Graph-Valued Regression: Prediction of unlabelled networks in a Non-Euclidean Graph-Space

Calissano, A.; Feragen, A; Vantini, S.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it http://mox.polimi.it

Graph-Valued Regression: Prediction of unlabelled networks in a Non-Euclidean Graph-Space

Anna Calissano[a], Aasa Feragen[b] and Simone Vantini[c]

January 22, 2021

[a] MOX, Dipartimento di Matematica, Politecnico di Milano
Piazza Leonardo da Vinci 32, I-20133 Milano, Italy
E-mail: anna.calissano@polimi.it
[b] DTU Compute
Richard Petersens Plads , Bygning 324, 2800 Kgs. Lyngby
E-mail: afhar@dtu.dk
[c] MOX, Dipartimento di Matematica, Politecnico di Milano
Piazza Leonardo da Vinci 32, I-20133 Milano, Italy
E-mail: simone.vantini@polimi.it

Abstract

Understanding how unlabeled graphs depend on input values or vectors is of extreme interest in a range of applications. In this paper, we propose a regression model taking values in Graph Space, representing unlabeled graphs which can be weighted or unweighted, one or multi-layer, and have same or different numbers of nodes, as a function of real valued regressor. As Graph Space is not a manifold, well-known manifold regression models are not applicable. We provide flexible parametrized regression models for Graph Space, along with precise and computationally efficient estimation procedures given by the introduced Align All and Compute regression algorithm. We show the potential of the proposed model for two real datasets: a time dependent cryptocurrency correlation matrices and a set of bus mobility usage network in Copenhagen (DK) during the Covid-19 pandemic.

Keywords: Graph-valued data; Graph-Valued Regression; Intrinsic Geometric Statistics; Covid-19 Public Transport; Cryptocurrencies

Defining a graph-valued regression model corresponds to building a regression model between a set of of real values (i.e. regressors) and a set of graphs (i.e. responses), which are here modelled in the perspective of object oriented data analysis (1). As discussed by (2), complex data such as networks (or graphs) are often analysed in a *first generation setting*, where a single data point, or graph, is analyzed. Examples off such analysis include node classification or link prediction. In a *second generation setting*, the graphs themselves are the data objects, leading to the analysis of a population of graphs. This is the topic of the present paper. When analyzing a population of graphs, their nodes can be the same (a fixed set of labelled nodes) or different (with either variable sets of labelled nodes, or completely unlabelled graphs). In this work, we focus on graph-valued regression for unlabelled graphs, utilizing *Graph Space*, where unlabelled graphs are represented as equivalence classes obtained by applying a node permutation action to adjacency matrices or tensors (3; 4; 5).

For networks, the first generation setting embodies a well known scientific problem: The prediction of edges and nodes in a given graph. Statisticians and sociologists have been focusing on the analysis of random graphs since the 50s, starting from seminal works by (6). From the Erdős-Rényi model, many different others have been proposed to describe the theoretical distribution behind the network datum and its variability. Exponential Random Graph Models (7; 8) and Stochastic Actor Oriented Models (9; 10) are some examples. Dynamic Network Analysis (DNA) is another stream of literature aiming to model the temporal evolution of a network (see (11) for an overview). Aside from the exploration of generative models behind graphs, the effect of covariates should be taken into account in graph-on-variable regression model. Some examples are: a discrete partition of the space of the covariates to predict labelled networks (12); the definition of a regression model with continuous covariates (13); a regression model for graphs represented as Laplacian matrices (14); a Bayesian approach to the regression process of binary networks (15); a multi-linear regression for a set of tensor data (16). In machine learning, a frequently studied problem is the prediction of nodes and edges from scalars or vectors with Graph Neural Networks (17; 18; 19) or Variational Autoencoder for Graphs (20; 21).

Within the second generation analysis, a considerable amount of work considers population analysis of graphs, where the graph plays the role of the independent, or input, variable. This includes problems such as graph classification, or regressing real-valued properties from graph-valued input. Such problems are often tackled by embedding the graphs, explicitly or implicitly, in a Euclidean feature- or embedding space (22; 23; 18; 24; 25), where much of the relational information is lost. This approach is fine when the information needed to make the prediction can be encoded in a Euclidean feature space.

A more challenging problem is when the predicted dependent variable is a network. Predicting an unlabelled network from a set of variables requires the definition of an interpolating regression function between graphs. We tackle this using interpolation in *Graph Space*. For other types of nonlinear data, such problems are frequently handled using tangent space methods, where regression models are estimated in the Euclidean tangent space (26; 27) of the embedding space (in our case Graph Space). However, these suffer from distorted residuals (28), giving challenging conditions for model fitting. Another easily applicable approach is given by non-parametric kernel smoothing approaches or K-nearest neighbor regression, which have appeared both in manifold statistics (29) and in the more general, stratified, tree-spaces (30). In the context of high dimensional data such as networks, these methods suffer from the curse of dimensionality, which could make them poorly performing in practice. Additionally, as these methods require computing local means or neighborhoods for every test point, they can also have significant computational cost. In terms of parametric models, linear regression models have been generalized to geodesic (26; 31), polynomial (27) and more general parametric (32) regression models, defined exclusively on manifolds. Staying within the manifold-valued regression regime, more recent work also includes manifold-valued models with uncertainty quantification (33; 28). However, as proven in (4), Graph Space is not a manifold, so we cannot apply manifold methods directly.

We address this problem by designing an intrinsic, generalized linear regression model taking values in Graph Space. The resulting parametrized regression models can be given a high level of flexibility via nonlinear basis functions supplied by the user. We provide efficient estimates via an iterative method called Aligned All and Compute (AAC) for regression, which combines statistical precision by using intrinsic, non-distorted residuals, with computational benefits as estimation is effectively made in a Euclidean ”tangent space”.

The paper is organized as follows: In Section 1, we recall the Graph Space and its basic properties. Section 2 introduces the generalized geodesic regression both in theory and in terms of its implementation via the AAC for regression algorithm. All the introduced concepts are shown in practice with two real case study in Section 3: one regarding the analysis of correlation matrices of cryptocurrencies and the other describing the effect of Covid-19 on the usage of public transport system in Copenhagen, Denmark.

1 Graph Space

We consider graphs as triples $G = (V, E, a)$, where the node set V has at most n elements, and the edge set $E \subset V^2$ has maximal size n^2. The nodes and edges are attributed with elements of an attribute space A, which in this paper is assumed to be Euclidean, via an attribute map $a\colon E \to A$. Here, the map a allows us to describe attributes on both edges and nodes, as we use self loop edges (diagonal elements in the graphs' adjacency matrix) to assign attributes to nodes. From here on, we interchangeably use the term network and graph. A graph with scalar attributes is completely specified by a weighted adjacency matrix of dimension $n \times n$, residing in a space $X = \mathbb{R}^{n^2}$ of flattened adjacency matrices. If the attributes are vectors of dimension d, the graph is represented by a tensor of dimension $n \times n \times d$, residing in a space $X = \mathbb{R}^{n \times n \times d}$.

Graphs can have different numbers of nodes and different node labels or - order. In this paper, we assume the existence across the populations of at most n distinct nodes and we add fictionally null nodes to smaller networks, so that all graphs can be described by a fixed-size adjacency matrix. To deal with unlabelled nodes, matching two graphs corresponds to finding optimal permutations of their nodes as first introduced by (3). The group T of node permutations can be represented via permutation matrices, acting on X through matrix multiplication. The binary operation:

$$\cdot\colon T \times X \to X, (T, x) \mapsto Tx$$

thus defines an action of the group T on X. As in (17), the obtained quotient space X/T is called *Graph Space*, and each element of X/T is an unlabelled graph G, represented as an equivalence class $[x] = Tx$ which contains all the flattened adjacency matrices in X which can be obtained from x by permuting nodes. The map $\pi\colon X \to X/T$ given by $\pi(x) = [x]$ can be thought of as a *projection* of the Euclidean total space X onto the Graph Space X/T, and the total space X plays a similar role relative to Graph Space, as the tangent space does for manifolds, by providing a Euclidean space in which approximate computations can be carried out and projected back onto the space of interest – in our case the Graph Space X/T.

