an electron microscope, gained a lot of attention in recent years. Due to ground-breaking improvements regarding hardware and data processing, the field of research has highly improved. In 2017, the pioneers in the field of cryo-EM got the Nobel Prize in Chemistry[1]. Today, using cryo-EM many molecular structures can be displayed with near-atomic resolution. The big challenge with cryo-EM is enormous noise, which makes calculation challenging. During the Master Thesis, the aim is to exploit Graph Learning on the cryo-EM reconstruction problem.

The following report resulted as the Master Thesis Preparation report. During the six weeks project, the aim was to familiarize with the research area, build up some mathematical foundation needed for the Thesis and define the project content as well as a project plan.

The report is structured the following: In chapter 4, the overall foundation for the Master Thesis will be given, focusing on Graph Learning, Graph Denoising, some mathematical methods and definitions as well as an introduction to cryo-EM. Chapter **??** is dedicated to the problem setup and some preliminaries of the problem. Moreover, the base idea of the Master Thesis is defined. Up to this point, the underlying problem has been defined and some related work can be given in chapter 5. To end the report, project plan and work packages are introduced in chapter 6.

[1] https://www.nobelprize.org/prizes/chemistry/2017/press-release/

2
Imaging methods

In the current chapter, imaging methods *computed tomography* and *cryo-electron microscopy* (cryo-EM) will be introduced. Further, their observation model is defined in a mathematically way and their reconstruction is observed. Application of cryo-EM is the major motivation for the Master Thesis, as the problem is not easy to solve due to dealing with enormous noise and other difficulties.

2.1 Computed tomography

Computed tomography is a well established imaging method. Using X-ray source, fan shaped beams are produced which scan the imaging object, resulting in many measurements taken over straight lines [3].

Tomography reconstruction: Tomographic reconstruction[6] is a popular inverse problem. The aim is to reconstruct an imaged object from observed measurements. The reconstruction object can be in two-dimension (2D) or in three-dimension (3D). In the Master Thesis, the focus on computed tomography will be on 2D case, which is called *classical tomography reconstruction.*

2D tomographic reconstruction: Mathematically, the observed measurements can be defined as follows:

$$y_i[j] = R(x, \theta_i, s_j) + \eta_i[j], \text{ with } 1 \leq i \leq N, \text{ and } 1 \leq j \leq M \tag{2.1}$$

where N is number of observations and M the observation dimension. Then, $x \in L^2(\Omega)$ is the original object with $\Omega \subset \mathbb{R}^2$ and L^2 is the Lebesgue space. Further, $y_i \in \mathbb{R}^M$ is the i-th observation with $y_i[j] \in \mathbb{R}$ the j-th element of the observation. $R(\cdot; \theta, s) : L^2(\Omega) \to L^2(\tilde{\Omega}), x \mapsto R(x; \theta, s)$ refers to the Radon Transform with $\tilde{\Omega} \subset \mathbb{R}$, θ as the observation angle from the x-axis and s_j as the sampling point. η refers to noise and is defined as $\eta_i[j] \sim \mathcal{N}(0, \sigma^2)$.

Filter Backprojection: Filter Backprojection [6] is a reconstruction method, typically used in classical tomography reconstruction. It allows to inverse the Radon Transform and enables reconstruction of the original object. The algorithm fails when working with noisy data.

2.2 Cryo-EM

Cryo-EM is another imaging method, that enables the view of molecules in near-atomic resolution. In the Master Thesis, only single-particle cryo-EM[9] is considered, when writing from cryo-EM it always refer to single-particle cryo-EM. During imaging process, molecules are frozen in a thin layer of ice, where they are randomly oriented and positioned. Random orientation and positioning of molecules makes reconstruction challenging but the freezing process allows to observe molecules in a stable state where they are not moving. With an electron microscope, two-dimensional tomographic projection images of the molecules are observed in the ice, which are called *micrograph.* The frozen molecules are fragile and the electron microscope needs to work with very low power (electron dose), resulting in highly noisy images. The resulting signal-to-noise ration (SNR) is typically smaller than 1, which indicates that there is more noise than signal[18]. Further, observed molecules are not equal is the sense that there are some structural varieties.

3D cryo-EM reconstruction: Similar to tomographic reconstruction, cryo-EM reconstruction problem[2] is defined. It can be seen as a 3D reconstruction problem as the original object $x \in L^2(\Omega)$ to be reconstructed is in 3D. To keep the notation from previous section, now $\Omega \subset \mathbb{R}^3$ and $\tilde{\Omega} \subset \mathbb{R}^2$.
Mathematically, the observed measurements can be defined as follows:

$$y_i = \Pi_z(Rot(x;\theta_i)) + \eta_i, \text{ with } 1 \leq i \leq N \tag{2.2}$$

