Statistical analysis of spatio-temporal and multi-dimensional data from a network of sensors

Yiye Jiang

THÈSE DE DOCTORAT

PRÉSENTÉE À

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

PAR **Yiye JIANG**

SOUS LA DIRECTION DE **Jérémie BIGOT** ET **Sofian MAABOUT**

POUR OBTENIR LE GRADE DE **DOCTEUR**

SPÉCIALITÉ : STATISTIQUE APPLIQUÉE

ANALYSE STATISTIQUE DE DONNÉES SPATIO-TEMPORELLES ET MULTIDIMENSIONNELLES ISSUES D'UN RÉSEAU DE CAPTEURS

Soutenue le 7 Decembre 2022 à l'Institut de Mathématiques de Bordeaux

après avis des rapporteurs

Sophie ACHARD	Directrice de recherche, Université Grenoble Alpes
Alexander PETERSEN	Assistant professor, Brigham Young University ...

devant le jury composé de :

Jérémie BIGOT	Professeur, Université de Bordeaux	Directeur de thèse
Sofian MAABOUT ..	Maître de conférences, Université de Bordeaux ...	Co-Directeur de thèse
Bernard BERCU	Professeur, Université de Bordeaux	Président de jury
Sophie ACHARD ...	Directrice de recherche, Université Grenoble Alpes	Rapporteure
Fadila BENTAYEB .	Professeure, Université Lumière Lyon 2	Examinatrice
Nicolas TREMBLAY	Chargé de recherche, Université Grenoble Alpes ..	Examinateur

Analyse statistique de données spatio-temporelles et multidimensionnelles issues d'un réseau de capteurs

Résumé

Cette thèse porte sur l'analyse statistique des séries chronologiques multivariées de différentes natures de données, enregistrées par un réseau de capteurs. L'objectif général est de développer des approches permettant d'explorer et de représenter la dépendance spatiale des séries chronologiques multivariées. Nous nous intéressons tout d'abord à l'identification d'une telle structure de dépendance et à sa représentation par un graphe. En particulier, nous considérons la modélisation des données matricielles et distribuées, c'est-à-dire à chaque instant du temps un vecteur ou une distribution est observé par un des capteurs dans le réseau. À cette fin, nous proposons deux nouveaux modèles autorégressifs (AR) où les graphes gouvernent la dépendance régressive, pour caractériser respectivement les séries chronologiques matricielles et celles multivariées-distribuées. En fittant les modèles aux données, les graphes de dépendance peuvent ensuite être inférés. Pour le modèle AR matriciel, nous nous concentrons sur online learning. Nous considérons notamment la tendance dans le modèle de données en tant que paramètres supplémentaires, puis nous proposons des algorithmes online qui peuvent mettre à jour les estimations du graphe et de la tendance simultanément lors de l'arrivée de nouvelles observations. Pour le modèle AR multivarié-distribué, nous nous appuyons sur les statistiques dans l'espace de Wasserstein pour traiter les objets non-euclidiens de données. Les modèles de régression proposés aident à la prévision de données futures, et les graphes inférés aident à la représentation des données et à l'analyse ultérieure. Par ailleurs, nous étudions les séries chronologiques vectorielles qui sont observées sur un réseau. Ceci est motivé par le fait que, dans de nombreuses applications, les observations sur un réseau présentent une forte dépendance entre les nœuds, ce qui rend les données sur un sous-ensemble de nœuds hautement prévisibles par les données sur les autres. Nous sommes donc intéressés par l'évaluation et le classement de la prévisibilité des nœuds d'un réseau. En guise d'application, les classements dérivés comme stratégies data-drivens servent à la sélection des capteurs. De ce point de vue, la prise en compte du réseau neurone comme méthode de reconstruction est innovante.

Mots-clés: *analyse des séries temporelles, modèles de régression, analyse de données distribuées, données matricielles, graph learning*

Unité de Recherche: Université de Bordeaux, CNRS, Bordeaux INP, Institut de Mathématiques de Bordeaux (IMB), UMR 5251, 351 Cours de la Libération, 33405 Talence, France.

Statistical analysis of spatio-temporal and multi-dimensional data from a network of sensors

Abstract

This thesis focuses on the statistical analysis of multivariate time series of different data natures, recorded by a network of sensors. The general goal is to develop the approaches in order to explore and represent the spatial dependency of the multivariate time series. We are firstly interested in identifying such dependency structure and represent it by a graph. In particular, we consider the modelling of the matrix-variate and distributional data, that is a vector or a distribution is observed per time instant per sensor. To this end, we propose two novel auto-regressive (AR) models where the graphs govern the regression dependency, to respectively characterise the matrix-variate and distributional multivariate time series. By fitting the models to the data, the graphs of dependency can then be inferred. For the matrix-variate AR model, we focus on the online inference. We especially incorporate the trend into the data model as the extra parameters, then we propose the online algorithms which can update the graph and trend estimations simultaneously when the new observation arrives. For the distributional multivariate AR model, we rely on the statistics in Wasserstein space to handle the non-Euclidean object data. The proposed regression models help the prediction of further data, meanwhile, the inferred graphs help the data representation and further analysis. Secondly, we study the vector time series that are observed over a network. We are motivated by the fact that, in many applications, the observations over a network exhibit strong cross-node dependency, which makes the data on a subset of nodes highly-predictable by the data on the rest nodes. We are therefore interested in evaluating and ranking the predictability for the nodes in a network. As an application, the derived rankings can be used in the sensor selection as data-driven strategies. From this aspect, the consideration of the neural network as the reconstruction method is innovative.

Keywords: *time series analysis, regression models, distributional data analysis, matrix-variate data, graph learning*

Contents

Acknowledgements

First of all, I would like to express my gratitude to my supervisor Jérémie Bigot. He gave me this great opportunity to do a thesis on statistical learning and machine learning, which have been my passion since master. His rich knowledge in this domain is the inevitable key to my transition from a classical-model user to an advanced-model creator. In the most difficult periods where, externally, there broke out Covid pandemic, internally, my work had been stuck, it was his patience, responsibility and encouragement that helped me overcome the obstacles, and reached a higher level of research ability. In addition to the help in research, he has been willing to practice French with me. I am very grateful for this kindness, especially in the beginning, I had to stop and search the words during a conversation. Without that each time of discussion in French, I would not have turned fluent in this language during my thesis.

Secondly, I would like to thank my co-supervisor Sofian Maabout for his guidance, support and encouragement. He has been doing his best to help me even though our domains are not exactly the same. I enjoyed very much each of our discussions. Plus, the dinners I spent with his family are those of the warmest memories in Bordeaux.

Further, I would like to thank the referees Sophie Achard and Alexander Petersen for reporting my thesis. Despite their charged schedules, they agreed to be my reporters, that is a big affirmation to me. I appreciate it very much. I would also like to thank members of the jury Bernard Bercu, Fadila Bentayeb, and Nicolas Tremblay for their support with the defense.

Next, I would like to thank Professor Qing Liu. As the coordinator of the bilateral master program that I attended, he offered me the opportunity to exchange in France. Since then and during the following years in master and Ph.D, he did not guide me in an explicit way, but he pays attention to my achievements in his own way. He feels like a sensor member in my own family.

Now, I would like to thank my best of best friends, Abhinandan, Yingjing, and Yulin (in alphabetical order). They are like my stars, lighthouses and harbours, whenever I feel discouraged or lost, they always raise me up. They always make up my daily happiness with our causal conversations, even now when we are not physically closed anymore.

Next I would like to thank Paul, Nicoletta. Thank you for accompanying me all these years at Bordeaux, and our differentiable tears jokes ! Big thank-you also to Fatbio! My first collaborator. We were attending SEME, but during whihch it came lockdown, the inner/outer conditions of me were very tough. But you said it was totally OK, and you supported me a lot. We are friends until today. You are

always very kind ! And especially thank you for bringing me to the châteaux for wine visiting, even though every time I got super carsick. I will keep in heart all the appellations in Medoc !

I would also like to thank my colleague-friends: Theo, Lara, Baptiste, Emanuele, Lan, Hui, Rolando, Gaston, Pei, Gautier, Issa, Jean, Elsa (thank you for introducing me to the IMB community in the first day, without this first sentence, the story would have not been developed), Vasileios, Marc'Anto. You help me to integrate such that I am not an isolated node. Thanks very much !

最后，献给我的母亲，感谢您不辞辛劳的栽培以及岁月里的陪伴；献给我的家人们，感谢你们的温暖真诚，总是把我的心和肚子都填得满满的；以及献给我的母校厦门大学—— 我青春起航，学海扬帆的地方。

Chapter 1

Introduction (Français)

1.1 Données enregistrées sur un réseau de capteurs

Les données enregistrées par un réseau de capteurs sont devenues de plus en plus populaires ces dernières années avec des applications dans de nombreux domaines tels que l'analyse du trafic (Crovella and Kolaczyk, 2003; Yao et al., 2018; Fang et al., 2019), l'analyse du réseau cérébral (Huang et al., 2018; Wang et al., 2020), météorologie (Handcock and Wallis, 1994; Mei and Moura, 2016; Xu et al., 2018). Les réseaux de capteurs peuvent également se situer dans un espace abstrait, qui n'est pas lié à des entités physiques, comme le réseau social (Tabassum et al., 2018), le réseau de citations (Liu et al., 2019), et le réseau sémantique (Lake and Tenenbaum, 2010; Sarica et al., 2020). Ces capteurs enregistrent souvent une séquence d'observations dans le temps à une fréquence régulière, qui correspond à l'évolution du trafic ou de la météo par exemple. De telles observations sont donc dotées d'une nature à la fois spatiale et temporelle. Outre l'aspect spatio-temporel, l'observation nodale de chaque capteur peut également présenter une représentation riche en données. Trois types de données courants sont scalaire, vecteur et distribution, qui sont donc considérés par cette thèse. Dans ce qui suit, nous donnons une brève présentation de chacune de ces trois structures de données.

Scalaire est le type le plus usuel pour l'observation des nœuds en littérature (Kolaczyk, 2009; Shuman et al., 2013). Dans ce cas, pour un réseau de N nœuds, on observe une valeur $\boldsymbol{x}_{it} \in \mathbb{R}^*$ à chaque nœud $i = 1, ..., N$ et à instant du temps $t \in \mathbb{Z}$. La popularité du type scalaire vient du fait que les observations à chaque instant peuvent être représentées par le vecteur $\mathbf{x}_t := (\boldsymbol{x}_{it})_{i=1}^N$. On peut alors partir de plus d'outils existants, par exemple ceux de l'analyse multivariée des séries temporelles : (Lütkepohl, 2005; Brockwell and Davis, 2009; Neusser, 2016), pour considérer la dépendance temporelle avec la structure spatiale, par exemple, Bach and Jordan (2004); Perraudin and Vandergheynst (2017).

Deuxièmement, l'observation vectorielle fait référence au cas où un vecteur $\mathbf{x}_{it} = (\boldsymbol{x}_{it}^f)_{f=1}^F \in \mathbb{R}^F$ de F caractéristiques est observé à chaque nœud et in-

*La valeur peut également être complexe. Cependant, nous nous concentrons sur les observations réelles pour tous les cas tout au long de cette thèse.

stant. Les observations à chaque instant peuvent être représentées par la matrice $\mathbf{X}_t := (\mathbf{x}_{1t}, \ldots, \mathbf{x}_{Nt}) \in \mathbb{R}^{N \times F}$ avec la ligne et la colonne correspondant à la dimension spatiale et à la dimension de la caractéristique. Dans le temps, l'observation totale est matricielle. De telles observations matricielles sont également courantes dans les applications. Parmi les exemples, citons l'observation de la mesure "epoch × channel" à chaque sujet dans l'analyse de l'EEG (Zhou, 2014; Wang et al., 2020), et les observations de gène × tissu dans l'analyse des données d'expression génétique (Yin and Li, 2012).

Enfin, nous considérons le cadre de l'observation distribuée. À chaque nœud $i = 1, ..., N$ et instant $t \in \mathbb{Z}$, nous observons une distribution, c'est-à-dire, une mesure de probabilité $\boldsymbol{\mu}_{it} \in \mathcal{P}(\Omega)$ supportée sur un intervalle Ω de $\mathbb{R}$. En pratique, une distribution ne peut pas être observée directement et les données disponibles consistent plutôt en des échantillons i.i.d. qui sont générés par elle. Ainsi, pour fitter le modèle, nous devons tout d'abords extraire les distributions, plus précisément leur représentation telle que la densité, la fonction quantile ou la fonction de distribution cumulative, des échantillons par des méthodes numériques, telles que la régression linéaire locale pour les densités lisses (Fan and Gijbels, 2018).

Les données de distribution ont joué un rôle important dans de nombreux domaines scientifiques. Un exemple pertinent est l'analyse des distributions d'indicateurs supportés sur des intervalles d'âge, tels que la mortalité et la fertilité (Mazzuco and Scarpa, 2015; Shang and Haberman, 2020), observés à travers différents pays au fil des ans dans les études démographiques. Un autre exemple important est l'analyse des distributions des rendements boursiers quotidiens dans le domaine des séries chronologiques financières (Kokoszka et al., 2019; Zhang et al., 2021). D'autres exemples incluent les distributions des corrélations entre les paires de voxels au sein des régions du cerveau (Petersen and Müller, 2016) et les distributions des prix des maisons sur plusieurs mois (Zhu and Müller, 2021). Néanmoins, malgré l'existence déja longtemps des données distributionnelles, le développement de leurs outils adaptés n'est qu'un domaine de recherche récemment apparu.

Dans les Figures 1.1 et 1.2, nous démontrons les trois types d'observations avec des ensembles de données réels.

L'objectif général de cette thèse est de développer des approches pour les données de ces diverses formes observées sur un réseau, afin d'explorer et de représenter la dépendance spatiale des N caractéristiques indexées par les noeuds. Dans la section 1.2, nous présentons les problématiques considérés de cette thèse, ainsi que les principales contributions réalisées pour chaque problème.

1.2 Problématiques et principales contributions

Les études menées dans cette thèse s'articulent autour de deux problèmes principaux: l'identification de la structure de dépendance qui régit le processus stochastique sur un réseau, et la compréhension de la prévisibilité des nœuds dans les signaux temporels du graphe.

Le premier problème peut être rattaché au domaine de recherche de *graph learning*

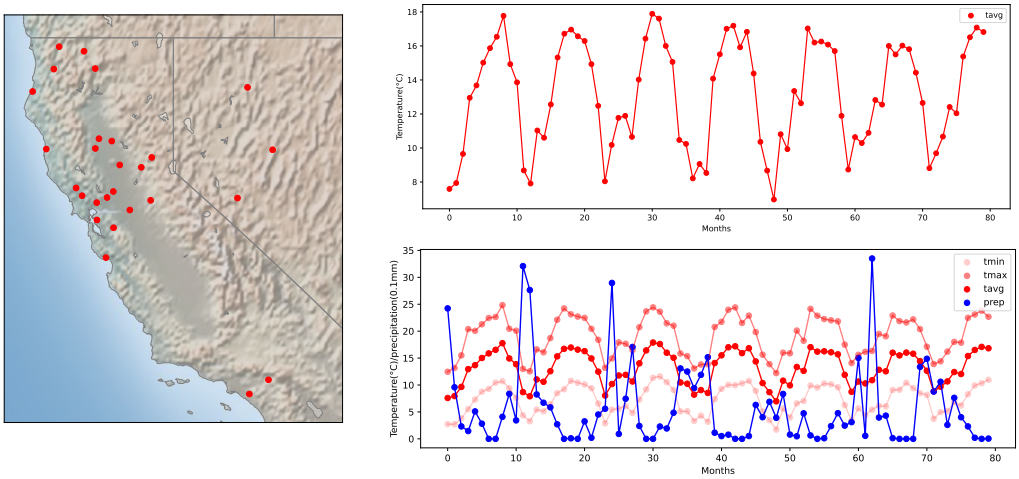

Figure 1.1: *Relevés climatologiques mensuels des stations météorologiques de Californie.* À gauche se trouve le réseau de stations météo de Californie. En haut à droite sont démontrée les observations *scalaires* sur une certaine station (capteur/nœud) i, où une valeur $\boldsymbol{x}_{it} \in \mathbb{R}$ de température moyenne est enregistrée à chaque instant t à i, conduisant à une série temporelle scalaire. En bas à droite sont illustrées les observations *vectorielles* sur une certaine station i, où un vecteur $\mathbf{x}_{it} \in \mathbb{R}^4$ de température min/max/moy et de précipitation est enregistré par temps t à i, conduisant à 4 série temporelle scalaire.

(Dong et al., 2019), où l'objectif central est que, étant donné les observations de multiples caractéristiques représentées par des variables ou processus stochastiques, nous souhaitons construire ou inférer la relation entre les caractéristiques qui prend la forme d'un graphe, les caractéristiques étant appelées nœuds. Dans ce cadre, lorsque l'observation du nœud est de type scalaire, de nombreuses approches ont été proposées pour apprendre de tels graphes. En particulier, pour les processus scalaires, les modèles vectoriels auto-régressifs (VAR) ont été largement adaptés pour déduire la structure d'une relation pertinente, à savoir la causalité de Granger. Le graphe résultant est appelé graphe causal. Nous nous intéressons ensuite à l'extension des modèles VAR aux modèle AR multivarié-distribué et celui matriciel pour servir l'objectif du graph learning. Les modèles dérivés répondent à la demande de développement de modèles personnalisés, étant donné que les séries temporelles matricielles et ceux distribuées deviennent plus populaires, et qu'elles n'ont pas encore été largement étudiées dans la littérature.

Le second problème est motivé par le fait que, dans de nombreuses applications, les observations enregistrées sur un réseau présentent une forte dépendance entre les nœuds, ce qui rend les données observées sur un sous-ensemble de nœuds hautement prévisibles par les données sur les autres nœuds. Ces nœuds sont à l'origine de la redondance des données du réseau, que nous pouvons donc arrêter d'enregistrer pour des raisons de stockage de données par exemple. Nous sommes alors intéressés par le classement de la prévisibilité pour les nœuds d'un réseau, étant donné un

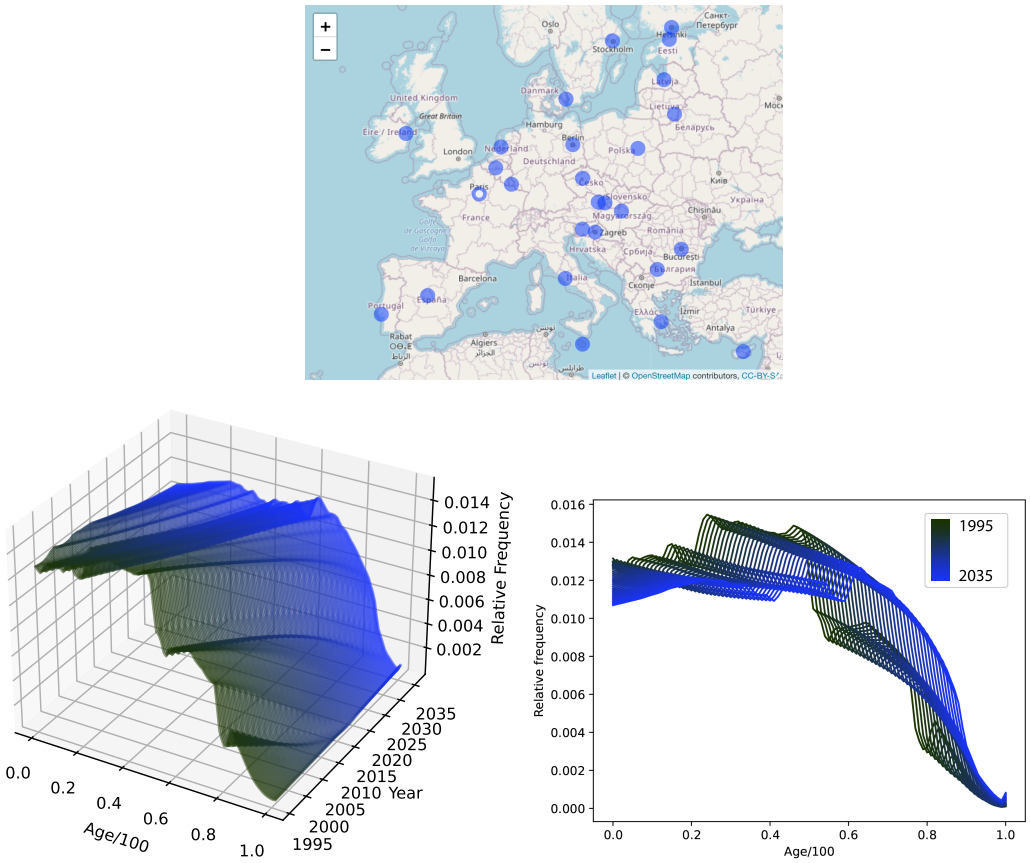

Figure 1.2: *Enregistrements annuels des distributions d'âge des pays de l'Union Européenne.* En haut se trouve le réseau des 27 pays de l'Union Européenne. En bas se trouvent les observations *distribuées* enregistrées à $i =$ France au cours du temps. Une distribution d'âge $\boldsymbol{\mu}_{it} \in \mathcal{P}([0,1])$ est enregistrée à chaque instant t. En bas à gauche, nous visualisons la série chronologique résultante de distribution univariée avec une surface dans le système de coordonnées Âge $\times$ Année $\times$ Fréquence relative. Les données brutes dans ce graphique consistent en 40 distributions annuelles. Nous les complétons avec des échantillons interpolés pour dessiner la surface. En bas à droite, nous montrons la projection de la série chronologique brute sur le plan Âge $\times$ Fréquence relative. On voit que la population vieillit avec le temps.

historique des observations du réseau, en prenant éventuellement en compte en plus la connectivité des nœuds si elle est disponible. Comme application, le classement dérivé peut être utilisé dans la sélection de capteurs, qui est un sujet important dans le domaine du traitement du signal sur graphe, voir par exemple Joshi and Boyd (2009); Sakiyama et al. (2019b).

Dans les sous-sections 1.2.1 et 1.2.2, nous établissons les cadres mathématiques des deux problèmes avec des revues de la littérature non exhaustives, qui permettent de présenter les principales contributions de cette thèse. Tout au long de ce manuscrit, nous utilisons la notation en gras pour distinguer les objets aléatoires dans les modèles contextuels, du reste des constantes.

1.2.1 Structure de dépendance spatiale et modèles auto-régressifs (AR)

Cette section correspond aux préprints Jiang et al. (2021); Jiang (2022).

1.2.1.1 Modèles VAR caractérisés par les graphes causaux

Dans cette section, nous rappelons les modèles VAR usuels ainsi que leurs applications au graph learning à partir des observations nodales de type scalaire. Dans des sections suivantes, nous présenterons les extensions proposées.

Dans l'analyse des séries chronologiques multivariées, le modèle auto-régressif vectoriel fait référence à l'équation différentielle stochastique:

$$\mathbf{x}_t = \mathrm{b} + \sum_{l=1}^{p} A^l \mathbf{x}_{t-l} + \mathbf{z}_t, \quad t \in \mathbb{Z},$$

où $\mathbf{z}_t \sim \mathrm{WN}(0, \Sigma)$ est un processus de bruit blanc avec la variance Σ, et $\mathrm{b} \in \mathbb{R}^N, A^l \in \mathbb{R}^{N \times N}$ sont les coefficients. Comme indiqué précédemment, les modèles VAR sont largement appliqués pour inférer la structure de causalité de Granger de N processus univariés $(\boldsymbol{x}_{it})_t$ pour tout $i = 1, \ldots, N$. La causalité de Granger est définie par paire: $(\boldsymbol{x}_{it})_t$ est dit Granger cause $(\boldsymbol{x}_{jt})_t$, $j \neq i$ si le dernier peut être prédit plus efficacement avec la connaissance du premier dans le passé et à présent prise en compte. La définition plus technique voir Définition 3.2.4. Le graphe causal fait alors référence à un tel graphe où chaque nœud représente une série temporelle univariée, et les arêtes représentent la causalité de Granger. Si les processus sont générés par un VAR(p) stationnaire, $(\boldsymbol{x}_{it})_t$ ne cause pas $(\boldsymbol{x}_{jt})_t$ si et seulement si toutes les ji-ièmes entrées des matrices de coefficients vrais $A^l_{ji} \neq 0, l = 1, \ldots, p$, (Lütkepohl, 2005, Corollaire 2.2.1). Ainsi, nous pouvons retrouver la topologie du graphe à partir de la structure de sparsité commune dans A^l. En petite dimension, cette structure peut être identifiée par le test de Wald, qui teste les contraintes linéaires des coefficients.

En grande dimension, l'inférence du graphe causal exact de Granger est principalement envisagée dans Bolstad et al. (2011); Zaman et al. (2020). Bolstad et al. (2011) proposent de considérer la pénalité group lasso, $\lambda \sum_{i \neq j} \|(A^1_{ij}, \ldots, A^p_{ij})\|_{\ell_2}$, dans le problème usuel des moindres carrés pour les modèles VAR(p), afin d'inférer la structure de sparsité commune des matrices de coefficients $A^l, l = 1, \ldots, p$. Zaman et al. (2020) développent la procédure online pour ce problème d'estimation.

Mei and Moura (2016) définissent une variante du modèle VAR, où les structures sparses des coefficients ne correspondent pas directement à la topologie du graphe, mais à la topologie des voisinages de l-arêtes. Plus précisément, ils supposent que $A^l = c_{l0}I + c_{l1}W + c_{ll}W^l$, où W est la matrice d'adjacence à inférer, et I est la matrice d'identité. De tels modèles peuvent ainsi capturer l'influence de plus de nœuds. L'estimation de la matrice d'adjacence sous-jacente s'appuie sur la pénalité Lasso pour promouvoir la sparsité.

Nous considérons tout d'abord l'extension du modèle VAR(1) au processus matriciel $(\boldsymbol{X}_t)_{t\in\mathbb{Z}} \in \mathbb{R}^{N\times F}$. À cette fin, nous pouvons appliquer directement les modèles vectoriels à la représentation vectorielle $\mathrm{vec}(\boldsymbol{X}_t)$. Cependant, cela n'est pas favorable. D'une part, traiter indifféremment les données de dimensions distinctes peut ignorer et donc gaspiller les informations de la structure intrinsèque des données. D'un point de vue technique, cela entraîne une croissance quadratique du nombre de paramètres dans le modèle, nécessitant alors un grand nombre d'échantillons pour une estimation robuste, ce qui n'est pas facile à satisfaire dans la pratique. Ainsi, la modélisation des données matricielles nécessite des techniques supplémentaires, que nous présenterons dans la section 2.2.1.2.

1.2.1.2 Somme de Kronecker et Product Cartésien de graphes

Dans la littérature, pour étendre les modèles vectoriels aux observations de variables matricielles (plus généralement de variables tensorielles), une pratique courante consiste à appliquer les modèles vectoriels aux données vectorisées, puis à imposer certaines structures aux matrices de paramètres qui encodent des informations sur les dimensions des données. Les structures les plus considérées sont la somme de Kronecker (SK) et/ou le produit de Kronecker (PK). Par exemple, pour étendre la distribution Gaussienne multivariée à la distribution gaussienne matricielle, Gupta and Nagar (2018) vectorisent d'abord la variable aléatoire à valeur matricielle $\mathbf{X} \in \mathbb{R}^{N\times F}$, et suppose que $\mathrm{vec}(\mathbf{X}) \sim \mathcal{N}(\mu, \Omega)$, puis impose la structure PK sur la matrice de variance : $\Omega = V \otimes U$, où $U \in \mathbb{R}^{N\times N}, V \in \mathbb{R}^{F\times F}$. Les deux matrices de sous-variance U et V sont associées séparément aux dimensions des lignes et des colonnes de la matrice de données. Par ailleurs, Kalaitzis et al. (2013) proposent un modèle Gaussien matriciel différent : $\mathrm{vec}(\mathbf{X}) \sim \mathcal{N}(\mu, P^{-1})$, où $P = \Psi \oplus \Theta$, et $\Psi \in \mathbb{R}^{N\times N}, \Theta \in \mathbb{R}^{F\times F}$. C'est-à-dire, ils imposent une structure SK à la matrice de précision[†]. Étant donné que la structure SK d'une matrice d'adjacence correspond au produit cartésien de sous-graphes, l'intégration de la structure SK conduit à un modèle graphique Gaussien qui est interprétable pour les observations de variables matricielles, où la structure de dépendance conditionnelle totale se factorise en sous-structures distinctes des dimensions de lignes et des dimensions de colonnes.

Par conséquent, pour étendre le modèle VAR(1) au processus matriciels $(\boldsymbol{X}_t)_t$, nous pouvons appliquer le VAR(1) usuelle au processus vectorisé $(\mathrm{vec}\boldsymbol{X}_t)_t$, puis imposer la structure SK ou PK à la matrice de coefficients A. Puisque la structure de sparsité

[†]Les reparamétrisations des distributions gaussiennes en terme de leurs matrices de précision sont utilisées dans le domaine du graph learning spécialisé dans les données i.i.d. Gaussiennes, parce que la structure de sparsité de la matrice de précision encode la relation d'indépendance conditionnelle des variables aléatoires.

de A encode la topologie du graphe causal des NF processus $(\boldsymbol{x}_{it}^f)_t, i = 1, \ldots, N, f = 1, \ldots, F$, SK et PK impliquent tous les deux que ce graphe total peut se factoriser en deux sous-graphes, mais de manière différente. Pour expliquer cette différence, nous rappelons la définition de la somme de Kronecker. Soit $A_{\mathrm{F}} \in \mathbb{R}^{F \times F}, A_{\mathrm{N}} \in \mathbb{R}^{N \times N}$ deux matrices carrées, et que I_k désigne la matrice identité $k \times k$. La somme de Kronecker entre A_{F} et A_{N} est définie comme suit

$$A_{\mathrm{F}} \oplus A_{\mathrm{N}} = A_{\mathrm{F}} \otimes I_N + I_F \otimes A_{\mathrm{N}},$$

où $\otimes$ est le produit de Kronecker. Comme indiqué précédemment, les structures PK et SK des matrices d'adjacence impliquent que les graphes correspondants sont les produits des graphes composants. Plus précisément, lorsque $A_{\mathrm{F}}, A_{\mathrm{N}}$ sont les matrices d'adjacence de deux graphes $\mathcal{G}_{\mathrm{F}}, \mathcal{G}_{\mathrm{N}}$, les PK $A_{\mathrm{F}} \otimes A_{\mathrm{N}}$ et SK $A_{\mathrm{F}} \oplus A_{\mathrm{N}}$ sont respectivement les matrices d'adjacence de leur produit tensoriel du graphe $\mathcal{G}_{\mathrm{N}} \times \mathcal{G}_{\mathrm{F}}$ et du produit Cartésien du graphe $\mathcal{G}_{\mathrm{N}} \square \mathcal{G}_{\mathrm{F}}$ (Sandryhaila and Moura, 2014a). Nous illustrons ces deux produits de graphes dans la figure 2.3. Pour les définitions formelles des produits cartésiens et tensoriels des graphes, nous nous référons à Hammack et al. (2011); Chen and Chen (2015); Imrich and Peterin (2018).

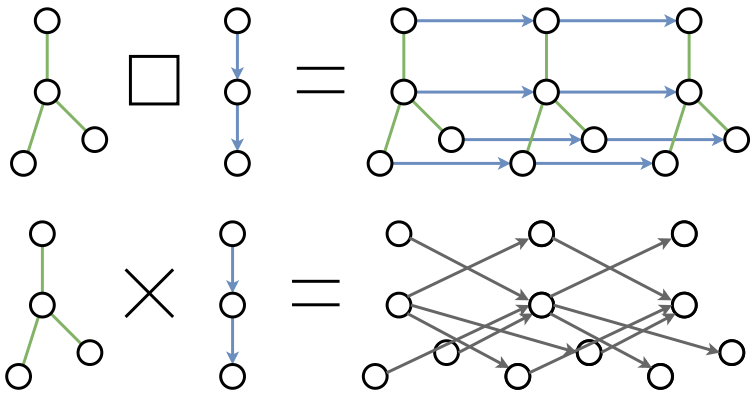

Figure 1.3: *Comparaison des produits cartésien et tensoriel des graphes.* L'ensemble des nœuds de tous les deux produits de graphe est le produit Cartésien des ensemble des nœuds des composantes. Au-dessus, des différents règles d'adjacence s'appliquent. L'exemple est basé sur Sandryhaila and Moura (2014a, Figure 2).

La figure 2.3 montre que les graphes de produits induits par le PK et le SK sont très différents. Par exemple, la structure en treillis du produit cartésien préserve les sous-graphes dans toutes les sections des deux dimensions. En revanche, le produit tensoriel se concentre sur la connexion inter-dimensionnelle, tout en abandonnant la dépendance intra-dimensionnelle. Cette dernière propriété fait en fait référence, dans la littérature sur les processus Gaussiens, «the cancellation of inter-task transfer», voir par exemple Bonilla et al. (2007, Section 2.3). Par conséquent, lorsque les nœuds représentent $(\boldsymbol{x}_{it}^f)_t, i = 1, \ldots, N, f = 1, \ldots, F$, imposer la structure PK (Chen et al., 2021a) implique de supposer qu'il n'existe aucune dépendance de causalité entre $(\boldsymbol{x}_{it}^f)_t, i = 1, \ldots, N$ pour chaque f fixé, qui représente les observations de la caractéristique f à différents nœuds dans le réseau. En revanche, les matrices de coefficients dotées de la

structure SK sont capables de prendre en compte ces dépendances dans l'inférence, qui sont en fait présentes dans de nombreuses applications.

1.2.1.3 Produit causal de graphe et modèle AR(1) matriciel

La première contribution consiste en l'extension du modèle AR(1) vectoriel aux séries temporelles vectorielles, $\mathbf{X}_t \in \mathbb{R}^{N \times F}$, $t \in \mathbb{Z}$, en imposant[‡] la structure SK sur la matrice de coefficient A dans le modèle VAR(1) appliqué à $\mathrm{vec}(\mathbf{X}_t)$:

$$\mathrm{offd}(A) = A_{\mathrm{F}} \oplus A_{\mathrm{N}}, \tag{1.2.1}$$

où $\mathrm{offd}(A)$ est la matrice de même taille que A, dont la partie non diagonale est identique à celle de A, tandis que ses éléments diagonaux sont égaux à zéro. Comme indiqué dans la section 1.2.1.1, pour un VAR(1) stationnaire, la structure de sparsité du coefficient définit le graphe causal des NF composantes de $\mathrm{vec}(\mathbf{X}_t)$. Par intégration du SK, nous supposons ensuite que ce graphe causal total se factorise comme le produit cartésien en le graphe spatial $\mathcal{G}_{\mathrm{N}}$ et le graphe de caractéristiques $\mathcal{G}_{\mathrm{F}}$, où $\mathcal{G}_{\mathrm{N}}$ est préservé dans chaque caractéristique, et $\mathcal{G}_{\mathrm{F}}$ est préservé dans chaque nœud. Un travail relatif qui étend également le modèle VAR(1) au modèle AR(1) matriciel est Chen et al. (2021a), où ils proposent d'imposer la structure PK aux matrices d'adjacence. Cependant, comme nous l'avons indiqué dans la section 1.2.1.2, la PK n'est pas capable de capturer la causalité intra-dimensionnelle.

Étant donné les échantillons $\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_t$, nous développons ensuite deux approches pour estimer $A_{\mathrm{N}}, A_{\mathrm{F}}$ et identifier la structure de sparsité, afin d'inférer la topologie du graphe. Les approches servent respectivement pour l'apprentissage en petite et grande dimension. Nous considérons particulièrement un cadre d'apprentissage général où la sparsité partielle est poursuivie dans l'estimation de seulement A_{N}. Ceci est motivé par un très petit nombre de caractéristiques F généralement présentes dans les applications. Ainsi, le graphe de caractéristiques peut être raisonnablement supposé entièrement connecté. D'autre part, puisque la contrainte de sparsité partielle est également un cas techniquement plus compliqué pour la méthode d'apprentissage proposée en grande dimension. Donc étant donné sa résolution, l'adaptation au cas de sparsité totale ne nécessite pas de nouvelles techniques.

En petite dimension, nous proposons l'estimateur par moindres carrés ordinaire (en équivalent moindres carrés généralisée) projeté afin de profiter des propriétés asymptotiques de l'estimateur classique. Plus précisément, nous d'abords construisons une base orthogonale $(U_k)_k$ de l'espace $\mathcal{K}_{\mathcal{G}}$, qui est un espace linéaire formé par tous les matrices de type $NF \times NF$ à coefficient dans $\mathbb{R}$ qui ont la structure (1.2.1). En nous appuyant sur $(U_k)_k$, nous sommes en mesure d'exprimer, $\mathrm{Proj}_{\mathcal{G}}$, la projection sur $\mathcal{K}_{\mathcal{G}}$ de manière explicite, ce qui définit en outre les estimateurs $\widehat{\mathbf{A}_{\mathrm{N},t}}$ et $\widehat{\mathbf{A}_{\mathrm{F},t}}$ par le biais du SK. Un tel $\widehat{\mathbf{A}_{\mathrm{N},t}}$ est une fonction linéaire par éléments de l'estimateur des moindres carrés ordinaire (MCO) classique de A. En appliquant le théorème de Cramer-Wold sur la TCL de l'estimateur MCO classique, nous dérivons ensuite la TCL pour $\widehat{\mathbf{A}_{\mathrm{N},t}}$. Cette TCL permet en outre d'établir un test de Wald pour identifier la structure de sparsité de A_{N}.

[‡]Nous ne montrons que la structure clé dans cette introduction. La structure complète supposée pour A voir le chapitre 4.

1.2.1.4 Nouveau type de Lasso: Lasso matriciel structuré et ses algorithmes d'Homotopie

Comme discuté à la fin de la section 1.2.1.1, une pratique dans la littérature pour identifier la structure de sparsité des coefficients VAR en régime de grande dimension est d'adopter des estimateurs Lasso. Celui utilisé dans Bolstad et al. (2011); Zaman et al. (2020) est défini comme le minimiseur du problème Lasso (1.2.2) dans le cas VAR(1).

$$\min_{A} \frac{1}{2t} \sum_{\tau=1}^{t} \|\mathbf{x}_\tau - A\mathbf{x}_{\tau-1}\|_{\ell_2}^2 + \lambda_t \|A\|_{\ell_1} , \tag{1.2.2}$$

où $\mathbf{x}_\tau$ est un vecteur d'échantillon, qui peut être $\mathrm{vec}(\boldsymbol{X}_\tau)$ par exemple. Lasso (1.2.2) est le Lasso le plus standard dans la littérature (Hastie et al., 2009, Section 3.4.2). Une grande variété de cadres issus de l'analyse et l'optimisation convexes ont été adaptés pour calculer ses solutions pour différents scénarios, par exemple, Descente par coordonnée (Friedman et al., 2010), méthodes proximales (Beck and Teboulle, 2009), ainsi qu'une technique plus Lasso-orientée «least angle regression» (Efron et al., 2004). Cependant, le Lasso (1.2.2) n'est pas capable d'estimer A structuré et à la composante A_{N} sparse. Par conséquent, motivés par l'estimation, nous proposons le nouveau problème de Lasso-type (1.2.3)

$$\mathbf{A}(t, \lambda_t) = \arg\min_{A \in \mathcal{K}_{\mathcal{G}}} \frac{1}{2t} \sum_{\tau=1}^{t} \|\mathbf{x}_\tau - A\mathbf{x}_{\tau-1}\|_{\ell_2}^2 + \lambda_t F \|A_{\mathrm{N}}\|_{\ell_1} , \tag{1.2.3}$$

La résolution ordinaire de Lasso (1.2.3) peut être effectuée en appliquant par exemple la descente de gradient proximal (Parikh and Boyd, 2014). Dans le cadre de l'algorithme, la contrainte de structure et la sparsité partielle ne posent pas de difficultés supplémentaires, puisque seul le gradient par rapport à $\mathbb{R}^{NF \times NF}$ est calculé dans la forward étape. Ensuite, la backward étape revient à un Lasso standard après projection du gradient sur $\mathcal{K}_{\mathcal{G}}$, comme le montre l'équation (1.2.4).

$$\begin{aligned}
\mathbf{A}^{k+1} &= \mathrm{prox}(\mathbf{A}^k - \eta^k \nabla f(\mathbf{A}^k)), \\
&= \arg\min_{A \in \mathcal{K}_{\mathcal{G}}} \frac{1}{2\eta^k} \left\| A - \left(\mathbf{A}^k - \eta^k \nabla f(\mathbf{A}^k)\right) \right\|_{\ell_2}^2 + \lambda_t F \|A_{\mathrm{N}}\|_{\ell_1} \\
&= \arg\min_{A \in \mathcal{K}_{\mathcal{G}}} \frac{1}{2\eta^k} \left\| A - \mathrm{Proj}_{\mathcal{G}} \left(\mathbf{A}^k - \eta^k \nabla f(\mathbf{A}^k)\right) \right\|_{\ell_2}^2 + \lambda_t F \|A_{\mathrm{N}}\|_{\ell_1} \\
&\iff \begin{cases}
\mathbf{A}_{\mathrm{N}}^{k+1} = \arg\min_{A_{\mathrm{N}}} \left\| A_{\mathrm{N}} - \mathrm{Proj}_{\mathcal{G}_{\mathrm{N}}} \left(\mathbf{A}^k - \eta^k \nabla f(\mathbf{A}^k)\right) \right\|_{\ell_2}^2 + 2\eta^k \lambda_t \|A_{\mathrm{N}}\|_{\ell_1} , \\
\mathbf{A}_{\mathrm{F}}^{k+1} = \mathrm{Proj}_{\mathcal{G}_{\mathrm{F}}} \left(\mathbf{A}^k - \eta^k \nabla f(\mathbf{A}^k)\right) , \\
\mathrm{diag}(\mathbf{A}^{k+1}) = \mathrm{Proj}_{\mathrm{D}} \left(\mathbf{A}^k - \eta^k \nabla f(\mathbf{A}^k)\right) ,
\end{cases}
\end{aligned} \tag{1.2.4}$$

où $\nabla f(\mathbf{A}^k) = \mathbf{A}^k \widehat{\mathbf{\Gamma}}_t(0) - \widehat{\mathbf{\Gamma}}_t(1)$, nous notons $\mathbf{A}^{k+1}(t, \lambda_t)$ par $\mathbf{A}^{k+1}$ pour éviter la notion lourde.

À ce point, nous nous concentrons sur la proposition d'algorithmes permettant de mettre à jour rapidement les solutions précédentes lors des changement de la valeur de l'hyperparamètre ou du terme des données. Pour bénéficier des solutions

précédentes d'un problème, il faut considérer les méthodes spécifiques. Pour le Lasso standard (1.2.2), le cadre des méthodes de continuation par homotopie (Osborne et al., 2000) a été exploré (Malioutov et al., 2005; Garrigues and Ghaoui, 2008) pour calculer la mise à jour rapide. Puisque l'algorithme d'homotopie est dérivé de la condition d'optimalité, qui est par rapport aux matrices dans $\mathcal{K}_{\mathcal{G}}$ pour Lasso (1.2.3), nécessitant de considérer le gradient avec la structure, ainsi les algorithmes d'homotopie existants pour Lasso (1.2.2) ne sont pas applicables. Par conséquent, dans notre travail, nous calculons d'abord la condition d'optimalité de Lasso (1.2.3), basée sur l'expression de la projection sur $\mathcal{K}_{\mathcal{G}}$. Nous dérivons ensuite les deux algorithmes d'homotopie, respectivement pour les trajets de mise à jour $\mathbf{A}(t, \lambda_1) \rightarrow \mathbf{A}(t, \lambda_2)$ et $\mathbf{A}(t, \lambda_2) \rightarrow \mathbf{A}(t+1, \lambda_2)$, ainsi qu'une procédure de réglage adaptatif de l'hyperparamètre de régularisation à l'arrivée d'une nouvelle observation $\mathbf{x}_{t+1}$. Les dérivations ne dépendent pas de la contrainte de structure spécifique, ni de la régularisation de sparsité particulière, et peuvent donc être appliquées à d'autres espaces de structure linéaire $\mathcal{K}_{\mathcal{G}}$ et à la conception de la sparsité.

1.2.1.5 Apprentissage online des graphes et des tendances

Dans les applications où le stockage des données est limité et où l'inférence rapide sur les données arrivant séquentiellement est exigée, la résolution ordinaire qui traite l'ensemble des données en une seule fois échoue. Ainsi, les algorithmes personnalisés qui remplissent ces deux conditions sont nécessaires. Cette classe d'algorithmes est appelée *online*. En revanche, la résolution ordinaire est appelée *off-line*.

Pour la résolution online de Lasso (1.2.3), les algorithmes d'Homotopie précédemment dérivés constituent la méthode pertinente, lors de l'exécution des trois étapes dans l'ordre :

$$\begin{aligned} &\text{Étape 1 : } \lambda_t \rightarrow \lambda_{t+1}, \quad \text{Étape 2 : } \mathbf{A}(t, \lambda) \rightarrow \mathbf{A}(t, \lambda_{t+1}), \\ &\text{Étape 3 : } \mathbf{A}(t, \lambda_{t+1}) \rightarrow \mathbf{A}(t+1, \lambda_{t+1}), \end{aligned} \tag{1.2.5}$$

Cependant, quand nous passons à l'inférence online, la formulation du problème (1.2.3) ne peut pas s'adapter complètement, en raison du terme de données $\frac{1}{t}\sum_{\tau=1}^{t} \|\mathbf{x}_\tau - A\mathbf{x}_{\tau-1}\|_{\ell_2}^2$. Le terme de données implique que la série temporelle $(\mathbf{x}_\tau)_\tau$, $\tau \in \mathbb{Z}$ est supposée admettre une moyenne invariante en temps, qui est zéro. Néanmoins, lorsque les $(\mathbf{x}_\tau)_\tau$ prennent des données brutes, cette hypothèse d'échantillon est très souvent fausse. Au lieu de cela, la série temporelle brute présente généralement la tendance, c'est-à-dire une fonction moyenne variant dans le temps. Ainsi, dans l'apprentissage online, une étape *detrend* est nécessaire, qui approche la fonction de tendance en utilisant l'ensemble des données, puis l'enlève des données brutes. Ensuite, la série chronologique traitée sera plus proche des hypothèses du modèle. Cependant, comme le principe de l'apprentissage online n'exige pas la présence de toutes les données, une telle étape de pré-traitement est interdite. Ainsi, nous devons considérer la tendance comme un paramètre explicite supplémentaire aux paramètres du graphe $A_{\mathrm{N}}, A_{\mathrm{F}}$ dans le modèle online.

Le présent travail se concentre principalement sur un type particulier de tendance, à savoir la tendance périodique, qui est fréquemment rencontrée dans la pratique. Par exemple, une récurrence annuelle tous les 12 mois peut être trouvée dans de

nombreux jeux de données, enregistrés mensuellement sur des années. Du point de vue de la modélisation, la tendance périodique est la fonction moyenne:

$$\mathbb{E}\mathbf{x}_\tau = \mathrm{b}^0_m,\ m = 0, ..., M-1,\ m = \tau \bmod M,$$

où M est la longueur de la période. Nous proposons donc de reformuler Lasso (1.2.3) en incorporant la tendance périodique b^0_m, $m = 0, \ldots, M-1$, sous la forme de l'équation (1.2.6).

$$\mathbf{A}(t, \lambda_t), \mathbf{b}^0_m(t, \lambda_t) = \underset{A \in \mathcal{K}_\mathcal{G}, \mathrm{b}^0_m}{\arg\min} \frac{1}{2t} \sum_{m=0}^{M-1} \sum_{\tau \in I_{m,t}} \|(\mathbf{x}_\tau - \mathrm{b}^0_m) - A(\mathbf{x}_{\tau-1} - \mathrm{b}^0_{m-1})\|^2_{\ell_2} + \lambda_t F \|A_\mathrm{N}\|_{\ell_1}, \tag{1.2.6}$$

où $I_{m,t} = \{\tau = 1, ..., t : \tau \bmod M = m\}$. Notez que b^0_{-1} désigne b^0_{M-1}. Le modèle d'échantillon sous-jacent devient

$$\mathbf{x}_{t-1} - \mathrm{b}^0_m = A(\mathbf{x}_{t-1} - \mathrm{b}^0_{m-1}) + \mathbf{z}_t,\ A \in \mathcal{K}_\mathcal{G},\ m = t \bmod^{\S} M,$$

qui est un modèle VAR(1) augmenté avec une tendance périodique.

Nous adaptons ensuite les algorithmes du Pipeline (1.2.5) au Problème (1.2.6), qui permet enfin l'apprentissage online des graphes et des tendances à partir des séries temporelles matricielles.

1.2.1.6 Statistiques de Wasserstein pour les données distribuées

Deuxièmement, nous aimerions étendre le modèle VAR(1) à une collection de N mesures de probabilité dépendantes du temps $(\boldsymbol{\mu}^i_t)_{t \in \mathbb{Z}}, i = 1, \ldots, N$ (exemple de telles données voir la section 1.1), qui peut en outre identifier et représenter les liens de dépendance significatifs entre les processus dans un graphe orienté pondéré de N noeuds. Au lieu d'un vecteur, la caractéristique du nœud devient maintenant une distribution. Pour modéliser de telles données distributives, nous devons adopter des outils plus avancés.

Puisque les distributions peuvent être caractérisées par les fonctions telles que, les densités, les fonctions quantiles, et les fonctions de distribution cumulative, alors pour analyser les séries temporelles distribuées, on peut chercher à étudier une de ses représentations fonctionnelles avec les outils de l'analyse des séries temporelles fonctionnelles (Bosq, 2000). Cependant, en raison de leurs contraintes non linéaires, telles que la monotonicité et la positivité, les fonctions de représentation des distributions ne constituent pas des espaces linéaires. Par conséquent, les notions de base des modèles VAR standard, telles que l'additivité et la multiplication scalaire, ne s'adaptent pas, de manière directe. Cela entraîne l'échec des modèles dérivés pour les éléments aléatoires d'un espace de Hilbert. Une approche existante consiste à faire correspondre les densités des distributions à des fonctions non contraintes dans l'espace de Hilbert par la transformation de «log quantile density» (LQD) (Petersen and Müller, 2016), puis à appliquer les outils fonctionnels (Kokoszka et al., 2019). Cependant, LQD ne prend pas en compte la géométrie de l'espace de distribution,

[§]Le modulo d'un nombre entier négatif est défini par le reste positif dans ce cas, par exemple, $-1 \bmod M = M - 1$.

et peut donc conduire à des déformations de la distance. Les approches récentes considèrent une telle géométrie en adoptant la métrique de Wasserstein (Bigot et al., 2017; Panaretos and Zemel, 2016; Petersen and Müller, 2019b). Nous établissons donc le modèle de série temporelle dans l'espace de Wasserstein. Dans ce qui suit, nous présentons la principale contribution dans ce piste de travail. Pour l'introduction technique des statistiques dans l'espace de Wasserstein, nous nous référons à la section 3.3.

1.2.1.7 Modèle AR(1) multivarié de Wasserstein et la contrainte N-simplexe

Il existe également d'autres travaux qui se sont penchés sur les modèles de séries temporelles distribuées. Jusqu'à présent, ils se concentrent principalement sur le cas univarié, c'est-à-dire lorsque $N = 1$. En général, les modèles distribués multivariés sont encore peu développés. Notez que dans le cas distribué, deux notions multivariées sont pertinentes, l'une se réfère à la dimension du support de la mesure, tandis que l'autre se réfère au nombre de mesures. Les modèles associés à ces deux notions sont peu développés. Dans ce manuscrit, le mot *multivarié (multivariate)* fait référence à la seconde notion.

Nous nous référons aux travaux récents Chen et al. (2021b); Zhang et al. (2021); Zhu and Müller (2021), qui développent le modèle AR distribué univarié dans l'espace de Wasserstein $\mathcal{W}_2(\mathbb{R})$. Ils proposent de mapper les distributions aléatoires dans l'image logarithmique de l'espace de Wasserstein supporté sur leur moyenne de Fréchet présumée commune. Ils construisent ensuite les modèles de séries temporelles fonctionnelles univariées en termes de cartes logarithmiques. La moyenne commune de Fréchet est la distribution de référence idéale dans le sens où elle annule les espérances des cartes logarithmiques dans la formule de régression. Cependant, pour les séries temporelles multivariées, les différents processus ne peuvent pas avoir une moyenne de Fréchet commune. De plus, il est très peu probable qu'une telle mesure de référence idéale existe (cela sera expliqué dans le chapitre 5). D'autre part, une interception supplémentaire dans un tel modèle de régression fonctionnelle causera de grandes difficultés, lorsque l'on veut conserver le modèle de régression et surtout les prédictions dans l'image logarithmique de l'espace de Wasserstein. Ainsi, pour traiter les moyens de processus inégaux, nous sommes motivés par la formulation équivalente du modèle VAR(1) construit sur les série centrées qui n'inclut donc pas de terme d'interception (détails voir l'équation (3.2.5) dans la section des préliminaires)

$$\mathbf{x}_t - \mathrm{u} = A(\mathbf{x}_{t-1} - \mathrm{u}) + \mathbf{z}_t.$$

Nous proposons tout d'abord un moyen de centrer toutes les distributions brutes par leurs moyennes de processus (Fréchet) de sorte que leurs moyennes deviennent la distribution uniforme, à savoir la mesure de Lebesgue. Ensuite, nous construisons le modèle sur les données centrées $\widetilde{\boldsymbol{\mu}}_t^i$, $i = 1, \ldots, N$, $t \in \mathbb{Z}$. Pour chaque i, nous définissons la relation de régression clé pour la réponse $\widetilde{\boldsymbol{\mu}}_t^i$ et les prédicteurs $(\widetilde{\boldsymbol{\mu}}_{t-1}^j)_j$, $j = 1, \ldots, N$ dans l'espace tangent de la mesure de Lebesgue, comme la combinaison linéaire des cartes logarithmiques de $(\widetilde{\boldsymbol{\mu}}_{t-1}^j)_j$, pondérées par les coefficients $(A_{ij})_j$. La réponse $\widetilde{\boldsymbol{\mu}}_t^i$ est égale à la carte exponentielle de la combinaison

linéaire, poussée par une fonction de distorsion aléatoire. Par conséquent, le total des coefficients du système de régression définit une matrice $A = (A_{ij})_{ij}$ comme dans le modèle VAR(1). Pour les modèles de régression fonctionnels généraux avec réponse fonctionnelle, les coefficients peuvent être des fonctions concurrents $A_{ij}(\cdot)$ ou plus généralement des surfaces $A_{ij}(\cdot, \cdot)$, voir Wang et al. (2015, Equations (14) et (15), respectivement). Néanmoins, nous avons l'intention d'utiliser la forme matricielle pour les coefficients de régression afin d'y représenter directement la structure du graphe.

Comme l'application exponentielle n'est pas injective, le modèle proposé ci-dessus a donc un problème d'identifiabilité. Ainsi, nous conservons en outre le modèle dans l'image logarithmique en ajoutant l'hypothèse N-simplexe A1 aux coefficients (voir ci-dessous), étant donné l'injectivité de l'application exponentielle restreinte sur l'image logarithmique et la convexité de l'image logarithmique. En fait, en sortant de l'image logarithmique, le calcul dans le problème d'estimation n'est pas traçable non plus.

Assumption A1. $\sum_{j=1}^{N} A_{ij} \leqslant 1$ and $0 \leqslant A_{ij} \leqslant 1$.

Nous montrons ensuite que le système de fonctions aléatoires itérées (Wu and Shao, 2004) associé au modèle de série temporelle proposé admet une solution unique, qui est de plus stationnaire en tant que processus fonctionnel dans l'espace de Hilbert de N-fonctions quantiles muni du produit scalaire L_2 usuel, étant donné les hypothèses supplémentaires de contraction sur l'application de régression, qui peuvent être reliées à la condition de stationnarité des modèles VAR.

Enfin, nous nous appuyons sur la méthode des moindres carrés pour dériver l'estimateur de A, qui minimise la somme des résidus au carré sous la contrainte du simplexe : A1. Nous fournissons également la garantie de consistance pour l'estimateur proposé.

1.2.1.8 Application au graph learning à partir de séries temporelles distribuées

En raison de la contrainte simplexe A1, l'estimateur dérivé porte naturellement la sparsité. Ainsi, en interprétant la matrice des coefficients comme la matrice d'adjacence de N nœuds, nous pouvons extraire un graphe orienté pondéré de son estimation, qui représente la structure de dépendance de N processus distribués $(\boldsymbol{\mu}_t^i)_{t\in\mathbb{Z}}, i = 1, \ldots, N$. La Figure 1.4 montre le graphe de structure d'âge inféré des pays à partir des données illustrées dans La figure 1.2. La liste complète des pays étudiés se trouve au chapitre 5.