Any metric d_X on X defines a quotient pseudo-metric

$$d_{X/T}([x_1], [x_2]) = \min_{t \in T} d_X(tx_1, x_2)$$

on X/T. Since the permutation group T is finite, $d_{X/T}$ is a metric, and the Graph Space X/T is a geodesic space. However, it is neither a vector space, nor a manifold, and its Alexandrov curvature is unbounded from above as shown in (4). As a consequence, well-known strategies from manifold regression (e.g. (26; 31; 27; 32)) are not directly applicable to Graph Space. We define an intrinsic regression model taking values on Graph Space, along with computational tools that allow its practical estimation to be done via an iterative *Align All and Compute (AAC) algorithm for Regression* which operates on the Euclidean total space X.

2 Generalized Linear Regression for Graph Space

Given a sample $(s_1,[x_1]),\dots,(s_k,[x_k])$, where $(s_i,[x_i]) \in \mathbb{R}^p \times X/T$, we aim to describe the relationship:

$$f\colon \mathbb{R}^p \to X/T$$

minimizing:

$$\sum_{i=1}^{k} d^2_{X/T}([x_i], f(s_i)) \tag{1}$$

over all the possible functions belonging to a prescribed family. In this section, we describe how such families of functions in Graph Space X/T can be defined and parametrized, and how to estimate the resulting regression model.

First, we recall the definition of *generalized geodesics*:

Definition 1 (Generalized Geodesics)**.** Denote by $\Gamma(X) := \{\gamma\colon \mathbb{R}^p \to X\}$ the set of all straight lines ($p = 1$), planes, or hyper-planes ($p > 1$) in the total space X. A curve, surface or hypersurface δ is a *generalized geodesic*, or *generalized geodesic subspace*, on the Graph Space X/T, if it is a projection of a straight line, plane, or hyper-plane on X:

$$\Gamma(X/T) = \{\delta = \pi \circ \gamma : \gamma \in \Gamma(X)\}. \tag{2}$$

Next, we consider two potential classes of regression models.

Definition 2 (Generalized Linear Regression Models)**.** Consider the regression model

$$f\colon \mathbb{R}^p \to X/T, \quad s \mapsto f(s) \in X/T$$

where $f \in \Gamma(X/T)$ is a generalized geodesic. This can be written as $f := f_\beta(s) = \pi \circ h_\beta(s)$, where $\pi\colon X \to X/T$ is the canonical projection from total to quotient space, and $h_\beta : \mathbb{R}^p \to X$ is a linear regression on X of the form:

$$h_\beta(s) = \sum_{j=0}^{p} \beta_j \phi_j(s) \tag{3}$$

where the $\phi_j\colon \mathbb{R}^p \to \mathbb{R},\ j = 1, \ldots, p$ are continuous, possibly non-linear, basis functions, for edge- and node-wise coefficients $\beta_j \in X$. Denote by $\mathscr{F}(X/T) := \{f_\beta : \mathbb{R}^p \to X/T\}$, the family of such models. Note that $\mathscr{F}(X/T)$ contains the family $\Gamma(X/T)$ of generalized geodesic regression models.

To simplify the notation, we omit the β writing $h_\beta(s) = h(s)$ and $f_\beta(s) = f(s)$ in the following paragraphs.

Remark 1. The regression model defined in Definition 2 includes the special case where the basis functions are the identity functions:

$$h_\beta(s) = \sum_{j=0}^{p} \beta_j s_j \tag{4}$$

By using the concept of generalized linear models and the concept of alignment with respect to a regression model, the Generalized Linear Regression Model is defined in the following way:

Definition 3 (Generalized Linear Regression)**.** Given a sample $(s_1, [x_1]), \ldots, (s_k, [x_k])$ where $(s_i, [x_i]) \in \mathbf{R}^p \times X/T$, their *Generalized Linear Regression* $f \in \mathscr{F}(X/T)$ is the one that minimizes the residuals as specified by Equation (1).

2.1 From intrinsic residuals on X/T to Euclidean residuals on X

Given a sample $(s_1,x_1), \ldots, (s_k,x_k), (s_i,x_i) \in \mathbb{R}^p \times X$ consisting of independent variables $s_i \in \mathbb{R}^p$ and regressors given by specific graph representatives $x_i \in X$, the modelling of a regression line $h\colon \mathbb{R}^p \to X$ is well known in statistics as a multiple output regression model. This regression line can be projected onto a generalized geodesic in the quotient space. However, this procedure depends entirely on how the representatives x_i for the graphs have been selected – since for any node permutation t_i, the representation $t_i x_i \in X$ is also a representative of the same graph. Here, we introduce the concept of optimal alignment with respect to a regression line in order to select the optimal representatives $t_i x_i \in [x_i], t_i \in T, i = 1, \ldots, k$ for

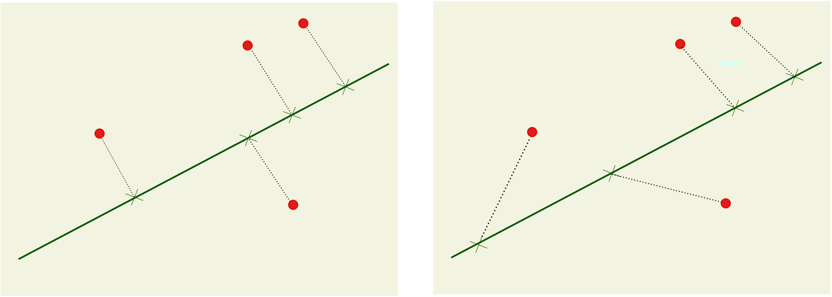

Figure 1: Alignment Differences between the Geodesic Principal Component Alignment and the Geodesic Regression Alignment

the graphs $[x_i]$. The original concept of optimal alignment was introduced to minimize the distance between a geodesic and an equivalence class in the context of quotient space PCA (34). In a regression problem, we instead seek to minimize the prediction residual, which is the distance between the predicted points along the regression line $f(s_i)$ and the observed datum $[x_i]$. The optimal representative tx of the equivalent class $[x]$ is the point that minimizes the distance not with respect to the whole regression line (i.e. the projection along the line), but with respect to the predicted point along the Generalized Linear Regression model:

Definition 4 (Alignment with respect to a regression model)**.** Consider $(s_i, [x_i]) \in \mathbb{R}^p \times X/T, \quad i = 1, \dots, k$, $t \in T$, and $f\colon \mathbb{R}^p \to X/T$ a generalized linear model in X/T with associated $h\colon \mathbb{R}^p \to X$. The graph representative $t_i x_i \in X$ is in *optimal regression position* with respect to the regression line f on X if

$$d_X(t_i x_i, h(s_i)) = d_{X/T}([x_i], f(s_i)). \tag{5}$$

Figure 1 illustrates the conceptual difference between alignment with respect to a Generalized Geodesic Principal Component (see (4; 34) for definition and details) and to a regression model. In our case, the alignment distance is not the distance between an observation and its orthogonal projection onto the line, but the distance between the observation and the associated prediction.

In Theorem 1 we show that this estimation strategy actually corresponds to regressing with intrinsic residuals from X/T.

2.2 Estimation: The Align All and Compute Algorithm for Regression

Inspired by the AAC algorithm for Generalized Geodesic Principal Components defined in (4), we next define an AAC algorithm for regression, where the alignment procedure is adapted to regression.