where $\Pi : L^2(\Omega) \to L^2(\tilde{\Omega}), x \mapsto \int x(\cdot,\cdot,z)dz$ is the projection operator and $Rot : L^2(\Omega) \to L^2(\Omega), Rot_\theta(x) = \big((x_1,x_2,x_3) \mapsto x(x_1R^1, x_2R^2, x_3R^3)\big)$ is the rotation operator modelling the rotation during freezing. Further, $\theta_i = [\theta_i^1, \theta_i^2, \theta_i^3]$ where entries $[\theta_i^1, \theta_i^2, \theta_i^3 \in \mathbb{R}$ and $R = [R^1, R^2, R^3] \in SO(3)$. $\eta_i[j,k] \sim \mathcal{N}(0, \sigma^2 I)$ corresponds again to the noise of the observation.
As y_i is not observable directly, discretization is needed:

$$\begin{aligned} y_i &= \left(\Pi_z(Rot(x;\theta_i)) + \eta_i\right)(\Delta), \text{ with } 1 \leq i \leq N \\ y_i[j,k] &= \Pi_z(Rot(x;\theta_i))_{j,k} + \eta_i[j,k], \text{ with } 1 \leq i \leq N \text{ and } 1 \leq j,k \leq M \end{aligned} \tag{2.3}$$

where $\Delta \subset \tilde{\Omega}^{M^2}$ is the sampling grid and M is the first and second dimension of the sampling grid.

Extended formula: The equation 2.2 is a simplified version of the cryo-EM reconstruction problem. First of all, the point spread function (PSF) of the microscope is not taken into account. Moreover, the structural variety is not taken into account, the underlying object x is not the same for every observation but can be seen as a random signal from an unknown distribution defined over all possible molecules structures.

The extended version can be defined as follows:

$$y_i = h_i \circ \Pi_z(Rot(x_i; \theta_i)) + \eta_i, \text{ with } 1 \leq i \leq N \tag{2.4}$$

where h_i is the PSF of the microscope and $\circ$ defines the convolution. During the Master Thesis, the equation **??** is used, not the extended version.

Difference to tomographic reconstruction: The two problems are highly related, but the cryo-EM reconstruct is more challenging. During CT observation, the patient is asked to not move and therefore, the angles of projection are known. Whereas, in cryo-EM this information will be lost during the freezing process. Secondly, the high level of noise makes cryo-EM much more challenging regarding tomographic reconstruction.

2.3 Abstract from

As the tomographic reconstruction and the cryo-EM reconstruction are rather similar, the aim of the Master Thesis will be to design an algorithm, that can be applied in both scenarios. Therefore, a abstract form of the problems will be defined in the following. First of all, a similar notation as before is used, but in a more general way $x \in L^2(\Omega)$ where $\Omega \subset \mathbb{R}^D$ with D as the dimension of the space and $\tilde{\Omega} \subset \mathbb{R}^{D-1}$.

$$y_i = \left(A(x, \theta_i) + \eta_i\right)(\Delta) \tag{2.5}$$

where $y_i \in \tilde{\Omega}^M$ is the observed measurements, M the measurement dimension, $x \in L^2(\Omega)$ our original object, A a non-linear operator $A : L^2(\Omega) \to L^2(\tilde{\Omega}), x \mapsto A(x; \theta)$ and η $\mathcal{N}(O, \sigma^2 I)$ gaussian noise. $\Delta \subset \tilde{Omega}^{M^2}$ is a term for discretization.

Classical tomography reconstruction: For classical tomography parameters are defined defined with $D = 2$ and $\theta \in \mathbb{R}^1$. Further, $A(\cdot)$ is the Radon transform, defined in equation 2.1. A distance measure between measurements can be set up by using the l2-norm $\|y_i - y_j\|$.

Cryo-Em reconstruction: For cryo-EM parameters are defined with $D = 3$ and $\theta \in \mathbb{R}^3$. Further, $A(\cdot)$ can be defined as $\Pi_z\left(Rot(x; \theta)\right)$ where Rot is the 3D rotation and Π_z the tomographic projection.
As measurements are drawn with some random 3D rotation and projection, it can happen that two samples are equivalent up to 2D rotation. Consider a first example y_1, which has no 3D rotation and a second sample y_2 with a rotation only in in x-y plane by 45°. The two samples have a defined in-plane rotation g, such that $gy_1 = y_2$. Therefore, in our distance measure we add this term of in-plan rotation: $min_{g \in SO(1)} \|g * y_i - y_j\|$, which is inspired by the work of [10].

High noise regime: Cryo-EM measurements are highly noisy, which makes reconstruction challenging. There are different ways to reduce noise from measurements, most of them are related to averaging. Averaging need to consider similar measurements and ignore

diverse ones. In the defined abstract model, averaging over paired measurements from θ should be a good averaging model. But how can it be achieved?
One idea would be to measure distances between observation (therefore introduced above). Another way is to find a low-dimensional embedding which maps our measurements y to some θ. When talking from low-dimensional embeddings, there is no way around Graph Learning, which will be introduced in the following chapter.

During the Master Thesis, high-noise regime is the domain of interest. The main practical application is cryo-EM, where an algorithm for denoising is expected to boost quality of the overall 3D-reconstruction. The aim of the Master Thesis is to introduce a denoising algorithm, which is able to work well even on highly noisy data, where cryo-EM is major field of interest.