1.2.1.9 Modèle régressif distribution-à-distribution multivarié

L'approche proposée de centralisation de distributions aléatoires peut également servir au développement du modèle régressif distribution-à-distribution multivarié pour les données i.i.d., qui est également très peu développé. Les travaux existants sur le modèle régressif distribution-à-distribution considèrent principalement le cas d'un prédicteur et une réponse. Dans Chen et al. (2019), la régression ridge à noyau des

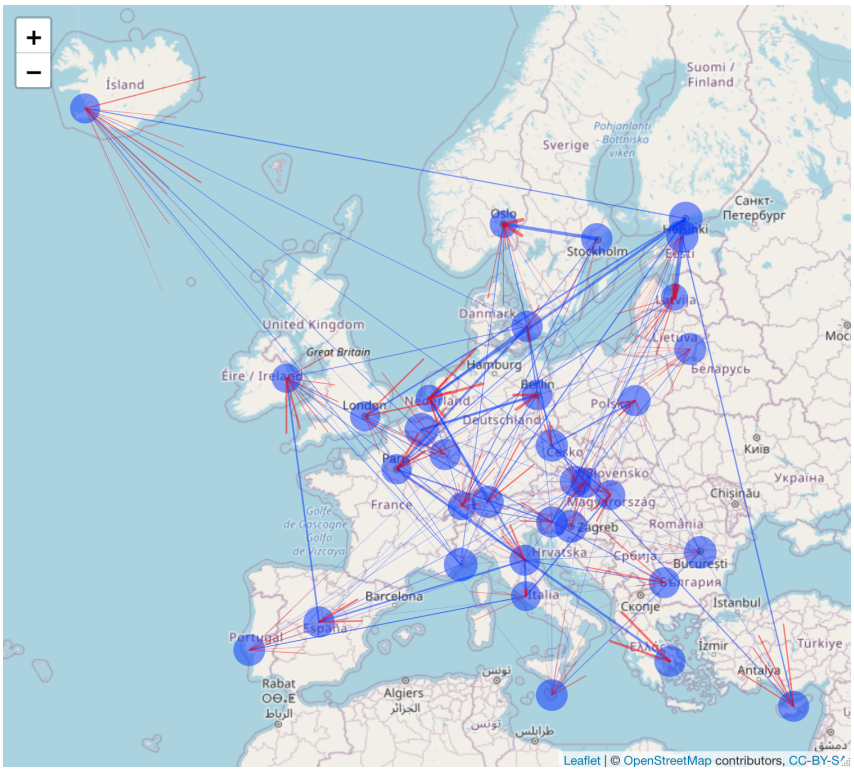

Figure 1.4: *Graphe de la structure d'âge inféré.* Une arête implique la similarité des structures d'âge entre les deux pays de 1996 à 2036 (projeté). L'orientation peut être comprise comme la propagation des changements de la structure. Une arête plus épaisse correspond à une relation plus étroite entre les pays liés, par rapport aux autres arêtes entrantes. La taille du cercle bleu autour d'un pays correspond à la similarité de ses structures d'âge entre deux années consécutives. Un cercle plus grand représente une évolution plus lente.

fonctions de densité transformées par LQD sont construites. Même si la régression peut être étendue au cas multivarié en élargissant le domaine du noyau reproduisant, mais comme indiqué dans la section 1.2.1.6, la transformation LQD ne tient pas compte de la géométrie de l'espace de distribution. Plus récemment, les deux Chen et al. (2021b); Ghodrati and Panaretos (2022) s'appuient sur la métrique de Wasserstein. Chen et al. (2021b) effectue la régression en utilisant l'opérateur de coefficient qui établit une correspondance entre l'espace tangent de la moyenne de Fréchet du prédicteur et celui de la réponse. En revanche, Ghodrati and Panaretos (2022) effectue la régression directement dans l'espace de Wasserstein. Le prédicteur est pushforwardé par la carte de coefficient optimale à la moyenne de Fréchet conditionnelle de la réponse. Il n'est pas simple d'étendre les deux modèles à un plus grand nombre de prédicteurs puisque les opérations correspondantes sont toutes deux directionnelles. C'est-à-dire qu'il n'est pas évident de mapper entre plusieurs espaces tangents de prédicteurs ou de pushforwarder parmi les mesures de prédicteurs. Ainsi, la méthode de centralisation des données proposée dans ce travail peut fournir une nouvelle piste pour traiter

plusieurs prédicteurs et réponses dans ce cadre de régression plus général.

1.2.2 Prévisibilité des nœud dans les signaux temporels du graphe

Pour mesurer la prévisibilité des nœuds d'un réseau, nous introduisons tout d'abord le problème de prédiction pour les observations du réseau $(\boldsymbol{x}_{it})_t$, $i \in \mathcal{N} := \{1, \ldots, N\}$, où nous reconstruisons hypothétiquement les observations au temps t sur un sous-ensemble I de noeuds, $\boldsymbol{x}_{it}$, $i \in I \subset \mathcal{N}$, étant donné les enregistrements dans le passé et à présent sur le reste des noeuds $(\boldsymbol{x}_{j\tau})_{j\tau}$, $j \in I^c, t - H \leqslant \tau \leqslant t$. Nous souhaitons comparer la prévisibilité évaluée par les deux méthodes de prédiction les plus populaires dans les domaines de la statistique et du traitement de signaux sur graphes : la régression ridge à noyau et le réseau de neurones. De plus, lorsque les arêtes des noeuds ne sont pas données, nous considérons la régression linéaire. Pour les deux méthodes de régression, nous obtenons le classement par l'adaptation gloutonne du problème de sélection de capteurs de I optimal, en mettant en relation les erreurs de prédiction des observations sur I avec les critères de sélection. Pour le réseau de neurones, nous mesurons la prévisibilité des noeuds de manière plus fine en attribuant un score à chaque noeud. Nous proposons d'adopter le schéma «dropout» pour faire varier les noeuds *éteints* dans I, qui permet en outre d'obtenir l'erreur de prédiction agrégée pour chaque noeud, en une seule formation. Dans la suite, nous présentons ces contributions dans un contexte plus technique. Cette section correspond au preprint Jiang et al. (2020).

1.2.2.1 Régression ridge à noyau des prédicteurs de temps et de nœuds

Tout d'abord, nous adaptons la régression ridge à noyau au problème de prédiction posé en régressant l'observation $x_{i\tau} \in \mathbb{R}$ sur les indices de nœud et de temps $i \in \mathcal{N}$ et $\tau \in \mathbb{Z}$. Nous proposons un design de noyau sur $\mathcal{N} \times \mathbb{Z}$, qui prend en compte la structure du graphe et définit la similarité des points temporels. La régression ridge à noyau approche alors la fonction de régression par les fonctions dans l'espace de Hilbert à noyau reproduisant (désignés sous l'acronyme issu du titre anglais RKHS, pour Reproducing Kernel Hilbert Space) induit $\mathcal{H}_k$, avec l'estimateur correspondant donné par

$$\widehat{f}_t = \min_{f \in \mathcal{H}_k} \sum_{i \in I^c, t-H \leqslant \tau \leqslant t} \|x_{i\tau} - f(i,\tau)\|_{\ell^2}^2 + \lambda \|f\|_{\mathcal{H}_k}^2. \tag{1.2.7}$$

Par la prédiction basée sur le RKHS, la prévisibilité d'un nœud provient de la similarité entre ses observations et les observations récentes sur son voisinage. Pour l'optimisation sous forme d'équation (1.2.7), le théorème de representer indique la forme explicite de la solution. Ainsi, étant donné un historique des observations du réseau, nous dérivons l'expression de l'erreur de prédiction totale $\sum_{i \in I} \sum_{H+1 \leqslant t \leqslant T} (x_{it} - \widehat{f}_t(i,t))^2/T$, qui dépend de l'auto-covariance empirique du processus $(x_{it})_t, i \in \mathcal{N}$ et de la matrice de Gram du noyau proposé. L'erreur dérivée donne un moyen de quantifier la prévisibilité des nœuds, qui dépend toutefois du choix de I. Nous proposons de nous appuyer sur le problème de sélection de capteurs en prenant comme critère l'erreur totale de prédiction, pour prendre en compte d'autres compositions et cardinalités possibles de I. Pour une cardinalité donnée $1 \leqslant p \leqslant N - 1$, le problème recherche

l'ensemble optimal $I^*(p)$ qui minimise un certain critère parmi tous les ensembles I de nœuds p. En particulier, puisque la stratégie gloutonne adaptée sélectionne les nœuds un par un, dont l'ordre forme les ensembles optimaux de cardinalité de 1 à $N-1$ de manière incrémentale. Nous proposons donc de classer les noeuds selon l'ordre produit par une telle stratégie gloutonne. Le classement résultant peut être interprété comme suit : pour chaque cardinalité p, nous attribuons aux noeuds de l'ensemble optimal $I^*(p)$ score 1, les autres noeuds score 0. Le total des scores pour tous les p de 1 à N forme le classement.

1.2.2.2 Faire varier l'ensemble éteint I par le dropout

Puisque les réseaux de neurones ont été largement adaptés aux problèmes de prédiction dans le domaine du traitement de signaux sur graphes, nous sommes particulièrement intéressés par l'évaluation de sa prédictibilité sur des nœuds{footnotePour le prédicteur de réseau, nous ne considérons que le problème de prédiction avec $H = 0$, car nous n'avons pas observé empiriquement l'amélioration de la reconstruction pour des H plus grands, lors de l'application du classement inféré à la sélection de capteurs.. Pour le prédicteur de réseau, nous pourrions définir le coût d'entraînement comme l'erreur de prédiction totale donnée dans la section 1.2.2.1 et le relier encore aux critères de sélection de capteurs de façon à reprendre la stratégie gloutonne. Cependant, cela conduit à $\mathcal{O}(N^2)$ fois l'entraînement du réseau, ce qui est beaucoup plus coûteux que les opérations matricielles. Par conséquent, nous concevons un schéma d'entraînement pour le prédicteur de réseau, qui varie la composition de I et donne donc les scores de tous les nœuds en un seul entraînement. L'idée principale est illustrée dans la Figure 1.5. En appliquant le dropout et le dropout à l'inverse sur l'entrée et la loss de batch, le réseau entraîné agrège les modèles de prédiction liés à différents I. Dans notre expérimentations, des milliers de compositions de I ont été échantillonnées pendant l'entraînement pour N de l'ordre de la centaine. Par conséquent, l'erreur de prédiction donnée par un tel réseau représente la prévisibilité globale d'un noeud, lorsqu'il est prédit dans différentes I. Nous proposons d'utiliser le réseau entraîné pour prédire les observations de l'ensemble de validation, et de prendre l'erreur de prédiction de chaque noeud comme son score, mesuré par la métrique R^2. La Figure 1.6 montre les scores dérivés pour les stations de vélos à Paris, calculés à partir d'un historique d'observation de 3533 heures, qui enregistre le ratio de vélos disponibles à chaque station et à chaque heure. Ces scores montrent le résultat intéressant que, les stations à haute prédictibilité forment plusieurs groupes géographiques. Nous nous concentrons sur les deux groupes. L'un suit l'itinéraire principal du Métro 1 et du RER A vers le quartier d'affaires central de La Défense (points blancs jaunes sur la rive droite de la Seine). L'autre se situe autour de la gare de Paris Montparnasse (points jaunes blancs sur la rive gauche). Ces deux groupes géographiques seront retrouvés au chapitre 5 par le modèle AR de Wasserstein proposé dans son graphe inféré sur le même jeu de données, où les formules de régression prédisant les stations à l'intérieur des groupes sont beaucoup plus significatives, avec une plus grande amplitude des coefficients non nuls.

Ces observations confirment l'efficacité des deux méthodes proposées.

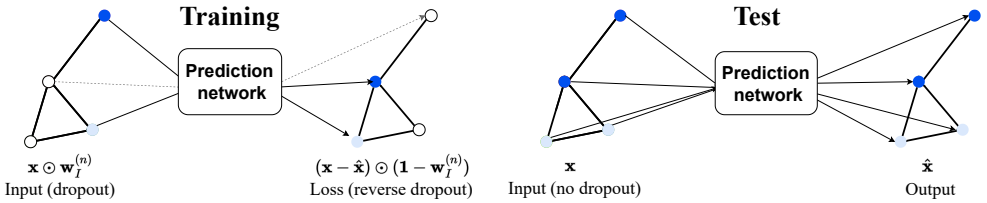

Figure 1.5: *Application du dropout à l'apprentissage de la prévisibilité.* À gauche, l'étape d'optimisation sur le $n$ième batch, où $\mathrm{x} = (x_i)_{i \in \mathcal{N}}$ représente l'entrée et $\mathbf{w}_I^{(n)}$ est le vecteur de dropout masquant l'entrée sur les noeuds de I. I est rééchantillonné à chaque batch n. Après la convergence de l'entraînement, le réseau est testé sur l'ensemble de validation comme illustré à droite, dont l'erreur de reconstruction de chaque noeud donne son score.

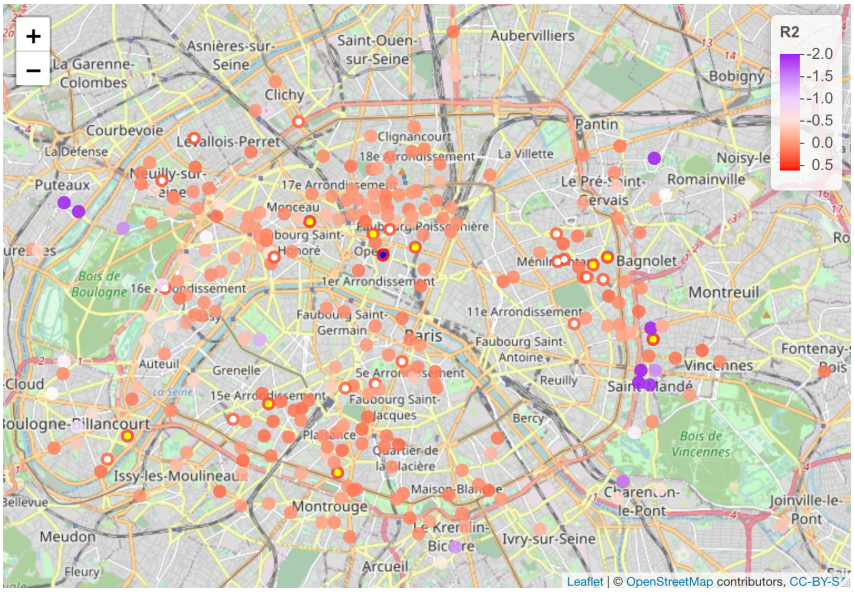

Figure 1.6: *Scores de prévisibilité des nœuds pour le réseau de velibs à Paris.* Les scores les plus élevés sont indiqués en rouge et les plus bas en violet. Les nœuds de top 1, 10 et 27 sont respectivement en bleu, jaune et blanc.

1.2.2.3 Variance partielle d'un processus stochastique multivarié

Enfin, nous aimerions dériver la méthode d'apprentissage pour la prévisibilité des nœuds, lorsque les informations sur la connectivité des nœuds dans un réseau ne sont pas disponibles. Nous pourrions d'abord appliquer une méthode de graph learning pour l'observation scalaire des noeuds, comme les modèles VAR, pour inférer les arêtes, puis réutiliser les méthodes proposées dans les sections 1.2.2.1 et 1.2.2.2. Cependant, puisque nous utilisons deux fois les mêmes données, le bruit des données sera amplifié et provoquera un biais plus important dans les modèles fittés. Nous avons donc l'intention de considérer les signaux temporels comme processus multivariés et de

nous appuyer sur des méthodes de prédiction sans graphe.

Une notion immédiate en statistique qui mesure la prévisibilité d'une variable aléatoire $\boldsymbol{x}_i$ étant donné d'autres variables $\boldsymbol{x}_j, j \neq i$ est le *variance partielle de* $\boldsymbol{x}_i$ *étant donné* $\boldsymbol{x}_j, j \neq i$, qui est essentiellement défini comme la variance du résidu de $\boldsymbol{x}_i$ prédit par $\boldsymbol{x}_j, j \neq i$ par la régression linéaire. Nous nous basons ensuite sur la variance partielle de $\boldsymbol{x}_{it}, i \in I \subset \mathcal{N}$ étant donné $(\boldsymbol{x}_{j\tau})_{j\tau}$, $j \in I^c, t - H \leqslant \tau \leqslant t$ pour quantifier la prévisibilité du noeud lorsque le graphe n'est pas donné. La variance partielle totale de $\boldsymbol{x}_{it}, i \in I$ étant donné $(\boldsymbol{x}_{j\tau})_{j\tau}$, $j \in I^c, t - H \leqslant \tau \leqslant t$ est une contrepartie de l'erreur de prédiction totale donnée dans la section 1.2.2.1. En particulier, lors de l'estimation de la variance partielle totale de $\boldsymbol{x}_{it}, i \in I$ étant donné $(\boldsymbol{x}_{j\tau})_{j\tau}, j \in I^c, t - H \leqslant \tau \leqslant t$ par l'auto-covariance empirique du processus $(\boldsymbol{x}_{it})_t, i \in \mathcal{N}$, l'estimateur s'écrit comme un cas spécial de l'erreur de prédiction totale dérivée dans la section 1.2.2.1, avec le kernel défini par ailleurs par l'auto-covariance empirique, et $\lambda = 0$. Nous considérons alors la variance partielle totale comme un autre critère pour le problème de sélection de capteurs, et nous nous appuyons sur la stratégie gloutonne pour obtenir le classement souhaité comme dans la Section 1.2.2.1.

1.2.2.4 Application à la sélection de capteurs

Étant donné la dérivation des classements, pour toute cardinalité $1 \leqslant p \leqslant N - 1$, les nœuds dans les top p peuvent également être considérés comme ensembles optimaux, sur lesquels les observations peuvent être prédites collectivement le mieux par les nœuds restants. Une telle stratégie de sélection prend en compte l'aspect de dépendance qui peut être estimé à partir des données, en plus de la structure du graphe. Dans les expérimentations du chapitre 6, nous montrons que la stratégie proposée présente un avantage dans la reconstruction des données manquantes par rapport aux méthodes d'échantillonnage standard dans le domaine du traitement de signaux sur graphes qui emploient uniquement la structure du graphe, comme Anis et al. (2016a); Puy et al. (2018).

1.3 Organization de la thèse

Chapitre 3 Nous présentons les préliminaires qui permettront de commencer l'analyse des chapitres 4 - 6.

Chapitre 4 Nous développons les approches pour les données de réseau d'observation vectorielle des nœuds. Nous proposons le modèle AR(1) matriciel et dérivons les procédures online pour inférer son graphe de caractérisation à partir des données séquentielles à tendance, en petite et grande dimension. Ce chapitre est lié au preprint Jiang et al. (2021).

Chapitre 5 Nous nous concentrons sur l'analyse des observations de nœuds distribués. Nous proposons le modèle AR(1) multivarié de Wasserstein et l'estimateur

sparse de ses coefficients, avec les résultats théoriques. Ce chapitre est lié au preprint Jiang (2022).

Chapitre 6 Ce chapitre est consacré au problème de l'apprentissage de la prévisibilité des nœuds à partir des données de réseau de type scalaire. Nous proposons des stratégies de classement pour les prédictions du modèle linéaire, du modèle RKHS et du réseau de neurones. Ce chapitre est lié au preprint Jiang et al. (2020).

Chapter 2

Introduction

2.1 Data recorded over a network of sensors

Data recorded over a network of sensors have become increasingly popular in recent years with applications in many areas such as traffic analysis (Crovella and Kolaczyk, 2003; Yao et al., 2018; Fang et al., 2019), brain network analysis (Huang et al., 2018; Wang et al., 2020), and meteorology (Handcock and Wallis, 1994; Mei and Moura, 2016; Xu et al., 2018). The sensor networks can also represent an abstract space, which are not related to physical entities, such as a social network (Tabassum et al., 2018), citation network (Liu et al., 2019), or semantic network (Lake and Tenenbaum, 2010; Sarica et al., 2020). These sensors often record a sequence of observations along time at a regular frequency, which corresponds to the evolution of traffic or weather, for example. Such observations are therefore endowed with both spatial and temporal nature. Apart from the spatio-temporal aspect, the nodal observation from each sensor can also exhibit rich data representation. Three common data types are scalar, vector, and distribution, which are therefore considered by this thesis. In the following, we give the brief presentation for each of the three data structures.

Scalar is the most usual type for node observation in the literature (Kolaczyk, 2009; Shuman et al., 2013). In this case, for a network of N nodes, we observe a value $\boldsymbol{x}_{it} \in \mathbb{R}^*$ at each node $i = 1, ..., N$ and time instant $t \in \mathbb{Z}$. The popularity of the scalar-type comes from the fact that the observations at each time can be represented by the vector $\mathbf{x}_t := (\boldsymbol{x}_{it})_{i=1}^N$. One can then start off with existing tools, for instance those from multivariate time series analysis (Lütkepohl, 2005; Brockwell and Davis, 2009; Neusser, 2016), to consider the temporal dependency with the spatial structure, for example, Bach and Jordan (2004); Perraudin and Vandergheynst (2017).

Secondly, the vectorial observation refers to the case where a vector $\mathbf{x}_{it} = (\boldsymbol{x}_{it}^f)_{f=1}^F \in \mathbb{R}^F$ of F features is observed at each node and time instant. Because the observations at each time can be represented by the matrix $\mathbf{X}_t := (\mathbf{x}_{1t}, \ldots, \mathbf{x}_{Nt}) \in \mathbb{R}^{N \times F}$ with the row and column corresponding to the spatial dimension and the additional

*The value can also be complex. However, we focus on real observations for all the cases throughout this thesis.

feature dimension. Time-wise, the total observation is matrix-variate. Such matrix-variate observations are also common in applications. Some examples include the observation of measurement epoch × channel for each subject in EEG analysis (Zhou, 2014; Wang et al., 2020), and the observations of gene × tissue in gene expression data analysis (Yin and Li, 2012).

Lastly, we consider the setting of distributional observation. At each node $i = 1, ..., N$ and time instant $t \in \mathbb{Z}$, we observe a distribution, namely, a probability measure $\boldsymbol{\mu}_{it} \in \mathcal{P}(\Omega)$ supported on an interval Ω of $\mathbb{R}$. In practice, a distribution can not be directly observed and instead the available data consist of i.i.d. samples that are generated by it. Thus, for model fitting, we need to primarily retrieve the distributions, more precisely their representation such as density, quantile function, or cumulative distribution function, from the samples by numeric methods, such as the local linear regression for smooth densities (Fan and Gijbels, 2018).

Distributional data have been playing important roles in many scientific fields. A pertinent example is the analysis of the indicator distributions supported over age intervals, such as mortality and fertility (Mazzuco and Scarpa, 2015; Shang and Haberman, 2020), observed across different countries over years in demographic studies. Another important example is the analysis of daily stock return distributions from financial time series (Kokoszka et al., 2019; Zhang et al., 2021). Other examples include the distributions of correlations between pairs of voxels within brain regions (Petersen and Müller, 2016) and the house price distributions overs months (Zhu and Müller, 2021). Nevertheless, despite the long existence of the distributional data, the development of their tailored tools is only a recently emerging field of research.

In Figures 2.1 and 2.2, we demonstrate the three types of observations with real data sets.

The general goal of this thesis is to develop the approaches for the network data of these diverse forms, in order to explore and represent the spatial dependency of the N features indexed by the nodes. In Section 2.2, we present the main problems considered by this thesis, as well as the main contributions achieved for each problem.

2.2 Problems and main contributions

The studies in this thesis are carried on around two main problems: identification of the dependency structure which governs the stochastic process over a network, and understanding the node predictability in the time-dependent graph signals.

The first problem can be related to the research domain of *graph learning* (Dong et al., 2019), where the central purpose is that, given the observations of multiple features represented by random variables or processes, we would like to build or infer the cross-feature relationship that takes the form of a graph, with the features termed as nodes. In this framework, when the node observation is scalar-type, many approaches have been proposed to learn such graphs. Especially, for the scalar random processes, the vector auto-regressive (VAR) models have been widely adapted to infer the structure of one pertinent relationship, namely Granger causality. The resulting graph is referred to as causal graph. Our interests are then to extend the VAR models

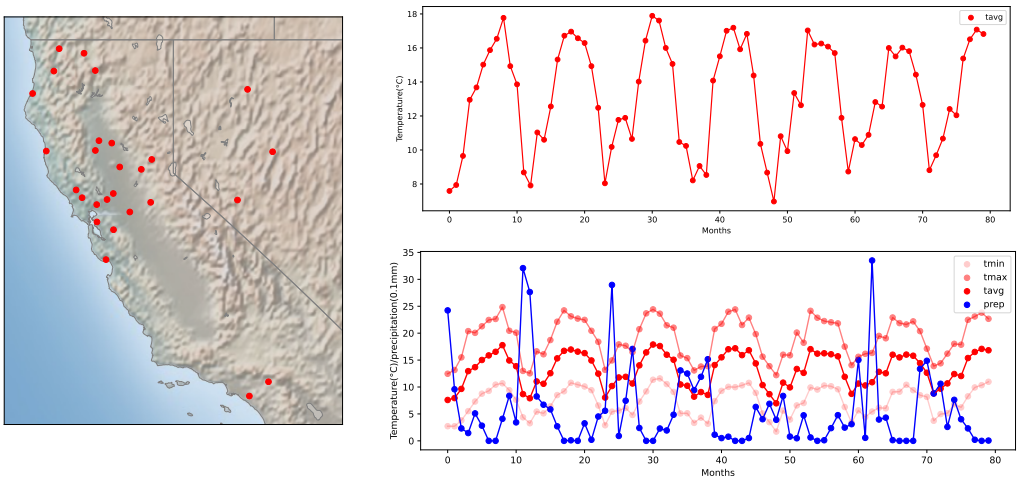

Figure 2.1: *Monthly climatological records of weather stations in California.* On the left is the network of weather stations in California. On the upper right are demonstrated the *scalar* observations on a certain station (sensor/node) i, where a value $\boldsymbol{x}_{it} \in \mathbb{R}$ of average temperature is recorded at each time t at i, leading to a scalar time series. On the lower right are demonstrated the *vectorial* observations on a certain station i, where a vector $\mathbf{x}_{it} \in \mathbb{R}^4$ of min/max/avg temperature and precipitation is recorded per time t at i, leading to 4 scalar time series.

to the matrix-variate and multivariate distributional AR models to serve the purpose of graph learning. The derived models respond to the demand of developing the tailored models, as the matrix-variate and distributional time series become more popular, which have not been widely studied yet in the literature.

The second problem is motivated by the fact that, in many applications, the observations recorded over a network exhibit strong cross-node dependency, which makes the data observed on a subset of nodes highly-predictable by the data on the remaining nodes. These nodes cause the redundancy in the network data, so that we can stop recording for the sake of data storage, for instance. We are then interested in ranking the predictability for the nodes in a network, given a history of network observations, possibly taking into account additionally the connectivity of nodes if available. As an application, the derived ranking can be used in sensor selection, which is an important topic in the graph signal processing domain, see for example Joshi and Boyd (2009); Sakiyama et al. (2019b).

In Subsections 2.2.1 and 2.2.2, we develop the mathematical frameworks of the two problems with non-exhaustive literature reviews, which allow to present the main contributions of this thesis. Throughout this manuscript, we use the bold notation to distinguish random objects in the models discussed in the context from the constants.

2.2.1 Spatial dependency structure and auto-regressive (AR) models

This section corresponds to the preprints Jiang et al. (2021); Jiang (2022).

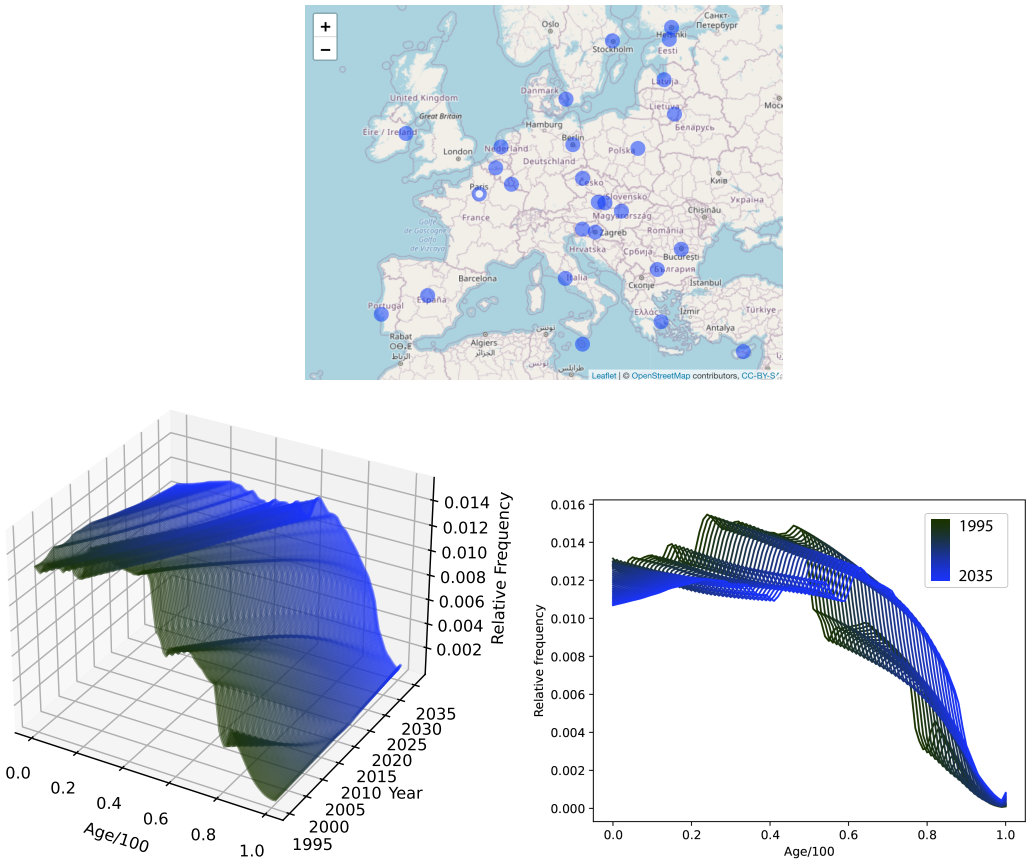

Figure 2.2: *Annual records of age distributions of EU countries.* On the top is the network of the 27 countries in the European union. At the bottom are the *distributional* observations recorded at i = France along time. A distribution of age $\boldsymbol{\mu}_{it} \in \mathcal{P}([0,1])$ is recorded at each time t. On the lower left, we visualize the resulting univariate distributional time series with a surface in the coordinate system of Age $\times$ Year $\times$ Relative frequency. The raw data in this plot consist in 40 annual distributions. We complete them with interpolated samples to draw the surface. On the lower right, we show the projection of the raw time series onto the Age $\times$ Relative frequency plane. We can see that the population is aging along time.

2.2.1.1 VAR models characterized by the causal graphs

In this section, we recall the classical VAR models and their applications in graph learning from scalar-valued nodal observations in literature. In the following sections, we will present the proposed extensions.

In multivariate time series analysis, the vector auto-regressive model refers to the stochastic difference equation:

$$\mathbf{x}_t = \mathrm{b} + \sum_{l=1}^{p} A^l \mathbf{x}_{t-l} + \mathbf{z}_t, \quad t \in \mathbb{Z},$$

where $\mathbf{z}_t \sim \mathrm{WN}(0, \Sigma)$ is a white noise process with variance Σ, and $\mathrm{b} \in \mathbb{R}^N, A^l \in \mathbb{R}^{N \times N}$ are the coefficients. As stated earlier, VAR models are widely applied to infer the Granger causality structure of N univariate processes $(\boldsymbol{x}_{it})_t$ for $i = 1, \ldots, N$. The Granger causality is defined pairwise: $(\boldsymbol{x}_{it})_t$ is said to Granger cause $(\boldsymbol{x}_{jt})_t$, $j \neq i$ if $(\boldsymbol{x}_{jt})_t$ can be predicted more efficiently with the knowledge of $(\boldsymbol{x}_{it})_t$ in the past and present taken into account. More technical definition see Definition 3.2.4. The causal graph then refers to such a graph where each node represents a univariate time series, and the edges represent Granger causality. If the processes generated by a stationary VAR(p) model, time series $(\boldsymbol{x}_{it})_t$ does not cause $(\boldsymbol{x}_{jt})_t$ if and only if all the ji-th entries of the true coefficient matrices $A^l_{ji} = 0$, $l = 1, \ldots, p$, (Lütkepohl, 2005, Corollary 2.2.1). Thus, we can retrieve the graph topology from the common sparsity structure in A^l. In low-dimensional regime, this structure can be identified through Wald test, which tests the linear constraints for the coefficients.

In the high dimensional regime, the inference of the exact Granger causal graph is mainly considered in Bolstad et al. (2011); Zaman et al. (2020). Bolstad et al. (2011) propose to consider the group lasso penalty, $\lambda \sum_{i \neq j} \|(A^1_{ij}, \ldots, A^p_{ij})\|_{\ell_2}$, to the usual least squares problem of VAR(p) models, in order to infer the common sparsity structure of coefficient matrices $A^l, l = 1, \ldots, p$. Zaman et al. (2020) develop the online procedure for this estimation problem. Mei and Moura (2016) define a variant of VAR model, where the sparse structure of coefficients A^l does not directly equal the graph topology, but the topology of l-hop neighbourhoods [†]. More specifically, they suppose that $A^l = c_{l0} I + c_{l1} W + \ldots + c_{ll} W^l$, where W is the adjacency matrix to infer, and I is the identity matrix. Such models can thus capture the influence from more nodes. The estimation of the underlying adjacency matrix relies on the Lasso penalty to promote the sparsity.

We firstly consider the extension of the VAR(1) model to the matrix-variate process $(\boldsymbol{X}_t)_{t \in \mathbb{Z}} \in \mathbb{R}^{N \times F}$. To this end, we may apply straightforwardly the vector models to the vectorized representation $\mathrm{vec}(\boldsymbol{X}_t)$. However, this is not favorable. On one hand, dealing with the data from separate dimensions indifferently can ignore, and hence waste, the information of the intrinsic structure of data. In terms of technical complexity, it leads to the quadratical growth of the number of parameters in the model, requiring then large number of samples for robust estimation, which is not easy to satisfy in practice. Thus the modelling of the matrix-variate data requires additional techniques, which we will present in Section 2.2.1.2.

[†]For a node, its neighbours are in the 1-hop neighbourhood of the node. All the neighbours of its neighbours are in the 2-hop neighbourhood of the node, so far and so forth.

2.2.1.2 Kronecker sum and Cartesian product graph

In the literature, to extend vector models to matrix-variate observations (more generally tensor-variate), a popular practice is to apply the vector models onto vectorized data, then impose certain structures on parameter matrices which encode information on data dimensions. The most considered structures are Kronecker sum (KS) or/and Kronecker product (KP). For example, to extend multivariate Gaussian distribution to matrix Gaussian distribution, Gupta and Nagar (2018) first vectorize matrix-valued random variable $\mathbf{X} \in \mathbb{R}^{N\times F}$, and suppose that $\mathrm{vec}(\mathbf{X}) \sim \mathcal{N}(\mu, \Omega)$, then impose the KP structure on the variance matrix: $\Omega = V \otimes U$, where $U \in \mathbb{R}^{N\times N}, V \in \mathbb{R}^{F\times F}$. The two sub-variance matrices U and V are separately associated with the row and column dimensions of the data matrix. Besides, Kalaitzis et al. (2013) propose a different matrix Gaussian model: $\mathrm{vec}(\mathbf{X}) \sim \mathcal{N}(\mu, P^{-1})$, where $P = \Psi \oplus \Theta$, and $\Psi \in \mathbb{R}^{N\times N}, \Theta \in \mathbb{R}^{F\times F}$. That is they impose a KS structure on the precision matrix[‡]. Since the KS structure in an adjacency matrix corresponds to the Cartesian product of subgraphs, the embedding of KS structure leads to an interpretable Gaussian graphical model for matrix-variate observations, where the total conditional dependence structure factorizes into the separate sub-structures of the raw and column dimensions.

Therefore, to extend VAR(1) model to matrix-variate process $(\boldsymbol{X}_t)_t$, we can apply the classical VAR(1) onto the vectorized process $(\mathrm{vec}\boldsymbol{X}_t)_t$, then impose KS or KP structure on the coefficient matrix A. Since the sparsity structure of A encodes the topology of causal graph of NF processes $(\boldsymbol{x}_{it}^f)_t, i = 1, \ldots, N, f = 1, \ldots, F$, KS and KP both imply that, this total graph can factorize into two subgraphs, but in different ways. To explain this difference, we recall the definition of the Kronecker sum. Let $A_{\mathrm{F}} \in \mathbb{R}^{F\times F}, A_{\mathrm{N}} \in \mathbb{R}^{N\times N}$ be two square matrices, and let I_k denote the $k \times k$ identity matrix. The Kronecker sum between A_{F} and A_{N} is defined as

$$A_{\mathrm{F}} \oplus A_{\mathrm{N}} = A_{\mathrm{F}} \otimes I_N + I_F \otimes A_{\mathrm{N}},$$

where $\otimes$ is the Kronecker product. As mentioned before, the KP and KS structures in adjacency matrices imply the corresponding graphs are the product of the component graphs. More specifically, when $A_{\mathrm{F}}, A_{\mathrm{N}}$ are the adjacency matrices of two graphs $\mathcal{G}_{\mathrm{F}}, \mathcal{G}_{\mathrm{N}}$, the KP $A_{\mathrm{F}} \otimes A_{\mathrm{N}}$ and KS $A_{\mathrm{F}} \oplus A_{\mathrm{N}}$ are respectively the adjacency matrices of their tensor product graph $\mathcal{G}_{\mathrm{N}} \times \mathcal{G}_{\mathrm{F}}$ and Cartesian product graph $\mathcal{G}_{\mathrm{N}} \square \mathcal{G}_{\mathrm{F}}$ (Sandryhaila and Moura, 2014a). We illustrate these two graph products in Figure 2.3. For the formal definitions of Cartesian and tensor products of graphs, we refer to Hammack et al. (2011); Chen and Chen (2015); Imrich and Peterin (2018).

Figure 2.3 shows that, the product graphs led by the KP and KS differ greatly. For example, the lattice-like structure of the Cartesian product preserves the subgraphs in all sections of both dimensions. By contrast, the tensor product focuses on the cross-dimensional connection, yet abandoning the intra-dimensional dependency. This later property actually refers to, in the Gaussian process literature, the cancellation of inter-task transfer, see for example, Bonilla et al. (2007, Section 2.3). Therefore when

[‡]Reparametrized versions of Gaussian distributions with respect to their precision matrices are used in the graph learning domain for Gaussian i.i.d. data, because the sparsity structure of the precision matrix encodes the conditional independence relationship of random variables.

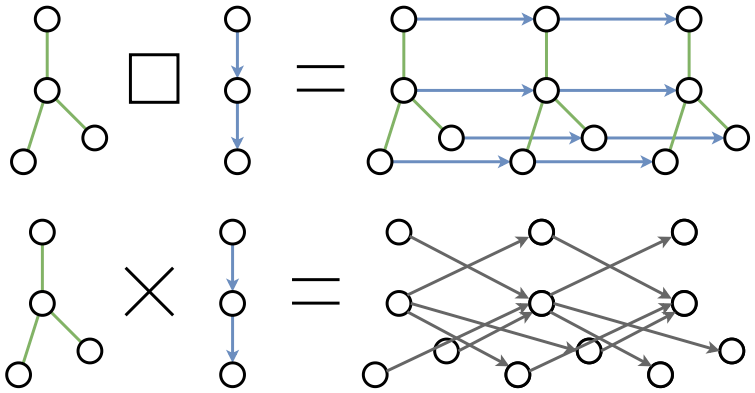

Figure 2.3: *Comparison of the Cartesian and the tensor products of graphs.* The node set of both product graphs is the Cartesian product of the components' node sets, yet follows the different adjacencies. The example is based on Sandryhaila and Moura (2014a, Figure 2).

the nodes represent $(\boldsymbol{x}_{it}^f)_t, i = 1, \ldots, N, f = 1, \ldots, F$, imposing KP structure (Chen et al., 2021a) implies assuming no causality dependencies among $(\boldsymbol{x}_{it}^f)_t, i = 1, \ldots, N$ for each f fixed, which represent the observations of the feature f at different nodes across the network. By contrast, the coefficient matrices endowed with the KS structure is able to take such dependencies into account in the inference, which are in effect present in many applications.

2.2.1.3 Causal product graphs and matrix-variate AR(1) models

The first contribution lies in the extension of vector AR(1) model to the matrix-variate time series, $\mathbf{X}_t \in \mathbb{R}^{N \times F}$, $t \in \mathbb{Z}$, by imposing[§] the KS structure on the coefficient matrix A of the VAR(1) model applied to $\text{vec}(\mathbf{X}_t)$:

$$\text{offd}(A) = A_{\text{F}} \oplus A_{\text{N}}, \tag{2.2.1}$$

where $\text{offd}(A)$ is the matrix of the same size as A, whose off-diagonal part is identical to that of A, while its diagonal elements are equal to zero. As stated in Section 2.2.1.1, for a stationary VAR(1), the sparsity structure of the coefficient defines the causal graph of the NF components of $\text{vec}(\mathbf{X}_t)$. By embedding of the KS, we then assume this total causal graph factorizes as the Cartesian product into the spatial graph $\mathcal{G}_{\text{N}}$ and the feature graph $\mathcal{G}_{\text{F}}$, where $\mathcal{G}_{\text{N}}$ is preserved in each feature, and $\mathcal{G}_{\text{F}}$ preserved in each node. A related work which also extends the VAR(1) model to the matrix-variate AR(1) model is Chen et al. (2021a), where they propose to impose the KP structure onto the adjacency matrices. However, as we indicated in Section 2.2.1.2, the KP is not able to capture the intra-dimensional causality.

Given the samples $\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_t$, we then develop two approaches to estimate $A_{\text{N}}, A_{\text{F}}$ and identify the sparsity structure, in order to infer the graph topology. The

[§]We only show the key structure in this summary. The complete structure assumed for A see Chapter 4.

approaches are designed, respectively, for the low and high dimensional learning. We especially consider a general learning framework where the partial sparsity is pursued in the estimation of only A_{N}. This is motivated by merely a very small number of features F usually present in applications. Thus, the feature graph can be reasonably assumed fully-connected. On the other hand, since the partial sparsity constraint is also a technically more complicated case for the proposed high dimensional learning method, given its corresponding resolution, the adaption to the case of fully sparsity does not require novel techniques.

In low dimension, we propose the projected OLS (equivalently GLS) estimator to benefit the asymptotic properties of the classical estimators. More specifically, we first construct an orthogonal basis $(U_k)_k$ of the space $\mathcal{K}_{\mathcal{G}}$, which is a linear space formed by all the $\mathbb{R}^{NF\times NF}$ matrices of the structure (2.2.1). Relying on $(U_k)_k$, we are able to express, $\mathrm{Proj}_{\mathcal{G}}$, the projection onto $\mathcal{K}_{\mathcal{G}}$ explicitly, which furthermore defines the estimators $\widehat{\mathbf{A}_{\mathrm{N},t}}$ and $\widehat{\mathbf{A}_{\mathrm{F},t}}$ through the KS. Such $\widehat{\mathbf{A}_{\mathrm{N},t}}$ is element-wise linear function of the classical OLS estimator of A. By applying the Cramer-Wold theorem on the CLT of the classical OLS estimator, we then derive the CLT for $\widehat{\mathbf{A}_{\mathrm{N},t}}$. This CLT moreover allows the establishment of Wald test to identify the sparsity structure of A_{N}.

2.2.1.4 Novel Lasso type: Structured matrix-variate Lasso and its Homotopy Algorithms

As discussed at the end of Section 2.2.1.1, a common practice in the literature to identify the sparsity structure of VAR coefficients in high dimensional regime is to adopt Lasso estimators. The one used in Bolstad et al. (2011); Zaman et al. (2020) is defined as the minimizer of Lasso problem (2.2.2) in the VAR(1) case.

$$\min_{A} \frac{1}{2t}\sum_{\tau=1}^{t} \|\mathbf{x}_\tau - A\mathbf{x}_{\tau-1}\|_{\ell_2}^2 + \lambda_t \|A\|_{\ell_1}\,, \tag{2.2.2}$$

where $\mathbf{x}_\tau$ is a vector of sample, which can be taken as $\mathrm{vec}(\boldsymbol{X}_\tau)$ for example. Lasso (2.2.2) is the most standard Lasso in literature (Hastie et al., 2009, Section 3.4.2). A wide variety of frameworks from convex analysis and optimization have been adapted to compute its solutions for different scenarios, for example, coordinate descent (Friedman et al., 2010), proximal gradient methods (Beck and Teboulle, 2009), and a more Lasso-specific technique least angle regression (Efron et al., 2004). However, Lasso (2.2.2) is not able to estimate the structured A with the sparse component A_{N}. Therefore motivated by the estimation, we propose the novel Lasso type problem (2.2.3)

$$\mathbf{A}(t,\lambda_t) = \operatorname*{arg\,min}_{A\in\mathcal{K}_{\mathcal{G}}} \frac{1}{2t}\sum_{\tau=1}^{t} \|\mathbf{x}_\tau - A\mathbf{x}_{\tau-1}\|_{\ell_2}^2 + \lambda_t F \|A_{\mathrm{N}}\|_{\ell_1}\,, \tag{2.2.3}$$

The ordinary resolution of Lasso (2.2.3) can be done by applying for example the proximal gradient descent (Parikh and Boyd, 2014). In the algorithm framework, the structure constraint and the partial sparsity do not pose the additional difficulties, since only the gradient with respect to $\mathbb{R}^{NF\times NF}$ is calculated in the forward step.

Then the backward step amounts to a standard Lasso after projecting the gradient onto $\mathcal{K}_{\mathcal{G}}$, as shown in Equation (2.2.4).

$$\begin{aligned}
\mathbf{A}^{k+1} &= \operatorname{prox}(\mathbf{A}^k - \eta^k \nabla f(\mathbf{A}^k)), \\
&= \arg\min_{A \in \mathcal{K}_{\mathcal{G}}} \frac{1}{2\eta^k} \left\| A - \left(\mathbf{A}^k - \eta^k \nabla f(\mathbf{A}^k)\right) \right\|_{\ell_2}^2 + \lambda_t F \left\| A_{\mathrm{N}} \right\|_{\ell_1} \\
&= \arg\min_{A \in \mathcal{K}_{\mathcal{G}}} \frac{1}{2\eta^k} \left\| A - \operatorname{Proj}_{\mathcal{G}} \left(\mathbf{A}^k - \eta^k \nabla f(\mathbf{A}^k)\right) \right\|_{\ell_2}^2 + \lambda_t F \left\| A_{\mathrm{N}} \right\|_{\ell_1} \\
&\iff \begin{cases}
\mathbf{A}_{\mathrm{N}}^{k+1} = \arg\min_{A_{\mathrm{N}}} \left\| A_{\mathrm{N}} - \operatorname{Proj}_{\mathcal{G}_{\mathrm{N}}} \left(\mathbf{A}^k - \eta^k \nabla f(\mathbf{A}^k)\right) \right\|_{\ell_2}^2 + 2\eta^k \lambda_t \left\| A_{\mathrm{N}} \right\|_{\ell_1}, \\
\mathbf{A}_{\mathrm{F}}^{k+1} = \operatorname{Proj}_{\mathcal{G}_{\mathrm{F}}} \left(\mathbf{A}^k - \eta^k \nabla f(\mathbf{A}^k)\right), \\
\operatorname{diag}(\mathbf{A}^{k+1}) = \operatorname{Proj}_{\mathrm{D}} \left(\mathbf{A}^k - \eta^k \nabla f(\mathbf{A}^k)\right),
\end{cases}
\end{aligned} \tag{2.2.4}$$

where $\nabla f(\mathbf{A}^k) = \mathbf{A}^k \widehat{\mathbf{\Gamma}}_t(0) - \widehat{\mathbf{\Gamma}}_t(1)$, we denote $\mathbf{A}^{k+1}(t, \lambda_t)$ by $\mathbf{A}^{k+1}$ to avoid the heavy notation.

At this point, we focus on providing the algorithms to quickly update the previous solutions for the change in the hyperparameter value or the data term. To benefit the previous solutions of a problem, one needs to consider the specific methods. For standard Lasso (2.2.2), the framework of homotopy continuation methods (Osborne et al., 2000) has been explored (Malioutov et al., 2005; Garrigues and Ghaoui, 2008) to calculate the fast updating. Since the homotopy algorithm is derived from the optimality condition, which is with respect to the matrices in $\mathcal{K}_{\mathcal{G}}$ for Lasso (2.2.3), requiring to consider the gradient with the structure, thus the existing homotopy algorithms for Lasso (2.2.2) are not applicable. Therefore in our work, we first calculate the optimality condition of Lasso (2.2.3), based on the expression of projection onto $\mathcal{K}_{\mathcal{G}}$. Then we derive the two Homotopy algorithms, respectively, for the updating paths $\mathbf{A}(t, \lambda_1) \to \mathbf{A}(t, \lambda_2)$ and $\mathbf{A}(t, \lambda_2) \to \mathbf{A}(t+1, \lambda_2)$, together with an adaptive tuning procedure for the regularization hyperparameter at the arrival of new observation $\mathbf{x}_{t+1}$. The derivations do not depend on the specific structure constraint, nor on the particular sparsity regularization, and thus can be applied to other linear structure space $\mathcal{K}_{\mathcal{G}}$ and the design of sparsity.

2.2.1.5 Online graph and trend learning

In the applications where the data storage is limited and fast inference upon the sequentially arriving data is demanded, the ordinary resolution which processes the entire data set at once fails. Thus the tailored algorithms which fulfill these two conditions are needed. This class of algorithm are referred to as *online*. By contrast, the ordinary resolution is referred to as *off-line*.

For the online resolution of Lasso (2.2.3), the previously derived homotopy algorithms constitute the pertinent method, when performing the three steps in the order:

$$\begin{aligned}
&\text{Step 1}: \ \lambda_t \to \lambda_{t+1}, \quad \text{Step 2}: \ \mathbf{A}(t, \lambda_t) \to \mathbf{A}(t, \lambda_{t+1}), \\
&\text{Step 3}: \ \mathbf{A}(t, \lambda_{t+1}) \to \mathbf{A}(t+1, \lambda_{t+1}),
\end{aligned} \tag{2.2.5}$$

However, when passing to the online inference, the formulation of the problem

(2.2.3) can not fully adapt, due to the data term $\frac{1}{t}\sum_{\tau=1}^{t}\|\mathbf{x}_\tau - A\mathbf{x}_{\tau-1}\|_{\ell_2}^2$. The data term implies that the time series $(\mathbf{x}_\tau)_\tau$, $\tau \in \mathbb{Z}$ is supposed to admit a time-invariant mean, which is zero. Nevertheless, when the $(\mathbf{x}_\tau)_\tau$ take raw data, this sample assumption is very often not true. Instead, the raw time series is usually present with the trend, that is a time-variant mean function. Thus, in off-line learning, a *detrend* step is needed, which approximates the trend function using the entire data set, then removes it from the raw data. Then the actual time series fitted to the model will be closer to the model assumptions. However, since the principle of online learning does not require the presence of all data, such pre-processing step is forbidden. Thus, we need to consider the trend as the explicit parameters additional to the graph parameters $A_{\mathrm{N}}, A_{\mathrm{F}}$ in the online model.

The present work focuses primarily on a particular type of trend, that is the periodic trend, which is frequently encountered in practice. For example, an annual recurrence for every 12 months can be found in many data sets, recorded monthly over years. From the aspect of modelling, the periodic trend is the mean function:

$$\mathbb{E}\mathbf{x}_\tau = \mathrm{b}_m^0,\ m = 0, ..., M-1,\ m = \tau \bmod M,$$

where M is the length of period. We thus propose to reformulate Lasso (2.2.3) by incorporating the periodic trend b_m^0, $m = 0, \ldots, M-1$, as Equation (2.2.6).

$$\mathbf{A}(t,\lambda_t), \mathbf{b}_m^0(t,\lambda_t) = \underset{A\in\mathcal{K}_{\mathcal{G}}, \mathrm{b}_m^0}{\arg\min} \frac{1}{2t}\sum_{m=0}^{M-1}\sum_{\tau\in I_{m,t}} \|(\mathbf{x}_\tau - \mathrm{b}_m^0) - A(\mathbf{x}_{\tau-1} - \mathrm{b}_{m-1}^0)\|_{\ell_2}^2 + \lambda_t F\|A_{\mathrm{N}}\|_{\ell_1}, \tag{2.2.6}$$

where $I_{m,t} = \{\tau = 1, ..., t : \tau \bmod M = m\}$. Note that b_{-1}^0 denotes b_{M-1}^0. The underlying sample model becomes

$$\mathbf{x}_{t-1} - \mathrm{b}_m^0 = A(\mathbf{x}_{t-1} - \mathrm{b}_{m-1}^0) + \mathbf{z}_t,\ A \in \mathcal{K}_{\mathcal{G}},\ m = t \bmod^{\P} M,$$

which is an augmented VAR(1) model with periodic trend.

We then adapt the algorithms in Pipeline (2.2.5) to Problem (2.2.6), which finally allows the online graph and trend learning from the matrix-variate time series.

2.2.1.6 Wasserstein statistics for distributional data

Secondly, we would like to extend the VAR(1) model for a collection of N time-dependent probability measures $(\boldsymbol{\mu}_t^i)_{t\in\mathbb{Z}}, i = 1, \ldots, N$ (example of such data see Section 2.1), which can additionally identify and represent the significant dependency links among the processes in a directed weight graph of N nodes. Instead of a vector, now the node feature becomes a distribution; to model such distributional data, we need to adopt more advanced tools.

Since distributions can be characterized by certain functions, such as densities, quantile functions, and cumulative distribution functions, then to analyze the distributional time series, one may turn to study one of its functional representations with the tools from functional time series analysis (Bosq, 2000). However, due to

[¶] The modulo of a negative integer is defined by the positive reminder in this case, for example, $-1 \bmod M = M - 1$.

their nonlinear constraints, such as monotonicity and positivity, the representing functions of distributions do not constitute linear spaces. Consequently, basic notions in standard VAR models, such as additivity and scalar multiplication, do not adapt, in a straightforward manner. This causes models devised for random elements of a Hilbert space to fail. One existing approach is to map the densities of distributions to unconstrained functions in the Hilbert space by the log quantile density (LQD) transformation (Petersen and Müller, 2016), and then apply the functional tools (Kokoszka et al., 2019). However LQD does not take into account the geometry of the distribution space, thus, it can lead to deformations in the distance. Recent approaches consider such geometry by adopting the Wasserstein metric (Bigot et al., 2017; Panaretos and Zemel, 2016; Petersen and Müller, 2019b). We therefore establish the time series model in the Wasserstein space. In the following, we present the main contribution in this line of work. For the technical introduction of the statistics in Wasserstein space, we refer to Section 3.3.

2.2.1.7 Wasserstein multivariate AR(1) model and the N-simplex constraint

There are also other works which have considered the distributional time series models. They so far mainly focus on the univariate case, that is when $N = 1$. Generally, the multivariate distributional models are still barely developed. Note that in the distributional case, two multivariate notions are relevant, one refers to the dimension of the measure support, while the other one refers to the numbers of measures. Models associated to both notions are undeveloped. In this manuscript, the word *multivariate* refers to the second notion.

We refer to the recent works Chen et al. (2021b); Zhang et al. (2021); Zhu and Müller (2021), which develop the univariate distributional AR model in the Wasserstein space $\mathcal{W}_2(\mathbb{R})$. They propose to map the random distributions to the Logarithmic image of Wasserstein space at their presumed common Fréchet mean. Then they construct the univariate functional time series models in terms of the logarithmic maps. The common Fréchet mean is the ideal reference distribution in the sense that it cancels out the expectations of the logarithmic maps in the regression formula. However, for multivariate time series, different processes can not have one common Fréchet mean. Moreover such ideal reference measure is very unlikely to exist (will be explained in Chapter 5). On the other hand, an extra intercept in such functional regression model will cause great difficulties, when one wants to retain the regression model especially the predictions in the logarithmic image of Wasserstein space. Thus to deal with the unequal process means, we are motivated by the equivalent formulation of VAR(1) model built on the centered series which thus does not include the intercept term (details see Equation (3.2.5) in the preliminaries section)

$$\mathbf{x}_t - \mathrm{u} = A(\mathbf{x}_{t-1} - \mathrm{u}) + \mathbf{z}_t.$$

We firstly propose a way to center all the raw distributions by their process means (Fréchet) so that their means turn to be the uniform distribution namely the Lebesgue measure. Secondly we build the model on the centered data $\widetilde{\boldsymbol{\mu}}_t^i$, $i = 1, \ldots, N$, $t \in \mathbb{Z}$. For each i, we define the key regression relationship for the response $\widetilde{\boldsymbol{\mu}}_t^i$ and the

predictors $(\widetilde{\boldsymbol{\mu}}_{t-1}^j)_j$, $j = 1, \ldots, N$ in the tangent space of the Lebesgue measure, as the linear combination of the logarithmic maps of $(\widetilde{\boldsymbol{\mu}}_{t-1}^j)_j$, weighted by coefficients $(A_{ij})_j$. The response $\widetilde{\boldsymbol{\mu}}_t^i$ equals the exponential map of the linear combination, pushforwarded by a random distortion function. Therefore the total coefficients of the regression system defines a matrix $A = (A_{ij})_{ij}$ as in VAR(1) model. For general functional regression models with function response, the coefficients can be concurrent $A_{ij}(\cdot)$ or the most general a surface $A_{ij}(\cdot, \cdot)$, see Wang et al. (2015, Equations (14) and (15), respectively). Nevertheless, we purpose to use matrix form for the regression coefficients in order to represent the graph structure directly.

Since the exponential map is not injective, the model proposed above has an identifiability problem. Thus we furthermore retain the model in the logarithmic image by adding the N-simplex assumption A2 to the coefficients (see below), given the injectivity of the restricted exponential map on the logarithmic image and the convexity of the logarithmic image. Actually, when falling out of the logarithmic image, the calculation in the estimation problem is not tractable.

