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

28.11.2021

Table of Contents

1 Introduction

Inverse problems aim to estimate an original signal that went through a system, based on the output signal observation. Machine learning (ML) is a tool to model and solve such inverse problems. 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, pioneers in the field of cryo-EM got the Nobel Prize in Chemistry[1]. Today, using cryo-EM many molecular structures can be observed with near-atomic resolution. The big challenge with cryo-EM is enormous noise.

The following report resulted from Master Thesis Preparation. During the six week project, goal is to familiarize with research area, build up mathematical foundation and define project content as well as a project plan. The report is structured the following:
In chapter 2, the main motivation of the Master Thesis is given. The two imaging methods computed tomography and cryo-EM are introduced and an abstract model. Chapter 3 is dedicated to graphs, were the connection of graphs to computed tomography and cryo-EM is established. Further, the problem of ”Graph Denoising” is defined and methods like graph construction and Graph Laplacian are introduced. Finally, in chapter 4 project content is shortly concluded, work packages are defined and project schedule is given. Throughout the report, red boxes are used to denote important statements regarding final Master Thesis project.

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

2

Imaging methods

In current chapter, imaging methods *computed tomography* and *cryo-electron microscopy* (cryo-EM) will be introduced. Further, their observation model is defined in a mathematic way and reconstruction is presented. Application of cryo-EM is 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 (CT) 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).

> The focus in computed tomography during the Thesis will be on 2D case, which is called *classical tomography reconstruction.*

2D tomographic reconstruction: Mathematically, observations are 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 $y_i \in \mathbb{R}^M$ is i-th observation with $y_i[j] \in \mathbb{R}$ j-th element of the observation and M the observation dimension. Further, N corresponds to number of observations.
Then, $x \in L^2(\Omega)$ is the original object with $\Omega \subset \mathbb{R}^2$ and L^2 the Lebesgue space.
$R(\cdot; \theta, s) : L^2(\Omega) \to L^2(\tilde{\Omega}), x \mapsto R(x; \theta, s)$ refers to the Radon Transform [18] with $\tilde{\Omega} \subset \mathbb{R}$, $\theta_i \in \mathbb{S}^1$ as observation angle and $s_j \in \mathbb{R}$ as sampling point. Finally, $\eta_i \in \mathbb{R}^M$ refers to gaussian noise and is defined as $\eta_i[j] \sim \mathcal{N}(0, \sigma^2) \in \mathbb{R}$.

Filter Backprojection: Filter Backprojection [6] is a reconstruction method, typically used in classical tomography. It allows to inverse the Radon Transform and enables reconstruction of the original object x. The algorithm fails when working with noisy data [15].

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 about 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 makes reconstruction challenging, but freezing allows observation in a stable state where molecules are not moving. With an electron microscope, two-dimensional tomographic projection images of molecules in the ice are observed, which are called *micrograph*. Frozen molecules are fragile and 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 [15]. Further, observed molecules are not equal in the sense that there are some structural varieties between molecules (isotopes). Wile observing the same molecule in ice many times, single observation could be from different isotopes.

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 follow notation from previous section, now $\Omega \subset \mathbb{R}^3$ and $\tilde{\Omega} \subset \mathbb{R}^2$.
Mathematically, observation is defined as follows:

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

where $y_i \in \mathbb{R}^M$ with M as observation dimension.
Then, $\Pi_z : L^2(\Omega) \to L^2(\tilde{\Omega}), x \mapsto \int x(\cdot,\cdot,z)dz$ is projection operator from z-axis and $Rot : L^2(\Omega) \to L^2(\Omega), Rot(x,\theta_i) = \big((x_1,x_2,x_3) \mapsto x(x_1 R^1_{\theta_i}, x_2 R^2_{\theta_i}, x_3 R^3_{\theta_i})\big)$ is rotation operator modelling the rotation during freezing. Further, $\theta_i = [\theta_i^{(1)}, \theta_i^{(2)}, \theta_i^{(3)}]$ where entries $\theta_i^{(1)}, \theta_i^{(2)}, \theta_i^{(3)} \in \mathbb{S}^1$ and $R_{\theta_i} = R_{e_x}(\theta_i^{(1)}) R_{e_y}(\theta_i^{(2)}) R_{e_z}(\theta_i^{(3)}) = [R^1_{\theta_i}, R^2_{\theta_i}, R^3_{\theta_i}] \in SO(3)$ is the 3D rotation matrix (see A.1 for further details). $\eta_i \sim \mathcal{N}(0, \sigma^2 I) \in \mathbb{R}^M$ corresponds to noise of observation.
As y_i is not observable directly, discretization is needed:

