Humboldt-Universität zu Berlin

Chair of statistics

Extension of generalized additive models to graph-valued data with application to Covid-19 impacts on European air transportation

Master thesis
for acquiring the degree of
Master of Science (M.Sc.)
in Statistics

Berlin, August 25, 2022

Submitted by:
Marco Simnacher
Student number:
609970

First examiner:
Prof. Dr. Sonja Greven
Second examiner:
Dr. Matthias Eckardt

Contents

List of abbreviations

AAC	Align all and compute
AIC	Akaike's information criterion
AL	Artificial label
Covid-19	Corona Virus Disease 2019
ADS-B	Automatic Dependent Surveillance–Broadcast
EF	Exponential family
EU	European Union
FD	Fundamental domain
GA	Graduate assignment algorithm from `graphspace` package
GAS	GAS algorithm from `graphspace` package
GAS1	GAS1 algorithm from `graphspace` package
GAM	Generalized additive model
GCV	Generalized cross validation
GLM	Generalized linear model
GLMM	Generalized linear mixed models
IRLS	Iterative (re-)weighted least squares
LEQR	Humboldt Lab for Empirical and Quantitative Research
LMM	Linear mixed models
LS	Least squares
MLE	Maximum likelihood estimator
PIRLS	Penalized iterative (re-)weighted least squares
PLS	Penalized least squares
REML	Restricted maximum likelihood
US	United States of America

List of Figures

List of Tables

1. Introduction

Statistics on the space of unlabeled networks can be allocated to object-oriented data analysis as in Marron & Dryden (2021) and to statistics for objects with geometric structure summarized hereafter from Feragen et al. (2019). On the one hand, this active research area is appealing by its interdisciplinarity in geometry, topology, stochastics, statistics and computer science. On the other hand, its practical importance increases with the rapid raise in available complex data such as networks, trees and shapes. One approach to analyse such geometric, complex objects treats them as points in their non-Euclidean spaces. To develop novel statistical methodologies on these spaces, one idea is to exploit their geometric structure and derive generalizations of least squares objectives. Afterwards, the objectives can be optimised for instance by taking advantage of appropriate mappings to related Euclidean spaces.

Less generally, there exist recent geometrical and topological results on the spaces of graph-valued objects facilitating the derivation of statistics treating the networks as points in these spaces, which is also called second generation approach. In first attempts, the generalization of least squares objectives for unlabeled networks is enabled utilizing a mapping between labeled and unlabeled networks and applying a Procrustean analysis framework. In this spirit, a sample Fréchet mean, a geodesic principal component analysis and a linear regression model for unlabeled networks as output are already developed together with estimation methods called align all and compute algorithms in Calissano et al. (2020, 2022). This master thesis investigates the possibility to extend the linear regression framework to an additive and a generalized additive framework. Both extensions aim for more flexible regression models while the latter would additionally allow for an appropriate treatment of the support in the graph space. Aside from that, the goal is to evolve first ideas for a deeper understanding of the regression functions as well as possibilities for interpretation and validation. Along the way, topological and geometrical results are utilized, probabilistic relationships exposed and questioned, challenges insinuated and if possible, exemplified.

Alongside the theoretical, interdisciplinary allures of statistics on the space of unlabeled networks, researchers can be motivated by applying the models to varieties of complex data. As described in Kolaczyk et al. (2020), networks are increasingly observed in larger sets. For example, social networks can be collected over time and users might be interested in patterns of groups of these networks as outlined in Kolaczyk & Csárdi (2020); Kolaczyk et al. (2020). In another example, brain arterial networks are collected for people of different age as presented in Guo et al. (2021). Furthermore, biological networks as well as epidemiological networks might offer an interesting collection for the application of the derived novel statistical methods. Moreover, transportation and mobility networks can be of interest as for example in Calissano et al. (2022). The latter motivated the application of the extension of generalized additive models to the air transportation network of parts of Europe in this thesis. The aim was to extend the current

implementation of the linear regression framework to the derived generalized additive framework, speed up computation time and apply it to real world data.

Before we start, we point out important terminology that is used throughout the thesis. The terms graph and network are used interchangeably. We refer to node labeled networks as labeled networks and networks without node labels as unlabeled networks. The space of labeled networks is named total space and the space of unlabeled networks graph space. When we refer to node attributes this is equal to the attribute of the loop of a node, that is the attribute of the edge between the node and itself. We call node and edge attributes together also graph attributes. Instead of 'the graph attribute with index a' we usually write 'the graph attribute a'. We do not discriminate between linear and smooth function on graph attribute level and on graph level. This means, we write for example sometimes a linear function in the total space although this refers usually to elementwise linear functions for the graph attributes. For the most part, we do not differentiate between permuting nodes or permuting node labels although the latter is more precise. Furthermore, we write usually passengers instead of air passengers as well as passengers instead of number of passengers.

Finally, the thesis is structured as follows: In section 2, we present an overview of the existing literature. Section 3 is the central part of the thesis. Therein, we introduce the graph space, the existing linear regression framework in the graph space and give a brief review on regression models for one-dimensional responses. Afterwards, the linear regression framework in the graph space is extended to additive and generalized additive models. Section 4 is essential to understand the model on a deeper level and discusses important general considerations. Later, section 5 describes the application of the generalized additive framework to the air transportation network of parts of Europe on country level during the Covid-19 pandemic. Ultimately, section 6 summarizes the results and discusses possible further research questions.

2. Literature review

The proposed regression framework expands the existing methodology-oriented literature by: 1. extending the linear regression framework in the graph space to additive models 2. presenting insights into the possibility for an extension to generalized additive models (GAMs) 3. outlining the motivation for graph space modeling as well as investigating ideas to interpret and validate the regression frameworks. Moreover, the application extends the existing implementation and application-oriented literature by: 1. an implementation of the developed GAM framework 2. an approximation for daily and weekly air passenger networks 3. an unlabeled network analysis of parts of the European air transportation system during the Covid-19 pandemic.

We review literature related to unlabeled networks, to labeled networks, references to extend the single target models in the regression framework and finally, literature related to air transportation networks and their development during the Covid-19 pandemic. During the review, we mention existing applications to give the reader a first idea of the possible practical relevance of the models. To the best of our knowledge, there exists the following related work.

Most related to this thesis is the work Calissano et al. (2022). Therein, the authors introduced the linear regression model in the graph space and its computation by the align all and compute (AAC) algorithm, which are both extended in this thesis. In addition, the authors apply the model to networks of crypto currency exchange rates, passing networks in football and bus transportation networks from Copenhagen. The application on the bus transportation networks is conceptually similar to our application on air transportation networks.

Moreover, the geometry of the graph space and similar AAC algorithms for the computation of the mean and principal components in the graph space are analysed in Calissano et al. (2020). Therein, the methods are applied to simulated and mobility datasets. Overall, the investigations in Calissano et al. (2020, 2022) focus mainly on real valued, possibly multi-dimensional graph attributes. Therefore, the authors base their work on geometrical results on the space from Jain (2016a). Aside from that, the same author published first results for the statistical analysis in the graph space with Euclidean graph attributes in Jain (2016b). Therein, an early iterative align all and compute algorithm for the mean can be found. The approach is formulated in a similar manner as the AAC algorithms in Calissano et al. (2020, 2022), namely as a majorize-minimize mean algorithm. Additionally, the author investigates existence and uniqueness criteria for the Fréchet mean in the graph space. Besides these works on graph space with Euclidean graph attributes, more general geometrical results on a larger class of spaces were already obtained in Jain & Obermayer (2009).

Similar geometrical and topological considerations as above can be found in Kolaczyk et al. (2020). However, the authors restrict their analysis to the special case of one-dimensional graph attributes on the positive real line, this is (id est, i.e.) to positive, continuously weighted networks. Then, they define the theoretical and empirical Fréchet mean to analyse averages of

networks. Moreover, asymptotic and uniqueness results on the Fréchet mean are given. Besides, the examples for so called fundamental domains (FDs) given by the authors allow for a better understanding of the space. Furthermore, this graph space is shown to be a stratified space. We refer the reader to Feragen & Nye (2020) for a general description of statistics on stratified spaces with focus on trees.

In contrast to the restriction to positive, real valued graph attributes, the works Guo et al. (2022); Guo & Srivastava (2020); Guo et al. (2021) extend the attribute space to Riemannian manifolds. In particular, the authors investigate shapes as edge attributes and call the corresponding space graph shape space. In this space, the authors define the concept of geodesics and means. Then, they introduce algorithms for the computation of the mean and principal components. Furthermore, they apply their framework to a database encoding letters as graphs, to molecular graphs, wikipedia graphs, brain arterial networks and neurons.

So far, the references utilize the possibility of mathematically representing labeled graphs by their adjacency matrices to derive statistical methods. An alternative way of mathematical representing labeled graphs are Laplacian matrices. With Laplacian matrices, labeled graphs can be treated as manifold-valued data and one can fall back on statistical methods for manifolds. This is done for example in Ginestet et al. (2017) where the geometry of the space of labeled networks represented by their Laplacian matrices is investigated and asymptotic results on network means as well as testing schemes are derived. In Severn et al. (2022), the authors derive a mean, a principal component analysis, a central limit theorem and hypothesis testing for labeled graphs represented by Laplacian matrices. Moreover, possible metrics are investigated. The authors apply their methods to the text of two novels where words are nodes and edges are weights for their pairwise cooccurrence. Similarly, the authors in Severn et al. (2021) introduce an extension of the Nadaraya-Watson estimator for regression with labeled graphs as output. The regression framework is applied to networks from an email dataset to discover anomalies over time and to networks of word occurrences of two authors over time. In contrast to this and to avoid the extrinsic methods in the former two papers, the authors in Zhou & Müller (2021) derived an intrinsic regression method by adapting the general Fréchet regression framework from Petersen & Müller (2019). The intrinsic approach and the adaption of the general Fréchet regression made it possible to derive theoretic results such as rates of convergence for the estimators. The framework is applied to neuroimaging data and taxi networks. Finally, we emphasize that the methods developed so far for Laplacian matrices correspond to the case of labeled networks. The connection between the labeled graph representation via adjacency matrix and via Laplacian matrix is derived for example (exempli gratia, e.g.) in Guo et al. (2021, section 2.2) and the authors proved that the developed methods are not equivalent since e.g. geodesics do not translate in general and adjacency matrices are not necessarily elements of the space of Laplacian matrices.

Up to now, we described parts of the existing literature corresponding to unlabeled and labeled graph modeling, respectively. Beyond, we extend separately each of the single target models of the multi output regression model in the total space to extend the linear regression model in the graph space to more general regression models. In particular, the single target models that are considered in this thesis are additive models and GAMs. Therefore, early work on additive models can be found in Friedman & Stuetzle (1981). Furthermore, GAMs include additive models

as special case and were first introduced in Hastie & Tibshirani (1986) as an flexible extension of generalized linear models which were developed in Nelder & Wedderburn (1972). We refer the reader to the textbook Fahrmeir et al. (2013) for a stepwise extension of regression models as it is done briefly in this thesis in subsection 3.3 and to the textbook Wood (2017) for an application- and computation-oriented reference.

As an alternative to the extension of the single target regression models and to account for the dependency between the graph attributes, we could replace the single target models by a more sophisticated multi output model. A survey on multi output regression models can be found e.g. in Borchani et al. (2015).

Ultimately, we apply the extended regression model to the air passenger network of parts of the European Union (EU) during the Covid-19 pandemic. Multiple articles investigate the impact of the Covid-19 pandemic on air transportation systems. A survey can be found in Sun et al. (2021). The idea to aggregate multiple airports on country level was introduced in Wandelt & Sun (2015) where the air transportation network is modeled with countries as nodes. Moreover, we approximate the number of passengers between countries. A similar approximation for the passengers is given in Suzumura et al. (2020). Finally, the accessible analysis of the Covid-19 pandemic on air transportation is also enabled by Strohmeier et al. (2021) with the processing of freely available flight data which is described in Schäfer et al. (2014).

3. Regression framework

The aim of this section is to extend the linear regression framework in the graph space to an additive and generalized additive regression framework. First, we introduce the graph space and summarize important geometrical results on this space which were derived in Calissano et al. (2020); Guo et al. (2021); Jain (2016a,b); Kolaczyk et al. (2020) and are summarized under a common notation here. Afterwards, the linear regression framework from Calissano et al. (2022) is introduced together with the align all and compute (AAC) algorithm to minimize an empirical error in the graph space. Next, we review regression frameworks for one-dimensional responses, covering linear models up to GAMs. Finally, we derive the extension of the linear regression framework to additive and generalized additive models in the graph space, combining the three previous subsections. Examples with visualisations, numeric algorithms and an idea for the proof of algorithm 2 for additive models can be found in the appendix 6.

3.1. Graph space

To be able to work with unlabeled networks as data objects, we introduce the space in which they live. This means, we start by defining labeled graphs and the set of all node permutations to define unlabeled graphs as equivalence classes. We observe, that the space of these unlabeled graphs together with an appropriate metric is a metric geodesic space but no manifold. The geometric introduction of the space is helpful since, although the space is no manifold, the derivation of statistical methods can be build on geodesics using metric geometry e.g. from Bridson & Haefliger (2013). Then, uniqueness of the metric in the graph space is considered briefly by the introduction of FDs, before the subsection closes with discussions on general attribute spaces for the graph attributes and the addition of null nodes for graphs of different sizes.

The introduction of unlabeled graphs by their labeled representations and the set of label permutations has the advantage of a straightforward definition of the graph space as a quotient space. A possible disadvantage can be its potential counter intuitiveness with respect to (w.r.t.) its implications on the results since we analyse unlabeled instead of labeled graphs. Thus, this derivation of unlabeled graphs has to be treated carefully especially when we translate it to the interpretation of the results. We discuss this in more detail in section 4. Moreover, another, more general introduction of the space of unlabeled graphs can be found in Jain (2016a). Therein, the mathematical representation of graphs is not fixed to the matrix representation with adjacency matrices and thus, more flexible. Moreover, alignment of graphs is defined more generally with injective partial maps and a metric is induced from so called optimal alignment kernels. In the end, the corresponding metric space obtained in this way is shown to be isometric to the quotient space derived in here, compare (cp.) Jain (2016a, theorem 4.4), at least for an Euclidean attribute space. Due to the additional flexibility for the alignment of graphs, this and even more

general spaces as in Jain & Obermayer (2009) might be helpful if the model derived in this thesis is translated for example to bipartite or hypergraphs. Nevertheless, since the introduction of unlabeled graphs as set of permuted labeled graphs is more straightforward w.r.t. the obtained quotient space and thus, more practical to work with and the introduction by optimal alignment kernels was proven to be isometric with the quotient space, we will focus solely on the former.

First, let $G = (V, E, a)$ be a graph, where V is the set of nodes and $E \subset V \times V$ the set of edges. We assume that the graph has at most $|V| = k, k \in \mathbb{N}$ nodes and $|E| = k^2$ edges. Moreover, the attribute mapping $a : V \times V \to \mathcal{A}$ assigns node and edge attributes to the graph. The attribute space $\mathcal{A}$, which we call **total space**, is assumed to be the Euclidean space $\mathbb{R}^{k^2 d}$ for some $d \in \mathbb{N}$ where d corresponds to the dimension of the graph attributes. In the case of $d = 1$, a **labeled graph** is mathematically represented by its adjacency matrix $\breve{Y} \in \mathcal{A}_{adj} = \mathbb{R}^{k \times k}$ or its flattened adjacency matrix $Y \in \mathcal{A} = \mathbb{R}^{k^2}$. In the case $d > 1$, a labeled graph can be represented by a tensor in the space $\mathbb{R}^{k \times k \times d}$. We use $\breve{Y}$ for the rather rare instances of graphs in adjacency matrix or tensor representation. If there is no edge between two nodes i and j, the edge is assigned zero edge weight, i.e. $a_{ij} = 0_d$ where 0_d denotes the d-dimensional vector of zeros. Thereby, all graphs are treated implicitly as complete graphs to obtain adjacency matrices for labeled graphs which can be used for computations. The graphs defined in this way can include loops, be weighted or unweighted and directed or undirected. In the following, all graphs are assumed to have the same number of nodes $|V| = k$. At the end of this subsection, this assumption and the assumption that the total space $\mathcal{A}$ is an Euclidean space are relaxed and discussed.

Next, we want to obtain mathematical representations of unlabeled graphs as all possible permutations of the adjacency matrices of their labeled graphs. Therefore, we define first permutations of adjacency matrices and permutations of flattened adjacency matrices. Following Guo et al. (2021), permutation of adjacency matrices $\breve{Y} \in \mathcal{A}_{adj}$ can be achieved by permutation matrices P of size $k \times k$ which permute the order of the columns of the adjacency matrix and thus, the order of the nodes. In particular, a permutation matrix has exactly one entry that equals one in each column and each row. Then, we define the set of permutation matrices as $\mathcal{P}$. A group on this set can be obtained by defining the neutral element as the $k \times k$ identity matrix, the group operation as the matrix multiplication and then, the inverse of $P \in \mathcal{P}$ is $P^\top$. Thus, in the case of $d = 1$ the following defines a group action on the set of labeled graphs $\mathcal{A}_{adj} = \mathbb{R}^{k \times k}$:

$$\mathcal{P} \times \mathcal{A}_{adj} \to \mathcal{A}_{adj}, \quad (P, \breve{Y}) \mapsto P \breve{Y} P^\top.$$

The group and its group action for general Euclidean spaces ($d > 1$) can be found in Jain (2016a).

We want to define an analogous group action on the set of flattened adjacency matrices $\mathcal{A}$, i.e. permutation of flattened adjacency matrices. We define the set of all permutations on the flattened adjacency matrices as $\mathcal{T}$ where each element $T \in \mathcal{T}$ is implicitly defined by the corresponding permutation matrix on the adjacency matrices. Then, the group operation is defined implicitly w.r.t. the matrix multiplication of the corresponding elements from $\mathcal{P}$. In summary, the group action of the permutation on the flattened adjacency matrix is implicitly

defined by

$$\cdot : \mathcal{T} \times \mathcal{A} \to \mathcal{A}, \quad (T, Y) \mapsto TY,$$

where TY is the flattened adjacency matrix of $P\check{Y}P^\top$ when $T \in \mathcal{T}$ is the corresponding permutation to $P \in \mathcal{P}$. Hence, for any labeled graph $Y \in \mathcal{A}$ we obtain the corresponding orbit

$$[Y] = \{TY \mid T \in \mathcal{T}\}.$$

The orbits are disjoint and define an equivalence relation $\sim$ by:

$$Y_1 \sim Y_2 \iff \exists\, T \in \mathcal{T} : TY_1 = Y_2.$$

The equivalence classes defined by these equivalence relations have as representative a labeled graph and consist of all labeled graphs that can be obtained through node permutations of the representative. Thus, an **unlabeled graph** can be mathematically represented with the set of all permuted (flattened) adjacency matrices of one of its labeled representatives. This means, an unlabeled graph consists of all labeled graphs with the same attribute structure.

We call the resulting quotient space **graph space** and define it as

$$\mathfrak{G} = \{[Y] \mid Y \in \mathcal{A}\}.$$

Following (Jain 2016a), this space is an orbit space and multiplication and addition between elements of it are not well defined. A more detailed topological and geometrical treatment of the quotient space for $\mathcal{A} = \mathbb{R}^{k^2}_{\geq 0}$ is covered in Kolaczyk et al. (2020).

To end the transition from labeled to unlabeled graphs, we define the projection of a representative onto its equivalence class as

$$\pi : \mathcal{A} \to \mathfrak{G}, Y \mapsto [Y].$$

The projection assigns labeled graphs their unlabeled counterparts and can be used to obtain approximate computations by projecting from the total space $\mathcal{A}$ into the graph space $\mathfrak{G}$. Thus, the total space satisfies a similar role for the graph space as tangent spaces for manifolds as discussed in Calissano et al. (2022).

In a next step, we want to define a metric on $\mathfrak{G}$ to compute averages which is fundamental for the development of further statistical methods. Let $d_{\mathcal{A}}$ be any metric on $\mathcal{A}$. Then, the quotient pseudo-metric

$$d_{\mathfrak{G}}([Y_1], [Y_2]) = \min_{T \in \mathcal{T}} d_{\mathcal{A}}(TY_1, Y_2) \tag{3.1}$$

defines a metric on the graph space $\mathfrak{G}$ which was shown in Calissano et al. (2020). The metric can be understood as follows: we choose representatives Y_1, Y_2 of the unlabeled graphs $[Y_1], [Y_2]$, reorder the flattened adjacency matrix of the first representative to the second through a permutation T^* such that it minimizes the metric $d_{\mathcal{A}}$ in the total space and then, the metric $d_{\mathcal{A}}$

of the aligned graphs T^*Y_1, Y_2 equals the metric $d_{\mathfrak{G}}$ in the graph space. We note that it does not matter if we reorder the first to the second flattened adjacency matrix or vice versa since the group action of $\mathcal{T}$ acts by isometry under $d_{\mathcal{A}}$ and $\mathcal{T}$ is a group. In particular, this follows from the equivalent results on $\mathcal{P}$ in Guo et al. (2021) and

$$\min_{T\in\mathcal{T}} d_{\mathcal{A}}(TY_1, Y_2) = \min_{P\in\mathcal{P}} d_{\mathcal{A}_{adj}}(P\check{Y}_1P^\top, \check{Y}_2) = \min_{P\in\mathcal{P}} d_{\mathcal{A}_{adj}}(P\check{Y}_2P^\top, \check{Y}_1) = \min_{T\in\mathcal{T}} d_{\mathcal{A}}(TY_2, Y_1) \quad (3.2)$$

where $\check{Y}_1, \check{Y}_2 \in \mathcal{A}_{adj}$ are the adjacency matrices corresponding to the flattened adjacency matrices $Y_1, Y_2 \in \mathcal{A}$.

This metric implies the important terms optimal position or optimal alignment as described in Huckemann et al. (2010) for manifolds or in Calissano et al. (2020) for the graph space. We say a labeled graph $TY_1 \in \mathcal{A}$ is in optimally aligned w.r.t. a labeled graph $Y_2 \in \mathcal{A}$ if

$$d_{\mathcal{A}}(TY_1, Y_2) = \min_{\tilde{T}\in\mathcal{T}} d_{\mathcal{A}}(\tilde{T}TY_1, Y_2) = d_{\mathfrak{G}}([Y_1], [Y_2])$$

where the last equality follows directly from the definition of the metric of $\mathfrak{G}$.

Furthermore, we can define (generalized) geodesics in the graph space. We give two definitions for geodesics, the former is used solely to be able to summarize the geodesic regression model in $\mathfrak{G}$ from Calissano et al. (2022) while the latter is used otherwise and defines geodesics between two elements of the graph space. We do not distinguish between the definitions in notation since it is clear from the context which definition is used. For the first definition and following Calissano et al. (2022), assume that $\gamma : \mathbb{R}^p \to \mathcal{A}$ is a line ($p = 1$), plane or hyperplane ($p > 1$) and denote by $\Gamma(\mathcal{A})$ the set of lines, planes or hyperplanes in $\mathcal{A}$. Then, the set of generalized geodesics $\Gamma(\mathfrak{G})$ in $\mathfrak{G}$ is defined as $\Gamma(\mathfrak{G}) = \{[\gamma] = \pi \circ \gamma \mid \gamma \in \Gamma(\mathcal{A})\}$ where π is the projection of an element γ from $\Gamma(\mathcal{A})$ on its equivalence class. In contrast to this, the second definition of a geodesic is obtained by aligning the representatives of two elements of the graph space first and then taking the shortest distance between the two aligned representatives as in Guo et al. (2022). In particular, let $[Y_1], [Y_2] \in \mathfrak{G}$ and T^* the optimal alignment of Y_1 to Y_2. A geodesic between $[Y_1]$ and $[Y_2]$ is $[\gamma] : [0, 1] \to \mathfrak{G}$ where $\gamma : [0, 1] \to \mathcal{A}$ is the shortest path between T^*Y_1 and Y_2. This means, a geodesic in the graph space is the equivalence class of the shortest path between optimally aligned labeled graphs in the total space. We note that the main difference between the definitions can be understood using FDs which are described in the next subsection. Then, we can observe that $\gamma \in \Gamma(\mathcal{A})$ in the first definition does not necessarily stay in one FD and contrary to this, γ as shortest path in $\mathcal{A}$ in the second definition stays in one FD. Furthermore, a related discussion can be found in appendix B and apart from this, the discussion is out of the scope of this thesis. Besides this, the authors in Calissano et al. (2020) were able to show that the graph space is a metric geodesic space. This ensures that for every two graphs in $\mathfrak{G}$ the geodesic exists.

3.1.1. Fundamental domains and uniqueness

Next, we want to introduce briefly the concept of ordinary graphs and their FDs. In this subsection, we follow closely Jain (2016a, section 4.4) and translate some of the results to the

graph space and the notation of the thesis. Ordinary graphs and FDs can be used to analyse the geometry of the graph space from a generic perspective in the topological sense as was done in Jain (2016a) or to obtain results with probability one in a measure theoretic sense. The latter helps us in subsection 3.4 to obtain insights into the convergence results of AAC algorithms.

First, we define the isotropy or stabilizer group of a labeled graph $Y \in \mathcal{A}$ as

$$T_Y = \{T \in \mathcal{T} | TY = Y\}.$$

Then, we define a labeled graph $Y \in \mathcal{A}$ as ordinary graph if $T_Y = \{Id\}$ where Id corresponds to the neutral element of $\mathcal{T}$, i.e. the isotropy group is given by the identity permutation. In this case, we call the isotropy group also trivial stabilizer. This means, there exists no permutation except the identity such that the labeled graph can be represented mathematically with the same adjacency matrix after the permutation. Furthermore, we define a labeled graph $Y \in \mathcal{A}$ as singular graph if the graph is not ordinary.

Following Jain (2016a, proposition 4.17), we define the FD for an ordinary graph $Y \in \mathcal{A}$ as

$$D_Y = \{Z \in \mathcal{A} | d^2_{\mathcal{A}}(Z, Y) \leq d^2_{\mathcal{A}}(TZ, Y) \text{ for all } T \in \mathcal{T}\} \tag{3.3}$$

where $d^2_{\mathcal{A}}$ is assumed in the following to be the squared Euclidean distance. Furthermore, we denote by D^o_Y the interior of the FD and by ∂D_Y the boundary of the FD. Then, it was proven in Ratcliffe et al. (1994, theorem 6.6.13) and translated to the graph space in Jain (2016a, cp. proposition 4.17) that the FD $D_Y \subseteq \mathcal{A}$ is a closed set and:

1. $\mathfrak{G} = \bigcup_{T \in \mathcal{T}} TD_Y \quad \text{where} \quad TD_Y = \{TZ \in \mathcal{A} | Z \in D_Y\}$
2. $TD^o_Y \cap D^o_Y = \emptyset \text{ for all } T \in \mathcal{T} \backslash Id \quad \text{where} \quad TD^o_Y = \{TZ \in \mathcal{A} | Z \in D^o_Y\}.$

Ratcliffe et al. (1994, 6.6.11) showed that there exists a set $F \subseteq \mathcal{A}$ (called fundamental set) with the property that there is precisely one labeled representative $Y \in \mathcal{A}$ of each unlabeled graph $[Y] \in \mathfrak{G}$ in F and $D^o_Y \subseteq F \subseteq D_Y$. Moreover, all labeled graphs in the interior of a FD are ordinary graphs and all singular graphs are on the boundaries of FDs, as was shown in Jain (2016a, Proposition 4.18 4)) and by definition of singular graphs. We note that there might be ordinary graphs on the boundaries of the FDs for graphs with graph attributes in $\mathbb{R}$. A corresponding example can be found in appendix A. Additionally, we note that all labeled graphs in the interior of the FD D_Y of an ordinary graph Y are closer to Y than to any other representative of $[Y]$ and, all labeled graphs on the boundary of the FD D_Y are closest to Y and at least one other representative $\tilde{Y} = TY$ of $[Y]$.

These definitions allow for a detailed investigation of the uniqueness of the permutation that minimizes the metric (3.1) in the graph space and thus, the uniqueness of the metric. The following enhances the discussion in Guo et al. (2021, section 2.1). The permutation $T \in \mathcal{T}$ that minimizes the metric (3.1) between two unlabeled graphs $[Y_1], [Y_2] \in \mathfrak{G}$ is unique if, fixing the representative Y_1 of $[Y_1]$ and assuming that Y_1 is an ordinary graph, the aligned representative of $Y_2 \in D_{Y_1}$ is not on the boundary of the FD of Y_1, i.e. $Y_2 \in D^o_{Y_1}$. Clearly, if one of the graphs is a singular graph, then there exist multiple permutations that minimize the metric. In general,

if $Y_2 \in \partial D_{Y_1}$, there exist at least one permutation for which the distance between Y_1 and Y_2 and TY_2, respectively, are the same (assuming that Y_2 is already aligned to Y_1). An analogous result can be obtained if Y_2 is an ordinary graph and $Y_1 \in \partial D_{Y_2}$ due to the equality in (3.2). If Y_1 and Y_2 are both singular graphs, there does not exist a FD for them by our definition. Then, the metric (3.1) is not unique by definition of singular graphs. Nevertheless, it can be helpful to choose an arbitrary ordinary graph $Z \in \mathcal{A}$ and assume that $Y_1, Y_2 \in D_Z$. Although this is a drawback, we can still achieve uniqueness with probability one under additional assumptions by a more detailed investigation of the boundaries of FDs.

Therefore, we introduce FDs and in particular their boundaries geometrically. As was shown for example in Jain (2016a, Proposition 4.18 1)), the FD of an ordinary graph Z is a convex polyhedral cone and can be written as

$$D_Z = \bigcap_{T \in \mathcal{T}} H_{T,Z}$$

where

$$H_{T,Z} = \{Y \in \mathcal{A} | d_{\mathcal{A}}(Z, Y) \leq d_{\mathcal{A}}(TZ, Y)\} \tag{3.4}$$

are half spaces and their boundaries are hyperplanes in $\mathcal{A}$. Thus, the boundary of D_Z is defined by the finite union over all permutations of the hyperplanes intersected with D_Z, i.e.

$$\partial H_{T,Z} = \bigcup_{T \in \mathcal{T}} \{Y \in \mathcal{A} | d_{\mathcal{A}}(Z, Y) = d_{\mathcal{A}}(TZ, Y)\} \cap D_Z.$$

Next, to obtain uniqueness with probability one of the metric (3.1), we define in general a push forward measure $\mathbb{P}$ on the graph space as $\mathbb{P}(B) = m(\pi^{-1}(B))$ for all B in the σ-algebra of $\mathfrak{G}$ where π is again the projection from the total space into the graph space. The consecutive proof idea is stated more formally in appendix D to obtain an idea of a proof of the convergence of the AAC algorithm for additive models. First, we assume that the graphs $[Y_1], [Y_2]$ are sampled from a probability measure $\mathbb{P}_{[Y]}$ that is absolutely continuous w.r.t. the push forward measure p of the Lebesgue measure m on $\mathcal{A}$. We are interested in the probability to sample unlabeled graphs $[Y_1], [Y_2]$ that have representatives on the boundary of the FDs of each other or that both have representatives that are singular graphs. The probability to sample labeled graphs Y_1, Y_2 on the boundary of the FDs of each other or that both are singular graphs is zero. This is due to the definition of FDs as union of finitely many hyperplanes in equation (3.4). In particular, the boundary of the FD ∂D_{Y_1} corresponding to Y_1 is given by the union of finitely many hyperplanes and we have probability zero to sample Y_2 from these hyperplanes using σ-additivity on the finite union and the absolute continuity w.r.t. the Lebesgue measure of m on the lower dimensional hyperplanes (lower dimensional w.r.t. $\mathcal{A}$). Then, since there exist only finitely many FDs corresponding to the representatives of $[Y_1]$, the probability of all of these FDs in m is zero and therefore, the push forward measure is also zero. This follows analogously, if we consider the case that Y_1, Y_2 are singular graphs and Z is an arbitrary ordinary graph. Thus, the probability to sample graphs with representatives on the boundary of FDs is zero in the case

$\mathcal{A} = \mathbb{R}^{k^2}$ and for the defined probability measures. This leads to almost sure uniqueness of the metric (3.1) in this case.

Now, we illustrate what happens if we weaken the assumptions. A typical assumption would be to assume graph attributes in $\mathbb{R}_{\geq 0}$ as investigated in Kolaczyk et al. (2020). There, one has to decide if the coordinate hyperplanes should be included in the boundaries of FDs. Then, also ordinary graphs on the coordinate hyperplanes could be on these types of boundaries. If we do not include the coordinate hyperplanes as boundaries of the FDs but instead in the interior of the FDs, Kolaczyk et al. (2020, Proposition 4.2) proves that only singular graphs are on the boundaries of arbitrary FDs for graphs in the restricted graph attribute space $\mathbb{R}_{\geq 0}$. As long as we assume again that the probability measures are absolute continuous w.r.t. the Lebesgue measure, this still leads to almost sure uniqueness. In contrast to this, we could allow for discrete graph attributes and a corresponding probability measure assigning positive probability to single numbers. Accordingly, we obtain a positive probability for singular graphs. Then, the minimizer in (3.1) is not unique with probability larger than zero. This happens also if we assume positive probability on just one number such as zero, i.e. if the probability measure on $\mathcal{A}$ is not absolutely continuous w.r.t. the Lebesgue measure anymore. This should be considered when we add null nodes or even more importantly, when we extend graphs to complete graphs by adding edges with null attribute. In summary, we see that the assumption of absolute continuity w.r.t. to the Lebesgue measure in the total space is critical for almost sure uniqueness of the metric (3.1).

Finally, we give some more examples for the usefulness of the definition of ordinary graphs and FDs. In Calissano et al. (2022, theorem 1), the concept is used indirectly to prove almost sure convergence of the AAC algorithm given in algorithm 1. Similarly, this is done in Calissano et al. (2020) for the AAC algorithms for mean and principal components in the graph space. Besides, the concept is used in Jain (2016b, section 4.3) to derive sufficient conditions for uniqueness of the population and sample mean. Therein, the automorphism group corresponds to our isotropy group and asymmetric graphs relate to ordinary graphs while symmetric graphs correspond to singular graphs. In a similar spirit, in Kolaczyk et al. (2020) ordinary graphs are called distinct graphs and FDs are utilized to understand the quotient space structure of the graph space. Then, FDs are used to show almost sure consistency of the sample Fréchet mean to a corresponding population Fréchet mean. Moreover, uniqueness of the Fréchet mean is investigated inside a subset of FDs. In contrast to this, Guo et al. (2021, section 2.1) discusses the uniqueness of the metric in the graph space informally w.r.t. a total space with continuous and discrete elements, respectively, while referring indirectly to ordinary graphs. This is slightly comparable to the our discussion above. Lastly, we use FDs in subsection 3.4 and appendix D in a discussion on the developed regression frameworks and to investigate the convergence of algorithm 2. Additionally, we examine in appendix B the regression functions in the graph space utilizing FDs.