Assumption A2. $\sum_{j=1}^N A_{ij} \leqslant 1$ and $0 \leqslant A_{ij} \leqslant 1$.

We then show that the iterated random function system (Wu and Shao, 2004) associated to the proposed time series model admits a unique solution, which is moreover stationary as the functional process in the Hilbert space of N-tuple of quantile functions endowed with the L_2 inner product, given the additional contraction assumptions on the regression map, which can be related to the stationarity condition of VAR models.

Lastly, we rely on the least squares method to derive the estimator of A, which minimizes the sum of squared residuals under the simplex constraint A2. We also provide the consistency guarantee for the proposed estimator.

2.2.1.8 Application in graph learning from distributional time series

Due to the simplex constraint A2, the derived estimator naturally carries sparsity. Thus when interpreting the coefficient matrix as the adjacency matrix of N nodes, we can retrieve a weighted directed graph from its estimate, which represents the dependency structure of N distributional processes $(\boldsymbol{\mu}_t^i)_{t\in\mathbb{Z}}, i = 1, \ldots, N$. Figure 2.4 shows the inferred age structure graph of countries from the data illustrated in Figure 2.2. The full list of the investigated countries can be found in Chapter 5.

2.2.1.9 Multivariate distribution-to-distribution regressive model

The proposed data centeralization approach for random distribution can also serve in the development of the multivariate distribution-to-distribution regressive model for the i.i.d. data, which is also very little developed. The existing works on the distribution-to-distribution regressive model mainly consider the 1 predictor 1 response case. In Chen et al. (2019), the kernel ridge regression of the transformed density functions by LQD are constructed. Even though the regression can be extended to multivariate case by expanding the domain of the reproducing kernel, as stated in Section 2.2.1.6, LQD transformation does not take into account the geometry

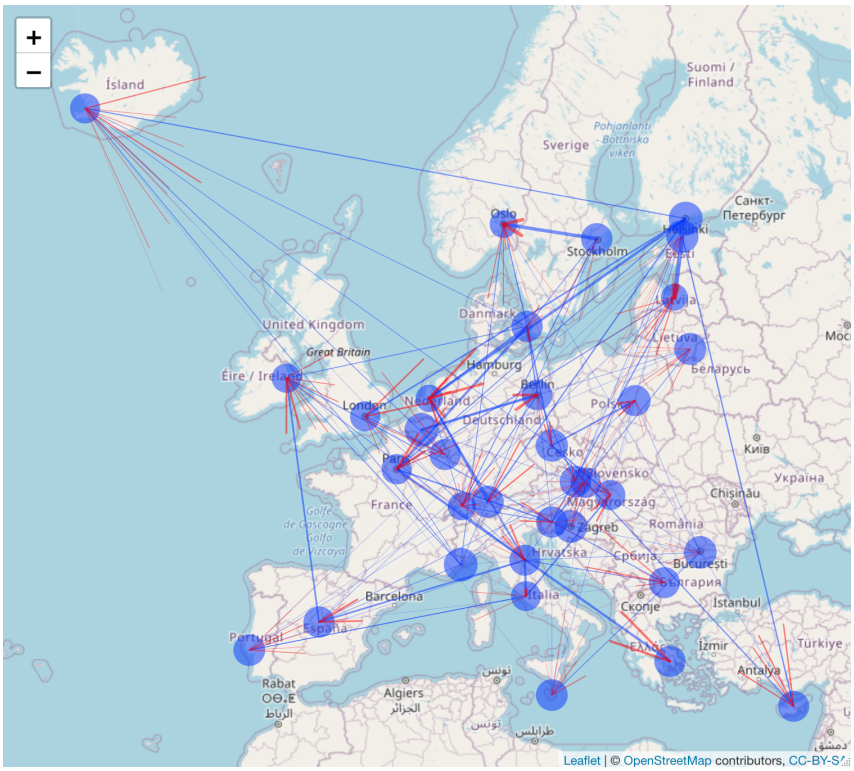

Figure 2.4: *Inferred age structure graph.* An edge implies the similarity of the age structures between the countries from 1996 to 2036 (projected). The direction can be understood as the propagation of changes in the structure. Thicker edge corresponds closer relationship between the linked countries, with respect to other incoming edges. The size of the blue circle around a country corresponds to the similarity of its age structures between two consecutive years. Bigger circle represents slower evolution.

of the distributional space. More recently, both Chen et al. (2021b); Ghodrati and Panaretos (2022) rely on the Wasserstein metric. Chen et al. (2021b) performs the regression using the coefficient operator which maps from the Tangent space of the predictor's Fréchet mean to that of the response's. By contrast, Ghodrati and Panaretos (2022) performs the regression directly in Wasserstein space. The predictor is pushforwarded by the coefficient optimal map to the conditional Fréchet mean of the response. It is not straightforward to extend the two models to more predictors since the related operations are both directional. That is, it is not evident how to map between multiple predictor tangent spaces or push among the predictor measures. Thus, the proposed data centering method in this work can provide a new trick to handle multiple predictors and responses in this more general regression framework.

2.2.2 Node predictability in time-dependent graph signals

To measure the predictability of nodes in a network, we firstly introduce the prediction problem for the network observations $(\boldsymbol{x}_{it})_t$, $i \in \mathcal{N} := \{1, \ldots, N\}$, where we hypothetically reconstruct the observations at time t over a subset I of nodes,

$\boldsymbol{x}_{it}$, $i \in I \subset \mathcal{N}$, given the past and present recordings on the rest of nodes $(\boldsymbol{x}_{j\tau})_{j\tau}$, $j \in I^c, t-H \leqslant \tau \leqslant t$. We would like to compare the predictability evaluated by the two popular prediction methods in the statistic and graph signal processing domains: kernel ridge regression and neural network. Additionally, when the edges of nodes are not given, we consider linear regression. For the two regression methods, we obtain the ranking through the greedy adaptation of the sensor selection problem of optimal I, by relating the prediction errors of observations over I to the selection criteria. For the neural network, we measure the node predictability in a finer way by providing each node a score. We propose to adopt the dropout scheme to vary the *turned-off* nodes in I, obtaining the aggregated prediction error for each node, in one training. In the following, we present these contributions in a more technical context. This section corresponds to the preprint Jiang et al. (2020).

2.2.2.1 Kernel ridge regression of time and node predictors

Firstly, we adapt the kernel ridge regression to the introduced prediction problem by regressing the observation $x_{i\tau} \in \mathbb{R}$ on the node and time indices $i \in \mathcal{N}$ and $\tau \in \mathbb{Z}$. We propose a kernel design on $\mathcal{N} \times \mathbb{Z}$ that takes into account the graph structure and defines the similarity of time points. The kernel ridge regression then approximates the regression function by the functions in the induced reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$, with the corresponding estimator given by

$$\widehat{f}_t = \min_{f \in \mathcal{H}_k} \sum_{i \in I^c, t-H \leqslant \tau \leqslant t} \|x_{i\tau} - f(i,\tau)\|_{\ell^2}^2 + \lambda \|f\|_{\mathcal{H}_k}^2. \tag{2.2.7}$$

By the RKHS-based prediction, the predictability of a node comes from the similarity between its observations and the recent observations over its neighbourhood. For the optimization as Equation (2.2.7), the representer theorem indicates the explicit form of solution. Thus, given a history of network observations, we derive the expression of the total prediction error $\sum_{i \in I} \sum_{H+1 \leqslant t \leqslant T} (x_{it} - \widehat{f}_t(i,t))^2/T$, which depends on the sample auto-covariance of process $(x_{it})_t, i \in \mathcal{N}$ and Gram matrix of the proposed kernel. The derived error gives a way to quantify the predictability of nodes, which however depends on the choice of I. We propose to rely on the sensor selection problem taking the total prediction error as the criteria, to take into account other possible compositions and cardinalities of I. For a given cardinality $1 \leqslant p \leqslant N-1$, the problem searches the optimal set $I^*(p)$ which minimizes a certain criteria among all the sets I of p nodes. Especially, since the adapted greedy strategy selects the nodes one by one, their order forms the optimal sets of cardinality from 1 to $N-1$ incrementally. Thus we propose to rank the nodes as the order produced by such a greedy strategy. The resulting ranking can be interpreted as follows: for every given cardinality p, we assign the nodes in the optimal set $I^*(p)$ score 1, the rest nodes score 0. The total scores for all p's from 1 to N forms the ranking.

2.2.2.2 Varying the prediction set I by dropout

Since the neural network has been widely adapted to prediction problems in the domain of graph signal processing, we are especially interested in evaluating its

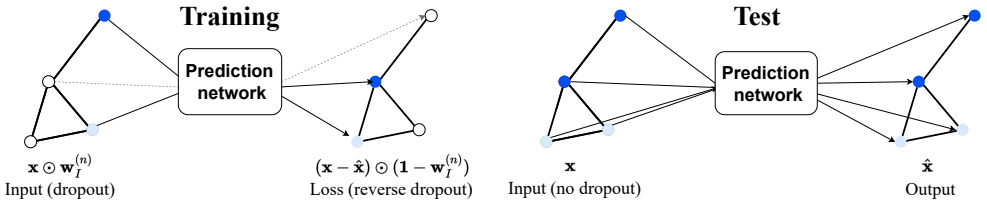

Figure 2.5: *Application of the dropout in predictability learning.* On the left represents the the optimization step on the n-th batch, where $\mathrm{x} = (x_i)_{i \in \mathcal{N}}$ represents the input and $\mathbf{w}_I^{(n)}$ is the dropout vector masking the input over nodes in I. I is sampled again at every batch n. After the training converges, the network is tested on the validation set as illustrated on the right, whose reconstruction errors of each node gives its score.

predictability over nodes[‖]. For the network predictor, we may set the training loss as the total prediction error given in Section 2.2.2.1 and then still relate it to the criteria in sensor selection so as to rely on the greedy strategy. However, this leads to $\mathcal{O}(N^2)$ times network training, which is much more costly than the matrix operations. Therefore, we design a training scheme for the network predictor, which varies the composition of I hence gives the scores of all nodes within one training. The key idea is illustrated in Figure 2.5. By applying the dropout and the reverse dropout on the input and the batch loss, the trained network aggregates the prediction models related to different I's. In our experiment, thousands of compositions of I have been sampled during the training for N on the order of one hundred. Therefore, the prediction error given by such network represents the overall predictability of a node, when it is predicted within different I's. We propose to use the trained network to predict the network observations of the validation set, and take the prediction error of each node as its score, measured by the metric R^2. Figure 2.6 shows the derived scores for the bike stations in Paris, calculated from a history of observation of 3533 hours, which records the ratio of available bike at each station and hour. These scores show the interesting result that the stations with high predictability forms several geographical groups. We focus on two groups. One goes along the main itinerary of Metro 1 and RER A to the central business district La defense (yellow white points on the right bank of the Seine river). The other is located around the train station Paris Montparnasse (yellow white points on the left bank). These two geographical groups will be found again in Chapter 5 by the proposed Wasserstein AR model in its graph inferred on the same data set, where the regression formulas predicting the stations inside the groups are much more significant, with larger magnitude of the nonzero coefficients.

These observations strongly support the effectiveness of both proposed methods.

[‖]For the network predictor, we only consider the prediction problem with $H = 0$, because we did not observe empirically the improvement in the reconstruction for larger H, when applying the learned ranking in the sensor selection.

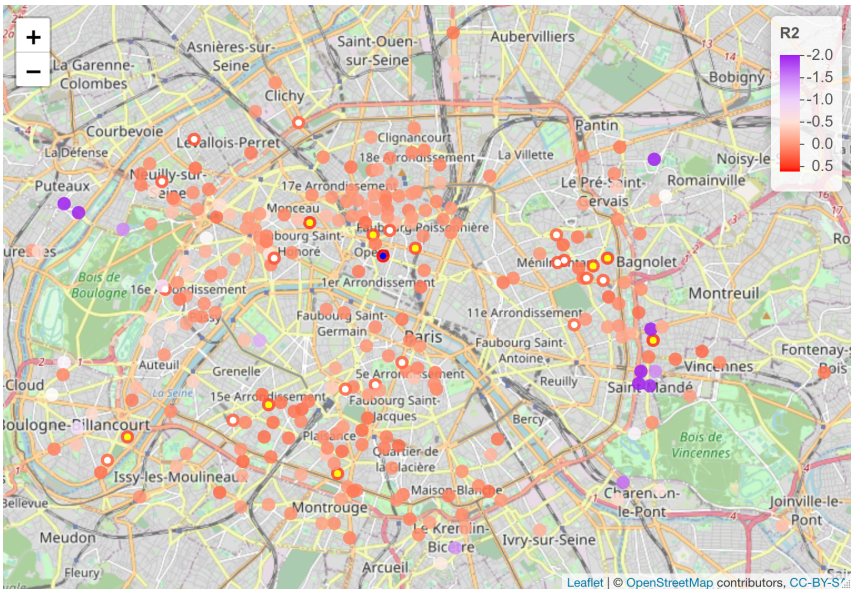

Figure 2.6: *Scores of node predictability for bike-sharing network in Paris.* The higher scores are shown in red and the lower in purple. The top 1, 10, and 27 nodes are in blue, yellow, and white respectively.

2.2.2.3 Partial variance of multivariate stochastic process

Lastly, we would like to derive the learning method for the node predictability, when the information on the node connectivity in a network is not available. We may firstly apply a graph learning method for the scalar-type node observation, such as VAR models, to infer the edges, then reuse the methods proposed in Sections 2.2.2.1 and 2.2.2.2. However, since this pipeline uses same data twice, the data noise will be amplified and cause larger bias in the fitted models. Thus we purpose to consider the time-dependent signals as multivariate process, and rely on the graph-free prediction methods.

An immediate notion in statistics which measures the predictability of a random variable $\boldsymbol{x}_i$ given other variables $\boldsymbol{x}_j, j \neq i$ is the *partial variance of* $\boldsymbol{x}_i$ *given* $\boldsymbol{x}_j, j \neq i$, which is essentially defined as the variance of residual of $\boldsymbol{x}_i$ predicted by $\boldsymbol{x}_j, j \neq i$ through the linear regression. We then rely on the partial variance of $\boldsymbol{x}_{it}$, $i \in I \subset \mathcal{N}$ given $(\boldsymbol{x}_{j\tau})_{j\tau}$, $j \in I^c, t-H \leqslant \tau \leqslant t$ to quantify the node predictability when the graph is not given. The total partial variance of $\boldsymbol{x}_{it}, i \in I$ given $(\boldsymbol{x}_{j\tau})_{j\tau}$, $j \in I^c, t-H \leqslant \tau \leqslant t$ is a counterpart of the total prediction error given in Section 2.2.2.1. Especially, when estimating the total partial variance of $\boldsymbol{x}_{it}, i \in I$ given $(\boldsymbol{x}_{j\tau})_{j\tau}$, $j \in I^c, t-H \leqslant \tau \leqslant t$ by the sample auto-covariance of process $(\boldsymbol{x}_{it})_t, i \in \mathcal{N}$, the estimator can be written as a special case of the derived total prediction error in Section 2.2.2.1, with the kernel defined otherwise by the sample auto-covariance, and $\lambda = 0$. We then regard the total partial variance as another criteria for the sensor selection problem, and rely on the greedy strategy to obtain the desired ranking as in Section 2.2.2.1.

2.2.2.4 Application in sensor selection

Given the derivation of the rankings, for any cardinality $1 \leqslant p \leqslant N-1$, the top p nodes can also be considered as the optimal sets over which the observations can be predicted collectively the best by the rest of the nodes. Such a selection strategy considers the aspect of dependency that can be estimated from the data, in additional to the graph structure. In the experiments of Chapter 6, we show that the proposed strategy exhibits advantages in the reconstruction of missing data when compared to standard sampling methods in graph signal processing that rely only on the graph structure, such as Anis et al. (2016a); Puy et al. (2018).

2.3 Organization of the thesis

Chapter 3 We present the necessary backgrounds to start the analysis of Chapters 4 - 6.

Chapter 4 We develop the approaches for the network data of vectorial node observations. We propose the matrix-variate AR(1) model and derive the online procedures to infer its characterising graph from the sequential trend data, in both low- and high-dimensions. This chapter is related to the preprint Jiang et al. (2021).

Chapter 5 We focus on analysing the distributional node observations. We propose the Wasserstein multivariate AR(1) model and the sparse estimator of its coefficients, with theoretical results. This chapter is related to the preprint Jiang (2022).

Chapter 6 This chapter is devoted to the problem of learning node predictability from the scalar-type network data. We propose the ranking strategies for the linear, RKHS, and neural network predictions. This chapter is related to the preprint Jiang et al. (2020).

Chapter 3

Preliminaries

3.1 Brief introduction of graphs

The most common mathematical object used to represent a network of sensors is a graph. A graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}, A\}$ consists of a finite set of nodes (or vertices) $\mathcal{N} = (V_i)_i$ with $|\mathcal{N}| = N$, a set of edges $\mathcal{E} = (i, j)_{i,j}$, and an adjacency matrix $A = (A_{ij})_{i,j}$ which stores the connectivity information. For an undirected graph $\mathcal{G}$, if there is an edge (i, j) connecting nodes i and j, the entry A_{ij} equals the weight of the edge (equals 1 if the graph is unweighted); otherwise $A_{ij} = 0$. In this case, the adjacency matrix is symmetric. By contrast, the adjacency matrix of directed graph is asymmetric. In this case, A_{ij} represents the weight (or 1) if and only if there is an edge from node i to node j (or defined as from node j or node i in some applications). A *graph signal* on $\mathcal{G}$ is defined as a mapping $h : \mathcal{N} \to \mathbb{R}$ with $h(V_i)$ representing the observation at node i.

An important notion which allows the development of tools for graph signal processing is the *graph Laplacian*. Here we focus on the case of undirected graphs with real weights, which is the most commonly considered. For the graph Laplacian defined on directed graphs, we refer to Wu (2005); Bauer (2012). The *combinatorial Laplacian* is defined as

$$L = D - A, \tag{3.1.1}$$

where A is the adjacency matrix and $D \in \mathbb{R}^{N \times N}$ is the *degree matrix*, that is, a diagonal matrix whose i-th diagonal element is equal to the sum of weights of all the edges incident to node i. One can normalize the combinatorial Laplacian to obtain the *symmetric normalized Laplacian*, which is defined as

$$L_{sym} = (D^+)^{\frac{1}{2}} L (D^+)^{\frac{1}{2}}, \tag{3.1.2}$$

where D^+ is the Moore–Penrose inverse of D. It is easy to find that the graph Laplacians defined in Equations (3.1.1) and (3.1.2) are positive semi-definite, thus

allowing orthonormal eigendecomposition with non-negative eigenvalues as

$$L = \Phi \begin{bmatrix} 0 & & & \\ & \lambda_1 & & \\ & & \ddots & \\ & & & \lambda_{N-1} \end{bmatrix} \Phi^\top := \Phi \Lambda \Phi^\top.$$

With real matrix and real eigenvalues, the eigenvectors can be chosen as all real. As for the classical one-dimensional Laplacian operator, the eigenvectors Φ of the graph Laplacian define the *Fourier basis* in the graph domain. Thus the Graph Fourier transform is given by

$$\hat{h} = \Phi^\top h,$$

where $h = h(V_i)_i \in \mathbb{R}^N$ is a graph signal.

3.2 Stationary process and vector AR models

3.2.1 Stationary process

A multivariate stochastic process $(\mathbf{x}_t)_t$ is a collection of random vectors in $\mathbb{R}^N$ indexed by $t \in \mathbb{Z}$. We can characterize $(\mathbf{x}_t)_t$ by its means and auto-covariance matrices (if these exist), defined respectively in Equations (3.2.1) and (3.2.2) as

$$\mathrm{u}_t = \mathbb{E}\mathbf{x}_t, \ \forall t, \in \mathbb{Z}. \tag{3.2.1}$$

$$\Gamma(t, t-h) = \mathbb{E}(\mathbf{x}_t - \mathrm{u}_t)(\mathbf{x}_{t-h} - \mathrm{u}_{t-h})^\top, \ \forall t, h \in \mathbb{Z}. \tag{3.2.2}$$

The stochastic process $(\mathbf{x}_t)_t$ is *stationary* if and only if

$$\mathbb{E}\|\mathbf{x}_t\|_{\ell_2}^2 < \infty, \ \mathrm{u}_t = \mathrm{u}_0, \ \text{and } \Gamma(t, t-h) = \Gamma(0, -h), \ \forall t, h \in \mathbb{Z}.$$

The stationarity is in the weak-sense throughout this manuscript, otherwise stated if not. Since the mean u_t and auto-covariance matrices $\Gamma(t, t-h)$ for the stationary process do not depend on the time instant t, we denote them furthermore by u and $\Gamma(h), h \in \mathbb{Z}$. Especially, the auto-covariance $\Gamma(0)$ at $h = 0$ will also be called variance in this manuscript. Given the time series data $\mathbf{x}_{1-p}, ..., \mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_T$, $p \in \mathbb{Z}$ and $1 \leqslant p \leqslant T-1$, we can then estimate the stationary mean u and stationary auto-covariances $\Gamma(h)$ by the sample mean and sample auto-covariances, defined as

$$\hat{\mathbf{u}} = \frac{1}{T}\sum_{\tau=1}^{T} \mathbf{x}_\tau, \ \text{and } \widehat{\mathbf{\Gamma}}(h) = \begin{cases} \frac{1}{T}\sum\limits_{\tau=1}^{T} (\mathbf{x}_\tau - \hat{\mathbf{u}})(\mathbf{x}_{\tau-h} - \hat{\mathbf{u}})^\top, \ h = 0, ..., p, \\ \widehat{\mathbf{\Gamma}}(-h)^\top, \ h = -1, ..., -p. \end{cases} \tag{3.2.3}$$

Note that, notation-wise, we do not distinguish between the stochastic process and a time series which is a realization of a stochastic process. When stationarity holds, these estimators are consistent under a mild condition, as stated in Theorem 3.2.1.

Theorem 3.2.1. *(Neusser, 2016, Theorems 11.1, 11.3) Let $(\mathbf{x}_t)_t, t \in \mathbb{Z}$ be a stationary process, that can be written as*

$$\mathbf{x}_t = \mathrm{u} + \sum_{j=-\infty}^{+\infty} \Psi_j \mathbf{z}_{t-j},$$

where $\mathbf{z}_t \sim \mathrm{IID}(0, \Sigma_z)$ with bounded forth moments, $\sum_{j=-\infty}^{+\infty} \|\Psi_j\|_{\mathbf{F}} < \infty$ and $\sum_{j=-\infty}^{+\infty} \Psi_j \neq 0$. $\mathrm{IID}(0, \Sigma_z)$ denotes independent and identically distributed with mean and variance respectively 0 and Σ_z. Then we have

$$\hat{\mathbf{u}} \xrightarrow{p} \mathrm{u}, \text{ and } \hat{\mathbf{\Gamma}}(h) \xrightarrow{p} \Gamma(h).$$

The summability condition is quite general, and in particular, it is fulfilled by the VAR process introduced in Section 3.2.2.

3.2.2 Vector AR model of order 1 and estimation of coefficients

A powerful tool to analyze multiple scalar time series is VAR model. Let $(\mathbf{x}_t)_t, t \in \mathbb{Z}$ be a random process in $\mathbb{R}^N$. $(\mathbf{x}_t)_t$ is called VAR(1) process if it is the solution of the stochastic difference equation

$$\mathbf{x}_t = \mathrm{b} + A\mathbf{x}_{t-1} + \mathbf{z}_t, \quad t \in \mathbb{Z}, \tag{3.2.4}$$

where $\mathbf{z}_t \sim \mathrm{WN}(0, \Sigma)$ is a white noise process with the variance Σ not singular, $A \in \mathbb{R}^{N \times N}$ is the coefficient matrix, and $\mathrm{b} \in \mathbb{R}^N$ is the intercept to cancel out the unequal expectations in the regressive formula, in other words, nonzero means $\mathbb{E}\mathbf{x}_t$ and $A\mathbb{E}\mathbf{x}_t$.

When $\mathbb{E}\mathbf{x}_t$ are known to be time-invariant, that is $\mathbb{E}\mathbf{x}_t = \mathrm{u}$, $t \in \mathbb{Z}$, the VAR(1) model admits another formulation reparameterized in the unknown process mean u and the coefficient matrix A. In this case, we can find by taking expectation on both sides of Equation (3.2.4) that, the constant b is determined by u and A

$$\mathrm{b} = (I - A)\mathrm{u}, \tag{3.2.5}$$

where I is the identity matrix. Plugging this relation in the regressive formula (3.2.4), we then obtain the VAR (1) model reparametrized by u and A

$$\mathbf{x}_t - \mathrm{u} = A(\mathbf{x}_{t-1} - \mathrm{u}) + \mathbf{z}_t, \quad t \in \mathbb{Z}. \tag{3.2.6}$$

Therefore we can directly study the centered series $\mathbf{x}_t - \mathrm{u}$, and consider the VAR(1) model without intercept.

The core of time series analysis is the study of stationary processes. Thus it is important to establish the stationarity condition for the VAR models. We present in Theorem 3.2.2 such a condition.

Theorem 3.2.2. *When $\|A\|_2 < 1$, the stochastic difference equation defined in Equation (3.2.4) admits the unique stationary solution*

$$\mathbf{x}_t = (I - A)^{-1} b + \sum_{j=0}^{\infty} A^j \mathbf{z}_{t-j},$$

where the infinite sum exists in L^2.

A stationary VAR(1) process thus refers to such a unique solution. Since stationarity implies the time-invariant process mean, the reparameterized formulation (3.2.6) then holds for stationary VAR(1) process. Multiplying $(\mathbf{x}_t - \mathrm{u})^\top$ on the right and taking expectation on both sides, we can obtain the following representation of the coefficient matrix

$$A = \Gamma(1)\left[\Gamma(0)\right]^{-1}, \tag{3.2.7}$$

where $\Gamma(0)$, and $\Gamma(1) \in \mathbb{R}^{N\times N}$ are the auto-covariance matrices defined in Equation (3.2.2) of the stationary process.

The VAR(1) model can then be extended to VAR(p) with p time lags by

$$\mathbf{x}_t = \mathrm{b} + \sum_{l=1}^{p} A^l \mathbf{x}_{t-l} + \mathbf{z}_t, \quad t \in \mathbb{Z}.$$

However, VAR(p) models will not be exploited in the present work yet. For more details on VAR models, we refer to the following books: Lütkepohl (2005); Brockwell and Davis (2009); Neusser (2016).

Given the time series data $\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_T$, we can then estimate the coefficient A by the general least squares (GLS) method or by the ordinary least squares (OLS) method, defined respectively through the optimisation problems (3.2.8) and (3.2.9) below. For the regression model (3.2.4) and (3.2.6), GLS and OLS estimators are equivalent (Lütkepohl, 2005, Equation 3.2.8), because the predictors in each equation of $\boldsymbol{x}_{it}$ are the same.

$$\widehat{\mathbf{A}}, \hat{\boldsymbol{b}} = \arg\min_{A,b} \sum_{\tau=1}^{T} (\mathbf{x}_\tau - \mathrm{b} - A\mathbf{x}_{\tau-1})\Sigma^{-1}(\mathbf{x}_\tau - \mathrm{b} - A\mathbf{x}_{\tau-1})^\top, \tag{3.2.8}$$

$$\widehat{\mathbf{A}}, \hat{\boldsymbol{b}} = \arg\min_{A,b} \sum_{\tau=1}^{T} (\mathbf{x}_\tau - \mathrm{b} - A\mathbf{x}_{\tau-1})(\mathbf{x}_\tau - \mathrm{b} - A\mathbf{x}_{\tau-1})^\top, \tag{3.2.9}$$

where Σ is the true noise variance given in Model (3.2.4). Through the straightforward calculation, the explicit forms of such minimizers are found to be

$$\widehat{\mathbf{A}} = \widehat{\mathbf{\Gamma}}(1)\left[\widehat{\mathbf{\Gamma}}(0)\right]^{-1}, \text{ and } \widehat{\mathbf{b}} = (I - \widehat{\mathbf{A}})\hat{\mathbf{u}},$$

where $\hat{\mathbf{u}}, \widehat{\mathbf{\Gamma}}(\cdot)$ are the sample mean and sample auto-covariances defined in Equation (3.2.3). Analogously, when stationarity holds, these estimators are consistent, given the representation (3.2.7) and the consistency result 3.2.1. Furthermore they permit the central limit theorem (CLT), as stated in Theorem 3.2.3.

Theorem 3.2.3. *(Lütkepohl, 2005, Proposition 3.1) Let $(\mathbf{x}_t)_t$ be a stationary VAR(1) process defined as the solution of Equation (3.2.4), with additionally $\mathbf{z}_t \sim \mathrm{IID}(0, \Sigma)$ with bounded forth moments. Then we have*

$$\frac{1}{\sqrt{T}} vec \begin{pmatrix} \widehat{\mathbf{b}} - \mathrm{b} \\ \widehat{\mathbf{A}} - A \end{pmatrix} \xrightarrow{d} \mathcal{N}\left(0, \begin{pmatrix} 1 & \mathrm{u}^\top \\ \mathrm{u} & \mathbb{E}\mathbf{x}_t\mathbf{x}_t^\top \end{pmatrix}^{-1} \otimes \Sigma\right)$$

Applying the formula of the block matrix inversion, we can obtain the marginal distribution of $\widehat{\mathbf{A}}$

$$\frac{1}{\sqrt{T}} vec\left(\widehat{\mathbf{A}} - A\right) \xrightarrow{d} \mathcal{N}\left(0, \Gamma(0)^{-1} \otimes \Sigma\right).$$

3.2.3 Granger causality and Wald test

The formal definition of the Granger causality is:

Definition 3.2.4. (Lütkepohl, 2005, Section 2.3.1) Let $\boldsymbol{z}_t \in \mathbb{R}$, $\boldsymbol{y}_t \in \mathbb{R}$ be two univariate processes. Suppose that Ω_t is the information set containing all the relevant information in the universe available up to and including period t. Let $\boldsymbol{z}_t(h|\Omega_t)$ be the optimal (mean squared error sense) h-step predictor of $\boldsymbol{z}_t$ at origin t, based on the information in Ω_t. The corresponding forecast mean squared error is denoted by $\Sigma_z(h|\Omega_t)$. The process $\boldsymbol{y}_t$ is said to cause process $\boldsymbol{z}_t$ in Granger's sense if

$$\Sigma_z(h|\Omega_t) < \Sigma_z(h|\Omega_t \backslash \{\boldsymbol{y}_\tau | \tau \leqslant t\}), \quad \text{for at least one } h \in \mathbb{N}^+,$$

where $\Omega_t \backslash \{\boldsymbol{y}_\tau | \tau \leqslant t\}$ is the set containing all the relevant information except for the information in the past and present of $\boldsymbol{y}_t$.

As mentioned in Section 2.2.1.1 in the introduction, for a stationary VAR(p) model, time series $(\boldsymbol{x}_{it})_t$ causes $(\boldsymbol{x}_{jt})_t$ if and only if all the ji-th entries of the true coefficient matrices $A^l_{ji} \neq 0$, $l = 1, \ldots, p$, (Lütkepohl, 2005, Corollary 2.2.1). Thus, the causal graph is defined by the sparsity structure of the coefficient matrices A^l of a stationary VAR process. In low-dimensional regime, this structure can be identified by testing the zero constraints for the entries of A^l via the Wald test. The Wald test considers generally the hypotheses*:

$$H_0 : C\text{vec}(A) = \text{c} \text{ versus } H_1 : C\text{vec}(A) \neq \text{c},$$

where $C \in \mathbb{R}^{P \times N^2}$ is of rank P with P an arbitrary integer, and $\text{c} \in \mathbb{R}^P$. The test statistic is given by

$$\boldsymbol{\lambda}_W = T\hat{\boldsymbol{\alpha}}^\top \left[C\hat{\boldsymbol{\Sigma}}_W C^\top\right]^{-1} \hat{\boldsymbol{\alpha}},$$

where $\hat{\boldsymbol{\alpha}} = C\text{vec}(A) - \text{c}$, $\hat{\boldsymbol{\Sigma}}_W = \left[\widehat{\boldsymbol{\Gamma}}(0)\right]^{-1} \otimes \widehat{\boldsymbol{\Sigma}}$ with $\widehat{\boldsymbol{\Sigma}}$ some consistent estimator of the variance of noise Σ. Based on the CLT in Theorem 3.2.3, the asymptotic distribution of $\boldsymbol{\lambda}_W$ is

Corollary 3.2.5. *(Lütkepohl, 2005, Proposition 3.5)*

$$\boldsymbol{\lambda}_W \xrightarrow{d} \chi^2(P) \text{ Under } H_0.$$

3.3 Statistics in Wasserstein space

In this section, we provide the required preliminaries to establish the proposed Wasserstein model in Chapter 5 for distributions on an interval of $\mathbb{R}$. For comprehensive reviews on the statistics Wasserstein in space, we refer to Bigot (2020); Panaretos and Zemel (2020); Petersen et al. (2022).

*We only demonstrate Wald test with VAR(1) model, as the preliminaries of the present work. For the test of VAR(p) model, see the cited book.

Let Ω be a (possibly unbounded) interval in $\mathbb{R}$, and $\mathcal{B}(\Omega)$ the associated σ-algebra made of Borel sets of Ω. Let μ be a probability measure (namely a distribution) over $(\Omega, \mathcal{B}(\Omega))$ with cumulative distribution function (cdf) F_μ. Then the (generalized) quantile function is defined as the left continuous inverse of F_μ, denoted by F_μ^{-1}, that is

$$F_\mu^{-1}(p) := \inf\{x \in \Omega : F_\mu(x) \geqslant p\}, \quad p \in (0,1).$$

The Wasserstein space $\mathcal{W} := \mathcal{W}_2(\Omega)$ is defined as the set of probability measures over $(\Omega, \mathcal{B}(\Omega))$ with finite second moment, that is endowed with the $\mathcal{L}^2$ Wasserstein distance

$$d_W(\mu,\nu) = \left(\int_0^1 \left[F_\mu^{-1}(p) - F_\nu^{-1}(p)\right]^2 dp\right)^{1/2}, \quad \mu,\nu \in \mathcal{W}_2(\Omega). \tag{3.3.1}$$

It is well known that $\mathcal{W}$ is a complete and separable metric space.

3.3.1 Tangent bundle

The space $\mathcal{W}$ has a pseudo-Riemannian structure (Ambrosio et al., 2008). Let $\gamma \in \mathcal{W}$ be an absolutely continuous measure, the tangent space at γ is defined as

$$\mathrm{Tan}_\gamma = \overline{\{t(F_\mu^{-1} \circ F_\gamma - id) : \mu \in \mathcal{W},\ t > 0\}}^{\mathcal{L}^2_\gamma(\Omega)},$$

where id is the identity function, $\mathcal{L}^2_\gamma(\Omega)$ is the Hilbert space of γ square integrable functions on Ω, with inner product $\langle\cdot,\cdot\rangle_\gamma$ defined by $\langle f, g\rangle_\gamma := \int_\Omega f(x)g(x)\, d\gamma(x),\ f, g \in \mathcal{L}^2_\gamma(\Omega)$, and the induced norm $\|\cdot\|_\gamma$. The exponential and the logarithmic maps at γ are then defined as follows.

Definition 3.3.1. The exponential map $\mathrm{Exp}_\gamma : \mathrm{Tan}_\gamma \to \mathcal{W}$ is defined as

$$\mathrm{Exp}_\gamma\, g = (g + id)\#\gamma, \tag{3.3.2}$$

where for any measurable function $T : \Omega \to \Omega$ and $\mu \in \mathcal{W}$, $T\#\mu$ is the pushforward measure on Ω defined as $T\#\mu(A) = \mu(\{x \in \Omega : T(x) \in A\})$, for any set $A \in \mathcal{B}(\Omega)$.

Definition 3.3.2. The logarithmic map $\mathrm{Log}_\gamma : \mathcal{W} \to \mathrm{Tan}_\gamma$ is defined as

$$\mathrm{Log}_\gamma\, \mu = F_\mu^{-1} \circ F_\gamma - id.$$

Note that the exponential map (3.3.2) is not a local homeomorphism (Ambrosio et al., 2004). Nevertheless, when restricted to the image of the logarithmic map, it becomes an isometry (Bigot et al., 2017) as stated in the following proposition.

Proposition 3.3.3. *Let $\gamma \in \mathcal{W}$ be any absolutely continuous measure. Then $\mathrm{Exp}_\gamma|_{\mathrm{Log}_\gamma \mathcal{W}}$ is an isometric homeomorphism from $\mathrm{Log}_\gamma \mathcal{W}$ to $\mathcal{W}$, with the inverse map Log_γ, satisfying*

$$d_W(\mu,\nu) = \|\mathrm{Log}_\gamma\, \mu - \mathrm{Log}_\gamma\, \nu\|_\gamma.$$

The Wasserstein distance in Equation (3.3.1) can be interpreted as $\|\operatorname{Log}_{Leb}\mu - \operatorname{Log}_{Leb}\nu\|_{Leb}$, which is the distance between Logarithmic maps of μ and ν in the tangent space at $\gamma = Leb$ the uniform distribution namely the Lebesgue measure, over $[0,1]$, if $[0,1] \in \Omega$. For the general definition of Wasserstein distance (between probability measures supported in general metric spaces) related to the optimal transport theory, see the references Villani (2021); Panaretos and Zemel (2020). We can also remark from the above results that the space of all quantile functions of measures in $\mathcal{W}$, namely $\operatorname{Log}_{Leb}\mathcal{W} + id$, is a complete separable metric space with respect to $\|\cdot\|_{Leb}$.

We recall below the important properties of $\operatorname{Log}_\gamma \mathcal{W}$ (Bigot et al., 2017) that are needed in the construction of our proposed model.

Proposition 3.3.4. *$\operatorname{Log}_\gamma \mathcal{W}$ is a closed and convex subset of $\mathcal{L}^2_\gamma(\Omega)$.*

Proposition 3.3.5. *Let $g \in \operatorname{Tan}_\gamma$, then $g \in \operatorname{Log}_\gamma \mathcal{W}$ if and only if $g + id$ is nondecreasing γ-almost everywhere.*

3.3.2 Fréchet means in Wasserstein space

Definition 3.3.6. Let $\mu_1, \ldots, \mu_T$ be measures in $\mathcal{W}$. The empirical Fréchet mean of $\mu_1, \ldots, \mu_T$, denoted by $\bar{\mu}$, is defined as the unique minimizer of

$$\min_{\nu \in \mathcal{W}} \frac{1}{T} \sum_{t=1}^{T} d_W^2(\mu_t, \nu).$$

It is well known that the empirical Fréchet mean $\bar{\mu}$ admits a simple expression through its quantile function that satisfies

$$F_{\bar{\mu}}^{-1}(p) = \frac{1}{T} \sum_{t=1}^{T} F_{\mu}^{-1}(p),\ p \in (0,1).$$

Definition 3.3.7. A random measure $\boldsymbol{\mu}$ is any measurable map from a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ to the metric space $\mathcal{W}$, endowed with its Borel σ-algebra.

Definition 3.3.8. Let $\boldsymbol{\mu}$ be a random measure from probability space $(\Omega, \mathcal{F}, \mathbb{P})$ to $\mathcal{W}$. Assume that $\boldsymbol{\mu}$ is square integrable, namely $\mathbb{E}d_W^2(\boldsymbol{\mu}, \nu) < \infty$ for some (thus for all) $\nu \in \mathcal{W}$. Then, the population Fréchet mean of $\boldsymbol{\mu}$, denoted by $\mu_\oplus$, is defined as the unique minimizer of

$$\min_{\nu \in \mathcal{W}} \mathbb{E}\left[d_W^2(\boldsymbol{\mu}, \nu)\right].$$

Note that $\mu_\oplus$ also admits a simple expression through its quantile function as

$$F_{\mu_\oplus}^{-1}(p) = \mathbb{E}\left[F_{\boldsymbol{\mu}}^{-1}(p)\right],\ p \in (0,1).$$

In the remarks below, we point out two important facts about Fréchet mean and the Logarithmic map.

Remark 3.3.9. The random measure $\boldsymbol{\mu}$ has zero expectation in the tangent space at its Fréchet mean, that is

$$\mathbb{E}\operatorname{Log}_{\mu_\oplus}\boldsymbol{\mu} = 0,\ \mu_\oplus \text{ almost everywhere.}$$

Remark 3.3.10. The Fréchet mean can be calculated using standard expectation in the tangent space at any absolute continuous measure $\gamma \in \mathcal{W}$, that is

$$\mu_{\oplus} = \mathrm{Exp}_{\gamma}(\mathbb{E}\,\mathrm{Log}_{\gamma}\,\boldsymbol{\mu}).$$

3.4 Theory on Iterated random function system

After establishing the multivariate AR model in the Wasserstein space, we would like to analyse the existence, uniqueness and then stationarity of its solution as in Theorem 3.2.2 for the VAR(1) model. We rely on the iterated random function (IRF) system theory in Wu and Shao (2004), where they propose the geometric moment contraction that is a sufficiently mild condition for a general IRF system to be stable in a metric space. Since the results in Wu and Shao (2004) are not presented in terms of the solution of a time series model, thus we also refer to Zhu and Müller (2021, Theorem 1), where even designed for their own model, they essentially represent the stability result, hence inducing a solution for the general time series model in metric space under the geometric moment contraction condition. We then adopt the notation of Zhu and Müller (2021, Theorem 1) to show the induced solution is the unique solution with some moment finite, and finally show the stationarity of such unique solution when the metric space is moreover induced from some Hilbert space. To analyse the proposed AR model, we apply this entire theory by imposing the assumptions under which the model satisfies the geometric moment contraction condition. Additionally, we will also rely on Theorem 3 in Wu and Shao (2004) to derive the consistency of the proposed estimator of the model coefficient. However, we will not repeat the theorem in this section, since the presentation of the theorem allows the direct application onto the estimator of a time series parameter. We refer the readers to Wu and Shao (2004) for the theorem.

In Sections 3.4.1 and 3.4.2, we present the above theory in the corresponding progressive results. We consolidate the proofs of Wu and Shao (2004) and Zhu and Müller (2021) to more detailed ones, which are given in Sections 3.4.3. To distinguish the contributions, we cite the original works for each result. The results without citation are developed by this work.

3.4.1 Time series in metric space

Let $(\mathcal{X}, d)$ be a complete separate metric space with Borel set X, an iterated random function (IRF) system in the state space $(\mathcal{X}, d)$ is defined as

$$\mathbf{X}_t = \boldsymbol{\Phi}_{\epsilon_t}(\mathbf{X}_{t-1}), \quad t \in \mathbb{Z}, \tag{3.4.1}$$

where $\boldsymbol{\epsilon}_t, t \in \mathbb{Z}$ are i.i.d. random objects taking values in a measurable space Θ, $\boldsymbol{\epsilon}_t$ is almost surely independent of $\mathbf{X}_{t-1}$ for all $t \in \mathbb{Z}$, $\boldsymbol{\Phi}_{\epsilon}(\cdot) := \boldsymbol{\Phi}(\cdot, \epsilon)$ is the ϵ-section of a jointly measurable function $\boldsymbol{\Phi} : \mathcal{X} \times \Theta \to \mathcal{X}$. Note that $\mathbf{X}_t, t \in \mathbb{Z}$ can also be seen as a $\mathcal{X}$-valued nonlinear auto-regressive process.

We now define $\widetilde{\mathbf{\Phi}}_{t,m} := \mathbf{\Phi}_{\epsilon_t} \circ \mathbf{\Phi}_{\epsilon_{t-1}} \circ \ldots \circ \mathbf{\Phi}_{\epsilon_{t-(m-1)}}$ the same way as Theorem 1 in Zhu and Müller (2021), and we recall Conditions C1 and C2 from Zhu and Müller (2021) as follows.

Condition C1. (Condition 1 in Wu and Shao (2004)) There exists $Y^0 \in \mathcal{X}$ and $\alpha > 0$, such that

$$I(\alpha, Y^0) := \mathbb{E}d^\alpha(Y^0, \mathbf{\Phi}_{\epsilon_t}(Y^0)) < \infty.$$

Condition C2. (Theorem 1 in Zhu and Müller (2021); Condition 2 in Wu and Shao (2004)) There exists $X^0 \in \mathcal{X}$, $\alpha > 0$, $r = r(\alpha) \in (0,1)$, and $C = C(\alpha) > 0$, such that for all $t \in \mathbb{Z}$, we have

$$\mathbb{E}d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\mathbf{\Phi}}_{t,m}(X)) \leqslant Cr^m d^\alpha(X^0, X), \quad \forall X \in \mathcal{X}, m \in \mathbb{N}. \tag{3.4.2}$$

Note that $\boldsymbol{\epsilon}_t, t \in \mathbb{Z}$ are i.i.d, thus for any fixed $X \in \mathcal{X}$, we have $\widetilde{\mathbf{\Phi}}_{t,m}(X) \overset{d}{=} \widetilde{\mathbf{\Phi}}_{t',m}(X), \forall t, t' \in \mathbb{Z}$. Thus Condition C2 is not a uniform requirement imposed for $t \in \mathbb{Z}$. We recall firstly in Lemma 3.4.1 and Lemma 3.4.2 below, the stability results given in Wu and Shao (2004, Theorem 2) and Zhu and Müller (2021, Theorem 1).

Lemma 3.4.1. *(Wu and Shao (2004, Theorem 2); Zhu and Müller (2021, Theorem 1)) Assuming that Conditions C1 and C2 hold, it follows that, for any fixed $t \in \mathbb{Z}$, $\lim_{m\to\infty} \widetilde{\mathbf{\Phi}}_{t,m}(X^0)$ exists almost surely and is denoted by $\widetilde{\boldsymbol{X}}_t$. Moreover, for any fixed $t, t' \in \mathbb{Z}$, $\widetilde{\boldsymbol{X}}_t \overset{d}{=} \widetilde{\boldsymbol{X}}_{t'}$. We denote this time-invariant marginal distribution as π.*

Lemma 3.4.2. *(Wu and Shao (2004, Theorem 2); Zhu and Müller (2021, Theorem 1)) Assuming that Conditions C1 and C2 hold, then the limits in Lemma 3.4.1 do not depend on the departure point X^0, that is, for any fixed $X \in \mathcal{X}$ and any fixed $t \in \mathbb{Z}$, $\widetilde{\mathbf{\Phi}}_{t,m}(X) \overset{m\to\infty}{\longrightarrow} \widetilde{\boldsymbol{X}}_t$, in d, almost surely.*

The random process $\widetilde{\boldsymbol{X}}_t, t \in \mathbb{Z}$ is then a solution to the IRF system (3.4.1). This result is indicated in Zhu and Müller (2021, Theorem 1), however, they do not elaborate the proof. We thus provide a proof in Section 3.4.3.3.

Lemma 3.4.3. *(Zhu and Müller (2021, Theorem 1)) Suppose that Conditions C1 and C2 hold, then $\widetilde{\boldsymbol{X}}_t = \lim_{m\to\infty} \widetilde{\mathbf{\Phi}}_{t,m}(X^0)$, $t \in \mathbb{Z}$ is a solution of IRF system (3.4.1), almost surely.*

We gather all the above results in Theorem 3.4.4.

Theorem 3.4.4. *(Existence; Wu and Shao (2004, Theorem 2); Zhu and Müller (2021, Theorem 1)) Suppose that Conditions C1 and C2 hold, then the IRF system (3.4.1) almost surely admits a solution $\widetilde{\boldsymbol{X}}_t$, $t \in \mathbb{Z}$, with the same marginal distribution π.*

The uniqueness result provided in Zhu and Müller (2021, Theorem 1) is no longer valid for the general system (3.4.1), which is also not mentioned in Wu and Shao (2004, Theorem 2). Thus, we provide the uniqueness result for the general system in Theorem 3.4.5. For the proof we refer to Section 3.4.3.4.

Theorem 3.4.5. *(Uniqueness) Suppose that Conditions C1 and C2 hold, then, if there is another solution* $\boldsymbol{S}_t$, $t \in \mathbb{Z}$, *such that*

$$\mathbb{E}\left[d^{\beta}(\boldsymbol{S}_t, Z^0)\right] < M, \quad t \in \mathbb{Z}, \tag{3.4.3}$$

for some $M, \beta > 0$, *and some* $Z^0 \in \mathcal{X}$ *(thus for all), then for all* $t \in \mathbb{Z}$

$$\widetilde{\boldsymbol{X}}_t = \boldsymbol{S}_t, \quad \textit{in } d, \quad \textit{almost surely.}$$

Since $\widetilde{\boldsymbol{X}}_t, t \in \mathbb{Z}$ is the unique solution of IRF system (3.4.1), we hereafter denote this solution directly by $\mathbf{X}_t, t \in \mathbb{Z}$. Lastly, we recall the geometric moment contraction result given in Wu and Shao (2004, Theorem 2), which will be used later on in the proof of consistency of the estimators. We reproduce their proof using our notation, in Section 3.4.3.5.

Proposition 3.4.6. *(Wu and Shao (2004, Theorem 2)) the IRF system (3.4.1) is geometric moment contracting in the following sense: let* $\mathbf{X} \sim \pi$ *be independent of* $\mathbf{X}^1 \sim \pi$, *where* π *is the shared marginal distribution in Theorem 3.4.4,* $\mathbf{X}, \mathbf{X}^1$ *are independent of* $\boldsymbol{\epsilon}_m, m \geqslant 1$. *Let* $\mathbf{X}_m(\mathbf{X}), \mathbf{X}_m(\mathbf{X}^1), m \geqslant 1$, *denote the sequences generated by the model (3.4.1) starting respectively from* $\mathbf{X}, \mathbf{X}^1$. *Then, for all* $m \geqslant 1$, *there exist constants* $D > 0, s \in (0,1)$, *such that*

$$\mathbb{E}d^{\alpha}(\mathbf{X}_m(\mathbf{X}), \mathbf{X}_m(\mathbf{X}^1)) \leqslant Ds^m. \tag{3.4.4}$$

3.4.2 Time series in Hilbert space

Now, we study the stationarity of the functional time series $\mathbf{X}_t$, $t \in \mathbb{Z}$. Theorem 3.4.4 has already implied that $\mathbf{X}_t$ is stationary in the sense that $\mathbf{X}_t \overset{d}{=} \pi$, $t \in \mathbb{Z}$. However, the stationarity in weak sense for time series requires additionally the auto-covariance to be time-invariant. To this end, we need furthermore a Hilbert structure in $\mathcal{X}$ to be able to define the notion of covariance between two random objects. Thus, we suppose $(\mathcal{X}, d)$ is furthermore a Hilbert space, whose inner product and the induced norm are denoted respectively by $\langle , \rangle$ and $\|\|$. To make the previous results presented in the metric space $(\mathcal{X}, d)$ valid as the results for the Hilbert space $(\mathcal{X}, \langle , \rangle)$, we assume $(\mathcal{X}, d)$ is indeed the induced metric space of $(\mathcal{X}, \langle , \rangle)$, namely, $d(X, Y) = \|X - Y\|$, $X, Y \in \mathcal{X}$.

We recall the conventional definition of stationarity for process in a separable Hilbert space, see for example Zhang et al. (2021, Definition 2.2). We can now give the stationarity result in Theorem 3.4.8. The proof can be found in Section 3.4.3.6.

Definition 3.4.7. A random process $\{\mathbf{V}_t\}_t$ in a separable Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ is said to be stationary if the following properties are satisfied.

1. $\mathbb{E}\|\mathbf{V}_t\|^2 < \infty$.
2. The Hilbert mean $U := \mathbb{E}[\mathbf{V}_t]$ does not depend on t.
3. The auto-covariance operators defined as

$$\mathcal{G}_{t,t-h}(V) := \mathbb{E}\langle \mathbf{V}_t - U, V \rangle (\mathbf{V}_{t-h} - U), \quad V \in \mathcal{H},$$

do not depend on t, that is $\mathcal{G}_{t,t-h}(V) = \mathcal{G}_{0,-h}(V)$ for all t.

Theorem 3.4.8. *Suppose that Conditions C1 and C2 hold with $\alpha \geqslant 2$, then the unique solution $\mathbf{X}_t, t \in \mathbb{Z}$ given in Theorems 3.4.4 and 3.4.5 is stationary in $(\mathcal{X}, \langle , \rangle)$ in the sense of Definition 3.4.7.*

The condition $\alpha \geqslant 2$ is to ensure the existence of a second moment of $\mathbf{X}_t$, from which we can define the auto-covariance operators of the time series. Note that for any $t \in \mathbb{Z}$

$$\begin{aligned}
\mathbb{E}\|\mathbf{X}_t\|^2 &\leqslant \mathbb{E}\|\mathbf{X}_t - \mathbf{\Phi}_{\epsilon_t}(X^0)\|^2 + \mathbb{E}\|\mathbf{\Phi}_{\epsilon_t}(X^0) - \mathbf{\Phi}_{\epsilon_t}(Y^0)\|^2 \\
&\qquad + \mathbb{E}\|\mathbf{\Phi}_{\epsilon_t}(Y^0) - Y^0\|^2 + \mathbb{E}\|Y^0\|^2 \\
&= \mathbb{E}d^2(\mathbf{X}_t, \mathbf{\Phi}_{\epsilon_t}(X^0)) + \mathbb{E}d^2(\mathbf{\Phi}_{\epsilon_t}(X^0), \mathbf{\Phi}_{\epsilon_t}(Y^0)) + \mathbb{E}d^2(\mathbf{\Phi}_{\epsilon_t}(Y^0), Y^0) + \|Y^0\|^2 \\
&\overset{(a)}{\leqslant} \left(\mathbb{E}d^\alpha(\mathbf{X}_t, \mathbf{\Phi}_{\epsilon_t}(X^0))\right)^{\frac{2}{\alpha}} + \left(\mathbb{E}d^\alpha(\mathbf{\Phi}_{\epsilon_t}(X^0), \mathbf{\Phi}_{\epsilon_t}(Y^0))\right)^{\frac{2}{\alpha}} \\
&\qquad + \left(\mathbb{E}d^\alpha(\mathbf{\Phi}_{\epsilon_t}(Y^0), Y^0)\right)^{\frac{2}{\alpha}} + \|Y^0\|^2 \\
&\overset{(b)}{\leqslant} \left(\mathbb{E}d^\alpha(\mathbf{X}_t, \mathbf{\Phi}_{\epsilon_t}(X^0))\right)^{\frac{2}{\alpha}} + \left(Cr d^\alpha(X^0, Y^0)\right)^{\frac{2}{\alpha}} + I(\alpha, Y^0)^{\frac{2}{\alpha}} + \|Y^0\|^2.
\end{aligned}$$

Inequality (a) comes from the condition $\alpha \geqslant 2$ and Jensen's inequality. Inequality (b) comes from Conditions C1 and C2. Thus, by the upperbound (3.4.5), we have $\mathbb{E}\|\mathbf{X}_t\|^2 < \infty$ is uniformly bounded over $t \in \mathbb{Z}$.

3.4.3 Proofs

3.4.3.1 Proof of Lemma 3.4.1

We are first to show that

$$I(\alpha, X^0) := \mathbb{E}d^\alpha(X^0, \mathbf{\Phi}_{\epsilon_t}(X^0)) < \infty.$$

We have

$$\begin{aligned}
I(\alpha, X^0) &\leqslant \mathbb{E}d^\alpha(X^0, Y^0) + \mathbb{E}d^\alpha(Y^0, \mathbf{\Phi}_{\epsilon_t}(X^0)) \\
&\leqslant \mathbb{E}d^\alpha(X^0, Y^0) + \mathbb{E}d^\alpha(Y^0, \mathbf{\Phi}_{\epsilon_t}(Y^0)) + \mathbb{E}d^\alpha(\mathbf{\Phi}_{\epsilon_t}(Y^0), \mathbf{\Phi}_{\epsilon_t}(X^0)) \\
&\overset{(a)}{\leqslant} d^\alpha(X^0, Y^0) + I(\alpha, Y^0) + Cr d^\alpha(Y^0, X^0) < \infty.
\end{aligned}$$

Inequality (a) comes from Condition C2 applied to $\mathbb{E}d^\alpha(\mathbf{\Phi}_{\epsilon_t}(Y^0), \mathbf{\Phi}_{\epsilon_t}(X^0))$. Then, by Inequality (3.4.2), we have for all $m \in \mathbb{N}$

$$\begin{aligned}
&\mathbb{E}\left[d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\mathbf{\Phi}}_{t,m+1}(X^0))\right] = \mathbb{E}\left(\mathbb{E}\left[d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\mathbf{\Phi}}_{t,m}(\mathbf{\Phi}_{\epsilon_{t-m}}(X^0)))|\epsilon_{t-m}\right]\right) \\
&\overset{\text{Condition C2}}{\leqslant} Cr^m \mathbb{E}\left[d^\alpha(X^0, \mathbf{\Phi}_{\epsilon_{t-m}}(X^0))\right] = I(\alpha, X^0) Cr^m.
\end{aligned}$$

Then by the Markov inequality, we have

$$\mathbb{P}\left[d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\mathbf{\Phi}}_{t,m+1}(X^0)) \geqslant r^{\frac{m}{2}}\right] \leqslant r^{-\frac{m}{2}} \mathbb{E}\left[d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\mathbf{\Phi}}_{t,m+1}(X^0))\right] \lesssim r^{\frac{m}{2}}.$$

Thus

$$\sum_{m=1}^{\infty} \mathbb{P}\left[d(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\mathbf{\Phi}}_{t,m+1}(X^0)) \geqslant r^{\frac{m}{2\alpha}}\right] \lesssim \sum_{m=1}^{\infty} r^{\frac{m}{2}} = \frac{r^{\frac{1}{2}}}{1 - r^{\frac{1}{2}}} < \infty.$$

Applying the Borel-Cantelli lemma, we have

$$\mathbb{P}\left[d(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\mathbf{\Phi}}_{t,m+1}(X^0)) \geqslant r^{\frac{m}{2\alpha}} \text{ infinitely often}\right] = 0.$$

Thus, we have that

$$\mathbb{P}\left[d(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\mathbf{\Phi}}_{t,m+1}(X^0)) \geqslant r^{\frac{m}{2\alpha}} \text{ happen finitely times}\right] = 1.$$

This implies

$$\mathbb{P}\left[d(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\mathbf{\Phi}}_{t,m+1}(X^0)) \overset{m\to\infty}{\longrightarrow} 0\right] = 1.$$

Thus $\widetilde{\mathbf{\Phi}}_{t,m}(X^0)$ is a Cauchy sequence in $(\mathcal{X}, d)$, almost surely. By the completeness of $(\mathcal{X}, d)$, there exists a $\widetilde{\boldsymbol{X}}_t \in \mathcal{X}$, such that $\widetilde{\mathbf{\Phi}}_{t,m}(X^0) \overset{m\to\infty}{\longrightarrow} \widetilde{\boldsymbol{X}}_t$ almost surely. Moreover, since for any fixed $t, t' \in \mathbb{Z}$, $\widetilde{\mathbf{\Phi}}_{t,m}(X^0) \overset{d}{=} \widetilde{\mathbf{\Phi}}_{t',m}(X^0)$, $\forall m \in \mathbb{N}$. Thus $\lim_{m\to\infty} \widetilde{\mathbf{\Phi}}_{t,m}(X^0) \overset{d}{=} \lim_{m\to\infty} \widetilde{\mathbf{\Phi}}_{t',m}(X^0)$, namely $\widetilde{\boldsymbol{X}}_t \overset{d}{=} \widetilde{\boldsymbol{X}}_{t'}$, almost surely.