3
Graph Denoising

3.1 Graph Foundations

A graph is defined as $G = \langle V, E\rangle$, where V is a set of vertices (or nodes) and E is a set of edges (or links). Edges are defined as a set of tuples (i, j), where i and j determine the index of the vertices in the graph.

Edges can be either *directed* or *undirected* and that has to do with the position of the nodes in the edge. In a directed graph, a edge points explicitly from one node to another, which means that edge $(i, j) \neq (j, i)$. In undirected graphs the ordering does not matter and $(i, j) = (j, i)$.

Moreover, edges can have weights, which is a method to define importance to the neighbours of a node. If edges are dealing with weights, we are talking from a *weighted* graph. The *neighbourhood* of a node $\mathcal{N}(\rangle) = \forall x$ is defined as all the adjacent node of i, which means that there is an edge between the nodes.

Adjacency matrix: To do calculations with graphs, it is common to translate graphs in a well suitable mathematically form, which are matrices. The adjacency matrix can be seen as a way of representing graphs as a matrix. The (binary) adjacency matrix of graph G is defined as follows:

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

The matrix A has dimension $\mathbb{R}^{N\times N}$ and the indices of the matrix correspond to the nodes of the graph. If there is an edge between two nodes, the entry in the matrix will be set to 1, otherwise to 0. This leads to an unweighted graph, as the weight of all edges will be 1. When the graph is undirected, the resulting matrix will be symmetric and has a complete set of eigenvalues and eigenvectors. The set of eigenvalues are also called *spectrum* of the graph.

Degree matrix: The degree matrix of G is defined as follows:

$$D_{ij} = \begin{cases} deg(v_i) & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

Where $deg(v_i)$ is the degree of the node, formally the number of incoming edges of node v_i.

Graph Laplacian: The graph Laplacian is a matrix that represents the graph and can be used to find many important properties of the graph, which are not handled here but a good overview can be found by [20, 23]. It is defined as follows:

$$L = D - A \tag{3.3}$$

Cryo-EM reconstruction needs to define algorithms, which works well in high noise regimes. During Master Thesis only on single-particle cryo-EM is considered, so speaking from cryo-EM it refers to single-particle cryo-EM.

3.1.1 Graph Construction

3) General framework for constructing a graph: a) Each vertex is associated to some feature/signal/whatever $x \in \mathcal{X}$, for some space arbitrary space $\mathcal{X}$. b) We construct the graph G_0 using: $d(x_i, x_j) < \tau$, τ is a threshold, or k-NN (defined it here in one line). c) For simplification, and because it covers already most cases, we will work on $\S = R^M$ d) There are some applications, as we will see (or have seen if already introduced), where we have only access to a noisy version of the true signal: $y = x + \eta$, where $y, x \in R^N$, and eta i.i.d follow Gaussian $(0, sigma^2)$. e) Then we defined the noisy graph G as in b)

When data is not available as a graph, it can be constructed from the data. First of all, every element of the dataset can be seen as a node and only the decision of how edges will be constructed is necessary. One popular approach is k-nearest neighbour (KNN) graph construction. The parameter k defines how many edges every node will have at the end. The neighbourhood of node i is defined as $\mathcal{N}_i$ and consists of the nodes, with the k smallest similarity measure.

$$A_{ij} = \begin{cases} 1 & \text{if } j \in \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

Instead of using a fixed parameter k as in KNN approach, one could define a threshold π and connection nodes if the similarity measure is smaller than π. This is another common way to define graphs and results in a graph, where not all nodes have the same amount of edges.

3.2 Graph Denoising

TODO: Not the name in literature

Data acquired by real-world observations are often noisy, which can lead to poor performance on data analysis tasks. Graph constructed by noisy data is called a noisy graph, as it includes the noise from observations.

Graph denoising is the task to reconstruct the original graph from a noisy one.

Denoising in general has often to do with averaging and graphs are a well suited data structure for this task[4].

Noise: A noisy observation is defined as: $y_n = y + \eta$, where η is the observation noise (often assumed to be gaussian distributed) and y the noiseless observation.

Denoising: When we talk from denoising, we want to reconstruct the true observation from a given noisy observation. Reconstruction is done via averaging, that can be performed locally, by the calculus of variations or in the frequency domain[4].

Noisy Graph: For every noisy graph, there exists an original graph $G = \langle V, E \rangle$.
The noisy graph can be defined as follows:

$$\begin{aligned} G_{noisy} &= \langle V, E_{noisy} \rangle, \\ \text{with } E_{noisy} &= E \setminus E^- \cup E^+, \\ E^- &\subseteq E, \\ E^+ \cap E &= \emptyset \end{aligned} \tag{3.5}$$

The noisy graph consists of the same vertices as the original graph. From the original graphs edges, some are removed (denoted by E^-) and some are added (denoted by E^+).
The adjacency matrix of G_{noisy} is denoted by $\bar{A}_{ij}$. The task of graph denoising, can therefore be written as:

$$\bar{A} \xrightarrow[method]{Graph-denoising} \tilde{A} \approx A \tag{3.6}$$

Where $\bar{A}$, $\tilde{A}$, A denotes the adjacency matrix from noisy input graph, denoised graph and original graph respectively.