While the regression model of Definition 3 is framed intrinsically in the Graph Space X/T, we obtain a simple estimation procedure by the *Align All and Compute Algorithm for Regression* (AAC) (4), which combines the euclidean regression model estimation in the total space X with iterative alignment to the current model estimate.

Recall that we are minimizing the sum of squared residuals loss function

$$\sum_{i=1}^{k} d^2_{X/T}([x_i], f(s_i)).$$

The AAC algorithm optimizes the loss with respect to one argument at a time: first with respect to t_i, freezing $f(s_i)$ (i.e. aligning the points), and consequently optimizing with respect to $f(s_i)$ freezing the optimally aligned points $t_i x_i$, by minimizing the corresponding loss function on the total space X:

$$\sum_{i=1}^{k} d^2_X(t_i x_i, h(s_i)) \tag{6}$$

As a result (proven in Theorem 1 below), the loss decreases in every step, which is crucial for its convergence.

In Algorithm 1, the detailed steps of the implementation are shown.

The AAC for Regression is implemented as a method in the *GraphSpace* python package (35).

The following Theorem 1 proves the convergence in finite time of the AAC to a local minimum. See A for proof.

Theorem 1. *Let Graph Space X/T be endowed with a probability measure η which is absolutely continuous with respect to the the push forward of the Lebesgue measure m on X, and let λ be a probability measure absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^p$. Let the sample $\{(s_1,[x_1]),\dots,(s_k,[x_k])\}, (x_i,[x_i]) \in \mathbb{R}^p \times X/T$ be sampled from $\lambda \times \eta$.*

Assume that the AAC for Regression (Algorithm 1) fits the regression model f_β defined in Definition 2. Assume moreover that the basis functions $\phi_j \colon \mathbb{R}^p \to \mathbb{R}$ satisfy the following properties:

Algorithm 1: AAC to compute the Generalized Geodesic Regression

```
Data: $\{(s_1,[x_1]),\dots,(s_k,[x_k])\}$
Result: Generalized Geodesic Regression $f(s) \in \mathscr{F}(X/T)$
Select randomly $t_i x_i \in [x_i]$ among $\{[x_1],\dots,[x_k]\}$, $t_i \in T$;
Align all the observations to the representative $t_i x_i$, obtaining a set of
  points $\{t_1 x_1, t_2 x_2, \dots, t_k x_k\} \in X$ in optimal position with respect to $t_i x_i$;
Perform a Regression $\{(s_1,t_1x_1),(s_2,t_2x_2),\dots,(s_k,t_kx_k)\}$ in $X$ obtaining
  $h(s) \in \Gamma(X)$ solving 6;
Project as $f(s) = \pi \circ h(s)$;
Set $\tilde{f}(s) = f(s)$
while $\delta > \varepsilon$ do
    Align all the points $\{[x_1],[x_2],\dots,[x_k]\}$ with respect to the generalized
      geodesic regression $\tilde{f}(s)$, obtaining a new set of aligned points
      $t_1x_1, t_2x_2, \dots, t_kx_k \in X$;
    Perform GGR on $\{(s_1,t_1x_1),(s_2,t_2x_2),\dots,(s_k,t_kx_k)\}$ in $X$ obtaining
      $h(s) \in \Gamma(X)$ by solving 6;
    Project onto $\mathscr{F}(X/T)$ as $f(s) = \pi \circ h(s)$;
    Compute a step as the distance between the sum of square prediction
      errors $\delta = \mathscr{D}(\tilde{f}(s), f(s))$;
    Align all the observations wrt $f(s)$, obtaining a set of points
      $\{t_1x_1, t_2x_2, \dots, t_kx_k\} \in X$ as explained in 5;
    Set $\tilde{f}(s) = f(s)$.
Return $f(s) \in \mathscr{F}(X/T)$
```

i) $\phi_0 := 1$

ii) *Sample s from* λ *and let* $(\beta_0, \dots, \beta_p) \neq (\tilde{\beta}_0, \dots, \tilde{\beta}_k)$*. Then, with probability* 1*,*

$$\sum_{j=0}^{p} \beta_j \phi_j(s) \neq \sum_{j=0}^{p} \tilde{\beta}_j \phi_j(s).$$

iii) *The matrix*

$$\Phi(S) = \begin{bmatrix} 1 & \phi_1(s_1) & \dots & \phi_p(s_1) \\ \vdots & \ddots & & \vdots \\ 1 & \phi_1(s_k) & \dots & \phi_p(s_k) \end{bmatrix}$$

has full rank.

Under these circumstances, we claim that

a) *The AAC algorithm terminates in finite time, and*

b) *With probability 1, the estimated regression curve* f_β *returned by the AAC algorithm is a local minimum of the function*

$$\beta \mapsto \sum_{i=1}^{k} d^2_{X/T}([x_i], f_\beta(s_i)). \tag{7}$$

Note that the assumptions *i)* and *ii)* are reasonable and hold both for the linear basis functions used in ordinary least square regression model, as well as for e.g. polynomial basis functions.

3 Case Studies

In this section, we show the potential of the model applied to two real datasets. To understand the AAC approach, we will compute at each iteration of Algorithm 1 two errors: the *Regression Error* and the *Post Alignment Error*. The *Regression Error* is the with-in sample prediction error at step *m*. It is computed as the distance between the prediction along the regression line at step *m* (i.e. $h_m(s_i)$) and the observations used to fit the current regression:

$$\sum_{i=1}^{k} d^2_X(h_m(s_i), t_{i(m)} x_i) \tag{8}$$

The *Post Alignment Error* is the distance between the prediction along the line at step m, and the graph representative optimally aligned with respect to this prediction, as defined in Definition 5. Note also, that this distance coincides with the distance in Graph Space X/T between the graphs represented by $h_m(s_i)$ and x_i, where the graph $[h_m(s_i)]$ coincides with $f_m(s_i) = \pi(h_m(s_i))$.

$$\sum_{i=1}^{k} d_X^2(h_m(s_i), t_{i(m+1)}x_i) = \sum_{i=1}^{k} d_{X/T}^2([h_m(s_i)], [x_i]) \tag{9}$$

In other words, this is an intrinsic residual between the Graph Space regression model f and the observation $[x_i] \in X/G$.

Note also that the aligned points obtained at step m are the points used to fit the regression line at step $m+1$.

3.1 Cryptocurrency correlation networks

The analysis of how the stock market evolves in time is a broadly discussed and complex problem, and correlation networks are commonly used to model currency interdependencies (36). From (37), we collect the prices in USD of the nine cryptocurrencies Bitcoin, Dash, Digibyte, Dogecoin, Litecoin, Vertcoin, Stellar, Monero, Verge from the first recorded price of bitcoin on July 18th 2010, until April 3rd 2020. Based on the price data, we compute correlation networks describing how pairs of crypto-currencies vary in price over time by computing, for every 7 days, their correlation over a time period of the following 20 days. In this way, we obtain a set of 506 correlation matrices, split into a training set of 400 matrices and a test set of 106.

The convergence of the algorithm is illustrated in Figure 2, where the plot shows that more than one step is required to converge to the optimal aligned final model. What this tells us, in particular, is that the AAC algorithm is necessary – it would not suffice to align all graphs with a representative of the mean and carry out a single regression model in X; this would suffer from the same distorted residuals as tangent space regression for manifold data.