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

where $\Delta \subset \tilde{\Omega}^{M^2}$ is the sampling grid with dimension M^2. Further, $y[j,k]$, $\eta[j,k]$ and $\Pi_z(\cdot)_{j,k}$ $\in \mathbb{R}$ with j,k as indices of the sampling grid.

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

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

where h_i is the PSF of the microscope and $\circ$ defines the convolution. Further, $x_i \in X$ where X is the set of all possible molecule structures.

During Master Thesis, equation 2.3 is used, not the extended version.

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

2.3 Abstract form

As tomographic reconstruction and cryo-EM reconstruction are rather similar, goal of the Master Thesis will be to design an algorithm, that can be applied in both scenarios.
Therefore, an 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 with $x \in L^2(\Omega)$ where $\Omega \subset \mathbb{R}^D$ with D as the dimension of the space of original object and $\tilde{\Omega} \subset \mathbb{R}^{D-1}$ as the dimension of the space of observations.

$$y_i = (A(x, \theta_i) + \eta_i)\,(\Delta), \text{ with } 1 \leq i \leq N \\ , \tag{2.5}$$

where $y_i \in \tilde{\Omega}^M$ is the i-th observation, M observation dimension, $x \in L^2(\Omega)$ original object, A a non-linear operator $A : L^2(\Omega) \to L^2(\tilde{\Omega}), x \mapsto A(x;\theta_i)$, $\theta_i \in \mathbb{S}^P$ projection angle vector with P the projection dimension and $\eta \sim \mathcal{N}(O, \sigma^2 I) \in \tilde{\Omega}^M$ gaussian noise. $\Delta \subset \tilde{\Omega}^M$ is a term for discretization.

Classical tomography reconstruction: Classical tomography parameters are defined with $D = 2$, $P = 1$.. Further, $A(\cdot)$ is the Radon Transform (see equation 2.1). A distance measure between measurements can be set up by using the $\ell 2$-norm $\|y_i - y_j\|$.

Cryo-Em reconstruction: Cryo-EM parameters are defined with $D = 3$, $P = 3$ and θ_i not only corresponds to a projection angle vector but also some rotation. Further, $A(\cdot)$ can be defined as $\Pi_z\,(\,Rot(\,x;\theta))$ 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 observation y_1, which has no 3D rotation and a second observation y_2 with a rotation in x-y plane by 45°. The two observations have a defined in-plane rotation g, such that $g\ y_1 = y_2$. Therefore, in distance measure term of in-plan rotation is added: $min_{g \in SO(1)} \|g\ y_i - y_j\|$, which is inspired by [10].

High noise regime: Cryo-EM observations are highly noisy, which makes reconstruction challenging. There are different ways to reduce noise from observations, most of them are related to averaging. Averaging need to consider similar observations and ignore diverse ones. In the defined abstract model, averaging over paired observations from θ should be a good averaging model. But how can it be achieved?
One idea would be to measure distances between observation. Another way is to find a low-dimensional embedding which maps our observations 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 domain of interest. Main practical application is cryo-EM, where an algorithm for denoising is expected to boost quality of the overall 3D-reconstruction. As cryo-EM is a 3D problem, computed tomography will be considered as well which allows to test on a corresponding 2D problem. The goal of the Master Thesis is to introduce a denoising algorithm, which is able to work well even on highly noisy data. Reconstruction of original object is not in the scope of the project.

3
Graph Denoising

Following chapter establishes connection between graphs and denoising in high-noise domains, such as cryo-EM. First, a broad definition of graphs is given and further, the term ”Graph Denoising” is introduced and explained. Finally, connection to Graph Laplacian is established and different opportunities exploiting for a good denoising algorithms are shown.

3.1 Graph Foundations

Real world data can be in graph structure, like social networks, citation networks, protein interaction networks or google search. If data is not available in graph structure, a graph can be artificially constructed with methods like k-nearest neighbours (k-NN) or others. A general framework for graph construction is introduced in section 3.1.2.