Connection to link prediction Link prediction is a task in Graph Learning. The idea is to predict existence of a link (edge) between two nodes. The task can be formulated as a missing value estimation task. A model M_p is learned from a given set of observed edges. The model finally maps links to probabilities $M_p : E' \rightarrow [0, 1]$ where E' is the set of potential links.
We define U as the set of all possible vertices of G, therefore $E \subseteq U$. Obviously, graph denoising can be seen as a link prediction problem.
The difference is, that in link prediction a model from a set of observed links is learned $E_{observed} \subseteq E$ and in graph denoising the model is learned from $E_{observed} \subseteq U$.

On could also say that link prediction problems are a subset of graph denoising problems.

Finally, the goal of the master thesis is to produce methods to estimate G_0 based on G. (You can put it in a tcolorbox or something similar to show that this is the most important things to read here). To do so, we will focus on a particular problem where this setting makes sense: cryo-EM. And to solve this problem, we will try to connect with recent machine learning techniques (if you follow the plan that proposed below)

5) Graph Laplacian and machine learning: this is one on the main idea to explore in the thesis. Graph Laplacian is a powerful tool, but relies on the noisy adjacency matrix. Can we learn the adjacency matrix such that the output of the Graph Laplacian is what we expected (link with spectrum folded).
6) Mathematical analysis of the link between Graph neural network and Graph Laplacian, as in the paper ”Simplifying graph convolutional networks”. Another strong idea to explore.

Graph Laplacian

3.3 Manifolds

2.3: use mathematic definitions. To avoid heavy definition that we won't need: Let $\mathcal{M} = \{f(x), f is C^K, f : R^d \to R^N\}$. In this thesis, we will consider C^k dfferentiable d-dimensional manifold defined by $\mathcal{M}$.
Embedding -¿ low dimensional embedding: mapping that goes from a high dimensional space to a low dimensional one. For instance, using the previous definition, an embedding on M is a function that maps $\mathcal{M} \to R^d$.
suggestion on Sections 2.3: 1) Definitions: manifold and then embedding, but special case (not general definition that is heavy and not needed). 2) 2 simple examples: circle (d=1, N=2) and one embedding, sphere (d=2,N=3) or more general d-dimensional sphere $S^d = x \in R^{d+1}, \|x\| = 1$. 3) link with signal processing (paragraph currently called ”Manifold assumption”)

4 Foundation

Before dealing with the problem setup of the Master Thesis itself, in the following chapter, a broad foundation of graphs and Graph Learning will be given. Moreover, some important mathematically concepts and methods will be introduced as well as cryo-EM.

K-hop neighbourhood: The adjacency matrix A has many nice properties, and one is referred to the k-hop neighbourhood. When calculating the k-th power of A, the k-hop neighbourhood of the graph is calculated. The resulting matrix gives the number of walks of length k from one node to another.

Normalization: When starting calculating with matrix A, it is sometimes necessary to normalize. With the degree matrix D and adjacency matrix A, all information for normalization are present. The normalization can be achieved in a row, column or symmetric way:

$$\begin{aligned} A_{row-norm} &= D^{-1}A \\ A_{col-norm} &= AD^{-1} \\ A_{sym} &= D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \end{aligned} \tag{4.1}$$

The normalization of the row and column will normalize the row and column to sum to 1 respectively. The symmetric normalization is well suited for undirected graphs, as it preserve the nice symmetric structure matrices.

Normalized Graph Laplacian: During computation, it is often needed to have a normalized version of the Graph Laplacian:

$$L_{sym} = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}L_{rw} \quad = I - D^{-1}A, \tag{4.2}$$

where L_{sym} is a symmetric normalization and L_{rw} is called a random walk normalization.

4.1 Math Foundation

In the following section, some mathematically concepts and methods will be explained.

4.1.1 Embedding

Mathematically, an embedding $f : X \to Y$ is defined as a structure-preserving mapping from one domain to another.
In graph theory, a graph embedding in the mapping from the graph G to a surface structure Σ.

4.1.2 Manifolds

A manifold is a topological space, where locally Euclidean distances make sense. More formally, a n-dimensional manifold is a topological space where each point has e neighbourhood, that is homeomorphic (mapping which preserves topological properties) to an subset of a n-dimensional Euclidean space.
Some example for a 1-D manifold is a line or a circle. 2-D manifolds can already become pretty complex and are basically any surfaces like planes, sphere but also the torus, Klein bottle or others.

Manifold assumption: The manifold assumption is a popular assumption for high-dimensional datasets. Even if a given dataset is in high-dimension and consists of many features, one can assume, that these data points are samples drawn from a low-dimensional manifold, which embeds the high-dimensional space.
Therefore, if one can approximate the underlying Manifold, one solved the dimensionality reduction as one can embed the data points in the low-dimensional manifold space.
There is a complete area of research devoted to this manifold assumption called Manifold Learning[5].

4.2 Graph Learning

As already mentioned, Graph Learning is a popular research area and got a lot of attention in recent years. It is a new way of applying Machine Learning (ML) with graphs as a data structure and many algorithms emerged from ML.
A lot of real data can be modelled as graphs. The data could have graph structure, like social networks. Or a graph can be artificially constructed with methods like k-nearest neighbours (KNN) or with some other similarity measure.