In Figure 3, we show, for each crypto-currency, the frequency of permutation with other crypto-currencies in the analysis. This carries information on which crypto-currencies have more interchangeable or unique roles in the market. We see that Bitcoin, which appeared significantly earlier than the other crypto-currencies, is very rarely interchanged with the rest. Similarly, Dash, Vertcoin and Monero are interchanged at noticeable rates. In Figure 4, we see the pre-

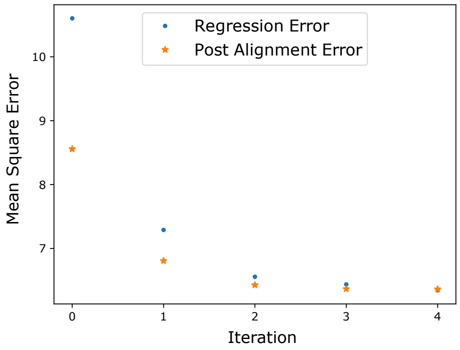

Figure 2: The Regression Error and Post Alignment Error are computed at every iteration of the model estimation procedure. At the final step, $R^2 = 0.381$. The plot shows that the model converges within relatively few iterations, but also that a single iteration, analogous to the tangent space approach from manifold statistics, would not have been sufficient.

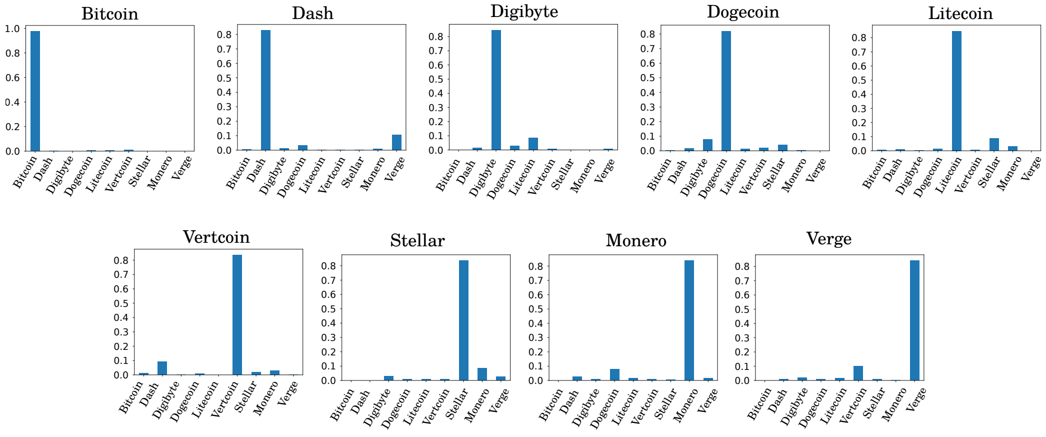

Figure 3: Permutation frequencies for the different cryptocurrencies.

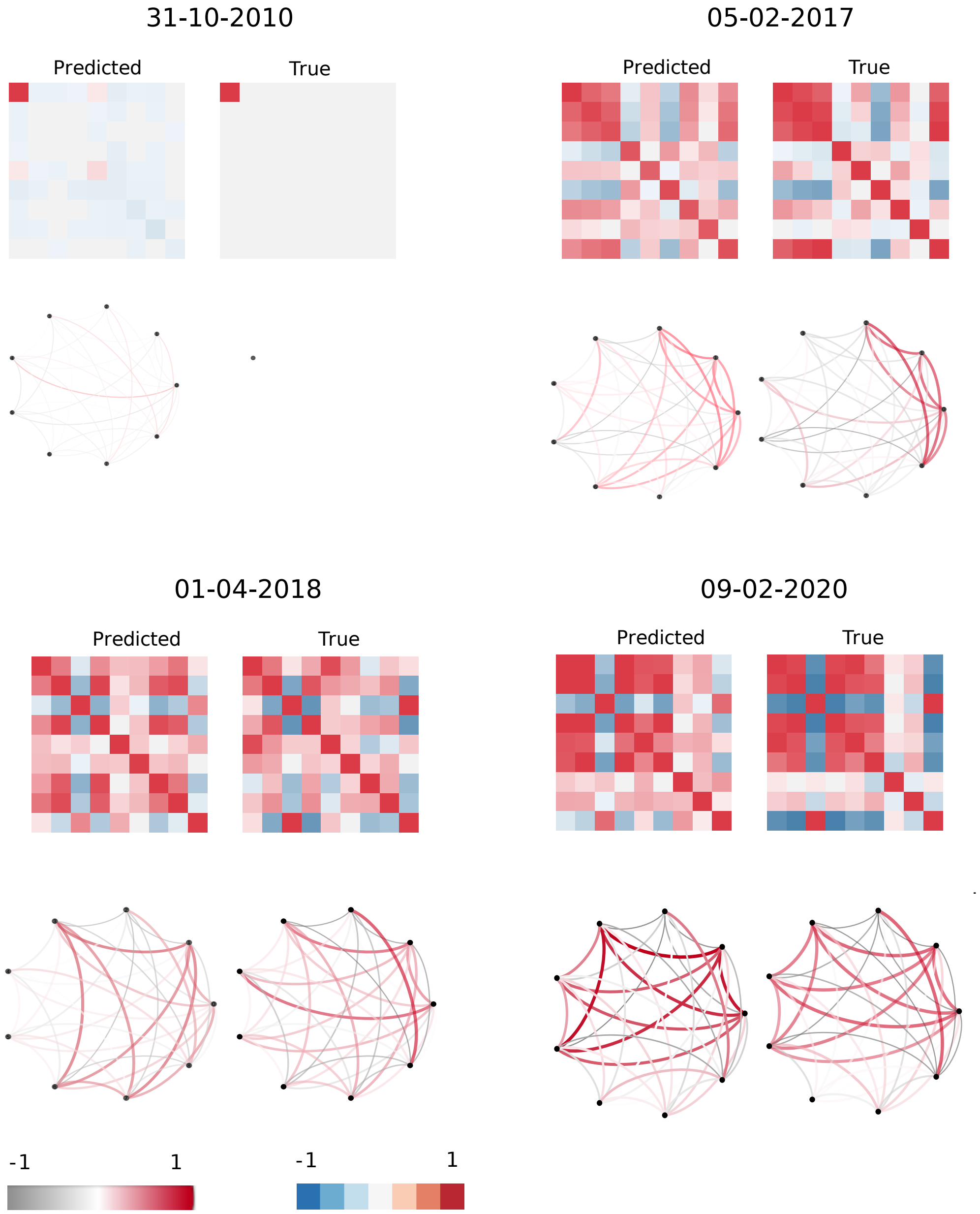

Figure 4: True and Predicted Cryptocurrency represented as heatmaps and networks.

dicted correlation networks at a number of different test time points, along with the ground truth network. The networks are illustrated both as adjacency matrices and plotted as networks for intuitive comparison; note that while the nodes may be permuted compared to their original order, the prediction and ground truth are aligned and thus comparable.

3.2 Public Transport and Covid-19 in Copenhagen, Denmark

In this example, we analyse public transport mobility networks in Copenhagen (Denmark) during the various phases of the Covid-19 epidemic in 2020. The mobility networks are derived from the *Rejsekort* (travel card) data provided by *Movia* - the bus company operating in the Copenhagen Region. The travel card registers the check in and check out on the buses, along with the corresponding bus stop. Our mobility networks are based on trips between the 27^{th} of February and the 13^{th} of May, 2020. We model the bus transport as the daily origin-destination matrix between the 10 different areas in the municipalities of Copenhagen and Frederiksberg. As shown in Figure 5, all the bus stops belonging to an area are aggregated into a single network node. The edges correspond to the number of people travelling between the areas during one day ($00:01$ to $23:59$). The regression model describes the relationship between the origin destination networks and the categorical variable indicating the Covid-19 lock-down phases in Denmark. After the first registered case the 27^{th} of February 2020, Denmark imposed a lock-down from the 13^{th} of March to the 13^{th} of April. During this month, the majority of the activities such as offices, gyms, and pubs were closed. During the following *Phase II*, a slow reopening has been taken place. The time regressors is modelled using three categorical variables describing the different phases. Notice that this regression problem corresponds to an Anova problem, where the analysis of variance is conducted on the set of origin destination networks as a function of a three level categorical variable describing the lockdown phases.