■

3.4.3.2 Proof of Lemma 3.4.2

Under the conditions of Lemma 3.4.1, we have $d(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\boldsymbol{X}}_t) \overset{m\to\infty}{\longrightarrow} 0$ almost surely, with $\widetilde{\boldsymbol{X}}_t \in (\mathcal{X}, d)$, $\forall m \in \mathbb{N}, t \in \mathbb{Z}$. Thus $\mathbb{E}d(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\boldsymbol{X}}_t) \overset{m\to\infty}{\longrightarrow} 0$. On the other hand, Since $\mathbb{E}d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m+j}(X^0), \widetilde{\mathbf{\Phi}}_{t,m+j+1}(X^0)) \lesssim r^{m+j}$ for any fixed $m > 0, j \geqslant 0$, then

$$\mathbb{E}\sum_{j=0}^{n} d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m+j}(X^0), \widetilde{\mathbf{\Phi}}_{t,m+j+1}(X^0)) \lesssim \sum_{j=0}^{n} r^{m+j} \lesssim \sum_{j=0}^{\infty} r^{m+j} = \frac{r^m}{1-r}.$$

Thus, for any $m, n \in \mathbb{N}$,

$$\begin{aligned} \mathbb{E}d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\boldsymbol{X}}_t) &\leqslant \mathbb{E}\sum_{j=0}^{n} d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m+j}(X^0), \widetilde{\mathbf{\Phi}}_{t,m+j+1}(X^0)) \\ &\quad + \mathbb{E}d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m+n+1}(X^0), \widetilde{\boldsymbol{X}}_t) \\ &\lesssim r^m. \end{aligned} \tag{3.4.5}$$

Then, for any fixed $m \in \mathbb{N}, X \in \mathcal{X}$, we have

$$d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X), \widetilde{\boldsymbol{X}}_t) \leqslant d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X), \widetilde{\mathbf{\Phi}}_{t,m}(X^0)) + d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\boldsymbol{X}}_t).$$

Applying the expectation on both sides of the above inequality, we have

$$\begin{aligned} \mathbb{E}d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X), \widetilde{\boldsymbol{X}}_t) &\leqslant \mathbb{E}d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X), \widetilde{\mathbf{\Phi}}_{t,m}(X^0)) + \mathbb{E}d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\boldsymbol{X}}_t) \\ &\lesssim Cr^m d^\alpha(X, X^0) + \mathbb{E}d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \widetilde{\boldsymbol{X}}_t) \lesssim r^m. \end{aligned}$$

Following the same steps as in the proof of Lemma 3.4.1, we then have almost surely,

$$d(\widetilde{\mathbf{\Phi}}_{t,m}(X), \widetilde{\boldsymbol{X}}_t) \overset{m\to\infty}{\longrightarrow} 0, \ \forall t \in \mathbb{Z}.$$

■

3.4.3.3 Proof of Lemma 3.4.3

We would like to show that, for any fixed $t \in \mathbb{Z}$, $d(\widetilde{\boldsymbol{X}}_t, \boldsymbol{\Phi}_{\epsilon_t}(\widetilde{\boldsymbol{X}}_{t-1})) = 0$, almost surely. Firstly, we have for any $m \in \mathbb{N}$

$$\begin{aligned}\mathbb{E}d^\alpha(\widetilde{\boldsymbol{X}}_t, \boldsymbol{\Phi}_{\epsilon_t}(\widetilde{\boldsymbol{X}}_{t-1})) &\leqslant \mathbb{E}d^\alpha(\widetilde{\boldsymbol{X}}_t, \tilde{\boldsymbol{\Phi}}_{t,m}(X^0)) + \mathbb{E}d^\alpha(\tilde{\boldsymbol{\Phi}}_{t,m}(X^0), \boldsymbol{\Phi}_{\epsilon_t}(\widetilde{\boldsymbol{X}}_{t-1})) \\ &= \mathbb{E}d^\alpha(\widetilde{\boldsymbol{X}}_t, \tilde{\boldsymbol{\Phi}}_{t,m}(X^0)) + \mathbb{E}d^\alpha(\boldsymbol{\Phi}_{\epsilon_t} \circ \tilde{\boldsymbol{\Phi}}_{t-1,m-1}(X^0), \boldsymbol{\Phi}_{\epsilon_t}(\widetilde{\boldsymbol{X}}_{t-1})) \\ &\leqslant \mathbb{E}d^\alpha(\widetilde{\boldsymbol{X}}_t, \tilde{\boldsymbol{\Phi}}_{t,m}(X^0)) + Cr\mathbb{E}d^\alpha(\tilde{\boldsymbol{\Phi}}_{t-1,m-1}(X^0), \widetilde{\boldsymbol{X}}_{t-1}).\end{aligned}$$

Since $d(\widetilde{\boldsymbol{X}}_t, \tilde{\boldsymbol{\Phi}}_{t,m}(X^0)) \overset{m\to\infty}{\longrightarrow} 0$ almost surely, for any $t \in \mathbb{Z}$. Thus, the last bound above tends to 0 as $m \to \infty$. Thus $\mathbb{E}d^\alpha(\widetilde{\boldsymbol{X}}_t, \boldsymbol{\Phi}_{\epsilon_t}(\widetilde{\boldsymbol{X}}_{t-1})) = 0$, which implies $d(\widetilde{\boldsymbol{X}}_t, \boldsymbol{\Phi}_{\epsilon_t}(\widetilde{\boldsymbol{X}}_{t-1})) = 0$ almost surely.

■

3.4.3.4 Proof of Theorem 3.4.5

We first show that Equation (3.4.2) holds for any $\alpha' \in (0, \alpha)$. For any $X \in \mathcal{X}$, we have

$$\mathbb{E}d^{\alpha'}(\tilde{\boldsymbol{\Phi}}_{t,m}(X^0), \tilde{\boldsymbol{\Phi}}_{t,m}(X)) \leqslant \left(\mathbb{E}d^\alpha(\tilde{\boldsymbol{\Phi}}_{t,m}(X^0), \tilde{\boldsymbol{\Phi}}_{t,m}(X))\right)^{\frac{\alpha'}{\alpha}} \leqslant C^{\frac{\alpha'}{\alpha}} r^{\frac{\alpha' m}{\alpha}} d^{\alpha'}(X^0, X). \tag{3.4.6}$$

Let $\gamma = \min\{\alpha, \beta\}$, then for any $t, m \in \mathbb{N}$, we have

$$\begin{aligned}\mathbb{E}d^\gamma(\widetilde{\boldsymbol{X}}_t, \boldsymbol{S}_t) = \mathbb{E}d^\gamma(\widetilde{\boldsymbol{X}}_t, \tilde{\boldsymbol{\Phi}}_{t,m}(\boldsymbol{S}_{t-m})) &\leqslant \mathbb{E}d^\gamma(\widetilde{\boldsymbol{X}}_t, \tilde{\boldsymbol{\Phi}}_{t,m}(X^0)) + \mathbb{E}d^\gamma(\tilde{\boldsymbol{\Phi}}_{t,m}(X^0), \tilde{\boldsymbol{\Phi}}_{t,m}(\boldsymbol{S}_{t-m})) \\ &\overset{Inequality(3.4.6)}{\leqslant} \mathbb{E}d^\gamma(\widetilde{\boldsymbol{X}}_t, \tilde{\boldsymbol{\Phi}}_{t,m}(X^0)) + C^{\frac{\gamma}{\alpha}} r^{\frac{\gamma m}{\alpha}} \mathbb{E}d^\gamma(X^0, \boldsymbol{S}_{t-m}).\end{aligned}$$

Since $d(\widetilde{\boldsymbol{X}}_t, \tilde{\boldsymbol{\Phi}}_{t,m}(X^0)) \overset{m\to\infty}{\longrightarrow} 0$ almost surely, for any $t \in \mathbb{Z}$, we have $\mathbb{E}d^\gamma(\widetilde{\boldsymbol{X}}_t, \tilde{\boldsymbol{\Phi}}_{t,m}(X^0)) \overset{m\to\infty}{\longrightarrow} 0$ for any $t \in \mathbb{Z}$. On the other hand, since $\gamma < \beta$, for all $t \in \mathbb{Z}$, we have $\mathbb{E}d^\gamma(\boldsymbol{S}_t, X^0) \leqslant \mathbb{E}d^\gamma(\boldsymbol{S}_t, Z^0) + d^\gamma(X^0, Z^0) \leqslant \left(\mathbb{E}d^\beta(\boldsymbol{S}_t, Z^0)\right)^{\frac{\gamma}{\beta}} + d^\gamma(X^0, Z^0) < M^{\frac{\gamma}{\beta}} + d^\gamma(X^0, Z^0)$. Therefore, $\mathbb{E}d^\gamma(\widetilde{\boldsymbol{X}}_t, \boldsymbol{S}_t) = 0$, which implies $d(\widetilde{\boldsymbol{X}}_t, \boldsymbol{S}_t) = 0$, almost surely, for all $t \in \mathbb{Z}$.

■

3.4.3.5 Proof of Proposition 3.4.6

We first show that given $\mathbf{X} \overset{d}{=} \mathbf{X}^1 \overset{d}{=} \pi$, we have for any $t \in \mathbb{Z}$, $\mathbf{X}_m(\mathbf{X}) \overset{d}{=} \mathbf{X}_m(\mathbf{X}^1) \overset{d}{=} \mathbf{X}_t$, almost surely for any $m \in \mathbb{N}$. From Theorem 3.4.4, we have $\mathbf{X}_t \overset{d}{=} \mathbf{X}_{t-m} \overset{d}{=} \pi$, almost surely for any $m \in \mathbb{N}, t \in \mathbb{Z}$, and $\mathbf{X}_t = \tilde{\boldsymbol{\Phi}}_{t,m}(\mathbf{X}_{t-m})$, almost surely for any $m \in \mathbb{N}, t \in \mathbb{Z}$. Thus, we obtain that $\pi = \tilde{\boldsymbol{\Phi}}_{t,m}(\pi)$, almost surely, which implies $\mathbf{X}_m(\mathbf{X}) \overset{d}{=} \mathbf{X}_m(\mathbf{X}^1) \overset{d}{=} \pi = \mathbf{X}_t$.

Therefore,

$$\begin{aligned}\mathbb{E}d^\alpha(\mathbf{X}_m(\mathbf{X}), \mathbf{X}_m(\mathbf{X}^1)) &= d^\alpha(\mathbf{X}_m(\mathbf{X}), \mathbf{X}_m(X^0)) + d^\alpha(\mathbf{X}_m(X^0), \mathbf{X}_m(\mathbf{X}^1)) \\ &= 2\mathbb{E}d^\alpha(\mathbf{X}_t, \mathbf{X}_m(X^0)) \overset{\boldsymbol{\epsilon}_t i.i.d.}{=} 2\mathbb{E}d^\alpha(\mathbf{X}_t, \tilde{\boldsymbol{\Phi}}_{t,m}(X^0)) \\ &\overset{Inequality(3.4.5)}{\lesssim} r^m.\end{aligned}$$

Thus, Bound (3.4.4) is checked with s taken as r, which completes the proof. ■

3.4.3.6 Proof of Theorem 3.4.8

We first show that the Hilbert mean $\mathbb{E}[\mathbf{X}_t]$ for time series $\mathbf{X}_t \in (\mathcal{X}, \langle , \rangle)$, $t \in \mathbb{Z}$ does not depend on time t. We are thus led to show that, for all $t, t' \in \mathbb{Z}$, $\mathbb{E}[\mathbf{X}_t] = \mathbb{E}[\mathbf{X}_{t'}]$. By the definition of Hilbert mean, this is equivalent to show that

$$\mathbb{E}\langle \mathbf{X}_t, X\rangle = \mathbb{E}\langle \mathbf{X}_{t'}, X\rangle, \quad \forall X \in \mathcal{X}.$$

Firstly, we show that $\forall t \in \mathbb{Z}, X \in \mathcal{X}$, $\mathbb{E}\langle \mathbf{X}_t, X\rangle = \lim_{m\to\infty} \mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0), X\rangle$. We have

$$\begin{aligned}
&\mathbb{E}|\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0) - \mathbf{X}_t, X\rangle| \leqslant \mathbb{E}\|\widetilde{\mathbf{\Phi}}_{t,m}(X^0) - \mathbf{X}_t\|\|X\| \\
&\overset{(a)}{\leqslant} \|X\|(\mathbb{E}\|\widetilde{\mathbf{\Phi}}_{t,m}(X^0) - \mathbf{X}_t\|^\alpha)^{\frac{1}{\alpha}} = \|X\| \left(\mathbb{E}d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), \mathbf{X}_t)\right)^{\frac{1}{\alpha}} \overset{m\to\infty}{\longrightarrow} 0.
\end{aligned}$$

Inequality (a) comes from the condition $\alpha \geqslant 2$ and Jensen inequality. Thus, for any $t \in \mathbb{Z}$,

$$\lim_{m\to\infty} \mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0) - \mathbf{X}_t, X\rangle = 0.$$

On the other hand, since $\boldsymbol{\epsilon}_t$ are i.i.d., there is $\widetilde{\mathbf{\Phi}}_{t,m}(X^0) \overset{d}{=} \widetilde{\mathbf{\Phi}}_{t',m}(X^0)$, $\forall t, t' \in \mathbb{Z}, m \in \mathbb{N}$. Thus, we have for any $t, t' \in \mathbb{Z}, X \in \mathcal{X}$

$$\mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0), X\rangle = \mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t',m}(X^0), X\rangle, \ \forall m \in \mathbb{N}.$$

Then $\forall t \in \mathbb{Z}, X \in \mathcal{X}$

$$\lim_{m\to\infty} \mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0), X\rangle = \lim_{m\to\infty} \mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t',m}(X^0), X\rangle, \ \forall m \in \mathbb{N},$$

which implies $\forall t \in \mathbb{Z}, X \in \mathcal{X}$

$$\mathbb{E}\langle \mathbf{X}_t, X\rangle = \mathbb{E}\langle \mathbf{X}_{t'}, X\rangle.$$

We denote $\mathbb{E}[\mathbf{X}_t]$ by U. Next, since $\mathbb{E}\|\mathbf{X}_t\|^2 < \infty$ for all $t \in \mathbb{Z}$, the auto-covariance operator $\mathcal{G}_{t,t+h}$ is well-defined. We are now to show $\mathcal{G}_{t,t+h}$ is time-invariant, which is equivalent to show

$$\langle \mathcal{G}_{t,t+h}(X), Y\rangle = \langle \mathcal{G}_{t',t'+h}(X), Y\rangle, \quad \forall X, Y \in \mathcal{X},$$

by Definition 3.4.7, that is

$$\mathbb{E}\langle \mathbf{X}_t - U, X\rangle\langle \mathbf{X}_{t+h} - U, Y\rangle = \mathbb{E}\langle \mathbf{X}_{t'} - U, X\rangle\langle \mathbf{X}_{t'+h} - U, Y\rangle, \quad \forall X, Y \in \mathcal{X}. \tag{3.4.7}$$

Analogously, we show firstly that $\forall t, h \in \mathbb{Z}, X, Y \in \mathcal{X}$

$$\mathbb{E}\langle \mathbf{X}_t - U, X\rangle\langle \mathbf{X}_{t+h} - U, Y\rangle = \lim_{m\to\infty} \mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0) - U, X\rangle\langle \widetilde{\mathbf{\Phi}}_{t+h,m}(X^0) - U, Y\rangle. \tag{3.4.8}$$

We have

$$\begin{aligned}
&\mathbb{E}\langle \mathbf{X}_t - U, X\rangle\langle \mathbf{X}_{t+h} - U, Y\rangle - \mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0) - U, X\rangle\langle \widetilde{\mathbf{\Phi}}_{t+h,m}(X^0) - U, Y\rangle \\
\leqslant &\mathbb{E}\langle \mathbf{X}_t - U, X\rangle\langle \mathbf{X}_{t+h} - U, Y\rangle - \mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0) - U, X\rangle\langle \mathbf{X}_{t+h} - U, Y\rangle \\
&\qquad + \mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0) - U, X\rangle\langle \mathbf{X}_{t+h} - U, Y\rangle \\
&\qquad\qquad - \mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0) - U, X\rangle\langle \widetilde{\mathbf{\Phi}}_{t+h,m}(X^0) - U, Y\rangle \\
= &\mathbb{E}\langle \mathbf{X}_t - \widetilde{\mathbf{\Phi}}_{t,m}(X^0), X\rangle\langle \mathbf{X}_{t+h} - U, Y\rangle \\
&\qquad\qquad + \mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0) - U, X\rangle\langle \mathbf{X}_{t+h} - \widetilde{\mathbf{\Phi}}_{t+h,m}(X^0), Y\rangle \\
\leqslant &\|X\|\|Y\|\mathbb{E}\|\mathbf{X}_t - \widetilde{\mathbf{\Phi}}_{t,m}(X^0)\|\|\mathbf{X}_{t+h} - U\| \\
&\qquad\qquad + \|X\|\|Y\|\mathbb{E}\|\widetilde{\mathbf{\Phi}}_{t,m}(X^0) - U\|\|\mathbf{X}_{t+h} - \widetilde{\mathbf{\Phi}}_{t+h,m}(X^0)\| \\
\leqslant &\|X\|\|Y\|\mathbb{E}d(\mathbf{X}_t, \widetilde{\mathbf{\Phi}}_{t,m}(X^0))d(\mathbf{X}_{t+h}, U) \\
&\qquad\qquad + \|X\|\|Y\|\mathbb{E}d(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), U)d(\mathbf{X}_{t+h}, \widetilde{\mathbf{\Phi}}_{t+h,m}(X^0)).
\end{aligned}$$

Since $d(\mathbf{X}_t, \widetilde{\mathbf{\Phi}}_{t,m}(X^0)) \overset{m\to\infty}{\longrightarrow} 0$ almost surely, for any $t \in \mathbb{Z}$, then $d(\widetilde{\mathbf{\Phi}}_{t,m}(X^0), U) \overset{m\to\infty}{\longrightarrow} d(\mathbf{X}_t, U)$, for any $t \in \mathbb{Z}$. Thus, the last bound tends to 0, as $m \to \infty$, which implies Equation (3.4.8).

On the other hand, since $\boldsymbol{\epsilon}_t$ are i.i.d., thus $\forall t, h \in \mathbb{Z}, X, Y \in \mathcal{X}$ and $\forall m \in \mathbb{N}$

$$\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0) - U, X\rangle\langle \widetilde{\mathbf{\Phi}}_{t+h,m}(X^0) - U, Y\rangle \overset{d}{=} \langle \widetilde{\mathbf{\Phi}}_{t',m}(X^0) - U, X\rangle\langle \widetilde{\mathbf{\Phi}}_{t'+h,m}(X^0) - U, Y\rangle,$$

which follows

$$\mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t,m}(X^0) - U, X\rangle\langle \widetilde{\mathbf{\Phi}}_{t+h,m}(X^0) - U, Y\rangle = \mathbb{E}\langle \widetilde{\mathbf{\Phi}}_{t',m}(X^0) - U, X\rangle\langle \widetilde{\mathbf{\Phi}}_{t'+h,m}(X^0) - U, Y\rangle. \tag{3.4.9}$$

Take limit on m on both sides of Equation (3.4.9), we obtain Equation (3.4.7). Thus the auto-covariance is time-invariant.

■

3.5 Reproducing kernel Hilbert space and kernel ridge regression

In this section, we provide a concise lecture on the Reproducing kernel Hilbert space (RKHS) and RKHS-based ridge regression. We only present the RKHS as a space of real functions, which we consider as the approximation family of the regression function of $x_{it}, (i,t) \in \mathcal{N} \times \mathbb{Z}$ in Chapter 6. For the general RKHS as a space of maps, whose reproducing kernel takes values of linear continuous operator, we refer to Kadri et al. (2016). We start off by defining a kernel.

Definition 3.5.1. Let $\mathcal{X}$ be a set. $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel on $\mathcal{X}$ if k is symmetric: $k(x,y) = k(y,x)$, and k is positive definite, that is $\forall\, x_1, x_2, ..., x_n \in \mathcal{X}$, the *Gram matrix* K defined by $K_{ij} = k(x_i, x_j)$ is positive semi-definite (PSD).

We would like to point out the following property of a kernel.

Remark 3.5.2. Given kernel k_1 on $\mathcal{X}_1$ and kernel k_2 on $\mathcal{X}_2$, then the mapping k_1k_2 whose value is defined by

$$k_1k_2\left[(x_1,y_1),(x_2,y_2)\right] = k_1(x_1,x_2)k_2(y_1,y_2)$$

is a also kernel on $\mathcal{X}_1 \times \mathcal{X}_2$.

For the proof, see (Christmann and Steinwart, 2008, Lemma 4.6 p.114). This product rule allows us to define a huge variety of kernels. Next, we introduce the *reproducing kernel feature map* ϕ and its related *feature space* $\mathcal{H}_k$.

Given a kernel k, define the map ϕ as[†]

$$\begin{cases} \phi : \mathcal{X} & \to & \mathbb{R}^{\mathcal{X}} \\ x & \mapsto & \phi(x) = k(x,\cdot) \end{cases}$$

Consider the functional space H_k

$$H_k = \text{span}(\{\phi(x) : x \in \mathcal{X}\}) = \{f(\cdot) = \sum_{i=1}^{n} \alpha_i k(x_i,\cdot)\}, \quad \text{where} \quad \alpha_i \in \mathbb{R}.$$

We define the inner product between $f = \sum_{i=1}^{n} \alpha_i k(x_i,\cdot)$ and $g = \sum_{j=1}^{m} \beta_j k(y_j,\cdot)$ in H_k as

$$\langle f,g\rangle = \sum_{i,j} \alpha_i\beta_j k(x_i,y_j).$$

It is easy to check that this inner product is valid. The induced norm is then $\|f\|_{H_k}^2 = \langle f,f\rangle$. We can rewrite the inner product and the induced norm into matrix form as

$$\langle f,g\rangle = \alpha^t G\beta, \quad \|f\|_{\mathcal{H}_k}^2 = \alpha^t K\alpha,$$

where $\alpha = (\alpha_1,...,\alpha_n)^t$, $\beta = (\beta_1,...,\beta_m)^t$, $G \in \mathbb{R}^{n\times m}$ with $G_{ij} = k(x_i,x_j)$ and K is the Gram matrix of inputs $x_1,x_2,...,x_n$.

Finally, we complete the inner product space H_k to a Hilbert space, denoted by $\mathcal{H}_k$. $\mathcal{H}_k$ is the RKHS uniquely defined by reproducing kernel (RK) k, meaning that k can only be the RK of $\mathcal{H}_k$. Conversely, if $\mathcal{H}_k$ is a RKHS[‡], then it has the unique RK. Note that the name *reproducing* comes from the reproducing property of k

$$\langle f,k(z,\cdot)\rangle = \sum_{x} \alpha_x k(x,z) = f(z).$$

This way of building a RKHS follows the proof of *Moore-Aronszajn theorem* below.

Theorem 3.5.3. *(Moore-Aronszajn) Suppose k is a symmetric, positive definite kernel on a set $\mathcal{X}$. Then there is a unique Hilbert space of functions on $\mathcal{X}$ for which k is a reproducing kernel.*

[†]This part of the thesis is from the lecture note Bartlett (2008).

[‡]There is an equivalent definition of RKHS, which deos not rely on k: a RKHS is a Hilbert space of maps from $\mathcal{X}$ to Hilbert space $\mathcal{Y}$ if its evaluation operator is linear continuous for every $x \in \mathcal{X}$.

We can find that RKHS has many favorable attributes to work with. We point out some of them, which are shown in the following remarks.

Remark 3.5.4.

- The principle does not assume additional structure on the set $\mathcal{X}$, which facilitates us to expand the input set by cartesian product later on.
- The point x in $\mathcal{X}$ is mapped to a high-dimensional feature space $\mathcal{H}_k$ with the feature vector $\phi(x)$. The feature vector can even be infinite-dimensional, such as the one in $\mathcal{H}_{k_{rbf}}$.
- The norm $\|f\|^2_{\mathcal{H}_k}$ implies how fast\smooth the function varies over $\mathcal{X}$ with respect to the geometry defined by the kernel[§].
- The composition $f(x) = \sum_y \alpha_y k(y, x)$ enables us to only set up linear models in the feature space, however we are still able to find a nonlinear function of x with the required characteristics, since the mapping can be nonlinear. This fact permits a feasible search for nonlinear predictor of the input space $\mathcal{X}$.

After introducing the approximation family $\mathcal{H}_k$, we would like to find the best estimator of the regression function of $\boldsymbol{y} \in \mathbb{R}$ on $\boldsymbol{x} \in \mathcal{X}$ given the observations. To accomplish this, we first set up *kernel regression* (3.5.1), then solve the related optimization problem with the help of the *representer theorem*.

Definition 3.5.5. Given the observations (x_1, y_1), (x_2, y_2), ..., $(x_n, y_n) \in \mathcal{X} \times \mathbb{R}$, the best fitted predictor is given by

$$\hat{f} = \underset{f \in \mathcal{H}_k}{\arg\min} \sum_{i=1}^{n} C(f(x_i), y_i) + g(\|f\|_{\mathcal{H}_k}) \tag{3.5.1}$$

where $C(x, a)$ is a convex cost function, $\|f\|_{\mathcal{H}_k}$ is the induced norm of inner product $< \cdot, \cdot >_{\mathcal{H}_k}$, $g : [0, +\infty) \to \mathbb{R}$ is a strictly monotonically increasing real-valued function.

If we choose the cost function $C(x, a)$ to be $\|x - a\|^2_{\ell^2}$, $g(x)$ to be λx^2 with nonnegative λ, then we have the kernel ridge regression (3.5.2).

$$\hat{f} = \underset{f \in \mathcal{H}_k}{\arg\min} \sum_{i=1}^{n} \|y_i - f(x_i)\|^2_{\ell^2} + \lambda \|f\|^2_{\mathcal{H}_k}. \tag{3.5.2}$$

From the third remark of 3.5.4, we know the regularization term $\lambda\|f\|^2_{\mathcal{H}_k}$ endows the potential optimizer $\hat{f}$ with the variation characteristic that we want, whose degree is quantified by λ. Therefore, the nature of using kernel ridge regression to predict is to diffuse the observed information along the input space.

Because all functions in $\mathcal{H}_k$ has the linear combination form $\sum_x \alpha_x k(x, \cdot)$. Thus once we get the optimal predictor $\hat{f} = \sum_x \hat{\alpha}_x k(x, \cdot)$, we can predict the output of unseen input z as $\hat{f}(z) = \sum_x \hat{\alpha}_x k(x, z)$. Since we know the kernel value beforehand, it is only left to find the optimal value for each coefficient α_x. The representer theorem, Schölkopf et al. (2001), indicates that, the only nonzero coefficients are

[§]Consider the inequality $|f(x) - f(y)| = |\langle f, k(x, \cdot) - k(y, \cdot)\rangle| \leqslant \|f\|_{\mathcal{H}_k} \|\phi(x) - \phi(y)\|_{\mathcal{H}_k}$.

those corresponds to the observations, namely, essentially the solution of problem (3.5.1) has the representation

$$\hat{f} = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot).$$

Plugging this representation into equation (3.5.2), the problem will be reduced to a common optimization problem in finite Euclidean space. We reformulate the problem using matrix notation as

$$\hat{\alpha} = \arg\min_{\alpha \in \mathbb{R}^n} \|\mathrm{y} - K_n\alpha\|_{\ell^2}^2 + \lambda \alpha^t K_n \alpha,$$

where $\mathrm{y} = (y_1, ..., y_n)^t$ and K_n is the Gram matrix of inputs $x_1, x_2, ..., x_n$. Through linear algebra, we get $\hat{\alpha} = (K_n + \lambda \mathrm{Id})^{-1}\mathrm{y}$. Then the resulting nonlinear predictor is $\hat{f}(z) = \sum_{i=1}^{n} \hat{\alpha} k(x_i, z)$, with $\|\hat{f}\|_{\mathcal{H}_k}^2 = \mathrm{y}^t(K_n + \lambda \mathrm{Id})^{-1} K_n (K_n + \lambda \mathrm{Id})^{-1}\mathrm{y}$.

3.6 Dropout

In Chapter 6, we rely on the dropout technique to aggregate the prediction models of different missing set I's during one network training so as to evaluate the overall predictability of nodes. To understand the motivation and the principle of such training design, in this section, we review the dropout Srivastava et al. (2014) which is a regularization technique designated to prevent overfitting based on the model aggregation. For the background on netural networks, we refer to Bishop and Nasrabadi (2006, Chapter 5).

The best way to reduce overfitting is model combination, which means fitting all possible models first, and then for each test sample, to aggregate the predictions from all trained models as the final prediction. Obviously, for a class of complex models such as neural networks, independent training and explicit aggregation are too costly. Dropout is then a general training framework, purposed to enable such varying and aggregation of network models within one training. The mechanism of dropout makes a single trained neural network aggregate exponentially many network models of different architectures with respect to the number of neurons in an approximate way, and gives the ensemble prediction values on the output neurons for targets.

To be more precise, we illustrate this technique with an example. Figure 3.1 is a small neural network, which consists in one input layer, one hidden layer and one output layer. Their respective neuron values are denoted by $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2)$, $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2)$ and $\boldsymbol{z}$, where $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2)$ represents an arbitrary input sample. Without dropout, the network forward propagation of the model given on the leftmost in Figure 3.1 is

$$\boldsymbol{z} = \sigma_2(v_1\boldsymbol{y}_1 + v_2\boldsymbol{y}_2), \quad \boldsymbol{y}_1 = \sigma_1(u_{11}\boldsymbol{x}_1 + u_{21}\boldsymbol{x}_2), \quad \boldsymbol{y}_2 = \sigma_1(u_{12}\boldsymbol{x}_1),$$

where $v_1, v_2, u_{11}, u_{21}, u_{12}$ are weights, which are trainable parameters. In this example, we omit bias terms. If we apply dropout on the hidden layer, the forward propagation becomes

$$\begin{aligned} &\boldsymbol{y}_1 = \sigma_1(u_{11}\boldsymbol{x}_1 + u_{21}\boldsymbol{x}_2), \quad \boldsymbol{y}_2 = \sigma_1(u_{12}\boldsymbol{x}_1), \\ &\widetilde{\boldsymbol{y}}_1 = \boldsymbol{w}_1\boldsymbol{y}_1, \quad \widetilde{\boldsymbol{y}}_2 = \boldsymbol{w}_2\boldsymbol{y}_2, \quad \boldsymbol{z} = \sigma_2(v_1\widetilde{\boldsymbol{y}}_1 + v_2\widetilde{\boldsymbol{y}}_2) \end{aligned}$$

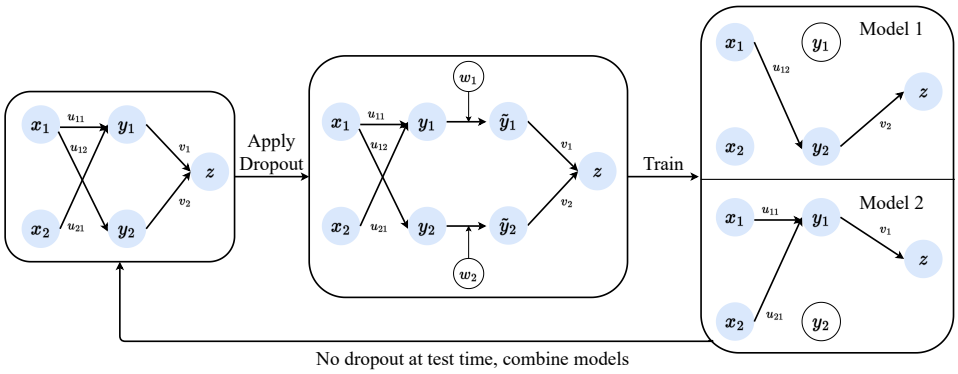

Figure 3.1: Training a network with dropout can be viewed as training a collection of network models which share weights. At test time, all neurons are turned on, leading to the aggregations of models. The final trained net without dropout outputs the aggregated prediction from these models.

where $\mathbf{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2)$ is the dropout vector. The components of the dropout vector are independent Bernoulli random variables, each of which has probability q of being 0, which is referred to as the *dropout rate*. The backpropagation is determined by the realization of $\mathbf{w}$ and gradient calculation. We denote the loss component of a sample as $J(\boldsymbol{z})$. Then, the associated gradient components of trainable parameters are computed using the *chain rule* as follows:

$$\frac{\partial J(v_1)}{\partial v_1} = \frac{\partial J(\boldsymbol{z})}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}(v_1)}{\partial v_1} = \frac{\partial J(\boldsymbol{z})}{\partial \boldsymbol{z}} \sigma_2'(v_1 \tilde{\boldsymbol{y}}_1 + v_2 \tilde{\boldsymbol{y}}_2) \boldsymbol{w}_1 \boldsymbol{y}_1,$$

$$\frac{\partial J(v_2)}{\partial v_2} = \frac{\partial J(\boldsymbol{z})}{\partial \boldsymbol{z}} \frac{\partial \boldsymbol{z}(v_2)}{\partial v_2} = \frac{\partial J(\boldsymbol{z})}{\partial \boldsymbol{z}} \sigma_2'(v_1 \tilde{\boldsymbol{y}}_1 + v_2 \tilde{\boldsymbol{y}}_2) \boldsymbol{w}_2 \boldsymbol{y}_2,$$

$$\frac{\partial J(u_{i1})}{\partial u_{i1}} = \frac{\partial J(\boldsymbol{z})}{\partial \boldsymbol{z}} \frac{\partial z(\boldsymbol{y}_1)}{\partial \boldsymbol{y}_1} \frac{\partial \boldsymbol{y}_1(u_{i1})}{\partial u_{i1}} = \frac{\partial J(\boldsymbol{z})}{\partial \boldsymbol{z}} \sigma_2'(v_1 \tilde{\boldsymbol{y}}_1 + v_2 \tilde{\boldsymbol{y}}_2) v_1 \boldsymbol{w}_1 \frac{\partial \boldsymbol{y}_1(u_{i1})}{\partial u_{i1}}, \; i = 1, 2,$$

$$\frac{\partial J(u_{12})}{\partial u_{12}} = \frac{\partial J(\boldsymbol{z})}{\partial \boldsymbol{z}} \frac{\partial z(\boldsymbol{y}_2)}{\partial \boldsymbol{y}_2} \frac{\partial \boldsymbol{y}_2(u_{12})}{\partial u_{12}} = \frac{\partial J(\boldsymbol{z})}{\partial \boldsymbol{z}} \sigma_2'(v_1 \tilde{\boldsymbol{y}}_1 + v_2 \tilde{\boldsymbol{y}}_2) v_2 \boldsymbol{w}_2 \frac{\partial \boldsymbol{y}_2(u_{12})}{\partial u_{12}}.$$

Therefore if $\boldsymbol{w}_1 = 0$ and $\boldsymbol{w}_2 = 1$, the gradient components of v_1, u_{11} and u_{21} are all 0. The gradient components of v_2, u_{12} are then calculated by backpropagating the loss component $J(\boldsymbol{z})$ through the model 1 in Figure 3.1. In practice, the dropout vector is re-sampled on a regular basis, such as each sample or each batch. The gradient used to update each parameter is then calculated as usual, that is the average over its gradient components computed from all the samples $t \in B_n$ used in an updating step n. For example

$$\frac{1}{|B_n|} \sum_{t \in B_n} \frac{\partial J(v_1; \mathrm{x}_t, \mathrm{w}_t)}{\partial v_1},$$

for v_1. B_n can have only one sample, but more commonly is a batch. When using the network in prediction (test time), all neurons are present with scaled-down weights, which indicates multiplying the outgoing weights of the dropout neurons by $1 - q$.

This ensures the expected neuron output in training remains the same as the actual one at test time.

Since our model re-samples the dropout vector at each batch, we present how the dropout does the model aggregation in this case with the example of Figure 3.1. B_n denotes a batch in the following, and we furthermore denote its dropout vector by $\mathrm{w}^{(n)}$. Because the dropout vector is identical for all samples in a batch, one batch contributes to the training of one specific network model. For example, sampling $\mathrm{w}^{(n)} = (0, 1)$ amounts to sampling and training model 1 with B_n. Then, sampling $\mathrm{w}^{(n+1)} = (1, 0)$ varies the architecture to model 2 in the next step with B_{n+1}. The parameters of a new model are initialized as the results of the preceding training phase. The ones connected to turned-on neurons are shifted towards the potential independent training results of this model. By doing this, the training needs to find the optimal weight values which are shared across models and gives the aggregated model prediction the best performance. Then when using the network in prediction (test time), no dropout leads to the aggregation of all models into the resulting trained network, which gives the overall predictions z_t for the target.

3.7 References on the involved convex optimization results

The work in this thesis uses several classical convex optimization results. To avoid the content redundancy of this manuscript, we refer to the following materials for the corresponding backgrounds.

Subdifferential, first order optimality condition Section $B.2$, Apidopoulos (2019)

Proximal operator, projection operator Section 1.1 and 1.2, Parikh and Boyd (2014)

Accelerated proximal gradient descent Section 4.3, Parikh and Boyd (2014)

Backtracking line search Section 9.2, Boyd et al. (2004)

Strong convexity Section 4, Zhou (2018)

Duality Section 5, Boyd et al. (2004)

Chapter 4

Online graph learning from matrix-variate time series

In this chapter, we develop the novel graph learning frameworks, which learns in an online fashion a Granger causal graph from matrix-variate time series $(\mathbf{x}_t)_t \in \mathbb{R}^{N \times F}$. The proposed approaches extend the existing works on the inference of causal graphs, relying on the VAR models, which we have reviewed in Section 2.2.1.1. In addition to learning the causality structure between stationary processes, in the graph learning domain, there is another main line of works, that is dedicated to the inference of conditional dependency structure between random variables. This line assumes that the observations $\mathbf{x}_t \in \mathbb{R}^N$ are independent and identically distributed from $\mathcal{N}(0, \Theta^{-1})$. The conditional dependency structure thus is encoded in the precision matrix Θ, which is promoted to be sparse during the estimation. The resulting models are known as the Gaussian graphical models (Meinshausen et al., 2006; Friedman et al., 2008). Moreover, Gaussian graphical model for the stationary process are studied in Bach and Jordan (2004) and Songsiri and Vandenberghe (2010), where they infer the conditional dependence graph from the inverse of its matrix-variate spectrum. This line of graph learning models has been firstly extended to matrix- (Kalaitzis et al., 2013) and tensor- (Greenewald et al., 2019; Wang et al., 2020) variate data. The extending works also rely on the KP/KS structure to impose a product graph structure in the precision matrix.

On the other hand, the developed graph learning frameworks are based on the novel matrix-variate AR(1) model, which is proposed by this work. At this point, we recall the other work in literature, Chen et al. (2021a), on the extension of the VAR(1) to the matrix-variate process, that we have mentioned in Section 2.2.1.2 of introduction. In contrast with our KS construction, Chen et al. (2021a) proposes to impose the KP structure in the coefficient matrix to extend VAR(1) model. The comparison of these two constructions has also been given in Section 2.2.1.2. Additional to this difference, our work is especially devised for the purpose of graph learning, thus we promote sparsity in our coefficient estimators and focus on the development of online inference, which is not considered by the work of Chen et al. (2021a).

In the rest of this chapter, we first introduce the novel matrix-variate AR model in Section 4.1. We also provide the closed form of the projection operator onto

the constrained coefficient space that is a key tool in the derivation of the learning approaches. In Section 4.2, we develop the main online graph learning frameworks. In Section 4.3, we propose the augmented data model to take into account the trends, and we adapt the previous frameworks to the inference of the augmented model. Lastly, we perform the numerical experiments using both synthetic and real data in Section 4.4. All proofs are gathered in the Appendix of this chapter.

Notations We present all the notations used in this chapter in advance.

vec	Vectorized representation of a matrix.
ivec	Inverse vectorized representation of a vector, such that $\text{ivec} \circ \text{vec} = id$.
$[\cdot]_{\cdot}$	Extraction by index. The argument in [] can be a vector or a matrix. For a vector, the index argument can be a scalar or an *ordered* list of integers. For example, $[\text{v}]_k$ extracts the k-th entry of v, while $[\text{v}]_K = ([\text{v}]_{k_i})_i$ extracts a sub-vector indexed by $K = (k_i)_i$ in order. For a matrix, the index argument can be a pair of scalars or a pair of *ordered* lists of integers. For example, $[M]_{k,k'}$ extracts the (k, k')-th entry of M, while $[M]_{K,K'} = ([M]_{k_i,k_j})_{i,j}$ extracts a sub-matrix indexed by $K = (k_i)_i$ in row order, and $K' = (k'_j)_j$ in column order. When $K = K'$, we denote $[M]_{K,K'}$ by $[M]_K$.
$[M]_{:,i}$	Extraction of the i-th column vector of matrix M.
$[M]_{i,:}$	Extraction of the i-th row vector of matrix M.
$\text{svec}(M)$	Vectorized representation of the upper diagonal part of matrix M, that is, $\left([\text{M}]_{1,2}, [\text{M}]_{1,3}, \cdots, [\text{M}]_{2,3}, \cdots\right)^\top$.
diag(M)	Diagonal vector of matrix M.

4.1 Causal product graphs and matrix-variate AR(1) models

We define formally the matrix-variate AR(1) model proposed in Section 2.2.1.3. The matrix-variate stochastic process $\mathbf{X}_t \in \mathbb{R}^{N \times F}$ is said to follow a matrix-variate AR(1) process if the multivariate process $\text{vec}(\mathbf{X}_t)$ is a VAR(1) process defined by A with the particular KS structure $\mathcal{K}_\mathcal{G}$

$$\begin{aligned}\mathcal{K}_\mathcal{G} \;=\; \big\{&\text{M} \in \mathbb{R}^{NF \times NF} : \exists\, \text{M}_\text{F} \in \mathbb{R}^{F \times F}, \text{M}_\text{N} \in \mathbb{R}^{N \times N}, \text{ such that,}\\ &\text{offd}(\text{M}) = \text{M}_\text{F} \oplus \text{M}_\text{N}, \text{ with, } \text{diag}(\text{M}_\text{F}) = 0,\ \text{diag}(\text{M}_\text{N}) = 0,\\ &\text{M}_\text{F} = \text{M}_\text{F}^\top,\ \text{M}_\text{N} = \text{M}_\text{N}^\top\big\}.\end{aligned}$$

By constraining $A \in \mathcal{K}_\mathcal{G}$, we impose the KS structure into the off-diagonal part of coefficient matrix A, modelling the total causality structure by a Cartesian product

graph $\mathcal{G}$ parameterized by the spatial graph $\mathcal{G}_{\mathrm{N}}$ and the feature graph $\mathcal{G}_{\mathrm{F}}$. The diagonal of A is free from the structure constraint. In return, we require no self-loops in the component graphs by imposing $\mathrm{diag}(\mathrm{M}_{\mathrm{F}}) = 0$, $\mathrm{diag}(\mathrm{M}_{\mathrm{N}}) = 0$. This is to primarily address the non-identifiability problem of Kronecker sum, since $AF \oplus AN = (AF + cI_F) \oplus (AN - cI_N)$ holds for any scalar c. On the other hand, a full parameterized diagonal adds the self-loops on all nodes of the total graph $\mathcal{G}$, which also brings more flexibility to the model.

Note that, the last constraint in $\mathcal{K}_{\mathcal{G}}$ requires the component graphs hence the product graph to be symmetric. This is because we notice that, the existing causal graphs are usually directed, which disables their further use in the methods, which require undirected graphs as prior knowledge, like kernel methods, and graph Fourier transform related methods. Therefore, we focus on learning undirected graphs. Nevertheless, we stress that the derived approaches do not depend on the specific structure of coefficient, and thus can be adapted to, for example, the relaxed constraint set without the symmetry assumption, which may suit more the applications in econometrics, as suggested by one reviewer in the recent feedback.

We then focus on the analysis of a stationary process. We recall the stationarity condition for the VAR(1) model in Theorem 3.2.2. We also need the conditions of CLT 3.2.3, which permits the consistent estimator and its Wald test. We conclude all these assumptions by data generating model (4.1.4).

We assume the samples $\mathbf{X}_t \in \mathbb{R}^{N \times F}$ are generated by the following model given the initial sample $\mathbf{X}_0$

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + \mathbf{z}_t, \text{ with } A \in \mathcal{K}_{\mathcal{G}}, \ \|A\|_2 < 1, \quad t \in \mathbb{N}^+, \tag{4.1.4}$$

where $\mathbf{x}_t = \mathrm{vec}\,(\mathbf{X}_t)$, $\|A\|_2$ equals the largest singular value of A, $\mathbf{z}_t \in \mathbb{R}^{NF} \sim \mathrm{IID}\,(0, \Sigma)$ is white noise with a non-singular covariance structure Σ and bounded fourth moments, and $\mathbf{x}_0 = \sum_{j=0}^{\infty} A^j \mathbf{z}_{t-j}$. Note that we assume the process mean is zero in this section and derive the main frameworks of Section 4.2. In Section 4.3, we will study the model with a non-zero, even time-variant, process mean, namely, a process trend, to adapt the main frameworks to the online learning.

Applying ivec( ) on both sides of Model (4.1.4), we can obtain its matrix representation

$$\mathbf{X}_t = D \circ \mathbf{X}_{t-1} + A_{\mathrm{N}}\mathbf{X}_{t-1} + \mathbf{X}_{t-1}A_{\mathrm{F}}^{\top} + \mathbf{Z}_t, \tag{4.1.5}$$

where $\circ$ is Hadamard product, $D \in \mathbb{R}^{N \times F} = \mathrm{ivec}(\mathrm{diag}(A))$, A_{N} and A_{F} are the adjacency matrices such that $\mathrm{offd}(A) = A_{\mathrm{F}} \oplus A_{\mathrm{N}}$, and $\mathbf{Z}_t = \mathrm{ivec}(\mathbf{z}_t)$. In model (4.1.5), $A_{\mathrm{N}}\mathbf{X}_{t-1}$ describes the spatial dependency, where each column of $\mathbf{X}_{t-1}$ can be viewed as a graph signal on the same spatial graph $\mathcal{G}_{\mathrm{N}}$. Similarly, each row of $\mathbf{X}_{t-1}$ can be seen as a graph signal on the feature graph $\mathcal{G}_{\mathrm{F}}$.

4.2 Online Graph Learning

In this section, we detail the two online learning frameworks presented in Sections 2.2.1.3 and 2.2.1.4, which infer A_{N} with sparsity and A_{F} respectively in low- and

high- dimensional regimes. In the following section, we firstly introduce the tools on constraint set $\mathcal{K}_\mathcal{G}$, which is crucial to derive the proposed frameworks.

4.2.1 Orthonormal basis and projection operator of $\mathcal{K}_\mathcal{G}$

$\mathcal{K}_\mathcal{G}$ defined as Equation (4.1.1) is a linear space of dimension $NF + \frac{1}{2}F(F-1) + \frac{1}{2}N(N-1)$. We now endow $\mathcal{K}_\mathcal{G}$ with the Frobenius matrix inner product, that is $\langle B, C\rangle_\mathbf{F} = tr(B^\top C)$. An orthogonal basis of $\mathcal{K}_\mathcal{G}$ is then given in the following Lemma.

Lemma 4.2.1. *The set of matrices U_k, $k \in K := \{1, \dots, NF + \frac{1}{2}F(F-1) + \frac{1}{2}N(N-1)\}$, defined below form an orthogonal basis of $\mathcal{K}_\mathcal{G}$*

$$U_k = \begin{cases} E_k, & k \in K_\mathrm{D} := \{1, ..., NF\}, \\ \frac{1}{2N} E_k \otimes I_N, & k \in K_\mathrm{F} := NF + \{1, \dots, \frac{1}{2}F(F-1)\}, \\ \frac{1}{2F} I_F \otimes E_k, & k \in K_\mathrm{N} := NF + \frac{1}{2}F(F-1) + \{1, \dots, \frac{1}{2}N(N-1)\}, \end{cases}$$

where, when $k \in K_\mathrm{D}$, $E_k \in \mathbb{R}^{NF \times NF}$, with $[E_k]_{i,j} = 1$, if $i = j = k$, otherwise 0, when $k \in K_\mathrm{F}$, $E_k \in \mathbb{R}^{F \times F}$ is almost a zero matrix except

$$\begin{cases} [E_k]_{1,2} = [E_k]_{2,1} = 1, \text{if } k = NF + 1, \\ [E_k]_{1,3} = [E_k]_{3,1} = 1, \text{if } k = NF + 2, \\ [E_k]_{2,3} = [E_k]_{3,2} = 1, \text{if } k = NF + F, \\ \vdots \\ [E_k]_{F-1,F} = [E_k]_{F,F-1} = 1, \text{if } k = NF + \frac{1}{2}F(F-1), \end{cases}$$

when $k \in K_\mathrm{N}$, $E_k \in \mathbb{R}^{N \times N}$ is almost a zero matrix except

$$\begin{cases} [E_k]_{1,2} = [E_k]_{2,1} = 1, \text{if } k = NF + \frac{1}{2}F(F-1) + 1, \\ [E_k]_{1,3} = [E_k]_{3,1} = 1, \text{if } k = NF + \frac{1}{2}F(F-1) + 2, \\ [E_k]_{N-1,N} = [E_k]_{N,N-1} = 1, \text{if } k = NF + \frac{1}{2}F(F-1) + \frac{1}{2}N(N-1). \end{cases}$$

In Figure 4.1, we give an example of this orthogonal basis of $\mathcal{K}_\mathcal{G}$ for $N = 3, F = 2$, where U_k are visualized with respect to their non-zero entries. We can find that each U_k relates to one variable of $\mathrm{diag}(M), M_\mathrm{F}$ and M_N, and characterises how it contributes to the structure of M by repeating at multiple entries. Thus, taking the inner product with U_k actually calculates the average value of an arbitrary matrix over these entries. This is important to understand how to project an arbitrary matrix onto $\mathcal{K}_\mathcal{G}$.

It is easy to verify that $\langle U_k, U_{k'}\rangle_\mathbf{F} = 0$ for any $k \neq k'$ in K, and $(U_k)_k$ spans $\mathcal{K}_\mathcal{G}$. Thus the normalized matrices $U_k/\|U_k\|_\mathbf{F}, k \in K$ forms an orthonormal basis of $\mathcal{K}_\mathcal{G}$. We introduce the orthogonal projection onto $\mathcal{K}_\mathcal{G}$ and provide an explicit formula to calculate it using $(U_k/\|U_k\|_\mathbf{F})_k$ in Proposition 4.2.2.

Proposition 4.2.2. *For a matrix $A \in \mathbb{R}^{NF \times NF}$, its orthogonal projection onto $\mathcal{K}_\mathcal{G}$ is defined by*

$$Proj_\mathcal{G}(B) = \underset{M \in \mathcal{K}_\mathcal{G}}{\arg\min} \|B - M\|_\mathbf{F}^2 .$$

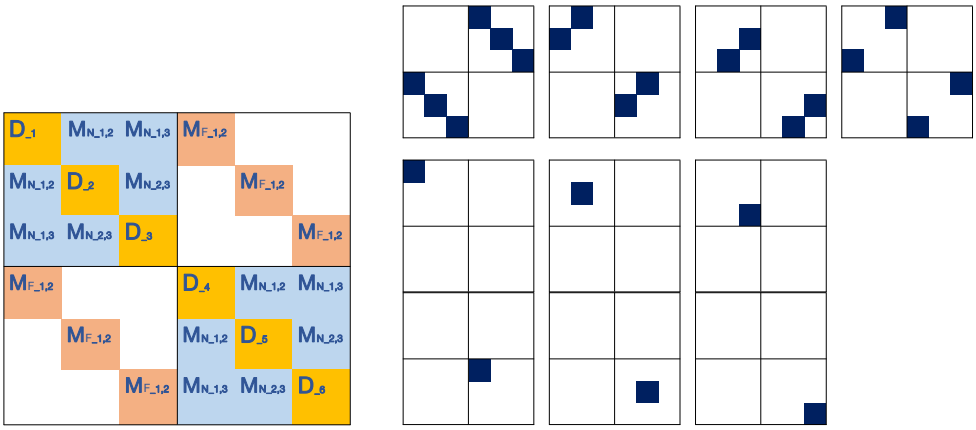

Figure 4.1: *Matrices $(U_k)_k$ as entry locators, which characterise the structure of $\mathcal{K}_\mathcal{G}$.*

Then given the orthonormal basis $U_k/\|U_k\|_\mathbf{F}, k \in K$, the projections can be calculated explicitly as

$$Proj_\mathcal{G}(B) = \sum_{k\in K} \langle U_k, B\rangle \frac{1}{\|U_k\|_\mathbf{F}^2} U_k.$$

The projection is very straightforward to understand. To obtain a variable in $\mathrm{diag}(M), M_F$ and M_N related to U_k, we use $\langle U_k, B\rangle$ to calculate the average value of B as explained previously. Then this average value is repeated at the corresponding entries to construct the structure, by multiplying locator $U_k/\|U_k\|_\mathbf{F}^2$.