Graph Learning: Graph Learning is a hot research area and got a lot of attention in recent years. It is a way of applying ML on graphs and algorithms emerged from ML but also other fields. 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, where model is learned from node and edge features as well as topology. The model can than be used for prediction or classification. Another common task is *community detection*, where the aim is to identify cluster of nodes within the input graph. Further, graphs are highly favoured for *dimensionality-reduction*, where graph algorithms provide a helpful tool, as ordinary algorithms like principle component analysis fail to establish a meaningful dimensionality-reduction.

3.1.1 Graph definition

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 vertices in the graph.

Graph properties: A graph can be either *directed* or *undirected*. In a directed one, an edge connects explicitly from one node to another, which means that edge $(i, j) \neq (j, i)$. In

undirected graphs ordering does not matter and $(i, j) = (j, i)$.
The *neighbourhood*, denoted by $\mathcal{N}(i)$, of a node i is defined as all adjacent nodes. In other words, there is an edge between neighbourhood nodes and i. Further, edges can have *weights*, which is a method to define importance to neighbours of a node. If edges are dealing with weights, the term *weighted* graph is used. The *degree* of a node are the number of incoming edges.

Adjacency matrix: To do calculations with graphs, it is common to translate graphs in a matrix, such as the adjacency matrix. The (binary) adjacency matrix of graph G is defined as follows:

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

Matrix A has dimension $\mathbb{R}^{N \times N}$ with N as number of nodes and indices of A correspond to nodes V. If there exists an edge between two nodes, entry in A will be set to 1, otherwise to 0. This leads to an unweighted graph, as weights of all edges will be 1, but could easily be extended by assigned not just values of 1 or 0. When the graph is undirected, the corresponding adjacency matrix will be symmetric. Eigenvalues of A are called *spectrum* of the graph.

3.1.2 Graph construction

When data is not available as a graph, it can be easily constructed. Consider data from space $\Omega \subset \mathbb{R}^M$, but could basically be any arbitrary space. Then, each node is associated with some element $x \in \Omega$. Further, the graph G can be constructed by using:

$$A_{ij} = \begin{cases} 1 & \text{if } d(x_i, x_j) < \tau \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

where x_i, x_j are nodes from indices i, j, d corresponds to a similarity measure and τ is a threshold, when to consider two nodes to be adjacent. K-NN is one possible implementation of a graph construction algorithm, where for every node, k neighbours will be defined. The neighbourhood $\mathcal{N}_i$ of node i is defined as k nodes with smallest similarity measured.

Noise regime In the case of noise, observation of x it not possible. Measurements will give access to $y = x + \eta$ where $y, x \in \Omega$ and the noise η is assumed to be drawn from gaussian distribution $\mathcal{N} \sim (0, \sigma^2)$. The *noisy graph* G_0 can be constructed as in equation 3.2, but replacing y with x:

$$A_{0_{ij}} = \begin{cases} 1 & \text{if } d(y_i, y_j) < \tau \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

3.2 Graph Denoising definition

First of all, *Graph Denoising* is not a common term in literature. In previous section, noisy graph G_0 was introduced and goal is to denoise this graph, which means to estimate original graph G from a given noisy graph G_0. This is our definition for Graph Denoising, which is rather related to signal or image denoising. Reconstruction of a true signal given noisy observation signal 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 G_0 can further be defined as $G_0 = \langle V, E_0 \rangle$, where $E_0 = E \setminus E_0^- \cup E_0^+$ with $E_0^- \subseteq E$ and $E_0^+ \cap E = \emptyset$.
G_0 consists of same nodes V as original graph G. From E some edges are removed (denoted by E_0^-) and some are added (denoted by E_0^+), which results is edges E_0.
Graph Denoising can therefore be written as $GD : A_0 \mapsto \tilde{A} \approx A$, where A_0, $\tilde{A}$, A denotes adjacency matrix from noisy input graph, denoised graph and original graph respectively.

The goal of the Master Thesis is to introduce a method to estimate original graph G with corresponding A based on observed noisy graph G_0 with A_0.