The regression is conducted both in the X space (i.e. without node permutation) and in the X/T space (i.e. with node permutation). The two analyses address two different research questions. In the X space regression, every neighbourhood maintains its own label. It is clear from both the within area trips and the between areas trips (see Figures 6 and 7) that the bus usage during lock-down almost disappears without a full recover in the phase two. If the regression is conducted on the X/T space, the neighbourhoods become interchangeable by allowing node permutation. In Figure 9, we focus on the permutations along time of *Indre By* - the shopping central area in Copenhagen. While before and after the lock-down,

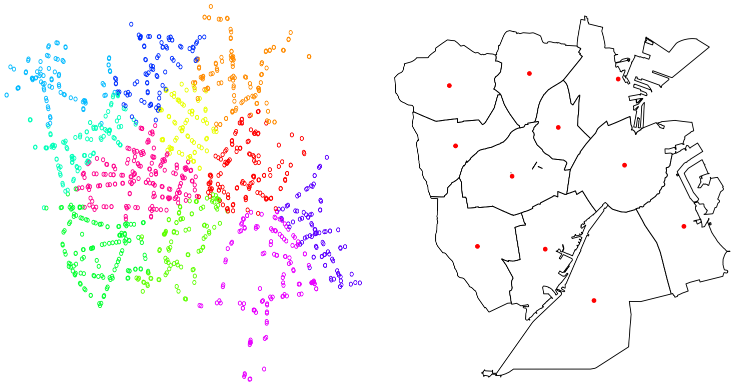

Figure 5: Bus stops in the different areas of Copenhagen and Frederiksberg.

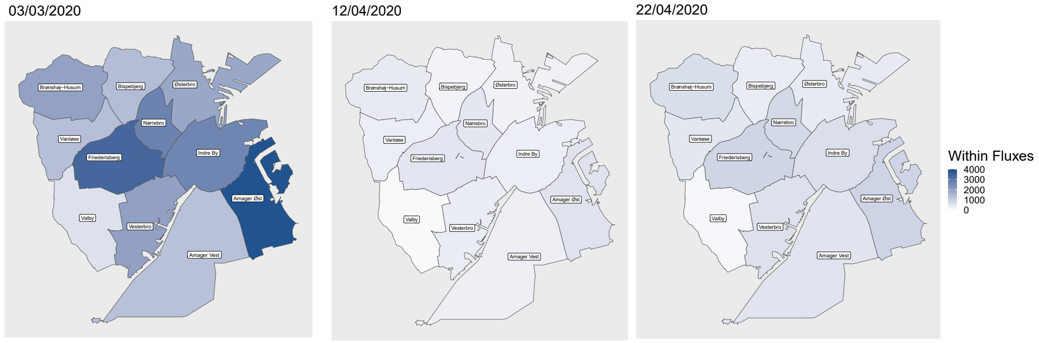

Figure 6: Prediction of the within area fluxes (i.e. the nodes attributes) of three days randomly sampled from the three periods: 03/03/2020, 12/04/2020, and 22/04/2020

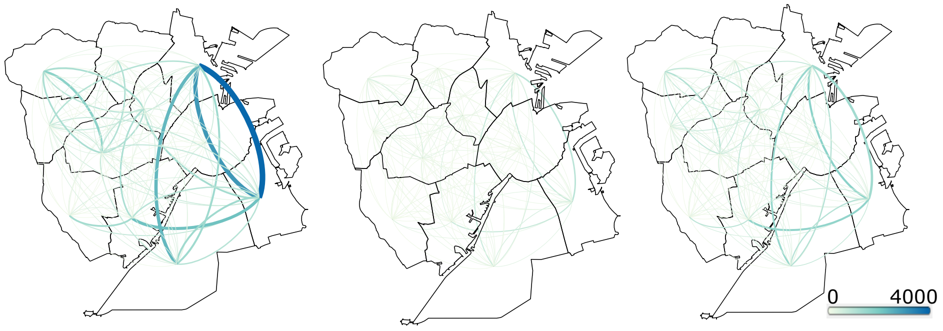

Figure 7: Prediction of the network (i.e. the nodes attributes) of three days randomly sampled from the three periods: 03/03/2020, 12/04/2020, and 22/04/2020

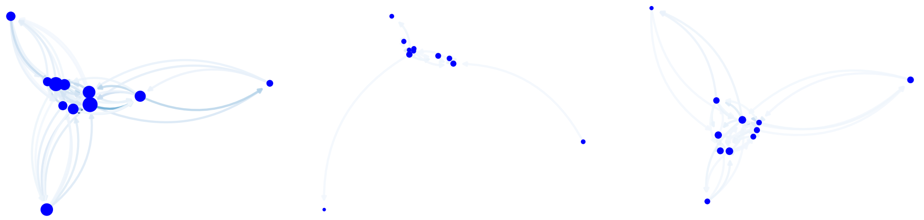

Figure 8: Prediction of the network of three days randomly sampled from the three periods: 03/03/2020, 12/04/2020, and 22/04/2020. The dimension of the node is proportional to the within area flux. The position of the nodes is computed using Spectral Layout of the Networkx python package.

its role in the network is unique, during the lock-down it becomes interchangeable with Valby and Bispebjerg - i.e. two mostly residential areas, showing how the city usage drastically changed. The unique role of Indre By in the city network is immediately recovered after during the Phase II. In Figure 8, the network after the lock-down shows the same structure of the pre-lock-down, but at a lower intensity. While the regression in the X space allows for a local interpretation of each neighbours role in the network in time, the regression in the X/T space offers an unlabeled network vision, focusing on the role that each area is playing in the whole system.

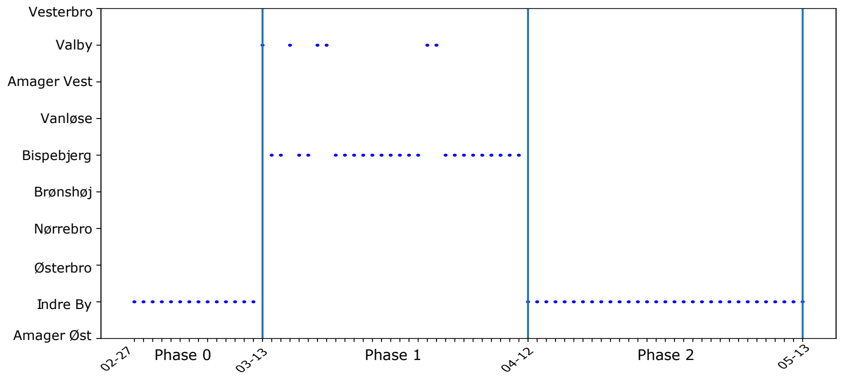

Figure 9: Most popular optimal matching with the node Indre By as a function time. Vertical Lines describe transitions between different phases of the lockdown.

4 Conclusion and Further Development

In this work, we developed a graph-valued regression model, allowing scalars or vectors as independent variable. The model is a generalized linear regression model, which allows for the use of nonlinear basis functions to create nonlinear regressors. The presented model is broadly applicable to every problem where a set of graphs varying according to an external factor. The model applies to graphs which are weighted or un-weighted, directed or un-directed, with coinciding or different nodes which are labelled or unlabelled. These graphs are modelled as points in the quotient space named Graph Space, obtained by applying permutation node action to the space of adjacency matrices. To define an intrinsic regression model, we implemented the Align All and Compute Algorithm for Regression in the Graph Space by iteratively aligning points to the regression line and estimating the Multiple Output Least Square Regression as a regression model on the total space. The experiments show the necessity of a multiple alignment procedure and the utility of the model in different application contexts. As a further development, this algorithmic framework can be extended to different regression strategies such as Gaussian Processes or Neural Networks. Notice that one of the main applications of this framework is time series, and indeed, more tailored time series regression models can be explored for this specific application.