Graph Learning Tasks: When a graph is available, one can start using Graph Learning algorithms for solving tasks. Popular tasks are node classification or link prediction within a graph. One tries to learn from node and edge features as well as the topology of the graph and tries to map information to a model, which allows prediction or classification.
Another popular task in Graph Learning is community detection, where the aim is to identify cluster of nodes within the input graph.
Further, graphs are highly popular for dimensionality-reduction. In higher dimensions, the euclidean distance is not helpful and therefore, algorithms for reducing the dimensionality, such that euclidean distance make sense again. are needed. Graph algorithms provide

a helpful tool in such scenarios, as ordinary algorithms like principle component analysis (PCA) fail.

Algorithm categories There are many algorithmic approaches, how to exploit graphs.
Graph Deep Learning is a derivation from Deep Learning. Basically, in Graph Deep Learning, Deep Learning algorithms are extended for the usage with graphs. After all, a model or some feature will be learned within a neural network, suitable for working with graphs.
Spectral graph theory[20] deals with learning properties and characteristics of graphs, in regard to the graphs eigenvalues and eigenvectors.
Manifold Learning [5] is a popular approach for dimensionality reduction on graphs. Using the manifold assumption section 4.1.2, an embedding of the graph for lower dimension is calculated, which can preserve most of the information, of the original graph.
Further, *Random Walks* is a concept, which is often used in Graph Learning. It is used to exploit topological information of a graph by randomly ”walking” (use edges to move from on node to another) over the graph. With sampling a lot of these walks, one can infer information about the graphs topology.

5 Related Work

In the following section, related work will be introduced.

5.1 Graph Deep Learning

As we have seen in the Foundation chapter 4, one can define the problem of Graph Denoising as a way of link predication. The state-of-the-art method for solving link prediction are graph deep learning approaches. Graph deep learning is a fast evolving field in research. With Graph Neural Networks (GNN)[13] the framework for GNN has been established.

Using Graph Convolutional Networks (GCN) [14] is a popular way for graph feature extraction. Basically, with GCN a new feature representation is iteratively learned for the node features (edges features are not taken into account). It can be seen as an averaging of nodes over their neighbourhood where all the neighbours get the same weight combined with some non-linear activation. To consider the node itself in the averaging process they apply to so-called "Renormalization trick", where self-loops are added to the adjacency matrix and after every layer, a normalization step is applied.The topology of the graph will not be adjusted during the learning process.

Veličković et al. [22] extended the concept of GCN with attention and not all the neighbouring nodes get the same weight (attention). Simple Graph Convolutional Network (SGC) [26] proposed a simplified version of GCN. They could verify their hypothesis that GCN is dominated by the local averaging step and the non-linear activation function between layers do not contribute to much to the success of GCN. Therefore, it can be seen as a way of power iteration over the adjacency matrix with normalization in every layer. Wang et al. [25] proposed a extended version of GCN by not operating on the same graph in every layer but adopting the underlying graph topology layer by layer.

5.2 Denoising

Denoising is an important part in practical application of ML, as observed signals are often noisy. Especially in computer vision it has a high importance, where observation noise in images is a major issue.

Non local means is a state-of-the-art image denoising method [4]. In the name of the method are two important concepts, namely the *mean* and *non local*.
For a given noisy image v, the denoised image is defined as $NL[v](i) = \sum w(i,j)\ v(j).$ where $w(i,j)$ is the weight between pixel i and j. The weight can be seen as a similarity measure of the two pixels. Moreover, these similarities are calculated over square neighbourhoods of the two pixels, where the l2-norm of the neighbourhood is used. Similar pixel neighbourhoods have a large weight and different neighbourhoods have a small weight. More general, the denoised image pixel i is computed as an weighted average of all pixels in the image, therefore, in a non local way.
Luo et al. [15] introduced PTDNET, a way of topological denoising in graphs. It can be seen as two neural networks, where the first is called denoising network and the second is a GNN. Firstly, in the denoising network noisy edges will be removed due to sampling subgraphs from a learned distribution on edges. The aim is to remove irrelevant edges. Further, in the GNN the node representation of the denoise graph is learned.