Furthermore the orthogonality of the basis implies the direct sum

$$\mathcal{K}_\mathcal{G} = \mathcal{K}_\mathrm{D} \oplus \mathcal{K}_\mathrm{F} \oplus \mathcal{K}_\mathrm{N}, \tag{4.2.1}$$

where $\mathcal{K}_\mathrm{D}$, $\mathcal{K}_\mathrm{F}$, and $\mathcal{K}_\mathrm{N}$ are respectively spanned by $(U_k)_{k\in K_\mathrm{D}}$, $(U_k)_{k\in K_\mathrm{F}}$, and $(U_k)_{k\in K_\mathrm{N}}$. Given the construction of $(U_k)_k$, Equation (4.2.1) actually reveals the product graph decomposition, note that equally we have

$$\mathcal{K}_\mathrm{D} = \{\mathrm{M} \in \mathbb{R}^{NF\times NF} : \mathrm{offd}(M) = 0\},$$

$$\begin{aligned}\mathcal{K}_\mathrm{F} \;=\; &\{\mathrm{M} \in \mathbb{R}^{NF\times NF} : \exists \mathrm{M_F} \in \mathbb{R}^{F\times F}, \text{ such that,}\\ &\mathrm{M} = \mathrm{M_F} \otimes I_N, \text{ with, } \mathrm{diag}(\mathrm{M_F}) = 0, \mathrm{M_F} = \mathrm{M_F^\top}\},\end{aligned}$$

$$\begin{aligned}\mathcal{K}_\mathrm{N} \;=\; &\{\mathrm{M} \in \mathbb{R}^{NF\times NF} : \exists \mathrm{M_N} \in \mathbb{R}^{N\times N}, \text{ such that,}\\ &\mathrm{M} = I_F \otimes \mathrm{M_N}, \text{ with, } \mathrm{diag}(\mathrm{M_N}) = 0, \mathrm{M_N} = \mathrm{M_N^\top}\}.\end{aligned}$$

The projection onto these subspaces can also be computed analogously

$$\mathrm{Proj}_\mathrm{D}(B) = \sum_{k\in K_\mathrm{D}} \langle U_k, B\rangle \frac{1}{\|U_k\|_\mathbf{F}^2} U_k, \text{ that is the diagonal part of } B.$$

$$\mathrm{Proj}_{\mathrm{F}}(B) = \sum_{k\in K_{\mathrm{F}}} \langle U_k, B\rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k = \left[\sum_{k\in K_{\mathrm{F}}} \langle U_k, B\rangle E_k\right] \otimes I_N,$$

$$\mathrm{Proj}_{\mathrm{N}}(B) = \sum_{k\in K_{\mathrm{N}}} \langle U_k, B\rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k = I_F \otimes \left[\sum_{k\in K_{\mathrm{N}}} \langle U_k, B\rangle E_k\right].$$

We use $\mathrm{Proj}_{\mathcal{G}_{\mathrm{F}}}(B)$ and $\mathrm{Proj}_{\mathcal{G}_{\mathrm{N}}}(B)$ to denote the small matrices $\sum_{k\in K_{\mathrm{F}}}\langle U_k, B\rangle E_k$ and $\sum_{k\in K_{\mathrm{F}}}\langle U_k, B\rangle E_k$, with an extra subscript $\mathcal{G}$, with which we will represent the proposed estimators of AF, AN in the following sections. Finally, we have

$$\mathrm{Proj}_{\mathcal{G}}(B) = \mathrm{Proj}_{\mathrm{D}}(B) + \mathrm{Proj}_{\mathcal{G}_{\mathrm{F}}}(B) \oplus \mathrm{Proj}_{\mathcal{G}_{\mathrm{N}}}(B). \tag{4.2.2}$$

4.2.2 Approach 1: Projected OLS estimators and Wald test

In the low dimensional regime, VAR model (4.1.4) can be estimated by the ordinary least squares (OLS) method, defined in (3.2.9). We denote the OLS estimator of A by $\check{\mathbf{A}}_t$, and we use $\widehat{\mathbf{A}}_t$ to denote the proposed projected OLS estimator. Assume that we start receiving samples $\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_t$ from time $\tau = 1$, we recall the OLS estimator for an intercept-free VAR(1) model

$$\check{\mathbf{A}}_t = \widehat{\mathbf{\Gamma}}_t(1)\left[\widehat{\mathbf{\Gamma}}_t(0)\right]^{-1},$$

where

$$\widehat{\mathbf{\Gamma}}_t(0) = \frac{1}{t}\sum_{\tau=1}^{t} \mathbf{x}_{\tau-1}\mathbf{x}_{\tau-1}^\top$$

and

$$\widehat{\mathbf{\Gamma}}_t(1) = \frac{1}{t}\sum_{\tau=1}^{t} \mathbf{x}_{\tau}\mathbf{x}_{\tau-1}^\top$$

are respectively the consistent estimators of auto-covariance matrices $\Gamma(0)$ and $\Gamma(1)$, with $\Gamma(h) = \mathbb{E}\left(\mathbf{x}_t\mathbf{x}_{t-h}^\top\right)$, $h \geqslant 0$. Moreover, the additional conditions in Model (4.1.4) permit the asymptotic properties

1. $\check{\mathbf{A}}_t \xrightarrow{p} A$,
2. $\sqrt{t}\,\mathrm{vec}(\check{\mathbf{A}}_t - A) \xrightarrow{d} \mathcal{N}(0, \Sigma_{ols})$, where $\Sigma_{ols} = [\Gamma(0)]^{-1} \otimes \Sigma$.

However, due to model misspecification and limited samples, $\check{\mathbf{A}}_t$ will not have the same structure as $A \in \mathcal{K}_{\mathcal{G}}$. Therefore, the projection of $\check{\mathbf{A}}_t$ onto $\mathcal{K}_{\mathcal{G}}$ needs to be performed, which leads to the projected OLS estimator:

$$\widehat{\mathbf{A}}_t := \mathrm{Proj}_{\mathcal{G}}(\check{\mathbf{A}}_t).$$

Given the representation (4.2.2), it is natural to define the estimators of $\mathrm{diag}(A)$, A_{F}, and A_{N} by $\mathrm{Proj}_{\mathrm{D}}(\check{\mathbf{A}}_t)$, $\mathrm{Proj}_{\mathcal{G}_{\mathrm{F}}}(\check{\mathbf{A}}_t)$, and $\mathrm{Proj}_{\mathcal{G}_{\mathrm{N}}}(\check{\mathbf{A}}_t)$, respectively, denoted by $\widehat{\mathbf{A}_{\mathrm{D}}}_{,t}$, $\widehat{\boldsymbol{A}_{\mathbf{F}}}_{,t}$, and $\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t}$. We now establish the Wald test with $\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t}$ to identify the sparsity structure of the true A_{N}. To this end, we provide the CLT in Theorem 4.2.3.

Theorem 4.2.3. *Assume samples* $\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_t$ *satisfy the assumptions of Model* (4.1.4)*, then the CLT holds for* $\widehat{\boldsymbol{A}_{\mathbf{N},t}}$*, as* $t \to +\infty$*,*

$$\sqrt{t}\, svec(\widehat{\boldsymbol{A}_{\mathbf{N},t}} - A_{\mathrm{N}}) \xrightarrow{d} \mathcal{N}(0, \Sigma_{\mathrm{N}}),$$

where

$$\Sigma_{\mathrm{N}} = \sum_{k,k' \in K_{\mathrm{N}}} vec(U_k)^\top \Sigma_{ols} vec(U_{k'}) \left(svec(E_k) svec(E_{k'})^\top\right).$$

The proof is done through applying Cramér-Wold theorem on $\sqrt{t}\,\mathrm{svec}(\widehat{\boldsymbol{A}_{\mathbf{N},t}} - A_{\mathrm{N}})$, given the linearity of $\mathrm{Proj}_{\mathcal{G}_{\mathrm{N}}}(\breve{\mathbf{A}}_t)$ and the CLT of classical OLS estimator $\breve{\mathbf{A}}_t$. For details, see Appendix 4.5.1, where we also derive a CLT for $\widehat{\mathbf{A}}_t$. It is straightforward to understand the asymptotic distribution of $\widehat{\boldsymbol{A}_{\mathbf{N},t}}$. The asymptotic covariance between its two entries is assigned the mean of covariance values $\mathrm{vec}(U_k)^\top \Sigma_{ols} \mathrm{vec}(U_{k'})$, following the construction of the corresponding estimators $\langle U_k, \breve{\mathbf{A}}_t \rangle$ and $\langle U_{k'}, \breve{\mathbf{A}}_t \rangle$ as averages as well.

Based on this large sample result, we now test the nullity of P given variables $[A_{\mathrm{N}}]_{i_k,j_k}$, $k = 1, ..., P$, with $i_k < j_k$ as

$$H_0 : \alpha = 0 \text{ versus } H_1 : \alpha \neq 0,$$

where $\alpha \in \mathbb{R}^P := \left(\cdots, [A_{\mathrm{N}}]_{i_k,j_k}, \cdots\right)^\top$. The test statistic is given by

$$\lambda_{W,t} = t\, \hat{\boldsymbol{\alpha}}_t^\top \left[\widehat{\boldsymbol{\Sigma}}_{W,t}\right]^{-1} \hat{\boldsymbol{\alpha}}_t, \tag{4.2.3}$$

where $\hat{\boldsymbol{\alpha}}_t \in \mathbb{R}^P := \left(\cdots, \left[\widehat{\boldsymbol{A}_{\mathbf{N},t}}\right]_{i_k,j_k}, \cdots\right)^\top$, and $\widehat{\boldsymbol{\Sigma}}_{W,t} \in \mathbb{R}^{P \times P}$ is defined as

$$\left[\widehat{\boldsymbol{\Sigma}}_{W,t}\right]_{k,k'} = \mathrm{vec}(U_{h_k})^\top \widehat{\boldsymbol{\Sigma}}_{ols,t} \mathrm{vec}(U_{h_{k'}}),$$

such that U_{h_k} is the matrix corresponding to variable $[A_{\mathrm{N}}]_{i_k,j_k}$,

$$\widehat{\boldsymbol{\Sigma}}_{ols,t} = \left[\widehat{\boldsymbol{\Gamma}}_t(0)\right]^{-1} \otimes \widehat{\boldsymbol{\Sigma}}_t, \text{ and } \widehat{\boldsymbol{\Sigma}}_t = \widehat{\boldsymbol{\Gamma}}_t(0) - \widehat{\boldsymbol{\Gamma}}_t(1) \left[\widehat{\boldsymbol{\Gamma}}_t(0)\right]^{-1} \widehat{\boldsymbol{\Gamma}}_t(1)^\top,$$

are the consistent estimators. CLT (4.2.3) implies the following result.

Corollary 4.2.4. *The asymptotic distribution of* $\lambda_{W,t}$ *as* $t \to +\infty$ *is given by*

$$\lambda_{W,t} \xrightarrow{d} \chi^2(P), \quad \textit{Under } H_0.$$

Remark 4.2.5. We can also consider the test statistic $\lambda_{F,t} := \lambda_{W,t}/P$ as suggested in Lütkepohl (2005, Section 3.6) in conjunction with the critical values from $F(P, t - NF - 1)$.

The Wald test above theoretically completes the approach. In practice, we propose to test the p entries of the smallest estimate magnitudes, jointly each time, as p grows from 1 to possibly largest value $|K_{\mathrm{N}}|$. Specifically, for a given estimation $\widehat{\boldsymbol{A}_{\mathbf{N}},t}$, we first sort its entries such that

$$|[\widehat{\boldsymbol{A}_{\mathbf{N}},t}]_{i_1,j_1}| \leqslant |[\widehat{\boldsymbol{A}_{\mathbf{N}},t}]_{i_2,j_2}| \leqslant \ldots \leqslant |[\widehat{\boldsymbol{A}_{\mathbf{N}},t}]_{i_{|K_{\mathrm{N}}|},j_{|K_{\mathrm{N}}|}}|.$$

Then, we set up the sequence of joint tests

$$H_0(1),\ H_0(2),\ ...,\ H_0(|K_{\mathrm{N}}|),\ \text{where } H_0(p) : \left([A_{\mathrm{N},t}]_{i_1,j_1}, \cdots, [A_{\mathrm{N},t}]_{i_p,j_p}\right)^\top = 0,$$

We perform these tests sequentially until $H(p_0+1)$ is rejected for some p_0. Lastly, we replace the entries $[\widehat{\boldsymbol{A}_{\mathbf{N}},t}]_{i_1,j_1}, ..., [\widehat{\boldsymbol{A}_{\mathbf{N}},t}]_{i_{p_0},j_{p_0}}$ with 0 in $\widehat{\boldsymbol{A}_{\mathbf{N}},t}$ as the final estimate of A_{N}. Note that searching for p_0 resembles root-finding, since the output from each point p is binary. Thus, the search can be accelerated by using the bisection, with the maximal number of steps about $\log_2(|K_{\mathrm{N}}|)$.

The previous procedure is performed at the t-th iteration, given the OLS estimator $\breve{\mathbf{A}}_t$ and the consistent estimator $\widehat{\boldsymbol{\Sigma}}_{ols,t}$. When the new sample $\mathbf{x}_{t+1}$ comes, $\breve{\mathbf{A}}_{t+1}$ and $\widehat{\boldsymbol{\Sigma}}_{ols,t+1}$ can be calculated efficiently by applying *Sherman Morrison formula* on $[\widehat{\boldsymbol{\Gamma}}_t(0)]^{-1}$. The pseudo code is given in Algorithm 2.

4.2.3 Approach 2: Structured matrix-variate Lasso and homotopy Algorithms

Another estimator of A can be obtained by minimizing the following Lasso problem

$$\mathbf{A}(t,\lambda) = \underset{A\in\mathcal{K}_{\mathcal{G}}}{\arg\min}\, L_{\lambda,t}(A),\ \text{where } L_{\lambda,t}(A) = \frac{1}{2t}\sum_{\tau=1}^{t} \|\mathbf{x}_\tau - A\mathbf{x}_{\tau-1}\|_{\ell_2}^2 + \lambda F\,\|A_{\mathrm{N}}\|_{\ell_1}\,. \tag{4.2.4}$$

We recall that only a subset of entries of A is penalized by the ℓ_1 norm. To solve this Lasso problem with structure constraint and the partial penalization, in an off-line fashion, we can adopt for example the projected gradient descent as formulated in Equation (2.2.4). Our goal is then to derive the homotopy algorithms, in order to update $\mathbf{A}(t,\lambda_t)$ to $\mathbf{A}(t+1,\lambda_{t+1})$ quickly upon the arrival of $\mathbf{x}_{t+1}$.

We recall again the standard Lasso (2.2.2) in Equation (4.2.5). We formulate it differently here as in the homotopy algorithm literature.

$$\boldsymbol{\theta}(t,\lambda) = \underset{\theta\in\mathbb{R}^d}{\arg\min}\, \frac{1}{2}\|\mathbf{y} - \mathbf{X}\theta\|_{\ell_2}^2 + t\lambda\|\theta\|_{\ell_1}, \tag{4.2.5}$$

where $\mathbf{y} = (\boldsymbol{y}_\tau)_{\tau=1}^t$, $\mathbf{X} = (\mathbf{x}_\tau)_{\tau=1}^t$, and $(\mathbf{x}_\tau, \boldsymbol{y}_\tau) \in \mathbb{R}^{d+1}$ is the sample at time τ. A family of homotopy algorithms are devised for Lasso (4.2.5) to quickly update the existing solution when the problem evolves. We refer to Malioutov et al. (2005) which updates the solution from $\boldsymbol{\theta}(t,\lambda_1)$ to $\boldsymbol{\theta}(t,\lambda_2)$, Garrigues and Ghaoui (2008) which proposes an homotopy algorithm updating $\boldsymbol{\theta}(t,\lambda_2)$ to $\boldsymbol{\theta}(t+1,\lambda_2)$. The adaptive λ selection rules, which update λ at the arrival of new samples are also studied in Monti et al. (2018); Garrigues and Ghaoui (2008).

As mentioned in Section 2.2.1.4, the existing homotopy algorithms do not adapt to novel Lasso (4.2.4), due to the calculation of optimality condition. Thus, we first calculate the optimality conditions of the novel Lasso in Section 4.2.3.1.

4.2.3.1 Optimality Conditions

The key point in deriving the optimality conditions arising from the variational problem (4.2.4) is to transfer the structure of A onto the data vector $\mathbf{x}_{\tau-1}$, using an orthonormal basis of $\mathcal{K}_{\mathcal{G}}$. We introduce the auxiliary variable A^0, such that $A = \mathrm{Proj}_{\mathcal{G}}(A^0)$, and rewrite Problem (4.2.4) with respect to A^0

$$\min_{A^0 \in \mathbb{R}^{NF \times NF}} \frac{1}{2t} \sum_{\tau=1}^{t} \left\| \mathbf{x}_\tau - \sum_{k \in K} \langle U_k, A^0 \rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k \mathbf{x}_{\tau-1} \right\|_{\ell_2}^2 + \lambda \left\| \sum_{k \in K_{\mathrm{N}}} \langle U_k, A^0 \rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k \right\|_{\ell_1}. \tag{4.2.6}$$

Problem (4.2.6) is weakly convex, since a minimizer of (4.2.4) can be projected from infinitely many minimizers of (4.2.6). We still use $L_{\lambda,t}$ to denote the objective function above. A minimizer $\mathbf{A}^0$ of (4.2.6) satisfies the optimality conditions

$$\begin{aligned} 0 \in \frac{\partial L_{\lambda,t}}{\partial A^0} = & \sum_{k,k' \in K} \langle U_k, U_{k'} \widehat{\mathbf{\Gamma}}_t(0) \rangle \langle \frac{1}{\|U_{k'}\|_{\mathbf{F}}^2} U_{k'}, \mathbf{A}^0 \rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k \\ & - \sum_{k \in K} \langle U_k, \widehat{\mathbf{\Gamma}}_t(1) \rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k + \lambda \sum_{k \in K_{\mathrm{N}}} \partial \left| \langle U_k, \mathbf{A}^0 \rangle \right| \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k. \end{aligned} \tag{4.2.7}$$

Assume $\mathbf{A}^0$ is a matrix which satisfies Equation (4.2.7), hence a minimizer of Problem (4.2.6). Then $\mathbf{A} = \mathrm{Proj}_{\mathcal{G}}(\mathbf{A}^0)$ is a minimizer of Lasso (4.2.4). We denote its active set $\{k \in K_{\mathrm{N}} : \langle U_k, \mathbf{A}^0 \rangle \neq 0\}$ by K_{N}^1, that is all the non-zero variables of $\boldsymbol{A}_{\mathbf{N}}$, and its non-active set by K_{N}^0, that is $K_{\mathrm{N}} \backslash K_{\mathrm{N}}^1$. Since $\{U_k\}_{k \in K}$ is an orthogonal family, Equation (4.2.7) is equivalent to

$$0 = \sum_{k \in K_{\mathrm{D}} \bigcup K_{\mathrm{F}}} \left[\sum_{k' \in K} \langle U_k, U_{k'} \widehat{\mathbf{\Gamma}}_t(0) \rangle \langle \frac{1}{\|U_{k'}\|_{\mathbf{F}}^2} U_{k'}, \mathbf{A}^0 \rangle - \langle U_k, \widehat{\mathbf{\Gamma}}_t(1) \rangle \right] \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k, \tag{4.2.8}$$

$$\begin{aligned} 0 = \sum_{k \in K_{\mathrm{N}}^1} & \left[\sum_{k' \in K} \langle U_k, \widehat{\mathbf{\Gamma}}_t(0) \rangle \langle \frac{1}{\|U_{k'}\|_{\mathbf{F}}^2} U_{k'}, \mathbf{A}^0 \rangle - \langle U_k, \widehat{\mathbf{\Gamma}}_t(1) \rangle \right] \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k \\ & + \lambda \sum_{k \in K_{\mathrm{N}}^1} \mathrm{sign} \langle U_k, \mathbf{A}^0 \rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k. \end{aligned} \tag{4.2.9}$$

$$\begin{aligned} 0 = \sum_{k \in K_{\mathrm{N}}^0} & \left[\sum_{k' \in K} \langle U_k, U_{k'} \widehat{\mathbf{\Gamma}}_t(0) \rangle \langle \frac{1}{\|U_{k'}\|_{\mathbf{F}}^2} U_{k'}, \mathbf{A}^0 \rangle - \langle U_k, \widehat{\mathbf{\Gamma}}_t(1) \rangle \right] \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k \\ & + \lambda \sum_{k \in K_{\mathrm{N}}^0} \partial \left| \langle U_k, \mathbf{A}^0 \rangle \right| \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k, \text{ where } \partial \left| \langle U_k, \mathbf{A}^0 \rangle \right| \in [-1, 1] \end{aligned} \tag{4.2.10}$$

To furthermore derive the optimality conditions of Lasso (4.2.4) in terms of $\mathbf{A}$, we introduce the projections onto sub-spaces $\mathcal{K}_{\mathrm{N}^1} := \mathrm{span}\{U_k : k \in K_{\mathrm{N}}^1\}$ and $\mathcal{K}_{\mathrm{N}^0} := \mathrm{span}\{U_k : k \in K_{\mathrm{N}}^0\}$, denoted respectively by $\mathrm{Proj}_{\mathrm{N}^1}$ and $\mathrm{Proj}_{\mathrm{N}^0}$. Note that Equation (4.2.1) in fact admits

$$\mathcal{K}_{\mathcal{G}} = \bigoplus_{k \in K} \mathrm{span}\{U_k\}.$$

Thus

$$\mathrm{Proj}_{\mathrm{N}^1}(B) = \sum_{k\in K_{\mathrm{N}}^1} \langle U_k, B\rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k = I_F \otimes \left[\sum_{k\in K_{\mathrm{N}}^1} \langle U_k, B\rangle E_k\right],$$

and

$$\mathrm{Proj}_{\mathrm{N}^0}(B) = \sum_{k\in K_{\mathrm{N}}^0} \langle U_k, B\rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k = I_F \otimes \left[\sum_{k\in K_{\mathrm{N}}^0} \langle U_k, B\rangle E_k\right].$$

Then Equations (4.2.8), (4.2.9), and (4.2.10) are equivalent respectively to

$$\mathrm{Proj}_{\mathrm{DF}}\left(\mathbf{A}\widehat{\mathbf{\Gamma}}_t(0) - \widehat{\mathbf{\Gamma}}_t(1)\right) = 0,$$

$$\mathrm{Proj}_{K_{\mathrm{N}}^1}\left(\mathbf{A}\widehat{\mathbf{\Gamma}}_t(0) - \widehat{\mathbf{\Gamma}}_t(1)\right) + \lambda I_F \otimes \left[\sum_{k\in K_{\mathrm{N}}^1} \mathrm{sign}\langle E_k, \boldsymbol{A}_{\mathbf{N}}\rangle\, E_k\right] = 0,$$

$$\mathrm{Proj}_{K_{\mathrm{N}}^0}\left(\mathbf{A}\widehat{\mathbf{\Gamma}}_t(0) - \widehat{\mathbf{\Gamma}}_t(1)\right) + \lambda I_F \otimes \left[\sum_{k\in K_{\mathrm{N}}^0} \partial|\langle E_k, \boldsymbol{A}_{\mathbf{N}}\rangle|\, E_k\right] = 0,$$

where $\mathbf{A} \in \mathcal{K}_{\mathcal{G}}$, $\mathrm{Proj}_{\mathrm{DF}} = \mathrm{Proj}_{\mathrm{D}} + \mathrm{Proj}_{\mathrm{F}}$, and $\partial|\langle E_k, \boldsymbol{A}_{\mathbf{N}}\rangle| \in [-1, 1]$. The optimality conditions above are an extension of those for standard Lasso (4.2.5), while the former are furthermore refined to the unpenalized variables versus the penalized variables.

4.2.3.2 Homotopy from $\mathbf{A}(t, \lambda_1)$ to $\mathbf{A}(t, \lambda_2)$

To develop the homotopy algorithm for the change in λ value, we need to get the formulas of the active variables indexed by K_{N}^1 in terms of λ. To this end, we need to rely on representation (4.2.8), (4.2.9), and (4.2.10), directly in terms of each variable $\langle U_k, \mathbf{A}^0\rangle$. We firstly reorganize all the model variables into a vector

$$\mathbf{a}^s := \left(\langle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k, \mathbf{A}^0\rangle\right)_{k\in K} = \left(\langle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k, \mathbf{A}\rangle\right)_{k\in K}.$$

Note that $\mathbf{a}^s$ is in fact the scaled Lasso solution by the time the variable repeats. Then optimality conditions (4.2.8), (4.2.9), and (4.2.10) are essentially a system of linear equations of unknown $\mathbf{a}^s$, with λ in the coefficients. Thus we aim to firstly represent this linear system in vector form, in order to solve the unknowns. We shall introduce the following notations.

Notations of Proposition 4.2.6. $\mathbf{\Gamma}_0 \in \mathbb{R}^{|K|\times|K|}$ is a large matrix defined as

$$[\mathbf{\Gamma}_0]_{k,k'} = \langle U_k, U_{k'}\widehat{\mathbf{\Gamma}}_t(0)\rangle.$$

$\boldsymbol{\gamma}_1 \in \mathbb{R}^{|K|}$ is a long vector defined as

$$[\boldsymbol{\gamma}_1]_k = \langle U_k, \widehat{\mathbf{\Gamma}}_t(1)\rangle.$$

$\mathbf{w} \in \mathbb{R}^{|K^1|}$ is a long vector where $[\mathbf{w}]_k$ is defined as

$$\begin{cases} = 0, & k \in K_{\mathrm{D}} \bigcup K_{\mathrm{F}}, \\ = \operatorname{sign}[\mathbf{a}^s]_k, & k \in K_{\mathrm{N}}^1, \\ \in [-1, 1], & k \in K_{\mathrm{N}}^0. \end{cases}$$

We define $K^1 := K_{\mathrm{D}} \bigcup K_{\mathrm{F}} \bigcup K_{\mathrm{N}}^1$, that are all the *non-zero* variables. Note that except the computational coincidence, the variables in $K_{\mathrm{D}} \bigcup K_{\mathrm{F}}$ are usually non-zero. Then we denote the extractions

$$\begin{aligned} &\boldsymbol{\Gamma}_0^1 = [\boldsymbol{\Gamma}_0]_{K^1}, \boldsymbol{\Gamma}_0^0 = [\boldsymbol{\Gamma}_0]_{K_{\mathrm{N}}^0, K^1}, \boldsymbol{\gamma}_1^1 = [\boldsymbol{\gamma}_1]_{K^1}, \boldsymbol{\gamma}_1^0 = [\boldsymbol{\gamma}_1]_{K_{\mathrm{N}}^0}, \\ &\mathbf{a}_1^s = [\mathbf{a}^s]_{K^1}, \mathbf{w}_1 = [\mathbf{w}]_{K^1}, \mathbf{w}_0 = [\mathbf{w}]_{K_{\mathrm{N}}^0}. \end{aligned} \tag{4.2.11}$$

We can endow any orders to the elements in K^1, K_{N}^0 to extract the rows/columns/entries above, only if the orders are used consistently to all the extractions. With these notations, we now can retrieve a system of linear equations from Equations (4.2.8), (4.2.10), (4.2.9) of unknowns $\mathbf{a}_1^s$. Each equation is obtained by equating the entries of one U_k. The resulting system is given in Proposition 4.2.6.

Proposition 4.2.6. *A minimizer of Lasso problem (4.2.4) satisfies the linear system*

$$\begin{cases} \boldsymbol{\Gamma}_0^1 \mathbf{a}_1^s - \boldsymbol{\gamma}_1^1 + \lambda \mathbf{w}_1 = 0, \\ \boldsymbol{\Gamma}_0^0 \mathbf{a}_1^s - \boldsymbol{\gamma}_1^0 + \lambda \mathbf{w}_0 = 0. \end{cases}$$

The representation of the optimality conditions in Equation (4.2.6) are similar to those of classical Lasso (Garrigues and Ghaoui, 2008; Malioutov et al., 2005), where $\boldsymbol{\Gamma}_0$, $\boldsymbol{\gamma}_1$ with the embedded structures correspond to $\mathbf{X}^\top \mathbf{X}$, $\mathbf{X}^\top \mathbf{y}$ in the optimality conditions of standard Lasso. However in our case, the non-zero and sign pattern are only with respect to the entries of A_{N}, thus $\mathbf{w}_1$, which is the equivalent of sign vector, has $|K_{\mathrm{D}}| + |K_{\mathrm{F}}|$ zeros.

Suppose that $\mathbf{A}(t, \lambda)$ is the unique solution for a fixed λ of the optimization problem (4.2.4), then we invert $\boldsymbol{\Gamma}_0^1$ in Proposition 4.2.6 and get the formulas of $\mathbf{a}_1^s$

$$\begin{cases} \mathbf{a}_1^s = \left[\boldsymbol{\Gamma}_0^1\right]^{-1} (\boldsymbol{\gamma}_1^1 - \lambda \mathbf{w}_1) \\ \lambda \mathbf{w}_0 = \boldsymbol{\gamma}_1^0 - \boldsymbol{\Gamma}_0^0 \mathbf{a}_1^s. \end{cases} \tag{4.2.12}$$

Formula (4.2.12) is determined by the active set and the sign pattern of the optimal solution at λ. It shows that $\mathbf{a}_1^s$ is a piecewise linear function of λ, while $\mathbf{w}_0$ is also a piecewise smooth function.

Therefore, with the assumptions that $[\mathbf{a}^s]_{K_{\mathrm{N}}^1} \neq 0$ (element-wise), and $|[\mathbf{w}]_{K_{\mathrm{N}}^0}| < 1$ (element-wise), due to continuity properties, there exists a range (λ_l, λ_r) containing λ, such that for any $\lambda' \in (\lambda_l, \lambda_r)$, element-wise, $[\mathbf{a}^s]_{K_{\mathrm{N}}^1}$ remains nonzero with the signs unchanged, and $[\mathbf{w}]_{K_{\mathrm{N}}^0}$ remains in $(-1, 1)$. Hence, Formula (4.2.12) is the closed form of all the optimal solutions $\mathbf{A}(t, \lambda')$, for $\lambda' \in (\lambda_l, \lambda_r)$. λ_l, λ_r are taken as the closest critical points to λ. Each critical point is a λ value which makes either an $[\mathbf{a}^s]_k$, $k \in K_{\mathrm{N}}^1$ become zero, or a $[\mathbf{w}]_k$, $k \in K_{\mathrm{N}}^0$ reach 1 or -1. By letting

$[\mathbf{a}^s]_k = 0$, $k \in K_{\mathrm{N}}^1$ and $[\mathbf{w}]_k = \pm 1$, $k \in K_{\mathrm{N}}^0$ in Formula (4.2.12), we can compute all critical values. We now use k_i to denote the orders of K^1, K_{N}^0 that we used in the extraction (4.2.11). The critical values are then given by

$$\begin{aligned}
\lambda_{k_i}^0 &= \left[\left[\mathbf{\Gamma}_0^1\right]^{-1}\boldsymbol{\gamma}_1^1\right]_i \Big/ \left[\left[\mathbf{\Gamma}_0^1\right]^{-1}\mathbf{w}_1\right]_i, \quad k_i \in K^1 \text{ such that } k_i \in K_{\mathrm{N}}^1, \\
\lambda_{k_i}^+ &= \frac{\left[\boldsymbol{\gamma}_1^0 - \mathbf{\Gamma}_0^0\left[\mathbf{\Gamma}_0^1\right]^{-1}\boldsymbol{\gamma}_1^1\right]_i}{\left[1 - \mathbf{\Gamma}_0^0\left[\mathbf{\Gamma}_0^1\right]^{-1}\mathbf{w}_1\right]_i}, \quad k_i \in K_{\mathrm{N}}^0, \\
\lambda_{k_i}^- &= \frac{\left[\boldsymbol{\gamma}_1^0 - \mathbf{\Gamma}_0^0\left[\mathbf{\Gamma}_0^1\right]^{-1}\boldsymbol{\gamma}_1^1\right]_i}{\left[-1 - \mathbf{\Gamma}_0^0\left[\mathbf{\Gamma}_0^1\right]^{-1}\mathbf{w}_1\right]_i}, \quad k_i \in K_{\mathrm{N}}^0.
\end{aligned} \tag{4.2.13}$$

Thus, the closet critical points from both sides are

$$\begin{aligned}
\lambda_l := \max\big\{ &\max\{\lambda_k^0, k \in K_{\mathrm{N}}^1 : \lambda_k^0 < \lambda\}, \\
&\max\{\lambda_k^+, k \in K_{\mathrm{N}}^0 : \lambda_k^+ < \lambda\}, \max\{\lambda_k^-, k \in K_{\mathrm{N}}^0 : \lambda_k^- < \lambda\}\big\}, \\
\lambda_r := \min\big\{ &\min\{\lambda_k^0, k \in K_{\mathrm{N}}^1 : \lambda_k^0 > \lambda\}, \\
&\min\{\lambda_k^+, k \in K_{\mathrm{N}}^0 : \lambda_k^+ > \lambda\}, \min\{\lambda_k^-, k \in K_{\mathrm{N}}^0 : \lambda_k^- > \lambda\}\big\}.
\end{aligned} \tag{4.2.14}$$

If $\lambda_l = \varnothing$ then $\lambda_l := 0$, while if $\lambda_r = \varnothing$ then $\lambda_r := +\infty$. After λ' leaves the region by adding or deleting one variable to or from the active set, we update in order the corresponding entry in $\mathbf{w}$, K^1, K_{N}^0, and the solution formula (4.2.12) (Sherman Morrison formula for one rank update of $[\mathbf{\Gamma}_0^1]^{-1}$) to calculate the boundary of the new region as before. We proceed in this way until we reach the region covering the λ value at which we would like to calculate the Lasso solution, and use Formula (4.2.12) in this final region to compute the $\mathbf{a}_1^s$ with the desired λ value. Lastly, we retrieve the matrix-form optimal solution based on $\mathbf{a}_1^s$ and the latest K^1. This completes the first homotopy algorithm. For detailed algorithm, see the appendix of this chapter.

4.2.3.3 Homotopy from $\mathbf{A}(t, \lambda)$ to $\mathbf{A}(t+1, \lambda)$

In Garrigues and Ghaoui (2008), a continuous variable μ is introduced in the standard Lasso formulation (4.2.5) leading to the optimization Problem (4.2.15) below

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\theta\|_{\ell_2}^2 + \frac{1}{2}(\mu \boldsymbol{y}_{t+1} - \mu \mathbf{x}_{t+1}^\top \theta)^2 + t\lambda\|\theta\|_{\ell_1}, \tag{4.2.15}$$

in order to let the problem of learning from t samples evolve to that of learning from $t+1$ samples, as μ goes from 0 to 1. Therefore, representing the Lasso solution as a continuous function of μ permits the development of homotopy algorithm, which computes the path $\boldsymbol{\theta}(t, \lambda)$ to $\boldsymbol{\theta}(t+1, \frac{t}{t+1}\lambda)$.

This homotopy algorithm is derived based on the fact that, the term of new sample will only result in a rank-1 update in the covariance matrix as $\mathbf{X}^\top\mathbf{X} + \mu^2 \mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top$, because only 1 response variable is present. Thus, the corresponding matrix inverse in the closed form of optimal solution can be still expressed as an explicit function of μ using the Sherman Morrison formula, which furthermore allows the calculation

of critical points of μ. However, for the matrix-variate Lasso (4.2.4), a new sample will cause NF rank change* in $\mathbf{\Gamma}_0$, that is the number of response variables in the Lasso problem †. To formally understand this change, we rewrite $\mathbf{\Gamma}_0$ as the sum of t reorganized samples analogous to usual $\widehat{\mathbf{\Gamma}}_t(0)$

$$\mathbf{\Gamma}_0 = \frac{1}{t}\sum_{\tau=1}^{t} \widetilde{\mathbf{X}}_{\tau-1}\widetilde{\mathbf{X}}_{\tau-1}^\top, \text{ where } \widetilde{\mathbf{X}}_{\tau-1} \in \mathbb{R}^{|K|\times NF} \text{ with } [\widetilde{\mathbf{X}}_{\tau-1}]_{k,i} = [U_k]_{i,:}\mathbf{x}_{\tau-1},$$

note that a new $\mathbf{x}_{t+1}$ corresponds to the change $\widetilde{\mathbf{X}}_t\widetilde{\mathbf{X}}_t^\top$ in $\mathbf{\Gamma}_0$, which is a rank NF matrix. Thus it is impossible to express $\left[\mathbf{\Gamma}_0^1\right]^{-1}$ as an explicit and simple function of one single μ. However, note that each column (rank) $[\widetilde{\mathbf{X}}_t]_{:,i}$ corresponds to introducing new sample of one response variable $\boldsymbol{x}_{t+1,i} := [\mathbf{x}_{t+1}]_i$ at node i in $\mathcal{G}$, by rewriting the incremental term of Lasso (4.2.4)

$$\begin{aligned}\|\mathbf{x}_{t+1} - A\mathbf{x}_t\|_{\ell_2}^2 &= \left\|\mathbf{x}_{t+1} - \sum_{k\in K}\langle U_k, A^0\rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k\mathbf{x}_t\right\|_{\ell_2}^2 \\ &= \sum_{i=1}^{NF}\left(\boldsymbol{x}_{t+1,i} - \sum_{k\in K}\langle U_k, A^0\rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2}[U_k]_{i,:}\mathbf{x}_t\right)^2\end{aligned}$$

Therefore, we propose to introduce NF continuous variables $\mu_1, ..., \mu_{NF}$ in Lasso (4.2.4), and to consider the following problem

$$\mathbf{A}_{\lambda,t}(\mu_1, ..., \mu_{NF}) = \arg\min_{A\in\mathcal{K}_\mathcal{G}} L_{\lambda,t}(\mu_1, ..., \mu_{NF}),$$

$$\begin{aligned}\text{where } \; L_{\lambda,t}(\mu_1, ..., \mu_{NF}) = &\frac{1}{2(t+1)}\sum_{\tau=1}^{t}\|\mathbf{x}_\tau - A\mathbf{x}_{\tau-1}\|_{\ell_2}^2 + \lambda F\|A_{\mathrm{N}}\|_{\ell_1} \\ &+ \frac{1}{2(t+1)}\sum_{i=1}^{NF}\mu_i\left(\boldsymbol{x}_{t+1,i} - \sum_{k\in K}\langle U_k, A^0\rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2}[U_k]_{i,:}\mathbf{x}_t\right)^2.\end{aligned} \tag{4.2.16}$$

Given solution $\mathbf{A}(t,\lambda)$, we first apply the homotopy Algorithm of Section 4.2.3.2 on it with $\lambda_1 = \lambda$ and $\lambda_2 = \frac{t+1}{t}\lambda$ to change the constant before the old data term from $\frac{1}{t}$ to $\frac{1}{t+1}$. Then, we have $\mathbf{A}(t, \frac{t+1}{t}\lambda) = \mathbf{A}_{\lambda,t}(0, ..., 0)$ and $\mathbf{A}(t+1,\lambda) = \mathbf{A}_{\lambda,t}(1, ..., 1)$. We let evolve the optimization problem (4.2.4) from time t to $t+1$ by sequentially varying all μ_i from 0 to 1, along the paths

$$L_{\lambda,t}(0, 0, ..., 0) \to L_{\lambda,t}(1, 0, ..., 0) \to L_{\lambda,t}(1, 1, ..., 1) = L_{\lambda,t+1}.$$

Proposition 4.2.7. *A minimizer* $\mathbf{A}_{\lambda,t}(..., 1, \mu_i, 0, ...)$ *of* $\min_{A\in\mathcal{K}_\mathcal{G}} L_{\lambda,t}(..., 1, \mu_i, 0, ...)$ *satisfies the linear system*

$$\begin{cases}\mathbf{\Gamma}_0^1(\mu_i)\mathbf{a}_1^s - \boldsymbol{\gamma}_1^1(\mu_i) + (1+\frac{1}{t})\lambda\mathbf{w}_1 = 0 \\ \mathbf{\Gamma}_0^0(\mu_i)\mathbf{a}_1^s - \boldsymbol{\gamma}_1^0(\mu_i) + (1+\frac{1}{t})\lambda\mathbf{w}_0 = 0,\end{cases}$$

*On the other hand, this implies that $\mathbf{\Gamma}_0$ will quickly become non-singular from the initial time, as new samples $\mathbf{x}_\tau$ come in.

†More general, a new sample will cause a rank-NF update in the corresponding matrix $I_{NF}\otimes\widehat{\mathbf{\Gamma}}_t(0)$ in Lasso (2.2.2).

where $\mathbf{a}^s, K_{\mathrm{N}}^0, K_{\mathrm{N}}^1, K^1, \mathbf{w}$ *are with respect to* $\mathbf{A} = \mathbf{A}_{\lambda,t}(..., 1, \mu_i, 0, ...)$*, defining furthermore the extractions through* (4.2.11)*,*

$$\mathbf{\Gamma}_0(\mu_i) = \mathbf{\Gamma}_0 + \frac{1}{t}\sum_{n=1}^{i-1}[\widetilde{\mathbf{X}}_t]_{:,n}[\widetilde{\mathbf{X}}_t]_{:,n}^\top + \frac{\mu_i}{t}[\widetilde{\mathbf{X}}_t]_{:,i}[\widetilde{\mathbf{X}}_t]_{:,i}^\top,$$

and

$$\boldsymbol{\gamma}_1(\mu_i) = \boldsymbol{\gamma}_1 + \frac{1}{t}\sum_{n=1}^{i-1}\boldsymbol{x}_{t+1,n}[\widetilde{\mathbf{X}}_t]_{:,n} + \frac{\mu_i}{t}\boldsymbol{x}_{t+1,i}[\widetilde{\mathbf{X}}_t]_{:,i},$$

with $\mathbf{\Gamma}_0, \boldsymbol{\gamma}_1$ *are the same ones as in Proposition 4.2.6.*

The optimal conditions given in Proposition 4.2.7 show that, each path only relates to the one rank change: $\frac{\mu_i}{t}[\widetilde{\mathbf{X}}_t]_{:,i}[\widetilde{\mathbf{X}}_t]_{:,i}^\top$, for the latest updated $\mathbf{\Gamma}_0$. Thus we can apply the Sherman Morrison formula on $\left[\mathbf{\Gamma}_0^1(\mu_i)\right]^{-1}$ to retrieve the smooth function of μ_i, and express $\mathbf{a}_1^s$ and $\mathbf{w}_0$ as smooth functions of μ_i, which furthermore makes the calculation of the critical points of μ_i explicit. To leverage these continuity properties, we still assume $[\mathbf{a}^s]_{K_{\mathrm{N}}^1} \neq 0$ (element-wise), and $|\,[\mathbf{w}]_{K_{\mathrm{N}}^0}\,| < 1$ (element-wise). For the algorithm of path $\mathbf{A}_{\lambda,t}(0, ..., 0)$ to $\mathbf{A}_{\lambda,t}(1, ..., 1)$, it is sufficient to impose such assumption only on $\mathbf{A}_{\lambda,t}(0, ..., 0)$. By arguing as in Section 4.2.3.2, we can derive the homotopy algorithm for the whole data path. The detailed Algorithm 5 is given in the appendix of this chapter.

4.2.3.4 Update from λ_t to λ_{t+1}

Given the previous solution $\mathbf{A}(t, \lambda_t)$, one way to select the hyperparameter value λ is to introduce the empirical objective function (Monti et al., 2018; Garrigues and Ghaoui, 2008), which takes the form

$$f_{t+1}(\lambda) = \frac{1}{2}\|\mathbf{x}_{t+1} - \mathbf{A}(t, \lambda)\mathbf{x}_t\|_{\ell_2}^2,$$

and to employ the updating rule

$$\lambda_{t+1} = \lambda_t - \eta \frac{\mathrm{d}f_{t+1}(\lambda)}{\mathrm{d}\lambda}\Big|_{\lambda=\lambda_t},$$

where η is the step size. For convenience, we write $\frac{\mathrm{d}f_{t+1}(\lambda)}{\mathrm{d}\lambda}\big|_{\lambda=\lambda_t}$ as $\frac{\mathrm{d}f_{t+1}(\lambda_t)}{\mathrm{d}\lambda}$. Analogously, we adopt the notation $\frac{\mathrm{d}\mathbf{A}(t,\lambda_t)}{\mathrm{d}\lambda}$ to denote the derivative with respect to λ, taken at value $\lambda = \lambda_t$. The objective function f_{t+1} can be interpreted as an one step prediction error on unseen data. Since the Lasso solution is piece-wise linear with respect to λ, it follows that when λ is not a critical point, the derivative can be calculated as

$$\begin{aligned}\frac{\mathrm{d}f_{t+1}(\lambda_t)}{\mathrm{d}\lambda} &= \left\langle \mathbf{G}_t, \frac{\mathrm{d}\mathbf{A}(t,\lambda_t)}{\mathrm{d}\lambda} \right\rangle \\ &= \left\langle \mathrm{Proj}_{\mathcal{G}}(\mathbf{G}_t), \frac{\mathrm{d}\mathbf{A}(t,\lambda_t)}{\mathrm{d}\lambda} \right\rangle = -\left[\mathbf{a}_1^{\mathbf{G}_t}\right]^\top \left[\mathbf{\Gamma}_0^1\right]^{-1}\mathbf{w}_1,\end{aligned}$$

where $\mathbf{a}_1^{\mathbf{G}_t} \in \mathbb{R}^{|K^1|}$ is defined as $\left(\mathbf{a}_1^{\mathbf{G}_t}\right)_i = \langle U_k, \mathbf{G}_t \rangle$, $k_i \in K^1$, with K^1, $\mathbf{w}_1$, $\left[\mathbf{\Gamma}_0^1\right]^{-1}$ associated with $\mathbf{A}(t, \lambda_t)$, and

$$\mathbf{G}_t = \left(\mathbf{A}(t, \lambda_t)\mathbf{x}_t - \mathbf{x}_{t+1}\right)\mathbf{x}_t^\top.$$

The derivatives of the entries of $\mathbf{A}(t, \lambda)$ indexed by K^1 at λ_t can be calculated through the formula (4.2.12) of $\mathbf{a}_1^s$. By contrast, the derivatives of the entries of $\mathbf{A}(t, \lambda)$ indexed by K_{N}^0 all equal zero. To obtain the non-negative parameter value, we project λ_{t+1} onto interval $[0, +\infty)$ by taking $\max\{\lambda_{t+1}, 0\}$, whenever the result from Equation (4.2.3.4) is negative.

Note that λ_{t+1} defined in Equation (4.2.3.4) can be interpreted as the online solution from the projected stochastic gradient descent derived for the batch problem

$$\lambda_n^* = \arg\min_{\lambda \geqslant 0} \frac{1}{2n} \sum_{t=1}^{n} \|\mathbf{x}_{t+1} - \mathbf{A}(t, \lambda)\mathbf{x}_t\|_{\ell_2}^2.$$

Therefore, the sublinear regret property of projected stochastic gradient descent implies that, when η is given as $\mathcal{O}(\frac{1}{\sqrt{n}})$, we have

$$\frac{1}{2n} \sum_{t=1}^{n} \|\mathbf{x}_{t+1} - \mathbf{A}(t, \lambda_t)\mathbf{x}_t\|_{\ell_2}^2 - \frac{1}{2n} \sum_{t=1}^{n} \|\mathbf{x}_{t+1} - \mathbf{A}(t, \lambda_n^*)\mathbf{x}_t\|_{\ell_2}^2 = \mathcal{O}(\frac{1}{\sqrt{n}}).$$

Equation (4.2.3.4) implies that in the sense of average one step prediction error defined as Equation (4.2.3.4), the adaptive hyperparameter sequence $\{\lambda_t\}_t$ will perform almost as well as the best parameter λ_n^*, for a large number of online updates, with sufficiently small step size η. This completes the online procedure in the high dimensional domain, which we conclude in Algorithm 1.

Algorithm 1 Online Structured matrix-variate Lasso

Input: $\mathbf{A}(t, \lambda_t)$, $\mathbf{\Gamma}_0$, $\boldsymbol{\gamma}_1$, K_{N}^1(ordered list), $\mathbf{w}_{\mathrm{N}}^1$, λ_t, $\left[\mathbf{\Gamma}_0^1\right]^{-1}$, $\mathbf{x}_{t+1}$, $\widetilde{\mathbf{X}}_t$, t, where K_{N}^1, $\mathbf{w}_{\mathrm{N}}^1$, $\left[\mathbf{\Gamma}_0^1\right]^{-1}$ are associated with $\mathbf{A}(t, \lambda_t)$, and $\mathbf{w}_{\mathrm{N}}^1 = [\mathbf{w}]_{K_{\mathrm{N}}^1}$.
Select λ_{t+1} according to Section 4.2.3.4.
Update $\mathbf{A}(t, \lambda_t) \to \mathbf{A}(t, \frac{t+1}{t}\lambda_{t+1})$ using Algorithm 4.
Update $\mathbf{A}(t, \frac{t+1}{t}\lambda_{t+1}) \to \mathbf{A}(t+1, \lambda_{t+1})$ using Algorithm 5.
Output: $\mathbf{A}(t+1, \lambda_{t+1})$, $\mathbf{\Gamma}_0$, $\boldsymbol{\gamma}_1$, K_{N}^1, $\mathbf{w}_{\mathrm{N}}^1$, λ_{t+1}, $\left[\mathbf{\Gamma}_0^1\right]^{-1}$.

4.3 Augmented model for periodic trends

The online methods derived previously are based on the data process (4.1.4), which assumes the samples $(\mathbf{x}_\tau)_{\tau \in \mathbb{N}}$ have the time-invariant mean zero. In this section, we propose a more realistic data model which considers the trends, and adapt the online methods for stationary data to this augmented model.

A common detrend strategy in offline learning is subtracting the sample mean from observations of all time points, that is, given a raw time series $\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_t$, we

first calculate its sample mean as $\bar{\mathbf{x}}_t = \sum_{\tau=1}^{t} \mathbf{x}_\tau/t$, then use $\mathbf{x}'_\tau := \mathbf{x}_\tau - \bar{\mathbf{x}}_t$, $\tau = 0, ..., t$, in the proposed methods. As mentioned in the introduction, the detrend step is not admissible in the online fashion, however, it inspires us to consider the trend in an additional step. We focus primarily on the periodic trend, then the proposed model is

$$\mathbf{x}_t = \mathrm{b}_m^0 + \mathbf{x}'_t, \text{ where } \mathbf{x}'_t \sim \mathrm{VAR}(4.1.4), \tag{4.3.1}$$

where $m = t \bmod M \in \{0, \ldots, M-1\}$, M is the length of period which is a hyperparameter to be preassigned. The observations of Model (4.3.1) is $\mathbf{x}_t$, while $\mathbf{x}'_t$ has the similar role as the unobserved state in the state space models however the observation equation here is much simplified. Therefore the estimators are built on the series $\mathbf{x}_t$. Note that Model (4.3.1) admits another reparameterization with intercept

$$\mathbf{x}_t = \mathrm{b}_m + A\mathbf{x}_{t-1} + \mathbf{z}_t, \tag{4.3.2}$$

where

$$\mathrm{b}_m = \mathrm{b}_m^0 - A\mathrm{b}_{m-1}^0, \quad m = t \bmod M,$$

extending the relation (3.2.5) of intercept and mean to the time-variant mean. Note that b_{-1}^0 denotes b_{M-1}^0.

4.3.1 New OLS estimators and asymptotic distributions

For the augmented model (4.3.1), we propose a new OLS estimator of A, which is based on the new sample auto-covariances, together with the OLS estimator of b_m^0. Because two crucial properties to derive the Wald tests in Section 4.2.2 are the consistency of sample auto-covariances $\widehat{\mathbf{\Gamma}}_t(0)$, $\widehat{\mathbf{\Gamma}}_t(1)$, and the CLT of OLS estimator $\breve{\mathbf{A}}_t$, we derive the corresponding asymptotic results for the new estimators, and show that these asymptotics are exactly the same as in the stationary case. Therefore, all the results and procedures presented in Section 4.2.2 can be applied directly on the new estimators. We first define the estimator of A, still denoted as $\breve{\mathbf{A}}_t$, using general least squares (GLS) method

$$\breve{\mathbf{A}}_t, \widehat{\mathbf{b}}_{m,t} = \underset{A, \mathrm{b}_m}{\arg\min} \sum_{m=0}^{M-1} S_m(A, \mathrm{b}_m), \tag{4.3.3}$$

where

$$S_m = \sum_{\tau \in I_{m,t}} \tilde{\mathbf{z}}_\tau^\top \Sigma^{-1} \tilde{\mathbf{z}}_\tau, \quad \tilde{\mathbf{z}}_\tau = \mathbf{x}_\tau - \mathrm{b}_m - A\mathbf{x}_{\tau-1},$$

with $I_{m,t} = \{\tau = 1, ..., t : \tau \bmod M = m\}$, and Σ^{-1} the true white noise covariance given in Model (4.1.4). Note that $\tilde{\mathbf{z}}_\tau$ represents the residual of the prediction of sample $\mathbf{x}_\tau$. The explicit forms of $\breve{\mathbf{A}}_t, \widehat{\mathbf{b}}_{m,t}$ can be found through straightforward calculation, which yields new sample auto-covariances, denoted still as $\widehat{\mathbf{\Gamma}}_t(0)$, $\widehat{\mathbf{\Gamma}}_t(1)$, and the estimator of trend $\widehat{\mathbf{b}}_{m,t}^0$. Specifically, we have

$$\begin{cases} \breve{\mathbf{A}}_t = \widehat{\mathbf{\Gamma}}_t(1) \left[\widehat{\mathbf{\Gamma}}_t(0)\right]^{-1}, \\ \widehat{\mathbf{b}}_{m,t} = \bar{\mathbf{x}}_{m,t} - \breve{\mathbf{A}}_t \underline{\mathbf{x}}_{m-1,t} \;\Rightarrow\; \widehat{\mathbf{b}}_{m,t}^0 = \underline{\mathbf{x}}_{m,t} (\text{or } \bar{\mathbf{x}}_{m,t}), \end{cases} \tag{4.3.4}$$

with

$$\widehat{\mathbf{\Gamma}}_t(0) = \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} \left(\frac{\sum\limits_{\tau \in I_{m,t}} \mathbf{x}_{\tau-1}\mathbf{x}_{\tau-1}^\top}{p_{m,t}} - \underline{\mathbf{x}}_{m-1,t}\underline{\mathbf{x}}_{m-1,t}^\top \right),$$

$$\widehat{\mathbf{\Gamma}}_t(1) = \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} \left(\frac{\sum\limits_{\tau \in I_{m,t}} \mathbf{x}_{\tau}\mathbf{x}_{\tau-1}^\top}{p_{m,t}} - \bar{\mathbf{x}}_{m,t}\underline{\mathbf{x}}_{m-1,t}^\top \right),$$

$$p_{m,t} = |I_{m,t}|,\ \bar{\mathbf{x}}_{m,t} = \sum_{\tau \in I_{m,t}} \frac{\mathbf{x}_\tau}{p_{m,t}},\ m = 0, ..., M-1,$$

$$\underline{\mathbf{x}}_{m-1,t} = \sum_{\tau \in I_{m,t}} \frac{\mathbf{x}_{\tau-1}}{p_{m,t}},\ m = 0, ..., M-1.$$

Note that $\underline{\mathbf{x}}_{-1,t}$ denotes $\underline{\mathbf{x}}_{M-1,t}$. It is also straightforward to understand the new auto-covariance estimators. Each $S_m(A, \mathrm{b}_m)$ leads to an OLS problem of regression equation (4.3.2). Its minimization introduces two sample covariance matrices. The weighted average of all such sample auto-covariance matrices for $m = 0, \ldots, M-1$ is the new sample auto-covariance for Model (4.3.1). $p_{m,t}$ denotes the number of times that the samples from the m-th state point in the period have been predicted in the sense of Equation (4.3.3). As t grows, $\underline{\mathbf{x}}_{m,t}$ becomes $\bar{\mathbf{x}}_{m,t}$ quickly, and $p_{m,t}$ becomes $\frac{t}{M}$. For the augmented model, GLS and OLS estimators are still identical, with the latter defined as

$$\arg\min_{A, \mathrm{b}_m} \sum_{m=0}^{M-1} \sum_{\tau \in I_{m,t}} \tilde{\mathbf{z}}_\tau^\top \tilde{\mathbf{z}}_\tau.$$

The estimators given by Formula (4.3.3) enjoy the asymptotic properties in Proposition 4.3.1.

Proposition 4.3.1. *The following asymptotic properties hold for the estimators* $\widehat{\mathbf{\Gamma}}_t(0)$, $\widehat{\mathbf{\Gamma}}_t(1)$, $\breve{\mathbf{A}}_t$, $\widehat{\mathbf{b}}_{m,t}^0$, *as* $t \to +\infty$,

1. $\widehat{\mathbf{\Gamma}}_t(0) \xrightarrow{p} \Gamma(0)$, $\widehat{\mathbf{\Gamma}}_t(1) \xrightarrow{p} \Gamma(1)$,
2. $\widehat{\mathbf{b}}_{m,t}^0 \xrightarrow{p} \mathrm{b}_m^0$, $\breve{\mathbf{A}}_t \xrightarrow{p} A$,
3. $\sqrt{t}\, vec(\breve{\mathbf{A}}_t - A) \xrightarrow{d} \mathcal{N}(0, [\Gamma(0)]^{-1} \otimes \Sigma)$,

where $\Gamma(h) = \mathbb{E}\left(\mathbf{x}_t'[\mathbf{x}_{t-h}']^\top\right)$, $h \geqslant 0$, $\Sigma = \mathbb{E}\left(\mathbf{z}_t\mathbf{z}_t^\top\right)$.

The proofs of the above results are given in the appendix of this chapter. Thus, Theorem 4.2.3 and the bisection Wald test procedure are still valid using $\mathrm{Proj}_{\mathcal{G}_\mathrm{N}}(\breve{\mathbf{A}}_t)$ and $\widehat{\mathbf{\Gamma}}_t(0)$, $\widehat{\mathbf{\Gamma}}_t(1)$ defined in this section. On the other hand, $\widehat{\mathbf{\Gamma}}_t(0)$ and $\widehat{\mathbf{\Gamma}}_t(1)$ satisfy the one rank update formulas:

$$\begin{aligned}
\widehat{\mathbf{\Gamma}}_{t+1}(0) &= \frac{t}{t+1}\widehat{\mathbf{\Gamma}}_t(0) + \frac{1}{t+1}\left[\frac{p_{\bar{m},t}}{p_{\bar{m},t}+1}(\mathbf{x}_t - \underline{\mathbf{x}}_{\bar{m}-1,t})(\mathbf{x}_t - \underline{\mathbf{x}}_{\bar{m}-1,t})^\top\right] \\
\widehat{\mathbf{\Gamma}}_{t+1}(1) &= \frac{t}{t+1}\widehat{\mathbf{\Gamma}}_t(1) + \frac{1}{t+1}\left[\frac{p_{\bar{m},t}}{p_{\bar{m},t}+1}(\mathbf{x}_{t+1} - \underline{\mathbf{x}}_{\bar{m},t})(\mathbf{x}_t - \underline{\mathbf{x}}_{\bar{m}-1,t})^\top\right],
\end{aligned} \tag{4.3.5}$$

where $\bar{m} = (t+1) \bmod M$. Thus, when new sample comes, $[\widehat{\mathbf{\Gamma}}_{t+1}(0)]^{-1}$ can still be calculated efficiently given the matrix inverse at the previous time. The details of the extended low dimensional learning procedure see Algorithm 3 in the appendix.