3.1.2. Attribute space and null nodes

Ultimately, we want to discuss the assumption that the total space is an Euclidean space and the assumption of equal size of all graphs, i.e. $|V| = k$.

Although we chose an Euclidean space $\mathbb{R}^{k^2 d}$ as total space $\mathcal{A}$, one can define $\mathcal{A}$ more generally.

More general formulations can be found e.g. in Guo et al. (2021) where the attribute space is allowed to be any Riemannian manifold on which distances, covariances and averages can be defined. This generalization allows for example for modeling shapes as edges. An application on brain arterial networks can be found in Guo et al. (2022); Guo & Srivastava (2020). There, arteries in the networks are modeled as shapes which are again modeled as equivalence classes in a similar fashion as unlabeled networks and labeled networks in the current framework, where the set of permutations is replaced by a set of rotations. The joint group action of permutation and rotation leads again to a quotient space. If there exists no edge between two nodes, the edge is assigned the null attribute of the total space $\mathcal{A}$. Geodesics between two points can be defined by geodesics between two points in the total space $\mathcal{A}$ and their respective projections. The authors call the resulting quotient space elastic graph shape space.

The assumption of equal size of the graphs can be relatively easily weakened by adding null nodes to the graphs. Null nodes are nodes with null attribute as node attribute and null attribute as edge weights to all other nodes, i.e. isolated nodes with node attributes equal to the null attribute of the graph attributes. To illustrate this, let $Y_1, Y_2 \in \mathcal{A}$, the number of nodes of Y_1 be k_1, the number of nodes of Y_2 be k_2 and $k_2 < k_1$. We could add null nodes in two ways: i) add $k_1 - k_2$ null nodes to Y_2 as e.g. in Calissano et al. (2022); Jain & Obermayer (2012); Jain (2016a) or ii) add k_2 null nodes to Y_1 and k_1 null nodes to Y_2 such that both graphs have $k_1 + k_2$ nodes as e.g. in Guo et al. (2021). In general, the former would be less precise if we have nodes that do not correspond structurally to other nodes in Y_1 and Y_2. The latter is more computationally expensive. To be more precise, we can choose the relevant option problem specific in the following way: If there exists at least one observation $[Y_j] \in \mathfrak{G}$ with all possible nodes, it is enough to add null nodes as described in i). Contrary, let $|Y_j| = k_{\max}$ be the observation with the most nodes. If there might exist another graph $[Y_l]$ with $|Y_l| \leq k_{\max}$ but there are nodes in $[Y_l]$ that do not correspond to existing nodes in $[Y_j]$ but instead correspond to null nodes not yet added to $[Y_j]$, then one should add null nodes as described in ii).

In summary, this subsection introduced the graph space $\mathfrak{G}$ as quotient space, a metric on $\mathfrak{G}$, geodesics on $\mathfrak{G}$ and the concept of optimal alignment. Then, we introduced FDs and analysed uniqueness of the metric in $\mathfrak{G}$. Finally, the assumptions of Euclidean graph attributes and the addition of null nodes were considered. We emphasize again the transition from labeled to unlabeled graphs, i.e. from $\mathcal{A}$ to $\mathfrak{G}$ by the projection π. Its motivation and implications on the investigated objects are discussed in more detail in section 4.

3.2. Linear regression in the graph space

This subsection reviews the linear regression framework introduced in Calissano et al. (2022). First, we define the Fréchet mean in the graph space and review the least squares minimization problem for a multi output regression model with J-dimensional real valued output. Next, the regression model in the graph space is defined and the family of regression functions is described. For that purpose, we introduce the term optimal alignment w.r.t. regression functions. Finally, we describe the AAC algorithm and state its convergence result. For the purpose of readability, we restrict this subsection to one-dimensional graph attributes and the flattened adjacency matrix

representatives of labeled graphs, i.e. $\mathcal{A} = \mathbb{R}^{k^2}$. For higher dimensional graph attributes, replace the flattened adjacency matrices by the vectorizations of the tensor representations of the labeled graphs.

Since addition between arbitrary elements in the graph space is not well defined, we define the Fréchet mean to obtain a definition of a sample mean in $\mathfrak{G}$ and a first statistical tool in the graph space. Following Fréchet (1948), the empirical Fréchet mean of a sample $\{[y_1], \dots, [y_n]\} \in \mathfrak{G}$ can be defined as minimizer of the function

$$F_n([y]) = \sum_{i=1}^{n} d^2_{\mathfrak{G}}([y_i], [y]).$$

This means, the Fréchet mean is defined as:

$$[\bar{y}] \in \arg\min_{[y]\in\mathfrak{G}} \sum_{i=1}^{n} d^2_{\mathfrak{G}}([y_i], [y]).$$

Similar to the empirical Fréchet mean as a sample mean in the graph space, we can develop a linear regression model using the metric (3.1) in the graph space. The idea is to define the linear regression function f as minimizer of the sum of squared distances in $\mathfrak{G}$ between observations $[y_i] \in \mathfrak{G}, i = 1, \dots, n$ and the regression function $f(x_i) \in \mathfrak{G}, i = 1, \dots, n$ evaluated at their corresponding covariate values. As preface and to connect the regression model to known concepts, we review the multi output regression model with target in the Euclidean space as minimizer of a least squares problem.

Following Fletcher (2013), the multi output linear model with explanatory variables $\mathbf{X} \in \mathbb{R}^p$ and target variables $Y \in \mathbb{R}^J$ can be defined as

$$Y = f(\mathbf{X}) + \varepsilon \tag{3.5}$$

where $f : \mathbb{R}^p \to \mathbb{R}^J,\ x \mapsto (1, x^\top)\beta$ is an affine function through $\mathbb{R}^J$ depending on an unobservable parameter vector $\beta = ((\beta_{0,1}, \dots, \beta_{p,1})^\top, \dots, (\beta_{0,J}, \dots, \beta_{p,J})^\top) \in \mathbb{R}^p \times \dots \times \mathbb{R}^p$ and $\varepsilon \in \mathbb{R}^J$ is an unobservable error vector. For a sample $(x_i, y_i) \in \mathbb{R}^p \times \mathbb{R}^J, i = 1, \dots, n$ of (3.5), we obtain the least squares estimate of the regression function f as minimizer of

$$\sum_{i=1}^{n} d^2_{\mathbb{R}^J}(y_i, f(x_i))$$

where $d^2_{\mathbb{R}^J}$ is the squared Euclidean distance in $\mathbb{R}^J$ and f depends implicitly on β. This model was extended to a regression model with target in manifolds using exponential maps in Fletcher (2013). In the following, we extend this model and the corresponding least squares estimation to responses living in the graph space.

In particular, we want to derive a similar minimization problem in $\mathfrak{G}$ using the projection π from the total space onto the graph space in a similar way as exponential maps. Since $\mathfrak{G}$ is not a manifold, we cannot use the results from Fletcher (2013) and related literature on geodesic regression on manifolds. In our setting, we observe unlabeled graphs together with covariates,

this means we observe a sample $\{(x_1, [y_1]), \ldots, (x_n, [y_n])\} \in \mathbb{R}^p \times \mathfrak{G}$. Following Calissano et al. (2022), the goal is to find a function

$$f : \mathbb{R}^p \to \mathfrak{G}$$

which minimizes

$$\sum_{i=1}^{n} d^2_{\mathfrak{G}}([y_i], f(x_i)). \tag{3.6}$$

This means, we minimize the sum of squared distances between the observed unlabeled graph and the fitted unlabeled graphs at their corresponding covariate. In other words, we minimize the sum of squared prediction errors in the graph space.

So far, we did not define the family of functions over which we minimize. In the following, we define the family of functions as generalized geodesics first and extend it to a larger family by allowing for possibly non-linear, continuous basis functions for the graph attributes in total space afterwards. Thereby, we follow closely Calissano et al. (2022) and all results are summarized from there. In subsection 3.4, we allow for smooth functions as regression functions for the graph attributes in the total space and optimize similar objectives as in (3.6). The extensions allow step wise for more flexible regression models in the graph attribute, hence in the total space $\mathcal{A}$ and thus, more flexible regression functions in the graph space $\mathfrak{G}$.

We start by introducing the **geodesic regression model** in $\mathfrak{G}$. Therefore, let $\mathcal{J} = \{1, \ldots, J\}, J = k^2$ be the set of node and edge attribute indices w.r.t. the flattened adjacency matrix of labeled graphs. We define the regression model as

$$f : \mathbb{R}^p \to \mathfrak{G}, \quad x \mapsto f(x)$$

where $f(x) = \pi \circ h(x)$ and π is defined as the projection from the total space to the graph space as in subsection 3.1. We model $h : \mathbb{R}^p \to \mathcal{A}$ as a multi output regression model where we regress linearly all outputs separately on the same covariates. This means we obtain in the total space the regression model

$$\begin{aligned} h(x) = h_\beta(x) &= (h_{\beta_a}(x))_{a \in \mathcal{J}} \in \mathcal{A} \\ h_{\beta_a}(x) &= (1, x^\top)\beta_a \in \mathbb{R} \end{aligned}$$

where $\beta_a = (\beta_{0,a}, \ldots, \beta_{p,a})^\top, a \in \mathcal{J}$ are unknown parameter vectors for each graph attribute a. We usually omit the dependence of h on β in the notation, such that we write from now on $f = \pi \circ h = \pi \circ (h_a)_{a \in \mathcal{J}}$. We note that h and $\beta_{j,\cdot}, j = 0, \ldots, p$ take values in $\mathcal{A}$ and f takes values in $\mathfrak{G}$. Furthermore, since h is defined as a line ($p = 1$), plane or hyperplane ($p > 1$) in $\mathcal{A}$, f is a generalized geodesic and we write $f \in \Gamma(\mathfrak{G})$. Thus, the family of functions over which we aim to minimize the empirical error (3.6) is $\Gamma(\mathfrak{G})$.

Next, we want to allow for a more flexible regression model and enlarge the family of functions that we minimize (3.6) over. Therefore, we extend the functions that are allowed for $h_a, a \in \mathcal{J}$.

Let $h : \mathbb{R}^p \to \mathcal{A}, x \mapsto h(x) = (h_a(x))_{a\in\mathcal{J}}$ where $h_a : \mathbb{R}^p \to \mathbb{R}$ can be any function of the form

$$h_a(x) = \sum_{j=0}^{p} \phi_j(x)\beta_{j,a} \tag{3.7}$$

where $\beta_{j,a}$ are unknown parameters corresponding to basis function $j = 0, \ldots, p$ and graph attribute $a \in \mathcal{J}$, $\phi_0(x) = 1$ is the basis function for the intercept and $\phi_j : \mathbb{R}^p \to \mathbb{R}, j = 1, \ldots, p$ can be basis functions that are non-linear and continuous. For a sample $(x_i, [y_i]) \in \mathbb{R}^p \times \mathfrak{G}, x_i = (x_{i1}, \ldots, x_{ip})^\top, i = 1, \ldots, n$, the regression model in the total space can be written with intercept in matrix formulation as

$$\begin{bmatrix} y_{1,1} & \ldots & y_{1,J} \\ \vdots & \ddots & \vdots \\ y_{n,1} & \ldots & y_{n,J} \end{bmatrix} = \begin{bmatrix} 1 & \phi_1(x_1) & \ldots & \phi_p(x_1) \\ \vdots & \ddots & & \vdots \\ 1 & \phi_1(x_n) & \ldots & \phi_p(x_n) \end{bmatrix} \begin{bmatrix} \beta_{0,1} & \ldots & \beta_{0,J} \\ \vdots & \ddots & \vdots \\ \beta_{p,1} & \ldots & \beta_{p,J} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,1} & \ldots & \varepsilon_{1,J} \\ \vdots & \ddots & \vdots \\ \varepsilon_{n,1} & \ldots & \varepsilon_{n,J} \end{bmatrix} \tag{3.8}$$

where $y_i = (y_{i,1}, \ldots, y_{i,J}), i = 1, \ldots, n$ are representatives of $[y_i], i = 1, \ldots, n$ and $(\varepsilon_{1,a}, \ldots, \varepsilon_{n,a})^\top$ are independent and identically, normally distributed error terms with conditional mean zero and constant variance, independent of each other for each graph attribute $a \in \mathcal{J}$. The error terms and the regression models on graph attribute level of the total space are more specifically defined in the classical linear model in the next subsection 3.3.

Finally, the multi output regression model can be projected into the graph space as before. In particular, we define

$$f : \mathbb{R}^p \to \mathfrak{G}, x \mapsto f(x) = \pi \circ h(x)$$

where h is allowed to be functions as defined above in (3.7). We write $f \in \mathcal{F}(\mathfrak{G})$. We observe that $\Gamma(\mathfrak{G}) \subset \mathcal{F}(\mathfrak{G})$ These models were first described in Calissano et al. (2022), are called here **linear regression in $\mathfrak{G}$** and the function f defined in this way minimizing (3.6) is called linear regression function in $\mathfrak{G}$. Following Borchani et al. (2015), we refer to separate regression models on each graph attribute as in (3.7) as single target (regression) models in a multi output regression model.

Align all and compute algorithm

To be able to estimate the regression function f, we define first the concept of optimal alignment w.r.t. a regression function similar to optimal alignment w.r.t. a graph. For this, the graphs are optimal aligned w.r.t. the regression function evaluated at their corresponding covariate values. Thus, for one observation and a regression function, we are again in the case of optimally aligning two graphs. To be precise, a representative of an unlabeled graph is optimally aligned w.r.t. a regression function if the representative has minimal distance in the total space to the function evaluated at the corresponding covariate value. Following Calissano et al. (2022), let $(x, [y]) \in \mathbb{R}^p \times \mathfrak{G}$, $f \in \mathcal{F}(\mathfrak{G})$ be a regression function in the graph space with corresponding regression function h in the total space and $T \in \mathcal{T}$ a permutation of the representative of $[y]$.

Then, we call a representative $Ty \in \mathcal{A}$ optimally aligned w.r.t. f or equivalently h in $\mathcal{A}$ if

$$d^2_{\mathcal{A}}(Ty, h(x)) = d^2_{\mathfrak{G}}([y], f(x)).$$

Now we can describe the AAC algorithm to obtain the regression function f. The algorithm was first described in Calissano et al. (2022) after similar algorithms for the computation of the sample Fréchet mean and geodesic principal components were developed in Calissano et al. (2020) which were inspired by the generalized procrustes analysis developed in Gower (1975).

The idea of the algorithm is as follows: in an initialization step, all observations are aligned w.r.t. a randomly chosen representative of one observation in the sample, yielding an aligned sample $(x_1, T_1 y_1), \ldots, (x_n, T_n y_n) \in \mathbb{R}^p \times \mathcal{A}$. Then, the regression function h in the total space is computed by minimizing

$$\sum_{i=1}^{n} d^2_{\mathcal{A}}(T_i y_i, h(x_i)). \tag{3.9}$$

We allow for regression functions as defined in (3.7) for each graph attribute and thus, this corresponds to separate minimization of J single target regression least-square objectives

$$\sum_{i=1}^{n} d^2_{\mathbb{R}}(t_i y_{i,a}, h_a(x_i))$$

where $t_i y_{i,a}$ is the a-th graph attribute of observation $T_i y_i, i = 1, \ldots, n$. The obtained linear regression function h is projected into the graph space and a regression function $f = \pi \circ h$ is obtained. Afterwards, all observations are aligned w.r.t. the regression function f. The steps of alignment and regression are repeated till the algorithm converges. We say that the algorithm has converged if the distance of the sum of squared prediction errors between the regression function of the last step $\tilde{f}$ and the regression function in the current step f falls below a threshold. The algorithm is given in algorithm 1 and described in the following in more detail.

Overall, the algorithm can be split into two parts, one for initialization and one that repeats the alignment and regression until convergence. The initialization part executes an initial alignment of the representatives of the unlabeled graphs and a regression on these representatives. Afterwards, the representatives are aligned w.r.t. the regression function. This is done to obtain an initial prediction error which is needed in the first step of the second part of the algorithm. Afterwards, the algorithm repeats regressing the aligned representatives on the covariates and the alignment of the representatives w.r.t. the current regression function. The problem of aligning graphs w.r.t. the regression function is analogous to the problem of aligning two graphs with each other multiple times.

The problem of alignment of graphs is known as NP-complete graph matching problem, see e.g. Calissano et al. (2020). There exist multiple algorithms which solve this problem approximately (Jain 2016b), e.g. the graduate assignment algorithm (Gold & Rangarajan 1996) or the generalized Bron Kerbosch algorithm (Jain & Obermayer 2011) which are both implemented in the graph space package in python. Since the algorithm relies on inexact graph matching this means that the alignment might be incorrect and the convergence of algorithm 1 only holds approximately.

Nevertheless, similar algorithms also rely on this inexact graph matching as stated in (Calissano et al. 2020).

Algorithm 1: AAC algorithm to compute the linear regression function $f \in \mathfrak{G}$ from Calissano et al. (2022)

Data: $(x_1, [y_1]), \ldots, (x_n, [y_n]) \in \mathbb{R}^p \times \mathfrak{G}$, thresholds κ, δ
Result: $f \in \mathfrak{G}$
Randomly pick one graph $[y_l] \in \mathfrak{G}$ and its representative $T_l y_l \in \mathcal{A}$;
Align the representatives $T_j y_j, j \in \{1, \ldots, n\} \backslash l$ of all other graphs to $T_l y_l$;
Obtain h from a regression of $(x_1, T_1 y_1), \ldots, (x_n, T_n y_n)$ in $\mathcal{A}$ minimizing (3.9);
Set $\tilde{f} = \pi \circ h$;
Align all representatives w.r.t. h obtaining $\tilde{T_1} y_1, \ldots, \tilde{T_n} y_n$;
while $\delta > \kappa$ **do**
  Obtain h by minimizing (3.9) for $(X_1, \tilde{T_1} y_1), \ldots, (X_n, \tilde{T_n} y_n)$ and $f = \pi \circ h$;
  Align all representatives w.r.t. f obtaining $\tilde{T_1} y_1, \ldots, \tilde{T_n} y_n$;
  Compute the distance of the sum of squared prediction errors $\delta = \Delta(\tilde{f}, f)$;
  Set $\tilde{f} = f$;
end
Return $f = \tilde{f}$.

One major argument in favor of the AAC algorithm given in algorithm 1 is that the authors in Calissano et al. (2022) showed that it terminates in finite time and that it converges to a local minimum of the empirical error (3.6) viewed as a function of the regression parameters β with probability one under additional assumptions. In particular, following Calissano et al. (2022, theorem 1), let $\mathbb{P}_{[Y]}$ be a probability measure on $\mathfrak{G}$ and $\mathbb{P}_X$ be a probability measure on $\mathbb{R}^p$. Furthermore, let P_X be absolutely continuous w.r.t. the Lebesgue measure on $\mathbb{R}^p$ and let $P_{[Y]}$ be absolutely continuous w.r.t. the push forward measure p of the Lebesgue measure m on $\mathcal{A}$. We assume that $(x_1, [y_1]), \ldots, (x_n, [y_n]) \in \mathbb{R}^p \times \mathfrak{G}$ is sampled from the distribution $\mathbb{P}_X \times \mathbb{P}_{[Y]}$.

Additionally, we assume algorithm 1 fits the linear regression in $\mathfrak{G}$, where we allow for regression functions as defined in (3.7) in $\mathcal{A}$ and their corresponding projections $f = \pi \circ h$ in $\mathfrak{G}$. For the basis functions in (3.7) we assume the following: they include an intercept; if we sample x from $\mathbb{P}_X$ and we have two different parameter vectors β and $\tilde{\beta}$, the resulting h_{β_a} and $h_{\tilde{\beta}_a}$ are unequal with probability one; the matrix in (3.8) has full rank. Then, algorithm 1 stops in finite time and the estimated regression function is a local minimum of the least squares objective (3.6) as a function of β with probability one. A proof of the result can be found in Calissano et al. (2022). The proof, the introduced probability measures and the reasonability of the assumptions can become more clear from the idea of an extension of the proof to additive models in appendix D, the discussions on FDs in subsection 3.1.1 and 3.4 and the discussion in section 4. Moreover, we note that the convergence holds up to numerical imprecision through inexact graph matching algorithms.

In summary, this subsection reviewed the linear regression model in $\mathfrak{G}$ and the corresponding AAC algorithm to fit the model. The aim of the next subsection will be to introduce generalized additive models for one-dimensional responses. Afterwards, the single target regression models of the current subsection can be extended to generalized additive models and we obtain an additive

and an idea of a generalized additive regression framework by projecting the corresponding multi output models into the graph space. While the extensions correspond to a similar extension of the family of function as was done above from $\Gamma(\mathfrak{G})$ to $\mathcal{F}(\mathfrak{G})$, the objectives which are optimized differ from (3.6).

3.3. Univariate response regression models

This subsection reviews additive and generalized additive models (GAMs) for one-dimensional responses, which were first introduced in Friedman & Stuetzle (1981) and Hastie & Tibshirani (1986), respectively. The aim is to present concisely the transition from the linear regression model to GAMs while introducing least squares and likelihood based estimation procedures to fit additive models and GAMs. Therefore, we describe briefly the classical linear regression model and its extensions to generalized linear models, linear mixed models, generalized linear mixed models and smoothers. On the way, we integrate the penalized iterative (re-)weighted least squares algorithm to estimate model parameters and the restricted maximum likelihood (REML) estimator to be able to estimate smoothing parameters. Finally, we arrive at the optimization objectives and their estimation procedures for additive models and GAMs. In the next subsection, the models are used as single target regression models of the graph attributes and the objectives are translated to a multi output setting in the total space to obtain afterwards an additive and a generalized additive regression framework in $\mathfrak{G}$. For a more detailed description, we refer the reader to Fahrmeir et al. (2013); Wood (2017) where all results of this subsection are summarized from, unless otherwise stated. For clarity, we usually do not cite them throughout the text. Instead, we note that the classical linear model is summarized from Fahrmeir et al. (2013, chp. 3), generalized linear models from Fahrmeir et al. (2013, chp. 5.4, 5.8) and Wood (2017, chp. 3.1, 3.2), linear and generalized linear mixed models from Fahrmeir et al. (2013, chp. 7) and especially Fahrmeir et al. (2013, chp. 7.6) for REML estimation, smoothers from Fahrmeir et al. (2013, chp. 8) and Wood (2017, chp. 5) and ultimately, additive models and GAMs from Fahrmeir et al. (2013, chp. 9) and Wood (2017, chp. 5, 6).

In this subsection, we are in general interested in the relationship of an one-dimensional response variable Y and a $(p+1)$-dimensional covariate vector $\mathbf{X}$. In contrast to the multi output model in (3.5), we stress that the response in this subsection is one-dimensional. To model the relationship, we can decompose the response variable to

$$Y = f(\mathbf{X}) + \varepsilon$$

where ε is a random error term and f is the function of interest. To be able to estimate the function f, we assume to observe independently $(y_i, x_i), x_i = (1, x_{i1}, \ldots, x_{ip})^\top, i = 1, \ldots, n, n \in \mathbb{N}$ from the response Y and covariate vector $\mathbf{X}$. All in all, we work with a random response vector $y = (y_1, \ldots, y_n)^\top$, an random error vector $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^\top$ and a random design or model matrix $X = (x_1, \ldots, x_n)^\top$ and the corresponding observations for estimation.

The **classical linear regression model** assumes a linear relationship between Y and $\mathbf{X}$, and,

consequently, between y and X, i.e.

$$\begin{aligned} \text{for } i = 1, \ldots, n \qquad & y_i = x_i^\top \beta + \varepsilon_i \\ \text{or in matrix notation} \qquad & y = X\beta + \varepsilon \end{aligned} \tag{3.10}$$

where β is an unobservable $(p+1)$-dimensional parameter vector. Estimation of the function f corresponds to estimation of the parameter vector β. The model assumes additivity and linearity in the covariates. Additionally, we assume that X has full rank $p+1$. This implies that $p+1 \leq n$. Furthermore, $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^\top$ is assumed to be a vector of independent and identically distributed (i.i.d.) random variables with, conditional on X, mean zero and constant variance σ^2. Moreover, we assume that the errors are normally distributed conditional on X. The assumptions on the errors can be summarized by $\varepsilon|X \sim \mathcal{N}(0_n, \sigma^2 \mathbb{I}_n)$ where 0_n is the n-dimensional vector of zeros and $\mathbb{I}_n$ is the $n \times n$ identity matrix. Thus, we obtain for the conditional distribution of y given X:

$$y|X \sim \mathcal{N}(X\beta, \sigma^2 \mathbb{I}_n).$$

The estimation of the parameter vector β in the classical linear regression model can be accomplished using a least squares (LS) approach by minimizing the LS objective

$$LS(\beta) = (y - X\beta)^\top (y - X\beta) = \sum_{i=1}^{n} (y_i - x_i^\top \beta)^2. \tag{3.11}$$

Besides, we could utilize the distributional assumption on the response deriving a maximum likelihood estimator (MLE). Therein, given σ^2, we maximize the log-likelihood w.r.t. β

$$\ell(\beta) \propto -\frac{1}{2\sigma^2} (y - X\beta)^\top (y - X\beta)$$

where $\propto$ denotes that equality holds up to additive constants. Minimization of the LS objective and maximization of the log-likelihood w.r.t. β yield the same estimator for the parameter vector:

$$\hat{\beta} = (X^\top X)^{-1} X^\top y.$$

Finally, we can observe that the function f is modeled by modeling the conditional mean of y, i.e. we model $f(X) = E[y|X] = X\beta$. This holds not only for the classical linear regression model, but also for its extensions. In general, the type of regression models analysed in this thesis can be called mean regression models and could be extended to e.g. quantile regression models described in Fahrmeir et al. (2013, chp. 10). In the following, we will switch between the model formulations $y = f(X) + \varepsilon$ and a direct model for $E[y|X]$ to obtain extensions of the classical linear model. The former makes distributional assumptions on the error term, while the latter assumes directly distributions on the response.

3.3.1. Generalized linear models

A first extension of model (3.10) is to allow for non-normal responses and some amount of non-linearity using distributions of the exponential family (EF) and a link function between the conditional mean of the response and the linear predictor $\eta_i = x_i^\top \beta$. This leads to so called **generalized linear models** (GLMs) which were developed first in Nelder & Wedderburn (1972).

First, we define briefly distributions of the EF to obtain a general class of distributions for the response. We say a variable y is distributed with an univariate distribution of the EF if its density can be written in the form

$$f(y|\theta) = \exp\left(\frac{y\theta - b(\theta)}{\phi} w + c(y, \phi, w)\right) \tag{3.12}$$

where θ is called natural parameter, ϕ is some dispersion or scale parameter, w is a known value, b is a function of θ such that $f(y|\theta)$ can be normalized and the first two derivatives $b'(\theta)$ and $b''(\theta)$ exist and finally, c is a function that does not depend on θ and can therefore be neglected here. In the following, we omit the conditioning on the natural parameter and write $f(y) = f(y|\theta)$ and $y \sim \mathrm{EF}(\mu, \phi)$ to denote that y is distributed with a distribution of the EF with mean μ and dispersion parameter ϕ. The EF contains many different distributions as special cases, for example the Bernoulli, Poisson, normal and gamma distribution.

Next, we define GLMs. From here on, we write $\mu_i = E[y_i|x_i],\ i = 1, \ldots, n$. Then, we assume for a **GLM** that the response variables are conditionally independent and distributed with $y_i|x_i \sim EF(\mu_i, \phi),\ i = 1, \ldots, n$ and we model the conditional mean as

$$\mu_i = h(x_i^\top \beta), \quad i = 1, \ldots, n$$

where h is an one-to-one and twice differentiable function of the linear predictor. For the so called link function $g = h^{-1}$ we can rewrite this into

$$g(\mu_i) = x_i^\top \beta, \quad i = 1, \ldots, n. \tag{3.13}$$

In summary, we determine a particular GLM by a distribution from the EF, a link function and covariates. When we choose the identity link for g and a normal distribution for the response variables, we obtain again the classical linear model (3.10).

Finally, the model can be estimated by its MLE. First, we derive the log-likelihood and its connection to the model parameters β. From the general form for a density of EFs in (3.12), the log-likelihood for GLMs is given by

$$\ell(\beta) \propto \sum_{i=1}^{n} \frac{y_i\theta_i - b(\theta_i)}{\phi} w_i \tag{3.14}$$

where we utilized the conditional independence of the y_i to obtain the sum. We note that the dependence of the log-likelihood on β is implicitly given by the dependence of the natural parameters θ_i on β due to $h(x_i^\top \beta) = \mu_i = b'(\theta_i)$. The equality can be proven by taking the first derivative of the log-likelihood function of the natural parameter w.r.t. θ and utilizing the

result that the expectation of the first derivative of the log-likelihood equals zero as explained in Wood (2017, chp. 3.1.1). Since $h(x_i^\top \beta)$ models the conditional mean, this results in the desired equation. The connection $h(x_i^\top \beta) = b'(\theta_i)$ is important since it offers the possibility to link the model parameters β to the canonical parameters of EFs.

However, the MLE of (3.14) for GLMs cannot be derived in general analytically. Instead, one can use an iterative, numeric procedure for estimation called iterative (re-)weighted least squares (IRLS) algorithm based on the Newton-Raphson method or similarly, the Fisher scoring algorithm. A description of the IRLS algorithm can be found in appendix C.

Generalized linear mixed models

A second extension of model (3.10) are linear mixed models (LMMs). The aim of this part is to introduce generalized linear mixed models (GLMMs) and briefly relate the penalized IRLS algorithm and REML estimation to them. This is of interest to us since the parameters of smooth terms in additive models and GAMs can be treated as random effects and then, estimation of the smoothing parameters of these models can be based on the estimation of the covariance matrix of the random effects. Nevertheless, we keep this part as short as possible since it is closer related to the estimation and especially implementation of the models. For the explicit connection between the choice of smoothing parameters for additive models and LMMs, we refer the reader e.g. to Fahrmeir et al. (2013, chp. 8.1.9). For a recent REML estimation of the smoothing parameters, we refer the reader to Wood (2011).

For **LMMs**, we rewrite model (3.10) to

$$y = Z\gamma + Ub + \varepsilon,$$

where y is a response vector, Z and U are design matrices, γ is a parameter vector of fixed effects, b one of random effects; additionally, we assume conditional on Z

$$\begin{pmatrix} b \\ \varepsilon \end{pmatrix} | Z \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} G & 0 \\ 0 & R \end{pmatrix} \right)$$

where G and R are covariance matrices.

As before, we can extend LMMs to **GLMMs** to account for non-normal responses by

$$g(\mu_i) = z_i^\top \gamma + u_i^\top b \tag{3.15}$$

where we assume conditional independent responses with $y_i | z_i, b_i \sim \text{EF}(\mu_i, \phi), i = 1, \ldots, n$, Z, U, γ, b are defined as for the LMMs, z_i and u_i are rows of the model matrices Z and U, respectively, and we assume again

$$b | Z \sim \mathcal{N}(0, G)$$

where G is a covariance matrix.

The parameter vectors γ and b from the GLMMs (3.15) can be estimated, given the covariance

matrix G, by maximizing the penalized log-likelihood

$$\ell_{\text{pen}}(\gamma, b) = \ell(\gamma, b) - \frac{1}{2} b^\top G^{-1} b \tag{3.16}$$

where $\ell(\gamma, b)$ is defined as the log-likelihood for GLMs in (3.14) with $\eta_i = z_i^\top \gamma + u_i^\top b$ instead of $\eta_i = x_i^\top \beta$. The penalty term $\frac{1}{2} b^\top G^{-1} b$ in the penalized log-likelihood arises due to the normal assumption on the vector of random effects. As for GLMs, we cannot derive a general MLE for γ and b from GLMMs analytically. However, the penalized log-likelihood can be maximized numerically similarly as for GLMs while including the penalty term in each iteration. In particular, the corresponding penalized IRLS (PIRLS) algorithm maximizing numerically (3.16) given the covariance matrix G can be found in appendix C. Estimation of the covariance matrix G can be achieved by maximizing a restricted maximum likelihood. This is also described in more detail in appendix C. Then, one can e.g. update in each iteration of the PIRLS algorithm the corresponding variance matrix using the REML. This is presented in more detail e.g. in Breslow & Clayton (1993).

3.3.2. Smoothers

Another approach to extend model (3.10) is to allow for smooth functions of the covariates, weakening the linearity assumption. Therefore, it is necessary to introduce so called smoothers to allow for the smooth terms in additive and generalized additive models. Furthermore, we explicitly introduce P-splines used in subsection 4.2 to derive an intrinsic penalty term in the graph space and cubic splines together with tensor product smooths used in the implementation of the application in section 5.

Initially, we restrict to

$$y_i = f(x_i) + \varepsilon_i$$

where f is a smooth function of an one-dimensional covariate and the error terms satisfy the same assumptions as for the classical linear model in (3.10). This will be extended to possibly multivariate covariates in the second part of this subsection using tensor product smooths.

In general, we model the smooth function f in two steps. In the first step, we define a basis representation of the smooth function. This can be achieved using splines of a specific degree and a number of knots which partition the domain of the function for a corresponding covariate. Examples for such basis representations are the truncated power series basis or the B-spline basis. Given such a basis representation, the smoothness of the function is determined by the number of knots, the position of the knots, and the degree of the splines. In a second step, one defines a measure of wiggliness. This basis representation would lead w.r.t. estimation e.g. to a LS objective. Then, one usually allows for overly flexible functions in the basis representations and adds the wiggliness as penalty term to the LS objective leading to a penalized least squares (PLS) objective. This can be summarized as approximating a smooth function while penalizing for its wiggliness. To concretize this, we describe first the procedure for penalized B-splines (P-splines) which were developed in Eilers & Marx (1996). For step one and two, we define in the following

a B-spline basis and $k-$order differences of coefficients of the B-splines as measure of wiggliness. In the following, we restrict to equidistant knots since they lead to a simple difference penalty.