5.3 Problem setup section

In the last section of related work, papers which aim to solve part of the Master Thesis problem will be introduced.
Coifman et al. [8] introduces a Laplacian-based algorithm, with which reconstruction of a planar object from projections at random unknown directions is possible. It can be seen as an algorithm for solving classical tomography, where the problem is extended by the fact that projection angles are unknown. They could order projections of the Shepp-Logan phantom by using the Graph Laplacian and used this fact to successfully reconstruct the phantom, even if observations are noisy. The proposed algorithm is not directly applicable to the reconstruction of cryo-EM as projection in 3D do not have a proper ordering.
Miolane et al. [16] introduced a way of estimation camera parameter (PSF) as well as the unknown rotation in cryo-EM reconstruction problem. They combined a Variational Autoencoder (VAE) with a Generative Adversarial Network (GAN). The VAE can be seen as learning a manifold to fit the observations which was used as an input to the GAN.
Diffusion maps[7] (DM) is a non-linear approach for calculating low-dimensional manifolds for (high-dimensional) datasets. The process is based on the concept of random-walks and works as follows: First of all, matrix P is calculated which contains the probability from moving from one node to another. Similar to the k-neighbourhood of A introduced in the foundation chapter, with power of P^t probabilities of reaching node i in t hops can be calculated. The diffusion distance at time t can be seen as a distance measure for two nodes in the diffusion space with added connectivity. It approximates the euclidean distance in the diffusion space and therefore allows to compare node embeddings regarding their euclidean distance. With diagonalization of P and selecting the first n largest eigenvalues/eigenvectors the embedding can be computed. Vector DM (VDM)[19] generalize the concept of DM for vector fields. Multi-Frequency Vector Diffusion Maps (MFVDM)Fan and Zhao [10] can be seen as an extension to Vector Diffusion Maps (VDM)[19], which works well even on highly noisy environments. [10] was successfully used in cryo-EM setting, where it was used for

denoising purpose[11].

6
Master Thesis project

In the last chapter of the Thesis Preparation report, the project plan will be introduced as well as a broad overview of different work packages. Further, the project timeline can be seen as a Gantt chart. Probably, there are some parts which will not work out as expected and adjustments are needed throughout the Thesis, the project plan can be seen as a rough guideline.

6.1 Problem conclusion:

Conlusion from red boxes

6.2 Work packages

Implement algorithm for 2D case: The first step will be, to familiarize with the problem and implement the algorithm for 2D.

Evaluate 2D case on toy dataset and implement baselines: As a second step, the implemented 2D algorithm will be tested on a toy dataset, where noise is added to the images by hand. As the aim is to work with highly noisy images, the noise level can be selected and increased when working with toy datasets. The evaluation in 2D is crucial and needs to in a satisfying matter. It does not make sense to continue with 3D implement, when the simply 2D case is not handled well enough. Therefore, if the evaluation results are not satisfying, the algorithm needs to be iteratively adjusted, such that the evaluation will be in a good enough quality.

Implement algorithm for 3D case: After successfully evaluating the algorithm in 2D, the aim is to extend the algorithm to work in 3D as well.

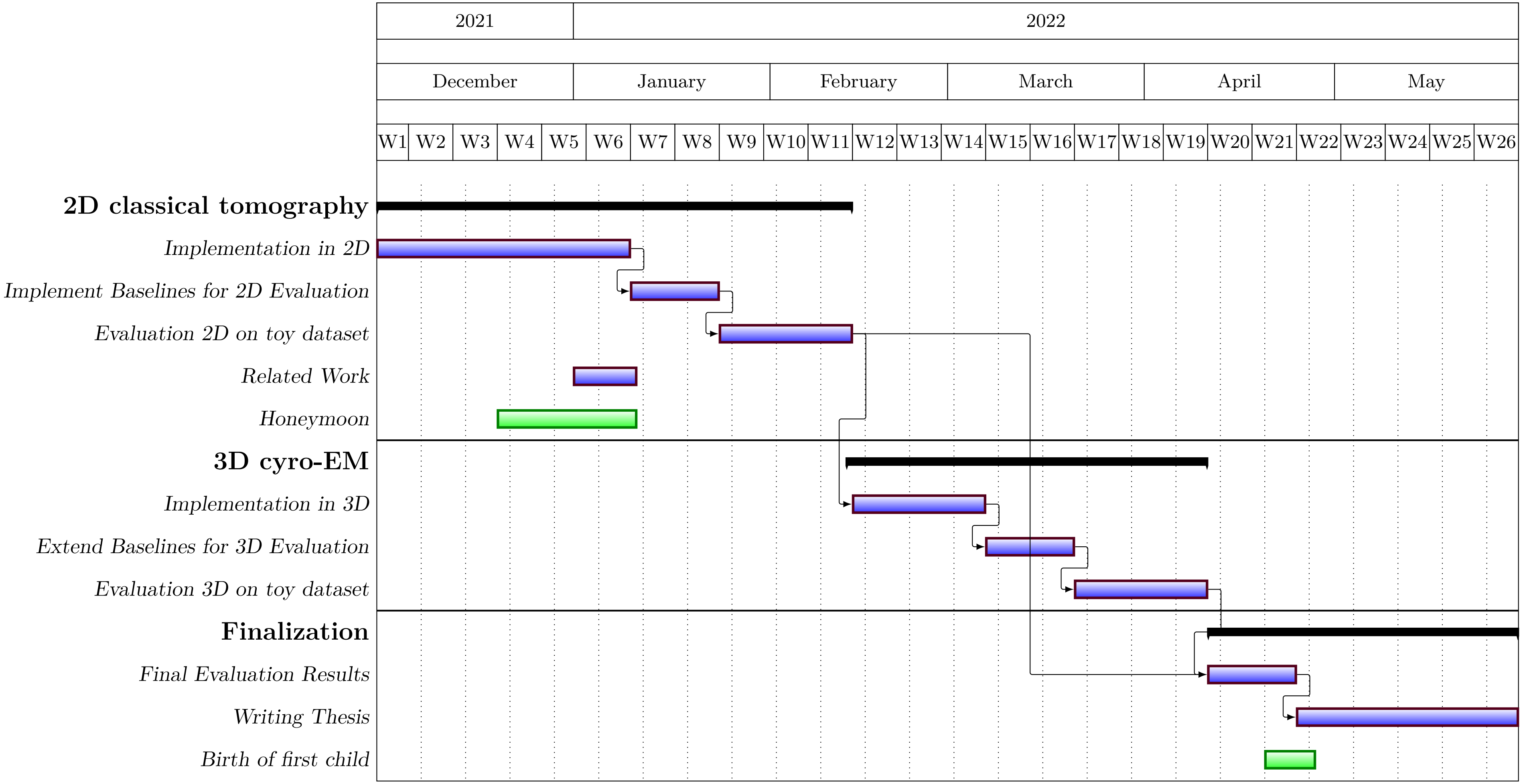
Evaluate 3D case on toy dataset and adjust baselines: Again, the implementation will be evaluated on a toy dataset, where noise can be adjusted by hand.