5 Aknowledgment

We thank Movia for the data provided in the second case study, Per Lindblad Johansen and Niklas Christoffer Petersen for their precious help on the data. The research was supported by Centre for Stochastic Geometry and Advanced Bioimaging, funded by a grant from the Villum Foundation.

A Proof of Theorem 1

Theorem 1. *Let Graph Space X/T be endowed with a probability measure η which is absolutely continuous with respect to the the push forward of the Lebesgue measure m on X, and let λ be a probability measure absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}$. Let the sample $\{(s_1,[x_1]),\dots,(s_k,[x_k])\} \subset \mathbb{R}^p \times X/T$ be sampled from $\lambda \times \eta$.*

Assume that the AAC for Regression (Algorithm 1) fits the regression model f_β defined in Definition 2. Assume moreover that the basis functions $\phi_j \colon \mathbb{R}^p \to \mathbb{R}$ satisfy the following properties:

i) $\phi_0 := 1$

ii) Sample s from λ and let $(\beta_0,\dots,\beta_p) \neq (\tilde{\beta}_0,\dots,\tilde{\beta}_k)$. Then, with probability 1,

$$\sum_{j=0}^{p} \beta_j \phi_j(s) \neq \sum_{j=0}^{p} \tilde{\beta}_j \phi_j(s).$$

iii) The matrix

$$\Phi(S) = \begin{bmatrix} 1 & \phi_1(s_1) & \dots & \phi_p(s_1) \\ \vdots & \ddots & & \vdots \\ 1 & \phi_1(s_k) & \dots & \phi_p(s_k) \end{bmatrix}$$

has full rank.

Under these circumstances, we claim that

a) The AAC algorithm terminates in finite time, and

b) With probability 1, the estimated regression curve f_β returned by the AAC algorithm is a local minimum of the function

$$\beta \mapsto \sum_{i=1}^{k} d^2_{X/T}([x_i], f_\beta(s_i)). \tag{10}$$

Remark 2. Equation (4) describes a length n^2 vector (corresponding to a flattened $n \times n$ matrix) of linear regression models whose a^{th} entry is given by

$$h_a(s) = \sum_{j=0}^{p+1} \beta_j(a)\phi_j(s).$$

Note that the index a corresponds to either a node or an edge, and we thus have one regression model for each edge and each node, Generalized geodesic regression on X/T fits linear regression to aligned data point representatives in X, and corresponds to basis functions $\phi_0(s) = 1$ and $\phi_i(s) = s_{(i)}$ where $s_{(i)}$ denotes the i^{th} coordinate of s. However, the model also enables using more general nonlinear basis functions $\phi_i(s)$ as known from Euclidean statistics, leading to linear regression models with potentially nonlinear regressors, such as e.g. polynomial regression.

Remark 3. Conditions ii)-iii) are reasonable: In particular, they hold both for the standard basis of $\mathbb{R}^p$, as well as polynomial basis functions.

Proof. Now we turn to the proof of Theorem 1. First, we prove convergence in finite time. Algorithm 1 consists of two steps repeated iteratively, fitting the generalised regression model (4) to the observations $\{(s_1,[x_1]),\dots,(s_k,[x_k])\} \in \mathbb{R}^p \times X/T$.

Consider the squared error loss function

$$\sum_{i=1}^{k} d_X^2(h^{cur}(s_i), x_i^{cur}), \tag{11}$$

where h^{cur} is our current estimate of the regression model in X, $h^{cur}(s_i) \in X$ is the corresponding regression estimate corresponding to input s_i, and x_i^{cur} is the current representative in X of the sample network $[x_i]$. Note that the first step of Algorithm 1, which aligns output representatives of $[x_i]$ to the corresponding predicted value $h(s_i)$ along the current estimation of the regression line, cannot increase the value of (11) as an improved alignment would indeed lower it. Similarly, the second step of Algorithm 1, which is the re-estimation of the generalized geodesic regression given the new alignments, also cannot increase the value of (11) as, again, an improved estimate would lower its value.

Moreover, if the value of (11) stays fixed two iterations in a row, the algorithm will terminate. Thus, the iterative algorithm will never see the same set of sample-wise alignments twice without terminating. As there are only finitely many such alignments, the algorithm is forced to terminate in finite time.

Next, we turn to proving that the estimated regression model f_β is, indeed, a local minimum. We need to show that for some $\varepsilon > 0$, $\|\tilde{\beta} - \beta\| < \varepsilon$ implies that, with probability 1,

$$\sum_{i=1}^{k} d_{X/T}^2([x_i], f_\beta(s_i)) \leq \sum_{i=1}^{k} d_{X/T}^2([x_i], f_{\tilde{\beta}}(s_i))$$

We shall rely on the following lemma:

Lemma 2. *Given representatives $x_1,\dots,x_k$ of $[x_1],\dots,[x_k]$ with generalized linear regression model $f(s) = \pi \circ h(s)$ obtained minimizing* (11)*, $h(s)\colon \mathbb{R} \to X$, the following holds with probability 1:*

For all $i = 1,\dots,k$ and for all $t \in T \setminus T_{x_i}$,

$$d_X(h(s_i),x_i) \neq d_X(h(s_i),tx_i),$$

where T_{x_i} is the stabilizer $T_{x_i} = \{t \in T | tx_i = x_i\}$.

If the lemma holds, then we may define

$$\nu = \min\{d_X(h(s_i),tx_i) - d_X(h(s_i),x_i) \mid i = 1,\dots,k,\ t \in T \setminus T_{x_i}\} > 0.$$

Since the map $\beta \mapsto h_\beta(s)$ is continuous for any fixed $s \in \mathbb{R}$, where $h = h_\beta$ depends on the weights β as in Eq. 4, we can find some $\varepsilon > 0$ such that $\|\beta - \tilde{\beta}\| < \varepsilon$ indicates $d_X(h(s_i),\tilde{h}(s_i)) < \frac{\nu}{2}$ for all observed independent variables s_i, $i = 1,\dots,k$.

We now consider $\tilde{\beta} \in B(\beta,\varepsilon)$; we wish to show that for all $i = 1,\dots,k$ and all $t \in T \setminus T_{x_i}$, we have $d(\tilde{h}(s_i),x_i) < d(\tilde{h}(s_i),tx_i)$, namely that the optimal representative of $[x_i]$ is left unchanged for all $i = 1,\dots,k$, even if we perturb the regression model. This would complete the proof.

Note that by the definition of ν, we have for any $i = 1,\dots,k$ and $t \in T \setminus T_{x_i}$

$$d(h(s_i),x_i) \le d(h(s_i),tx_i) - \nu.$$

We compute

$$d(\tilde{h}(s_i),x_i) \le \underbrace{d(\tilde{h}(s_i),h(s_i))}_{<\frac{\nu}{2}} + d(h(s_i),x_i) < \frac{\nu}{2} + d(h(s_i),x_i) \le \frac{\nu}{2} + d(h(s_i),tx_i) - \nu$$

$$< -\frac{\nu}{2} + \underbrace{d(h(s_i),\tilde{h}(s_i))}_{<\frac{\nu}{2}} + d(\tilde{h}(s_i),tx_i) < -\frac{\nu}{2} + \frac{\nu}{2} + d(\tilde{h}(s_i),tx_i) = d(\tilde{h}(s_i),tx_i),$$

where the second and fourth inequalities follow from the triangle inequality. This completes the proof of Theorem 1 under the assumption that Lemma 2 holds. □

The proof of Lemma 2 relies on the following :

Lemma 3. *Let $\beta \in \mathbb{R}^{(p+1)\times J}$ be the parameters of the output of AAC as stated in Theorem 1. This β encodes $p+1$ flattened matrices (of dimension J) of coefficients, where β_j is the j^{th} flattened matrix, and $\beta_j(a)$ is the j^{th} coordinate corresponding to the a^{th} node or edge, and we denote by $\beta(a)$ the $(p+1)$-dimensional vector of coefficients for the node or edge a.*

Then, with probability 1, $\beta(a_1) \neq \beta(a_2) \in \mathbb{R}^{(p+1)}$ *for all* $a_1 \neq a_2 \in \{1,\dots,J\}$, *giving* $t\beta \neq \beta$ *for all* $t \in T \setminus \{Id\}$.