The idea of B-spline bases is to define piecewise polynomials and glue them together at the knots under additional smoothness constraints. In the first step, we define a B-spline basis with l parameters to parameterize the smooth function f. Following Wood (2017, chp. 5.3.3), define $l + m + 2$ knots $x_1 < x_2 < \ldots < x_{l+m+2}$. Then, the $(m + 1)$-th order spline can be written as

$$f(x) = \sum_{j=1}^{l} B_j^m(x)\beta_j \tag{3.17}$$

where we define the piecewise polynomial B-spline basis functions recursively as:

$$B_i^m(x) = \frac{x - x_i}{x_{i+m+1} - x_i} B_i^{m-1}(x) + \frac{x_{i+m+2} - x}{x_{i+m+2} - x_{i+1}} B_{i+1}^{m-1}(x), \quad i = 1, \ldots, l$$

and

$$B_i^{-1}(x) = \begin{cases} 1 & x_i \leq x \leq x_{i+1} \\ 0 & \text{else.} \end{cases}$$

In the second step, we want to define a measure for the wiggliness of the function. We can measure the wiggliness of a function by its variability and thus, derivatives of the function can be used. Since the second derivative of the function measures its curvature,

$$\int (f''(z))^2 dz \tag{3.18}$$

is a common measure for wiggliness of a function. More general, also derivatives of higher or lower order can be taken as a measure of wiggliness. Since B-splines are piecewise polynomials, their derivatives are easily obtained. In particular, one can show that the k-th derivative of a spline can be expressed with k-th order differences of basis coefficients and B-spline basis function of lower order, as described e.g. in De Boor & De Boor (1978); Fahrmeir et al. (2013). Thus, we obtain with the estimates of the coefficients of (3.17) also estimates of the derivatives of the function f. Moreover, Eilers & Marx (1996) could show that second-order differences of B-spline coefficients used in the penalty term of the PLS objective is closely connected to using (3.18) in the penalty term of the PLS objective. More general, k-th order difference penalties are good approximations of the integrated square of k-th derivatives. These results lead to the use of k-th order differences in the penalty.

The differences of k-th order can be recursively defined by $\Delta\beta_j = \beta_j - \beta_{j-1}$, $\Delta^2\beta_j = \beta_j - 2\beta_{j-1} + \beta_{j-2}$ and $\Delta^k\beta_j = \Delta(\Delta^{k-1}\beta_j)$. Utilizing this and omitting the superscript of the B-splines, we obtain the following PLS objective to estimate the smooth function f:

$$PLS(\lambda) = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{l} \beta_j B_j(x_i) \right)^2 + \lambda \sum_{j=k+1}^{l} (\Delta^k \beta_j)^2$$

where λ is the smoothing parameter that controls the smoothness of the function. For $\lambda \to 0$, the

penalization term tends to zero and the B-splines are fitted without penalization. For $\lambda \to \infty$, the approximated k-th derivatives tend to zero and a function in the $(k-1)$-null space of the penalty, i.e. a polynomial of degree $k-1$, is approximated when minimizing the corresponding PLS objective.

We translate the model into matrix notation. We do this here for a B-spline basis of degree one, i.e. we choose $m = 1$ for the B-spline basis functions in (3.17), and second-order differences. Then, we can write the B-spline basis representation and the matrix of second-order differences as

$$B = \begin{pmatrix} B_1(x_1) & \dots & B_l(x_1) \\ \vdots & & \vdots \\ B_1(x_n) & \dots & B_l(x_n) \end{pmatrix} \quad \text{and} \quad P = \begin{pmatrix} 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 & 1 \end{pmatrix}.$$

This results in the penalty matrix $S = P^\top P$ and we obtain as PLS objective in matrix notation

$$PLS(\lambda) = (y - B\beta)^\top (y - B\beta) + \lambda \beta^\top S \beta$$

where $\beta = (\beta_1, \dots, \beta_l)^\top$ is the l-dimensional parameter vector that we want to estimate for fixed smoothing parameter λ. Given a smoothing parameter λ, the PLS estimator of the parameters is obtained by minimizing the PLS objective, i.e. taking the derivative of PLS w.r.t. β and setting it to zero, and one can show that this results in

$$\hat{\beta} = (B^\top B + \lambda S)^{-1} B^\top y.$$

Another example to model the smooth function f are cubic (regression) splines. Cubic splines are, given knots, functions f that are twice continuously differentiable at these knots and that are cubic polynomials on the half open intervals between the knots as described e.g. in Fahrmeir et al. (2013, chp. 8.1). They could be constructed using e.g. a B-spline basis. Nevertheless, we will describe in the following another construction from Wood (2017, chp. 5.3.1) since this construction corresponds to the implementation in the R package `mgcv` which is also used in the application in section 5. We do not present the detailed construction but instead, give a brief summary.

In a first step, we define for l knots $x_1, \dots, x_l$ the cubic spline function $f(x)$ in terms of a parameter vector β by

$$f(x) = a_j^-(x)\beta_j + a_j^+(x)\beta_{j+1} + c_j^-(x)F_j\beta + c_j^+(x)F_{j+1}\beta, \quad \text{if } x_j \le x \le x_{j+1}$$

where $a_j^-, a_j^+, c_j^-, c_j^+, F_j$ are defined as in Wood (2017, chp. 5.3.1). This can be rewritten to

$$f(x) = \sum_{i=1}^{l} b_i(x)\beta_i.$$

In a second step, we define the measure for the wiggliness of the function with such that

$$\int_{x_1}^{x_k} f''(x)^2 dx = \beta^\top D^\top B^{-1} D\beta$$

where the matrices B and D are defined in Wood (2017, table 5.1) and we obtain again a penalty matrix $S = D^\top B^{-1} D$.

In general, we can use various basis representations for the smooth function f and different penalties. Following Fahrmeir et al. (2013, chp. 8.1.3), the above derivations are translated straightforward and we obtain

$$\begin{aligned} y &= B\beta + \varepsilon \\ PLS(\lambda) &= (y - B\beta)^\top (y - B\beta) + \lambda \beta^\top S\beta \end{aligned}$$

where y is the response, ε the error vector, B the matrix obtained from the basis representation of f and the covariate x, β is the parameter vector corresponding to the basis representation B, S is the penalty matrix to measure the wiggliness and λ is the smoothing parameter. Finally, we obtain the general PLS estimator

$$\hat{\beta} = (B^\top B + \lambda S)^{-1} B^\top y.$$

Extension to tensor product smooths

We describe the construction of tensor product smooths to extend smooths of one-dimensional covariates to covariate vectors. In particular, tensor product smooths allow for smooths of multiple one-dimensional covariates and therefore, are able to model interactions of covariates smoothly. Another advantage of tensor product smooths compared to other multivariate smoothing approaches is that the covariates can be of different scale since each is modeled with its own penalty term. Thus, they avoid the necessity to rescale the covariates. In general, there exist different construction approaches and we summarize the one from Wood (2017, chp. 5.6.1,5.6.2). The construction allows for various univariate basis representations and penalties. In the following, we restrict the construction for simplicity to

$$y_i = f(x_i, z_i, u_i) + \varepsilon_i$$

where f is a smooth function of three covariates and the error terms satisfy the same assumptions as for the classical linear model in (3.10). Next, the construction consists again of the two steps of basis representation and penalty description as for univariate smooths.

In a first step, we have to construct a basis representation analogous as for univariate smooths. Therefore, we assume that the marginal smooths of the covariates x, z, u are

$$f_x(x) = \sum_{i=1}^{I} \alpha_i a_i(x), \quad f_z(z) = \sum_{l=1}^{L} \delta_l d_l(z), \quad f_u(u) = \sum_{k=1}^{K} \beta_k b_k(u)$$

where a_i, d_l, b_k are basis functions and $\alpha_i, \delta_i, \beta_k$ are the corresponding parameters. In particular,

these marginal smooths imply marginal smooth model matrices B_x, B_z, B_u as in the last part of this subsection for univariate smoothers explicitly shown for the B-spline basis. Then, the joint smooth of x and z can be constructed if we let f_x change smoothly with z by modeling α_i as a smooth function of z and we obtain

$$f_{xz}(x,z) = \sum_{i=1}^{I} \alpha_i(z) a_i(x) = \sum_{i=1}^{I} \sum_{l=1}^{L} \delta_{il} d_l(z) a_i(x).$$

Analogously, for v and f_{xz}, the smooth for all three covariates is given by

$$f_{xzu}(x,z,u) = \sum_{i=1}^{I} \sum_{l=1}^{L} \sum_{k=1}^{K} \beta_{ilk} b_k(u) d_l(z) a_i(x).$$

If we order the coefficient vector β_{ilk} appropriately, one can show that we obtain a model matrix B for the smooth f_{xzu} by the row-wise Kronecker product of the marginal smooth model matrices B_x, B_z, B_u.

In the second step, we define the penalty for tensor product smooths. As for the basis representations of the marginal smooths, we assume to have the marginal smooth penalties

$$J_x(f_x) = \alpha^\top S_x \alpha, \quad J_z(f_z) = \delta^\top S_z \delta, \quad J_u(f_u) = \beta^\top S_u \beta$$

where α, δ, β are the coefficient vectors from the marginal smooths f_x, f_z, f_u and S_x, S_z, S_u are penalty matrices analogous as the one in the last part for univariate smooths. Next, we assume that $f_{x|zu}$ is the smooth change of f_{xzu} in x while holding z, u constant. Then, we could measure the wiggliness of the smooth f_{xzu} by adding the average wiggliness of the smooth in each direction. For example, the average wiggliness in x is the average wiggliness measured by $J_x(f_{x|zu})$ for all values of z, u, cp. also the description in Wood (2006, Figure 1). If we allow for separate smoothing parameters in each direction, a measure for wiggliness of the joint smooth f_{xzu} can be given by

$$J(f_{xzu}) = \lambda_x \int_{z,u} J_x(f_{x|zu}) dz du + \lambda_z \int_{x,u} J_z(f_{z|xu}) dx du + \lambda_u J_j(f_{u|xz}) dx dz.$$

Thus, the wiggliness measure can be obtained by the wiggliness measures for the marginal smooths. Moreover, this wiggliness measure allows for separate smoothing parameters in each direction and thus, different scaling of the the covariates.

In a last step, we define very briefly penalty matrices $\tilde{S}_x, \tilde{S}_z, \tilde{S}_u$ approximating the integrals over the wiggliness measures in each direction. This can be achieved by a reparametrization of the basis representations of f_x, f_z, f_u w.r.t. to evenly spaced covariates which also reparametrizes the corresponding univariate penalty matrices to S'_x, S'_z, S'_u. Then, the integrals can be approximated by

$$\int_{z,u} J_x(f_{x|zu}) dz du \approx h \beta^\top \tilde{S}_x \beta$$

where $\tilde{S}_x = S'_x \otimes \mathbb{I}_K \otimes \mathbb{I}_L$, the symbol $\otimes$ denotes the Kronecker product, h is some constant

computed w.r.t. to the spacing of the covariates and β is the coefficient vector obtained from the joint smooth f_{xzu}. Thus, we obtain for the penalty term the sum over the three penalty matrices $\tilde{S}_x, \tilde{S}_z, \tilde{S}_u$ and three separate smoothing parameters $\lambda_x, \lambda_z, \lambda_u$.

Finally, we can obtain a basis representation B and penalty matrix S for the smooth f_{xzu} of a covariate vector as for univariate smooths. Thus, we can minimize again a PLS objective or fall back to other estimation procedures as described in the next subsection.

In summary, we described how smooth relationships between a response and a covariate (vector) can be modeled constructing basis representations and wiggliness measures. There exist many basis representations and penalty terms used as wiggliness measures. For more examples, we refer the reader e.g. to Fahrmeir et al. (2013, chp. 8) or Wood (2017, chp. 5). Next, we extend this to model the smooth relationship between a response and multiple possibly multivariate covariates in additive models. Furthermore, we allow for non-normal responses to obtain GAMs at the end.

3.3.3. Additive and generalized additive models

As already indicated at the start of subsection 3.3.2, a third approach to gain more flexibility in model (3.10) is to drop the linearity assumption in the linear predictor to obtain additive models. In particular, we model in the following for $i = 1, \dots, n$ covariates $z_i = (1, z_{i1}, \dots, z_{iq})^\top$ as in model (3.10) linearly and covariates $x_i = (x_{i1}, \dots, x_{ip})^\top$ with smooth functions $f_1, \dots, f_p$.

Then, for $(y_i, z_i^\top, x_i^\top), i = 1, \dots, n$ the standard **additive model** can be defined as:

$$y_i = \gamma_0 + \sum_{k=1}^{q} \gamma_k z_{ik} + \sum_{j=1}^{p} f_j(x_{ij}) + \varepsilon_i \tag{3.19}$$

where $\gamma = (\gamma_0, \dots, \gamma_q)^\top$ is an unobservable parameter vector and $f_1 \dots, f_p$ are unobservable smooth functions. Additionally, $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)^\top$ is a random error term satisfying the same assumptions as for the linear regression model (3.10), i.e. the error term consists of conditionally normally i.i.d. random variables with mean zero and constant variance.

As described in the last subsection 3.3.2, we can parameterize each smooth function $f_j, j = 1, \dots, p$ using a basis representation $B^{[j]}$. Accounting for identifiablility constraints as in Wood (2017, chp. 6.1), we can reparameterize the basis representations $B^{[j]}$ to $\mathcal{B}^{[j]}, j = 1, \dots, p$ to obtain one large model matrix for model (3.19)

$$X = (Z : \mathcal{B}^{[1]} : \dots : \mathcal{B}^{[p]})$$

where : denotes that the matrices are bind next to each other. Then, the additive model (3.19) can be rewritten as

$$y_i = X_i \beta + \varepsilon_i, \quad i = 1, \dots, n$$

where X_i is a row of the model matrix X and $\beta^\top = (\gamma^\top, b^\top)$ is the parameter vector consisting of the coefficient vector γ corresponding to linearly modeled covariates and the coefficient vector $b^\top = (b_1^\top, \dots, b_p^\top)$ consisting of coefficient vectors b_j corresponding to basis representations $\mathcal{B}^{[j]}$

of covariates modeled via smooth functions.

To penalize for wiggliness of the smooth functions, we choose at least one penalty matrix $S^{[j]}$ for each smooth function $f_j, j = 1 \ldots, p$. All penalty matrices are enlarged to matrices $\mathcal{S}^{[j]}$ where $S^{[j]}$ is a block diagonal entry at the elements corresponding to b_j from β and the matrix is otherwise filled with zeros. Then, we can write the corresponding sum of penalties as

$$\sum_{j=1}^{p} \lambda_j \beta^\top \mathcal{S}^{[j]} \beta$$

where λ_j is the smoothing parameter corresponding to the smooth function f_j and β is the parameter vector of the parameterized additive model above.

Finally, estimation of the additive model (3.19) can be achieved by minimizing the PLS objective

$$PLS(\beta) = (y - X\beta)^\top (y - X\beta) + \sum_{j=1}^{p} \lambda_j \beta^\top \mathcal{S}^{[j]} \beta \tag{3.20}$$

w.r.t. β for given smoothing parameters $\lambda_j, j = 1, \ldots, p$. This can be done using e.g. a backfitting algorithm as first described in Friedman & Stuetzle (1981) or by directly setting the derivative of the PLS objective to zero to obtain an analytical solution of the PLS (3.20) as described in Fahrmeir et al. (2013, chp. 9.6.1). Furthermore, the smoothing parameters can be estimated using e.g. a REML approach treating the parameters of the smooth functions as random effects as described for GLMMs.

Generalized additive models

Ultimately, we derive **generalized additive models** by joining GLMs from (3.13) and additive models from (3.19). For $i = 1, \ldots, n, (y_i, z_i^\top, x_i^\top)$, x_i and z_i as for additive models in (3.19), $\eta_i = z_i^\top \gamma + f_1(x_{i1}) + \ldots + f_p(x_{ip})$, we assume $y_i | x_i, z_i \sim \text{EF}(\mu_i, \phi)$ and that the y_i's are conditionally independent given η_i. Then, we obtain GAMs as

$$g(\mu_i) = z_i^\top \gamma + \sum_{j=1}^{p} f_j(x_{ij}) \tag{3.21}$$

where γ is an unobservable parameter vector, $f_1, \ldots, f_p$ are unobservable smooth functions and g is a link function as defined for GLMs (3.13).

For estimation, we rewrite the GAMs (3.21) analogous as for additive models. First, we use basis representations $B^{[1]}, \ldots, B^{[p]}$ for the smooth functions $f_1, \ldots, f_p$ and account for identifiability constraints to obtain $\mathcal{B}^{[1]}, \ldots, \mathcal{B}^{[p]}$ and thus, the model matrix

$$X = (Z : \mathcal{B}^{[1]} : \ldots : \mathcal{B}^{[p]}).$$

This results in the rewritten model

$$g(\mu_i) = X_i \beta, \quad y_i | x_i, z_i \sim \text{EF}(\mu_i, \phi) \tag{3.22}$$

where X_i is a row of the model matrix X, $\beta^\top = (\gamma^\top, b_1^\top, \ldots, b_p^\top)$ is the unobservable parameter vector that contains the coefficient vector γ corresponding to the covariate vector z_i at its first $k+1$ entries and the coefficient vectors b_j corresponding to the basis representations $\mathcal{B}^{[j]}, j = 1, \ldots, p$ and the rest is defined as in (3.21).

Analogous as for additive models, we choose at least one penalty matrix $S^{[j]}$ for each smooth function f_j. All penalty matrices are enlarged to matrices $\mathcal{S}^{[j]}$ where $S^{[j]}$ is a block diagonal entry at the elements corresponding to b_j from β and the matrix is otherwise filled with zeros. Then, we can write the corresponding sum of penalties as

$$\sum_{j=1}^{p} \lambda_j \beta^\top \mathcal{S}^{[j]} \beta$$

where λ_j is the smoothing parameter corresponding to the smooth function f_j.

Finally, estimation of model (3.21) can be based on the penalized log-likelihood objective

$$\ell_{\text{pen}}(\beta) = \ell(\beta) - \sum_{j=1}^{p} \lambda_j \beta^\top \mathcal{S}^{[j]} \beta \tag{3.23}$$

where $\ell(\beta)$ corresponds to the log-likelihood of model (3.22). Again, there exists no analytical solution. Nevertheless, given the smoothing parameter vector $\lambda = (\lambda_1, \ldots, \lambda_p)$, the parameter vector β can be estimated, similar as for GLMMs, using a PIRLS algorithm since the objective is equivalent to the objective in (3.16) for given λ. The estimation of the smoothing parameters can be based on a REML objective similar as for the estimation of the covariance matrix parameters of the GLMMs. For more details, we refer the reader e.g. to Wood (2011) for a REML based approach or to Wood (2017, chp. 6) for a comparison of multiple approaches to estimate the smoothing parameters.

In summary, the subsection 3.3 reviewed different extensions of the classical linear model (3.10) under a common notation. First, GLMs were set up to define distributions of the EF and their connection to the responses. Second, GLMMs were introduced to connect to numerical procedures such as REML and PIRLS and obtain a base for additive models and GAMs as large LMMs and GLMMs in the context of estimation. Then, smoothers were presented to obtain basis representations and penalties for the wiggliness of smooth functions used in additive models and GAMs. Finally, the models were connected to derive additive models and GAMs together with their PLS objective and penalized log-likelihood, respectively. Overall, additive models offer a flexible extension w.r.t. the linearity assumption in the covariates allowing for non-linear covariate effects. GAMs extend this further by allowing also for non-normal responses. This is useful when we model binary, count or categorical data but even so when we want to model continuous data with a skewed distribution. This flexibility comes with a larger computational cost. Moreover, the results for GAMs are not exact but instead depend on asymptotic theory as explained in Wood (2017). In the next subsection, these additive models and GAMs are used as single target models to derive the corresponding regression frameworks in the graph space.

3.4. Extended regression models in the graph space

In this subsection we combine the former three subsections 3.1, 3.2 and 3.3 to develop additive and generalized additive regression frameworks in $\mathfrak{G}$. The subsection is structured by differentiating between two extensions: 1) additive models in $\mathfrak{G}$ and 2) generalized additive models in $\mathfrak{G}$. For the first extension, we derive a similar objective as the LS objective (3.6) for linear regression models in $\mathfrak{G}$. This is done starting with the PLS objective (3.20) for single target additive models and deriving the corresponding PLS objective in the graph space. Afterwards, we state a possible result on the almost sure convergence of the AAC algorithm for additive models (algorithm 2) to a local minimum of the PLS objective viewed as a function of the model parameters. Finally, we explicitly describe three cases for which the algorithm does not converge to understand the regression frameworks in $\mathfrak{G}$ more thoroughly. For the second extension, we start with the single target GAMs and their penalized log-likelihoods (3.23) and derive the corresponding objective in the graph space. Finally, we discuss possible challenges of GAMs in the graph space and analyse the possibility for a GAM AAC algorithm as given in algorithm 3.

3.4.1. Additive models in the graph space

As in subsection 3.2, we model the multi output regression model in the total space $\mathcal{A} = \mathbb{R}^J$ by J single target additive models, i.e. we model $h : \mathbb{R}^r \to \mathcal{A}, x \mapsto h(x) = (h_a(x))_{a\in\mathcal{J}}$ with J independent additive models for each graph attribute where $r = 1 + q + p$. In particular, for each $a \in \mathcal{J}, i = 1, \ldots, n$ and $(y_{i,a}, (z_i^\top, x_i^\top)) \in \mathbb{R} \times \mathbb{R}^r$ where $z_i = (1, z_{i1}, \ldots, z_{iq})^\top, x_i = (x_{i1}, \ldots, x_{ip})^\top$, the separate single target additive models are defined as

$$y_{i,a} = h_a((z_i^\top, x_i^\top)^\top) + \varepsilon_{i,a} = \gamma_{0,a} + \sum_{k=1}^{q} \gamma_{k,a} z_{ik} + \sum_{j=1}^{p} f_{j,a}(x_{ij}) + \varepsilon_{i,a}, \quad i = 1, \ldots, n \tag{3.24}$$

where $f_{1,a}, \ldots, f_{p,a}, \gamma_a = (\gamma_{0,a}, \ldots, \gamma_{q,a})^\top$ and $\varepsilon_a = (\varepsilon_{1,a}, \ldots, \varepsilon_{n,a})^\top$ are defined as for the additive models in (3.19).

If we parameterize the smooth functions $f_{1,a}, \ldots, f_{p,a}$ using basis functions, we can rewrite for each $a \in \mathcal{J}$ the single target additive models analogously to the derivations for additive models in the last subsection to

$$h_a((z_i^\top, x_i^\top)^\top) = X_i \beta_a \tag{3.25}$$

where X_i is a row of the design matrix $X = (Z : \mathcal{B}^{[1]} : \ldots : \mathcal{B}^{[p]})$, $\mathcal{B}^{[1]}, \ldots, \mathcal{B}^{[p]}$ are basis representations of the smooths $f_{1,a}, \ldots, f_{p,a}$ (independent of $a \in \mathcal{J}$) accounting for identifiability constraints and $\beta_a^\top = (\gamma_a^\top, b_a^\top)$ is the parameter vector consisting of the coefficient vector γ_a corresponding to linearly modeled covariates and the coefficient vector $b_a^\top = (b_{1,a}^\top, \ldots, b_{p,a}^\top)$ consisting of coefficient vectors $b_{j,a}, j = 1, \ldots, p$ corresponding to covariates modeled via smooth functions. From here on, we write $h_a(X_i) \coloneqq h_a((z_i^\top, x_i^\top)^\top)$ for clarity.

We can estimate separately for each $a \in \mathcal{J}$ the model (3.24) by minimizing w.r.t. the parameter

vector β_a given smoothing parameters the objective

$$PLS_a = PLS(\beta_a) = \sum_{i=1}^{n}(y_{i,a} - h_a(X_i))^2 + \sum_{j=1}^{p} \lambda_{j,a}\beta_a^\top \mathcal{S}^{[j]}\beta_a$$

where $\mathcal{S}^{[j]}$ is a penalty matrix defined (independent of $a \in \mathcal{J}$) as for additive models in the last subsection for each smooth function $f_{j,a}, j = 1, \ldots, p$, and $\lambda_{j,a}$ are the smoothing parameters corresponding to the smooths $f_{j,a}, j = 1, \ldots, p$. A similar penalization objective for a multi output model with separate ridge regressions can be found in Borchani et al. (2015, chp. 2.1.1) describing the multi output ridge regression developed in Hoerl & Kennard (1970).

In a next step, we derive the corresponding PLS objective in the total space. Let $d^2_{\mathcal{A}}$ be the squared Euclidean distance in the total space $\mathcal{A}$. Then, for $i = 1, \ldots, n, (y_i^\top, (z_i^\top, x_i^\top)) \in \mathcal{A} \times \mathbb{R}^r$ with $y_i = (y_{i,1}, \ldots, y_{i,J})^\top$, the PLS objectives for the graph attributes imply in the total space $\mathcal{A}$ the following objective which we want to minimize:

$$\begin{aligned} PLS_{\mathcal{A}} = \sum_{a=1}^{J} PLS_a &= \sum_{a=1}^{J}\left[\sum_{i=1}^{n}(y_{i,a} - h_a(X_i))^2 + \sum_{j=1}^{p}\lambda_{j,a}\beta_a^\top \mathcal{S}^{[j]}\beta_a\right] \\ &= \sum_{i=1}^{n} d^2_{\mathcal{A}}(y_i, h(X_i)) + \sum_{a=1}^{J}\sum_{j=1}^{p}\lambda_{j,a}\beta_a^\top \mathcal{S}^{[j]}\beta_a \end{aligned} \tag{3.26}$$

where $h(X_i) = (h_a(X_i))_{a\in\mathcal{J}}$. The objective is minimized separately for each $a \in \mathcal{J}$ w.r.t. $\beta_a^\top = (\gamma_a^\top, b_a^\top)$ while choosing smoothing parameter vectors $\lambda_a = (\lambda_{1,a}, \ldots, \lambda_{p,a})$. Practically, the objective can be minimized separately for each $a \in \mathcal{J}$ using e.g. a direct approach or a backfitting algorithm for the parameter vectors β_a and a REML approach for the smoothing parameters λ_a as described for additive models with one-dimensional responses.

After projecting from the total space into the graph space, we aim to minimize for $([y_i], (z_i^\top, x_i^\top)) \in \mathfrak{G} \times \mathbb{R}^r,\ i = 1, \ldots, n$ in the graph space $\mathfrak{G}$ the PLS objective

$$PLS_{\mathfrak{G}} = \sum_{i=1}^{n} d^2_{\mathfrak{G}}([y_i], f(X_i)) + \sum_{a=1}^{J}\sum_{j=1}^{p}\lambda_{j,a}\beta_a^\top \mathcal{S}^{[j]}\beta_a \tag{3.27}$$

where the metric $d_{\mathfrak{G}}$ is defined as in (3.1) and $f(X_i) := \pi \circ h(X_i)$. Then, we define the **additive regression function in the graph space** as the function $f : \mathbb{R}^r \to \mathfrak{G}$, $f = \pi \circ h$ where $h = (h_a)_{a\in\mathcal{J}}$, h_a is defined as in (3.25), that minimizes the PLS objective (3.27). The corresponding regression models are called additive models in $\mathfrak{G}$. Minimization of the PLS (3.27) minimizes an intrinsic error in the graph space while penalizing for parameters in the total space. Moreover, we note that we allow for different degrees of smoothness when modeling each graph attribute in the total space. We study this objective further in subsection 4.2 while mentioning preliminary considerations in the following and in subsection 3.4.2.

Align all and compute algorithm for additive models

For estimation of the additive regression function f in the graph space, we extend algorithm 1 to additive models in $\mathfrak{G}$ to minimize the PLS objective (3.27). In the following part, we state briefly

the extended AAC algorithm given in algorithm 2. Then, we give a corresponding convergence result and a scheme to prove it. In the end, three cases when the algorithm fails to converge are described.

Algorithm 2: AAC algorithm to compute additive regression functions $f \in \mathfrak{G}$

Data: $((z_1^\top, x_1^\top), [y_1]), \dots, ((z_n^\top, x_n^\top), [y_n]) \in \mathbb{R}^r \times \mathfrak{G}$, thresholds κ, δ
Result: $f \in \mathfrak{G}$
Randomly pick one graph $[y_l] \in \mathfrak{G}$ and its representative $T_l y_l \in \mathcal{A}$;
Align the representatives $T_j y_j, j \in \{1, \dots, n\} \backslash l$ of all other graphs to $T_l y_l$;
Obtain h from a regression of $(X_1, T_1 y_1), \dots, (X_n, T_n y_n)$ in $\mathcal{A}$ minimizing (3.26);
Set $\tilde{f} = \pi \circ h$;
Align all representatives w.r.t. $\tilde{f}$ obtaining $\tilde{T_1} y_1, \dots, \tilde{T_n} y_n$;
while $\delta > \kappa$ **do**
  Obtain h by minimizing (3.26) for $(X_1, \tilde{T_1} y_1), \dots, (X_n, \tilde{T_n} y_n)$ and $f = \pi \circ h$;
  Align all representatives w.r.t. f obtaining $\tilde{T_1} y_1, \dots, \tilde{T_n} y_n$;
  Compute the distance of the sum of squared prediction errors $\delta = \Delta(\tilde{f}, f)$;
  Set $\tilde{f} = f$;
end
Return $f = \tilde{f}$.

Following closely the theorem for AAC algorithm for linear regression models from Calissano et al. (2022), we state the next theorem to obtain a similar result for algorithm 2. In the following and as before, the functions f and h for additive models in $\mathfrak{G}$ depend implicitly on β where $\beta = (\beta_1, \dots, \beta_J)$ and $\beta_a, a \in \mathcal{J}$ is defined as described in model (3.25).

Theorem 1. *Let $\mathcal{A} = \mathbb{R}^J$ be the total space, $\mathcal{T}$ the set of permutations and $\mathfrak{G} = \mathcal{A}/\mathcal{T}$ the corresponding graph space. Additionally, we denote with p a push forward measure of the Lebesgue measure m on $\mathcal{A}$. Moreover, let $\mathbb{P}_{[Y]}$ be a probability measure on $\mathfrak{G}$ absolutely continuous w.r.t. p and $\mathbb{P}_X$ a probability measure on $\mathbb{R}^r$ absolutely continuous w.r.t. the Lebesgue measure on $\mathbb{R}^r$. For $i = 1, \dots, n$, $((z_i^\top, x_i^\top), [y_i]) \in \mathbb{R}^r \times \mathfrak{G}$ is sampled independently from the distribution $\mathbb{P}_X \times \mathbb{P}_{[Y]}$.*

Next, we assume that $f = \pi \circ h$ is the regression function obtained by the AAC algorithm 2. Additionally, for the parameterized single target additive models from (3.25) of the graph attributes, we assume that the design matrix X has full rank and that for $(z_i^\top, x_i^\top) \sim \mathbb{P}_X$ and $\beta_a \neq \tilde{\beta}_a$, we have almost surely $X_i \beta_a \neq X_i \tilde{\beta}_a$ for all $a \in \mathcal{J}$.

Then,

1. *The AAC algorithm 2 stops in finite time.*

2. *The AAC algorithm 2 converges almost surely to a local minimum of*

$$\beta \mapsto \sum_{i=1}^{n} d^2_{\mathfrak{G}}([y_i], f(X_i)) + \sum_{a=1}^{J} \sum_{j=1}^{p} \lambda_{j,a} \beta_a^\top \mathcal{S}^{[j]} \beta_a \tag{3.28}$$

Proof. The scheme for a proof, which is translated from Calissano et al. (2022) to additive models, can be found in D. □

The idea of the proof for 1. is that the PLS objective (3.26) for the graph attributes in the current iteration cannot increase by refitting the model to realigned observations or by realigning observations to the current regression function. Then, since the number of permutations is finite, the algorithm stops in finite time. The idea for 2. is to show that, with probability one, in a small ball around the parameter vector β the alignment of the observations w.r.t. the corresponding regression function h obtained by algorithm 2 does not change. This means, they would be still optimally aligned when changing the regression function by a small enough amount. Since we minimize the PLS objective (3.26) in the total space, this implies that the algorithm stops at a local minimum of the PLS objective (3.27) in the graph space. Finally, we note that the convergence holds up to numerical imprecision through inexact graph matching algorithms and algorithms to compute the parameters of the single target additive models.

Next, we want to analyse which cases are neglected in the almost sure results of the AAC algorithms. An important part of the proof is to show that the probability of the set of representatives which would have to be realigned to be optimally aligned w.r.t. the regression function in a small ball around the obtained regression function from the AAC algorithm 2 is zero. This set consists of observations $((z_i, x_i), [y_i])$ which satisfy for some representatives $d^2_{\mathcal{A}}(y_i, h(X_i)) = d^2_{\mathcal{A}}(Ty_i, h(X_i))$. As discussed for FDs in subsection 3.1.1, there are three non disjoint cases for which $d^2_{\mathcal{A}}(y_i, h(X_i)) = d^2_{\mathcal{A}}(Ty_i, h(X_i))$ could happen. First, if $h(X_i)$ is an ordinary graph and the optimally aligned representative y_i of $[y_i]$ is on the boundary of the FD of $h(X_i)$; second, if y_i is an ordinary graph and $h(X_i)$ is on the boundary of the FD of y_i; third, if $h(X_i)$ and y_i are both singular graphs. In the following, we describe these cases to obtain a better understanding for the algorithm, its proof and possible issues.