4.3.2 Augmented structured matrix-variate Lasso and the optimality conditions

To adapt the Lasso-based approach to Model (4.3.2), the corresponding trend and graph estimators can be obtained by minimizing the augmented Matrix-variate Lasso problem

$$\mathbf{A}(t,\lambda), \mathbf{b}_m(t,\lambda) = \underset{A\in\mathcal{K}_\mathcal{G}, \mathrm{b}_m}{\arg\min} \frac{1}{2t} \sum_{m=0}^{M-1} \sum_{\tau\in I_{m,t}} \|\mathbf{x}_\tau - \mathrm{b}_m - A\mathbf{x}_{\tau-1}\|_{\ell_2}^2 + \lambda F \|A_{\mathrm{N}}\|_{\ell_1}. \quad (4.3.6)$$

As in the extension of our first approach, the extra bias terms $\mathrm{b}_m, m = 0, ..., M-1$, do not affect the core techniques, rather they force the methods to consider the M means in the sample autocovariances. Since b_m only appear in the squares term, the minimizers $\mathbf{b}_m(t,\lambda)$ have the same dependency with $\mathbf{A}(t,\lambda)$ as in Equation (4.3.4). Thus the trend b_m^0 can still be estimated by $\underline{\mathbf{x}}_{m,t}$, and we extend the algorithms in Section 4.2.3 to update the batch solution of augmented Lasso (4.3.6) from $\mathbf{A}(t,\lambda_t)$ to $\mathbf{A}(t+1,\lambda_{t+1})$, given new sample $\mathbf{x}_{t+1}$. To compute the regularization path $\mathbf{A}(t,\lambda_t) \to \mathbf{A}(t,(1+\frac{1}{t})\lambda_{t+1})$, Proposition 4.3.2 implies that Algorithm 4 can still be used, with the adjusted definitions of $\mathbf{\Gamma}_0$ and $\boldsymbol{\gamma}_1$.

Proposition 4.3.2. *A minimizer $\mathbf{A}(t,\lambda)$ of Lasso problem (4.3.6) satisfies the linear system*

$$\begin{cases} \mathbf{\Gamma}_0^1 \mathbf{a}_1^s - \boldsymbol{\gamma}_1^1 + \lambda \mathbf{w}_1 = 0 \\ \mathbf{\Gamma}_0^0 \mathbf{a}_1^s - \boldsymbol{\gamma}_1^0 + \lambda \mathbf{w}_0 = 0, \end{cases}$$

where $\mathbf{a}^s$ is the vectorized scaled Lasso solution $\mathbf{A}(t,\lambda)$, $\mathbf{w}, K^1, K_{\mathrm{N}}^0$ are also defined analogously from $\mathbf{A}(t,\lambda)$, while $\widehat{\mathbf{\Gamma}}_t(0)$ and $\widehat{\mathbf{\Gamma}}_t(1)$ used in the definitions of $\mathbf{\Gamma}_0$ and $\boldsymbol{\gamma}_1$ are the new sample auto-covariance matrices in Equation (4.3.4).

For the data path $\mathbf{A}(t,(1+\frac{1}{t})\lambda_{t+1}) \to \mathbf{A}(t+1,\lambda_{t+1})$, in the same spirit of Problem (4.2.16), we introduce variables $\mu_1, ..., \mu_{NF}$ to let evolve Lasso problem (4.3.6) from time t to $t+1$ through the following variational problem

$$\mathbf{A}_{\lambda_{t+1},t}(\mu_1,...,\mu_{NF}),\ \mathbf{b}_{m,\lambda_{t+1},t}(\mu_1,...,\mu_{NF}) = \underset{A\in\mathcal{K}_\mathcal{G}, \mathrm{b}_m}{\arg\min}\ L_{\lambda_{t+1},t}(\mu_1,...,\mu_{NF}),$$

$$\text{where } L_{\lambda_{t+1},t}(\mu_1,...,\mu_{NF}) = \frac{1}{2(t+1)} \sum_{m=0}^{M-1} \sum_{\tau\in I_{m,t}} \|\mathbf{x}_\tau - \mathrm{b}_m - A\mathbf{x}_{\tau-1}\|_{\ell_2}^2$$

$$+ \lambda_{t+1} F \|A_{\mathrm{N}}\|_{\ell_1} + \frac{1}{2(t+1)} \sum_{i=1}^{NF} \mu_i (\boldsymbol{x}_{t+1,i} - b_{\bar{m},i} - \sum_{k\in K} \langle U_k, A^0\rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} [U_k]_{i,:}\mathbf{x}_t)^2,$$

where $b_{\bar{m},i} = [\mathrm{b}_{\bar{m}}]_i$, $\bar{m} = (t+1) \bmod M$. To extend the homotopy Algorithm of data path, we first calculate the optimality conditions of Lasso $L_{\lambda_{t+1},t}(\mu_1,...,\mu_{NF})$

with respect to the constraint-free A^0 in Equation (4.3.7), then extract its vector representation in terms of $\mathbf{a}_1^s$.

A minimizer $\mathbf{A}^0$ such that $\mathbf{A} = \text{Proj}_{\mathcal{G}}(\mathbf{A}^0)$ of $L_{\lambda_{t+1},t}(\mu_1, ..., \mu_{NF})$ satisfies

$$
\begin{aligned}
0 \in & \frac{\partial L_{\lambda_{t+1},t}(\mu_1, ..., \mu_{NF})}{\partial A^0} \\
& = \frac{t}{t+1}\left[\sum_{k,k' \in K} \langle U_k, U_{k'} \widehat{\mathbf{\Gamma}}_t(0)\rangle \langle \frac{1}{\|U_{k'}\|_{\mathbf{F}}^2} U_{k'}, \mathbf{A}^0\rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k - \sum_{k \in K} \langle U_k, \widehat{\mathbf{\Gamma}}_t(1)\rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k\right] \\
& + \frac{1}{t+1} \sum_{i=1}^{NF} \mu_i \frac{p_{\bar{m},t}}{p_{\bar{m},t} + \mu_i} \left[\sum_{k,k' \in K} (\mathbf{x}_t - \underline{\mathbf{x}}_{\bar{m}-1,t})^\top [U_k]_{i,:}^\top [U_{k'}]_{i,:} (\mathbf{x}_t - \underline{\mathbf{x}}_{\bar{m}-1,t}) \langle \frac{1}{\|U_{k'}\|_{\mathbf{F}}^2} U_{k'}, \mathbf{A}^0\rangle \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k\right] \\
& - \frac{1}{t+1} \sum_{i=1}^{NF} \mu_i \frac{p_{\bar{m},t}}{p_{\bar{m},t} + \mu_i} \left[\sum_{k \in K} (\boldsymbol{x}_{t+1,i} - \underline{\boldsymbol{x}}_{\bar{m},t,i})(\mathbf{x}_t - \underline{\mathbf{x}}_{\bar{m}-1,t})^\top [U_k]_{i,:}^\top \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k\right] \\
& + \lambda_{t+1} \sum_{k \in K_{\mathrm{N}}} \partial \left|\langle U_k, \mathbf{A}^0\rangle\right| \frac{1}{\|U_k\|_{\mathbf{F}}^2} U_k,
\end{aligned}
\tag{4.3.7}
$$

where $\widehat{\mathbf{\Gamma}}_t(0)$, $\widehat{\mathbf{\Gamma}}_t(1)$ are the ones defined in Equation (4.3.4), and $\underline{\boldsymbol{x}}_{\bar{m},t,i} = \left[\underline{\mathbf{x}}_{\bar{m},t}\right]_i$. Note that, when $(t \bmod M) \neq m$, $\bar{\mathbf{x}}_{m,t} = \underline{\mathbf{x}}_{m,t}$.

The following remarks can then be made.

Remark 4.3.3. Subdifferential formula (4.3.7) is almost the same as its stationary counterpart except that the former uses centered data, as well as the appearance of term $\frac{p_{\bar{m},t}}{p_{\bar{m},t}+\mu_i}$.

Remark 4.3.4. Equation (4.3.7) implies the update formula (4.3.5). Since

$$L_{\lambda_{t+1},t}(1, ..., 1) = L_{\lambda_{t+1},t+1},$$

and $\frac{\partial L_{\lambda_{t+1},t+1}}{\partial A^0}$ is given in Equation (4.2.7), with $\lambda = \lambda_{t+1}$, $\widehat{\mathbf{\Gamma}}_{t+1}(0)$, $\widehat{\mathbf{\Gamma}}_{t+1}(1)$ defined alternatively in Equation (4.3.4). Thus, by equating the quantities in $\langle U_k, U_{k'} \cdot \rangle$ and $\langle U_k, \cdot \rangle$ in the corresponding subdifferential formulas respectively, update formula (4.3.5) can be induced.

We recall that we update the solution along the path

$$L_{\lambda_{t+1},t}(0, 0, ..., 0) \to L_{\lambda_{t+1},t}(1, 0, ..., 0) \to L_{\lambda_{t+1},t}(1, 1, ..., 1) = L_{\lambda_{t+1},t+1}.$$

At each step $L_{\lambda_{t+1},t}(..., 1, \mu_i, 0, ...)$, $\mu_i \in [0, 1]$, the optimal solution $\mathbf{A}_{\lambda_{t+1},t}(..., 1, \mu_i, 0, ...)$ is piece-wise smooth with respect to μ_i, element-wise. We retrieve the linear system of $\mathbf{a}_1^s$ in terms of μ_i for each $\mathbf{A}_{\lambda_{t+1},t}(..., 1, \mu_i, 0, ...)$ in Proposition 4.3.5.

Proposition 4.3.5. *A minimizer* $\mathbf{A}_{\lambda_{t+1},t}(..., 1, \mu_i, 0, ...)$ *of Lasso* $L_{\lambda_{t+1},t}(..., 1, \mu_i, 0, ...)$ *satisfies the linear system*

$$
\begin{cases}
\mathbf{\Gamma}_0^1(\mu_i)\mathbf{a}_1^s - \boldsymbol{\gamma}_1^1(\mu_i) + (1 + \frac{1}{t})\lambda_{t+1}\mathbf{w}_1 = 0 \\
\mathbf{\Gamma}_0^0(\mu_i)\mathbf{a}_1^s - \boldsymbol{\gamma}_1^0(\mu_i) + (1 + \frac{1}{t})\lambda_{t+1}\mathbf{w}_0 = 0,
\end{cases}
$$

where $\mathbf{a}^s, \mathbf{w}, K^1, K_{\mathrm{N}}^0$ *are with respect to* $\mathbf{A}_{\lambda_{t+1},t}(..., 1, \mu_i, 0, ...)$*, defining the extractions through* (4.2.11)*,*

$$\begin{aligned}\boldsymbol{\Gamma}_0(\mu_i) = \boldsymbol{\Gamma}_0 &+ \frac{1}{t}\sum_{n=1}^{i-1}\frac{p_{\bar{m},t}}{p_{\bar{m},t}+1}[\widetilde{\mathbf{X}}_t - \underline{\widetilde{\mathbf{X}}}_{\bar{m}-1,t}]_{:,n}[\widetilde{\mathbf{X}}_t - \underline{\widetilde{\mathbf{X}}}_{\bar{m}-1,t}]_{:,n}^\top \\ &+ \frac{\mu_i}{t}\frac{p_{\bar{m},t}}{p_{\bar{m},t}+\mu_i}[\widetilde{\mathbf{X}}_t - \underline{\widetilde{\mathbf{X}}}_{\bar{m}-1,t}]_{:,i}[\widetilde{\mathbf{X}}_t - \underline{\widetilde{\mathbf{X}}}_{\bar{m}-1,t}]_{:,i}^\top,\end{aligned}$$

$$\begin{aligned}\boldsymbol{\gamma}_1(\mu_i) = \boldsymbol{\gamma}_1 &+ \frac{1}{t}\sum_{n=1}^{i-1}\frac{p_{\bar{m},t}}{p_{\bar{m},t}+1}(\mathbf{x}_{t+1,n} - (\underline{\mathbf{x}}_{\bar{m},t})_n)[\widetilde{\mathbf{X}}_t - \underline{\widetilde{\mathbf{X}}}_{\bar{m}-1,t}]_{:,n} \\ &+ \frac{\mu_i}{t}\frac{p_{\bar{m},t}}{p_{\bar{m},t}+\mu_i}(\mathbf{x}_{t+1,i} - (\underline{\mathbf{x}}_{\bar{m},t})_i)[\widetilde{\mathbf{X}}_t - \underline{\widetilde{\mathbf{X}}}_{\bar{m}-1,t}]_{:,i},\end{aligned}$$

with $[\underline{\widetilde{\mathbf{X}}}_{\bar{m}-1,t}]_{k,i} = [U_k]_{i,:}\underline{\mathbf{x}}_{\bar{m}-1,t}$*,* $p_{\bar{m},t} := t \bmod M$*,* $\boldsymbol{\Gamma}_0, \boldsymbol{\gamma}_1$ *the same as Proposition 4.3.2.*

Therefore, the derived homotopy algorithm is essentially the previous homotopy Algorithm 5 with minor changes. The details see Algorithm 6 in the appendix.

Lastly, we derive the updating rule for the regularization parameter. We still consider the one step prediction error, which writes as the following objective function in the case of Model (4.3.2)

$$f_t(\lambda) = \frac{1}{2}\|\mathbf{x}_{t+1} - \mathbf{b}_{\bar{m}}(t,\lambda) - \mathbf{A}(t,\lambda)\mathbf{x}_t\|_{\ell_2}^2.$$

Given the previous solution $\mathbf{A}(t,\lambda_t)$ and $\mathbf{b}_{\bar{m}}(t,\lambda_t)$, we assume that λ_t is not a critical point. Then the derivative of f_t with respect to λ is calculated as

$$\begin{aligned}\frac{\mathrm{d}f_t(\lambda_t)}{\mathrm{d}\lambda} &= \left\langle \frac{\mathrm{d}f_t(\lambda)}{\mathrm{d}\mathbf{b}_{\bar{m}}(t,\lambda)}\Big|_{\lambda=\lambda_t}, \frac{\mathrm{d}\mathbf{b}_{\bar{m}}(t,\lambda_t)}{\mathrm{d}\lambda}\right\rangle + \left\langle \frac{\mathrm{d}f_t(\lambda)}{\mathrm{d}\mathbf{A}(t,\lambda)}\Big|_{\lambda=\lambda_t}, \frac{\mathrm{d}\mathbf{A}(t,\lambda_t)}{\mathrm{d}\lambda}\right\rangle \\ &= \left\langle \mathbf{G}_t^{\mathbf{b}}, -\frac{\mathrm{d}\mathbf{A}(t,\lambda_t)}{\mathrm{d}\lambda}\underline{\mathbf{x}}_{\bar{m}-1,t}\right\rangle + \left\langle \mathbf{G}_t, \frac{\mathrm{d}\mathbf{A}(t,\lambda_t)}{\mathrm{d}\lambda}\right\rangle \\ &= \left\langle [\mathbf{A}(t,\lambda_t)\mathbf{x}_t - \mathbf{x}_{t+1} + \mathbf{b}_{\bar{m}}(t,\lambda_t)][\mathbf{x}_t - \underline{\mathbf{x}}_{\bar{m}-1,t}]^\top, \frac{\mathrm{d}\mathbf{A}(t,\lambda_t)}{\mathrm{d}\lambda}\right\rangle,\end{aligned}$$

where $\mathbf{G}_t^{\mathbf{b}} = \mathbf{b}_{\bar{m}}(t,\lambda_t) - \mathbf{x}_{t+1} + \mathbf{A}(t,\lambda_t)\mathbf{x}_t$, $\mathbf{G}_t = [\mathbf{A}(t,\lambda_t)\mathbf{x}_t - \mathbf{x}_{t+1} + \mathbf{b}_{\bar{m}}(t,\lambda_t)]\mathbf{x}_t^\top$. Analogous to Section 4.2.3.4, we have $\langle \mathbf{G}_t, \frac{\mathrm{d}\mathbf{A}(t,\lambda_t)}{\mathrm{d}\lambda}\rangle = -\left[\mathbf{a}_1^{\mathbf{G}_t}\right]^\top \left[\boldsymbol{\Gamma}_0^1\right]^{-1}\mathbf{w}_1$. Using the same updating rules of the projected stochastic gradient descent presented in Section 4.2.3.4, we can compute the online solution λ_{t+1}. We can see that, the introduction of bias terms b_m into the original model makes them center the raw data automatically during the model fitting. This enables the direct learning over raw time series, while maintaining the performance of methods comparable to the stationarity-based ones.

We summarize the complete learning procedure of this subsection in Algorithm 6 in the appendix.

4.4 Experiments

We test the two proposed approaches for the online graph and trend learning on both synthetic and real data set.

4.4.1 Synthetic data

4.4.1.1 Evaluation procedures

We now present the evaluation procedure for the augmented model approaches. In each simulation, we generate a true graph A with the structure indicated by $\mathcal{K}_{\mathcal{G}}$. In particular, we impose sparsity on its spatial graph A_{N} by randomly linking a subset of node pairs. The values of non-zero entries in A_{N}, and the entries in A_{F}, $\mathrm{diag}(A)$ are generated in a random way. Additionally, we generate a trend over a period of M time points for each node and each feature. Therefore, the true b_m^0, $m = 0, ..., M-1$, consists in these NF trend vectors, each containing M elements. Then, we synthesize very few samples $\mathbf{x}_t$ from Model (4.3.1) until time t_0. Figure 4.2 shows an example of the synthetic time series $\mathbf{x}_t$, compared with its stationary source $\mathbf{x}'_t$ before adding the periodic trend. The graph and trend estimators proposed in Section 4.3 only use $\mathbf{x}_t$. We then set up the batch Lasso problem (4.3.6) with the generated samples and use its solution $\mathbf{A}(t_0, \lambda_0)$ to start the high-dimensional online procedure since the next synthetic sample. The batch problem is solved via the accelerated proximal gradient descent with the backtracking line search (Parikh and Boyd, 2014, Section 3.2.2). We especially set λ_0 as a large number so as to have an over sparse initial solution. Therefore, we expect to see a decreasing λ_t, together with a more accurate estimate $\mathbf{A}(t, \lambda_t)$ as t grows. The updating of $\hat{\mathbf{\Gamma}}_t(0)$ and $\hat{\mathbf{\Gamma}}_t(1)$ starts from $t = 1$, using Formula (4.3.5), since they are the only inputs of the proximal gradient descent algorithm. However, we wait until there are enough samples for $\hat{\mathbf{\Gamma}}_t(0)$ to be invertible, then we start the low-dimensional online procedure at time t with $[\hat{\mathbf{\Gamma}}_t(0)]^{-1}$. To analyse the performance of the proposed approaches, we define the average one step prediction error metric as,

$$\sum_{\tau=1}^{t} \frac{\|\mathbf{x}_{\tau+1} - \widehat{\mathbf{b}}_{m(\tau+1),\tau} - \widehat{\mathbf{A}}_\tau \mathbf{x}_\tau\|_2}{t\|\mathbf{x}_{\tau+1}\|_2}, \quad m(\tau+1) = (\tau+1) \bmod M, \tag{4.4.1}$$

and root mean square deviation (RMSD) as,

$$\frac{\|\widehat{\mathbf{A}}_t - A\|_{\mathbf{F}}}{\|A\|_{\mathbf{F}}}, \tag{4.4.2}$$

where $\widehat{\mathbf{A}}_\tau$ denotes the online estimates from either approach at time τ, and

$$\widehat{\mathbf{b}}_{m(\tau+1),\tau} = \begin{cases} \underline{\mathbf{x}}_{m(\tau+1),\tau} - \breve{\mathbf{A}}_\tau \underline{\mathbf{x}}_{m(\tau+1)-1,\tau}, & \text{in low-dimensional,} \\ \underline{\mathbf{x}}_{m(\tau+1),\tau} - \mathbf{A}(\tau, \lambda_\tau) \underline{\mathbf{x}}_{m(\tau+1)-1,\tau}, & \text{in high-dimensional.} \end{cases}$$

We collect the metric values along time. We perform such simulation multiple times to obtain furthermore the means and the standard deviations of error metrics at each

iteration (when the estimators are available) to better demonstrate the performance. The true graph A and trends b_m^0 are generated independently across these simulations.

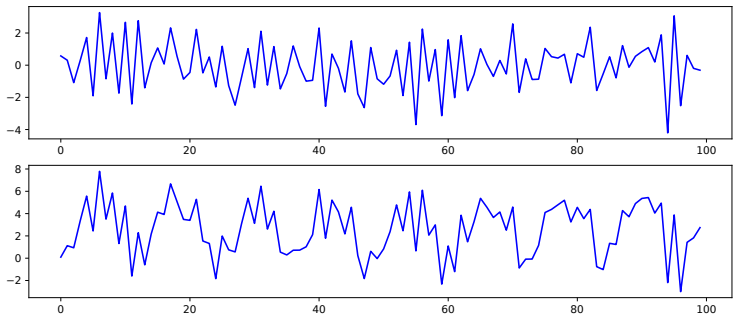

Figure 4.2: Top is the stationary time series from Model (4.1.5) at one component, bottom is the above time series with added periodic trend (M = 12).

4.4.1.2 Simulation results

We first visualize the representative estimates in heatmap with $N = 10$, $F = 4$, and $M = 12$ for illustration purpose. Then we plot the evolution of error metrics and regularization parameter of 30 simulations for $N = 20$, $F = 5$, and $M = 12$. Lastly, we report the running time. The hyperparameter settings are given in the captions of figures of corresponding results.

Figures 4.3 and 4.4 show the estimated graphs of two approaches when their corresponding online procedures start. In Figure 4.3, we can see that the batch solution which starts the high-dimensional procedure is over sparse due to the large λ_0. We can notice from Figure 4.4 that the two initial estimations of A_{F} are already satisfactory, especially the Lasso solution which uses only 20 samples. Actually, estimations of A_{F} and $\mathrm{diag}(A)$ converge to the truth very quickly in both cases when N is significantly larger than F. Figures 4.5 and 4.6 show that the estimations of A_{N} of both approaches tend to the true values as more samples are received. Meanwhile, Figure 4.7 shows the effectiveness of trend estimator $\underline{\mathbf{x}}_{m,t}$ defined in Equation (4.3.4).

We now show the numeric results of 30 simulations, with $N = 20$, $F = 5$, and $M = 12$. We test three different step sizes η, 5×10^{-7}, 1×10^{-6}, and 5×10^{-6}. With each value we perform 10 independent simulations. Figure 4.8 and 4.9 plot the evolution of error metrics (4.4.1) and (4.4.2), respectively. For better visualization effect, since the performance of the low-dimensional procedure does not depend on η, we only show one mean metric curve instead of 3, in the two figures, which is calculated from the results of these 30 simulations.

Figures 4.8 and 4.9 show the convergence of A_{N} estimations of both procedures. Moreover, for the high-dimensional procedure, the step size η determines the convergence speed. Especially, we can see from Figure 4.8 that the RMSD of the high-dimensional procedure with $\eta = 5 \times 10^{-6}$ decreases the most quickly for the

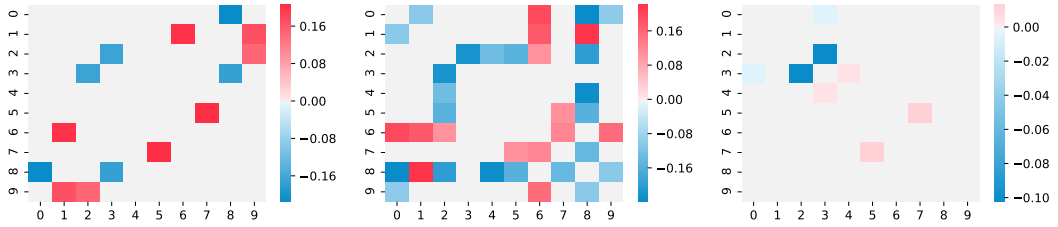

Figure 4.3: *Initial spatial graph estimates which start the online procedures.* True A_{N} (left), $\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,91}$ of the low-dimensional procedure (middle), and $\mathbf{A}_{\mathrm{N}}(20, 0.05)$ of the high-dimensional procedure (right) are represented by heatmaps. Simulation settings: $N = 10$, $F = 4$, number of model parameters = 571, significance level of χ^2 test in Corollary 4.2.4 = 0.1.

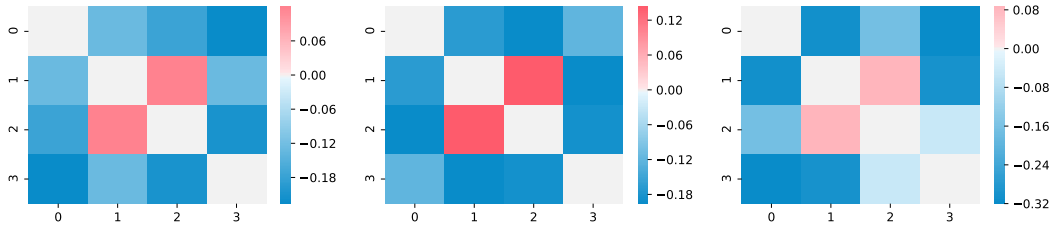

Figure 4.4: *Initial feature graph estimates which start the online procedures.* True A_{F} (left), $\widehat{\boldsymbol{A}_{\mathbf{F}}}_{,91}$ of the low-dimensional procedure (middle), and $\mathbf{A}_{\mathrm{F}}(20, 0.05)$ of the high-dimensional procedure (right). Simulation settings: $N = 10$, $F = 4$, number of model parameters = 571.

first 100 iterations, after which it starts to slow down and decrease more slowly than the RMSDs of the other two step sizes. For the low-dimensional procedure, when its estimator is available, the RMSD decreases very fast, and it shows the trend to keep decreasing for larger sample size. Nevertheless, the estimator of the low-dimensional procedure performs worse than the Lasso estimators in the sense of the prediction of unseen data, as shown in Figure 4.9. This is likely linked with the fact that the selection procedure updates the regularization parameter of the Lasso estimator towards the direction that minimizes the one step prediction error (4.4.1). The larger standard deviation is due to the larger magnitude of low-dimensional estimator, contrast to the Lasso estimator which is regularized by the ℓ_1 norm. This can also be observed in the scales of the $y-$axis in Figures 4.3 - 4.6.

For synthetic data, it is not surprising that the RMSD from the low-dimensional procedure will tend toward zero, because these data are precisely sampled from the model used in the method derivation. On the other hand, at each online iteration, the OLS estimation is calculated accurately. In contrast, for the homotopy algorithms, they still introduce small errors, possibly due to the following assumptions used in the derivation of the method: 1. the active elements of K_{N}^1 of the algorithm inputs

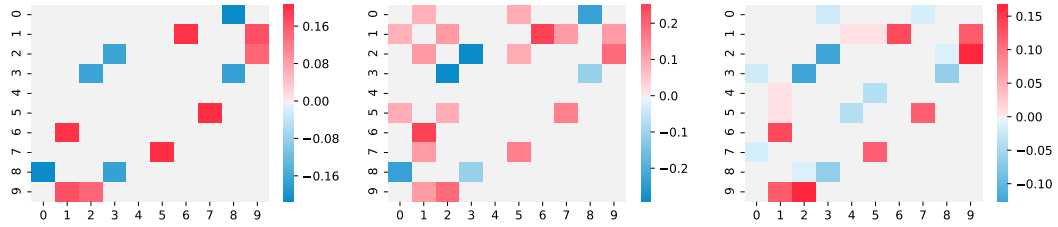

Figure 4.5: *Spatial graph estimated at the arrival of the* 182*-nd sample.* True A_{N} (left), $\widehat{\boldsymbol{A}_{\mathbf{N},182}}$ of the low-dimensional procedure (middle), and $\mathbf{A}_{\mathrm{N}}(182, 0.0286)$ of the high-dimensional procedure (right) are represented by heatmaps. Simulation settings: $N = 10$, $F = 4$, number of model parameters $= 571$, significance level of χ^2 test $= 0.1$, $\eta = 5 \times 10^{-6}$.

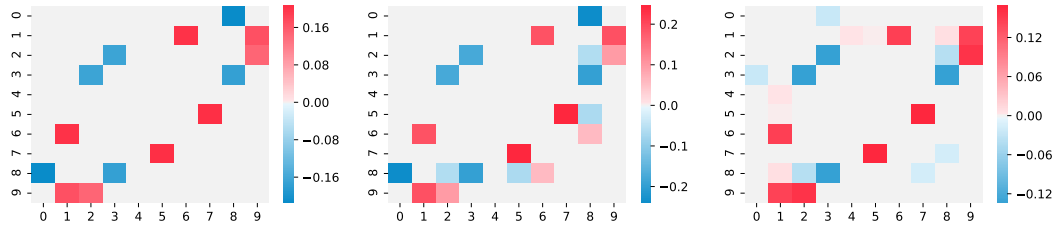

Figure 4.6: *Spatial graph estimated at the arrival of the* 591*-th sample.* True A_{N} (left), $\widehat{\boldsymbol{A}_{\mathbf{N},591}}$ of the low-dimensional procedure (middle), and $\mathbf{A}_{\mathrm{N}}(591, 0.0130)$ of the high-dimensional procedure (right) are represented by heatmaps. Simulation settings: $N = 10$, $F = 4$, number of model parameters $= 571$, significance level of χ^2 test $= 0.1$, $\eta = 5 \times 10^{-6}$.

are not zero[‡]; 2. the sub-derivatives of those zero elements are strictly within $(-1, 1)$; 3. every λ_t at which we calculate the derivative as in Section 4.2.3.4 is not a critical point. Thus, for example, small non-zero entry values in the inputs may cause the numerical errors. However, in real applications, the only available metric which allows the performance comparison is the prediction error (4.4.1).

Figure 4.10 demonstrates the performance of the updating method of the regularizing parameter λ, and the impact from different step size values η. The curves emphasize the convergence of the estimation updated by the high-dimensional procedure. Moreover, we can observe that λ_t are decreasing, which was expected from the experiment design. On the other hand, the results show that a larger step size will make the convergence faster, yet more affected by the noise, especially when the solution has converged.

[‡]This hypothesis means that, some zero $(\mathbf{a}_1^s)_{i(k)}, k \in K_{\mathrm{N}}^1$ should not satisfy the first equations of the optimality condition (4.2.6) and (4.2.7), due to the computation coincidence.

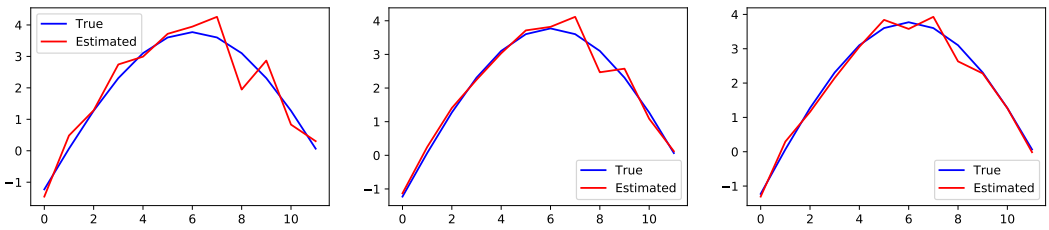

Figure 4.7: *Trend of the first node, first feature, estimated at different times.* Estimation at $t = 182$ (left), $t = 273$ (middle), $t = 591$ (right). Simulation settings: $N = 10$, $F = 4$, $M = 12$, number of model parameters $= 571$.

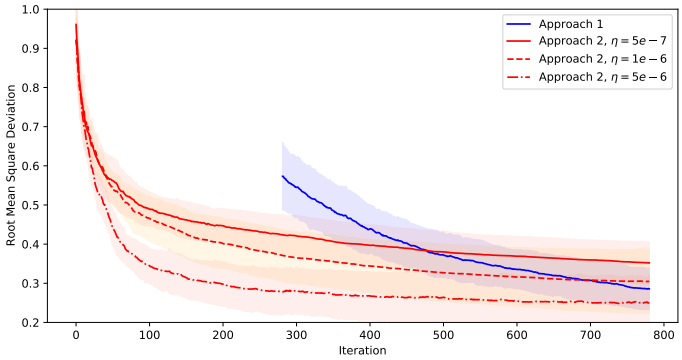

Figure 4.8: *Root mean square deviation.* The red curves are the mean RMSD of the high-dimensional procedure, taken over 10 simulations each. The blue curve is the mean RMSD of the low-dimensional procedure, taken over the same 30 simulations. The shaded areas represent the corresponding one standard deviations. Other simulation settings: $N = 20$, $F = 5$, $M = 12$, number of model parameters $= 1500$, significance level of χ^2 test $= 0.1$, $t_0 = 20$, $\lambda_0 = 0.03$. In the first high dimensional phase, the accurate estimator of the low-dimensional procedure is not available.

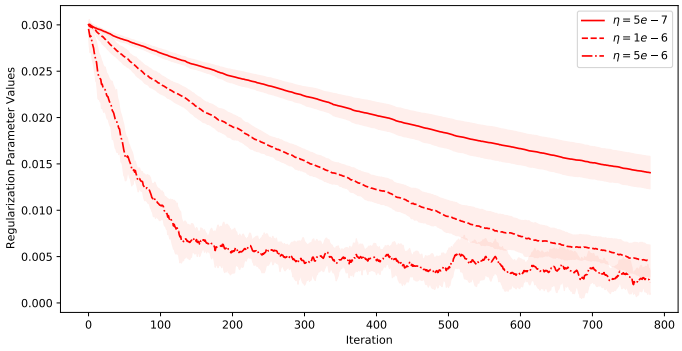

Figure 4.10: *Regularization parameter evolution.* The red curves are the mean regularization parameter values, taken over 10 simulations each. The shaded areas represent the corresponding one standard deviations. Other simulation settings: $N = 20$, $F = 5$, $M = 12$, number of model parameters $= 1500$, $t_0 = 20$, $\lambda_0 = 0.03$.

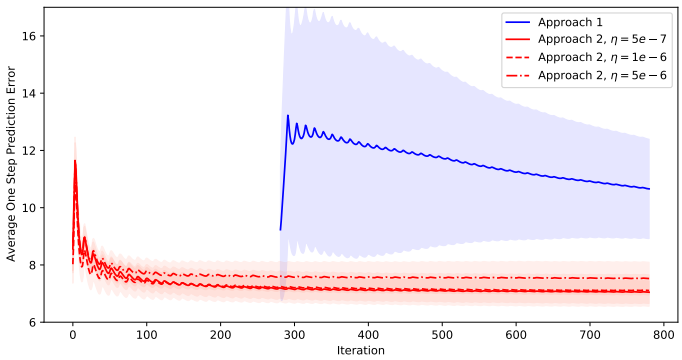

Figure 4.9: *Average one step prediction error.* The red curves are the mean prediction error of the high-dimensional procedure, taken over 10 simulations each. The blue curve is the mean prediction error of the low-dimensional procedure, taken over the same 30 simulations. The shaded areas represent the corresponding one standard deviations. Other simulation settings: $N = 20$, $F = 5$, $M = 12$, number of model parameters $= 1500$, significance level of χ^2 test $= 0.1$, $t_0 = 20$, $\lambda_0 = 0.03$.

We also compare the running time of a single online update for the two methods in Figure 4.11. Firstly, it is clear that updating the Lasso solutions by the homotopy algorithms saves considerable time, which is on average 0.20 seconds for the graph size $N = 20$, $F = 5$. The running time of the accelerated proximal gradient descent performed in the beginning of these simulations costs more than 3 seconds. By contrast, an update using the low-dimensional procedure takes 25 seconds on average. We can also notice that the high-dimensional procedure with larger step size runs slower, because the updated regularization parameter is quite different from the preceding one, as evidenced by the results in Figure 4.10.

Lastly, it is worthwhile to point out that, because the true A_{N} has a high level of sparsity, the Wald test will accept $H_0 : \alpha = 0$ more easily with lower significance levels, and we can observe the Wald estimator $\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t}$ rejects those false non-zero entries faster. Nevertheless, since we do not know the true graph sparsity for real data, the significance level can be regarded as the hyperparameter which controls the desired sparsity as well for the first approach.

4.4.2 Climatology data

We use the U.S. Historical Climatology Network (USHCN) data[§] to test our proposed approaches. The data set contains monthly averages of four climatology features, recorded at weather stations located across the United States, over years. The four features are: minimal temperature, maximal temperature, mean temperature, and precipitation. A snippet of the data set has been given in Figure 2.1, which illustrates these feature time series observed from a certain spatial location. A clear periodic trend can be seen from each scalar time series, with period length equal to 12 months. We can also notice that some observations are missing in the

[§]The data set is available at https://www.ncdc.noaa.gov/ushcn/data-access

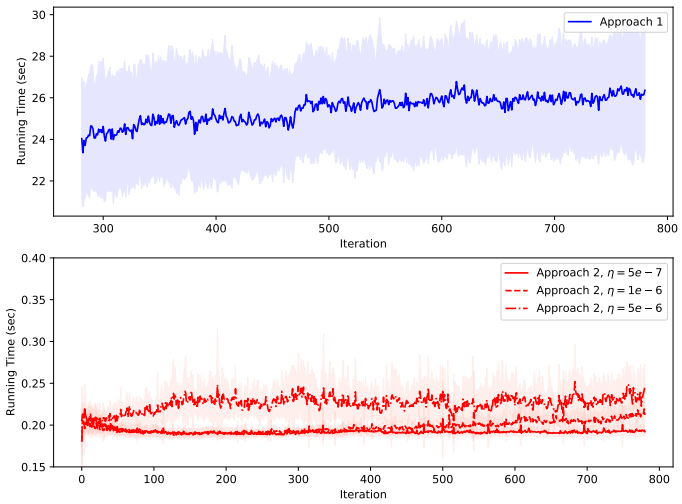

Figure 4.11: *Running time of each online update.* The red curves are the mean running time of the high-dimensional procedure, taken over 10 simulations each. The blue curve is the mean running time of the low-dimensional procedure, taken over the same 30 simulations. The shaded areas represent the corresponding one standard deviations. Other simulation settings: $N = 20$, $F = 5$, $M = 12$, number of model parameters $= 1500$, significance level of χ^2 test $= 0.1$, $t_0 = 20$, $\lambda_0 = 0.03$.

data set; to focus on the evaluation of learning approaches, we do not consider the stations with incomplete time series. Geographically, we picked data only from California and Nevada for this experiment. The summary of experiment setting thus is: $N = 27, F = 4, M = 12$, total number of time points $= 1523$ months (covering the years from 1894 to 2020). We apply the approaches from Section 4.3 on the raw time series to learn the weather graph of the region. The testing procedure using the real data is identical to the evaluation procedure with the synthetic data. We use the first $t_0 + 1$ observations to set up the corresponding batch Lasso problem, and use its solution to start the high-dimensional procedure. The low-dimensional procedure will be started once $\hat{\mathbf{\Gamma}}_t(0)$ becomes invertible. The average one step prediction error is calculated along online iterations. t_0 and λ_0 are always set as 20 and 0.03, respectively. Their values do not affect the methods' performance much, because of the adaptive tuning procedures of the regularization parameter.

Figure 4.13 and 4.14 show the spatial graphs learned by the two proposed approaches in Section 4.3 updated at different times. Figure 4.15 plots the evolution of regularization parameter value. We can see that, for the high-dimensional procedure, when more observations are received, it finds that more location pairs actually have a Granger causal effect on each other. On the other hand, compared to the estimated graphs from the high-dimensional procedure, those from the low-dimensional procedure vary more along time, which can be caused by the following facts: 1. in the early stage, $\hat{\mathbf{\Gamma}}_t(0)$ is still ill-conditioned, therefore its inverse brings unstable OLS solutions; 2. the low-dimensional procedure relies on large sample properties of the designed estimators. These points are also supported by the average one step

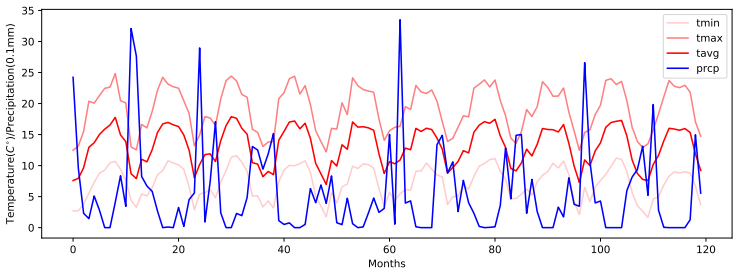

Figure 4.12: *Learning set for augmented model based approaches.* Monthly records of minimal temperature, maximal temperature, mean temperature, and precipitation from Station USH00040693 over a period over 20 years.

prediction curve given in Figure 4.16, where it is shown that, the prediction error of the low-dimensional procedure is significantly larger than the high-dimensional procedure, especially when the sample size is around 500 to 800.

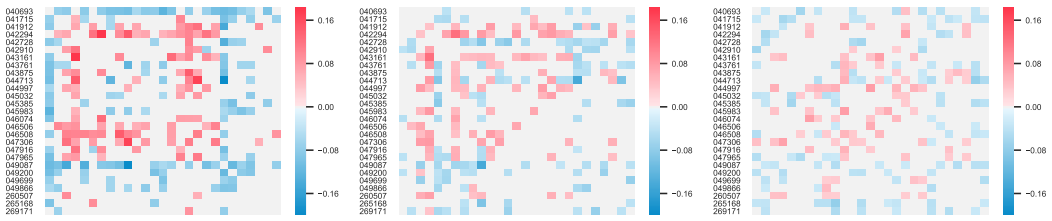

Figure 4.13: *Updated spatial graph by the low-dimensional procedure at different times.* $t = 507$ (left), $t = 1015$ (middle), and $t = 1522$ (right). Experiment settings: $N = 27$, $F = 4$, $M = 12$, number of model parameters $= 1761$, significance level of χ^2 test $= 0.1$, $\eta = 10^{-5}$, $t_0 = 20$, $\lambda_0 = 0.03$. The row labels are the 6-digit Cooperative Observer Identification Number of the corresponding weather stations.

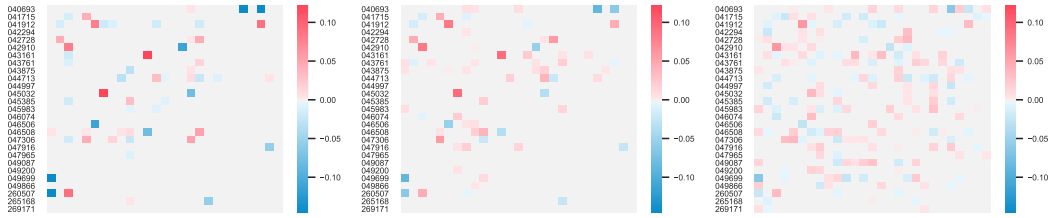

Figure 4.14: *Updated spatial graph by the high-dimensional procedure at different times.* $t = 507$ (left), $t = 1015$ (middle), and $t = 1522$ (right). Experiment settings: $N = 27$, $F = 4$, $M = 12$, number of model parameters $= 1761$, significance level of χ^2 test $= 0.1$, $\eta = 10^{-5}$, $t_0 = 20$, $\lambda_0 = 0.03$. The rows and columns correspond to the weather stations whose 6-digit Cooperative Observer Identification Number are given by the row labels.

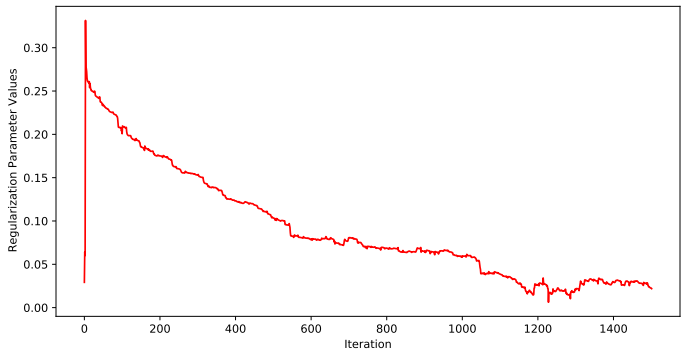

Figure 4.15: *Regularization parameter evolution.* Experiment settings: $N = 27$, $F = 4$, $M = 12$, number of model parameters $= 1761$, significance level of χ^2 test $= 0.1$, $\eta = 10^{-5}$, $t_0 = 20$, $\lambda_0 = 0.03$.

Next we show the last updated feature graphs in Figure 4.17. We can see that the estimated feature relationships from the two approaches coincide in tmin and tmax, tmin and tavg, tmin and prcp. However, the relationship between tavg and prcp is very weak in the Lasso estimation, while strong in the projected OLS estimation.

In particular, Figure 4.18 reports the evolution of estimated trends from one representative spatial location along time, where we can observe the increase of temperature from the past to the present.

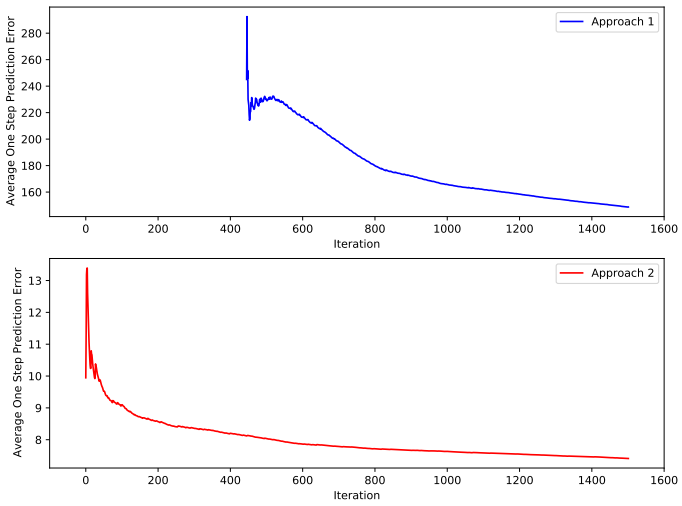

Figure 4.16: *Average one step prediction error of raw time series.* the low-dimensional procedure (top), and the high-dimensional procedure (bottom).

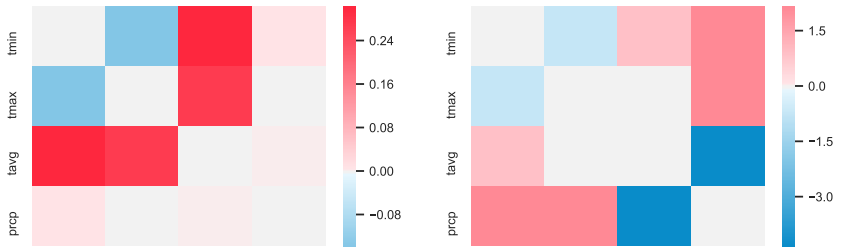

Figure 4.17: *Updated feature graph at* $t = 1522$. the low-dimensional procedure (left), and the high-dimensional procedure (right). Experiment settings: $N = 27$, $F = 4$, $M = 12$, number of model parameters $= 1761$, significance level of χ^2 test $= 0.1$, $\eta = 10^{-5}$, $t_0 = 20$, $\lambda_0 = 0.03$.

Lastly, in Figure 4.19, we plot the edge overlap (considering the signs of weights) of the two last updated spatial graphs, where we also visualize this spatial graph superimposed on the actual geographical graph. We can see that the remote weather stations have less dependency with other stations, while more edges appear within the area where lots of stations are densely located together. These observations imply that the inferred graphs provide the consistent weather patterns with geographical features. Furthermore, they validate the legitimacy of Models (4.1.4) and (4.3.1), as well as the effectiveness of the proposed learning methods.

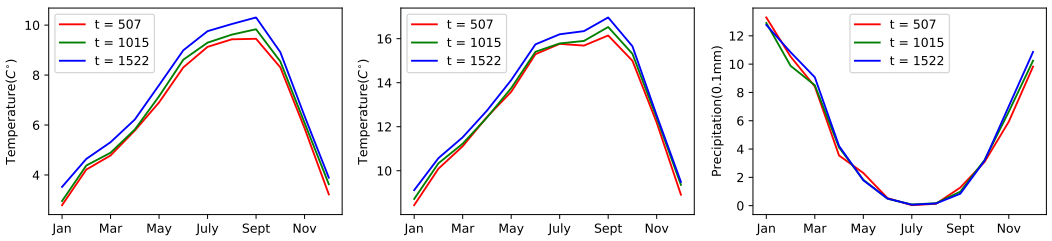

Figure 4.18: *Estimated trends along years.* On the left, middle, right are the estimated trends at different years of Station USH00040693 for minimal temperature, average temperature, and precipitation respectively. Experiment settings: $N = 27$, $F = 4$, $M = 12$.

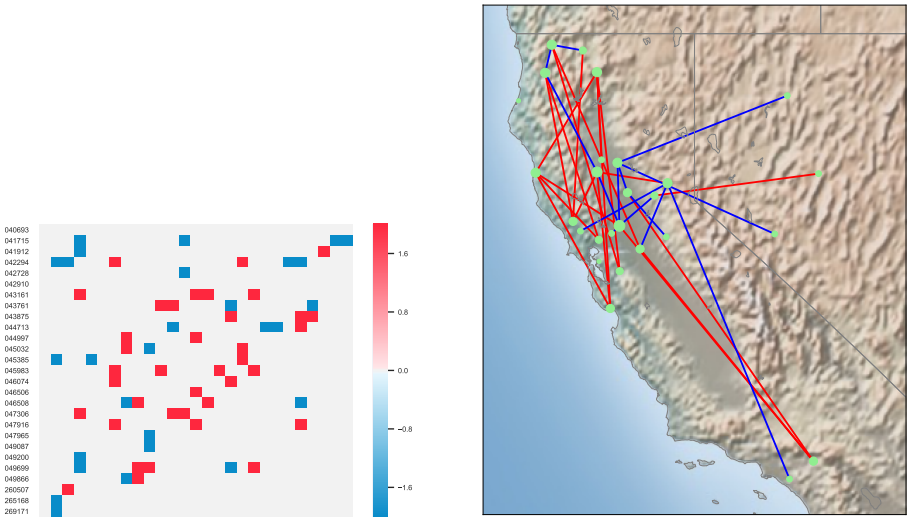

Figure 4.19: *Overlap spatial graph.* On the left is the adjacency matrix of an unweighed undirected graph which is the overlap of the two last updated spatial graphs in Figure 4.14, with the colors reporting the common edge signs. On the right is the visualization of this overlap spatial graph on the actually geographical map. The nodes with bigger sizes connect with more nodes.

4.5 Appendix

4.5.1 Proof of results in Section 4.2.2 and the CLT for $\widehat{\mathbf{A}}_t$

Proof of Theorem 4.2.3. By Cramér-Wold theorem, $\sqrt{t}\,\mathrm{vec}(\breve{\mathbf{A}}_t - A) \xrightarrow{d} \mathcal{N}(0, \Sigma_{ols})$ is equivalent to

$$\langle \Lambda, \sqrt{t}\left(\breve{\mathbf{A}}_t - A\right)\rangle \xrightarrow{d} \mathcal{N}(0, \mathrm{vec}(\Lambda)^\top \Sigma_{ols} \mathrm{vec}(\Lambda)), \quad \forall \Lambda \in \mathbb{R}^{NF\times NF}.$$

On the other hand, we can express the entries of $\mathrm{svec}\left(\sqrt{t}\left(\widehat{\boldsymbol{A}_{\mathbf{N}}} - A_{\mathrm{N}}\right)\right)$ as a linear function of $\breve{\mathbf{A}}_t$

$$\mathrm{svec}\left(\sqrt{t}\left(\widehat{\boldsymbol{A}_{\mathbf{N}}} - A_{\mathrm{N}}\right)\right) = \sum_{k\in K_{\mathrm{N}}} \langle U_k, \sqrt{t}\left(\breve{\mathbf{A}}_t - A\right)\rangle \mathrm{svec}(E_k).$$

Then for all $\lambda \in \mathbb{R}^{\frac{N(N-1)}{2}}$, we have

$$\lambda^\top \mathrm{svec}\left(\sqrt{t}\left(\widehat{\boldsymbol{A}_{\mathbf{N}}} - A_{\mathrm{N}}\right)\right) = \langle \sum_{k\in K_{\mathrm{N}}} \lambda^\top \mathrm{svec}(E_k) U_k, \sqrt{t}\left(\breve{\mathbf{A}}_t - A\right)\rangle.$$

Let Λ in Equation (4.5.1) be $\sum_{k\in K_{\mathrm{N}}} \lambda^\top \mathrm{svec}(E_k) U_k$, then we have

$$\lambda^\top \mathrm{svec}\left(\sqrt{t}\left(\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t} - A_{\mathrm{N}}\right)\right)\rangle \xrightarrow{d} \mathcal{N}(0, \mathrm{vec}(\Lambda)^\top \Sigma_{ols} \mathrm{vec}(\Lambda)).$$

Note that, $\mathrm{vec}(\Lambda) = \sum_{k\in K_{\mathrm{N}}} \lambda^\top \mathrm{svec}(E_k)\mathrm{vec}(U_k)$. Thus $\mathrm{vec}(\Lambda)^\top \Sigma_{ols} \mathrm{vec}(\Lambda) = \lambda^\top \Sigma_{\mathrm{N}} \lambda$. Use Cramér-Wold theorem again, we can get the theorem result.

■

Theorem 4.5.1. *(CLT for $\widehat{\mathbf{A}}_t$)*

$$\sqrt{t}\,\mathrm{vec}(\widehat{\mathbf{A}}_t - \mathbf{A}) \xrightarrow{d} \mathcal{N}(0, \Sigma_{\mathcal{G}})$$

where $\Sigma_{\mathcal{G}} = \sum_{k,k'\in K} \mathrm{vec}(U_k)^\top \Sigma_{ols} \mathrm{vec}(U_{k'}) \left[\mathrm{vec}(U_k)\mathrm{vec}(U_{k'})^\top\right]$.

Proof: The proof is similar as before. Because, we can express any entries of $\widehat{\mathbf{A}}_t$ as a linear function of $\breve{\mathbf{A}}_t$:

$$\widehat{\mathbf{A}}_t = \sum_{k\in K} \langle U_k, \breve{\mathbf{A}}_t\rangle U_k.$$

Thus, for all $\Lambda' \in \mathbb{R}^{NF\times NF}$, we have

$$\langle \Lambda', \sqrt{t}\left(\widehat{\mathbf{A}}_t - A\right)\rangle = \langle \sum_{k\in K} \langle \Lambda', U_k\rangle U_k, \sqrt{t}\left(\breve{\mathbf{A}}_t - A\right)\rangle.$$

Let Λ in Equation (4.5.1) be $\sum_{k\in K}\langle \Lambda', U_k\rangle U_k$, then

$$\langle \Lambda', \sqrt{t}\left(\widehat{\mathbf{A}}_t - A\right)\rangle \xrightarrow{d} \mathcal{N}(0, \mathrm{vec}(\Lambda')^\top \Sigma_{\mathcal{G}} \mathrm{vec}(\Lambda')).$$

Use Cramér-Wold theorem again, we can get the theorem result. The distribution in this theorem is degenerate.

■

Proof of Corollary 4.2.4. The proof is an adaption of Lütkepohl (2005, Section 3.6). We first construct the following matrix:

$$C = \begin{pmatrix} \vdots \\ \operatorname{svec}(E_{h_k})^\top \\ \vdots \end{pmatrix} \in \mathbb{R}^{P \times \frac{N(N-1)}{2}}.$$

Then test H_0 versus H_1 equals to

$$H_0' : C\operatorname{svec}(A_{\mathrm{N}}) = 0 \text{ versus } H_1' : C\operatorname{svec}(A_{\mathrm{N}}) \neq 0.$$

Following CLT 4.2.3, we have

$$\sqrt{t}\, C\operatorname{svec}(\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t} - A_{\mathrm{N}}) \xrightarrow{d} \mathcal{N}(0, C\Sigma_{\mathrm{N}}C^\top).$$

Hence, when H_0' holds,

$$\sqrt{t}\, C\operatorname{svec}(\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t}) \xrightarrow{d} \mathcal{N}(0, C\Sigma_{\mathrm{N}}C^\top).$$

Then by Proposition C.2 (4) in Lütkepohl (2005), we have

$$\sqrt{t}\left[C\hat{\boldsymbol{\Sigma}}_{\mathrm{N},t}C^\top\right]^{-\frac{1}{2}} C\operatorname{svec}(\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t}) \xrightarrow{d} \mathcal{N}(0, I_P),$$

where $\hat{\boldsymbol{\Sigma}}_{\mathrm{N},t} = \sum_{k,k' \in K_{\mathrm{N}}} \operatorname{vec}(U_k)^\top \hat{\boldsymbol{\Sigma}}_{ols,t} \operatorname{vec}(U_{k'}) \left(\operatorname{svec}(E_k)\operatorname{svec}(E_{k'})^\top\right)$ is the consistent estimator of $\boldsymbol{\Sigma}_{\mathrm{N}}$. Then by continuous mapping theorem:

$$t\,\hat{\boldsymbol{\alpha}}_t^\top \left[C\hat{\boldsymbol{\Sigma}}_{\mathrm{N},t}C^\top\right]^{-1} \hat{\boldsymbol{\alpha}}_t \xrightarrow{d} \chi^2(P).$$

Note that $C\operatorname{svec}(\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t}) = \hat{\boldsymbol{\alpha}}_t$, and $(\operatorname{svec}(E_k))_{k \in K_{\mathrm{N}}}$ are orthonormal basis in $\mathbb{R}^{\frac{N(N-1)}{2}}$, thus we have $C\hat{\boldsymbol{\Sigma}}_{\mathrm{N},t}C^\top = \hat{\boldsymbol{\Sigma}}_{W,t}$.