Proof. From the analytical solution of a linear regression model in X (see Eq. (3)), we recall that

$$\hat{\beta} = (\Phi(S)^T\Phi(S))^{-1}\Phi(S)^T X.$$

Since, by the assumptions of the theorem, $\Phi(S)$ has full rank, so does $(\Phi(S)^T\Phi(S))^{-1}\Phi(S)^T$. Thus, if $\beta(a_1) = \beta(a_2)$ for some $a_1 \neq a_2$, then the corresponding elements of X belong to the same fiber of $(\Phi(S)^T\Phi(S))^{-1}\Phi(S)^T$, which happens with probability 0. □

Now, we are ready to prove the final Lemma 2.

Proof of Lemma 2. In order to prove the lemma, we will show that the set

$$\mathscr{X}_T = \left\{ \Big((s_1,[x_1]),\dots,(s_k,[x_k]) \Big) \in (\mathbb{R}^p \times X/T)^k \;\middle|\; \begin{array}{c} d(h(s_i),x_i) = d(h(s_i),tx_i) \\ \text{for some representatives } x_1,\dots,x_k, \\ i = 1,\dots,k \text{ and } t \in T \setminus T_{x_i} \end{array} \right\}$$

has measure $(\lambda \times \eta)_k(\mathscr{X}_T) = 0$, where $(\lambda \times \eta)_k$ is the product measure induced by $(\lambda \times \eta)$ on $\underbrace{(\mathbb{R}^p \times X/T) \times \ldots \times (\mathbb{R}^p \times X/T)}_{k}$.

For each element $t \in T$, denote by $X^t = \{x \in X | tx = x\}$ the fixed point set of t. Note that $(\lambda \times \eta)_k(\mathscr{X}_T) = (\lambda \times m)_k((Id_{\mathbb{R}^p} \times \pi)^{-1}(\mathscr{X}_T))$, and that

$$(Id_{\mathbb{R}^p} \times \pi)^{-1}(\mathscr{X}_T) = \bigcup_{i=1}^{k} \bigcup_{t \in T} \mathscr{X}_{i,t},$$

where

$$\mathscr{X}_{i,t} = \left\{ (s_1,x_1,\dots,s_k,x_k) \in (\mathbb{R}^p \times X) \times \ldots \times \underbrace{(\mathbb{R}^p \times X \setminus X^t)}_{i^{th}} \times \ldots \times (\mathbb{R}^p \times X) \;\middle|\; \begin{array}{c} d_X(h(s_i),x_i) \\ = d_X(h(s_i),tx_i) \end{array} \right\}$$

and

$$\mathscr{X}_{i,t} \subset \underbrace{(\mathbb{R}^p \times X) \times \cdots \times (\mathbb{R}^p \times X)}_{k}.$$

The preimage $\mathscr{F}_i^{-1}(0)$ of the function

$$\mathscr{F}_i\colon \underbrace{X \times \cdots \times X}_{k} \to \mathbb{R}, \qquad (s_1,x_1,\ldots,s_k,x_k) \mapsto d_X^2(h(s_i),x_i) - d_X^2(h(s_i),tx_i)$$

satisfies

$$\mathscr{F}_i^{-1}(0) \cap (\mathbb{R}^p \times X) \times \ldots \times \underbrace{(\mathbb{R}^p \times X \setminus X^t)}_{i^{th}} \times \ldots \times /\mathbb{R}^p \times X) = \mathscr{X}_{i,t}.$$

We show that $\mathscr{F}_i$ is a submersion on $(\mathbb{R}^p \times X) \times \ldots \times \underbrace{(\mathbb{R}^p \times X \setminus X^t))}_{i^{th}} \times \ldots \times (\mathbb{R}^p \times X)$ by showing that it has nonzero gradient.

Note that

$$\begin{aligned}
\mathscr{F}_i(s_1,x_1,\ldots,s_k,x_k) &= d_X^2(h(s_i),x_i) - d_X^2(h(s_i),tx_i) \\
&= \|h(s_i) - x_i\|^2 - \|h(s_i - tx_i\|^2 \\
&= (h(s_i) - x_i)^T(h(s_i) - x_i) - (h(s_i) - tx_i)^T(h(s_i) - tx_i) \\
&= 2h(s_i)^T(tx_i - x_i).
\end{aligned}$$

It follows that

$$\nabla_{x_i}\mathscr{F}_i(s_1,x_1,\ldots,s_k,x_k) = 2h(s_i)^T(t - I).$$

We would like to show that $2h(s_i)^T(t-I)$ is nonzero with probability 1. Note that $2h(s_i)^T(t-I) = 0$ if and only if $t^T h(s_i) = h(s_i)$, which also indicates that $th(s_i) = tt^T h(s_i) = h(s_i)$. By Lemma 3, we know that $t\beta \neq \beta$ with probability 1, and by the assumptions of Theorem 1, we then have $th(s_i) \neq h(s_i)$ with probability 1. Hence, we may conclude that with probability 1, $2h(s_i)^T(t-I) \neq 0$, giving

$$\nabla_{x_i}\mathscr{F}_i(s_1,x_1,\ldots,s_k,x_k) \neq 0.$$

It follows that $\mathscr{F}_i$ is a submersion on

$$(\mathbb{R}^p \times X) \times \ldots \times \underbrace{(\mathbb{R}^p \times X \setminus X^t)}_{i^{th}} \times \ldots \times (\mathbb{R}^p \times X).$$

As a result, the set

$$\mathscr{F}_i^{-1}(0) \cap (\mathbb{R}^p \times X) \times \ldots \times \underbrace{(\mathbb{R}^p \times X \setminus X^t)}_{i^{th}} \times \ldots \times (\mathbb{R}^p \times X) = \mathscr{X}_{i,t}$$

has codimension 1 and, in particular,

$$m_k(\mathscr{X}_{i,t}) = m_k(f^{-1}(0) \cap (\mathbb{R}^p \times X) \times \ldots \times \underbrace{(\mathbb{R}^p \times X \setminus X^t)}_{i^{th}} \times \ldots \times (\mathbb{R}^p \times X) = 0.$$

But then,

$$\begin{aligned}(\lambda \times \eta)_k(\mathscr{X}_k) = (\lambda \times m)_k((Id_{\mathbb{R}^p} \times \pi)^{-1}(\mathscr{X}_T)) = (\lambda \times m)_k \left(\bigcup_{i=1}^k \bigcup_{t \in T} \mathscr{X}_{i,t} \right) \\ \leq \sum_{i=1}^k \sum_{t \in T} (\lambda \times m)_k(\mathscr{X}_{i,t}) = 0,\end{aligned}$$

which proves the lemma. □

References

[1] J. S. Marron, A. M. Alonso, Overview of object oriented data analysis, Biometrical Journal 56 (5) (2014) 732–753.

[2] H. Wang, S. J. Marron, Object oriented data analysis: Sets of trees, The Annals of Statistics 35 (5) (2007) 1849–1873.

[3] B. J. Jain, K. Obermayer, Structure spaces, Journal of Machine Learning Research 10 (Nov) (2009) 2667–2714.