Case 1: $h(X_i)$ is an ordinary graph and the optimally aligned representative y_i of $[y_i]$ is on the boundary of the FD of $h(X_i)$, i.e. $y_i \in \partial FD_{h(X_i)}$. Since the FD of $h(X_i)$ is defined as

$$D_{h(X_i)} = \bigcap_{T \in \mathcal{T}} \{Y \in \mathcal{A} | d_{\mathcal{A}}(Y, h(X_i)) \leq d_{\mathcal{A}}(TY, h(X_i)\},$$

we have that $y_i \in \partial FD_{h(X_i)}$ means

$$y_i \in \bigcup_{T \in \mathcal{T}} \{Y \in \mathcal{A} | d_{\mathcal{A}}(Y, h(X_i)) = d_{\mathcal{A}}(TY, h(X_i)\} \cap D_{h(X_i)}.$$

However,

$$\begin{aligned} & m\left(\bigcup_{T \in \mathcal{T}} \{Y \in \mathcal{A} | d_{\mathcal{A}}(Y, h(X_i)) = d_{\mathcal{A}}(TY, h(X_i)\} \cap D_{h(X_i)}\right) \\ \leq & \sum_{T \in \mathcal{T}} m\left(\{Y \in \mathcal{A} | d_{\mathcal{A}}(Y, h(X_i)) = d_{\mathcal{A}}(TY, h(X_i)\}\right) = 0 \end{aligned}$$

where the first part follows by σ-additivity of the probability measure m, since $D_{h(X_i)} \subseteq \mathcal{A}$ and because we can estimate from above the probability omitting the condition that $h(X_i)$ is an ordinary graph (cp. also case 2 below) and its probability, and the second part follows since $\mathcal{T}$ is finite and hyperplanes in $\mathcal{A}$ have probability zero w.r.t. m since m is absolute continuous w.r.t. the Lebesgue measure on $\mathcal{A}$.

Case 2: The optimally aligned y_i (w.r.t. $h(X_i)$) is an ordinary graph and $h(X_i)$ is on the boundary of the FD of y_i, i.e. $h(X_i) \in FD_{y_i}$. To consider this case, a first step is to write the parameter matrix $\beta = (\beta_1, \dots, \beta_J)$ corresponding to the regression function $h(X)$ in the total space obtained by algorithm 2 to

$$\beta = \left[\left(X^\top X + \sum_{j=1}^{p} \lambda_{j,1} \mathcal{S}^{[j]} \right)^{-1} X^\top \begin{pmatrix} y_{1,1} \\ \vdots \\ y_{n,1} \end{pmatrix}, \dots, \left(X^\top X + \sum_{j=1}^{p} \lambda_{j,J} \mathcal{S}^{[j]} \right)^{-1} X^\top \begin{pmatrix} y_{1,J} \\ \vdots \\ y_{n,J} \end{pmatrix} \right] \tag{3.29}$$

where all terms are defined as in (3.26). Utilizing this notation, the aim would be to show that the probability of

$$h(X_i) = X_i \beta \in \bigcup_{T \in \mathcal{T}} \{Y \in \mathcal{A} | d_{\mathcal{A}}(Y, y_i) = d_{\mathcal{A}}(TY, y_i\} \cap D_{y_i}$$

is zero, where D_{y_i} is the FD of y_i and X_i is a row of the model matrix of the parameterized additive models defined as in (3.25). This is similar to Calissano et al. (2022, proof of lemma 1,2) although we allow additionally for ordinary graphs on the boundary of the FDs while they account only for singular graphs $h(X_i)$.

Case 3: The optimally aligned y_i (w.r.t. $h(X_i)$) and $h(X_i)$ are singular graphs. This case can be considered by choosing an ordinary graph $Z \in \mathcal{A}$ which is optimally aligned with y_i and $h(X_i)$. Then, this is analogous to case 1 and 2.

Accounting for these three cases, the convergence can be proven using σ-additivity on the union over all n observations and the definition of the push forward measure. Besides that, we observe that graphs on boundaries of FDs can be a problem for the almost sure convergence of AAC algorithms.

3.4.2. Generalized additive models in the graph space

This subsection translates the penalized log-likelihood of single target GAMs to the corresponding objective in the graph space. Afterwards, the derived model and the possibility of an AAC algorithm is discussed. It could be possibly seen as a first step in the direction of GAMs in the graph space. Nevertheless, we will see that the inclusion of non-normal responses has to be treated more carefully.

We start by modeling $h : \mathbb{R}^r \to \mathcal{A}, x \mapsto h(x) = (h_a(x))_{a \in \mathcal{J}}, r = 1+q+p$ as a multi output model with J single target GAMs. We note that $\mathcal{A}$ does not necessarily equal $\mathbb{R}^J$. Then, for each $a \in \mathcal{J}, i = 1, \dots, n$ and $(y_{i,a}, (z_i^\top, x_i^\top))$ where $z_i = (1, z_{i1}, \dots, z_{iq})^\top, x_i = (x_{i1}, \dots, x_{ip})^\top, (z_i^\top, x_i^\top) \in \mathbb{R}^r$, the single target GAMs are defined as

$$h_a((z_i^\top, x_i^\top)^\top) = g(\mu_{i,a}) = \eta_{i,a} = z_i^\top \gamma_a + \sum_{j=1}^{p} f_{j,a}(x_{ij}), \quad i = 1, \dots, n \tag{3.30}$$

where $y_{i,a} | x_i, z_i \sim \mathrm{EF}(\mu_{i,a}, \phi_a)$, $y_{i,a}$ are conditionally independent for $i = 1, \dots, n$ and $g, f_{j,a}, \gamma_a$ are defined as for GAMs in (3.21). With basis representations $\mathcal{B}^{[j]}, j = 1 \dots, J$ (independent of

$a \in \mathcal{J}$) for the smooth functions $f_{1,a}, \ldots, f_{p,a}$ and the corresponding design matrix $X = (Z : \mathcal{B}^{[1]} : \ldots : \mathcal{B}^{[p]})$, we can rewrite the model to

$$h_a(X_i) = X_i \beta_a, \quad y_{i,a} | x_i, z_i \sim \text{EF}(\mu_{i,a}, \phi_a), \quad i = 1, \ldots, n, a \in \mathcal{J} \tag{3.31}$$

where X_i is a row of the design matrix X and the same convention as before for $h_a(X_i) \coloneqq h_a((z_i^\top, x_i^\top)^\top)$.

For each graph attribute $a \in \mathcal{J}$, we estimate the GAMs (3.30) separately by maximizing numerically their penalized log-likelihoods w.r.t. β_a for corresponding smoothing parameters $\lambda_{j,a}$, i.e. for each $a \in \mathcal{J}$

$$\ell_{\text{pen}}(\beta_a) = \ell(\beta_a) - \sum_{j=1}^{p} \lambda_{j,a} \beta_a^\top \mathcal{S}^{[j]} \beta_a$$

where $\beta_a = (\beta_{0,a}, \ldots, \beta_{r,a})^\top$ is a parameter vector consisting of coefficients for the covariates z_i and covariates modeled via smooths $f_{1,a}, \ldots, f_{p,a}$ and $\lambda_a = (\lambda_{1,a}, \ldots, \lambda_{p,a})$ is a vector of smoothing parameters corresponding to the smooths $f_{1,a}, \ldots, f_{p,a}$ of the graph attribute $a \in \mathcal{J}$; additionally, $\ell(\beta_a)$ is the log-likelihood corresponding to model (3.31). In particular, we can rewrite the penalized log-likelihood with the definition for densities of the EF (3.12) as

$$\ell_{\text{pen}}(\beta_a) = \sum_{i=1}^{n} \frac{y_{i,a}\theta_{i,a} - b(\theta_{i,a})}{\phi_a} w_{i,a} + c(y_{i,a}, \phi_a, w_{i,a}) - \sum_{j=1}^{p} \lambda_{j,a} \beta_a^\top \mathcal{S}^{[j]} \beta_a$$

As we described in subsection 3.3 for GLMs, the dependence of the first sum on β_a is implicit by the connection of the natural parameters and β_a and maximization can be achieved e.g. by a PIRLS algorithm to estimate β_a together with a REML based approach to estimate the smoothing parameter vector λ_a.

Next, we can write the corresponding regression model in the total space $\mathcal{A}$ in matrix notation as

$$h(X) = X\beta \tag{3.32}$$

where

$$h(X) = \begin{bmatrix} h_1(X_1) & \ldots & h_J(X_1) \\ \vdots & \ddots & \vdots \\ h_1(X_n) & \ldots & h_J(X_n) \end{bmatrix}$$

is a multi output matrix with $(h_a(X_1), \ldots, h_a(X_n))^\top, a \in \mathcal{J}$ as the linked mean for the graph attribute with index $a \in \mathcal{J}$; furthermore, $X = (Z : \mathcal{B}^{[1]} : \ldots : \mathcal{B}^{[p]})$ is a design matrix defined as

above for (3.31); moreover, the matrix of coefficient vectors is given by

$$\beta = (\beta_1, \dots, \beta_J) = \begin{bmatrix} \beta_{0,1} & \dots & \beta_{0,J} \\ \vdots & \ddots & \vdots \\ \beta_{r,1} & \dots & \beta_{r,J} \end{bmatrix}$$

where each column corresponds to the coefficient vector of the single target GAMs for one graph attribute.

Then, for $i = 1, \dots, n$, $(y_i^\top, (z_i^\top, x_i^\top)) \in \mathcal{A} \times \mathbb{R}^r$ with $y_i = (y_{i,1}, \dots, y_{i,J})^\top$ estimation of model (3.32) in the total space $\mathcal{A}$ is performed by maximizing the objective

$$\begin{aligned} \ell_{\mathcal{A}}(\beta) &= \sum_{a=1}^{J} \ell_{\text{pen}}(\beta_a) = \sum_{a=1}^{J} \left[\ell(\beta_a) - \sum_{j=1}^{p} \lambda_{j,a} \beta_a^\top \mathcal{S}^{[j]} \beta_a \right] \\ &= \sum_{a=1}^{J} \left[\sum_{i=1}^{n} \frac{y_{i,a}\theta_{i,a} - b(\theta_{i,a})}{\phi_a} w_{i,a} + c(y_{i,a}, \phi_a, w_{i,a}) - \sum_{j=1}^{p} \lambda_{j,a} \beta_a^\top \mathcal{S}^{[j]} \beta_a \right] \end{aligned} \tag{3.33}$$

w.r.t. β.

Ultimately, for $([y_i], (z_i^\top, x_i^\top)) \in \mathfrak{G} \times \mathbb{R}^r,\ i = 1, \dots, n$ we aim to maximize the objective

$$\begin{aligned} \ell_{\mathfrak{G}}(\beta) &= \min_{T \in \mathcal{T}} \ell_{\mathcal{A}}(T, \beta) \\ &= \min_{T \in \mathcal{T}} \sum_{a=1}^{J} \left[\sum_{i=1}^{n} \frac{t_{i,a} y_{i,a} \theta_{i,a} - b(\theta_{i,a})}{\phi_a} w_{i,a} + c(t_i y_{i,a}, \phi_a, w_{i,a}) - \sum_{j=1}^{p} \lambda_{j,a} \beta_a^\top \mathcal{S}^{[j]} \beta_a \right] \end{aligned} \tag{3.34}$$

in the graph space $\mathfrak{G}$ w.r.t. β while choosing optimal smoothing parameter vectors $\lambda_a, a \in \mathcal{J}$. Here, $t_{i,a}$ denotes the corresponding permutation of representative $T_i y_i$ on graph attribute level. We define the **generalized additive regression function in the graph space** as the function $f : \mathbb{R}^r \to \mathfrak{G}$ that maximizes objective (3.34) with $f = \pi \circ h$ and $h = (h_a)_{a \in \mathcal{J}}$, h_a is defined as in (3.31). The corresponding models are called GAMs in $\mathfrak{G}$.

Discussion and align all and compute algorithm for generalized additive models

Above, we introduced a GAM framework in the graph space to account appropriately for the space of the graph attributes. The approach taken to develop the framework is to derive objective (3.34) which corresponds to the single target penalized log-likelihoods from the single target GAMs in the total space. However, it is not clear or reasoned, if the projection from (3.33) to (3.34) is correct or why objective (3.34) should be meaningful in the graph space. In this part, we discuss this GAM framework. Afterwards, the possibility to develop an AAC algorithm for GAMs is examined based on algorithm 3 and a simple example. In the end, we address general issues observable during the example while highlighting briefly the difference between probabilistic and geometric regression approaches. This part expounds difficulties rather than providing solutions.

First, we want to gain some insights from the introduction of these GAMs. In particular, we can describe more detailed the difficulty of extending the existing linear and additive regression

Algorithm 3: AAC algorithm to compute generalized additive functions $f \in \mathfrak{G}$

Data: $((z_1^\top, x_1^\top), [y_1]), \ldots, ((z_n^\top, x_n^\top), [y_n]) \in \mathbb{R}^r \times \mathfrak{G}$, thresholds κ, δ
Result: $f \in \mathfrak{G}$
Randomly pick one graph $[y_l] \in \mathfrak{G}$ and its representative $T_l y_l \in \mathcal{A}$;
Align the representatives $T_j y_j, j \in \{1, \ldots, n\} \backslash l$ of all other graphs to $T_l y_l$;
Obtain h from a regression of $(x_1, T_1 y_1), \ldots, (x_n, T_n y_n)$ in $\mathcal{A}$ maximizing (3.33);
Set $\tilde{f} = \pi \circ h$;
Align all representatives w.r.t. $\tilde{f}$ obtaining $\tilde{T}_1 y_1, \ldots, \tilde{T}_n y_n$;
while $\delta > \kappa$ **do**
  Obtain h by maximizing (3.33) for $(X_1, \tilde{T}_1 y_1), \ldots, (X_n, \tilde{T}_n y_n)$ and $f = \pi \circ h$;
  Align all representatives w.r.t. f obtaining $\tilde{T}_1 y_1, \ldots, \tilde{T}_n y_n$;
  Compute the distance of the sum of squared prediction errors $\delta = \Delta(\tilde{f}, f)$;
  Set $\tilde{f} = f$;
end
Return $f = \tilde{f}$.

framework in the graph space to graphs with a total space that is different to $\mathcal{A} = \mathbb{R}^J$. We observe that the GAM regression functions $f(X_1), \ldots, f(X_n) \in \mathfrak{G}$ are obtained projecting combined means of the graph attributes from the total space to the graph space. A possible issue is that combining these single target means can lead to labeled points $h(X_i) \notin \mathcal{A}$. Among other things, we would have to translate the concept of optimal alignment to these points which was not considered so far in the derivations of the GAMs above during the projection step from (3.33) to (3.34). One could think about optimal aligning graphs w.r.t. points outside of the total space or by aligning graphs w.r.t. sampled graphs from the distributions with mean modeled by the GAMs.

Let us illustrate the former type of optimal alignment at an example of binary graphs. Thus, let $\mathcal{A} = \{0, 1\}^J$, $\mathcal{T}$ the set of permutations and $\mathfrak{G} = \mathcal{A}/\mathcal{T}$ the graph space. Then, we could say a graph $TY_1 \in \mathcal{A}, T \in \mathcal{T}$ is optimally aligned w.r.t. a point $Y_2 \in [0, 1]^J$ if

$$d_{\mathcal{A}}(TY_1, Y_2) = \min_{\tilde{T} \in \mathcal{T}} d_{\mathcal{A}}(\tilde{T}TY_1, Y_2) = d_{\mathfrak{G}}([Y_1], [Y_2])$$

where $d_{\mathcal{A}}$ could be e.g. the Euclidean metric. This definition would enable the alignment of graphs w.r.t. their fitted multi output means although we do not examine issues with the fitted multi output means not lying in the total space.

We concretize this further to investigate what could happen for an AAC algorithm as defined in algorithm 3. We assume to observe a sample of binary graphs with Bernoulli distributed graph attributes which are regressed independently solely on the intercept for each graph attribute. This means, we assume that each graph attribute $a \in \mathcal{J}$ of each graph $i = 1, \ldots, n$ is independently Bernoulli distributed, i.e. $Y_{i,a} \sim \text{Ber}(p_{i,a})$ with $p_{i,a} \in [0, 1]$. Then, our model for each $a \in \mathcal{J}$ is given by

$$h_a(x_i) = \beta_{0,a}.$$

Moreover, since we do not have any smooth terms and the Bernoulli distribution is a distribution of the EF as shown e.g. in (Fahrmeir et al. 2013, chp. 5.4), the model corresponds to a GLM and we define the model even more specific by a logit model

$$h_a(x_i) = g(p_{i,a}) = \beta_{0,a}$$

where g is the logit link function. For logit models, we refer the reader e.g. to Fahrmeir et al. (2013, chp. 5.1) and we use some of their results for the following calculations.

We restrict further to graphs with two nodes and edges, i.e. $J = 4, \mathcal{A} = \{0,1\}^4$. Additionally, we assume to observe a sample of size $n = 4$ with four unlabeled, directed graphs $(1,[y_1]),(1,[y_2]),(1,[y_3]),(1,[y_4]) \in \mathbb{N} \times \mathfrak{G}$ with the corresponding flattened representatives $y_1 = (1,0,0,0), y_2 = (0,0,0,1), y_3 = (0,1,1,0)$ and $y_4 = (1,1,1,1)$ where $y_{i,1}, y_{i,4}$ correspond to the two nodes and $y_{i,2}, y_{i,3}$ to the edges with $i = 1,\dots,4$ for the different representatives. Then, in the algorithm 3 in the initialization step, we assume to align all representatives w.r.t. the third one. Since all graphs are already optimally aligned, there is no realignment. In a regression step, we maximize for each $a \in \mathcal{J}$ separately the log-likelihood

$$\begin{aligned} \ell(\beta_a) &= \sum_{i=1}^{n} y_{i,a} \log(p_{i,a}) - y_{i,a} \log(1 - p_{i,a}) + \log(1 - p_{i,a}) \\ &= y_{i,a}\beta_{0,a} - \log(1 + \exp(\beta_{0,a})). \end{aligned} \tag{3.35}$$

Setting the corresponding score function to zero, we obtain

$$\hat{\beta}_{0,a} = 0, \quad \hat{p}_{i,a} = 0.5,$$

for all $i, a \in \{1,\dots,4\}$ and $\hat{p}_{i,a}$ do not differ for different i since we regressed only on the intercept. Thus, we obtain with $x_i = 1$ the fitted regression function

$$h(x_i) = (h_1(x_i), h_2(x_i), h_3(x_i), h_4(x_i)) = (0.5, 0.5, 0.5, 0.5), \quad i = 1,\dots,4.$$

If we realign the representatives $y_1,\dots,y_4$ in a next step of algorithm 3 w.r.t. these points $h(x_i)$ using the optimal alignment definition w.r.t. points from above, all are already optimally aligned and the algorithm stops. Nevertheless, we are not at a local maximum of the from (3.34) translated objective

$$\ell_{\mathfrak{G}}(\beta) = \min_{T \in \mathcal{T}} \sum_{a=1}^{J} \ell(\beta_a)$$

where $\ell(\beta_a)$ is defined as in (3.35). Instead, this objective would be maximized by permuting node one and two of the representatives y_1 or y_2 but not for both representatives at the same time. Thus, algorithm 3 does not converge to a local maximum of objective (3.34) viewed as a function of β for the considered Bernoulli distributions on the graph attributes although this sample would be observed with a positive probability in our example. This can happen since the probability of representatives on the boundary of FDs is not zero (y_3 is e.g. a singular graph).

The example highlights multiple issues for the existing regression frameworks in the graph space. First, when we extend the existing linear and additive regression frameworks to account in the single target models properly for the space of the graph attributes, this cannot be done straightforwardly by extending the single target models to their generalized forms. Theoretically, this is because the regression functions do not lie in the graph space, optimal alignment has to be redefined and a reasonable objective in the graph space has to be derived. Moreover, due to the probabilistic nature of the latter inherent in the likelihood maximization for the single target models, we lose the advantage of the plausible geometric (penalized) LS objectives in the linear and additive regression framework. Thus, to derive reasonable objectives for generalized regression models in the graph space, we require distributions on this graph space which are not simply investigated due to the quotient space structure of the space. A similar observation can be made for the practical part of the frameworks. The theorems on the AAC algorithms here and in Calissano et al. (2020, 2022) combine probabilistic assumptions on the sample with almost sure convergence results to local minima of (penalized) LS objectives. We saw in the example that even simple, practical distributional assumptions on the space of graph attributes can lead to difficulties w.r.t. the convergence of the algorithms. In particular, the absolute continuity w.r.t. Lebesgue measures of the probability measures in the theorems should be investigated when applying the AAC algorithms 1 and 2 because they do not necessarily converge to a local minimum of the corresponding objectives (3.6) and (3.27) viewed as functions of the parameters if graphs on the boundary of FDs are sampled with positive probability. Additionally, this combination of probabilistic assumptions on the sample to obtain almost sure convergence results and geometric regression approaches utilizing ((P)LS) objectives might be also problematic when we add edges with null attribute to obtain complete graphs because we obtain positive probability on null attributes leading to graphs such as singular graphs on the boundaries of FDs. These thoughts can be slightly related to the algebraic, geometric and probabilistic perspective on the Euclidean mean given e.g. in Fletcher (2020, section 2).

In summary w.r.t. the AAC algorithm, for a reasonable extension to GAMs the assumption of absolute continuity of $\mathbb{P}_{[Y]}$ w.r.t. the push forward measure of the Lebesgue measure has to be weakened. Then, major challenges are to develop a reasonable probabilistic objective in the graph space and that the regression function and/or the observed graphs could contain graphs on the boundaries of FDs with positive probability. Then, for example almost sure convergence to a local maximum of the objective (3.33) cannot be proven for the AAC algorithm 3. Furthermore, the linear and additive regression frameworks combine in their argument in favor of the AAC algorithm probabilistic and geometric regression perspectives which should be done thoughtfully.

4. Model considerations

This section examines the developed regression frameworks in more detail. First, we motivate modeling in the graph space in general and argue why an extension of the linear regression framework from Calissano et al. (2022) to additive models and GAMs is reasonable. Subsequently, possibilities to interpret the results are outlined. Then, we discuss alternative penalty terms of the extension to additive models. Afterwards, we describe which type of covariates are reasonable for the regression framework. We discuss computational issues of the AAC algorithms connected with the graph matching problem and give a recommendation for the maximal number of nodes. Finally, a description of the implementation of the AAC algorithm 3 is given.

4.1. Motivation and interpretation for graph space regression

First, we motivate models in the graph space reviewing existing applications and their abstractions. General data analysis in the graph space can be motivated from the perspective of object oriented data analysis as explained in Marron & Dryden (2021). An application arises e.g. if we observe unlabeled graphs. Then, we could create node labels by random enumeration to obtain corresponding representatives in the total space. Next, to work with these representatives of the unlabeled networks, they have to be aligned to be comparable again which is done in the graph space setting. In this case, the node labels possess no relevant information for the analysis and data analysis in the graph space could be reasonable. This motivated the application in Guo et al. (2021) on a data set of handwritten letters. Each letter is treated as unlabeled graph where nodes are connected by shapes such that one graph corresponds to one letter. Afterwards, the node labels are serially numbered to obtain representatives in the form of adjacency matrices. Then, the aim was to obtain pairwise matches, geodesics and means between the letters. This was achieved by realigning the representatives again and executing the computations in the total space between aligned graphs. The outcomes could be interpreted as results for the handwritten letters, i.e. the unlabeled graphs. In a similar fashion, the authors in Guo et al. (2022) matched the nodes of brain arterial networks for in general otherwise unknown matches. Other common examples for unlabeled networks are social networks in which we are interested in the patterns/structure in groups of related entities. Then, the labels are either not important or not available, cp. e.g. the discussion in (Kolaczyk et al. 2020). Finally, a similar application can be found in Beiler et al. (2015). Therein, the nodes are trees and the edges are fungal genets. Thus, the networks arise naturally as unlabeled networks and one is interested in the topology of the interaction networks.

Closer related to this thesis, we aim to model the relationship between covariates and unlabeled graphs in the regression setting. Thus, the regression model should account properly for the space of the graphs which leads to the graph space as space of the responses. An application arises e.g. for observations of labeled networks with different node labels but the same type of edge measurements as in Calissano et al. (2022, section 4.3). Therein, the authors regressed

passing networks of football teams on goals scored in a game. Although the players and thus, the node labels, differed between the games, unlabeled passing networks w.r.t. goals scored could be modeled by matching different players across labeled passing networks which have a similar role (more precisely, by matching players with nodes on their corresponding regression function evaluation, but for comprehensibility simplified here). Similarly, this might be possible for multiple logistics networks with various node labels. Different to this, we can analyse observations of labeled networks with fixed labels by restricting the analysis and observed networks from the beginning to the graph space as is done e.g. in two examples in Calissano et al. (2022, section 4.1,4.2) and in this thesis in section 5 for air transportation networks. In this case, we could model the labeled or the unlabeled networks – treating the observed labeled graphs as unlabeled graphs – but we have to adjust the interpretation of the results with regard to the analysed data object.

Given the former examples, an advantage of the unlabeled model might be, that it is easier to compare the outcome to the same models on data with the same type of edge measurements but different node labels, e.g. in our case the comparison of the European to the American or Asian air transportation networks. Then, since the node labels are arbitrary, a comparison might be more reasonable using models in the graph space because we do not have to match nodes between different data sets manually. This comparison over multiple data sets is again related to the football passing model. Furthermore, the treatment of labeled networks as unlabeled networks and labeled networks again could be related to the handwritten letter example when we treat e.g. country labels in air transportation networks as random enumeration. In contrast to this, we have to be careful when we interpret the results again w.r.t. to the node labels of the observed labeled networks, especially for regression models. This is due to the regression function is a function into the graph space and observations are not aligned w.r.t. each other but w.r.t. this regression function. This subtle aspect is also discussed at the end of this subsection for the interpretation of the model.

Another advantage of graph space modeling could be, that if we are interested in the development of e.g. the air transportation network as a whole, the development might be reproduced smoother in the graph space regression model. For example, if Germany passes severe restrictions on air transportation and its role in the European air traffic network is temporarily absorbed by France, the captured transition in the graph space model can be smoother due to the possible permutation of France and Germany. In summary, the regression frameworks require thoughtful handling of the object that we want to model compared to the one that we actually observe. If the object is the structure of the network, the graph space modeling might be appropriate.

Next, we want to review the motivation behind the extensions of the linear regression framework. First, we extended the linear regression framework to additive models in $\mathfrak{G}$ by allowing for additive models in the single target models. The main incentive behind this extension is analogous as for the Euclidean counterparts. Namely, additive models allow for a more flexible relationship between the response and the covariates. In our case, this means that we allow for smooth functions between the covariates and the graph attributes of the unlabeled graphs instead of solely linear relations.

Second, we discussed the possibility to extend additive models to GAMs in $\mathfrak{G}$ by extending the

single target models to GAMs. Again, the motivation to model non-normal responses from the equivalent extension for Euclidean responses transfers to our case. In particular, this extension is necessary to model responses with binary, categorical or count data. This means, we need the extension to GAMs to include graphs with discrete graph attributes. Moreover, this is also the case when we observe networks with graph attributes in $\mathbb{R}_{\geq 0}$. The linear regression framework does not account for such graph attributes and we could obtain graph attributes smaller than zero in our model. Hence, the necessity for GAMs for these types of graphs.

Finally, we want to emphasize that the existing results on the almost sure convergence of the AAC algorithms 1 and 2 to a local minimum should be treated carefully. The connection between the geometric approach of the regression model and the probabilistic reasoning for the AAC algorithms seems not straightforward. As we described for additive models and GAMs in subsection 3.4, the almost sure convergence does not hold as soon as graphs on boundaries of FDs are involved with positive probability. However, this could be already the case if we add edges with null attribute to the graphs since then, these edges are assigned positive probability. Thus, if we observe non complete graphs but extend them to complete graphs by adding edges with null attributes for unobserved edges, this could have impacts on the convergence of the AAC algorithms. Nevertheless, this is done and not discussed in existing applications and its impact on the convergence of the algorithm should be investigated in detail.

Interpretation of regression frameworks in the graph space

So far, there exist different possibilities to interpret the model in practice which we discuss until the end of this subsection. The first two, already existing approaches analyse the permutations of the observed graphs. The first is illustrated in (Calissano et al. 2022, Figure 3) and introduced by the same authors. It provides an interpretation on graph attribute level in the total space and is based on permutations of the nodes. The idea is to plot the permutation frequencies of the labeled nodes w.r.t. other labeled nodes. The approach is e.g. applicable if we observe labeled networks but model them in the graph space as described above. Then, one could say that labeled nodes which are permuted often with each other are similar w.r.t. their role in the labeled network while nodes that are not permuted have unique roles in the labeled network. Thus, this would aim to obtain a first idea of the roles of the nodes in the total space compared to each other. Similarly, we can plot the permutation of a node w.r.t. all other nodes against the covariate, e.g. the permutation of a node against time as in Calissano et al. (2022, Figure 7). Again, we obtain an interpretation possibility on which nodes have similar roles in the total space given covariate values. Both interpretations could be problematic since they work with the roles and labels of the nodes in the total space while modeling the networks in the graph space. For example, two nodes that are interchanged at specific covariate values do not necessarily have the same role in the networks w.r.t. each other but instead, one satisfied the role for one region of the covariate while the other satisfied the role for another region of the covariate, where the role is understood as the role of the node of the unlabeled regression function in the graph space. Moreover, both approaches focus on the observed networks instead of on the regression function itself. However, thoughtful interpretation of both approaches might be helpful especially

when observing labeled networks and treating the metric in the graph space as minimizer over permutations for node labels with inherent information.

Besides that, another possibility on graph attribute level might be to assign 'artificial labels' (ALs) to the nodes of unlabeled networks obtained from the regression function f evaluated at specific covariate values in the graph space. It is not yet clear, how to develop these ALs. Nevertheless, this approach aims for a direct interpretation of the regression function f in the graph space on graph attribute level. The advantage of such an approach is that the functional assumptions from the single target models in the total space might be captured. This means, we would enable an interpretation of the regression parameters β and the smooths from the total space in the graph space. We exemplify this in appendix B. Therein, we describe that the parameters of the total space can be interpreted indeed as piecewise parameters on the graph attributes in the graph space which are interchanged when the regression function h crosses a FD. Additionally, we present methods to obtain the crossing points.

A disadvantage of the former approach is that we have k^2 potential graph attributes to analyse, where k denotes the number of nodes. In contrast to this, we could interpret the model by analysing the whole networks from the function f evaluated at different covariate values in the graph space and their variation w.r.t. the covariates. The advantage would be that we analyse the unlabeled network on a higher level than the graph attributes. This could be reasonable since we model the unlabeled network as data object in our regression frameworks. This can be done e.g. by visualisations of the unlabeled networks along the covariates. Alternatively, we could use network summary statistics. Depending on the summary statistics, this would lower the potential functions to analyse, compared to k^2 of the former approach. A disadvantage might be that it seems difficult to preserve interpretability of parameters and smooths of the single target models from the total space in the interpretation of functions of the network summary statistics. Thus, we lose a general advantage of regression models, namely their interpretability via parameters and smooths. Nevertheless, this approach analyses again the regression function f in the graph space instead of the observed graphs and their permutations. We describe this approach briefly in section 5 and exemplify it in the respective figures E.12 and E.13 in the appendix.

Finally, we summarize the differences between the interpretations. First, we can separate approaches on the observed graphs and on the regression function in the graph space. Second, we can differentiate between interpretations on graph attribute and graph level. Apart from this, the distinction between interpretations referring to labels from the total space and ones referring to potential labels from the graph space is nuanced and should be thought through.

4.2. Discussion on the penalty term for additive models

This subsection discusses the penalty term in (3.27). So far, this term aims to penalize, separately for each graph attribute $a \in \mathcal{J}$, the wiggliness of the smooth functions $f_{1,a}, \ldots, f_{p,a}$ of the additive model in (3.24). This is done using parameter vectors β_a corresponding to the regression functions h_a from (3.25) with optimally aligned representatives $y_1, \ldots, y_n$. Thereby, we allow for different smoothing parameters $\lambda_{j,a}$ for each smooth $f_{1,a}, \ldots, f_{p,a}$, we penalize them independently for each graph attribute $a \in \mathcal{J}$ and the parameters are penalized on graph attribute level of labeled

graphs in the total space. In particular, if for fixed $a \in \mathcal{J}$, $j = 1, \ldots, p$ it holds $\lambda_{j,a} \to \infty$, then the smooth $f_{j,a}$ of covariate j tends for graph attribute a to a function in the null space of the penalty of this smooth. If for fixed j and all a it holds $\lambda_{j,a} \to \infty$, then the smooth of covariate j tends to a function in the null space of the smooth for the whole graph.

To obtain another perspective, we rewrite the penalty for a concrete example. We examine an additive model defined by an univariate smooth without linearly modeled covariates, with P-splines and B-spline basis and differences of order one. This can be translated from subsection 3.3.2 and we rewrite the penalty term from (3.27) to

$$\begin{aligned}
&\sum_{a=1}^{J} \sum_{j=1}^{p} \lambda_{j,a} \beta_a^\top \mathcal{S}^{[j]} \beta_a = \sum_{j=1}^{p} \sum_{a=1}^{J} \lambda_{j,a} \beta_a^\top \mathcal{S}^{[j]} \beta_a \\
\overset{(p=1)}{=} &\sum_{a=1}^{J} \lambda_{1,a} \sum_{u=2}^{l} (\beta_{u,a} - \beta_{u-1,a})^2 = \sum_{u=2}^{l} \sum_{a=1}^{J} \lambda_{1,a} (\beta_{u,a} - \beta_{u-1,a})^2 \\
= &\sum_{u=2}^{l} d_{\mathcal{A}}^2(\tilde{\beta}_u, \tilde{\beta}_{u-1})
\end{aligned}$$

where $\tilde{\beta}_u = (\beta_{u,1}, \ldots, \beta_{u,J})$ and $d_{\mathcal{A}}^2$ is the squared weighted Euclidean distance with weights $\lambda_{1,a}, a \in \mathcal{J}$. To clarify that $\tilde{\beta}_u$ differs from β_a in its form (indexing over rows instead of columns), we write the tilde. The second equality follows from the definition of penalties of order one for P-splines. Now, we can observe that this rewritten penalty treats the parameter vectors $\tilde{\beta}_u, u = 1, \ldots, l$ for the smooth as labeled graphs, where each labeled parameter graph $\tilde{\beta}_u$ corresponds to the covariate evaluated at the basis function B_u from subsection 3.3.2. Thus, we penalize the wiggliness of the function across the parameters treated as labeled graphs while simultaneously, still allowing for independent smoothing parameter $\lambda_{1,a}$ for each graph attribute.