Connection to link prediction: Link prediction is a task in Graph Learning. The goal is to predict existence of a link 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 $M_p : E' \to [0, 1]$ where E' is the set of potential links.
Further, U determines the set of all possible vertices of G, therefore $E \subseteq U$. Clearly, 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 $E' \subseteq E$ and in Graph Denoising model is learned from $E' \subseteq U$.

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

3.2.1 Non-local means:

In the following section, a short introduction to the state-of-the-art image denoising method non-local means is given [4]. 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 weight between pixel i and j. Weight can be seen as similarity measure of pixels, which are calculated over square neighbourhoods. Similar pixel neighbourhoods have a large weight and different neighbourhoods have a small weight. More general, denoised image pixel i is computed as an weighted average of all pixels in the image, therefore, in a non-local way.
Non-local means is not a denoising algorithm, which works with graph as a data structure. But, it uses a neighbourhood for averaging, which shows great potential of graphs as a data structure for denoising, as graphs can represent neighbours and neighbourhoods really well.

3.3 Graph Laplacian

Graph Laplacian is a matrix that represents a graph and can be used to find many important properties. It is a very powerful tool and therefore, a complete section is dedicated to it. A good introduction and overview can be found by [17, 20].
The matrix is defined as follows:

$$L = D - A, \tag{3.4}$$

where A is the adjacency matrix and D the degree matrix (diagonal matrix with degree of nodes as entries).

3.3.1 Manifolds

In high-dimensional data Euclidean distances are not meaningful, in the sense that they will not capture similar data points well. Graph Laplacian can be used to compute a *Manifold*, which can help in such scenarios. In manifold space, Euclidean distances make sense again. Let manifold M be defined as $\mathcal{M} = \{f(x), f \in C^K, f : \mathbb{R}^D \to \mathbb{R}^d\}$. Manifolds are a well established mathematical concept. In the Master Thesis, only C^k differentiable d-dimensional manifolds defined by $\mathcal{M}$ are considered. When $d \ll D$, manifolds define a *low-dimensional embedding*, which maps from high-dimensional space $\mathbb{R}^D$ to low-dimensional space $\mathbb{R}^d$.
Lets give two popular examples of manifolds, namely the *circle* and the *sphere*. The circle is a 1D manifold, where $d = 1$ and $D = 2$. A sphere is a 2D manifold, with $d = 2$ and $D = 3$. In figure 3.1(a), 200 samples are drawn from a uniform distribution of the unit-circle manifold and in figure 3.1(b), 400 samples are drawn from a uniform distribution of the unit-sphere manifold, as well as the sphere itself.

(a) Circle samples (b) Sphere samples

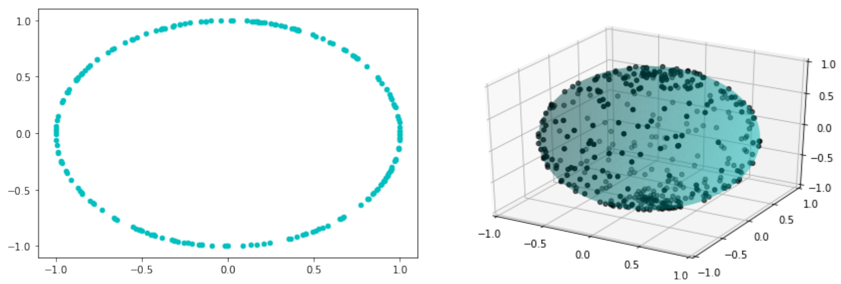


Figure 3.1: Samples drawn from 1D and 2D manifold.

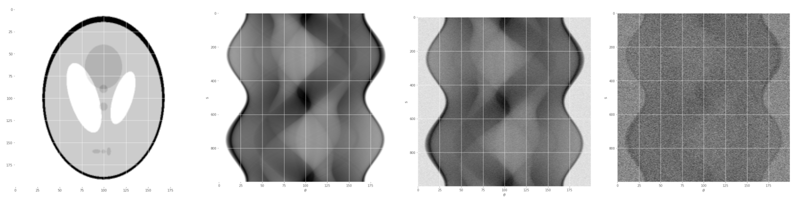
One popular algorithm for calculating manifolds is diffusion maps [7], which is a non-linear approach for calculating low-dimensional manifolds for (high-dimensional) datasets, using Graph Laplacian. Vector diffusion maps [16] generalize the concept of diffusion maps for vector fields. Multi-frequency vector diffusion maps [10] can be seen as an extension to vector diffusion maps, which works well even on highly noisy environments. Fan and Zhao [11] successfully applied multi-frequency vector diffusion Maps in cryo-EM setting, where it was used for denoising purpose.