Nice to have: Evaluate of real dataset If time allows and the 2D and 3D implementation are evaluated successfully on toy datasets, real data can be used for further evaluation. This step will only be done, if time allows.

Evaluate related work: As cryo-EM reconstruction is a hot research topic, related work can not only be considered during the start of the Thesis and needs to be evaluated throughout the Thesis.

Writing Thesis: Document implementation and evaluation result.

6.3 Gantt chart



Bibliography

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/arjovsky17a.html.

[2] Tamir Bendory, Alberto Bartesaghi, and Amit Singer. Single-particle cryo-electron microscopy: Mathematical theory, computational challenges, and opportunities. *IEEE signal processing magazine*, 37(2):58–76, 2020.

[3] David J Brenner and Eric J Hall. Computed tomography—an increasing source of radiation exposure. *New England journal of medicine*, 357(22):2277–2284, 2007.

[4] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.

[5] Lawrence Cayton. Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep*, 12(1-17):1, 2005.

[6] Rolf Clackdoyle and Michel Defrise. Tomographic reconstruction in the 21st century. *IEEE Signal Processing Magazine*, 27(4):60–80, 2010.

[7] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

[8] Ronald R Coifman, Yoel Shkolnisky, Fred J Sigworth, and Amit Singer. Graph laplacian tomography from unknown random projections. *IEEE Transactions on Image Processing*, 17(10):1891–1899, 2008.

[9] Allison Doerr. Single-particle cryo-electron microscopy. *Nature methods*, 13(1):23–23, 2016.

[10] Yifeng Fan and Zhizhen Zhao. Multi-frequency vector diffusion maps. In *International Conference on Machine Learning*, pages 1843–1852. PMLR, 2019.

[11] Yifeng Fan and Zhizhen Zhao. Cryo-electron microscopy image denoising using multi-frequency vector diffusion maps. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3463–3467. IEEE, 2021.

[12] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. Learning with a wasserstein loss. *arXiv preprint arXiv:1506.05439*, 2015.

[13] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

[14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[15] Dongsheng Luo, Wei Cheng, Wenchao Yu, Bo Zong, Jingchao Ni, Haifeng Chen, and Xiang Zhang. Learning to drop: Robust graph neural network via topological denoising. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 779–787, 2021.

[16] Nina Miolane, Frédéric Poitevin, Yee-Ting Li, and Susan Holmes. Estimation of orientation and camera parameters from cryo-electron microscopy images with variational autoencoders and generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 970–971, 2020.

[17] Frank Natterer. *The mathematics of computerized tomography.* SIAM, 2001.

[18] Amit Singer. Mathematics for cryo-electron microscopy. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3995–4014. World Scientific, 2018.

[19] Amit Singer and H-T Wu. Vector diffusion maps and the connection laplacian. *Communications on pure and applied mathematics*, 65(8):1067–1144, 2012.

[20] Daniel Spielman. Spectral graph theory. *Combinatorial scientific computing*, 18, 2012.

[21] Peter Toft. The radon transform. *Theory and Implementation (Ph. D. Dissertation)(Copenhagen: Technical University of Denmark)*, 1996.

[22] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[23] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4): 395–416, 2007.

[24] Lin-Wang Wang and Alex Zunger. Solving schrödinger's equation around a desired energy: Application to silicon quantum dots. *The Journal of Chemical Physics*, 100(3): 2394–2397, 1994.

[25] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[26] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

A
Mathematical tools

A.1 Power Iterations

Power iteration (also called power method) is a iteratively method, which approximates the biggest eigenvalue of a diagonalizable matrix A.
The algorithm starts with a random vector b_0 or an approximation of the dominant eigenvector.

$$b_{k+1} = \frac{Ab_k}{||Ab_k||} \tag{A.1}$$

TODO:convergence if there is only one largest eigenvalue and if b0 is not orthogonal to the eigenvector associated with the largest eigenvalue.
The algorithm not necessarily converges. The algorithm will converge, if A has an eigenvalue strictly grater than its other eigenvalues and the initial vector b_0 has a component in direction of an eigenvector, associated with the dominant eigenvector.