Next, we describe alternatives to the penalty in (3.27). The second last sentence provokes a possible alternative penalty. Instead of different smoothing parameters for each graph attribute $a \in \mathcal{J}$, we could restrict to one smoothing parameter $\tilde{\lambda}_1$ for the whole graph by

$$\sum_{u=2}^{l} \sum_{a=1}^{J} \lambda_{1,a} (\beta_{u,a} - \beta_{u-1,a})^2 = \sum_{u=2}^{l} \tilde{\lambda}_1 \sum_{a=1}^{J} (\beta_{u,a} - \beta_{u-1,a})^2 = \sum_{u=2}^{l} \tilde{\lambda}_1 d_{\mathcal{A}}^2(\tilde{\beta}_u, \tilde{\beta}_{u-1})$$

where we start from the end of the second line of the calculations above. This would measure the wiggliness of the smooth by the sum of the squared differences $\beta_{u,a} - \beta_{u-1,a}$ in the parameters on graph attribute level while choosing the smoothing parameter for this wiggliness over the whole labeled graph in the total space. Analogously, we could penalize for the wiggliness e.g. across nodes and their incoming edges by

$$\sum_{u=2}^{l} \sum_{a=1}^{J} \tilde{\lambda}_{1,a} (\beta_{u,a} - \beta_{u-1,a})^2 = \sum_{u=2}^{l} d_{\mathcal{A}}^2(\tilde{\beta}_u, \tilde{\beta}_{u-1})$$

where $d_{\mathcal{A}}^2$ is the squared weighted Euclidean distance with weights $\tilde{\lambda}_{1,a}, a \in \mathcal{J}$ and we assume for

$\tilde{\lambda}_{1,a} = (\tilde{\lambda}_{1,1}, \ldots, \tilde{\lambda}_{1,J})$ that

$$\tilde{\lambda}_{1,1} = \ldots = \tilde{\lambda}_{1,k};\ \tilde{\lambda}_{1,k+1} = \ldots = \tilde{\lambda}_{1,2k};\ \ldots;\ \tilde{\lambda}_{1,J-k+1} = \ldots = \tilde{\lambda}_{1,J}.$$

Analogous models with penalties such as the group lasso model for multiple outputs/tasks would be also possible by translating them to (3.27) and the above considerations straightforwardly. The considered group lasso can be found e.g. in Hastie et al. (2015, Example 4.2). Moreover, we could connect multiple penalties as done e.g. in Jacob et al. (2008) for a multi task model formulation.

Alternatively, we could penalize the parameters more directly in the graph space. Contrary to all developed regression frameworks in the graph space which only permuted observations, this would also permute parameter vectors viewed as graphs. To do this, we utilize the metric formulation of the penalty term derived above. Then, when we project the PLS (3.26) in metric formulation to the graph space, we minimize the penalty term over all permutations for the first and second metric to obtain the following PLS in the graph space

$$PLS_{\mathfrak{G}} = \sum_{i=1}^{n} d^2_{\mathfrak{G}}([y_i], f(X_i)) + \sum_{u=2}^{l} \tilde{d}^2_{\mathfrak{G}}([\tilde{\beta}_u], [\tilde{\beta}_{u-1}]).$$

where $\tilde{d}^2_{\mathfrak{G}}$ is the weighted form of the metric (3.1) in the graph space with weights $\lambda_{1,a}$. This penalty measures the wiggliness of the smooth by treating the parameters $\tilde{\beta}_u, u = 1, \ldots, l$ as unlabeled graphs. If we do this while allowing for different smoothing parameters $\lambda_{1,a}$ for each graph attribute, we penalize for the wiggliness of the smooth for each graph attribute in the graph space. In particular, for $\lambda_{1,a} \to \infty$ for all $a \in \mathcal{J}$, this forces the smooth to the null space of the penalty for the whole graph. More generally, the penalty measures the wiggliness in the graph space. As above, we can restrict the smoothing parameters and those, penalizing e.g. for the wiggliness of the smooth across the whole unlabeled graph.

This example can be straightforwardly generalized to multiple smooths. Moreover, a generalization to other penalties is possible as long as we can obtain a metric formulation of the parameter vectors treated as graphs in the total space. In general and for future research, it would be helpful to compare the PLS objective (3.27) e.g. to the Fréchet regression model developed in Petersen & Müller (2019) to verify its meaningfulness.

4.3. General remarks on the framework

First, the covariates should be independent of the graph attributes. This means, that the covariates should not differ for various graph attributes or equivalently, that the model matrix X is independent of the graph attributes $a \in \mathcal{J}$. Otherwise, the prediction of graph attributes could be not feasible since the covariate values used for prediction are potentially not inside the support of the covariates for some graph attributes used for fitting the single target models. Moreover, it would be unclear which covariate values to use for predictions since we used for the fit of single target models potentially multiple graph attributes in the total space with different covariate values. This relates again to the discussion on observing labeled or unlabeled graphs in

subsection 4.1.

Second, it is complex to validate the frameworks. Since the regression models regress each graph attribute separately on the covariates, we desire distributions on the graph attributes of the graph space. Due to the permutations, it is difficult to derive theoretical distributions for the graph attributes in the graph space when starting with distributions on graph attribute level in the total space. However, a rather simplified approach would be to use just node attributes and assign to all edges null attributes. Then, we can derive the distributions of the graph attributes in the graph space as the distributions corresponding to the order statistic of the nodes from the total space since the permutations order the nodes solely by size in this restricted case. For example, if we assume that all node attributes in the total space are sampled from a uniform distribution on the unit interval, the corresponding theoretical distributions of the j-th ordered graph attribute in the graph space would be the beta distribution with parameters j and $j+1-k$. For this result on the order statistic compare e.g. Bickel & Doksum (2015, Appendix B.2.9b). Nevertheless, it is in general not straightforward to compute these marginal distributions of the joint distribution on the graph space. This complicates e.g. simulation studies.

Another issue arises for the computational costs of the frameworks. Due to the iterative application of graph matching and fitting of multiple regression models, the AAC algorithms are computationally costly. For example in the application in section 5, although we were able to parallelize the graph matching and the estimation of the GAMs, running the AAC algorithm 3 with the in the `graphspace` package implemented GAS graph matching algorithm once on a Humboldt Lab for Empirical and Quantitative Research (LEQR) server with 12 physical cores at 3.4 GHz still took at least six days for three iterations for a network with 27 nodes that is observed over 127 weeks (for more details, cp. the next subsection and section 5). Thus, we recommend to restrict the analysed networks to at most 30 nodes. In our application, the bottleneck of the computational time was the alignment of the graphs w.r.t. each other and the regression function. In particular, for example the in the `graphspace` package implemented GA graph matching algorithm was not exact enough for our application and we had to switch to the slower implemented GAS or GAS1 algorithm. In general, using faster graph matching algorithms and implementing them in the `graphspace` package might be a worthy improvement. An alternative way to shorten computational time is to reduce the time to estimate the GAMs for each graph attribute in the current alignment. This could be done by reusing already estimated coefficient vectors of previous steps of the AAC algorithm. In the simplest case, if a node is not permuted from one step to the next for all observations, we can reuse the estimated GAM for this node attribute. Moreover, if two nodes are not permuted from one step to the next for all observations, we can reuse the estimated GAM for the edge attribute between these two nodes. Additionally, the search for smoothing parameters could be adjusted depending on smoothing parameters of the last iteration.

We note that although the regression framework might seem restrictive with a maximum of 30 nodes per graph, it can be helpful to aggregate nodes in larger networks to obtain smaller networks. This is for example done for the air transportation networks in section 5 by investigating them on country instead of airport level. After the model is fitted, we might want to decompose the estimated graph attributes back to their non aggregated form. Practically, this could e.g. be

done using multinomial models on the graph attributes.

As a final remark on the existing regression frameworks, we indicate that they should be investigated in more depth w.r.t. the projection of the regression function into the graph space and its implication on the frameworks. In particular, we assume currently reasonable regression functions in the total space for the multi output regression function h while we aim to minimize e.g. in the linear regression framework in $\mathfrak{G}$ an intrinsic LS objective in the graph space. However, it is unclear if the regression function f in $\mathfrak{G}$ corresponding to reasonable regression functions h in $\mathcal{A}$ is still sensible. This is related to the discussion on the penalty term in subsection 4.2 on whether to penalize wiggliness in the total or in the graph space. Furthermore, it is related to the discussions on geometric and probabilistic approaches above and at the end of subsection 3.4. If we review for example the linear regression model in the total space in matrix notation in (3.8), the error terms corresponding to the probabilistic regression perspective live in the total space instead of the graph space. This became even more apparent in the development of the additive and generalized additive regression frameworks since we started with the single target models and their objectives on graph attribute level in the total space. Additionally, it is discussed in appendix B.

4.4. Implementation of the framework

The graph space and the linear regression framework are implemented in python (Van Rossum & Drake 2009) in the `graphspace` package available via GitHub at Calissano (2020). The package is currently moved into the package `geomstats` described in Miolane et al. (2020), which is a package for geometry based statistics. The `graphspace` package allows for the definition and work with graphs and sets of graphs. For the alignment of pairwise graphs e.g. the generalized Bron Kerbosch algorithm (Jain & Obermayer 2011) and the graduate assignment algorithm (Gold & Rangarajan 1996) are available. Moreover, AAC algorithms to compute Fréchet means, geodesic principal components and the linear regression in the graph space are implemented. Based on the package, our main contribution was to implement GAMs as described in subsection 3.4 into the existing framework. In a first step, we used the `multiprocessing` package included in the standard library of python to parallelize the alignment of graphs and the estimation of GAMs. To the best of our knowledge, there did not exist a package in python to fit GAMs with a REML approach for the estimation of the smoothing parameters. Thus, we switched to the statistical software R (R Core Team 2022) and used the `mgcv` package for which a practical tutorial is available in Wood (2017). As described in AAC algorithm 3, model fitting is just one step in each iteration. Thus, it was necessary to run the R code during the compilation of the python code. Therefore, we used the `rpy2` package (Gautier et al. 2008). In summary, we fit the GAMs with the `mgcv` package in R and align the graphs within the `graphspace` package in python. Estimation and alignment are both parallelized.

5. European air transportation network during the Covid-19 pandemic

This section describes the application of the developed GAM framework to the air transportation network of parts of the EU aggregated on country level during the Corona Virus Disease 2019 (Covid-19) pandemic. In particular, we approximate for the response the weekly air passenger networks on country level for the ten countries in the EU with the most passengers between January 2019 and September 2021, treated as unlabeled networks in the analysis. As covariates, we consider time, new daily Covid-19 cases in the ten countries averaged by week, and months. The application aims to implement the GAM framework in the `graphspace` package in python and to apply it to real world data.

The application has various merits as well as flaws. First, we work with the total space $\mathcal{A} = \mathbb{R}^J_{\geq 0}$ and the corresponding graph space $\mathfrak{G} = \mathcal{A}/\mathcal{T}$. This is common since it corresponds to networks with one-dimensional, positive and continuous graph attributes. Furthermore, we observe a time series of networks together with Covid-19 cases which is interesting due to its dynamic structure as well as its epidemiology and mobility related background. However, as described e.g. in Kolaczyk & Csárdi (2020, chp. 11), dynamic network modeling is not well developed yet and the time related component of the data is not sophisticatedly modeled by us. Moreover, since we observe labeled graphs and treat them as unlabeled graphs, the reasoning to use a model in the graph space instead of the total space might be unclear. Some arguments in favor of the graph space can be found in subsection 4.1 while we have to keep in mind that the analysed objects differ in both cases. Another disadvantage is that the covariates are not necessarily real valued, although we derived all models for real valued covariates. Apart from this, the developed generalized additive framework is applied to real world data which comes with data related difficulties such as only approximate passenger numbers between countries. Additionally, the application is relatively large and complex which makes it more delicate to investigate the theoretical GAM framework instead of the application related challenges. Moreover, we described in subsection 3.4 that the GAM framework is not well developed yet and should be not employed without caveat. However, the application to real world data is an important step to show that the derived frameworks can be actually useful in practice.

The section is structured as follows: first, we describe the data. Next, we define the regression model by defining the single target GAMs and its implementation in python and R. Finally, we summarize the results.

5.1. Data description

This subsection describes the data sources, the extracted raw data and its preprocessing. A summary of the data sources can be found in table 5.1. Our aim is to analyse weekly air passenger networks for the ten countries with the most passengers between 2019 and 2021 in the EU. As explained e.g. in Sun et al. (2021, section 2.2), a common limitation for studies of air passenger

traffic is that there exists no openly accessible data source with information on the daily number of air passengers between and within countries. Especially when we analyse the air traffic during the Covid-19 pandemic, it is important to substitute the air passenger network not solely by the flight network. There exist different reasons why a substitution of the passenger network by the flight network is not reasonable. For example, airlines had to fly empty flights to keep their slots at airports during the Covid-19 pandemic. A demonstration of this problem can be found in the regulation "Council Regulation (EEC) No 95/93 on common rules for the allocation of slots at Community airports" (2020-03-31). Another difference between flight and passenger networks is that the flights are performed with different aircrafts and thus, possibly different passenger capacities per flight. Furthermore, the occupancy rates can differ between flights. This means, that different flights do not relate necessarily to the same number of passengers. Additionally, there exist cargo flights in the flight network and cargo and passenger traffic behaved differently during the pandemic, cp. e.g. Sun et al. (2021). To address these difficulties partly, we extracted the daily number of flights and used them as weights on the monthly passengers between two countries. In addition, this could be improved further using a similar approximation as in Suzumura et al. (2020). The authors also analysed the OpenSky network data and approximated the passengers by accounting for aircraft types and their corresponding capacities. However, at time of submission of this thesis, there was no data freely available to us that connected the type codes of the aircrafts to their capacities.

Data source	Data
Strohmeier et al. (2021)	Global, daily flights between 2019 and 2021 of OpenSky
Eurostat (2022a)	Monthly flights and passengers between reporting countries
Eurostat (2022b)	Glossary for EU country codes
Eurostat (2022c)	Yearly total air passengers by countries of the EU
Dong et al. (2020)	Daily Covid-19 data of John Hopkins University

Table 5.1.: Data sources

In a first step, we profited from the OpenSky network community and its crowdsourced project for global air traffic data introduced in Schäfer et al. (2014). We summarize in the following briefly the data description from Strohmeier et al. (2021). The project is based on receiver data capturing radar signals from aircrafts. In particular, the Automatic Dependent Surveillance–Broadcast (ADS-B) protocol provides every second identity, position, status and velocity of individual flights. This information can be captured by receivers and is collected in the OpenSky network database. Since 2020, ADS-B is required in many airspaces and thus, aircrafts are commonly equipped with it. So far, the data consisted of encoded ADS-B data, noise of the measurement process, duplicate measurements of flights due to multiple receivers and other quality obstacles. Thanks to the work in Strohmeier et al. (2021), this data was preprocessed and made available freely starting from January 2019. The processed data consists of the flights covered by the OpenSky database between January 2019 and February 2022 including an identification option for individual flights, type codes of the aircrafts, origin and destination airports and timestamps.

Next, we restricted the time period to January 2019 until September 2021 since the covariates were available during this time period. Furthermore, we included only flights with origin and

destination airports in the ten countries with the most air passengers during 2019 till 2021 which were obtained summing over yearly passenger data from Eurostat (2022c). Namely, the ten countries are Belgium, France, Germany, Greece, Italy, Netherlands, Poland, Portugal, Spain and Sweden. We confined the analysis to ten countries due to the computational costs of the graph matching algorithm GAS implemented in the `graphspace` package. Due to the same reason, we aggregated all airports for each country to obtain origin and destination countries instead of origin and destination airports. This results in a dataset of daily flights between the ten countries on country level between January 2019 and September 2021 covered by the OpenSky network.

Afterwards, we collected the monthly total number of passengers between two countries within the EU from Eurostat (2022a). Then, we propose the following, continuous approximation for the daily number of passengers between country i and j:

$$\begin{aligned} &\text{for } i \neq j: \\ &\qquad \tilde{y}^d_{i,j} = p^m_{i,j} \frac{f^d_{i,j}}{f^m_{i,j}} \\ &\qquad y^d_{i,j} = y^d_{j,i} = \tilde{y}^d_{j,i} + \tilde{y}^d_{i,j} \\ &\text{for } i = j: \\ &\qquad y^d_{i,j} = y^d_{i,i} = p^{nat,m}_i \frac{f^d_{i,j}}{f^m_{i,k}} \end{aligned} \tag{5.1}$$

where $k =$ number of countries, $i, j \in \{1, \ldots, k\}$, $y^d_{i,j}$ is the edge attribute between country i and j of day d, $p^m_{i,j}$ are the international passengers arriving at country j from country i in month m, $p^{nat,m}_i$ are the domestic passengers in country i in month m, $f^d_{i,j}$ and $f^m_{i,j}$ are the flights from country i to j and the superscripts d and m denote the corresponding day and month, respectively. For $i = j$, we obtain the daily domestic passengers of country i and for $i \neq j$, the daily international passengers between i and j. Afterwards, we computed weekly averages for each graph attribute.

Finally, the preprocessing for the response results in weekly flattened adjacency matrices of labeled, undirected, weighted networks y_i where $i \in \{1, \ldots, 143\}$ is the index for each week between January 2019 and September 2021, i.e. $n = 143$. To be precise, each labeled graph is given by the flattened adjacency matrix:

$$y_i = (y_{i,1}, \ldots, y_{i,J}), \qquad i = 1, \ldots, n.$$

The networks have $k = 10$ nodes and thus, $J = k^2 = 100$ graph attributes and we define $\mathcal{J} = \{1, \ldots, J\}$. The node labels are the country names and the graph attributes are approximated domestic and international weekly passengers for the ten countries.

Before we proceed, we evaluate the quality of the flight data from OpenSky network. Therefore, Eurostat provides monthly data on the total number of commercial flights between countries in Eurostat (2022a). To obtain an impression how many commercial flights are captured by the OpenSky network, we compute the ratios of OpenSky network flights and Eurostat commercial flights for each country and each month between January 2019 and September 2021. A heatmap

of these ratios can be found in figure 5.1. The countries are plotted on the vertical and the months with corresponding year on the horizontal axis. We see that for Greece the coverage of the OpenSky network is relatively low and the ratios vary between 0.1 and 0.5. Similarly, the ratios for Spain were just around 0.6 till January 2020. In contrast to this, the months April till June 2020 are clearly the brightest in the heatmap. This means, that the OpenSky network captured more flights than commercial flights measured by Eurostat. One reason for this is that the restrictions on the commercial air transportation network were severe during this time. Thus, on the one hand there were not many commercial flights measured by Eurostat while, on the other hand, OpenSky network still observed non-commercial flights such as e.g. cargo flights or empty flights of airlines to keep their airport slots. We try to account for this in our approximations (5.1) and would like to improve it in the future with the inclusion of aircraft capacities. Alternative ways to improve the data would be e.g. to account also for first and last seen locations of flights although their landing was not observed in the OpenSky network. Additionally, we could use a commercial flight tracker with better coverage such as https://www.flightradar24.com/. However, we have to keep in mind that our passenger networks are solely approximations and that weighting with flights can have severe, unclear impacts on the results.

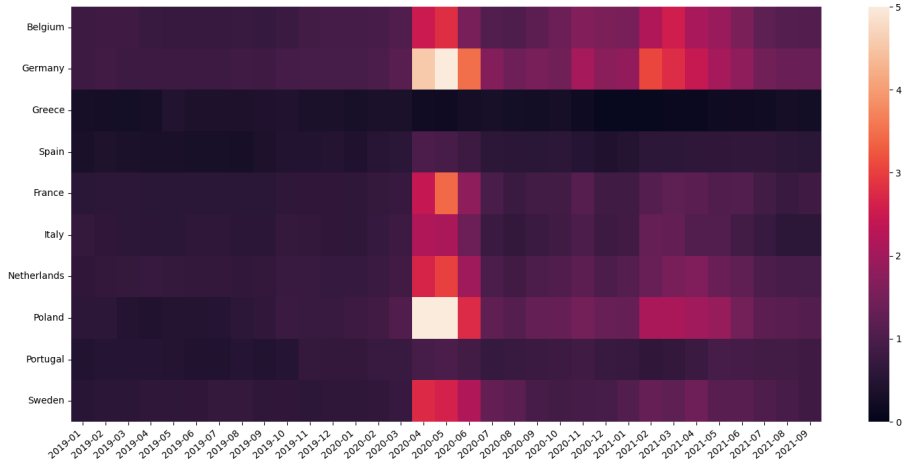

Figure 5.1.: Heatmap of the ratios of OpenSky network flights and Eurostat commercial flights for each of the ten countries with the most passengers between 2019 and 2021 in the EU and the months between January 2019 and September 2021. The brighter the entry, the larger the corresponding ratio.

In a last step, we assign covariates to the graphs and define a graph set in python with the `graphspace` package. Namely, we assign each labeled network the current week, new daily Covid-19 cases aggregated over all ten countries in this week, and the current month. This results in

$$(y_i,\ t_i,\ c_i,\ m_i), \quad i = 1, \dots, 143 \tag{5.2}$$

where $t_i \in \{1, \dots, 143\}$ is a vector for the current week encoded from 1 to 143, c_i is a vector of mean weekly new Covid-19 cases within the countries and $m_i \in \{1, \dots, 12\}$ a vector for the

current month. Daily new Covid-19 cases for the countries are computed using daily new Covid-19 cases on country level from Dong et al. (2020). We note that the covariates $t_i, c_i, m_i, i = 1, \ldots, n$ are chosen to be independent of the graph attributes, i.e. independent of $a \in \mathcal{J}$, for the reasons discussed in subsection 4.3.

An interesting extension of the application done in this thesis would be to consider e.g. daily instead of weekly data. This leads potentially to less smooth functions due to the additional variability in responses, especially through zero passengers at some days and hundreds of passengers at other days for some connections. Further, the computational costs of the model would increase, although just linearly in the graph matching step of the algorithm, which is currently the bottleneck w.r.t. computational costs. However, we would also obtain daily predictions of the network which might be more valuable for e.g. airlines to coordinate their aircraft fleets. Alternatively, the ratio of graph attributes $y_{i,a}, a \in \mathcal{J}$ and populations of the countries corresponding to this graph attribute a could be analysed. This would lead possibly to more similar graph attributes and thus, a larger possibility for permutations. However, the modeled object would differ to the analysed object in here. Moreover, we could consider the air transportation network of all member states of the EU. So far, this was not possible since the in `graphspace` implemented graph matching algorithm GAS was too slow (at least six days for three iterations on one of the LEQR servers), GAS1 was not able to match such large networks and GA was not exact enough such that regression errors from one iteration to the next in the AAC algorithm 3 could increase largely. The latter is a reason to be careful when applying the GA graph matching algorithm to a similar analysis.

5.2. Model

This subsection describes the generalized additive framework in $\mathfrak{G}$ applied to the approximations of air passenger networks treated as unlabeled networks and its implementation.

We present two ways of analysing the air transportation network similar to the analysis of the public transportation network in Copenhagen during the Covid-19 pandemic in Calissano et al. (2022). First, we can analyse the development of the labeled network during the Covid-19 pandemic. Therefore, the nodes are not permuted and it corresponds to an analysis with single target GAMs for each graph attribute in the total space $\mathcal{A}$. Alternatively, we model the development of the unlabeled network in $\mathfrak{G}$. Hence, we allow for permutation of the nodes such that the networks are optimally aligned w.r.t. the regression function. The difference in the approaches can be understood in terms of the data object that we observe. In the former approach, the passengers between the ten countries are the target. In the latter, the network structure is modeled and the country labels are not of interest. In a further analysis, this structure could be compared for example to the development of the network structure of the air passenger transportation network within the United States (US), i.e. within and between the states of the US. The structural comparison of the networks is easier since the networks are made independent of the countries when they are permuted. We describe in the following the theoretical model in the graph space.

For the graph space model, we consider the sample

$$([y_i], x_i) = ([y_i],\ t_i,\ c_i,\ m_i), \quad i = 1, \dots, n \tag{5.3}$$

obtained from (5.2), where we treat the labeled networks $y_i \in \mathcal{A} = \mathbb{R}^J_{\geq 0}$ as representatives of unlabeled networks $[y_i] \in \mathfrak{G} = \mathcal{A}/\mathcal{T}$ to be able to analyse the structure inherent in the networks and the covariates are also obtained from (5.2). In addition, the last four weeks are removed from (5.3) for the fit of our model such that $n = 139$ because we want to predict unseen observations afterwards, i.e. we split into training and test set. Then, we assume the following generalized additive model in $\mathfrak{G}$

$$f : \mathbb{R}^3 \to \mathfrak{G}, \quad x \mapsto f(x) = \pi \circ h(x)$$

where $h : \mathbb{R}^3 \to \mathcal{A}, x \mapsto h(x) = (h_a(x))_{a \in \mathcal{J}}$.

In a next step, it is necessary to decide on a distribution of the EF for the graph attributes $y_i, i = 1 \dots, n$ in the total space to model their linked mean by $h_a, a \in \mathcal{J}$. The distribution should be flexible enough to model each of the 100 graph attributes separately. For example, approximated passengers between the Netherlands and Greece vary between zero and 15000 and attain exactly zero, while approximated domestic passengers in Spain vary between 1000 and 165000 passengers. Hence, we can observe that we need a distribution on the positive real numbers including zero. Additionally, the distribution should be as flexible as possible, since the developments of the graph attributes w.r.t. the covariates differ strongly. This means, we should use a flexible distribution with the possibility of positive probability on zero.

The Tweedie distribution (Tweedie 1984) belongs, under certain assumptions on its parameters, to the EF. In particular, it is uniquely defined by a scale parameter, its mean and an additional parameter φ that determines the mean variance relationship. For detailed descriptions of the distribution, we refer the reader to Jørgensen (1987); Gilchrist & Drinkwater (2000) where we summarize briefly some results from. For an overview of related distributions given φ cp. e.g. Jørgensen (1987, table 1). For values $1 < \varphi < 2$, the distribution is a sum of N random variables that are gamma distributed while N is Poisson distributed, i.e. the distribution can be related to compound Poisson processes. For $\varphi \to 1$, the distribution tends to a Poisson distribution while for $\varphi \to 2$, the distribution tends to a gamma distribution. In summary, it can be used as a continuous distribution on the positive real line with potential positive probability at zero. Additionally, the Tweedie distribution is implemented in the `mgcv` package for parameter values $1 < \varphi < 2$. Therein, it allows for the estimation of the additional parameter φ via a REML approach as explained in Wood (2017, chp. 6.6) which makes the distribution flexible while still being computational feasible. Hence, we assume for the graph attributes in the total space

$$y_{i,a} | t_i, c_i, m_i \sim \mathrm{Tw}(\mu_{i,a}, \phi_a, \varphi_a), \quad a \in \mathcal{J}, i = 1, \dots, n$$

where Tw denotes the Tweedie distribution.

Although the Tweedie distribution might be a good fit for the data at hand, it can be computational expensive to estimate the additional parameter φ and we should compare the

model to other approaches. As described in Wood (2017, chp. 3.1.9, 6.6, 7.5), possible alternatives would be the negative binomial distribution or using a quasi-likelihood approach. Thus, we implemented in addition to the Tweedie distribution also single target GAMs assuming negative binomial distributed graph attributes in the total space. It is not straightforward to compare the results. To obtain a practical approach, we compare pairwise the AIC values of all fitted single target GAMs for Tweedie and negative binomial distributions in the last iteration of the AAC algorithm 3.

Finally, for each $a \in \mathcal{J}, i = 1, \ldots, n$ we assume the single target models:

$$g(\mu_{i,a}) = h_a(x_i) = f_1(t_i, c_i) + f_2(m_i) \tag{5.4}$$

where f_1, f_2 are smooth functions of the covariates, the covariates t_i, c_i, m_i are defined as in (5.2) and g is the link function of the mean as in subsection 3.4. In particular, f_1 is implemented with `mgcv` using tensor product smooths `te` with cubic spline marginal bases and basis dimensions 25 and five for the basis representations of time and Covid-19 cases, respectively. Furthermore, f_2 is implemented as cyclic cubic spline `cc` with 26 knots, where the knots for January are matched following Wood (2017, appendix C, exercise 5d)). Cyclic cubic splines are similar to cubic regression splines presented in subsection 3.3.2 and a more detailed introduction can be found e.g. in Wood (2017, chp. 5.3.2).

With these single target models, we wanted to analyse the effect of Covid-19 cases on the air passenger network while allowing for a variation in this effect over time. This led to the assumptions on f_1. Besides, the network varies over the months and thus, we included also f_2. Moreover, we tried to employ the considerations in Wood (2017) for the construction, especially of chapter 7. In particular, the `gam.check` function was used during the construction on different single target models to evaluate if we needed potentially larger basis dimensions which is recommended for values k close to one in the output of the function. Nevertheless, we note that the data could be modeled differently. One alternative would be to consider e.g. lagged Covid-19 cases similar to Wood (2017, chp. 7.4). Furthermore, time and Covid-19 cases do not necessarily have to be modeled using tensor product smooths.

5.3. Results

This subsection describes and interprets the results. The results are rather dissatisfying and reveal challenges for future research. Furthermore, we note that it is relatively difficult to present the results due to the complexity of the analysed object. Therefore, we mention first unexpected outcomes and corresponding potential reasons. Nevertheless, we present afterwards visualisations of the fitted and predicted whole networks as well as the graph attributes to give ideas for a first, mainly visual analysis of the results.

First, there was no permutation of nodes when we applied the model corresponding to (5.4) to the data. Nevertheless, algorithm 3 needed seven iterations till convergence. The regression errors in each iteration vary only slightly and are all similar to the regression error of the model without alignment in the total space. We guess that this is due to the inexact graph matching of

the implemented GAS algorithm and the imprecise mean estimates from the single target GAMs. However, this could be also arising due to a more severe issue within the algorithm 3. We tried different graph matching algorithms from the `graphspace` package to improve on this. However, the GA algorithm performed even worse and increased the regression errors enormously for the first few iterations while the GAS1 algorithm performed similarly to the GAS algorithm.

Second, our model in (5.4) is relatively flexible compared to e.g. a simple linear model with the same covariates. This means that the fit of each single target model is close to the graph attributes of the representatives in the current alignment. Therefore, two countries are only interchanged if they are particularly similar w.r.t. the passengers in their connections. Contrary, for a simple linear model with the same covariates, we aligned ten out of 139 graphs after five iterations. However, the regression error increased over the iterations for the simple linear model. This might be again due to the inexact graph matching or a more severe issue in the implementation of algorithm 1. Apart from this, we exemplify the difference in alignment between models of various flexibility further in appendix B.

Third, we removed the initial alignment in algorithm 3 of all to one of the observations. This was done since an alignment of all observations to one increased the regression errors considerably in all modeling approaches. One explanation for this is that the observed graphs differ strongly over time. An initial alignment of the graph from the first week with one during the first wave of Covid-19 differs strongly from an alignment of the graph of the first week w.r.t. the regression function evaluated at the corresponding covariate vector. Therefore, we recommend to omit the initial alignment of the graphs w.r.t. one observed graph in similar applications.

Finally, the passengers for the countries differ too much. Therefore, each country has already such a unique role in the network that it cannot be interchanged with other countries in a reasonable flexible model. This would possibly change if we include the populations of the countries in the graph attributes as described at the end of subsection 5.1. Furthermore, when we analysed the network of the ten countries with the least passengers between 2019 and 2021 in the EU with the same model, some of the countries were permuted. A possible reason for this might be that the approximated passengers within and between these ten countries are closer to zero and more similar. However, the regression error still increased in some iterations and we proceeded with the analysis of the ten countries with the most passengers. After these preliminary considerations, we offer in the following possibilities to evaluate the results. Since there occurred no alignments during the model fit, we can interpret the outcomes also by assigning the nodes the country labels from the total space. In general, this is not possible.

In a first step, we visualise the fitted and predicted networks of the model in figure 5.2 and figure 5.3. We obtained the plots as follows: first, to compute the coordinates of the nodes for each network, we applied the Kamada-Kawai layout implemented in Rudiger et al. (2022) obtained from the algorithm presented in Kamada & Kawai (1989). Furthermore, we centered the positions around the coordinates $(0, 0)$. In general, the positions of the nodes are arbitrary and one can employ alternatives such as a circular, spectral or spiral layout. Apart from this, the nodes and edges are scaled equally for all networks including the four predicted. Moreover, the dates about each network denote the first day of the week corresponding to the plot.

Twelve fitted networks can be found in figure 5.2. We plotted at least three networks of

the years 2019, 2020 and 2021. The networks corresponding to the first week in February and the second week in June are plotted for each year. In the first row, we can observe that the overall network structure of the nodes with the most domestic passengers in the Kamada-Kawai layout is visually relatively stable during 2019. In all three plots, there are four large nodes encompassing the networks. In the total space, these four nodes correspond to Spain, France, Italy and Germany. Spain has the most approximated fitted domestic passengers and corresponds to the larges node, for France and Italy a similar amount of domestic passengers can be observed and the two countries correspond to the two middle sized nodes and Germany has the least of all four and thus, corresponds to the node with the smallest size out of the four. In the second row, the network starts similarly to 2019 in February 2020. Afterwards, the network rapidly shrinks during March w.r.t. node and edge weights. However, the structure of the four nodes encompassing the network in the given layout remains similar till September 2020. Then, we can observe in the third row of figure 5.2 that edge and node sizes increase again although now, five nodes surround the inside. Finally, in the last row we see again a rather unusual plot in June. Contrary, the network recovered a similar structure as in 2019, given the Kamada-Kawai layout.

Overall, the plots during 2020 indicate that the air transportation network of the ten countries recovered rather slowly from the initial Covid-19 wave in the ten countries in spring 2020. The first network potentially similar in edge and node size to the ones in 2019 can be observed in August 2021. However, this visual analysis should be interpreted carefully since for example the coordinates for the layout are computed for each network separately.

In a next step, we analyse the predicted networks corresponding to the four covariate vectors at the end of the time series. The corresponding raw passenger networks of these weeks can be found on the left in figure 5.3 together with the predicted networks of the model on the right. It is difficult to interpret the predictions compared to the observed networks like this, although is seems that the structure is not well predicted. In contrast to this, the sizes of the three largest nodes seem reasonably predicted for the first and maybe second week. After week two, the sizes of the three largest nodes in the predictions increase fast compared to the observed graphs. Nevertheless, such a visual analysis and interpretation seems problematic, especially with the different positions of the nodes obtained from the layout. A potential solution would be to fix the nodes at some positions. This could be done e.g. by mapping them to their geographic locations. However, this is in general not appropriate for an analysis in the graph space, since the nodes lose their geographic allocation when we permute the labels.