Manifold assumption: Manifold assumption is a popular assumption for high-dimensional datasets. For a given dataset in high-dimension, one can assume that data points are samples drawn from a low-dimensional manifold, that embeds the high-dimensional space. Therefore, if underlying manifold can be approximated, a dimensionality reduction is established 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], but it is not only used there.

Manifold calculation: Manifold of a dataset can be calculated the following:

1. Construct k-NN graph from observations (see section 3.1.2).
2. Calculate the (normalized) Graph Laplacian.
3. Extract the second, third (and fourth) smallest eigenvectors.

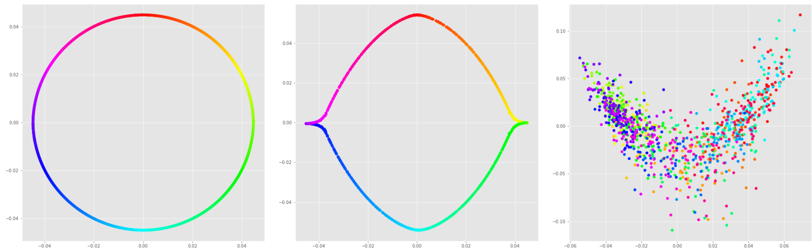
Therefore, it can be observed how the manifold of classical tomography and cryo-EM objects look like. In the following, the Shepp-Logan phantom (figure 3.2(a)) is used as an example of classical tomography. In figure 3.2(b), figure 3.2(c) and figure 3.2(d) sinogram of the original phantom and phantoms, where gaussian noise was added, are shown.



(a) Shepp-Logan phantom (b) Original sinogram (c) Noisy sinogram (d) Highly noisy sinogram

Figure 3.2: Shepp-Logan phantom and sinograms

In Figure 3.3(a) the manifold calculated from original phantom Graph Laplacian can be seen and it is a perfect circle. Further, in figure 3.3(b) noisy version with $\sigma = 2$ is plotted and the manifold is not a perfect circle, but circle like.
The more noise is added, the less manifold looks like a circle. In figure 3.3(c) the manifold for $\sigma = 100$ is plotted. In all plots, k-NN graphs have been constructed with $k = 10$.



(a) Original phantom manifold (b) Noisy phantom manifold (c) Highly-noisy phantom manifold

Figure 3.3: Shepp-Logan phantom manifolds

In the field of classical tomography and cryo-EM, the underlying low-dimensional manifold is well defined for none-noisy data. In the 2D case of classical tomography, the underlying manifold is a circle, whereas in 3D case of cryo-EM the manifold if defined as a sphere. This fact can be exploited during learning (e.g. by using Wasserstein loss function (see A.3)).

3.3.2 Connection to Machine Learning

Graph Laplacian is used for dimensionality reduction for high-dimensional data, as well as spectral clustering and semi-supervised learning. Coifman et al. [8] used Graph Laplacian in a complete other domain, namely in tomography. They showed that Graph Laplacian approximates the Laplace-Beltrami operator. Further, Graph Laplacian is depended on the adjacency matrix A, if A is noisy, Graph Laplacian will be noisy as well.

In the Master Thesis, further experience of Graph Laplacian in the domain of tomography and denoising should be gained. Additionally, connection with Machine Learning will be explored, hopefully allowing to learn a denoised adjacency matrix to fully enable the power of Graph Laplacian.

3.3.3 Graph Deep Learning

As already mentioned, Graph Denoising can be seen 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 neural networks with graphs has been established.
Using Graph Convolutional Networks (GCN) [14] for graph feature extraction is a popular way. With GCN a new feature representation is iteratively learned for the node features (edge features are not considered). It can be seen as an averaging of nodes over their neighbourhood where all neighbours get the same weight combined with some non-linear activation (e.g. ReLU). To consider the node itself in averaging they apply the 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

learning process.
Veličković et al. [19] extended the concept of GCN with attention and not all the neighbouring nodes get the same weight (attention). Simple Graph Convolutional Network [22] proposed a simplified version of GCN. They could verify their hypothesis that GCN is dominated by local averaging step and 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 (see A.2 for further information) over the adjacency matrix with normalization in every layer. Wang et al. [21] proposed an extension to GCN by not operating on the same graph in every layer but adopting underlying graph topology layer by layer.