[4] A. Calissano, A. Feragen, S. Vantini, Populations of unlabeled networks: Graph spacegeometry and geodesic principal components, MOX Report (2020).

[5] X. Guo, A. Srivastava, S. Sarkar, A quotient space formulation for statistical analysis of graphical data, arXiv preprint arXiv:1909.12907 (2019).

[6] P. Erdős, A. Rényi, On the evolution of random graphs, Publ. Math. Inst. Hung. Acad. Sci 5 (1) (1960) 17–60.

[7] G. Robins, T. Snijders, P. Wang, M. Handcock, P. Pattison, Recent developments in exponential random graph (p*) models for social networks, Social networks 29 (2) (2007) 192–215.

[8] D. Lusher, J. Koskinen, G. Robins, Exponential random graph models for social networks: Theory, methods, and applications, Cambridge University Press, 2013.

[9] T. A. Snijders, Statistical models for social networks, Annual Review of Sociology 37 (2011).

[10] T. A. Snijders, Stochastic actor-oriented models for network dynamics (2017).

[11] K. Carley, Dynamic network analysis, Summary in NRC Workshop on Dynamic social network modeling and analysis (2003) 133–145.

[12] H. Liu, X. Chen, L. Wasserman, J. D. Lafferty, Graph-valued regression, in: Advances in Neural Information Processing Systems, 2010, pp. 1423–1431.

[13] Y. Ni, F. C. Stingo, V. Baladandayuthapani, Bayesian graphical regression, Journal of the American Statistical Association 114 (525) (2019) 184–197.

[14] K. E. Severn, I. L. Dryden, S. P. Preston, Non-parametric regression for networks, arXiv preprint arXiv:2010.00050 (2020).

[15] D. Durante, D. B. Dunson, Nonparametric bayes dynamic modelling of relational data, Biometrika 101 (4) (2014) 883–898.

[16] P. D. Hoff, Multilinear tensor regression for longitudinal relational data, The annals of applied statistics 9 (3) (2015) 1169.

[17] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Transactions on Neural Networks 20 (1) (2008) 61–80.

[18] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, in: International Conference on Learning Representations, 2018.

[19] M. Zhang, Z. Cui, M. Neumann, Y. Chen, An end-to-end deep learning architecture for graph classification, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[20] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: European Semantic Web Conference, Springer, 2018, pp. 593–607.

[21] T. N. Kipf, M. Welling, Variational graph auto-encoders, NeurIPS Workshop on Bayesian Deep Learning (2016).

[22] H. Saigo, S. Nowozin, T. Kadowaki, T. Kudo, K. Tsuda, gboost: a mathematical programming approach to graph classification and regression, Machine Learning 75 (1) (2009) 69–89.

[23] H. Kashima, K. Tsuda, A. Inokuchi, Marginalized kernels between labeled graphs, in: Proceedings of the 20th international conference on machine learning (ICML-03), 2003, pp. 321–328.

[24] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, K. M. Borgwardt, Graph kernels, The Journal of Machine Learning Research 11 (2010) 1201–1242.

[25] H. Maron, H. Ben-Hamu, N. Shamir, Y. Lipman, Invariant and equivariant graph networks, in: International Conference on Learning Representations, 2018.

[26] P. T. Fletcher, Geodesic regression and the theory of least squares on riemannian manifolds, Int. J. Comput. Vis. 105 (2) (2013) 171–185. doi:10.1007/s11263-012-0591-y.

[27] J. D. Hinkle, P. T. Fletcher, S. C. Joshi, Intrinsic polynomials for regression on riemannian manifolds, Journal of Mathematical Imaging and Vision 50 (1-2) (2014) 32–52. doi:10.1007/s10851-013-0489-5.

[28] A. Mallasto, A. Feragen, Wrapped gaussian process regression on riemannian manifolds, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5580–5588.

[29] B. C. Davis, P. T. Fletcher, E. Bullitt, S. C. Joshi, Population shape regression from random design data, in: IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007, IEEE Computer Society, 2007, pp. 1–7. doi:10.1109/ICCV.2007.4408977. URL `https://doi.org/10.1109/ICCV.2007.4408977`

[30] S. Skwerer, Tree oriented data analysis, Ph.D. thesis, University of North Carolina at Chapel Hill Graduate School. (2014). doi:10.17615/w788-sx87.

[31] Y. Hong, R. Kwitt, N. Singh, B. Davis, N. Vasconcelos, M. Niethammer, Geodesic regression on the grassmannian, in: European Conference on Computer Vision, Springer, 2014, pp. 632–646.

[32] Y. Hong, R. Kwitt, N. Singh, N. Vasconcelos, M. Niethammer, Parametric regression on the grassmannian, IEEE transactions on pattern analysis and machine intelligence 38 (11) (2016) 2284–2297.

[33] Y. Hong, X. Yang, R. Kwitt, M. Styner, M. Niethammer, Regression uncertainty on the grassmannian, in: Artificial Intelligence and Statistics, 2017, pp. 785–793.

[34] S. Huckemann, T. Hotz, A. Munk, Intrinsic shape analysis: geodesic PCA for Riemannian manifolds modulo isometric Lie group actions, Statist. Sinica 20 (1) (2010) 1–58.

[35] A. Calissano, A. Feragen, S. Vantini, Graphspace python package, https://github.com/annacalissano/GraphSpace.git (2020).

[36] T. Mizuno, H. Takayasu, M. Takayasu, Correlation networks among currencies, Physica A: Statistical Mechanics and its Applications 364 (2006) 336 – 342. doi:https://doi.org/10.1016/j.physa.2005.08.079.
URL `http://www.sciencedirect.com/science/article/pii/S0378437105010058`

[37] C. Coins, Cryptocurrency database, accessed: 2020-09-04.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

01/2021 Pegoraro, M.; Beraha, M.
Projected Statistical Methods for Distributional Data on the Real Line with the Wasserstein Metric

Franco, N.r.; Massi, M.c.; Ieva, F.; Manzoni, A.; Paganoni, A.m.; Zunino, P.; Et al.
Development of a method for generating SNP interaction-aware polygenic risk scores for radiotherapy toxicity

83/2020 Hron, K.; Machalova, J.; Menafoglio, A.
Bivariate densities in Bayes spaces: orthogonal decomposition and spline representation

84/2020 Vergara, C.; Stella, S.; Maines, M.; Catanzariti, D.; Demattè, C.; Centonze, M.; Nobile, F.; Qua
Computational electrophysiology to support the mapping of coronary sinus branches for cardiac resynchronization therapy

85/2020 Cavinato, L.; Sollini, M.; Kirienko, M.; Biroli, M.; Ricci, F.; Calderoni, L.; Tabacchi, E.; Nann
PET radiomics-based lesions representation in Hodgkin lymphoma patients

82/2020 Vismara, F; Benacchio, T.; Bonaventura, L.
A seamless, extended DG approach for hyperbolic-parabolic problems on unbounded domains

81/2020 Antonietti, P. F.; Mascotto, L.; Verani, M.; Zonca, S.
Stability analysis of polytopic Discontinuous Galerkin approximations of the Stokes problem with applications to fluid-structure interaction problems

80/2020 Zingaro, A.; Dede', L.; Menghini, F.; Quarteroni, A.
Hemodynamics of the heart's left atrium based on a Variational Multiscale-LES numerical model

79/2020 Regazzoni, F.; Salvador, M.; Africa, P.c.; Fedele, M.; Dede', L.; Quarteroni, A.
A cardiac electromechanics model coupled with a lumped parameters model for closed-loop blood circulation. Part I: model derivation

78/2020 Regazzoni, F.; Salvador, M.; Africa, P.c.; Fedele, M.; Dede', L.; Quarteroni, A.
A cardiac electromechanics model coupled with a lumped parameters model for closed-loop blood circulation. Part II: numerical approximation