At this point, there exist mainly two different approaches to proceed with the interpretation as described already at the end of subsection 4.1. On the one hand, network summary statistics might be helpful to analyse the results in more depth. Depending on the summary statistics, this enables an interpretation treating the model as a model on the whole network. On the other hand, we can investigate the networks further on graph attribute level. The advantage is that, compared to the network summary statistics interpretation, a graph attribute approach relates closer to the functional assumptions of the single target models (5.4). The disadvantage is that there are k^2 graph attributes where k is the number of nodes. In particular, we have in our application 100 possible connections for each network and thus, 100 potential single target models to interpret. However, since we did not observe alignments during the model fit, the

single target models can be related directly to the countries in our specific case. In the following, we present one possibility to visualise the fitted and predicted single target functions, i.e. the results on graph attribute level in the graph space.

We visualise the predictions of specific graph attributes compared to their observed values. Usually, this has to be done by permuting the observed values such that they correspond to the graph attributes of the aligned graphs used to fit the model. Contrary, we use solely the raw passenger approximations from subsection 5.1 for the corresponding connections within and between the ten countries since there happened no alignment during the model fit. In figure 5.4, we present exemplarily the graph attributes corresponding to the connections between Germany and Greece, Germany and Portugal, Portugal and Sweden, Portugal and the Netherlands as well as the domestic passengers for Germany and Portugal. The title of each subplot denotes the connection the plot corresponds to. The horizontal axis is shared over all plots and shows the time while the tick labels are given by their month and year. The vertical axis displays the passengers. The ticks are printed at the five quantiles $0, 0.25, 0.5, 0.75, 1$ of the fitted single target models evaluated at the covariate values used for fitting. The points display the observed passengers, the dark lines the fitted regression function and the purple line at the right of each subplot shows the predicted regression function for the four excluded covariate vectors. Furthermore, analogous visualisations can be found in the figures E.8, E.9 and E.10 in the appendix to present the variety in the graph attributes.

We can observe in figure 5.4 three single target models related to connections with Germany on the left and three related to connections with Portugal on the right. Overall, the fitted passengers lie between zero for the connection between Portugal and Sweden and 73125 for the domestic passengers in Germany. Consistently over all connections, the fitted functions as well as the points show the clear decrease in passengers during February, March and April 2020. Similar although less consistent and fast, such a decrease can be observed starting around October 2020. This clearly displays the first two Covid-19 waves in Europe. Aside from that, the connections recover differently from the first wave. For the connections between Germany and Greece, Germany and Portugal and Portugal and the Netherlands the tourist season in the summer in 2020 is apparent and the passengers achieve roughly levels half as large as during the same time in 2019. For Germany and Greece, the passengers recovered in June 2021 even to the level before the Covid-19 pandemic. A similar trend can be seen for the domestic passengers in Portugal. Contrary, the domestic passengers in Germany recovered very slowly although this connection recorded the largest numbers of passengers compared to all six connections. Apart from this, the fitted regression function seems reasonably smooth for e.g. the connection between Germany and Portugal. Contrary, the fitted regression function for e.g. the domestic passengers in Germany seems too wiggly.

We present more visualisations in the appendix E. We can visualise e.g. also the tensor product smooths of the week together with the respective Covid-19 cases for each single target model. This is exemplified in figure E.11 for the connections between Germany and Greece and within Germany on the left as well as between Portugal and the Netherlands and within Portugal on the right. Besides, examples for network summary statistics can be found in figure E.12 and figure E.13.

In summary, this section presented the application of the derived GAM framework. Although the results are expandable, we could observe many important obstacles and challenges of the model that could be discussed in future research.

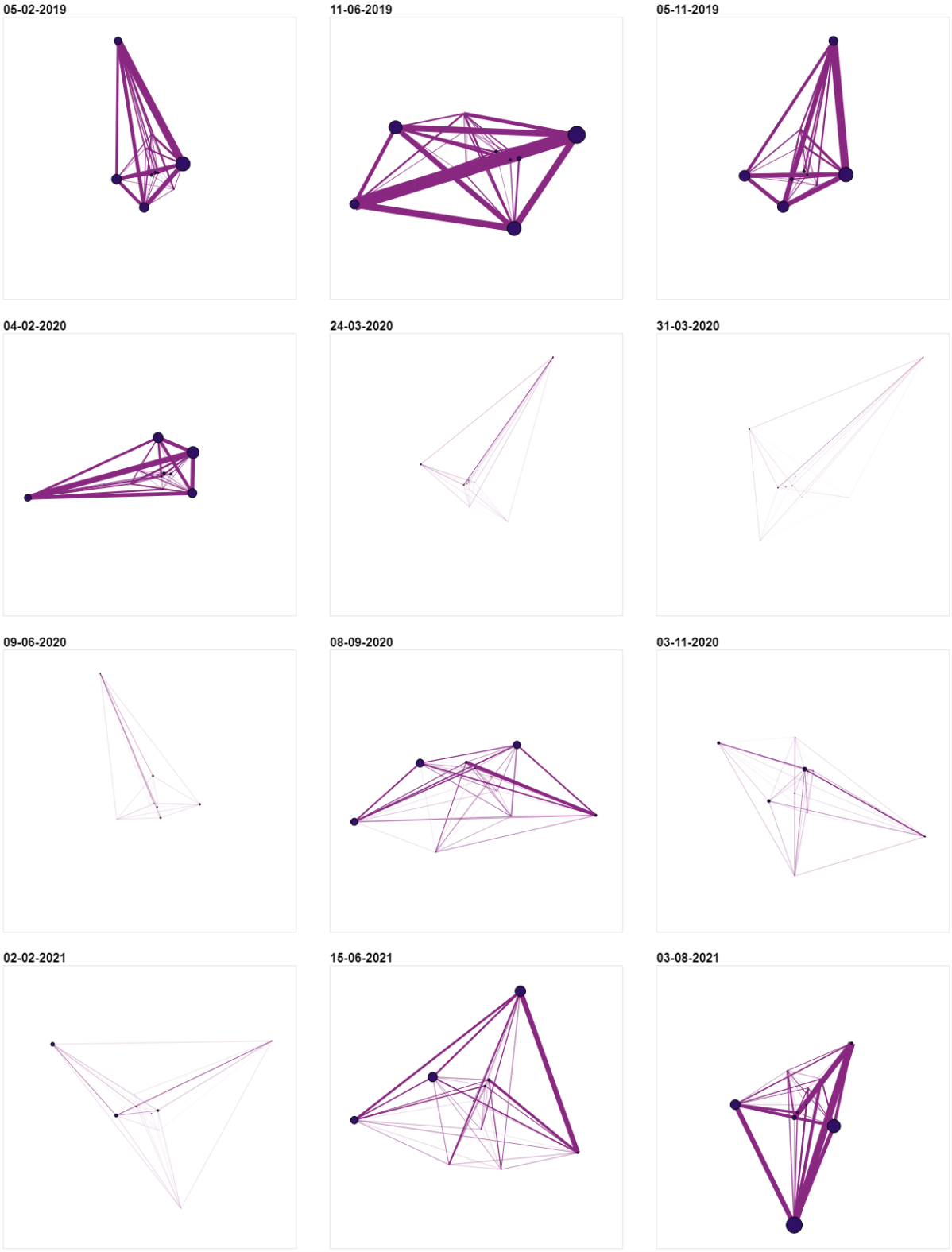

Figure 5.2.: Fitted networks of model (5.4) for selected covariate vectors. Nodes correspond to domestic passengers of nodes in the graph space, which is equivalent in our model to domestic passengers of the countries in the total space. Edges correspond to passengers between the nodes. The node and edge size is scaled by the passengers, nodes are colored darker and edged brighter. The date in the title denotes the date of the covariate vector corresponding to the plot.

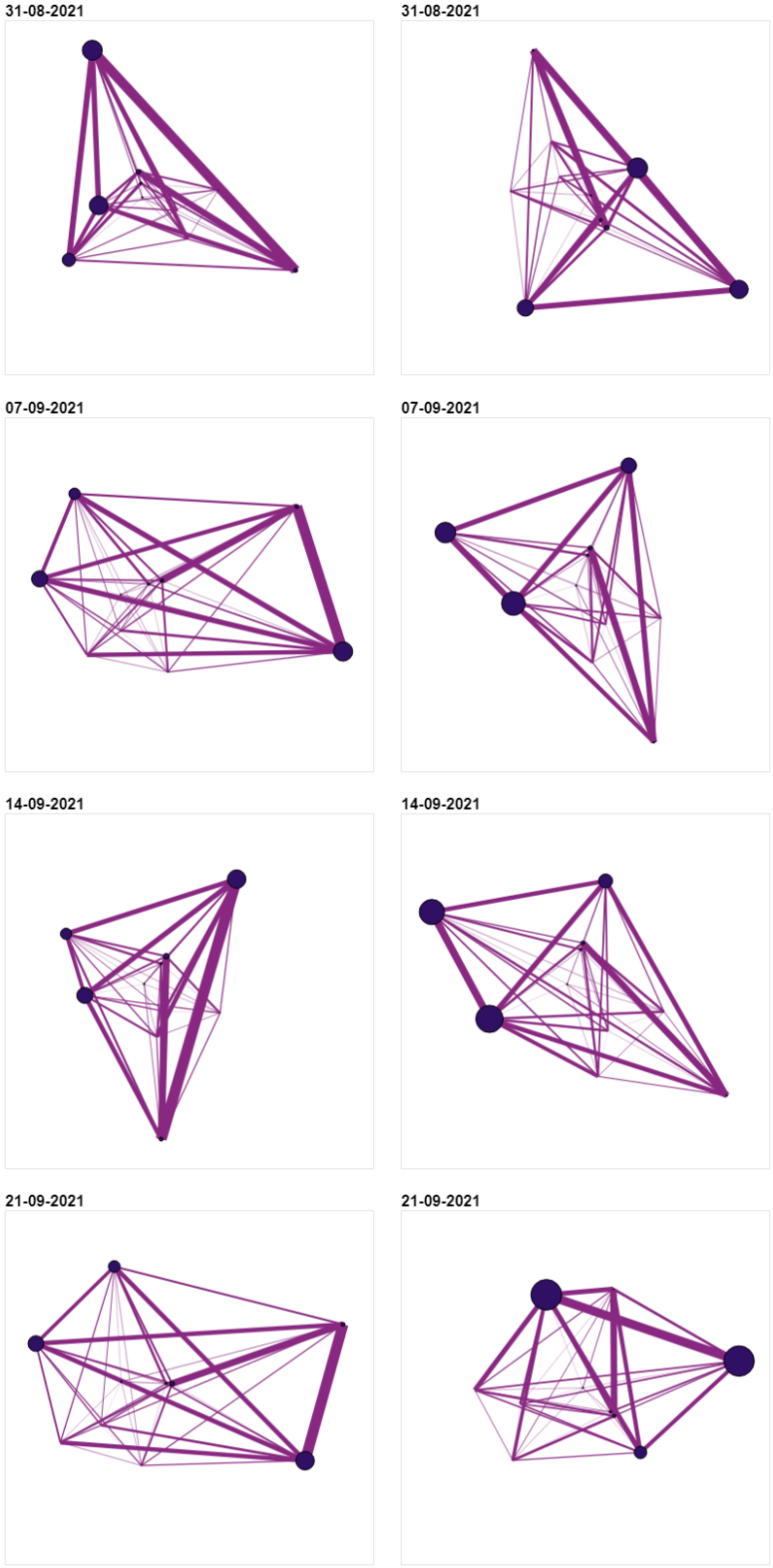

Figure 5.3.: Raw (left) and predicted (right) networks of model (5.4) for the four covariate vectors excluded for the fit (test data). Nodes, edges and titles have the same meaning as in figure 5.2.

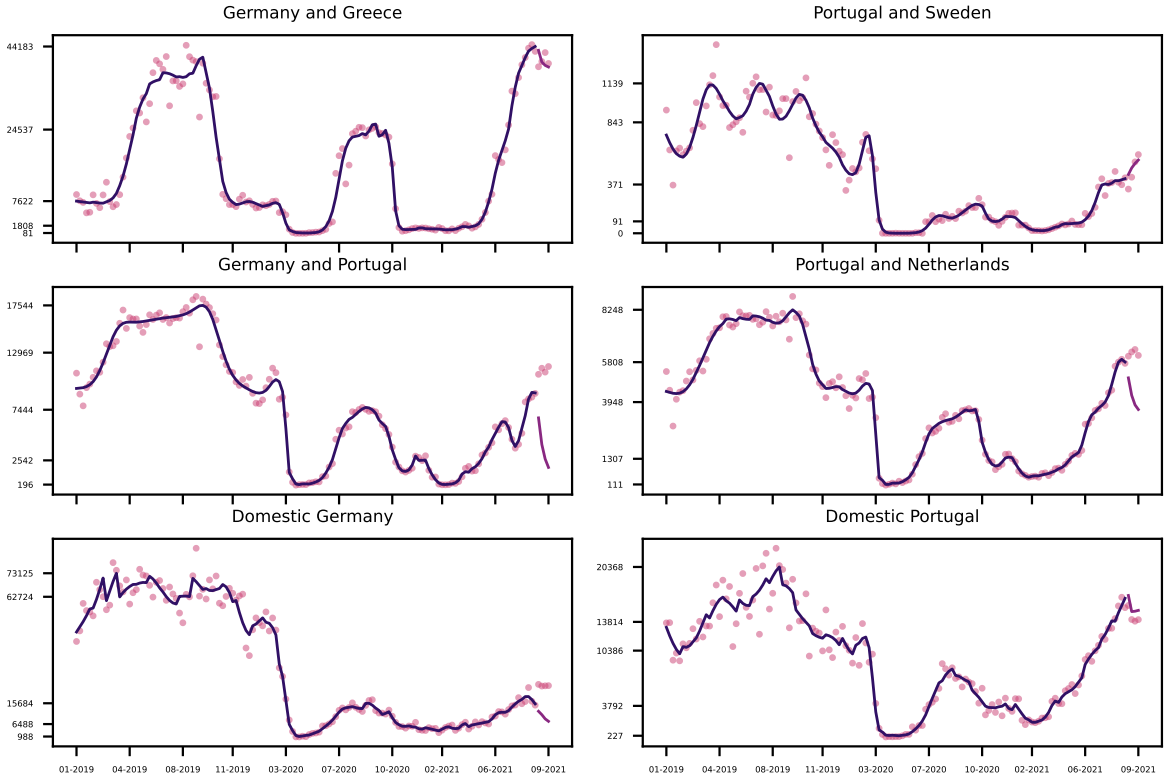

Figure 5.4.: Raw passengers (points), fit (line, dark violet) and predictions (line, purple) of selected single target models. The horizontal axis shows the time corresponding to the covariate vectors, is labeled by months and years at the ticks and is shared over all plots. The vertical axis represents the number of passengers and the ticks are printed at the five quantiles $0, 0.25, 0.5, 0.75, 1$ of the fitted regression function evaluated at the covariate values used for fitting. The titles show the depicted graph attributes. Since there are no permutations, the titles can refer directly to countries.

6. Conclusion

The methodology oriented part of this master thesis discussed the possibility to extend the linear regression framework of Calissano et al. (2022) to additive and generalized additive models. We started for both extensions from additive and generalized additive models for one-dimensional graph attributes and attained their corresponding objectives for optimisation in the graph space. In a first part, the translation to additive models led to a PLS objective in the graph space. Especially through the penalty term, the perspective on the parameters as graphs in the total as well as the graph space was apparent and therewith, a debate on the meaning of the parameters and thus, the functions arises naturally. In contrast, the extension to GAMs revealed difficulties in the transfer of single target models to the graph space connected to the support of the graph space as well as to differences between probabilistic and geometric regression approaches. Although this derivation of GAMs was naive due to the intrinsic probabilistic nature of the single target models and the absence of related probabilistic terminology in the graph space, the current exemplary applications of the regression frameworks in section 5 and in Calissano et al. (2022) demonstrate the necessity for such an extension. In summary, the highlights of this thesis from the methodology oriented perspective are the subsections 3.1.1, 3.4, 4.1 and 4.2 as well as appendix B.

The application oriented part applied the generalized additive regression framework to an air transportation network of parts of the EU during the Covid-19 pandemic. Therefore, we implemented single target REML based GAMs in the `graphspace` package utilizing the existing `mgcv` package and parallised the graph matching as well as the computation of the GAMs. The outcomes revealed potential challenges for the derived regression frameworks in future research.

Overall, merely first ideas of the regression functions could be obtained. Although the linear regression framework minimizes a reasonable LS objective, the family of functions we minimize over should be investigated more carefully. It is not straightforward to see how linear and, even more, smooth functions in the total space behave in the graph space. However, a first idea is developed in appendix B. Alternatively, an opportunity to investigate this further would be to compare the developed frameworks theoretically and empirically to similar methods such as Fréchet regression developed in Petersen & Müller (2019). Moreover, a more solid foundation for a framework related to GAMs would be obtained by a similar development of distributions on the graph space as described in Feragen et al. (2019, page 149) for manifolds. One such approach can be found e.g. in Paton et al. (2022). In this context, a translation of the framework to the literature on exchangeability in network models as in e.g. Crane (2018); Lauritzen et al. (2019) might be helpful to develop inferential results.

Apart from this, interpretation and evaluation of the results have to be investigated in more depth. In particular, the interpretation of the model w.r.t. to the data object that is analysed should be refined instead of an interpretation w.r.t. the graph in the total space. Therefore, the

interpretation could be split into interpretations on graph attribute level, conserving the implied functional assumptions of the single target models in the explanations, and interpretations on unlabeled graph level utilizing e.g. network summary statistics, emphasizing the viewpoint of the data object as whole unlabeled graph. Additionally, it is critical to investigate appropriate real world examples and develop simulation studies.

Ultimately, statistics for the graph space has only taken its first steps. There exist plenty other interesting future research questions for the regression frameworks alone. For example, the regression framework could be extended to include time series models accounting more sophisticated for networks observed over time. Moreover, the single target additive models could be extended further to functional additive models as introduced in Müller & Yao (2008). In contrast to this, the multi output model in the total space modeling each single target model separately could be replaced by a more sophisticated multi output model accounting for dependencies between the outputs as e.g. reviewed in Borchani et al. (2015). Alternatively, the framework could be extended w.r.t. to the graph. This could be done utilizing results from e.g. Crane (2018); Jain & Obermayer (2009); Jain (2016a) to allow for more complicated graphs such as bipartite, mixed or hypergraph structures. Similarly, the extension to other permutation invariances such as edge exchangeability are conceivable. Furthermore, the framework could be extended to graph-on-graph regression.

References

Beiler, K. J., Simard, S. W., & Durall, D. M. (2015). Topology of tree–mycorrhizal fungus interaction networks in xeric and mesic douglas-fir forests. *Journal of Ecology*, *103*(3), 616–628.

Bickel, P. J., & Doksum, K. A. (2015). *Mathematical statistics: basic ideas and selected topics, volumes i-ii package*. Chapman and Hall/CRC.

Bonacich, P. (1987). Power and centrality: A family of measures. *American journal of sociology*, *92*(5), 1170–1182.

Borchani, H., Varando, G., Bielza, C., & Larranaga, P. (2015). A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *5*(5), 216–233.

Breslow, N. E., & Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American statistical Association*, *88*(421), 9–25.

Bridson, M. R., & Haefliger, A. (2013). *Metric spaces of non-positive curvature* (Vol. 319). Springer Science & Business Media.

Calissano, A. (2020). *Graphspace package.* `https://github.com/annacalissano/GraphSpace`. GitHub.

Calissano, A., Feragen, A., & Vantini, S. (2020). Populations of unlabeled networks: Graph space geometry and geodesic principal components. *MOX Report*.

Calissano, A., Feragen, A., & Vantini, S. (2022). Graph-valued regression: Prediction of unlabelled networks in a non-euclidean graph space. *Journal of Multivariate Analysis*, *190*, 104950.

Council regulation (eec) no 95/93 on common rules for the allocation of slots at community airports. (2020-03-31). *OJ*, *L 99*, 1–4.

Crane, H. (2018). *Probabilistic foundations of statistical network analysis*. CRC Press.

De Boor, C., & De Boor, C. (1978). *A practical guide to splines* (Vol. 27). springer-verlag New York.

Dong, E., Du, H., & Gardner, L. (2020). An interactive web-based dashboard to track covid-19 in real time. *The Lancet infectious diseases*, *20*(5), 533–534.

Eilers, P. H., & Marx, B. D. (1996). Flexible smoothing with b-splines and penalties. *Statistical science*, *11*(2), 89–121.

Eurostat. (2022a). *Flights and passengers between reporting countries.* Luxembourg: Eurostat. (Accessed: 03-31-2022, `https://ec.europa.eu/eurostat/databrowser/view/avia_paocc/default/table?lang=en`)

Eurostat. (2022b). *Glossary of country codes of member states of the eu.* Luxembourg: Eurostat. (Accessed: 03-31-2022, `https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Glossary:Country_codes`)

Eurostat. (2022c). *Yearly passenger by member states of the eu.* Luxembourg: Eurostat. (Accessed: 08-01-2022, `https://ec.europa.eu/eurostat/databrowser/view/AVIA_PAOC__custom_3212581/default/table?lang=en`)

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. (2013). *Regression: Models, methods and applications*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Feragen, A., Hotz, T., Huckemann, S., & Miller, E. (2019). Statistics for data with geometric structure. *Oberwolfach Reports*, *15*(1), 125–186.

Feragen, A., & Nye, T. (2020). Statistics on stratified spaces. In *Riemannian geometric statistics in medical image analysis* (pp. 299–342). Elsevier.

Fletcher, P. T. (2013). Geodesic regression and the theory of least squares on riemannian manifolds. *International journal of computer vision*, *105*(2), 171–185.

Fletcher, P. T. (2020). Statistics on manifolds. In *Riemannian geometric statistics in medical image analysis* (pp. 39–74). Elsevier.

Fréchet, M. (1948). Les éléments aléatoires de nature quelconque dans un espace distancié. In *Annales de l'institut henri poincaré* (Vol. 10, pp. 215–310).

Friedman, J. H., & Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American statistical Association*, *76*(376), 817–823.

Gautier, L., Belopolsky, A., Moreira, W., & Warnes, G. (2008). *rpy2 package.* `https://github.com/rpy2/rpy2`. GitHub.

Gilchrist, R., & Drinkwater, D. (2000). The use of the tweedie distribution in statistical modelling. In J. G. Bethlehem & P. G. M. van der Heijden (Eds.), *Compstat* (pp. 313–318). Heidelberg: Physica-Verlag HD.

Ginestet, C. E., Li, J., Balachandran, P., Rosenberg, S., & Kolaczyk, E. D. (2017). Hypothesis testing for network data in functional neuroimaging. *The Annals of Applied Statistics*, 725–750.

Gold, S., & Rangarajan, A. (1996). A graduated assignment algorithm for graph matching. *IEEE Transactions on pattern analysis and machine intelligence*, *18*(4), 377–388.

Gower, h. C. (1975). Generalized procrustes analysis. *Psychometrika*, *40*(1), 33–51.

Guo, X., Bal, A. B., Needham, T., & Srivastava, A. (2022). Statistical shape analysis of brain arterial networks (ban). *The Annals of Applied Statistics*, *16*(2), 1130–1150.

Guo, X., & Srivastava, A. (2020). Representations, metrics and statistics for shape analysis of elastic graphs. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops* (pp. 832–833).

Guo, X., Srivastava, A., & Sarkar, S. (2021). A quotient space formulation for generative statistical analysis of graphical data. *Journal of Mathematical Imaging and Vision*, 1–18.

Hastie, T., & Tibshirani, R. (1986). Generalized additive models. *Statistical Science*, *1*(3), 297–310.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). Statistical learning with sparsity. *Monographs on statistics and applied probability*, *143*, 143.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67.

Huckemann, S., Hotz, T., & Munk, A. (2010). Intrinsic shape analysis: Geodesic pca for riemannian manifolds modulo isometric lie group actions. *Statistica Sinica*, 1–58.

Jacob, L., Vert, J.-p., & Bach, F. (2008). Clustered multi-task learning: A convex formulation. *Advances in neural information processing systems*, *21*.

Jain, B. J. (2016a). On the geometry of graph spaces. *Discrete Applied Mathematics*, *214*, 126–144.

Jain, B. J. (2016b). Statistical graph space analysis. *Pattern Recognition*, *60*, 802–812.

Jain, B. J., & Obermayer, K. (2009). Structure spaces. *Journal of Machine Learning Research*, *10*(11).

Jain, B. J., & Obermayer, K. (2011). Extending bron kerbosch for solving the maximum weight clique problem. *arXiv preprint arXiv:1101.1266*.

Jain, B. J., & Obermayer, K. (2012). Learning in riemannian orbifolds. *arXiv preprint arXiv:1204.4294*.

Jørgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society: Series B (Methodological)*, *49*(2), 127–145.

Kamada, T., & Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information processing letters*, *31*(1), 7–15.

Kolaczyk, E. D., & Csárdi, G. (2020). *Statistical analysis of network data with r*. Springer Nature.

Kolaczyk, E. D., Lin, L., Rosenberg, S., Walters, J., & Xu, J. (2020). Averages of unlabeled networks: Geometric characterization and asymptotic behavior. *The Annals of Statistics*, *48*(1), 514–538.

Lauritzen, S. L., Rinaldo, A., & Sadeghi, K. (2019). On exchangeability in network models. *Journal of Algebraic Statistics*, *10*(1).

Marron, J. S., & Dryden, I. L. (2021). *Object oriented data analysis*. Chapman and Hall/CRC.

Miolane, N., Guigui, N., Brigant, A. L., Mathe, J., Hou, B., Thanwerdas, Y., ... Pennec, X. (2020). Geomstats: A python package for riemannian geometry in machine learning. *Journal of Machine Learning Research*, *21*(223), 1-9.

Müller, H.-G., & Yao, F. (2008). Functional additive models. *Journal of the American Statistical Association*, *103*(484), 1534–1544.

Nelder, J. A., & Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, *135*(3), 370–384.

Paton, J., Hartle, H., Stepanyants, J., van der Hoorn, P., & Krioukov, D. (2022). Entropy of labeled versus unlabeled networks. *arXiv preprint arXiv:2204.08508*.

Petersen, A., & Müller, H.-G. (2019). Fréchet regression for random objects with euclidean predictors. *The Annals of Statistics*, *47*(2), 691–719.

R Core Team. (2022). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria.

Ratcliffe, J. G., Axler, S., & Ribet, K. (1994). *Foundations of hyperbolic manifolds* (Vol. 149). Springer.

Rudiger, P., Stevens, J.-L., Bednar, J. A., Nijholt, B., Mease, J., Andrew, ... Davis, B. (2022). *holoviz/holoviews: Version 1.15.0.* `https://doi.org/10.5281/zenodo.6817988`. Zenodo.

Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., & Wilhelm, M. (2014). Bringing up opensky: A large-scale ads-b sensor network for research. In *Ipsn-14 proceedings of the 13th international symposium on information processing in sensor networks* (pp. 83–94).

Severn, K. E., Dryden, I. L., & Preston, S. P. (2021). Non-parametric regression for networks. *Stat*, *10*(1), e373.

Severn, K. E., Dryden, I. L., & Preston, S. P. (2022). Manifold valued data analysis of samples of networks, with applications in corpus linguistics. *The Annals of Applied Statistics*, *16*(1), 368–390.

Strohmeier, M., Olive, X., Lübbe, J., Schäfer, M., & Lenders, V. (2021). Crowdsourced air traffic data from the opensky network 2019–2020. *Earth System Science Data*, *13*(2), 357–366.

Sun, X., Wandelt, S., Zheng, C., & Zhang, A. (2021). Covid-19 pandemic and air transportation: Successfully navigating the paper hurricane. *Journal of Air Transport Management*, 102062.

Suzumura, T., Kanezashi, H., Dholakia, M., Ishii, E., Napagao, S. A., Pérez-Arnal, R., & Garcia-Gasulla, D. (2020). The impact of covid-19 on flight networks. In *2020 ieee international conference on big data (big data)* (pp. 2443–2452).

Tweedie, M. C. (1984). An index which distinguishes between some important exponential families. In *Statistics: Applications and new directions: Proc. indian statistical institute golden jubilee international conference* (Vol. 579, pp. 579–604).

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. Scotts Valley, CA: CreateSpace.

Wandelt, S., & Sun, X. (2015). Evolution of the international air transportation country network from 2002 to 2013. *Transportation Research Part E: Logistics and Transportation Review*, *82*, 55–78.

Wood, S. N. (2006). Low-rank scale-invariant tensor product smooths for generalized additive mixed models. *Biometrics*, *62*(4), 1025–1036.

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(1), 3–36.

Wood, S. N. (2017). *Generalized additive models: an introduction with r*. CRC press.

Zhou, Y., & Müller, H.-G. (2021). Network regression with graph laplacians. *arXiv preprint arXiv:2109.02981*.

Appendix

This section summarizes additional results. In the first part, we give examples on labeled and unlabeled graphs and their visualisations. Additionally, we examine a specific example of a FD and an ordinary graph on the boundary of the FD that is not singular. Afterwards, we study the functions allowed in the regression frameworks in the graph space. Then, we state three numeric algorithms that can be used to approximate the parameters of the regression models with one dimensional response that are described in subsection 3.3. Next, we give a more explicit idea of the proof of the theorem for the AAC algorithm for additive models. Finally, more visualisations for the outcomes of the application are presented.

A. Illustrating examples for the graph space

This subsection studies examples to understand the theoretical concepts from subsection 3.1. The figures of this section, namely A.1, A.2 and A.3, are created with the open source browser-based application from www.draw.io.

First, we describe a labeled graph with $|V| = 3$ and thus, $|E| = 9$. Additionally, let the attribute space be $\mathcal{A} = \mathbb{R}^9$. We assume the graph to be undirected and without loops. We note that the graph could be mathematically represented as lower diagonal matrix (because of the symmetric structure of the adjacency matrix in case of an undirected graph) with zeros on the diagonal which can be helpful w.r.t. the implementation due to lower computational costs. Nevertheless, we still work here with the whole adjacency matrix.

In figure A.1, we present an example of the adjacency matrix of a labeled graph with three nodes on the right and its visualisation on the left. The node labels of the graph determine the order of the adjacency matrix and are written inside the nodes in the visualisation. For example, the labels determine that the undirected edge between node 1 and node 2 has edge attribute one and the corresponding entries in the adjacency matrix are depicted in row one, column two as well as in row two, column one. In general, if the node labels are no numbers but e.g. countries, each country could be assigned additionally a number. We note that the positions of the nodes in the visualisation are arbitrary, at least as long as the node labels do not carry additional spatial information that we add to the plot as e.g. possible for the countries in section 5 in the total space. Then, a

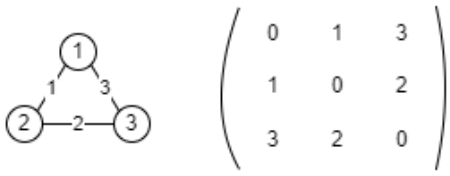

Figure A.1.: Visualisation and adjacency matrix of an undirected, labeled graph without loops. The entries in the nodes denote node labels while the entries on the edges denote edge weights.

permutation matrix $P \in \mathcal{P}$ that permutes node 2 and 3 is given by

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

Furthermore, the flattened adjacency matrix of the graph is given by $(0, 1, 3, 1, 0, 2, 3, 2, 0)$. The permutation $T \in \mathcal{T}$ of the flattened adjacency matrix that is equivalent to P interchanges the entries of node 2 and 3 and elements directed to them which results in $(0, 3, 1, 3, 0, 2, 1, 2, 0)$.

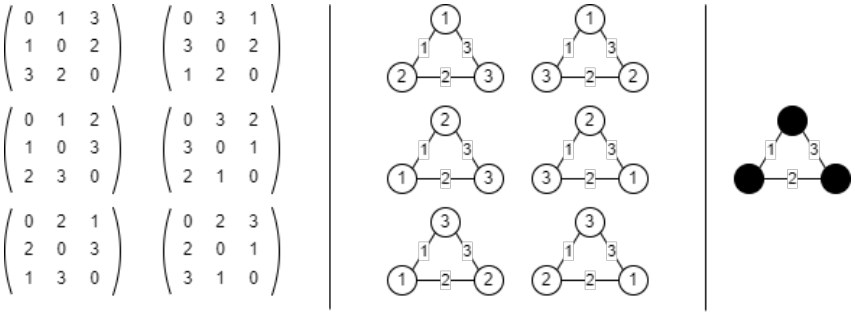

Figure A.2.: Adjacency matrices and visualisations of an undirected, unlabeled graph without loops. The entries in the nodes denote node labels while the entries on the edges denote edge weights.

Next, the corresponding unlabeled graph is illustrated in figure A.2. On the left, the adjacency matrices of the possible representatives of the unlabeled graph are displayed. The one in the left upper corner is the labeled graph studied above. The set of these six adjacency matrices could mathematically represent the unlabeled graph. In the middle, visualisations of these six possible representatives of the unlabeled graph are shown. Each representative can be obtained from another representative by permuting the nodes with a permutation matrix. For example, the second graph in the first row can be obtained from the first graph in the first row with the permutation matrix given in A.1. The unlabeled graph could be also visualised by removing the node labels which can be seen on the right in A.2. The left, middle and right part of figure A.2 correspond all to different representations of the unlabeled graph with three nodes and the depicted edge structure.

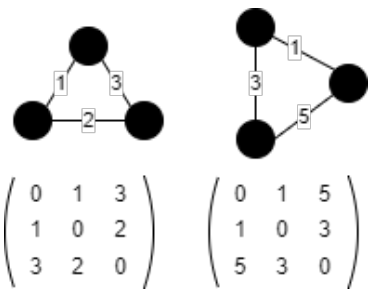

Figure A.3.: Two unlabeled graph visualisations with their optimally aligned adjacency matrices.

In a next step, we illustrate the metric in the graph space given in (3.1). Therefore, we depict two unlabeled graphs with three nodes in figure A.3. The positions of the nodes are different to emphasize that they are arbitrary in the visualisations. We assume that we use the Euclidean metric in the total space, such that larger differences between the entries are more penalized. Then, the entries are matched by size in this example as this would be e.g. also the case for graphs with node attributes but without edge attributes. Therefore, the adjacency matrices of the optimally aligned representatives of the labeled graphs are the ones displayed in the bottom row of figure A.3. Finally, the metric $d_{\mathfrak{G}}$ between two unlabeled graphs is computed as the sum of the pairwise distances of the entries of their aligned adjacency matrices.