In the Master Thesis, connection between Graph Laplacian and GNNs should be further studied. The fact, that Wu et al. [22] could simplify the existing GCN algorithm is motivation enough that similar connections can be drawn in other fields of Graph Deep Learning.

4
Master Thesis project

In the last chapter of the report, a short conclusion of the Thesis problem is given. Further, the project plan will be introduced as well as a broad overview of different work packages. Finally, the project schedule can be seen as a Gantt chart.

4.1 Problem conclusion

Cryo-Em and computed tomography within high noise regime is the field of interest. In both imaging methods, the final goal is to reconstruct an original object from observed noisy data. Better denoising will boost the overall quality of reconstruction.

The Master Thesis will therefore introduce a new method for denoising. We believe, that graphs are a well suited data structure for denoising. Additionally, we hope to explore further connection of the powerful Graph Laplacian and Machine Learning and use it to estimate a denoised graph from a noisy one. Further, the idea is to exploit the fact, that the manifold of the noise-less input data is known.

4.2 Evaluation:

The introduced denoising method will be evaluated during Master Thesis, where 2D and 3D scenario will be considered. A first evaluation will be done on artificial constructed toy-dataset. As the goal is to work with highly noisy data, 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 implementation, when the simple 2D case is not handled well enough. If time allows, real dataset from classical tomography and/or cryo-EM[2] can be evaluated as well.

During evaluation, two baselines are considered, which already solved part of the problem. The first one is a multi-frequency diffusion map approach[10, 11], which aims to denoise cryo-EM images. Secondly, [8] a Graph Laplacian approach solving classical tomography with random projection angles will be compared against (only possible in 2D case).

[2] https://www.ebi.ac.uk/emdb/

The evaluation process is a first broad idea. Any adjustments in baseline papers or dataset are possible during Master Thesis.

4.3 Work packages

In the upcoming section, some work packages are defined. Probably, there are some parts of the project which will not work out as expected and adjustments are needed throughout the Thesis. The project plan can be seen as a rough guideline.

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 images by hand.