■

4.5.2 Proof of Proposition 4.3.1

From Definition (4.3.4), we have

$$\begin{aligned}
\hat{\boldsymbol{\Gamma}}_t(0) &= \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} \left(\frac{\sum\limits_{\tau \in I_{m,t}} \mathbf{x}_{\tau-1}\mathbf{x}_{\tau-1}^\top}{p_{m,t}} - \underline{\mathbf{x}}_{m-1,t}\underline{\mathbf{x}}_{m-1,t}^\top \right) \\
&= \frac{\sum\limits_{\tau=1}^{t} \mathbf{x}_{\tau-1}\mathbf{x}_{\tau-1}^\top}{t} - \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} \left(\underline{\mathbf{x}}_{m-1,t}\underline{\mathbf{x}}_{m-1,t}^\top\right).
\end{aligned}$$

Plug $\mathbf{x}_t = \mathrm{b}_t^0 + \mathbf{x}'_t$ in the last equation above, we can get the formula only with respect with $\mathbf{x}'_t$

$$\widehat{\mathbf{\Gamma}}_t(0) = \widehat{\mathbf{\Gamma}}_t(0)' - \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} \left(\underline{\mathbf{x}}'_{m-1,t}[\underline{\mathbf{x}}'_{m-1,t}]^\top\right),$$

where $\underline{\mathbf{x}}'_{m-1,t} = \sum_{\tau \in I_{m,t}} \frac{\mathbf{x}'_{\tau-1}}{p_{m,t}}$, $m = 0, ..., M-1$, and $\widehat{\mathbf{\Gamma}}_t(0)' := \frac{\sum_{\tau=1}^{t} \mathbf{x}'_{\tau-1}[\mathbf{x}'_{\tau-1}]^\top}{t}$. Note that $\underline{\mathbf{x}}'_{-1,t} = \underline{\mathbf{x}}'_{M-1,t}$.

Similarly, denote $\frac{\sum_{\tau=1}^{t} \mathbf{x}'_{\tau}[\mathbf{x}'_{\tau-1}]^\top}{t}$ by $\widehat{\mathbf{\Gamma}}_t(1)'$, we have

$$\widehat{\mathbf{\Gamma}}_t(1) = \widehat{\mathbf{\Gamma}}_t(1)' - \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} \left(\bar{\mathbf{x}}'_{m,t}[\underline{\mathbf{x}}'_{m-1,t}]^\top\right),$$

with $\bar{\mathbf{x}}'_{m,t} = \sum_{\tau \in I_{m,t}} \frac{\mathbf{x}'_{\tau}}{p_{m,t}}$, $m = 0, ..., M-1$. Since $(\mathbf{x}'_t)_t$ is the causal solution of VAR (4.1.4), we have

(a') $\widehat{\mathbf{\Gamma}}_t(0)' \xrightarrow{p} \Gamma(0)$, $\widehat{\mathbf{\Gamma}}_t(1)' \xrightarrow{p} \Gamma(1)$,

(b') $\widehat{\mathbf{\Gamma}}_t(1)' \left[\widehat{\mathbf{\Gamma}}_t(0)'\right]^{-1} \xrightarrow{p} \Gamma(1)\left[\Gamma(0)\right]^{-1} = A$,

(c') $\sqrt{t}\,\mathrm{vec}\left(\widehat{\mathbf{\Gamma}}_t(1)' \left[\widehat{\mathbf{\Gamma}}_t(0)'\right]^{-1} - A\right) \xrightarrow{d} \mathcal{N}(0, [\Gamma(0)]^{-1} \otimes \Sigma)$.

Thus, to reach the results in Proposition 4.3.1, we need additionally the asymptotic properties of sample mean $\bar{\mathbf{x}}'_{m,t}$, which are given in Lemma 4.5.2.

Lemma 4.5.2. *(CLT of $\bar{\mathbf{x}}'_{m,t}$)*

$$\sqrt{p_{m,t}}\bar{\mathbf{x}}'_{m,t} \xrightarrow{d} \mathcal{N}(0, \Phi\Sigma_M\Phi^\top), \quad \forall m = 0, ..., M-1,$$

where $\Phi = \left(I_{NF} - A^M\right)^{-1}$, *and* $\Sigma_M = \sum_{h=0}^{M-1} A^h \mathbf{\Sigma} (A^h)^\top$. *Therefore,* $\bar{\mathbf{x}}'_{m,t} \xrightarrow{p} 0$.

Proof of Lemma 4.5.2. Because of the periodicity, $(\mathbf{x}'_\tau)_{\tau \in I_{m,\infty}}$ is also a stationary process from VAR: $\tilde{\mathbf{X}}_{t'} = \mathbf{A}^M \tilde{\mathbf{X}}_{t'-1} + \tilde{\mathbf{z}}_{t'}$, with $\tilde{\mathbf{z}}_{t'} \sim \mathrm{IID}(0, \Sigma_M)$, for all $m = 0, ..., M-1$. Thus, apply Proposition 3.3 in Lütkepohl (2005), we get the result. ■

Proof of Proposition 4.3.1.

(a) When $t \to \infty$, $\widehat{\mathbf{\Gamma}}_t(0) = \widehat{\mathbf{\Gamma}}_t(0)' - \sum_{m=0}^{M-1} \frac{1}{M} \left(\bar{\mathbf{x}}'_{m,t}[\bar{\mathbf{x}}'_{m,t}]^\top\right) \xrightarrow{p} \Gamma(0) - 0 = \Gamma(0)$, and $\widehat{\mathbf{\Gamma}}_t(1) = \widehat{\mathbf{\Gamma}}_t(1)' - \sum_{m=0}^{M-1} \frac{1}{M} \left(\bar{\mathbf{x}}'_{m,t}[\bar{\mathbf{x}}'_{m-1,t}]^\top\right) \xrightarrow{p} \Gamma(1)$, with $\bar{\mathbf{x}}'_{-1,t} := \bar{\mathbf{x}}'_{M-1,t}$.

(b) $\bar{\mathbf{x}}_{m,t} = \frac{\sum_{\tau \in I_{m,t}} \mathrm{b}_m^0 + \mathbf{x}'_\tau}{p_{m,t}} = \mathrm{b}_m^0 + \bar{\mathbf{x}}'_{m,t} \xrightarrow{p} \mathrm{b}_m^0$, $\forall m = 0, ..., M-1$. Since asymptotically, $\bar{\mathbf{x}}_{m,t} = \underline{\mathbf{x}}_{m,t}$, thus both means can be used to estimate b_m^0. On the other

hand, based on (a), using continuous mapping theorem on the matrix inverse, we have $\breve{\mathbf{A}}_t = \widehat{\mathbf{\Gamma}}_t(1)\left[\widehat{\mathbf{\Gamma}}_t(0)\right]^{-1} \xrightarrow{p} A$.

(c) When $t \to \infty$, $\breve{\mathbf{A}}_t$ equals

$$\left[\widehat{\mathbf{\Gamma}}_t(1)' - \sum_{m=0}^{M-1} \frac{p_{m,t}}{t}\left(\bar{\mathbf{x}}'_{m,t}[\bar{\mathbf{x}}'_{m-1,t}]^\top\right)\right]\left[\widehat{\mathbf{\Gamma}}_t(0)' - \sum_{m=0}^{M-1} \frac{p_{m,t}}{t}\left(\bar{\mathbf{x}}'_{m-1,t}[\bar{\mathbf{x}}'_{m-1,t}]^\top\right)\right]^{-1}.$$

Use Woodbury formula on the matrix inverse, we have

$$\begin{aligned}\sqrt{t}(\breve{\mathbf{A}}_t - A) = \sqrt{t}(\widehat{\mathbf{\Gamma}}_t(1)'\left[\widehat{\mathbf{\Gamma}}_t(0)'\right]^{-1} - A) - \sqrt{t}\sum_{m=0}^{M-1} \frac{p_{m,t}}{t}\left(\bar{\mathbf{x}}'_{m,t}[\bar{\mathbf{x}}'_{m-1,t}]^\top\right)\left[\widehat{\mathbf{\Gamma}}_t(0)'\right]^{-1} \\ + \frac{\sqrt{t}}{1-g}\widehat{\mathbf{\Gamma}}_t(1)'\left[\widehat{\mathbf{\Gamma}}_t(0)'\right]^{-1}\sum_{m=0}^{M-1} \frac{p_{m,t}}{t}\left(\bar{\mathbf{x}}'_{m-1,t}[\bar{\mathbf{x}}'_{m-1,t}]^\top\right)\left[\widehat{\mathbf{\Gamma}}_t(0)'\right]^{-1} \\ -\frac{\sqrt{t}}{1-g}\sum_{m=0}^{M-1} \frac{p_{m,t}}{t}\left(\bar{\mathbf{x}}'_{m,t}[\bar{\mathbf{x}}'_{m-1,t}]^\top\right)\left[\widehat{\mathbf{\Gamma}}_t(0)'\right]^{-1}\sum_{m=0}^{M-1} \frac{p_{m,t}}{t}\left(\bar{\mathbf{x}}'_{m-1,t}[\bar{\mathbf{x}}'_{m-1,t}]^\top\right)\left[\widehat{\mathbf{\Gamma}}_t(0)'\right]^{-1},\end{aligned}$$

where, $g = \mathrm{tr}(\sum_{m=0}^{M-1} \frac{p_{m,t}}{t}\left(\bar{\mathbf{x}}'_{m-1,t}[\bar{\mathbf{x}}'_{m-1,t}]^\top\right)\left[\widehat{\mathbf{\Gamma}}_t(0)'\right]^{-1})$. Based on the result of (c'), to reach the same asymptotic distribution, we only need to show that, the reminder terms, namely from the second term to the last term above, all converge to 0 in probability.

From Slutsky's theorem and Lemma 4.5.2, we have the asymptotic result:

$$\forall m, \frac{p_{m,t}}{\sqrt{t}}\left(\bar{\mathbf{x}}'_{m,t}[\bar{\mathbf{x}}'_{m-1,t}]^\top\right) = \frac{1}{\sqrt{M}}\left(\sqrt{p_{m,t}}\bar{\mathbf{x}}'_{m,t}\right)[\bar{\mathbf{x}}'_{m-1,t}]^\top \xrightarrow{p} 0.$$

Thus, $\sqrt{t}\sum_{m=0}^{M-1} \frac{p_{m,t}}{t}\left(\bar{\mathbf{x}}'_{m,t}[\bar{\mathbf{x}}'_{m-1,t}]^\top\right)\left[\widehat{\mathbf{\Gamma}}_t(0)'\right]^{-1} \xrightarrow{p} 0$.

Similarly, $\sqrt{t}\sum_{m=0}^{M-1} \frac{p_{m,t}}{t}\left(\bar{\mathbf{x}}'_{m-1,t}[\bar{\mathbf{x}}'_{m-1,t}]^\top\right)\left[\widehat{\mathbf{\Gamma}}_t(0)'\right]^{-1} \xrightarrow{p} 0$. Since, it is obvious that $\sum_{m=0}^{M-1} \frac{p_{m,t}}{t}\left(\bar{\mathbf{x}}'_{m-1,t}[\bar{\mathbf{x}}'_{m-1,t}]^\top\right) \xrightarrow{p} 0$, then use the properties of convergence in probability and continuous mapping theorem, we can show the reminder terms all converge to 0 in probability.

■

4.5.3 Bisection Wald test for the identification of sparsity structure of A_{N}

Algorithm 2

Input: $\mathbf{x}_{t+1}, \mathbf{x}_t$, $\widehat{\mathbf{\Gamma}}_t(0)$, $\widehat{\mathbf{\Gamma}}_t(1)$, $[\widehat{\mathbf{\Gamma}}_t(0)]^{-1}$, t.
#Update:
$\widehat{\mathbf{\Gamma}}_{t+1}(1) = \frac{t}{t+1}\widehat{\mathbf{\Gamma}}_t(1) + \frac{1}{t+1}\mathbf{x}_{t+1}\mathbf{x}_t^\top$, $\widehat{\mathbf{\Gamma}}_{t+1}(0) = \frac{t}{t+1}\widehat{\mathbf{\Gamma}}_t(0) + \frac{1}{t+1}\mathbf{x}_t\mathbf{x}_t^\top$,
$[\widehat{\mathbf{\Gamma}}_{t+1}(0)]^{-1} = \frac{t+1}{t}[\widehat{\mathbf{\Gamma}}_t(0)]^{-1} - \frac{t+1}{t}\frac{[\widehat{\mathbf{\Gamma}}_t(0)]^{-1}\mathbf{x}_t\mathbf{x}_t^\top[\widehat{\mathbf{\Gamma}}_t(0)]^{-1}}{t+\mathbf{x}_t^\top[\widehat{\mathbf{\Gamma}}_t(0)]^{-1}\mathbf{x}_t}$,
$\widehat{\mathbf{\Sigma}}_{t+1} = \widehat{\mathbf{\Gamma}}_{t+1}(0) - \widehat{\mathbf{\Gamma}}_{t+1}(1)\widehat{\mathbf{\Gamma}}_{t+1}(0)^{-1}\widehat{\mathbf{\Gamma}}_{t+1}(1)^\top$,
$\check{\mathbf{A}}_{t+1} = \widehat{\mathbf{\Gamma}}_{t+1}(1)[\widehat{\mathbf{\Gamma}}_{t+1}(0)]^{-1}$.
#Projection:
$\widehat{\mathbf{A}}_{t+1} = \mathrm{Proj}_{\mathcal{G}}(\check{\mathbf{A}}_{t+1})$, retrieve $\widehat{\mathbf{A}_{\mathrm{D}}}_{,t+1}, \widehat{\boldsymbol{A}_{\mathbf{F}}}_{,t+1}, \widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t+1}$ using Equation (4.2.2).
Sort such that: $|(\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t+1})_{i_1,j_1}| \leqslant ... \leqslant |(\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t+1})_{i_{|K_{\mathrm{N}}|},j_{|K_{\mathrm{N}}|}}|$.
#Bisection Wald test procedure:
Initialize $p_l = 1$, $p_r = |K_{\mathrm{N}}|$, $p_m = \texttt{Floor}(\frac{p_l+p_r}{2})$.
Construct the corresponding test statistic $\lambda_{W,t+1}$ or $\lambda_{F,t+1}$ using Equation (4.2.3).
Perform tests $H(1)$ and $H(|K_{\mathrm{N}}|)$ based on Corollary 4.2.4.
if $H(1)$, $H(|K_{\mathrm{N}}|)$ are not rejected **then**
  $\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t+1} = 0$,
else
  if $H(1)$, $H(|K_{\mathrm{N}}|)$ are both rejected **then**
    No changes are made to $\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t+1}$,
  else
    while $p_l + 1 < p_r$ **do**
      $p_m \leftarrow \texttt{Floor}(\frac{p_l+p_r}{2})$, perform $H(p_m)$.
      if $H(p_m)$ is not rejected **then**
        $p_l \leftarrow p_m$,
      else
        $p_r \leftarrow p_m$.
      end if
    end while
    Let $(\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t+1})_{i_1,j_1} = ... = (\widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t+1})_{i_{p_l},j_{p_l}} = 0$.
  end if
end if
$\widehat{\mathbf{A}}_{t+1} \leftarrow \widehat{\mathbf{A}_{\mathrm{D}}}_{,t+1} + \widehat{\boldsymbol{A}_{\mathbf{F}}}_{,t+1} \otimes \widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t+1}$.
$t \leftarrow t + 1$.
Output: $\widehat{\mathbf{A}}_{t+1}, \widehat{\mathbf{\Gamma}}_{t+1}(0), \widehat{\mathbf{\Gamma}}_{t+1}(1), \widehat{\mathbf{\Gamma}}_{t+1}(0)^{-1}$, t.

Note that, since multiplication with $\mathrm{vec}(U_{h_k})$ amounts to extracting elements in the matrix from the corresponding locations, in practice, we take the elements

directly from $\left[\widehat{\mathbf{\Gamma}}_t(0)\right]^{-1}$ and $\widehat{\mathbf{\Sigma}}_t$, to compose $\widehat{\mathbf{\Sigma}}_{W,t}$ as:

$$\begin{aligned}\left(\widehat{\mathbf{\Sigma}}_{W,t}\right)_{k,k'} &= \left(\widehat{\mathbf{\Sigma}}_{W,t}\right)_{k',k} \\ &= \langle \mathbf{\Sigma}_{ii}^{k,k'}, \mathbf{\Gamma}_{jj}^{k,k'} \rangle + \langle \mathbf{\Sigma}_{jj}^{k,k'}, \mathbf{\Gamma}_{ii}^{k,k'} \rangle + \langle \mathbf{\Sigma}_{ij}^{k,k'}, \mathbf{\Gamma}_{ji}^{k,k'} \rangle + \langle \mathbf{\Sigma}_{ji}^{k,k'}, \mathbf{\Gamma}_{ij}^{k,k'} \rangle,\end{aligned}$$

where $\mathbf{\Sigma}_{ii}^{k,k'} = \left[\widehat{\mathbf{\Sigma}}_t\right]_{I_k,I_{k'}}$, $\mathbf{\Gamma}_{jj}^{k,k'} = \left[\widehat{\mathbf{\Gamma}}_t(0)^{-1}\right]_{J_k,J_{k'}}$, $\mathbf{\Sigma}_{jj}^{k,k'} = \left[\widehat{\mathbf{\Sigma}}_t\right]_{J_k,J_{k'}}$, $\mathbf{\Gamma}_{ii}^{k,k'} = \left[\widehat{\mathbf{\Gamma}}_t(0)^{-1}\right]_{I_k,I_{k'}}$, $\mathbf{\Sigma}_{ij}^{k,k'} = \left[\widehat{\mathbf{\Sigma}}_t\right]_{I_k,J_{k'}}$, $\mathbf{\Gamma}_{ji}^{k,k'} = \left[\widehat{\mathbf{\Gamma}}_t(0)^{-1}\right]_{J_k,I_{k'}}$, and $\Sigma_{ji}^{k,k'} = \left[\widehat{\mathbf{\Sigma}}_t\right]_{J_k,I_{k'}}$, $\mathbf{\Gamma}_{ij}^{k,k'} = \left[\widehat{\mathbf{\Gamma}}_t(0)^{-1}\right]_{I_k,J_{k'}}$, with order indices $I_k := \{i_k, i_k + F, ..., i_k + (N-1)F\}$, $I_{k'} := \{i_{k'}, i_{k'} + F, ..., i_{k'} + (N-1)F\}$, $J_k := \{j_k, j_k + F, ..., j_k + (N-1)F\}$, $J_{k'} := \{j_{k'}, j_{k'} + F, ..., j_{k'} + (N-1)F\}$.

4.5.4 Extended algorithm 2 for the augmented model

Algorithm 3

Input: $\mathbf{x}_{t+1}, \mathbf{x}_t$, $\widehat{\mathbf{\Gamma}}_t(0)$, $\widehat{\mathbf{\Gamma}}_t(1)$, $[\widehat{\mathbf{\Gamma}}_t(0)]^{-1}$, $\bar{m}$, t, $\{p_{m,t}\}_{m=0}^{M-1}$, $\{\underline{\mathbf{x}}_{m,t}\}_{m=0}^{M-1}$.

Update $\widehat{\mathbf{\Gamma}}_{t+1}(0)$, $\widehat{\mathbf{\Gamma}}_{t+1}(1)$ from Equation (4.3.5).

$[\widehat{\mathbf{\Gamma}}_{t+1}(0)]^{-1} = \frac{t+1}{t}[\widehat{\mathbf{\Gamma}}_t(0)]^{-1} - \frac{t+1}{t}\frac{[\widehat{\mathbf{\Gamma}}_t(0)]^{-1}(\mathbf{x}_t - \underline{\mathbf{x}}_{\bar{m}-1,t})(\mathbf{x}_t - \underline{\mathbf{x}}_{\bar{m}-1,t})^\top[\widehat{\mathbf{\Gamma}}_t(0)]^{-1}}{t(1+1/p_{\bar{m},t}) + (\mathbf{x}_t - \underline{\mathbf{x}}_{\bar{m}-1,t})^\top[\widehat{\mathbf{\Gamma}}_t(0)]^{-1}(\mathbf{x}_t - \underline{\mathbf{x}}_{\bar{m}-1,t})}$,

$\breve{\mathbf{A}}_{t+1} = \widehat{\mathbf{\Gamma}}_{t+1}(1)[\widehat{\mathbf{\Gamma}}_{t+1}(0)]^{-1}$.

$\widehat{\mathbf{\Sigma}}_{t+1} = \widehat{\mathbf{\Gamma}}_{t+1}(0) - \widehat{\mathbf{\Gamma}}_{t+1}(1)[\widehat{\mathbf{\Gamma}}_{t+1}(0)]^{-1}\widehat{\mathbf{\Gamma}}_{t+1}(1)^\top$.

Step *Projection* to *Bisection Wald test procedure* are identical to Algorithm 2.

Let $\widehat{\mathbf{A}}_{t+1} = \widehat{\mathbf{A}_{\mathrm{D}}}_{,t+1} + \widehat{\boldsymbol{A}_{\mathbf{F}}}_{,t+1} \otimes \widehat{\boldsymbol{A}_{\mathbf{N}}}_{,t+1}$.

Update: $\underline{\mathbf{x}}_{\bar{m}-1,t+1} \leftarrow \frac{p_{\bar{m},t}}{p_{\bar{m},t}+1}\underline{\mathbf{x}}_{\bar{m}-1,t} + \frac{1}{p_{\bar{m},t}+1}\mathbf{x}_t$, and $\underline{\mathbf{x}}_{m,t+1} \leftarrow \underline{\mathbf{x}}_{m,t}, \forall m \neq \bar{m}-1$.

$p_{\bar{m},t+1} \leftarrow p_{\bar{m},t} + 1$, and $p_{m,t+1} \leftarrow p_{m,t}, \forall m \neq \bar{m}$,

$t \leftarrow t+1$.

Output: $\widehat{\mathbf{A}}_{t+1}, \widehat{\mathbf{\Gamma}}_{t+1}(0), \widehat{\mathbf{\Gamma}}_{t+1}(1), [\widehat{\mathbf{\Gamma}}_{t+1}(0)]^{-1}$, t, $\{p_{m,t}\}_{m=0}^{M-1}$, $\{\underline{\mathbf{x}}_{m,t}\}_{m=0}^{M-1}$.

4.5.5 Homotopy algorithm for regularization path $\mathbf{A}(t, \lambda_1)$ to $\mathbf{A}(t, \lambda_2)$

Algorithm 4

Input: $N, F, \mathbf{\Gamma}_0, \boldsymbol{\gamma}_1, K_{\mathrm{N}}^1$ (ordered list), $\mathbf{w}_{\mathrm{N}}^1, \lambda_1, \lambda_2, \left[\mathbf{\Gamma}_0^1\right]^{-1}$, where $K_{\mathrm{N}}^1, \mathbf{w}_{\mathrm{N}}^1, \left[\mathbf{\Gamma}_0^1\right]^{-1}$ are associated with $\mathbf{A}(t, \lambda_1)$, and $\mathbf{w}_{\mathrm{N}}^1 = [\mathbf{w}]_{K_{\mathrm{N}}^1}$.

Initialization: $\lambda \leftarrow \lambda_1$, $K_{\mathrm{N}}^0 \leftarrow K_{\mathrm{N}} \backslash K_{\mathrm{N}}^1$, $K^1 \leftarrow K_{\mathrm{D}} + K_{\mathrm{F}} + K_{\mathrm{N}}^1$, where $+$ is the ordered append of two lists.

#Computing the regularization path (the steps in parentheses are the modifications for the case $\lambda_1 > \lambda_2$):

while $\lambda < \lambda_2$ (or $\lambda > \lambda_2$) **do**

Generate $\mathbf{\Gamma}_0^0, \boldsymbol{\gamma}_1^1, \boldsymbol{\gamma}_1^0, \mathbf{w}_1$, based on Proposition 4.2.6.

Compute λ_r (or λ_l), based on Equations (4.2.13) and (4.2.14).

if $\lambda_r < \lambda_2$ (or $\lambda_l > \lambda_2$) **then**

$\lambda = \lambda_r$ (or $\lambda = \lambda_l$),

#Update the active set and the sign vector:

if $[\mathbf{a}_1^s]_i$ becomes zero for some $k_i \in K^1$ and $k_i \in K_{\mathrm{N}}^1$, namely, λ comes from $\{\lambda_k^0\}_k$ **then**

$K_{\mathrm{N}}^1 \leftarrow K_{\mathrm{N}}^1 \backslash \{k\}$, $K^1 \leftarrow K^1 \backslash \{k\}$, $K_{\mathrm{N}}^0 \leftarrow K_{\mathrm{N}}^0 + \{k\}$.

Remove $[\mathbf{w}_{\mathrm{N}}^1]_{i-|K_{\mathrm{D}}|-|K_{\mathrm{F}}|}$ from $\mathbf{w}_{\mathrm{N}}^1$.

Remove the i-th row together with the i-th column from $\mathbf{\Gamma}_0^1$, and use Sherman Morrison formula to update $\left[\mathbf{\Gamma}_0^1\right]^{-1}$.

else if $[\mathbf{w}_0]_i$ reaches 1 for some $k_i \in K_{\mathrm{N}}^0$, namely, λ comes from $\{\lambda_k^+\}_k$ **then**

$K_{\mathrm{N}}^0 \leftarrow K_{\mathrm{N}}^0 \backslash \{k\}$, $K_{\mathrm{N}}^1 \leftarrow K_{\mathrm{N}}^1 + \{k\}$, $K^1 \leftarrow K^1 + \{k\}$.

Append 1 to the end of sign vector $\mathbf{w}_{\mathrm{N}}^1$.

Append row $[\mathbf{\Gamma}_0]_{k,K^1}$, column $[\mathbf{\Gamma}_0]_{K^1,k}$ after the last row and last column $\mathbf{\Gamma}_0^1$, respectively, and use Sherman Morrison formula to update $\left[\mathbf{\Gamma}_0^1\right]^{-1}$.

else if $[\mathbf{w}_0]_k$ reaches -1 for some $k_i \in K_{\mathrm{N}}^0$, namely, λ comes from $\{\lambda_k^-\}_k$ **then**

$K_{\mathrm{N}}^0 \leftarrow K_{\mathrm{N}}^0 \backslash \{k\}$, $K_{\mathrm{N}}^1 \leftarrow K_{\mathrm{N}}^1 + \{k\}$, $K^1 \leftarrow K^1 + \{k\}$.

Append -1 to the end of sign vector $\mathbf{w}_{\mathrm{N}}^1$.

Append row $[\mathbf{\Gamma}_0]_{k,K^1}$, column $[\mathbf{\Gamma}_0]_{K^1,k}$ after the last row and last column $\mathbf{\Gamma}_0^1$, respectively, and use Sherman Morrison formula to update $\left[\mathbf{\Gamma}_0^1\right]^{-1}$.

end if

else

$\lambda = \lambda_2$.

end if

end while

Compute $\mathbf{a}_1^s$, using Equation (4.2.12) and the last updated $\left[\mathbf{\Gamma}_0^1\right]^{-1}, \boldsymbol{\gamma}_1^1, \mathbf{w}_1$. Retrieve $\mathbf{A}(t, \lambda_2)$ from this $\mathbf{a}_1^s$.

Output: $\mathbf{A}(t, \lambda_2)$, K_{N}^1, $\mathbf{w}_{\mathrm{N}}^1$, $\left[\mathbf{\Gamma}_0^1\right]^{-1}$.

4.5.6 Homotopy algorithm for data path $\mathbf{A}(t, \frac{t+1}{t}\lambda)$ to $\mathbf{A}(t+1, \lambda)$

Algorithm 5

Input: $N, F, \mathbf{\Gamma}_0, \boldsymbol{\gamma}_1, K_{\mathrm{N}}^1$ (ordered list), $\mathbf{w}_{\mathrm{N}}^1, \lambda, \left[\mathbf{\Gamma}_0^1\right]^{-1}, \mathbf{x}_{t+1}, \widetilde{\mathbf{X}}_t, t$, where K_{N}^1, $\mathbf{w}_{\mathrm{N}}^1, \left[\mathbf{\Gamma}_0^1\right]^{-1}$ are associated with $\mathbf{A}(t, \frac{t+1}{t}\lambda)$, and $\mathbf{w}_{\mathrm{N}}^1 = [\mathbf{w}]_{K_{\mathrm{N}}^1}$.
Initialization: $\lambda \leftarrow \lambda_1$, $K_{\mathrm{N}}^0 \leftarrow K_{\mathrm{N}} \backslash K_{\mathrm{N}}^1$, $K^1 \leftarrow K_{\mathrm{D}} + K_{\mathrm{F}} + K_{\mathrm{N}}^1$, where $+$ is the ordered append of two lists.
for $i = 1, ..., NF$ **do**
  $\mu \leftarrow 0$.
  while $\mu < 1$ **do**
    Generate $\mathbf{\Gamma}_0^0$, $\boldsymbol{\gamma}_1^1$, $\boldsymbol{\gamma}_1^0$, $\mathbf{w}_1$, based on Proposition 4.2.6.
    $\underline{\mathbf{a}}_1^s = \left[\mathbf{\Gamma}_0^1\right]^{-1}(\boldsymbol{\gamma}_1^1 - (1+\frac{1}{t})\lambda \mathbf{w}_1)$,
    $e = \boldsymbol{x}_{t+1,i} - ([\widetilde{\mathbf{X}}_t]_{K^1,i})^\top \underline{\mathbf{a}}_1^s$, $\mathbf{u} = \left[\mathbf{\Gamma}_0^1\right]^{-1}[\widetilde{\mathbf{X}}_t]_{K^1,i}$. $\alpha = ([\widetilde{\mathbf{X}}_t]_{K^1,i})^\top \mathbf{u}$.
    $\mu_{k_i}^0 = -t(\underline{\mathbf{a}}_1^s)_i / (\alpha(\underline{\mathbf{a}}_1^s)_i + e(\mathbf{u})_i)$, $k_i \in K^1$ such that $k_i \in K_{\mathrm{N}}^1$,
    $\mu_{k_i}^{\pm} = \frac{-t(\mathrm{b}^{\pm})_i}{e(\mathbf{\Gamma}_0^0\mathbf{u})_i - e(\widetilde{\mathbf{X}}_t)_{k,i} + \alpha(\mathrm{b}^{\pm})_i}$, $k_i \in K_{\mathrm{N}}^0$, $\mathrm{b}^{\pm} = \mathbf{\Gamma}_0^0 \underline{\mathbf{a}}_1^s - \boldsymbol{\gamma}_1^0 \pm (1+\frac{1}{t})\lambda$,
    $\mu' = \min\big\{\min\{\mu_k^0, k \in K_{\mathrm{N}}^1 : \mu_k^0 > \mu\}, \min\{\mu_k^+, k \in K_{\mathrm{N}}^0 : \mu_k^+ > \mu\}, \min\{\mu_k^-, k \in K_{\mathrm{N}}^0 : \mu_k^- > \mu\}\big\}$.
    if $\mu' = \varnothing$, $\mu' \leftarrow +\infty$.
    if $\mu' < 1$ **then**
      $\mu = \mu'$.
      if μ' is some μ_k^0 **then**
        $K_{\mathrm{N}}^1 \leftarrow K_{\mathrm{N}}^1 \backslash \{k\}$, $K^1 \leftarrow K^1 \backslash \{k\}$, $K_{\mathrm{N}}^0 \leftarrow K_{\mathrm{N}}^0 + \{k\}$.
        Remove $[\mathbf{w}_{\mathrm{N}}^1]_{i-|K_{\mathrm{D}}|-|K_{\mathrm{F}}|}$ from $\mathbf{w}_{\mathrm{N}}^1$.
        Remove the i-th row, the i-th column from $\mathbf{\Gamma}_0^1$, use Sherman Morrison formula to update $\left[\mathbf{\Gamma}_0^1\right]^{-1}$.
      else if μ' is some μ_k^+ (or μ_k^-) **then**
        $K_{\mathrm{N}}^0 \leftarrow K_{\mathrm{N}}^0 \backslash \{k\}$, $K_{\mathrm{N}}^1 \leftarrow K_{\mathrm{N}}^1 + \{k\}$, $K^1 \leftarrow K^1 + \{k\}$.
        Append 1 (or -1) to the end of sign vector $\mathbf{w}_{\mathrm{N}}^1$.
        Append row $[\mathbf{\Gamma}_0]_{k,K^1}$, column $[\mathbf{\Gamma}_0]_{K^1,k}$ after the last row and last column $\mathbf{\Gamma}_0^1$, respectively, and use Sherman Morrison formula to update $\left[\mathbf{\Gamma}_0^1\right]^{-1}$.
      end if
    else
      $\mu = 1$.
    end if
  end while
  $[\mathbf{\Gamma}_0^1]^{-1} \xleftarrow{\text{rank } 1 \text{ update}} [\mathbf{\Gamma}_0^1 + \frac{1}{t}[\widetilde{\mathbf{X}}_t]_{K^1,i}([\widetilde{\mathbf{X}}_t]_{K^1,i})^\top]^{-1}$
  $\mathbf{\Gamma}_0 \leftarrow \mathbf{\Gamma}_0 + \frac{1}{t}[\widetilde{\mathbf{X}}_t]_{:,i}[\widetilde{\mathbf{X}}_t]_{:,i}^\top$. $\boldsymbol{\gamma}_1 \leftarrow \boldsymbol{\gamma}_1 + \frac{1}{t}\boldsymbol{x}_{t+1,i}[\widetilde{\mathbf{X}}_t]_{:,i}$
end for
$\mathbf{a}_1^s = \underline{\mathbf{a}}_1^s + e\mathbf{u}/(t+\alpha)$. Retrieve $\mathbf{A}(t+1, \lambda)$ based on K^1 and $\mathbf{a}_1^s$.
$[\mathbf{\Gamma}_0^1]^{-1} \leftarrow \frac{t+1}{t}[\mathbf{\Gamma}_0^1]^{-1}$, $\mathbf{\Gamma}_0 \leftarrow \frac{t}{t+1}\mathbf{\Gamma}_0$, $\boldsymbol{\gamma}_1 \leftarrow \frac{t}{t+1}\boldsymbol{\gamma}_1$.
Output: $\mathbf{A}(t+1, \lambda)$, K_{N}^1, $\mathbf{w}_{\mathrm{N}}^1$, $\left[\mathbf{\Gamma}_0^1\right]^{-1}$, $\mathbf{\Gamma}_0$, $\boldsymbol{\gamma}_1$.

4.5.7 Online graph and trend learning from matrix-variate time series in high-dimensional regime

Algorithm 6

Input: $\mathbf{A}(t,\lambda_t)$, $\boldsymbol{\Gamma}_0$, $\boldsymbol{\gamma}_1$, K_{N}^1 (ordered list), $\mathbf{w}_{\mathrm{N}}^1$, λ_t, $\left[\boldsymbol{\Gamma}_0^1\right]^{-1}$, $\mathbf{x}_{t+1}$, $\widetilde{\mathbf{X}}_t$, $\bar{m}$, t, M, $(p_{m,t})_{m=0}^{M-1}$, $(\underline{\mathbf{x}}_{m,t})_{m=0}^{M-1}$, $\mathbf{b}_{\bar{m},t}$, where K_{N}^1, $\mathbf{w}_{\mathrm{N}}^1$, $\left[\boldsymbol{\Gamma}_0^1\right]^{-1}$ are associated with $\mathbf{A}(t,\lambda_t)$.
Select λ_{t+1} according to the end of Section 4.3.2.
Update $\mathbf{A}(t,\lambda_t) \rightarrow \mathbf{A}(t,\frac{t+1}{t}\lambda_{t+1})$ using algorithm 4.
Center $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_{t+1} - \underline{\mathbf{x}}_{\bar{m},t}$. Compose $\underline{\widetilde{\mathbf{X}}}_{\bar{m}-1,t}$ as $[\underline{\widetilde{\mathbf{X}}}_{\bar{m}-1,t}]_{k,i} = [U_k]_{i,:}\underline{\mathbf{x}}_{\bar{m}-1,t}$, and center $\widetilde{\mathbf{X}}_t \leftarrow \widetilde{\mathbf{X}}_t - \underline{\widetilde{\mathbf{X}}}_{\bar{m}-1,t}$.
Update $\mathbf{A}(t,\frac{t+1}{t}\lambda_{t+1}) \rightarrow \mathbf{A}(t+1,\lambda_{t+1})$ using algorithm 5, with modifications:
Line 8 change to $\alpha = [\widetilde{\mathbf{X}}_t]_{K^1,i}^\top \mathbf{u} + p_{\bar{m},t}$,
Line 28, 29 change respectively to:
$[\boldsymbol{\Gamma}_0^1]^{-1} \overset{\text{rank 1 update}}{\longleftarrow} [\boldsymbol{\Gamma}_0^1 + \frac{p_{\bar{m},t}}{t(p_{\bar{m},t}+1)}[\widetilde{\mathbf{X}}_t]_{K^1,i}[\widetilde{\mathbf{X}}_t]_{K^1,i}^\top]^{-1}$
$\boldsymbol{\Gamma}_0 \leftarrow \boldsymbol{\Gamma}_0 + \frac{p_{\bar{m},t}}{t(p_{\bar{m},t}+1)}[\widetilde{\mathbf{X}}_t]_{:,i}[\widetilde{\mathbf{X}}_t]_{:,i}^\top$
$\boldsymbol{\gamma}_1 \leftarrow \boldsymbol{\gamma}_1 + \frac{p_{\bar{m},t}}{t(p_{\bar{m},t}+1)}\boldsymbol{x}_{t+1,i}[\widetilde{\mathbf{X}}_t]_{:,i}$
Update $\underline{\mathbf{x}}_{\bar{m}-1,t+1} \leftarrow \frac{p_{\bar{m},t}}{p_{\bar{m},t}+1}\underline{\mathbf{x}}_{\bar{m}-1,t} + \frac{1}{p_{\bar{m},t}+1}\mathbf{x}_t$, and $\underline{\mathbf{x}}_{m,t+1} \leftarrow \underline{\mathbf{x}}_{m,t}, \forall m \neq \bar{m}-1$.
$p_{\bar{m},t+1} \leftarrow p_{\bar{m},t} + 1$, and $p_{m,t+1} \leftarrow p_{m,t}, \forall m \neq \bar{m}$,
$\bar{m}' \leftarrow (t+2) \bmod M$.
$\mathbf{b}_{\bar{m}',t+1} \leftarrow \underline{\mathbf{x}}_{\bar{m}',t+1} - \mathbf{A}(t+1,\lambda_{t+1})\underline{\mathbf{x}}_{\bar{m},t+1}$,
$t \leftarrow t+1$.
Output: $\mathbf{A}(t+1,\lambda_{t+1})$, $\boldsymbol{\Gamma}_0$, $\boldsymbol{\gamma}_1$, K_{N}^1, $\mathbf{w}_{\mathrm{N}}^1$, λ_{t+1}, $\left[\boldsymbol{\Gamma}_0^1\right]^{-1}$, t, $(p_{m,t+1})_{m=0}^{M-1}$, $(\underline{\mathbf{x}}_{m,t+1})_{m=0}^{M-1}$, $\mathbf{b}_{\bar{m}',t+1}$.

Chapter 5

Characterisation of distributional time series over nodes

In this chapter, we model the multivariate distributional time series $(\boldsymbol{\mu}_t^i)_{t\in\mathbb{Z}}$, $i = 1, \ldots, N$ as random processes in the Wasserstein space $\mathcal{W}$ defined in Section 3.3, and propose a novel auto-regressive model to describe their dependency. In Section 5.1, we introduce the proposed time series model and analyze the existence, uniqueness and stationarity of its solution. Estimators of the model coefficients are studied in Section 5.2. Lastly in Section 5.3, we implement the numerical experiments with simulated data to verify the consistency of the proposed estimator. Additionally, we fit the proposed model on two real data sets to illustrate its applications in graph learning from the multivariate distributional time series.

For the sake of reproducible research, Python code available at

https://github.com/yiyej/Wasserstein_Multivariate_Autoregressive_Model

implements the proposed estimator and the experiments carried out in this paper.

5.1 Wasserstein multivariate AR Models

5.1.1 Description of the model

As explained in Section 2.2.1.7, the proposed multivariate distributional model is inspired by the intercept-free formulation of VAR(1) in Equation (3.2.6), which is built on the centered series $(\mathbf{x}_t - \mathrm{u})_{t\in\mathbb{Z}}$. We first rewrite the formulation element-wise to motivate the distributional construction as

$$\widetilde{\boldsymbol{x}}_t^i = \sum_{j=1}^{N} A_{ij}\widetilde{\boldsymbol{x}}_{t-1}^j + \boldsymbol{z}_t^i,\ t \in \mathbb{Z},\ i = 1, \ldots, N, \tag{5.1.1}$$

where $\widetilde{\boldsymbol{x}}_t^i = \boldsymbol{x}_t^i - u_i$, and $\boldsymbol{x}_t^i, u_i, \boldsymbol{z}_t^i$ are respectively the i-th component of $\mathbf{x}_t, \mathrm{u}$ and $\mathbf{z}_t$. We now extend VAR model by replacing each scalar time series $(\boldsymbol{x}_t^i)_t$ with univariate distributional series $(\boldsymbol{\mu}_t^i)_t$. The key point is to propose the centering notion for $(\boldsymbol{\mu}_t^i)_t$

using their common Fréchet mean. We therefore primarily assume the Fréchet mean of the series is time-invariant as below, as the assumption of formulation (5.1.1).

Assumption A3. For each fixed $i = 1, \dots, N$, the random probability measures $\boldsymbol{\mu}_t^i$, $t \in \mathbb{Z}$ are square integrable and they have the same Fréchet mean denoted by $\mu_{i,\oplus}$.

We propose the following "data centering step" for $\boldsymbol{\mu}_t^i$, $t \in \mathbb{Z}$, $i = 1, \dots, N$. The centered measures are denoted by $\widetilde{\boldsymbol{\mu}}_t^i$, and they are defined through their quantile functions given as

$$\widetilde{\boldsymbol{F}}_{i,t}^{-1} = \boldsymbol{F}_{i,t}^{-1} \ominus F_{i,\oplus}^{-1} := \boldsymbol{F}_{i,t}^{-1} \circ (F_{i,\oplus}^{-1})^{-1}, \tag{5.1.2}$$

where $\boldsymbol{F}_{i,t}^{-1}$ is the quantile function of $\boldsymbol{\mu}_t^i$ extended at 0 (Bobkov and Ledoux, 2014, Section A1) by

$$\boldsymbol{F}_{i,t}^{-1}(0) := \inf\{x \in [0,1] : \boldsymbol{F}_{i,t}(x) > 0\},$$

and $F_{i,\oplus}^{-1}$ is the quantile function of $\mu_{i,\oplus}$. Note that in most cases the function $(F_{i,\oplus}^{-1})^{-1}$ is equal to the cdf $F_{i,\oplus}$. However, when $F_{i,\oplus}$ is only right-continuous but not continuous, $(F_{i,\oplus}^{-1})^{-1}$ is not equal to $F_{i,\oplus}$, since the left-continuous inverse only gives a left-continuous function. Thus, we shall keep the notation $(F_{i,\oplus}^{-1})^{-1}$.

The centering step (5.1.2) at the level of the quantile functions is thus analogous to the usual centering step in VAR models for Euclidean data. From the optimal transport point of view, the centered quantile function $\widetilde{\boldsymbol{F}}_{i,t}^{-1}$ is interpreted as the optimal transport map from the Fréchet mean $\mu_{i,\oplus}$ to the measure $\boldsymbol{\mu}_t^i$. The notation $\ominus$ in (5.1.2) as a difference operator between two increasing functions is taken from the recent work in Zhu and Müller (2021) on auto-regressive model for univariate (that is $N = 1$) distributional time series. We remark that the output of this difference operator remains an increasing function.

The function $\widetilde{\boldsymbol{F}}_{i,t}^{-1}$ needs to be defined over $[0,1]$ as a valid quantile function. Furthermore, as a result of data centering, we aim to turn the Fréchet mean of $\widetilde{\boldsymbol{F}}_{i,t}^{-1}$ to be Lebesgue measure. Thus we impose Assumption A4 below.

Assumption A4. All $\boldsymbol{\mu}_t^i$, $t \in \mathbb{Z}$, $i = 1, \dots, N$ are supported on the same closed and bounded interval $\mathcal{D} \subset \Omega$. Without loss of generality, we assume that $\mathcal{D} = [0,1]$.

Under Assumption A4, the support of $\boldsymbol{\mu}_t^i$, $t \in \mathbb{Z}$, $i = 1, \dots, N$ is a closed and bounded interval, and thus, their cdf are strictly increasing function on this support [Proposition A7](Bobkov and Ledoux, 2014). Consequently, all the quantile function $\boldsymbol{F}_{i,t}^{-1}$ are continuous. Thus, the Fréchet mean has a continuous quantile function $F_{i,\oplus}^{-1}$. The continuity of $F_{i,\oplus}^{-1}$ makes Equation (5.1.3) holds (Bobkov and Ledoux, 2014, Lemma A.3 5).

$$\mathbb{E}\left[\widetilde{\boldsymbol{F}}_{i,t}^{-1}(p)\right] = (\mathbb{E}\boldsymbol{F}_{i,t}^{-1})\left[(F_{i,\oplus}^{-1})^{-1}(p)\right] = p, \quad p \in (0,1). \tag{5.1.3}$$

Thus, all the centered distributional time series $(\widetilde{\boldsymbol{\mu}}_t^i)_{t\in\mathbb{Z}}$ have the same Fréchet mean, which equals to the Lebesgue measure. We then propose to build an auto-regressive model for multivariate distributional time series with respect to the centered data $\widetilde{\boldsymbol{\mu}}_t^i$

in the tangent space of the Lebesgue measure, that takes the following expression:

$$\widetilde{\boldsymbol{\mu}}_t^i = \boldsymbol{\epsilon}_{i,t} \# \operatorname{Exp}_{Leb} \left(\sum_{j=1}^{N} A_{ij} \operatorname{Log}_{Leb} \widetilde{\boldsymbol{\mu}}_{t-1}^j \right), \quad t \in \mathbb{Z},\ i = 1, \ldots, N, \tag{5.1.4}$$

where $\{\boldsymbol{\epsilon}_{i,t}\}_{i,t}$ are i.i.d. random distortion functions taking values in the space of extended quantile functions

$$\Pi = \{F^{-1} : [0,1] \to [0,1],\ \text{such that } F^{-1}|_{(0,1)} \in \operatorname{Log}_{Leb} \mathcal{W} + id,$$
$$F^{-1}(0) := \inf\{x \in [0,1] : F(x) > 0\}, \text{and} F^{-1}(1) := \sup\{x \in [0,1] : F(x) < 1\}\},$$

endowed with $\|\cdot\|_{Leb}$ and the induced Borel algebra, $\boldsymbol{\epsilon}_{i,t}$ is almost surely independent of $\boldsymbol{\mu}_{t-1}^i$, $i = 1, \ldots, N$, for all $t \in \mathbb{Z}$, and

$$\mathbb{E}\left[\boldsymbol{\epsilon}_{i,t}(x)\right] = x,\ x \in [0,1].$$

Note that all the univariate time series of log maps are centered to 0, namely, $\mathbb{E}[\operatorname{Log}_{Leb} \widetilde{\boldsymbol{\mu}}_t^i] = 0$, $\forall t \in \mathbb{Z}$, $i = 1, ..., N$ as in Model (5.1.1).

The pushforward in (5.1.4) under $\boldsymbol{\epsilon}_{i,t}$ is a valid approach to provide random distortions of probability measures as proposed in Petersen and Müller (2019a). This approach is also used in Chen et al. (2021b). An example of random distortion function satisfying the conditions in Equation (5.1.4) as well as in Assumption A6 imposed later on, can be found, for example in Chen et al. (2021b, Equation (38)). However, in these works, not many examples of valid random distortion functions which satisfy the conditions in Equation (5.1.4) are given. Thus, to demonstrate that the conditions imposed on the distortion function are not restrictive, we describe, in Section 5.3 on numerical experiments, a general mechanism to generate random distortion functions that satisfy both Equation (5.1.4) and Assumption A6.

For the purpose of graph learning from distributional time series, a $N \times N$ matrix of coefficient matrix $A := (A_{ij})_{i,j}$ captures the dependency structure between N features characterized by time-dependent probability measures, and represents the structure directly in a directed and weighted graph, in contrast with using more complex nonparametric coefficient construction, for example Wang et al. (2015, Equations (14) and (15)).

To fit Model (5.1.4), a least squares estimator of the matrix A can be constructed by minimizing the expected squared Wasserstein distance (3.3.1) between $\widetilde{\boldsymbol{\mu}}_t^i$ and its prediction $\operatorname{Exp}_{Leb}\left(\sum_{j=1}^N A_{ij} \operatorname{Log}_{Leb} \widetilde{\boldsymbol{\mu}}_{t-1}^j\right)$. When $\sum_{j=1}^N A_{ij} \operatorname{Log}_{Leb} \widetilde{\boldsymbol{\mu}}_t^{j-1}$ belongs to $\operatorname{Log}_{Leb} \mathcal{W}$, the quantile function of its Exponential map Exp_{Leb} is simply given by $\sum_{j=1}^N A_{ij} \left(\widetilde{\boldsymbol{F}}_{j,t-1}^{-1} - id\right) + id$. By contrast, when $\sum_{j=1}^N A_{ij} \operatorname{Log}_{Leb} \widetilde{\boldsymbol{\mu}}_t^{j-1}$ falls out of $\operatorname{Log}_{Leb} \mathcal{W}$, the dependency between the quantile function and the coefficients A_{ij} is non-tractable, see Cazelles et al. (2017, Proposition 3.1). On the other hand, retaining the model in $\operatorname{Log}_{Leb} \mathcal{W}$ will avoid the non-identifiability problem of parametric models, thanks to Proposition 3.3.3. Thus, it will be needed to assume that $\sum_{j=1}^N A_{ij} \operatorname{Log}_{Leb} \widetilde{\boldsymbol{\mu}}_t^{j-1} \in \operatorname{Log}_{Leb} \mathcal{W}$. Since $\widetilde{\boldsymbol{\mu}}_t^{j-1}$ can take any value in $\mathcal{W}$, imposing such assumption amounts to the following N-simplex constraint on the rows of A, given the convexity of the logarithmic image. Similar assumptions are imposed in

related works, see e.g. Chen et al. (2021b, Assuption (A1)) and Petersen and Müller (2019b, Assumption (A3)), to keep the regression model in the logarithmic image.

Assumption A5. $\sum_{j=1}^{N} A_{ij} \leqslant 1$ and $0 \leqslant A_{ij} \leqslant 1$.

An additional important advantage of Assumption (A5) is that it leads to least squares estimation of the matrix A under an ℓ_1 ball constraint on its coefficients. In this manner, the estimators of the coefficients A_{ij} will naturally be sparse, which identifies the significant dependency links among the processes.

Given Assumption A5, we can also build the auto-regressive model directly with respect to the quantile function $\widetilde{\boldsymbol{F}}_{i,t}^{-1}$ as

$$\widetilde{\boldsymbol{F}}_{i,t}^{-1} = \boldsymbol{\epsilon}_{i,t} \circ \left[\sum_{j=1}^{N} A_{ij} \left(\widetilde{\boldsymbol{F}}_{j,t-1}^{-1} - id \right) + id \right], \quad t \in \mathbb{Z},\, i = 1, \ldots, N. \tag{5.1.5}$$

When reducing to the univariate case ($N = 1$), Model (5.1.5) is similar to the auto-regressive model proposed in Zhu and Müller (2021, Model (4)), when regression coefficient belongs to $(0, 1]$.

Remark 5.1.1. To deal with the unequal expectations of random measures, we may alternatively find a reference measure at whose tangent space, a proposed intercept-free regression holds in expectation. Unfortunately, such a valid reference measure is very unlikely to exist. For example, consider the following regression model of three random measures, $\boldsymbol{\mu}, \boldsymbol{\nu}_1$, and $\boldsymbol{\nu}_2$

$$\mathbb{E}\left[\mathrm{Log}_{\gamma}(\boldsymbol{\mu}) \,|\, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2\right] = \lambda_1 \mathrm{Log}_{\gamma}(\boldsymbol{\nu}_1) + \lambda_2 \mathrm{Log}_{\gamma}(\boldsymbol{\nu}_2),$$

at the tangent space of an unknown reference measure γ to be identified. A valid measure γ that makes the model hold in expectation, needs to satisfy

$$F_{\gamma}^{-1} = \frac{F_{\oplus,\mu}^{-1} - \lambda_1 F_{\oplus,\nu_1}^{-1} - \lambda_2 F_{\oplus,\nu_2}^{-1}}{(1 - \lambda_1 - \lambda_2)}. \tag{5.1.6}$$

However, the right hand side of the above equation is not necessarily an increasing function, which implies that a valid reference measure fails to exist in all negative cases. Furthermore, when plugging relationship (5.1.6) in the regression model, we obtain its reparameterization analogous to the reparameterization (3.2.6) of VAR(1) model

$$\mathbb{E}\left[\boldsymbol{F}_{\mu}^{-1} - F_{\oplus,\mu}^{-1} \,|\, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2\right] = \lambda_1(\boldsymbol{F}_{\nu_1}^{-1} - F_{\oplus,\nu_1}^{-1}) + \lambda_2(\boldsymbol{F}_{\nu_2}^{-1} - F_{\oplus,\nu_2}^{-1}).$$

However, the centering in this reparameterized model is linear given through the subtraction in Hilbert space, which does not bring well-defined new measures. Indeed, such centering can be equivalently obtained by adding an intercept term to the regression model established in the tangent space of Lebesgue measure. Therefore, by contrast to these two equivalent operations, we propose to consider nonlinear centering of quantile functions, when developing multivariate regression models in Wasserstein space.

5.1.2 Existence, uniqueness and stationarity

To study the legitimacy of the iterated random functions (IRF) system defined by Model (5.1.5), we shall consider the product metric space

$$(\mathcal{X}\,,\,d) := (\mathcal{T}\,,\,\|\cdot\|_{Leb})^{\otimes N}\,,$$

where $\mathcal{T} := \mathrm{Log}_{Leb}\,\mathcal{W} + id$ is the space of all quantile functions of $\mathcal{W}$, equipped with the norm $\|\cdot\|_{Leb}$ of the tangent space at Lebesgue measure. Thus, we have

$$d(\mathbf{X},\mathbf{Y}) := \sqrt{\sum_{i=1}^{N}\|\mathbf{X}_i - \mathbf{Y}_i\|_{Leb}^2}, \quad \mathbf{X} = (\mathbf{X}_i)_{i=1}^N \in \mathcal{X},\ \mathbf{Y} = (\mathbf{Y}_i)_{i=1}^N \in \mathcal{X}. \tag{5.1.7}$$

The auto-regressive model (5.1.5) can be interpreted as an iterated random functions (IRF) system operating on the state space $(\mathcal{X}\,,\,d)$, written as

$$\mathbf{X}_t = \mathbf{\Phi}_{\epsilon_t}\left(\mathbf{X}_{t-1}\right), \tag{5.1.8}$$

where $\mathbf{X}_t = (\mathbf{X}_{i,t})_{i=1}^N$, $\boldsymbol{\epsilon}_t = (\boldsymbol{\epsilon}_{i,t})_{i=1}^N$, and $\mathbf{\Phi}_{\epsilon_t}\left(\mathbf{X}_{t-1}\right) = (\mathbf{\Phi}^i_{\epsilon_t}\left(\mathbf{X}_{t-1}\right))_{i=1}^N$ with

$$\mathbf{\Phi}^i_{\epsilon_t}\left(\mathbf{X}_{t-1}\right) := \boldsymbol{\epsilon}_{i,t} \circ \left[\sum_{j=1}^{N} A_{ij}\left(\mathbf{X}_{j,t-1} - id\right) + id\right].$$

We first study the existence and the uniqueness of the solution to the IRF system in the metric space $(\mathcal{X}\,,\,d)$.

For time series models in a Hilbert space, two standard assumptions that ensure the existence and the uniqueness of the solutions are the boundedness of the L_p norm of random additive noise and the contraction of the regression operator. For Model (5.1.8), the random noise $\epsilon_{i,t}$ is bounded between 0 and 1, and thus $\mathbb{E}\left[d^p\left(\mathbf{X},\boldsymbol{\epsilon}\right)\right]$ is bounded for all $\mathbf{X} \in \mathcal{X}$, which is the L_p norm equivalent in the metric space setting. Then, to have a contractive map $\mathbf{\Phi}_{\epsilon_t}$, we shall rely on an interplay between properties of the matrix A of coefficients and the random noise distortion since it is applied in a nonlinear way. More specifically, we impose Assumptions A6 and A7 below on Model (5.1.8).

Assumption A6. $\mathbb{E}\left[\boldsymbol{\epsilon}_{i,t}(x) - \boldsymbol{\epsilon}_{i,t}(y)\right]^2 \leqslant L^2(x-y)^2,\ \forall x,y \in [0,1],\quad t \in \mathbb{Z},\, i = 1,\ldots,N,$

Assumption A7. $\|A\|_2 < \frac{1}{L}$.

Note that, Assumption A6 implies that $\boldsymbol{\epsilon}_{i,t}$ is L-Lipschitz in expectation. For increasing functions from $[0,1]$ to $[0,1]$, the smallest L is 1 that is attained by the identity function. Therefore, Assumption A7 implies that $\|A\|_2 < 1$, which is the contraction assumption for VAR(1) model. We now state the existence and uniqueness results. The proofs for this section is given in the appendix.