Ordinary graphs and fundamental domains

First, we describe all ordinary graphs with two nodes, with node attributes but without edge attributes and their respective FDs. Although we assume that the graphs have no edge attributes, we represent them in the space $\mathcal{A} = \mathbb{R}^4$. Since we have only two nodes, $\mathcal{T}$ consists of two permutations. The non-identity permutation $T \in \mathcal{T}\backslash\{Id\}$ acts by switching the two nodes. Furthermore, as defined in subsection 3.1.1, an ordinary graph $Y = (Y_1, Y_2, Y_3, Y_4) \in \mathbb{R}^4$ is a graph with trivial stabilizer, i.e. $\{T \in \mathcal{T} | Y = TY\} = \{Id\}$. Thus, all graphs with different node attributes $Y_1 \neq Y_4$ are ordinary graphs since $Y = (Y_1, Y_2, Y_3, Y_4) \neq (Y_4, Y_3, Y_2, Y_1) = TY$, $T \in \mathcal{T}\backslash\{Id\}$, for example $Y = (Y_1, Y_2, Y_3, Y_4) = (3, 0, 0, 1)$. Contrary, all graphs with equal node attributes $Y_1 = Y_4$ are singular graphs since $Y = (Y_1, 0, 0, Y_1) = TY$, $T \in \mathcal{T}\backslash\{Id\}$. Besides this, the FD of an ordinary graph $Y \in \mathbb{R}^4$ with $Y_1 > Y_4$ is given by

$$D_Y = \{Z \in \mathbb{R}^4 | Z_1 \geq Z_4\}.$$

This can be shown using the definition for FDs in (3.3) and the result from Jain (2016a, proposition 4.18.7). To be precise, after the latter we obtain for general FDs:

$$D_Y = \{Z \in \mathcal{A} | Y^\top Z \geq Y^\top T Z \quad \text{for all} \quad T \in \mathcal{T}\} \tag{A.2}$$

and thus, in our example:

$$\begin{aligned} Y^\top Z = Y_1 Z_1 + Y_4 Z_4 &\geq Y_1 Z_4 + Y_4 Z_1 = Y^\top T Z \\ \iff Z_1(Y_1 - Y_4) &\geq Z_4(Y_1 - Y_4) \\ \iff Z_1 &\geq Z_4 \quad \text{for} \quad Y_1 > Y_4. \end{aligned}$$

Finally, the boundary and the interior are given by

$$\partial D_Y = \{Z \in \mathbb{R}^4 | Z_1 = Z_4\}, \quad D_Y^o = \{Z \in \mathbb{R}^4 | Z_1 > Z_4\}.$$

We note that the FDs become more complicated if we add edge attributes.

Next, we include edge attributes and examine an example for ordinary graphs that is on the boundary of FDs. This means, we assume that $\mathcal{A} = \mathbb{R}^4$ and give an example of two labeled graphs $Y_1, Y_2 \in \mathcal{A}$ that are ordinary graphs but $Y_1 \in \partial D_{Y_2}$ (and analogously $Y_2 \in \partial D_{Y_1}$). Let

$$Y_1 = (1, 1, -1, -1), \quad Y_2 = (1, -1, 1, -1).$$

We note that Y_1 and Y_2 are ordinary graphs since the only permutation except the identity permutation results in

$$TY_1 = (-1, -1, 1, 1), \quad TY_2 = (-1, 1, -1, 1)$$

and thus, $Y_1 \neq TY_1$ and $Y_2 \neq TY_2$ and hence, Y_1 and Y_2 have both a trivial stabilizer. Let now

$d^2_{\mathcal{A}}$ be the squared Euclidean norm. Then,

$$d^2_{\mathcal{A}}(Y_1, Y_2) = 2 = d^2_{\mathcal{A}}(Y_1, TY_2).$$

Thus, $Y_2 \in \partial D_{Y_1}$ (and $Y_1 \in \partial D_{Y_2}$) although Y_1 and Y_2 are ordinary graphs. As was shown in Kolaczyk et al. (2020, proposition 4.2), this can e.g. not happen if the graph attribute space is restricted to $\mathbb{R}_{\geq 0}$.

B. Informal analysis of the regression function

This section discusses the admissible regression functions f of the linear, additive and generalized additive regression models in the graph space described in section 3. Some of the following thoughts are the result of a brief talk with Lisa Steyer on the type of admissible regression functions.

As in the last section, we restrict first to the example of networks with two nodes, node attributes and without edge attributes, i.e. we set the edge attributes to zero. In particular, we assume $\mathcal{A} = \mathbb{R}^4$, $\mathfrak{G} = \mathcal{A}/\mathcal{T}$ and $J = 4, \mathcal{J} = \{1, 2, 3, 4\}$. First, we consider the linear regression model in the graph space corresponding to model (3.8) in the total space, restricted to an intercept and univariate covariate. This means, we study the functions $f : \mathbb{R} \to \mathfrak{G}, x \mapsto \pi \circ h(x)$ with $h : \mathbb{R} \to \mathcal{A}, x \mapsto (1, x)^\top \beta$, $x \in \mathbb{R}, \beta \in \mathbb{R}^2 \times \mathbb{R}^4$ and $h(x) = (h_a(x))_{a \in \mathcal{J}}, h_a : \mathbb{R} \to \mathbb{R}, x \mapsto (1, x)^\top \beta_a$, $\beta_a = (\beta_{0,a}, \beta_{1,a})^\top$. To be more precise, we assume for a sample $(x_i, [y_i]) \in \mathbb{R} \times \mathfrak{G}, i = 1, \ldots, n$ in matrix notation in the total space

$$\begin{bmatrix} y_{1,1} & 0 & 0 & y_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ y_{n,1} & 0 & 0 & y_{n,4} \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_{0,1} & \beta_{0,2} & \beta_{0,3} & \beta_{0,4} \\ \beta_{1,1} & \beta_{1,2} & \beta_{1,3} & \beta_{1,4} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,1} & \ldots & \varepsilon_{1,4} \\ \vdots & \ddots & \vdots \\ \varepsilon_{n,1} & \ldots & \varepsilon_{n,4} \end{bmatrix}$$

where all terms are described in detail in subsection 3.2. For this example, we are interested in $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)$. Due to the restriction to graphs without edge attributes, the parameters β_2 and β_3, which correspond to the edge attributes, are set to zero.

Next, we set $\beta_{0,1} = 3, \beta_{0,4} = 1$ and $\beta_{1,1} = 1, \beta_{1,4} = 2$. Then, we obtain e.g. the following predictions in the total space:

$$\begin{aligned} x = 0: &\quad h(0) = (3, 0, 0, 1) \\ x = 1: &\quad h(1) = (4, 0, 0, 3) \\ x = 2: &\quad h(2) = (5, 0, 0, 5) \\ x = 3: &\quad h(3) = (6, 0, 0, 7) \\ x = 4: &\quad h(4) = (7, 0, 0, 9) \end{aligned} \tag{B.3}$$

If we view the regression functions h evaluated at the covariate values as flattened adjacency matrices in the total space with node 1 and node 2, we see that for each unit increase in the covariate, node 1 increases by one and node 2 by two. This can be analogously derived w.r.t. the parameters as follows: As in subsection 4.2 and in contrast to the single target model perspective,

we treat the parameters as graphs. To denote this change in perspective, we use again the symbol tilde over the parameters, i.e. we are interested in

$$\tilde{\beta} = \begin{pmatrix} \tilde{\beta}_0 \\ \tilde{\beta}_1 \end{pmatrix}$$

with $\tilde{\beta} = \beta$. Then, our assumptions on the parameters lead to $\tilde{\beta}_0 = (3, 0, 0, 1)$ and $\tilde{\beta}_1 = (1, 0, 0, 2)$. Therefore, $\tilde{\beta}_0$ gives the intercept (graph) and $\tilde{\beta}_1$ the covariate effects of the model on the graph attributes in the total space.

So far, this was clear due to the separate single target linear regression functions $h_a(x), a \in \mathcal{J}$ on the graph attributes in the total space. Next, we study how this linearity translates into the graph space. The function f in the graph space is defined as projection of the function h in the total space. Although we mentioned in subsection 4.2 to treat the parameters $\tilde{\beta}_0, \tilde{\beta}_1$ as unlabeled graphs $[\tilde{\beta}_0], [\tilde{\beta}_1]$, this is not straightforward beyond the metric formulation in the penalty since addition in the graph space is not well defined and thus, for example $[\tilde{\beta}_0] + x[\tilde{\beta}_1]$ is not well defined. However, we can evaluate the function f e.g. at different covariate values to obtain unlabeled graphs at these covariate values. Then, we can analyse the change between two unlabeled graphs which is defined by the corresponding change in the total space. We could do this analogously as in (B.3) by

$$\begin{aligned} x = 0: &\quad f(0) = \{(3, 0, 0, 1), (1, 0, 0, 3)\} \\ x = 1: &\quad f(1) = \{(4, 0, 0, 3), (3, 0, 0, 4)\} \\ x = 2: &\quad f(2) = \{(5, 0, 0, 5), (5, 0, 0, 5)\} \\ x = 3: &\quad f(3) = \{(6, 0, 0, 7), (7, 0, 0, 6)\} \\ x = 4: &\quad f(4) = \{(7, 0, 0, 9), (9, 0, 0, 7)\}. \end{aligned} \tag{B.4}$$

Since unlabeled graphs are mathematically represented as sets of (flattened) adjacency matrices, this is rather difficult to interpret. Instead, we could look at the visualisations of the unlabeled graphs obtained from the regression function evaluated at $x = 0, 1, 2, 3, 4$. This is done in figure B.4. Therein, the plots are obtained analogously as in figure A.2 from the sets of flattened adjacency matrices in (B.4) which define the relative scaling for the size of the nodes. The network in the middle is encompassed by a red rectangle and we describe in the following the reason for this.

Figure B.4 shows the increase of the nodes arising due to the linear increase of the single target models in the covariates. One could be tempted to say that the upper node in the figure increases linearly by one unit while the lower node increases by two with each unit increase in the covariate. However, the positions of the nodes are arbitrary. For example, figure B.5 depicts the same graphs as in figure B.4 although the positions of the nodes of the graphs corresponding to $x = 3$ and $x = 4$ are interchanged. This emphasizes the important fact that in the graph space exist solely the information inherent in the graph attributes. Although this does not necessarily correspond e.g. to labels or spatial information on the nodes, the unlabeled graph contains relative information of the nodes w.r.t. each other which is constant over a set of adjacency

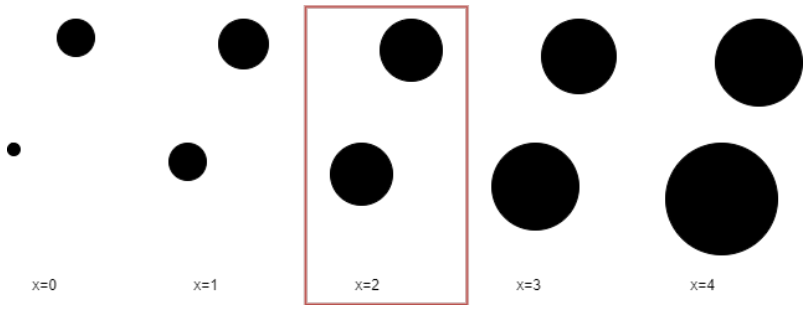

Figure B.4.: Linear regression function in the graph space evaluated at different covariate values for a graph with two nodes, node attributes and without edge attributes. The subtitles of the graphs denote the corresponding covariate values. The size of the node is scaled by its attribute. The red rectangle encompasses the singular graph.

matrices. For example, it is valid to say that the covariate effect on the larger node in figure B.4 and analogously in figure B.5 equals one for $x \leq 2$ and two for $x > 2$. Contrary, the covariate effect on the smaller node equals two for $x \leq 2$ and one for $x > 2$. In this example, the assignment of such 'artificial labels' (ALs) like 'larger' or 'smaller' on the unlabeled nodes in the graph space that define them uniquely w.r.t. all other nodes is easy since we can do this solely by the size of the nodes (because we set edge attributes and thus, permutations order solely by size). These ALs should define unlabeled nodes uniquely, invariant of the current alignment. Then, these ALs condense the FDs of one arbitrary ordinary graph and all its possible permutations into one space of the size of a FD that is never left by the regression function f. Inside of this space related to the ALs, the type of linearity in the graph attributes can change as we describe next. In summary, the ALs provide us the possibility to identify uniquely the graph attributes in the graph space. However, they are difficult to derive in general.

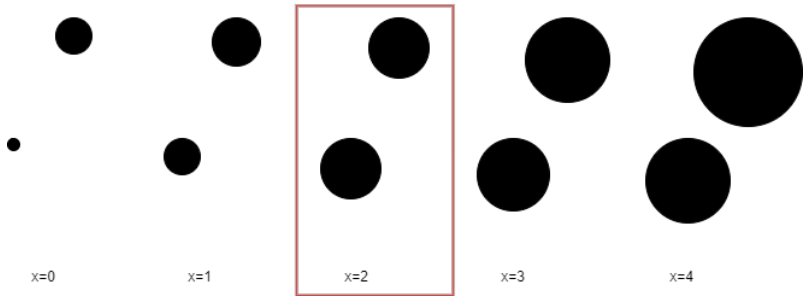

Figure B.5.: Linear regression function in the graph space evaluated at different covariate values for a graph with two nodes, node attributes and without edge attributes. The subtitles of the graphs denote the corresponding covariate value. The size of the node is scaled by its attribute. The red rectangle encompasses the singular graph. The nodes of the graphs corresponding to $x = 3$ and $x = 4$ are switched compared to figure B.4.

A bit more straightforward than the derivation of such ALs is the study of the type of change of the regression function f inside of the space of the ALs and thus, we start with this. First, we note that the covariate effects on graph attributes in the total space are fixed (fixed over the whole regression function) and linear due to the parameters $\tilde{\beta}$, here $\tilde{\beta}_0$ and $\tilde{\beta}_1$. However, the graph attributes in the graph space to which these fixed covariate effects correspond to can change. In fact, we can describe precisely when the fixed covariate effects switch between graph

attributes. This happens exactly when the regression function h in the total space crosses the boundary of the FD for itself a small step before. To be precise, as we derived in the last section, the FD of an ordinary graph $y \in \mathcal{A}$ with $y_1 > y_4$ is given by

$$D_y = \{z \in \mathcal{A} | z_1 \geq z_4\}.$$

which is therefore also the FD of $D_{h(x)}$ for $x < 2$ in our example. Let us fix now the FD at $D_{h(1)}$. For $x = 2$, we are on the boundary of the FD since $h(2) = (5, 0, 0, 5)$ is a singular graph, i.e. $h(2) \in \partial D_{h(1)}$. When we increase x further, we are inside the FD $D_{Th(1)}$, e.g. $h(3) \in D_{Th(1)}$. This is visualised by projecting it into a two dimensional coordinate system in the left plot in figure B.6, which is possible since we set the edge attributes to zero. Therein, the horizontal axis shows the attribute of node 1 and the vertical axis the attribute of node 2. The dashed line is the boundary of the FD and the purple line and rose points are the regression function h as well as h evaluated at $x = -1, 0, 1, 2, 3, 4$, all projected into two dimensions. We observe that $h(2) \in \partial D_{h(1)}$ and that h crosses the boundary of the FD at $x = 2$. The linearity in the labeled graph stays the same after crossing the boundary, i.e. the covariate effects on the graph attributes of node 1 and node 2 equal still one and two, respectively. This can be observed analogously in the left plot of figure B.7. Here, the graph attributes of node 1 and 2 are plotted against x and the dashed line marks the crossing of the boundary of the FD in the total space. The linearity in both regression functions stays constant.

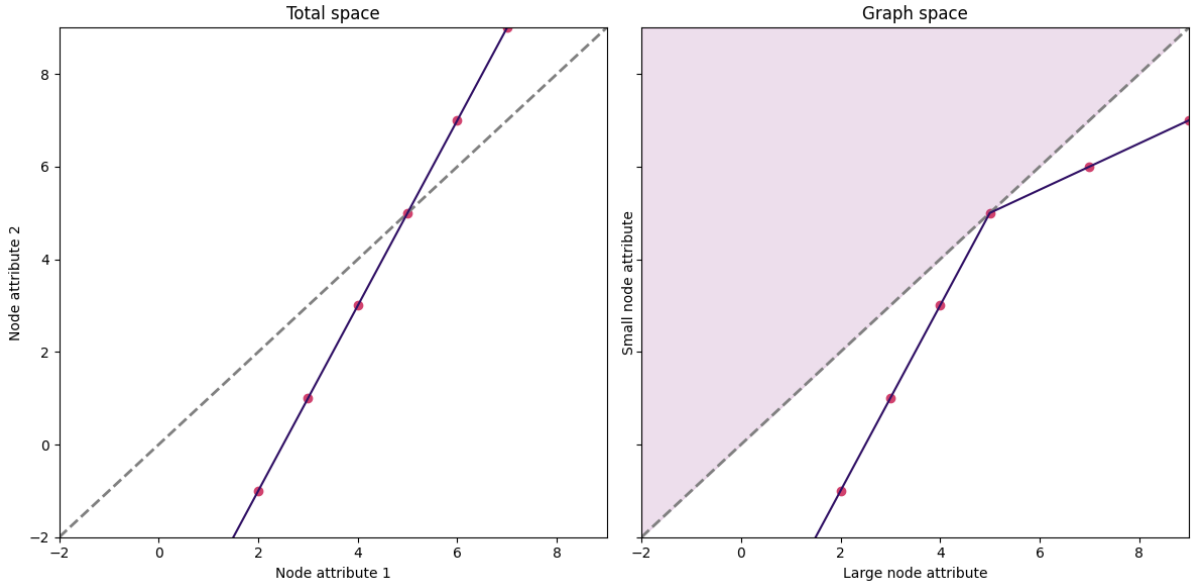

Figure B.6.: Regression functions of networks in total (left) and graph (right) space projected into two dimensions. Points denote the function evaluated at the covariate values $x = -1, 0, 1, 2, 3, 4$, from left to right. The dashed line denotes the boundaries of FDs in the total space.

In contrast to this, we can observe what happens to the regression function f in the graph space in the plots on the right in figure B.6 and B.7. In figure B.7, the purple function is the regression function for the large node and the rose function the one for the small node. As in the left plot, the dashed grey line marks the crossing of the boundary of the FD of the regression

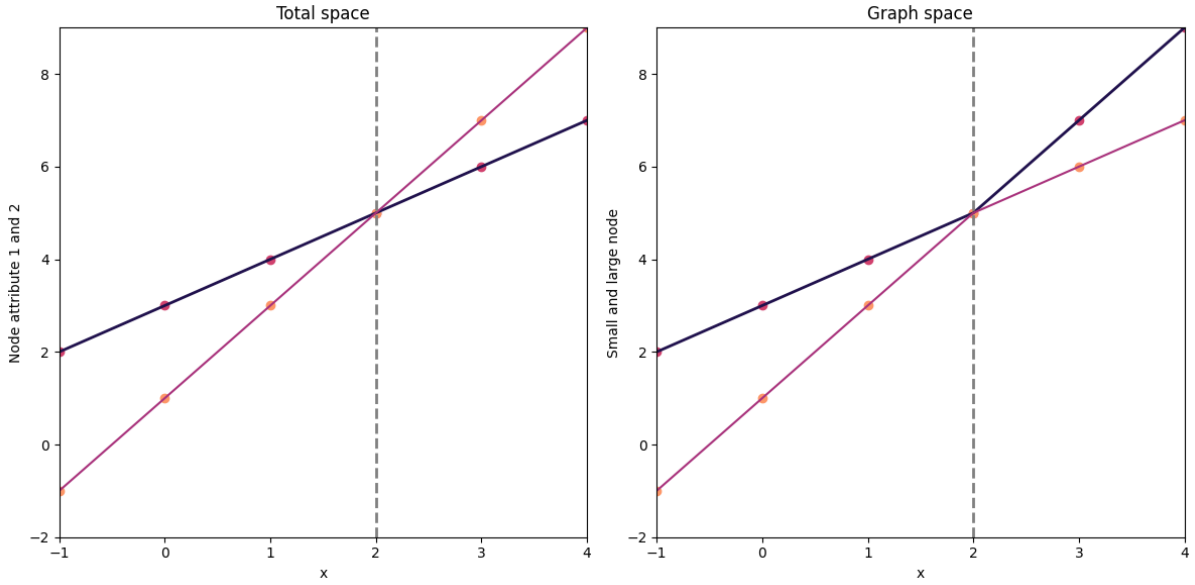

Figure B.7.: Regression functions of graph attributes in total and graph space. The left plot depicts node attribute 1 (purple, dark) and node attribute 2 (rose, bright) against x. The right plot shows the large node (purple, dark) and the small node (rose, bright) against x. Points denote the function evaluated at the covariate values $x = -1, 0, 1, 2, 3, 4$, from left to right. The dashed line marks the crossing of the boundary of the FD of the regression function h in the total space.

function h in the total space. After the crossing, the covariate effects on the unlabeled nodes interchange. In particular, the covariate effects on the graph attributes in the graph space are piecewise or locally linear where the pieces (or regions) depend on the regression function h in the total space and its crossing of boundaries of FDs. After the crossing, the covariate effects are not allowed to be arbitrary, but rather are interchanged between the graph attributes. This is visualised in the right plot of figure B.6. Therein, the attribute of the small node is plotted against the attribute of the large node. Due to the ALs 'small' and 'large', we constructed a space of the size of the FDs in the total space, that is not left by the regression function f, namely the right, white triangle and its extension outside the plot. However, the covariate effects on the graph attributes of the small and large node interchange at the dashed grey line. The left upper, rose triangle is obviously never reached by the function f by definition of small and large.

In summary, in case of the linear regression model, we allow for piecewise linear regression functions on the graph attributes in the graph space. The pieces depend on the crossing of boundaries of FDs of the regression function h in the total space. The covariate effects could be seen e.g. as fixed while interchanging the graph attributes in the graph space at crossings. Besides, we note that this result is independent of the knowledge of ALs for the graph attributes in the graph space and could be used thoughtfully e.g. on visualisations of unlabeled graphs along the regression function f. Furthermore, we emphasize that it allows us to utilize the interpretability of the parameters and single target models from the total space also in the graph space in an exactly defined way, as already briefly noted at the end of subsection 4.1 for the interpretation on graph attribute level in the graph space. This translates the interpretability obtained by using regression modeling to the graph space. Moreover, if we use at some point more sophisticated multi output models instead of the separate single target modeling, the

considerations can be potentially adapted straightforwardly.

These concepts on the form of the regression function f could be extended in various directions. First, they also hold for the additive models in the graph space derived in subsection 3.4. Therefore, instead of linear functions on the graph attributes in the total space, we allow for smooth functions. Still, if the corresponding function h crosses a FD in the total space, the smooth covariate effects on the graph attributes in the graph space are interchanged. We note again, that the smooth effects are solely interchanged while being fixed for the whole function h in the total space.

Second, we could study different types of networks. Therefore, we need to define their FDs. Afterwards, to apply the concepts discussed above, we have to analyse when the regression function h crosses a boundary of the FD. As a first example, we could include edge attributes and more nodes. Then, we use the result from Jain (2016a, proposition 4.18.7) as in the last subsection to define FDs. Thus, the general FD of an ordinary graph $Y \in \mathcal{A} = \mathbb{R}^J$ can be rewritten from (3.3) as

$$D_Y = \{Z \in \mathcal{A} | Y^\top Z \geq Y^\top T Z \quad \text{for all} \quad T \in \mathcal{T}\}.$$

Next, we want to derive a method to obtain the pieces at which we interchange the covariate effects. Therefore, we want to derive when the regression function $h(x)$ evaluated at x is on the boundary of the FD of itself a small step before, i.e. $h(x) \in D_{h(x-\epsilon)}, \epsilon > 0$. This is the case if for any permutation $T \in \mathcal{T} \backslash \{Id\}$

$$\begin{aligned} & \lim_{\epsilon \to 0} h(x-\epsilon)^\top h(x) = \lim_{\epsilon \to 0} h(x-\epsilon)^\top T h(x) \\ \iff \quad & \lim_{\epsilon \to 0} \sum_{a=1}^{J} h_a(x-\epsilon)(h_a(x) - t_a h_a(x)) = 0 \\ \iff \quad & \sum_{a=1}^{J} h_a(x)(h_a(x) - t_a h_a(x)) = 0 \end{aligned}$$

where t_a denotes the elementwise permutation of graph attribute a corresponding to T and the second equality follows from the continuity of $h_a, a \in \mathcal{J}$. Thus, we are interested in the functions

$$\zeta_T(x) = \sum_{a=1}^{J} h_a(x)(h_a(x) - t_a h_a(x)), \quad T \in \mathcal{T} \backslash \{Id\}$$

and in particular, their crossings with zero. This seems difficult to grasp at first due to the large number of possible permutations, namely $k!$ where k denotes the number of nodes. However, we could start with our regression function h at a particular graph, e.g. at the intercept $h(0)$. We assume that $h(0)$ is an ordinary graph, otherwise we use $h(0+\epsilon), \epsilon > 0$. Then, if we assume to cross the FD at a surface of the boundary instead of a corner (the FD is a covex polyhedral cone, cp. e.g. Jain (2016a, proposition 4.18.1)) which means that we allow just for pairwise permutations, we obtain for $h(0)$ a starting point with $\sum_{i=1}^{k}(k-i)$ possible permutations to investigate. For example for $k = 10$ as in section 5, we obtain 45 possibility permutations under the additional assumption. Then, we could e.g. plot each ζ_T as a function of x for each of the

45 considered permutations T. If one of the functions crosses zero, we know that we crossed a boundary of the FD and thus, we interchanged the covariate effects for the graph attributes in the graph space. We even know that the covariate effects of the graph attributes that are permuted in the total space are interchanged in the graph space. However, we neglected the corners of the FD to obtain a lower number of possible permutations. Including them would mean that we account also for non pairwise permutations of the nodes. In summary, the crossings of the functions ζ_T with zero provide us the covariate values, at which the covariate effects from the graph attributes in the total space interchange for the graph attributes in the graph space. There exist presumably more elegant possibilities than the computation of all $\zeta_T, T \in \mathcal{T}\backslash\{Id\}$. For example, one could in practice check if $h(x-c), c \in \mathbb{R}$, is still optimally aligned w.r.t. $h(x)$. If this is not the case, the covariate effects on the graph attributes in the graph space interchanged between $x-c$ and x. This could be done for different regions of the covariates to obtain approximations of the pieces.

Similar to the example above, we could restrict the graph attributes also to the positive real line, i.e. $\mathcal{A} = \mathbb{R}^J_{\geq 0}$. Then, we can derive the function ζ_T using the result on the form of the FDs in Kolaczyk et al. (2020, appendix B) again as:

$$\zeta_T(x) = \sum_{a=1}^{J} h_a(x)(h_a(x) - t_a h_a(x)), \quad T \in \mathcal{T}\backslash\{Id\}.$$

In general, this could be also helpful to extend the concepts to the GAMs in the graph space derived in 3.4 after defining appropriate FDs in graph spaces corresponding to total spaces where $\mathcal{A} \neq \mathbb{R}^J$.

Third, compared to the Fréchet mean in the graph space as derived e.g. in Calissano et al. (2020), the considerations reveal conceptual differences between Fréchet mean and regression function in the graph space. The Fréchet mean is defined as the network for which one fixed labeled representative minimizes the distance to all optimally aligned networks in its FD in the total space. Therefore, the fixed representative of the Fréchet mean lies in its specific FD in the total space. Contrary, for the regression function f, if we fix one representative h, this function is allowed to lie in multiple FDs. This controversy is also related to the distinction between the two definitions for geodesics given in subsection 3.1. The first definition projects lines, planes and hyperplanes, which could potentially cross boundaries of FDs in the total space, into the graph space. Contrary, in the second definition a geodesic between two unlabeled graphs is the projection of the shortest path between their two optimally aligned representatives. Therefore, the shortest path stays in the FD of the two graphs (assuming for simplicity that they are ordinary graphs, otherwise choose an ordinary graph whose boundary of the FD consists also of one or both of the graphs). Then, the second definition could be related to regression functions f in one of the two following ways: i) $f(x_i)$ and y_i lie in a common FD for all $i = 1, \ldots, n$ or ii) f is a piecewise geodesic between the crossings of h in the total space. Apart from this, i) can be related to interpretations on graph attribute level in the total space as described at the end of subsection 4.1 and used in Calissano et al. (2022) while ii) corresponds to interpretations in the graph space, utilizing the underlying regression function h in the total space.

Finally, the concepts help to explain at least intuitively that the results of the application

in section 5 indicate, that a permutation of the observed graphs in the AAC algorithms might be less probable for more flexible regression functions. Therefore, we change the perspective by adding observations to the described concepts, thinking about permuting these observations (instead of the regression function) and consider a flexible regression model that fits the data without error. Then, the representative h of the regression function f evaluated at the covariates of the data equals the representatives y_i^{cur} of the observed unlabeled graphs $[y_i]$ in the current iteration. Therefore, representatives of observed graphs would be just permuted if we allow for permutations with constant distance and if the representatives are singular graphs. The latter happens with probability zero, at least if we observe complete graphs. However, this is less clear if we obtain also some error in the fit of the flexible regression function to the data. Then, it seems natural that the representatives y_i^{cur} lie in the FDs of the flexible regression function h with a higher probability than in the FDs of the less flexible regression functions since the flexible single target functions are closer to the graph attributes of the representatives. Nevertheless, this might not be the case in general, especially e.g. around zero in $\mathbb{R}^J$ since the FDs are convex polyhedral cones as described e.g. in Jain (2016a); Kolaczyk et al. (2020) and thus, this could be investigated further.

In summary, these considerations try to describe explicitly the meaning of the projection of a linear or smooth regression function h into the graph space as well as the form of the obtained projected function f. We noticed that the parameters and thus, the covariate effects, are fixed in the total space. These fixed covariate effects are interchanged on the graph attributes in the graph space when $h(x)$ crosses a boundary of its FD $D_{h(x-\epsilon)}$. In particular, the covariate effects corresponding to the labeled nodes, which are permuted when crossing the boundary of the FD, are interchanged. This means that linear functions for the graph attributes in the total space imply piecewise linear functions for the graph attributes in the graph space, and analogously for smooth functions. The pieces depend on the the regression function h and could be e.g. computed analysing functions such as $\zeta_T, T \in \mathcal{T}\backslash\{Id\}$ or using more practical approaches. Furthermore, e.g. the linearity is fixed in the total space and not arbitrary on the pieces, but instead the fixed linearity from the total space is interchanged between pieces.

Finally, one could question further if such piecewise functions are reasonable as regression functions in the graph space. For example, would it be more appropriate to assume linearity in the covariate effects for the graph attributes in the graph space and could this be achieved? Or would a model with piecewise covariate effects for the graph attributes in the graph space, in which the covariate effects are allowed to differ arbitrarily, be more reasonable?

Before the discussion on the regression function f, we started with the presentation of ALs in our simplified example with two nodes. In the future, the aim would be to derive ALs in a general setting. We can e.g. derive pointwise, pairwise ALs for the regression function f with the help of the general form of FDs in (A.2). We exemplify this idea for the intercept $h(0) = (3, 1, 2, 1)$ and its permutation $Th(0) = (1, 2, 1, 3)$ at $x = 0$. Then, we obtain the AL for

arbitrary $y = (y_1, y_2, y_3, y_4) \in \mathcal{A}$ as

$$\begin{aligned}&\sum_{a=1}^{J} y_a(h_a(0) - t_a h_a(0)) = y_1(3-1) + y_2(1-2) + y_3(2-1) + y_4(1-3)\\ =&2y_1 - y_2 + y_3 - 2y_4,\end{aligned}$$

i.e. the AL would be that one node has a two times larger node attribute and a smaller edge attribute compared to the other node at $x = 0$, considering undirected graphs. This approach to ALs could be developed further in future research. Alternatively, we believed that compositional data analysis, some network summary statistics or line graphs could be helpful to construct ALs. However, the derivation of ALs was unfortunately out of the scope of this thesis.

C. Numeric algorithms for single target regression models

This section reviews briefly some numeric algorithms to approximate the parameters of the regression models of subsection 3.3. This is done solely for completeness from Fahrmeir et al. (2013) and can be found in related literature more detailed.

(Penalized) iterative (re-)weighted least squares

For GLMs, we can maximize the log-likelihood function (3.14) numerically with the iterative (re-)weighted least squares (IRLS) algorithm based on the Newton-Raphson method or similarly, the Fisher scoring algorithm. We describe the latter and denote in the following $\hat{\eta}_i^{(t)} = x_i\hat{\beta}^{(t)}$. Then, following Fahrmeir et al. (2013, chp. 5.8.2), in each iteration the IRLS estimator is

$$\hat{\beta}^{(t+1)} = (X^\top W^{(t)} X)^{-1} X^\top W^{(t)} \tilde{y}^{(t)}$$

where $\tilde{y}^{(t)} = (\tilde{y}_1(\hat{\beta}^{(t)}), \ldots, \tilde{y}_n(\hat{\beta}^{(t)}))^\top$ with $\tilde{y}_i(\hat{\beta}^{(t)}) = \hat{\eta}_i^{(t)} + (h_i'(\hat{\eta}_i^{(t)})^{(-1)}(y_i - \hat{\mu}_i(\hat{\beta}^{(t)}))$. Moreover, the weight matrix $W^{(t)} = (\tilde{w}_1^{(t)}, \ldots, \tilde{w}_n^{(t)})$ is defined with $\tilde{w}_i^{(t)} = (h'(\hat{\eta}_i^{(t)}))^2 \frac{w_i}{b''(\theta_i)\phi}$ where $b''(\theta_i)$ is the second derivative of $b(\theta_i)$ of the EF of y_i as well as w_i and ϕ are the respective parameters of the EF of y_i. Furthermore, h' is the derivative of the inverse link function h of the model (3.13).

In the context of GLMMs as in (3.15), we can maximize the penalized log-likelihood (3.16) as follows: let

$$X = (Z, U) \quad \text{and} \quad S = \begin{pmatrix} 0 & 0 \\ 0 & G^{-1} \end{pmatrix}.$$

Based on a Fisher scoring algorithm, we obtain a penalized IRLS (PIRLS) algorithm with the following estimators in each iteration:

$$\begin{pmatrix} \hat{\gamma}^{(t+1)} \\ \hat{b}^{(t+1)} \end{pmatrix} = (X^\top W^{(t)} X + S)^{-1} X^\top W^{(t)} \tilde{y}^{(t)}$$

where $\tilde{y}^{(t)}, W^{(t)}$ are defined as in the IRLS estimator for GLMs with $\eta_i = z_i^\top \gamma + u_i^\top b$.