A.2 Folded spectrum Method

TODO: I don't know what is the Hamiltonian matrix, so either you define it or don't talk about it. You can also introduce the folded spectrum just as a way to recover the eigenvector associated with a known eigenvalue. Epsilon often refers to a very small scalar.
Calculation of eigenvalues and eigenvectors of a given Hamiltonian matrix H is a fundamental mathematical problem. Often, we are interested in just the smallest values, which can be efficiently computed. But if we are interested in selected values, this can be hard. H is needed to be diagonalized (bring matrix H into diagonal form) which is computationally expensive and for big matrices impossible.
Currently, the best way to solve such problems is the Folded spectrum (FS)[24] method, which iteratively solves the problem. During calculation, the eigenvalue spectrum will be folded around a reference value ϵ.

$$v^{t+1} = v^t - \alpha(H - \epsilon I)^2 v^t, \tag{A.2}$$

with $0 < \alpha < 1$. When $t \to \infty$, then v^∞ will be the eigenvector with respect to the reference value ϵ.

A.3 Wasserstein metric

TODO: the difference between Wasserstein and KL divergence for instance is that it is defined (the value is finite) even if the two distributions have not the same support.

The Wasserstein metric is a distance measure between two probability distributions and it is used in ML as a loss function[12]. Intuitively, it can can be understood as the minimum cost to transfer the mass of one distribution to the other. Therefore, it is also known as the *earth mover's distance.*

As Arjovsky et al. [1] could show, ordinary distance measures like *Total Variation*, *Kullback-Leibler divergence* and *Jensen-Shannon divergence* are not sensible when learning with distributions supported by manifolds On the contrary, Wasserstein metric does a good job as loss function in such scenarios.

A.4 Fourier Transform

Fourier Analysis is the overall field of study, which deals with representing (or approximating) functions as sums of trigonometric functions. When the function is defined in such a way, we are talking from the *Fourier Domain.*

Fourier transform (FT) is the way of transforming signals to the Fourier Domain, which is popular in ML. Basically, with the Fourier transform, a signal can be decomposed to a *Fourier series*, which consists of many weighted sinusoids.

Fourier-slice theorem The Fourier-slice theorem [17] in 3D is defined as follows:

$$F_2 P_2 = S_2 F_3, \tag{A.3}$$

where $F2$ and $F3$ are FTs in 2D and 3D respectively, P2 is a projection operator ($P_2 : 3D \to 2D$) and $S2$ is the restriction operator.

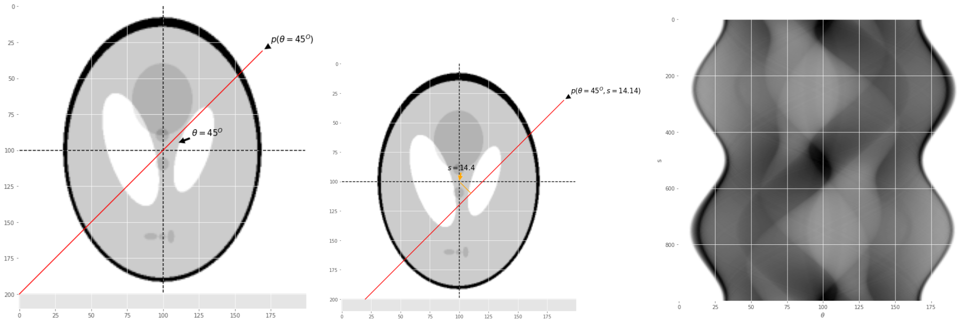
As pointed out by [2], the Fourier-slice theorem is the foundation of the reconstruction problem in computerized tomography (CT), which will be explained in section **??**. It states, that the 2D FT of the tomographic projection is the same as the 3D FT restricted to a 2D plane through the origin. Basically, for the CT reconstruction problem, acquiring samples from known viewing directions is the same as sampling the 3D Fourier-space. This concept is exploited by the filter BackProjection algorithms, see section **??**.

A.5 Radon Transform

The Radon Transform[21] is the main mathematically concept of tomographic reconstruction. It is an integral transformation and the inverse for classical tomography is well defined by the Fourier Transform.

For classical tomography, $R : f \rightarrow Rf, f(x, y) \mapsto Rf(\theta, s)$, where f is a 2D image and x and y can be seen as the coordinates within this image. Then, $Rf(f; \theta, s)$ defines a line, where s is the distance from the origin and θ is the angle to the x-axis.

In Figure A.1(a) and Figure A.1(b) on can see two plots of different values for θ and s, where $f(x, y)$ is the Shepp-Logan phantom. The complete $R(\theta = 45, s = 0)$, which is also called *sinogram*, can be see in Figure A.1(c)



(a) Radon Transform $R(x; \theta = 45, s = 0)$

(b) Radon Transform $R(x; \theta = 45, s = 14.14)$

(c) Shepp–Logan phantom sinogram of $R(x; \theta = 45, s = 0)$

Figure A.1: Examples, where the original object x is the Shepp-Logan phantom.