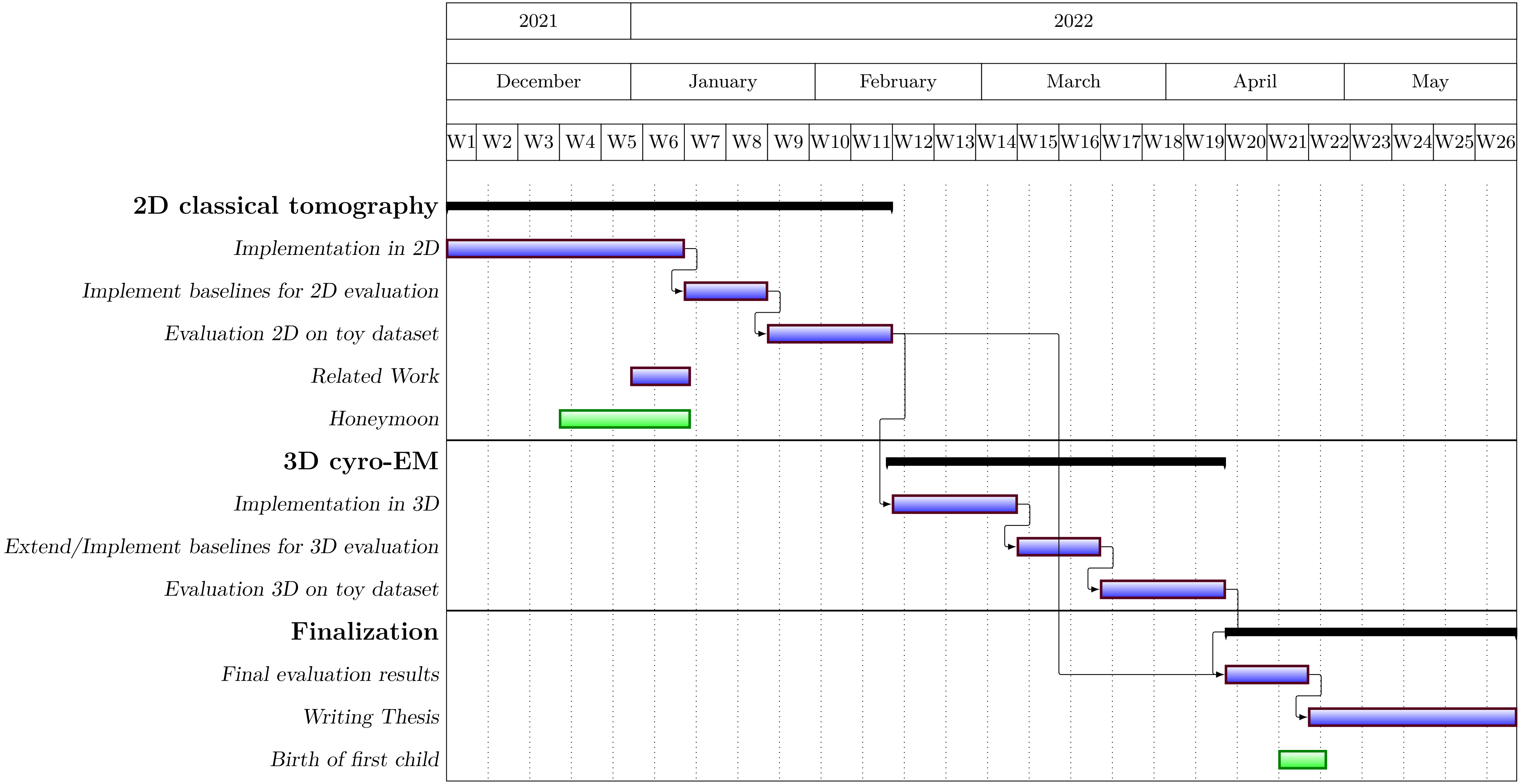
Implement algorithm for 3D case: After successfully evaluating the algorithm in 2D, the goal is to extend the algorithm to work in 3D as well.

Evaluate 3D case on toy dataset and adjust baselines: Again, 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 2D and 3D implementation are evaluated successfully on toy datasets, real data can be used for further evaluation.

Writing Thesis: Document implementation and evaluation result and write final Master Thesis report.

4.4 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] 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.

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

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

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

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

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

[21] 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.

[22] 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 3D rotation matrix

A rotation matrix is a transformation matrix used to perform rotations. In 3D case, matrix for rotating one single axis can be described as:

$$R_{e_x}(\theta)\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\theta & -\sin\theta\\ 0 & \sin\theta & cos\theta\end{bmatrix} \tag{A.1}$$

$$R_{e_y}(\theta)\begin{bmatrix}\cos\theta & 0 & \sin\theta\\ 0 & 1 & 0\\ -\sin\theta & 0 & cos\theta\end{bmatrix} \tag{A.2}$$

$$R_{e_z}(\theta)\begin{bmatrix}\cos\theta & -\sin\theta & \\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix} \tag{A.3}$$

where e_x, e_y, e_z corresponds to the axis unit-vector (for x: $(1, 0, 0)$, etc.) and $\theta \in \mathbb{R}$. To combine the single axis rotations, matrices can be multiplied with each other:

$$R(\theta) = R_{e_x}(\theta)R_{e_y}(\theta)R_{e_z}(\theta) \tag{A.4}$$

In equation A.4, angle θ is the same for all axis, which does not have to be.

A.2 Power Iterations

Power iteration, also called power method, is an iterative method that approximates largest eigenvalue of a diagonalizable matrix A.

The algorithm starts with a random vector b_0 or an approximation of the dominant eigenvector.

$$b_{k+1} = \frac{Ab_k}{||Ab_k||} \tag{A.5}$$

The algorithm not necessarily converges. The algorithm will converge if A has an eigenvalue strictly grater than its other eigenvalues and initial vector b_0 is not orthogonal to the eigenvector associated with the largest eigenvalue.

A.3 Wasserstein metric

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 that ordinary distance measurements 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.