Restricted maximum likelihood

Next, we describe the REML method from Fahrmeir et al. (2013, chp. 7.6) to estimate θ where θ are the unknown coefficients in the covariance matrix $G = G(\theta)$ for GLMMs (3.15). We denote for $i = 1, \ldots, n$

$$\beta^\top = (\gamma^\top, b^\top), \quad D = \operatorname{diag}\left(\ldots, \frac{\partial h(\eta_i)}{\partial \eta}, \ldots\right)$$
$$\Sigma = \operatorname{diag}(\ldots, \sigma_i^2, \ldots) \text{ with } \sigma_i^2 = \sigma_i^2(\eta_i) = \operatorname{Var}(y_i | x_i).$$

Then, we obtain for the working weights $W^{(t)}$ and the working observations $\tilde{y}^{(t)}$ (without iteration superscripts)

$$W = D\Sigma^{-1}D$$
$$\tilde{y} = X\hat{\beta} + D^{-1}(y - \mu) \quad \Longleftrightarrow \quad y - \mu = D(\tilde{y} - X\hat{\beta}).$$

Moreover, we can approximate for a GLMM the log-likelihood of β given θ by a Laplace approximation as

$$\begin{aligned}\ell(\beta|\theta) &\approx \frac{1}{2}(y-\mu)^\top \Sigma^{-1}(y-\mu) \\ &= (\tilde{y} - X\hat{\beta})^\top W (\tilde{y} - X\hat{\beta}).\end{aligned}$$

For fixed θ, the marginal distribution of the working observations is

$$\tilde{y} | X, \gamma \sim \mathcal{N}(X\beta, W^{-1})$$

and we obtain with $V(\theta) = UG(\theta)U^\top + W$ the so called approximate restricted log-likelihood

$$\ell_{\mathrm{R}}(\theta) = -\frac{1}{2}\log|V(\theta)| - \frac{1}{2}\log(|Z^\top V(\theta)^{-1} Z|) - \frac{1}{2}(\tilde{y}(\theta) - Z\hat{\gamma})^\top V(\theta)^{-1}(\tilde{y}(\theta) - Z\hat{\gamma})$$

which can be also maximized numerically using a Newton-Raphson or Fisher scoring method. The corresponding restricted log likelihood in case of LMMs leads, when maximized, to the REML estimator for the covariance matrix parameters θ.

Finally, β and θ can be numerically approximated by iterative computation of β using a PIRLS step given the current estimate of θ and one iteration of a Newton-Raphson or Fisher scoring algorithm to compute θ given the current estimate of β, until convergence. The algorithm is similarly summarized in Fahrmeir et al. (2013, chp. 7.6). Due to the possibility of a mixed model formulation of GAMs, this can be used to estimate the parameters including the smoothing coefficients in GAMs.

D. Proof idea for the AAC algorithm for additive models

First, we prove that algorithm 2 stops in finite time. The proof is analogous to the proof of convergence in finite time in Calissano et al. (2022). Moreover, we note that the idea and

structure of the proof is also used e.g. in Calissano et al. (2020) to prove a similar result for a principal component analysis in the graph space.

We denote by y_i^{cur} the current representative of $[y_i]$ and by h^{cur} the current estimate of the additive model obtained from the regression on $y_i^{cur-1}, i = 1, \dots, n$ in the current iteration of the AAC algorithm 2. Then, the algorithm consists of two parts. First, the additive model is estimated with the current representatives. If there was at least one representative that was permuted from $cur - 1$ to cur, we have

$$\begin{aligned} PLS_{\mathcal{A}}^{cur} &= \sum_{i=1}^{n} d_{\mathcal{A}}^2(y_i^{cur}, h^{cur}(X_i)) + \sum_{a=1}^{J}\sum_{j=1}^{p} \lambda_{j,a}^{cur} (b_{j,a}^{cur})^\top \mathcal{S}_j b_{j,a}^{cur} \\ &\leq \sum_{i=1}^{n} d_{\mathcal{A}}^2(y_i^{cur}, h^{cur-1}(X_i)) + \sum_{a=1}^{J}\sum_{j=1}^{p} \lambda_{j,a}^{cur-1} (b_{j,a}^{cur-1})^\top \mathcal{S}_j b_{j,a}^{cur-1} \end{aligned}$$

since otherwise the additive model of iteration $cur - 1$ would be estimated again. Thus, the estimation of the current additive model does not increase the current $PLS_{\mathcal{A}}^{cur}$. Second, the algorithm aligns all representatives to h^{cur} in each iteration. This lowers $PLS_{\mathcal{A}}^{cur}$ if there is at least one representative k that was realigned with permutation T_k since then

$$d_{\mathcal{A}}^2(T_k y_k, h^{cur}(X_k)) < d_{\mathcal{A}}^2(y_k, h^{cur}(X_k))$$

while the penalty term stays constant since it is independent of the permutation. Thus, this step also does not increase $PLS_{\mathcal{A}}^{cur}$.

Furthermore, if the prediction error stays the same in two iterations, the algorithm stops. Since the number of permutations for graphs with a bounded number of nodes is finite, we observe finitely many observations and we saw that each alignment is observed at most once and thus, the algorithm stops in finite time.

In a second step, we want to prove that the algorithm converges almost surely to a local minimum of (3.28). This could be done by adapting the proof of Calissano et al. (2022) to (3.28). However, we will follow only partly Calissano et al. (2020) and utilize for the rest of the proof the ideas developed for FDs. This leads to the same result but allows for an exact characterization w.r.t. FDs of the graphs that have probability zero in the proof of Calissano et al. (2022).

To prove the result, we have to show that there exists almost surely a $\epsilon > 0$ such that for $\|\beta - \tilde{\beta}\| < \epsilon$:

$$\begin{aligned} &\sum_{i=1}^{n} d_{\mathfrak{G}}^2([y_i], f(X_i)) + \sum_{a=1}^{J}\sum_{j=1}^{p} \lambda_{j,a} b_{j,a}^\top \mathcal{S}_j b_{j,a} \\ \leq\ &\sum_{i=1}^{n} d_{\mathfrak{G}}^2([y_i], \tilde{f}(X_i)) + \sum_{a=1}^{J}\sum_{j=1}^{p} \tilde{\lambda}_{j,a} \tilde{b}_{j,a}^\top \mathcal{S}_j \tilde{b}_{j,a}. \end{aligned}$$

The idea is to show that even if we change the parameters of the additive model in $\mathcal{A}$ by a small amount, the representatives of the observations are still optimally aligned w.r.t. the changed regression function with probability one. Then, since the additive model in $\mathcal{A}$ of the unchanged parameters minimizes (3.26), the proof follows.

First, let us define

$$v = \min_{T\in\mathcal{T},\, i\in\{1,\dots,n\}} \{d^2_{\mathcal{A}}(T_i y_i, h(X_i)) - d^2_{\mathcal{A}}(y_i, h(X_i))\} > 0. \tag{D.5}$$

We will prove later, that we can define v almost surely.

Next, we note that the function $\beta \mapsto h(x)$ is continuous for fixed $x \in \mathbb{R}^r$. Then, we can find a $\epsilon > 0$ such that for $\|\beta - \tilde{\beta}\| < \epsilon$ and for all $i = 1, \dots, n$:

$$d^2_{\mathcal{A}}(h(X_i), \tilde{h}(X_i)) < \frac{v}{2}.$$

Now we assume $\tilde{\beta} \in B(\beta, \epsilon)$ where $B(\beta, \epsilon)$ is a ball in $\mathcal{A}$ with radius ϵ around β.

We note that by the definition of v we have for all $T \in \mathcal{T}$ and all $i = 1, \dots, n$:

$$d^2_{\mathcal{A}}(y_i, h(X_i)) \leq d^2_{\mathcal{A}}(T_i y_i, h(X_i)) - v$$

Then, we obtain for all $i = 1, \dots, n$:

$$\begin{aligned}
& d^2_{\mathcal{A}}(y_i, \tilde{h}(X_i)) + \sum_{a=1}^{J}\sum_{j=1}^{p} \tilde{\lambda}_{j,a} \tilde{b}^{\top}_{j,a} \mathcal{S}_j \tilde{b}_{j,a} \\
\leq\ & d^2_{\mathcal{A}}(h(X_i), \tilde{h}(X_i)) + \sum_{a=1}^{J}\sum_{j=1}^{p} \tilde{\lambda}_{j,a} \tilde{b}^{\top}_{j,a} \mathcal{S}_j \tilde{b}_{j,a} + d^2_{\mathcal{A}}(y_i, h(X_i)) \\
\leq\ & d^2_{\mathcal{A}}(h(X_i), \tilde{h}(X_i)) + \sum_{a=1}^{J}\sum_{j=1}^{p} \tilde{\lambda}_{j,a} \tilde{b}^{\top}_{j,a} \mathcal{S}_j \tilde{b}_{j,a} + d^2_{\mathcal{A}}(y_i, Th(X_i)) - v \\
<\ & \frac{v}{2} + \sum_{a=1}^{J}\sum_{j=1}^{p} \tilde{\lambda}_{j,a} \tilde{b}^{\top}_{j,a} \mathcal{S}_j \tilde{b}_{j,a} + d^2_{\mathcal{A}}(y_i, Th(X_i)) - v \\
\leq\ & -\frac{v}{2} + \sum_{a=1}^{J}\sum_{j=1}^{p} \tilde{\lambda}_{j,a} \tilde{b}^{\top}_{j,a} \mathcal{S}_j \tilde{b}_{j,a} + d^2_{\mathcal{A}}(h(X_i), \tilde{h}(X_i)) + d^2_{\mathcal{A}}(y_i, T\tilde{h}(X_i)) \\
<\ & d^2_{\mathcal{A}}(y_i, T\tilde{h}(X_i)) + \sum_{a=1}^{J}\sum_{j=1}^{p} \tilde{\lambda}_{j,a} \tilde{b}^{\top}_{j,a} \mathcal{S}_j \tilde{b}_{j,a}.
\end{aligned}$$

This shows that for a small enough change of the parameters β of the additive model in $\mathcal{A}$, the representatives of $[y_1], \dots, [y_n]$ are still optimally aligned almost surely. Since h corresponding to the unchanged parameters minimizes

$$PLS_{\mathcal{A}} = \sum_{i=1}^{n} d^2_{\mathcal{A}}(y_i, h(X_i)) + \sum_{a=1}^{J}\sum_{j=1}^{p} \lambda_{j,a} b^{\top}_{j,a} \mathcal{S}_j b_{j,a}$$

the algorithm converges to a local minimum of (3.28).

To complete the proof, we have to show that we can define v almost surely as in (D.5), i.e. $v > 0$ almost surely. This means, we have to prove that for given representatives $\{y_1, \dots, y_n\}$ from $\{[y_1], \dots, [y_n]\}$ and a regression function $f = \pi \circ h$ obtained from algorithm 2 for all $T \in \mathcal{T}$

and $i = 1, \ldots, n$, it holds almost surely

$$d^2_{\mathcal{A}}(y_i, h(X_i)) \neq d^2_{\mathcal{A}}(Ty_i, h(X_i)).$$

This is equivalent to prove that we have with probability zero

$$d^2_{\mathcal{A}}(y_i, h(X_i)) = d^2_{\mathcal{A}}(Ty_i, h(X_i))$$

since all observations in the last iteration of algorithm 2 are optimally aligned w.r.t. to h. Thus, we want to show that the probability of the set

$$X_{\mathcal{T}} = \left\{ (x_1, [y_1]), \ldots, (x_n, [y_n]) \in \mathbb{R}^r \times \mathfrak{G} \;\middle|\; \begin{array}{l} d^2_{\mathcal{A}}(y_i, h(X_i)) = d^2_{\mathcal{A}}(Ty_i, h(X_i)) \\ \text{for at least one representative} \\ y_i,\ i = 1, \ldots, n, \text{ and } T \in \mathcal{T} \end{array} \right\}$$

w.r.t. to the probability measure $\mathbb{P}_X \times \mathbb{P}_Y$ is zero.

We would aim to do this similarly as in Calissano et al. (2022). In particular, this could be potentially done considering the three cases described in subsection 3.4. Therefore, the parameters in (3.29) and their probability to lie on the boundary of a FD could be studied in more depth, developing reasonable assumptions e.g. on the smoothing parameters. This would lead probably to a similar result as in Calissano et al. (2022, lemma 2). However, this was out of the scope of this thesis.

E. Additional visualisations of application outcomes

This section presents further visualisations of the outcomes of the application from section 5. We aim to present ideas to analyse the GAMs in the graph space. First, more regression functions from the single target models can be found in the figures E.8, E.9 and E.10. Afterwards, we display the tensor product smooths for time and Covid-19 cases in figure E.11. So far, the figures correspond to interpretations on graph attribute level. Alternatively, there is also the possibility to evaluate the results using network summary statistics. Therefore, we compute the eigenvector centrality of the nodes of the fitted and predicted networks of the combined models (5.4) which are shown in figure E.12. Intuitively, the eigenvector centrality measures the rank of a node by measuring the importance of the neighbors of the node. The measure was introduced in Bonacich (1987) and is also described e.g. in Kolaczyk & Csárdi (2020, chp. 4.2.2). This could be an interesting measure for the graph space since it potentially relates also to the permutation of two nodes. Similarly, we plot the degree of the nodes of the fitted and predicted networks of the model in figure E.13. A definition for the degree of nodes can be found in Kolaczyk & Csárdi (2020, chp. 4.2.1).

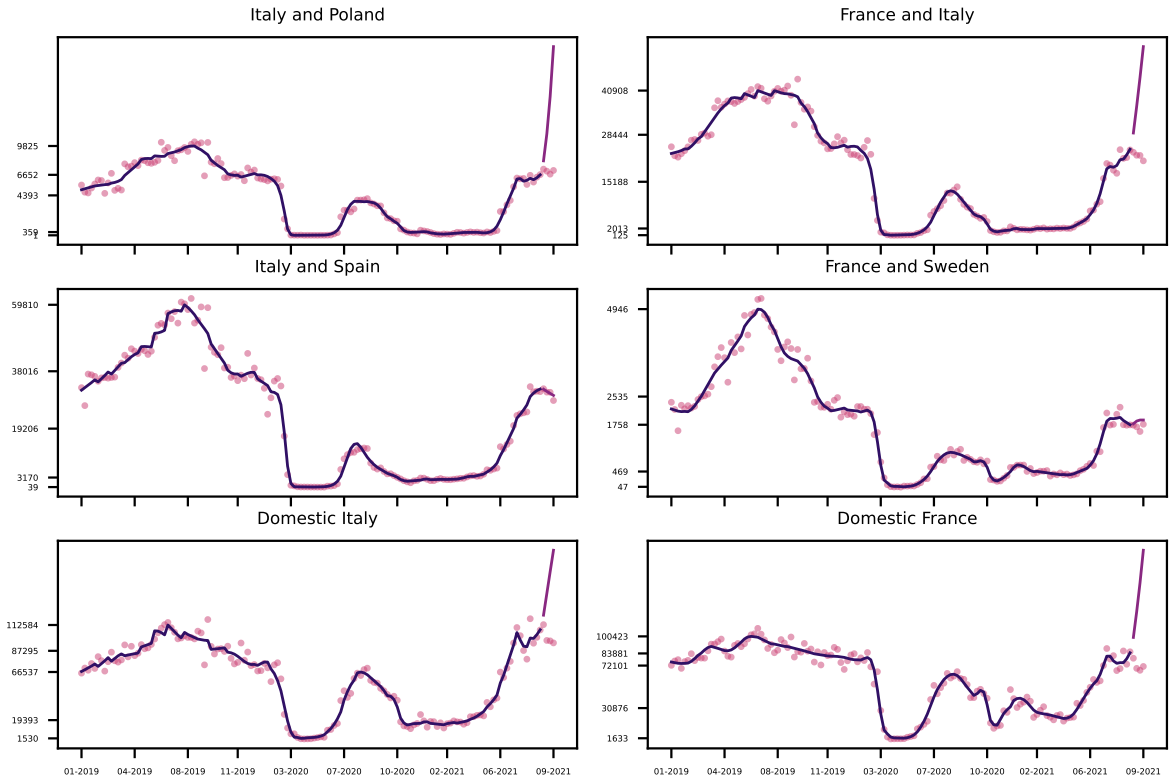

Figure E.8.: Raw passengers (points), fit (line, dark violet) and predictions (line, purple) of the single target models. The horizontal axis shows the time corresponding to the covariate vectors, is labeled by months and years at the ticks and is shared over all plots. The vertical axis represents the number of passengers and the ticks are printed at the five quantiles $0, 0.25, 0.5, 0.75, 1$ of the fitted regression function evaluated at the covariate values used for fitting. The titles show the depicted graph attributes. Since there are no permutations, the titles can refer directly to countries.

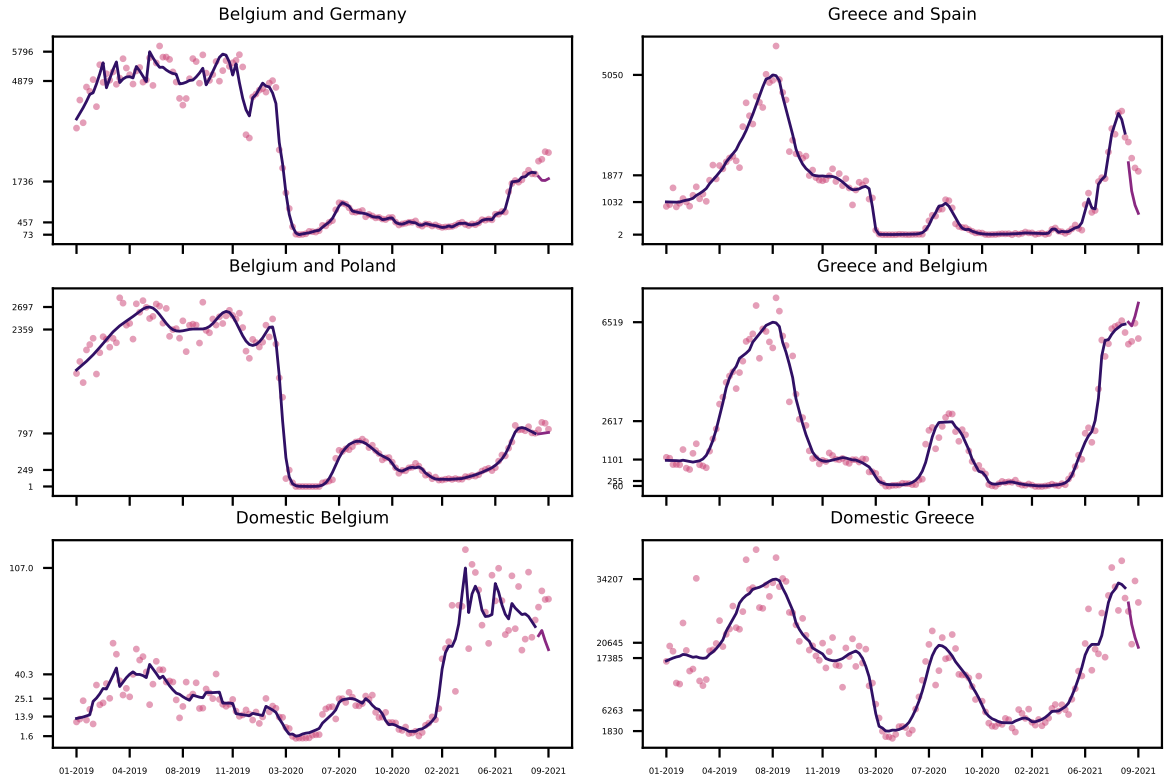

Figure E.9.: Raw passengers (points), fit (line, dark violet) and predictions (line, purple) of the single target models. The horizontal axis shows the time corresponding to the covariate vectors, is labeled by months and years at the ticks and is shared over all plots. The vertical axis represents the number of passengers and the ticks are printed at the five quantiles $0, 0.25, 0.5, 0.75, 1$ of the fitted regression function evaluated at the covariate values used for fitting. The tick for the 0.25 quantile of the plot 'Greece and Spain' was removed due to overlap. The titles show the depicted graph attributes. Since there are no permutations, the titles can refer directly to countries.

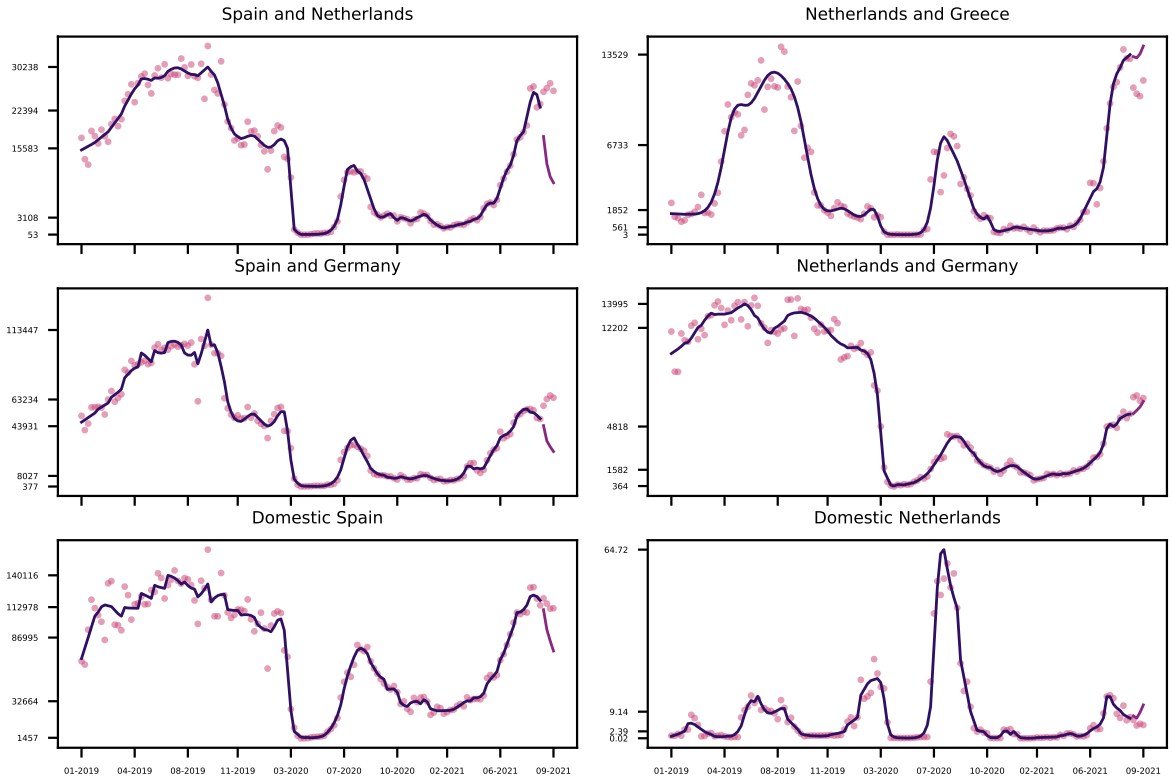

Figure E.10.: Raw passengers (points), fit (line, dark violet) and predictions (line, purple) of the single target models. The horizontal axis shows the time corresponding to the covariate vectors, is labeled by months and years at the ticks and is shared over all plots. The vertical axis represents the number of passengers and the ticks are printed at the five quantiles $0, 0.25, 0.5, 0.75, 1$ of the fitted regression function evaluated at the covariate values used for fitting. The tick for the 0.25 quantile of the plot 'Domestic Netherlands' was removed due to overlap. The titles show the depicted graph attributes. Since there are no permutations, the titles can refer directly to countries.

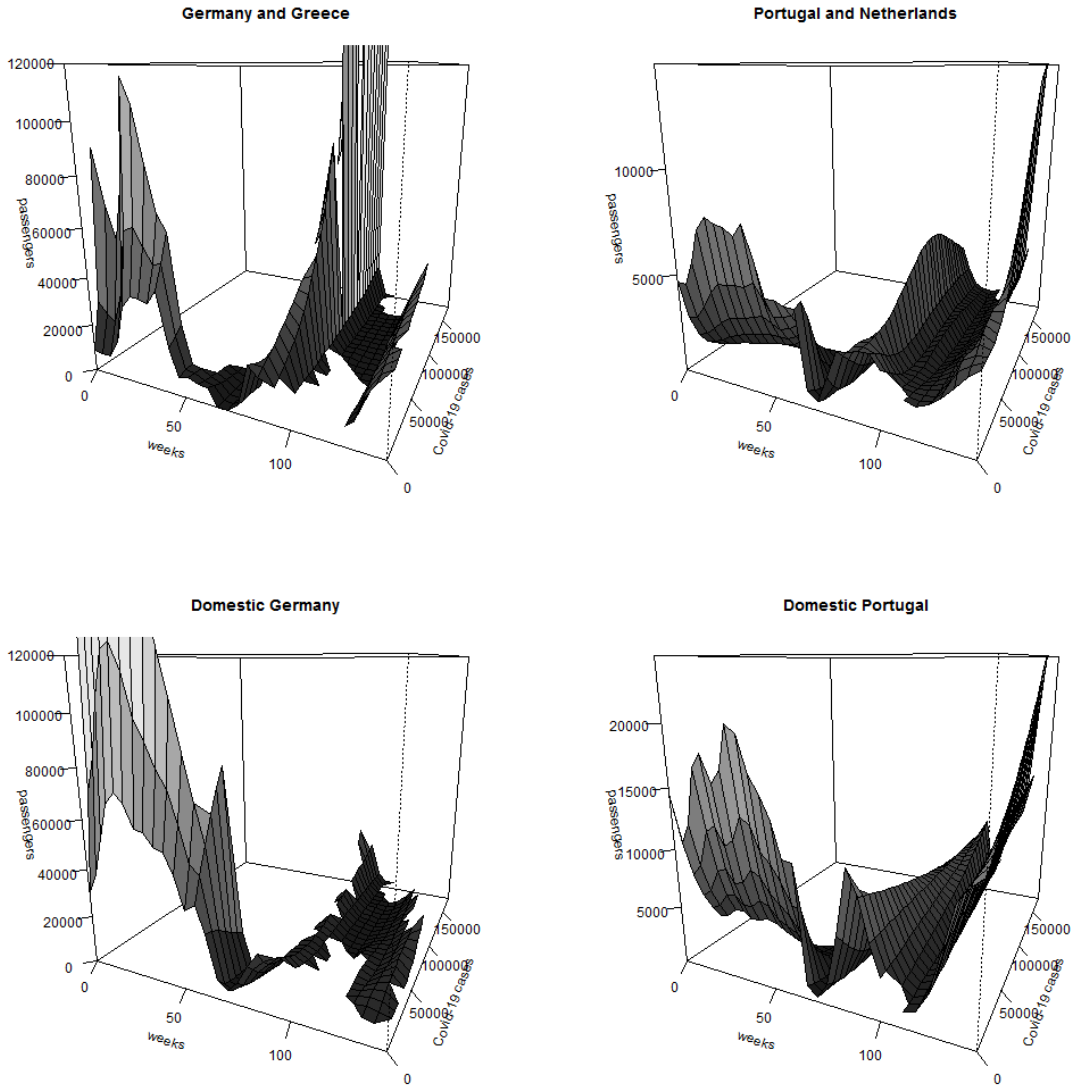

Figure E.11.: Tensor product smooths of weeks together with Covid-19 cases obtained from the vis.gam function from the `mgcv` package. The titles show the connections the smooth corresponds to. The parameter 'too.far' is set to 0.07 for the plots on the left and to 0.15 for the plots on the right to remove the predictions for covariates that are too far from the ones observed. Furthermore, the passenger limit for the plots on the left is set to 120000.

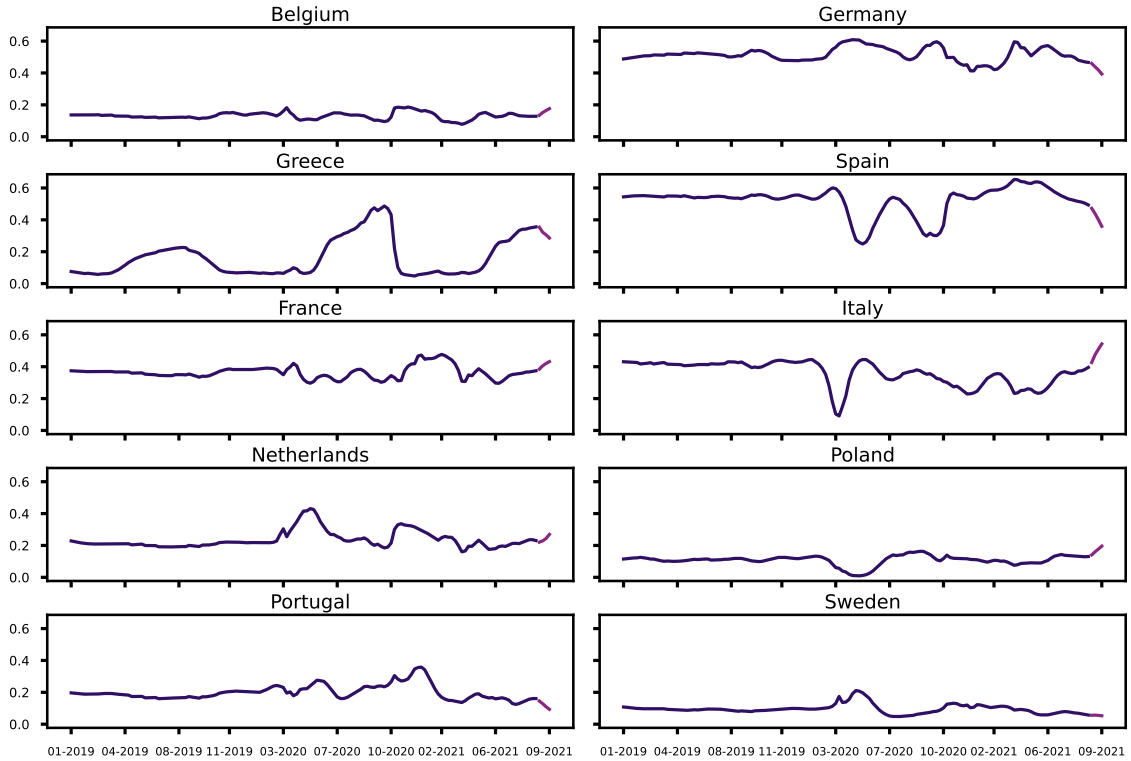

Figure E.12.: Eigenvector centrality of the nodes of the fitted and predicted networks plotted against time. All plots share the same vertical and horizontal axes. The dark violet line corresponds to the fitted regression function, the purple line to the predicted regression function.

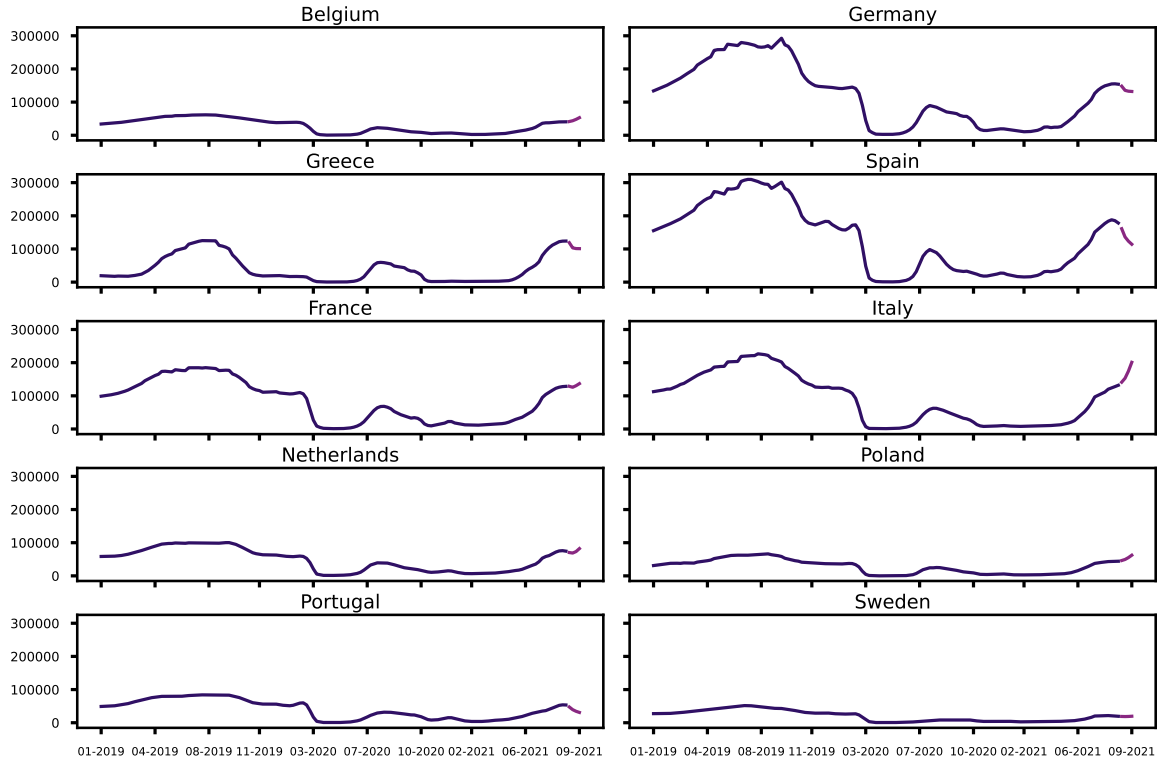

Figure E.13.: Degree of the nodes of the fitted and predicted networks plotted against time. All plots share the same vertical and horizontal axes. The dark violet line corresponds to the fitted regression function, the purple line to the predicted regression function.

I, Marco Simnacher, hereby declare that I have not previously submitted the present work for other examinations. I wrote this work independently. All sources, including sources from the Internet, that I have reproduced in either an unaltered or modified form (particularly sources for texts, graphs, tables and images), have been acknowledged by me as such. I understand that violations of these principles will result in proceedings regarding deception or attempted deception.

Berlin, August 25, 2022

.....

Marco Simnacher