Theorem 5.1.2. *Under Assumptions A5, A6 and A7, the IRF system (5.1.8) almost surely admits a solution $\mathbf{X}_t$, $t \in \mathbb{Z}$, with the same marginal distribution π, namely, $\mathbf{X}_t \overset{d}{=} \pi$, $\forall\, t \in \mathbb{Z}$, where the notation $\overset{d}{=}$ means equality in distribution. Moreover, if there exists another solution $\boldsymbol{S}_t$, $t \in \mathbb{Z}$, then for all $t \in \mathbb{Z}$*

$$\mathbf{X}_t \overset{d}{=} \boldsymbol{S}_t, \quad \textit{almost surely.}$$

Theorem 5.1.2 states that under Assumptions A6 and A7, a well-defined IRF system (5.1.8) (namely when Assumption A5 is satisfied) permits a unique solution in $(\mathcal{X}, d)$ almost surely. Next, we show that this solution is furthermore stationary as a functional time series in a Hilbert space. To this end, we need to assume that there is an underlying Hilbert space associated to $(\mathcal{X}, d)$, with its inner product inducing d as the norm. Such an Hilbert space exists, with corresponding inner product given by

$$\langle X, Y \rangle = \sum_{i=1}^{N} \langle X_i, Y_i \rangle_{Leb} .$$

Then, Theorem 5.1.3 below gives the stationarity result.

Theorem 5.1.3. *The unique solution given in Theorem 5.1.2 is stationary as a random process in $(\mathcal{X}, \langle \cdot, \cdot \rangle)$ in the sense of Definition 3.4.7.*

Besides, Proposition 5.1.4 below states that the stationary solution of the IRF system (5.1.8) satisfies the property (5.1.3) of the transformed series $\widetilde{\boldsymbol{F}}_{i,t}^{-1}, t \in \mathbb{Z}$. Thus, it is consistent to propose the IRF system (5.1.8) as the process that generated the data $\widetilde{\boldsymbol{F}}_{i,t}^{-1}$, which completes the building of Model (5.1.5) as valid approach to analyze multivariate distributional time series.

Proposition 5.1.4. *The stationary solution $\mathbf{X}_t$ of the IRF system (5.1.8) satisfies:*

1. $\mathbf{X}_{i,t}(p) \in [0,1], \quad \forall p \in (0,1)$,
2. $\mathbb{E}\left[\mathbf{X}_{i,t}(p)\right] = p, \quad \forall p \in (0,1)$.

Finally, we point out in Proposition 5.1.5 additional properties of the IRF system (5.1.8) that will serve in the following section of the estimation of coefficients in Model (5.1.5).

Proposition 5.1.5. *The matrix A of coefficients in the stationary IRF system (5.1.8) admits the representation*

$$A = \Gamma_H(1) \left[\Gamma_H(0)\right]^{-1}, \tag{5.1.9}$$

where $\Gamma_H(0)$, and $\Gamma_H(1) \in \mathbb{R}^{N \times N}$ are defined as

$$\begin{aligned} \left[\Gamma_H(0)\right]_{j,l} &= \mathbb{E}\langle \mathbf{X}_{j,t-1} - id\,,\, \mathbf{X}_{l,t-1} - id \rangle_{Leb} \\ \left[\Gamma_H(1)\right]_{j,l} &= \mathbb{E}\langle \mathbf{X}_{j,t} - id\,,\, \mathbf{X}_{l,t-1} - id \rangle_{Leb}, \end{aligned}$$

for $1 \leqslant j, l \leqslant N$.

Note that representation (5.1.9) of coefficient in the proposed model is analogous to the one for VAR(1) models given in Equation (3.2.7), with matrices $\Gamma_H(0), \Gamma_H(1)$ carrying out the information on the correlation. However, compared to the auto-covariance operators in Definition 3.4.7, matrices $\Gamma_H(0)$ and $\Gamma_H(1)$ rather reflect the average auto-covariance taking into account additionally the correlated level along the function domain. To avoid the heavy notation in the estimators of matrices $\Gamma_H(0)$ and $\Gamma_H(1)$, in the rest of this chapter, we abuse the notation by omitting the subscript $_H$ in these matrices.

5.2 Estimation of the regression coefficients

In this section, we develop the estimators of coefficient A, given $T+1$ samples $\boldsymbol{\mu}_t^i$, $t = 0, 1, \ldots, T$ for each feature $i = 1, \ldots, N$. We also show the consistency result of the proposed estimator. Note that we assume that the measures are fully observed, instead of indirectly observed through their samples.

5.2.1 A constrained least-square estimation method

As briefly explained before the statement of Assumption (A5), we could consider the estimator based on an unconstrained least squares method, which is defined as the minimizer of the sum of squared residuals measured by the Wasserstein distance:

$$\begin{aligned}\widetilde{\boldsymbol{A}}_{i:} &= \underset{A_{i:}}{\arg\min} \frac{1}{T}\sum_{t=1}^{T} d_W^2\left[\widetilde{\boldsymbol{\mu}}_t^i\,,\, \mathrm{Exp}_{Leb}\left(\sum_{j=1}^{N} A_{ij}\,\mathrm{Log}_{Leb}\,\widetilde{\boldsymbol{\mu}}_t^{j-1}\right)\right], \quad i = 1, \ldots, N, \\ &= \underset{A_{i:}}{\arg\min} \frac{1}{T}\sum_{t=1}^{T} \left\| \widetilde{\boldsymbol{F}}_{i,t}^{-1} - \sum_{j=1}^{N} A_{ij}\left(\widetilde{\boldsymbol{F}}_{j,t-1}^{-1} - id\right) - id \right\|_{Leb}^2, \quad i = 1, \ldots, N,\end{aligned} \tag{5.2.1}$$

Analogous to Proposition 5.1.5, the estimator $\widetilde{\boldsymbol{A}}$ defined in Equation (5.2.1) admits the expression

$$\widetilde{\boldsymbol{A}} = \widetilde{\boldsymbol{\Gamma}}(1)\left[\widetilde{\boldsymbol{\Gamma}}(0)\right]^{-1},$$

where

$$\left[\widetilde{\boldsymbol{\Gamma}}(0)\right]_{j,l} = \frac{1}{T}\sum_{t=1}^{T} \langle \widetilde{\boldsymbol{F}}_{j,t-1}^{-1} - id\,,\, \widetilde{\boldsymbol{F}}_{l,t-1}^{-1} - id \rangle_{Leb}$$

and

$$\left[\widetilde{\boldsymbol{\Gamma}}(1)\right]_{j,l} = \frac{1}{T}\sum_{t=1}^{T} \langle \widetilde{\boldsymbol{F}}_{j,t}^{-1} - id\,,\, \widetilde{\boldsymbol{F}}_{l,t-1}^{-1} - id \rangle_{Leb}.$$

Note that $\widetilde{\boldsymbol{A}}$ is the exact least squares estimator constructed from the stationary solution of Model (5.1.5) without any constraint. However, in practice, we do not know the population Fréchet mean $F_{i,\oplus}^{-1}$, thus we can not calculate the exact centered data $\widetilde{\boldsymbol{\mu}}_t^i$ as in method (5.1.2). Therefore, we propose to first estimate $F_{i,\oplus}^{-1}$ by the empirical Fréchet mean

$$\boldsymbol{F}_{\bar{\mu}_i}^{-1} = \frac{1}{T}\sum_{t=1}^{T} F_{\boldsymbol{\mu}_{i,t}}^{-1}, \tag{5.2.2}$$

and center $\boldsymbol{\mu}_{i,t}$ by $\boldsymbol{F}_{\bar{\mu}_i}^{-1}$ as in Equation (5.2.3), to obtain the transformed data $\widehat{\boldsymbol{\mu}}_{i,t}$.

$$\widehat{\boldsymbol{F}}_{i,t}^{-1} := \boldsymbol{F}_{i,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_i}^{-1} = \boldsymbol{F}_{i,t}^{-1} \circ \boldsymbol{F}_{\bar{\mu}_i}. \tag{5.2.3}$$

Using the data $\widehat{\boldsymbol{\mu}}_{i,t}$ in the least squares formula (5.2.1) we obtain an approximate least squares estimator $\widehat{\boldsymbol{A}}_o$ whose rows satisfy

$$\left[\widehat{\boldsymbol{A}}_o\right]_{i:} = \underset{A_{i:}}{\arg\min} \frac{1}{T}\sum_{t=1}^{T} \left\| \widehat{\boldsymbol{F}}_{i,t}^{-1} - \sum_{j=1}^{N} A_{ij}\left(\widehat{\boldsymbol{F}}_{j,t-1}^{-1} - id\right) - id \right\|_{Leb}^2, \quad i = 1, \ldots, N,$$

Note that, since the empirical Fréchet means of all $\widehat{\boldsymbol{\mu}}_{i,t}$ are also uniform, similarly to the population setting, we do not need to consider the additional terms in the estimation problem to cancel out the unequal empirical Fréchet means. Analogously,

$$\widehat{\boldsymbol{A}}_o = \widehat{\boldsymbol{\Gamma}}(1)\left[\widehat{\boldsymbol{\Gamma}}(0)\right]^{-1}, \tag{5.2.4}$$

where

$$[\widehat{\boldsymbol{\Gamma}}(0)]_{j,l} = \frac{1}{T}\sum_{t=1}^{T}\langle \widehat{\boldsymbol{F}}_{j,t-1}^{-1} - id\,,\, \widehat{\boldsymbol{F}}_{l,t-1}^{-1} - id\rangle_{Leb}$$

and

$$[\widehat{\boldsymbol{\Gamma}}(1)]_{j,l} = \frac{1}{T}\sum_{t=1}^{T}\langle \widehat{\boldsymbol{F}}_{j,t}^{-1} - id\,,\, \widehat{\boldsymbol{F}}_{l,t-1}^{-1} - id\rangle_{Leb}.$$

Finally, we add the coefficient constraints to the problem, corresponding to the simplex constraint (A5). Therefore, the estimator $\widehat{\boldsymbol{A}}$ that we finally propose is defined as

$$\widehat{\boldsymbol{A}}_{i:} = \underset{A_{i:}\in B_+^1}{\arg\min}\frac{1}{T}\sum_{t=1}^{T}\left\|\widehat{\boldsymbol{F}}_{i,t}^{-1} - \sum_{j=1}^{N}A_{ij}\left(\widehat{\boldsymbol{F}}_{j,t-1}^{-1} - id\right) - id\right\|_{Leb}, \quad i = 1,\dots,N, \tag{5.2.5}$$

where B_+^1 is N-dimensional simplex, that is the nonnegative orthant of the ℓ_1 unit ball B^1 in $\mathbb{R}^N$. Thus, an important advantage of this constraint is to promote sparsity in $\widehat{\boldsymbol{A}}_{i:}$, which will be illustrated in Section 5.3. The optimization problem (5.2.5) can be solved by the accelerated projected gradient descent (Parikh and Boyd, 2014, Chapter 4.3). The projection onto B_+^1 is given in Thai et al. (2015).

5.2.2 Consistency of the estimators

Now, we study the consistency of the proposed estimator $\widehat{\boldsymbol{A}}$. The main result of this section is Theorem 5.2.5. The details of its proof is given in the appendix of this chapter. Instead in this section, we resume the proof by the key intermediate results in its development.

The proof proceeds by firstly showing the consistency of the unconstrained least squares estimator $\widetilde{\boldsymbol{A}}$ (see Lemma 5.2.1) that uses the knowledge of the population Fréchet mean $F_{i,\oplus}^{-1}$. Secondly, we show $\boldsymbol{F}_{\bar{\mu}_i}^{-1} \xrightarrow{p} F_{i,\oplus}^{-1}$ element-wise (see Lemma 5.2.2), and aim to rely on this result to prove that $\widehat{\boldsymbol{A}}_o - \widetilde{\boldsymbol{A}} \xrightarrow{p} 0$ (see Theorem 5.2.3). Thirdly, we show a general result (see Theorem 5.2.4 below) on the consistency of constrained estimators that is not restricted to the proposed estimator in this work. Finally, we apply this result to $\widehat{\boldsymbol{A}}_o$ to derive the consistency of the proposed estimator $\widehat{\boldsymbol{A}}$.

Lemma 5.2.1. *Assume that $\boldsymbol{\mu}_t^i$, $i = 1,\dots,N$ satisfy Assumption A3 for $t = 0,1,\dots,T$. Assume also that the transformed sequence $\widetilde{\boldsymbol{F}}_t^{-1}$, $t = 0,1,\dots,T$ satisfies Model (5.1.8) with Assumption A5 true. Suppose additionally that $\widetilde{\boldsymbol{F}}_0^{-1} \overset{d}{=} \pi$ with π the stationary distribution defined in Theorem 5.1.2. Given Assumptions A6 and A7 hold true, we obtain*

$$\widetilde{\boldsymbol{A}} - A = \mathcal{O}_p(\frac{1}{\sqrt{T}}).$$

Lemma 5.2.2. *Under the conditions of Lemma 5.2.1, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\widetilde{\boldsymbol{F}}_{i,t-1}(p) - p = \mathcal{O}_p(\frac{1}{\sqrt{T}}), \quad \forall p \in (0,1),\ i = 1,\dots N. \tag{5.2.6}$$

Since $\widetilde{\boldsymbol{F}}_{i,t} = \boldsymbol{F}_{i,t}^{-1} \ominus F_{i,\oplus}^{-1}$, *we have equivalently,*

$$\boldsymbol{F}_{\bar{\mu}_i}^{-1}(p) - F_{i,\oplus}^{-1}(p) = \mathcal{O}_p(\frac{1}{\sqrt{T}}), \quad \forall p \in (0,1),\ i = 1,\dots N. \tag{5.2.7}$$

Theorem 5.2.3. *Under the conditions of Lemma 5.2.1*

$$\widehat{\boldsymbol{A}}_o - \widetilde{\boldsymbol{A}} \xrightarrow{p} 0,$$

which implies

$$\widehat{\boldsymbol{A}}_o - A \xrightarrow{p} 0.$$

Theorem 5.2.4. *Let* $\beta^* \in \mathbb{R}^n$ *be some constant of interest. Assume* $\hat{\boldsymbol{\beta}}_o$ *is an estimator, defined as:*

$$\hat{\boldsymbol{\beta}}_o = \underset{\beta\in\mathbb{R}^n}{\arg\min}\, \boldsymbol{f}_T(\beta)$$

which converges to β^* *in probability at* $\mathcal{O}_p(r_T)$. *We define then the constrained estimator* $\hat{\boldsymbol{\beta}}$ *as:*

$$\begin{aligned} \hat{\boldsymbol{\beta}} = \underset{\beta\in\mathbb{R}^n}{\arg\min} & \boldsymbol{f}_T(\beta) \\ \text{subject to: } & f_i(\beta) \leqslant 0, i = 1,...,m \\ & h_j(\beta) = 0, j = 1,...,p. \end{aligned} \tag{5.2.8}$$

If $\boldsymbol{f}_T$ *is strongly convex with strong convexity constant* $\boldsymbol{\mu}_T$ *satisfying* $\frac{1}{2\boldsymbol{\mu}_T} = \mathcal{O}_p(1)$, *the strong duality holds for Problem (5.2.8) and* β^* *satisfies the constraints, then* $\hat{\boldsymbol{\beta}}$ *is also a consistent estimator of* β^*, *with* $\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_o = \mathcal{O}_p(r_T)$.

The result is evident from the other direction, since $\hat{\boldsymbol{\beta}}_o \xrightarrow{p} \beta^*$ and β^* satisfies the constraints, then $\hat{\boldsymbol{\beta}}_o$ should gradually approach the constraint set as the simple size T increases. Once $\hat{\boldsymbol{\beta}}_o$ itself satisfies the constraint, $\hat{\beta}$ takes $\hat{\boldsymbol{\beta}}_o$, which can be only asymptotic though.

Many results establish conditions under which strong duality holds. For example when the objective function is convex with an open domain and the constraints $f_i,\ i = 1,...,m,\ h_j,\ j = 1,...,p$ are all affine, then strong duality holds as soon as the problem is feasible (Boyd et al., 2004, Section 5.2.3), which is the case of Problem (5.2.8). Note that, the problem can be represented in vector form as

$$\widehat{\boldsymbol{A}}_{i:} = \underset{A_{i:}\in B_+^1}{\arg\min}\, \frac{1}{2}\boldsymbol{A}_{i:}\left[\widehat{\boldsymbol{\Gamma}}(0)\right]\boldsymbol{A}_{i:}^\top - \left[\widehat{\boldsymbol{\Gamma}}(1)\right]_{i,:}\boldsymbol{A}_{i:}^\top, \quad i = 1,\dots,N.$$

Since Theorem 5.2.3 shows the constraint free least square estimator on the approximate series converges to A in probability, applying Theorem 5.2.4 on Theorem 5.2.3, we finally obtain the desired consistency result, given in Theorem 5.2.5.

Theorem 5.2.5. *Under the conditions of Lemma 5.2.1, assume* $\frac{1}{4\boldsymbol{\lambda}_T} = \mathcal{O}_p(1)$ *where* $\boldsymbol{\lambda}_T$ *is the smallest eigenvalue of* $\widehat{\boldsymbol{\Gamma}}(0)$*, then given the true coefficient* A *satisfies Assumption A5, namely,* $A_{i:} \in B_+^1$*,* $i = 1, \ldots, N$*, we have*

$$\widehat{\boldsymbol{A}} - A \xrightarrow{p} 0.$$

Note that the condition on the eigenvalue of $\widehat{\boldsymbol{\Gamma}}(0)$ is very light, because of the non-linearity in the calculus $\widehat{\boldsymbol{\Gamma}}(0)$, in practice with only a very few samples the matrix will become and stay invertible.

5.3 Numerical experiments

In Section 5.3.1, we firstly demonstrate the consistency result of the proposed estimator using synthetic data. Then, we fit the model on two real data sets in Section 5.3.2 and Section 5.3.3. For each data set, the estimated coefficient matrix $\widehat{\boldsymbol{A}}$ allows us to understand the dependency structure of the multivariate distributional time series. In particular, we visualize the learned structure between features using a directed weighted graph with adjacency matrix $\widehat{\boldsymbol{A}}$.

5.3.1 Simulations

5.3.1.1 Generation of the synthetic data

We firstly propose a mechanism to generate the valid random distortion functions. To this end, we consider the random functions defined by

$$\boldsymbol{\epsilon}_g = \frac{1+\boldsymbol{\xi}}{2} g \circ h^{-1} + \frac{1-\boldsymbol{\xi}}{2} h^{-1}, \tag{5.3.1}$$

where g is a non-decreasing right-continuous constant function from $[0,1]$ to $[0,1]$, h^{-1} is the left continuous inverse of $h = \frac{1}{2}(g + id)$, and $\boldsymbol{\xi} \sim U(-1,1)$ is a random variable. For any given function g, we can sample a family of distortion functions $\boldsymbol{\epsilon}_{i,t} \overset{i.i.d.}{\sim} \boldsymbol{\epsilon}_g$, when sampling $\boldsymbol{\xi}_{i,t} \overset{i.i.d.}{\sim} U(-1,1)$. This construction of random distortion functions is inspired by the one proposed in Zhu and Müller (2021, Equation (13)), however, we have modified their construction of h and of the random coefficients. It is easy to verify that

$$\mathbb{E}[\boldsymbol{\epsilon}_g] = \frac{1}{2}(g \circ h^{-1} + h^{-1}) = \frac{1}{2}(g + id) \circ h^{-1} = id.$$

To make $\boldsymbol{\epsilon}_{i,t}$ satisfy additionally Model (5.1.4) and Assumption A6, we require g to be furthermore continuous and differentiable. Then on the one hand, since g is continuous and non-decreasing, any generated $\boldsymbol{\epsilon}_g$ is non-decreasing and left-continuous. On the other hand, note that

$$[h^{-1}]' = \frac{1}{h' \circ h^{-1}} = \frac{1}{\frac{1}{2}(g'+1) \circ h^{-1}} = \frac{2}{g' \circ h^{-1} + 1}.$$

Thus, we have

$$\begin{aligned}\boldsymbol{\epsilon}_g{}' &= \frac{1+\boldsymbol{\xi}}{2}\left(g'\circ h^{-1}\right)\frac{2}{g'\circ h^{-1}+1}+\frac{1-\boldsymbol{\xi}}{2}\frac{2}{g'\circ h^{-1}+1}\\ &= \left(\frac{1+\boldsymbol{\xi}}{2}g'\circ h^{-1}+\frac{1-\boldsymbol{\xi}}{2}\right)\frac{2}{g'\circ h^{-1}+1}\\ &= 1+\boldsymbol{\xi}-\boldsymbol{\xi}\frac{2}{g'\circ h^{-1}+1} = 1+\boldsymbol{\xi}\left(1-\frac{2}{g'\circ h^{-1}+1}\right).\end{aligned}$$

This implies

$$|\boldsymbol{\epsilon}_g{}'| \leqslant 1+|\boldsymbol{\xi}|\left|1-\frac{2}{g'\circ h^{-1}+1}\right| \leqslant 2.$$

The bound comes from $\boldsymbol{\xi}\sim U(-1,1)$ and $g'\geqslant 0$, which is hence tight. Thus any $\boldsymbol{\epsilon}_g$ generated by Formula (5.3.1) is Lipschitz continuous, with the constant uniformly bounded by 2 over $\boldsymbol{\xi}$. Note that Assumption A6 requires the Lipschitz continuity only in expectation. Thus, the i.i.d. samples $\boldsymbol{\epsilon}_{i,t}$ of any $\boldsymbol{\epsilon}_g$ satisfy obviously Assumption A6 with the largest $L=2$. Figure 5.1 shows the function g used in the simulation and one realization of 30 i.i.d. samples of the resulting $\boldsymbol{\epsilon}_g$.

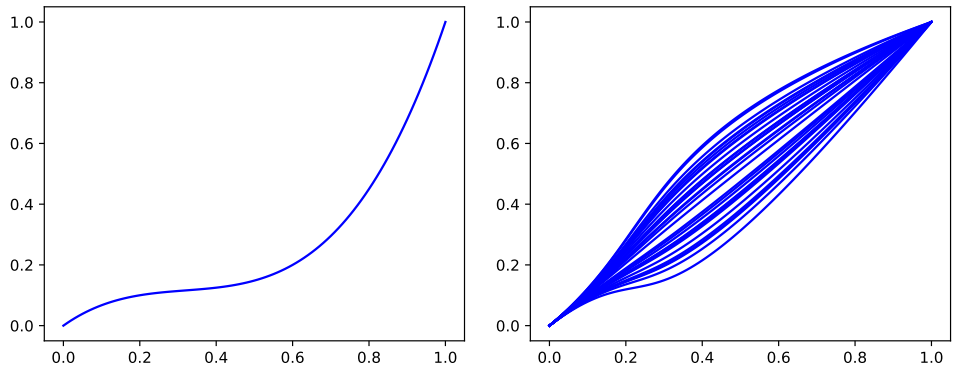

Figure 5.1: The function g on the left is given by the natural cubic spline passing through the points $(0,0),(0.2,0.1),(0.6,0.2),(1,1)$. On the right is one realization of 30 i.i.d. samples of the resulting $\boldsymbol{\epsilon}_g$.

Secondly, we present the procedure to generate the true coefficient matrix A. We first generate a sparse matrix with the weights all positive in a random way, denoted by A^0. We then normalize each row of A^0 by the row sum to fulfill Assumptions A5. We denote this last matrix still by A^0. Based on the previous mechanism for the random distortion function, we take $L=2$ in Assumption A7. Lastly, we scale down A^0 by $(2+\alpha)\|A^0\|_2$ to obtain a valid A. We test two values $\alpha=0.1$ and $\alpha=0.5$ in our experiments.

Given a valid matrix of coefficients A and samples of $\boldsymbol{\epsilon}_{i,t}$, we can then generate the "centered" quantile functions $\widetilde{\boldsymbol{F}}_{i,t}^{-1}$ from Model (5.1.5). Note that, $\widetilde{\boldsymbol{F}}_{i,t}^{-1}$ are only the simulations of the transformed data. Thus, we have to generate furthermore the population Fréchet mean $F_{i,\oplus}^{-1}$ of each univariate series in order to finally obtain the

synthesized "raw" data, as the inverse of transformation (5.1.2):

$$\boldsymbol{F}_{i,t}^{-1} = \widetilde{\boldsymbol{F}}_{i,t}^{-1} \oplus F_{i,\oplus}^{-1} := \widetilde{\boldsymbol{F}}_{i,t}^{-1} \circ F_{i,\oplus}^{-1}.$$

We set $F_{i,\oplus}^{-1}$ as the natural cubic spline of the points: $(0,0), (0.2,0.1), (0.6, 0.2 + 0.2i/N), (1,1)$, $i = 1, ..., N$. The empirical Fréchet mean $\boldsymbol{F}_{\bar{\mu}_i}^{-1}$ and the proposed estimator $\widehat{\boldsymbol{A}}$ are calculated on the synthesized "raw" data $\boldsymbol{F}_{i,t}^{-1}$. In Section 5.3.1.2, we aim to demonstrate the consistency result given in Theorem 5.2.5 with the synthetic data.

5.3.1.2 Experiment settings and results

In this experiment, we demonstrate the consistency of the proposed estimator $\widehat{\boldsymbol{A}}$ for two different values $N = 10$ and $N = 100$. For each N, we generate two true matrices A for $\alpha = 0.1$ and 0.5 respectively, according to the procedure presented in Section 5.3.1.1. With each A, we calculate the root mean square deviation (RMSD) successively

$$\frac{\|\widehat{\boldsymbol{A}} - A\|_F}{\|A\|_F}, \tag{5.3.2}$$

with the synthetic data that it generates along time. To furthermore study the mean and the variance of the RMSD (5.3.2), we run 100 independent simulations for the same A.

Note that the value of $\widehat{\boldsymbol{A}}$ we use in Equation (5.3.2) is the approximation obtained by the projected gradient descent applied to Problem (5.2.5). Thus the corresponding approximation error also accounts for the deviation which is on the order of the threshold we set in the stopping criteria. For all values of N, we use the same error threshold. We stop the algorithm as soon as the difference between the previous and the current updates in ℓ_2 norm reaches 0.0001, for the resolution of each row $\widehat{\boldsymbol{A}}_{i:}$.

We firstly show the evolution of RMSD for $N = 10, 100$ in Figures 5.2 and 5.3, respectively. We can see that all means and variances of the RMSD decrease towards zero as the sample size T increases, for each N and α value. This demonstrates empirically that, when the model assumptions A5, A6 and A7 hold true for the data, the proposed estimator $\widehat{\boldsymbol{A}}$ converges to A in probability, which is implied actually by the convergence of $\widehat{\boldsymbol{A}}$ to A in L^2.

Additionally, we can notice that, the RMSD for $\alpha = 0.1$ which corresponds to larger ℓ_2 norm of A has a smaller mean in both cases, and also a smaller variance for most of the sample sizes T investigated.

Also we would like to remark that, during the first few T values, the samples are insufficient for a meaningful estimation. Thus the projected gradient descent will terminate rapidly, as shown in the very beginning of Figures 5.4 and 5.5. The output $\widehat{\boldsymbol{A}}$ will be a zero matrix, since we initialize $\widehat{\boldsymbol{A}}$ as zero. This results in the low RMSD values during the early phase, since the true A is generated as a sparse matrix with small weights, as shown in Figures 5.2 and 5.3. As T increases, more entries of $\widehat{\boldsymbol{A}}$ become non-zero, which brings to the growth of RMSD. Upon the arrival of more new samples, $\widehat{\boldsymbol{A}}$ starts to converges, meanwhile the RMSD starts to decrease accordingly.

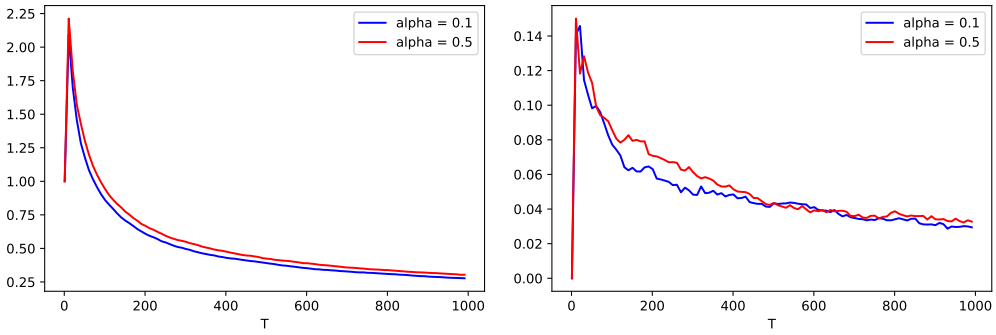

Figure 5.2: *Mean (left) and standard deviation (right) of RMSD for* $N = 10$. The mean and the variance are calculated over 100 simulations along time T, every 10 time instants.

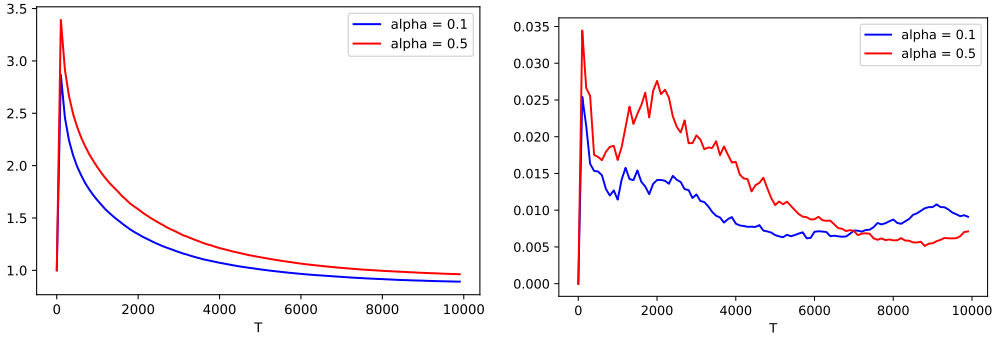

Figure 5.3: *Mean (left) and standard deviation (right) of RMSD for* $N = 100$. The mean and the variance are calculated every 100 time instants.

Lastly, we show in Figures 5.4 and 5.5 the complete execution time of the model fitting on the raw data with respect to the sample size T. We can see that the execution time increases linearly with respect to T, and A with the smaller ℓ_2 norm requires sightly less time ($\alpha = 0.5$) than the other. The linear increase comes mainly from the loop over time $t = 1, ..., T$ in calculating the empirical Fréchet mean (5.2.2) and in calculating the matrices $\widehat{\mathbf{\Gamma}}(0), \widehat{\mathbf{\Gamma}}(1)$ by their formulas in Equation (5.2.4). The running time of these calculations is determined by the granularity in the numerical methods to approximate the function composition, function inverse, and the inner product. The granularity applied during this simulation is 0.01, that is we input/output only the quantile function values at grid $0, 0.01, ..., 0.99$ to/from each numerical approximation.

5.3.2 Age distribution of countries

We firstly test the proposed model with the data of distributional observations illustrated in Figure 2.2 in the introduction. These data are from the US Census

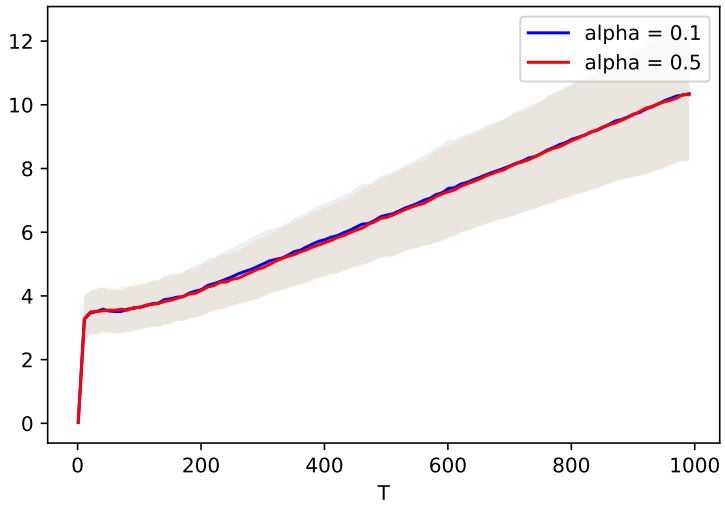

Figure 5.4: *Calculation time (in seconds) of* $\widehat{\boldsymbol{A}}$ *with respect to the sample size* T *for* $N = 10$. The calculation time counting starts from the computation of the empirical Fréchet means for Data transformation (5.2.3), and ends when the accelerated projected gradient descent of Problem (5.2.5) finishes for the last row $i = N$.

Bureau's International Data Base*, which provides the population estimates and projections for countries and areas by single year of age, over years. We would like to apply the proposed model on this international age distribution data to learn about the links among the changes in the age structures of different countries. Specially, we consider the countries and the micro-states in the European Union and/or Schengen Area. Because the corresponding data used during the model fitting starts in the 1990s, we also include the former European Union member United Kingdom. Note that, Vatican City is not included since it is not available in the data base. Therefore, the list of 34 countries in this study is: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Liechtenstein, Lithuania, Luxembourg, Malta, Monaco, Netherlands, Norway, Poland, Portugal, Romania, San Marino, Slovakia, Slovenia, Spain, Sweden, Switzerland, United Kingdom. Time-wise, we consider the 40 years between 1996 to 2035. Note that 1996 is the earliest year for which the data for all the considered countries is available.

To apply Model (5.1.4), we firstly represent the distribution of age population, of country i, at year T, by $\boldsymbol{\mu}_t^i$, with $T = 1, ..., 40$ and $i = 1, ..., 34$. Note that the age considered by the data base goes through 0 to 100-plus. Thus we take the 100-plus as 100, and moreover scale down the age by 100 to make the age distribution supported over $[0, 1]$. Then we retrieve the quantile function $\boldsymbol{F}_{i,t}^{-1}$ of $\boldsymbol{\mu}_t^i$ from the population counts by ages of country i recorded at year T, with the numeric methods. In particular, we retrieve the quantile functions using continuous functions which

*The data base in open access through https://www.census.gov/data/developers/data-sets/international-database.html.

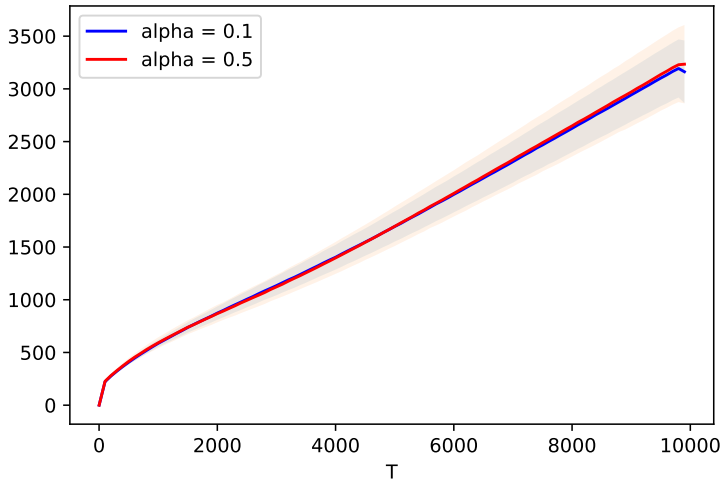

Figure 5.5: *Calculation time (in seconds) of* $\widehat{\boldsymbol{A}}$ *with respect to the sample size* T *for* $N = 100$.

take 0 on $p = 0$ and 1 on $p = 1$ so as to be consistent with Assumption A4 (for details see function `generate_qt_fun` defined in script *age_pop.py* in the code related to this paper).

We fit Model (5.1.5) on the retrieved functions $\boldsymbol{F}_{i,t}^{-1}$, $T = 1, ..., 40, i = 1, ..., 34$. We use the same stopping criteria as in the simulation, while we apply the granularity of 0.002. The complete execution time of model fitting takes around 78 seconds. Figure 2.4 in the introduction comes from the inferred coefficient A in this experiment. We recall this result in Figure 5.6 with a more technical caption and the interpretation with the data set.

Firstly, we can notice that for all countries $i \in \{1, ..., 34\}$, the weight of self-loop A_{ii} dominates the weights of incoming edges A_{ij}, $j = 1, ..., 34$, which are bounded by $0 \leqslant \sum_{j=1}^{34} A_{ij} \leqslant 1$. This is because the age structure of a country does not change much from one year to another. On the other hand, this also implies the age structure differs largely across countries. Nevertheless, there are still significant links between countries' age distribution. The first two largest weights excluding all the self-loops are respectively on the edges: Estonia → Latvia, and Sweden → Norway. To justify the inferred edges, we plot the evolution of age structure of these four countries in Figures 5.7 and 5.8.

We can see that within these four countries, the age structures between the linked countries are similar along time, by contrast, the structures between the unlinked countries are very different. Indeed, these two linked pairs consist both of the countries which share long distance of borders. Thus generally, the inferred edges in Figure 5.6 indicate the similarity of the age structures between countries from 1996 to 2036. Moreover, the directions of the edges imply, at the model level, that, when age structures in the outward countries (for example, Estonia, Sweden) change, it will induce relative changes in the inward countries (respectively, Latvia, Norway). These numeric findings can be furthermore explained in demography or not. On the other, we are interested in the neighbouring countries which are not linked. We verify for

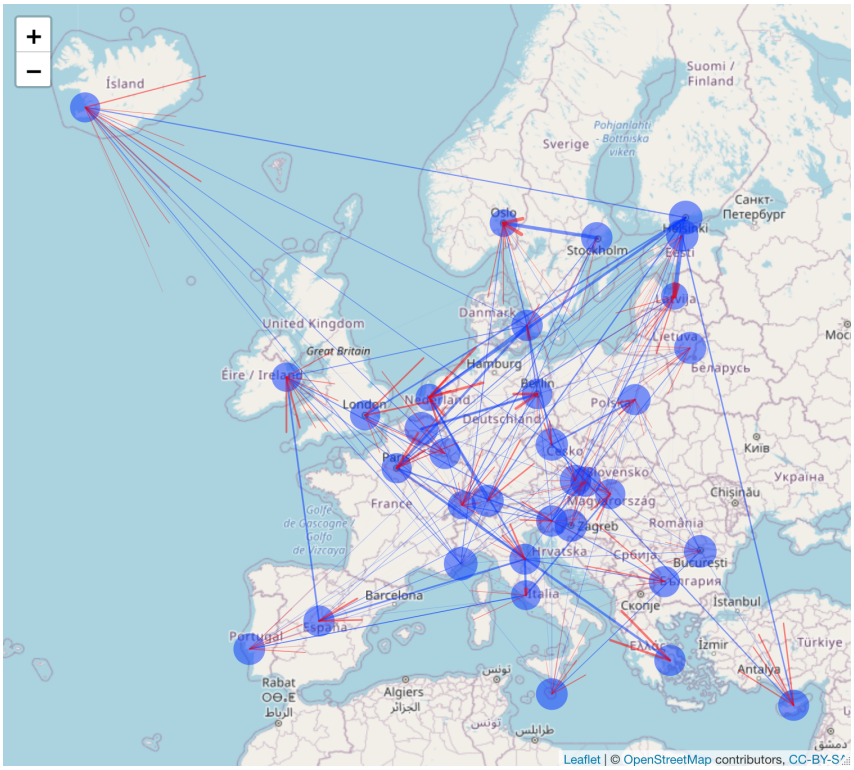

Figure 5.6: *Inferred age structure graph.* The non-zero coefficients A_{ij} are represented by the weighted directed edges from node j to node i. Thicker arrow corresponds to larger weights. The blue circles around nodes represent the weights of self-loop.

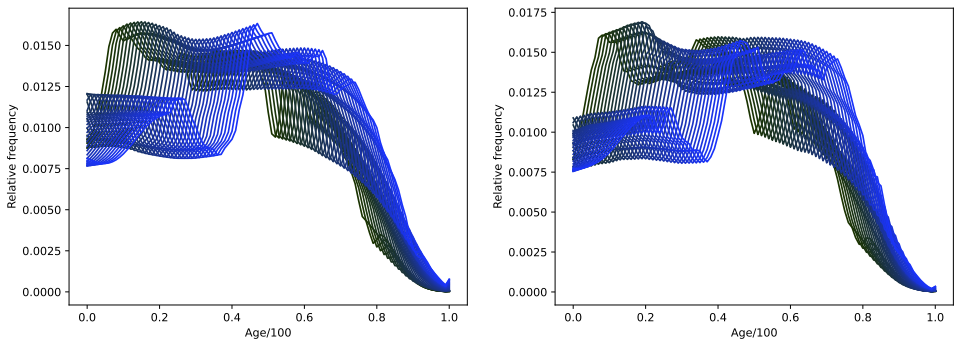

Figure 5.7: *Evolution of age structure from* 1996 *to* 2036 *(projected) of Estonia (left) versus Latvia (right).* Each curve connects the 101 relative frequencies from $0, 1/100, 2/100, ..., 1$, which represents the age structure of a considered year. Lighter curves correspond to more recent years.

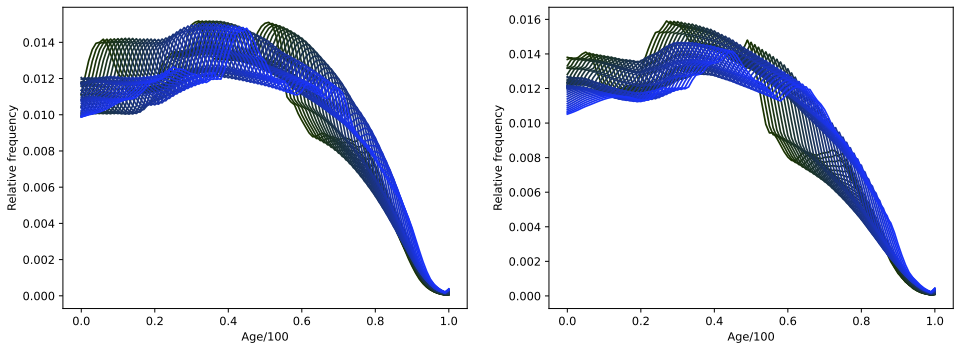

Figure 5.8: *Evolution of age structure from* 1996 *to* 2036 *(projected) of Sweden (left) versus Norway (right).*

example the age structures of France, Italy, in Figure 5.9. We can see that the age structures are as expected very different.

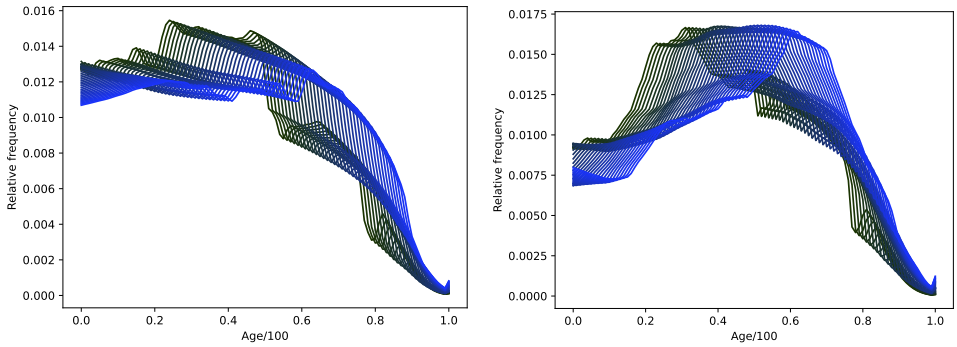

Figure 5.9: *Evolution of age structure from* 1996 *to* 2036 *(projected) of France (left) versus Italy (right).*

All these observations strongly support the usefulness of our model. Lastly, in Table 5.1, we provide the first 5 edges of the largest weights.

5.3.3 Bike-sharing network in Paris

Secondly, we test our model on the bike-sharing data set of Paris. The original data set is available at: https://maxhalford.github.io/blog/a-short-introduction-and-conclusion-to-the-openbikes-2016-challenge/. It contains the coordinates of all bike stations and their usage recordings. After data cleaning, we extract the variable *unused bike ratio* at the recording time for each station. To have the time series at the common time for all stations, we perform the linear interpolation with hourly observation frequency. The clean data set consists in the observations of unused bike ratio of 274 stations over 4417 consecutive hours. We are interested in the temporal evolution of the bike availability of stations, and would like to identify how the stations relate mutually in their evolution using the proposed method. To this end, we firstly represent the data by taking into account the distribution aspect. We

	From	To
1	Estonia	Latvia
2	Sweden	Norway
3	Belgium	Germany
4	Finland	Netherlands
5	France	Greece

Table 5.1: *Top* 5 *edges with the largest weights excluding all the self-loops*

introduce the temporal variable T which represents hours in a day. Accordingly, the distribution $\boldsymbol{\mu}_t^i$ considered by Model (5.1.4) represents the distribution of the bike availability of the station i, at hour T in a day, with $T = 1, ..., 24$ and $i = 1, ..., 274$. Thus the quantile function $\boldsymbol{F}_{i,t}^{-1}$ of distribution $\boldsymbol{\mu}_t^i$ can be retrieved from all the data points of station i recorded at hour T, by the numeric methods. We use the similar method as for the age distribution data to retrieve the valid quantile functions, however the implementation is different due to the availability of different quantity types (for details see function `generate_qt_fun` defined in script *bike_ net.py* in the code related to this paper).

Note that since these observations are distanced from each other in time, they can be considered approximately as being independent samples. For more comments on this point, we refer to Remark 5.3.1. We then fit Model (5.1.5) on data $\boldsymbol{F}_{i,t}^{-1}$, $T = 1, ..., 24, i = 1, ..., 274$. We use the same stopping criteria as previously, and we apply the granularity of 0.002. The complete execution time of model fitting takes around 20 minutes. We now demonstrate the visualization of the inferred matrix of coefficients A on the map of Paris, represented by the directed weighted graph. Since the edges of the complete graph will be densely located in the plot when fitting the graph to the paper size, for better visual effects, we show two subgraphs each of 50 nodes, in Figure 5.10 and Figure 5.11 respectively. The complete graph and the subgraphs are available in the interactive form at `https://github.com/yiyej/Wasserstein_Multivariate_Autoregressive_Model`.

These figures show some interesting links between the bike utilisation in different areas of Paris. For example, we can notice in Figure 5.10 that the regression relationship between the station *41604-lagny-saint-mande* and the stations along the flow from *Saint-Mandé* in southeast to *Neuilly-sur-Seine* in northwest are very significant. We recall the result of node predictability shown in Figure 2.6 and Chapter 6, which is also learned from this data set. This flow going through the main itinerary of Metro 1 and RER A in Paris corresponds to the first group of stations with high predictability. On the other hand, the second group is located at the bottom of Figure 5.10 around the train station Paris Montparnasse. We can see that, the result from this section also shows that the stations in this area have significant predictors, especially the bike stations around the other train station Gare du Nord, which is situated at the top of Figure 5.10. Note that, since Figure 5.10 shows only a subset of edges between the 50 randomly chosen nodes, the complete graph can have also re-found other patterns of the derived node ranks in Chapter 6.

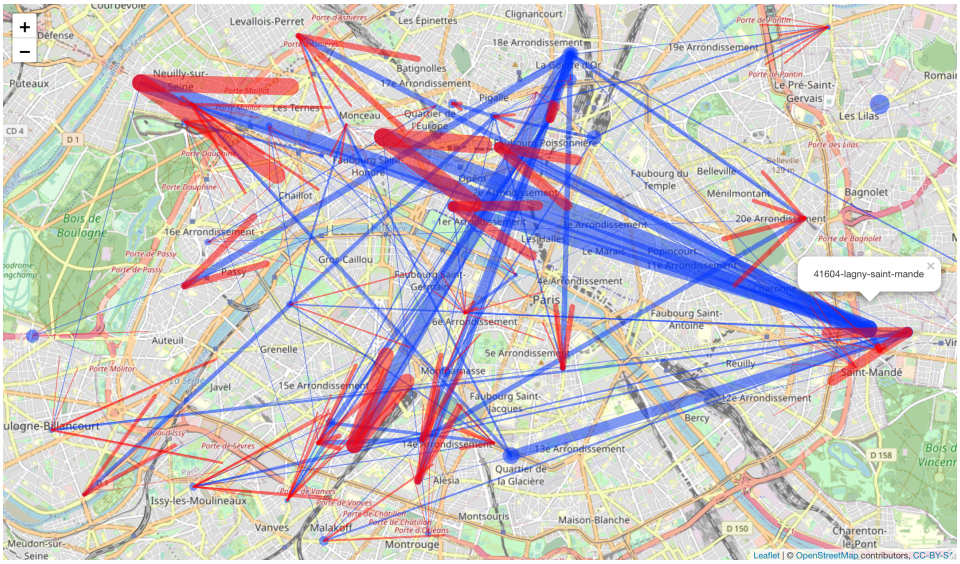

Figure 5.10: *Subgraph* 1 *(50 nodes chosen randomly).* For graphical meanings see the caption of Figure 5.6.

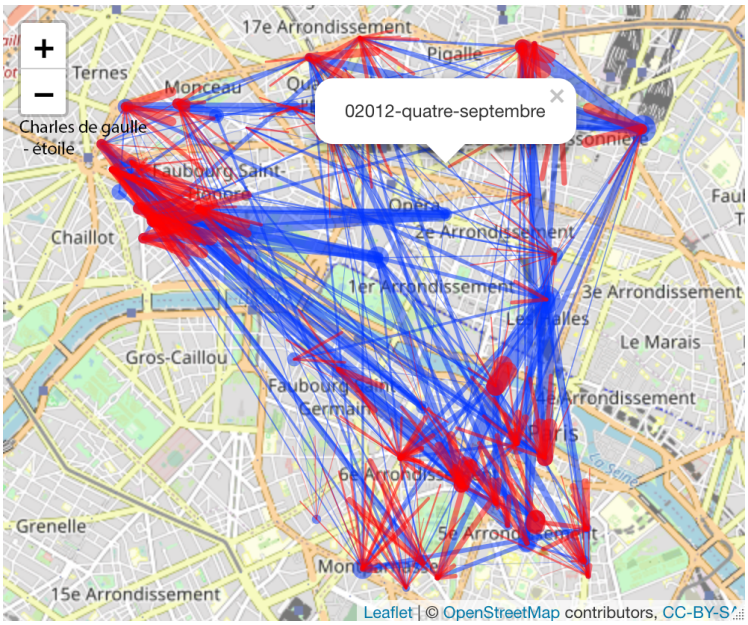

Figure 5.11: *Subgraph* 2 *(the first* 50 *nodes).* This subgraph exhibits a more specific area in Paris, isolating the stations numbered 1 to 50.

Additionally, we can see in Figure 5.11 that the stations near *Charles de gaulle - étoile* (upper left) need to be predicted by numerous stations together, among which is the station *02012-quatre-septembre*. By contrast, station *02012-quatre-septembre* can only be predicted by itself with no edges pointed in. These observations all support the effectiveness of the proposed AR model and its estimation method.

Remark 5.3.1. Note that another way to represent the data set by distributions is to consider every K consecutive observations as the i.i.d. samples of a distribution. Thus $T = 1, 2, ..., 185/K$ represents the ongoing time in hours. However, the consecutive observations are highly correlated. Moreover, the distribution retrieved in this manner does not clearly represent a random variable. Thus we do not go further with this data representation method.

5.4 Appendix

5.4.1 Proof of Theorem 5.1.2 and Theorem 5.1.3

With $(\mathcal{X}, d)$ given in Definition (5.1.7), $\mathbf{\Phi}_{\epsilon_t}$ defined in Equation (5.1.8), and Θ defined as the product space of N $(\Pi, \|\|_{Leb})$ with $(\Pi, \|\|_{Leb})$ defined in Equation (5.1.4), we show the resulting system (5.1.8) satisfies Conditions C1, C2 in Section 3.4.1.

The verification of Condition C1 does not require the additional assumptions. We consider any $X \in (\mathcal{X}, d)$, $\alpha = 2$, then

$$\mathbb{E}(d^2(X, \mathbf{\Phi}_{\epsilon_t}(X))) = \mathbb{E}\left(\sum_{i=1}^{N} \|X_i - \mathbf{\Phi}^i_{\epsilon_t}(X)\|^2_{Leb}\right).$$

Since $(\mathcal{X}, d)$ is a space of bounded functions from $(0, 1)$ to $[0, 1]$. Thus $\mathbb{E}(d^2(X, \mathbf{\Phi}_{\epsilon_1}(X)))$ is bounded.

We are now to show that under Assumptions A6 and A7, IRF System (5.1.8) satisfies Condition C2. We first examine the case of $m = 1$, we consider any $X, X^1 \in \mathcal{X}$, then

$$\begin{aligned}
&\mathbb{E}(d^\alpha(\widetilde{\mathbf{\Phi}}_{t,1}(X), \widetilde{\mathbf{\Phi}}_{t,1}(X^1)) \\
&= \mathbb{E}\left[\sum_{i=1}^{N} \left\| \boldsymbol{\epsilon}_{i,t} \circ \left[\sum_{j=1}^{N} A_{ij}\left(X_j - id\right) + id\right] - \boldsymbol{\epsilon}_{i,t} \circ \left[\sum_{j=1}^{N} A_{ij}\left(X^1_j - id\right) + id\right] \right\|^2\right] \\
&\overset{(b)}{=} \sum_{i=1}^{N} \int_0^1 \mathbb{E}\left(\boldsymbol{\epsilon}_{i,t} \circ \left[\sum_{j=1}^{N} A_{ij}\left(X_j - id\right) + id\right](p) - \boldsymbol{\epsilon}_{i,t} \circ \left[\sum_{j=1}^{N} A_{ij}\left(X^1_j - id\right) + id\right](p)\right)^2 dp \\
&\overset{Assumption A6}{\leqslant} \sum_{i=1}^{N} L^2 \int_0^1 \left(\left[\sum_{j=1}^{N} A_{ij}\left(X_j - id\right) + id\right](p) - \left[\sum_{j=1}^{N} A_{ij}\left(X^1_j - id\right) + id\right](p)\right)^2 dp \\
&= \sum_{i=1}^{N} L^2 \int_0^1 \left(\sum_{j=1}^{N} A_{ij}(X_j(p) - X^1_j(p))\right)^2 dp = L^2 \int_0^1 \sum_{i=1}^{N} \left(\sum_{j=1}^{N} A_{ij}(X_j(p) - X^1_j(p))\right)^2 dp
\end{aligned}$$

Exchange (b) of the integral and the expectation comes from that the integrand is bounded. Then for any fixed $p \in (0,1)$, $\sum_{i=1}^{N}\left(\sum_{j=1}^{N} A_{ij}(X_j(p) - X_j^1(p))\right)^2 = \|A\vec{x}_p\|_{\ell_2}^2$, where $\vec{x}_p = (X_1(p) - X_1^1(p), \ldots, X_N(p) - X_N^1(p))^\top$. Since $\|A\vec{x}_p\|_{\ell_2}^2 \leqslant \|A\|_2^2\|\vec{x}_p\|_{\ell_2}^2$, we have

$$
\begin{aligned}
&L^2\int_0^1 \sum_{i=1}^{N}\left(\sum_{j=1}^{N} A_{ij}(X_j(p) - X_j^1(p))\right)^2 dp \\
&\leqslant L^2\|A\|_2^2\int_0^1 \sum_{j=1}^{N}\left((X_j(p) - X_j^1(p))\right)^2 dp = L^2\|A\|_2^2 d^2(X, X^1).
\end{aligned}
$$

By Assumption A7, $L^2\|A\|_2^2 < 1$, thus by taking $r = L^2\|A\|_2^2$, $C = 1$, Equation (3.4.2) is checked for $m = 1$. Now, suppose for any fixed m, we have

$$
\mathbb{E}(d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m}(X), \widetilde{\mathbf{\Phi}}_{t,m}(X^1)) \leqslant r^m d^2(X, X^1). \tag{5.4.1}
$$

Then for $m+1$, we have

$$
\begin{aligned}
&\mathbb{E}(d^\alpha(\widetilde{\mathbf{\Phi}}_{t,m+1}(X), \widetilde{\mathbf{\Phi}}_{t,m+1}(X^1)) \\
&= \mathbb{E}\left[\sum_{i=1}^{N}\left\|\boldsymbol{\epsilon}_{i,t}\circ\left[\sum_{j=1}^{N} A_{ij}\left(\widetilde{\mathbf{\Phi}}_{t-1,m}^j(X) - id\right) + id\right] - \boldsymbol{\epsilon}_{i,t}\circ\left[\sum_{j=1}^{N} A_{ij}\left(\widetilde{\mathbf{\Phi}}_{t-1,m}^j(X^1) - id\right) + id\right]\right\|^2\right] \\
&= \sum_{i=1}^{N}\int_0^1 \mathbb{E}\Bigg(\boldsymbol{\epsilon}_{i,t}\circ\left[\sum_{j=1}^{N} A_{ij}\left(\widetilde{\mathbf{\Phi}}_{t-1,m}^j(X) - id\right) + id\right](p) \\
&\qquad\qquad - \boldsymbol{\epsilon}_{i,t}\circ\left[\sum_{j=1}^{N} A_{ij}\left(\widetilde{\mathbf{\Phi}}_{t-1,m}^j(X^1) - id\right) + id\right](p)\Bigg)^2 dp \\
&= \sum_{i=1}^{N}\int_0^1 \mathbb{E}[\mathbb{E}(\boldsymbol{\epsilon}_{i,t}\circ\left[\sum_{j=1}^{N} A_{ij}\left(\widetilde{\mathbf{\Phi}}_{t-1,m}^j(X) - id\right) + id\right](p) \\
&\qquad\qquad - \boldsymbol{\epsilon}_{i,t}\circ\left[\sum_{j=1}^{N} A_{ij}\left(\widetilde{\mathbf{\Phi}}_{t-1,m}^j(X^1) - id\right) + id\right](p))^2\Big|\boldsymbol{\epsilon}_\tau, \tau < t]dp \\
&\overset{Assumption A6}{\leqslant} \sum_{i=1}^{N} L^2\int_0^1 \mathbb{E}\Bigg(\left[\sum_{j=1}^{N} A_{ij}\left(\widetilde{\mathbf{\Phi}}_{t-1,m}^j(X) - id\right) + id\right](p) \\
&\qquad\qquad - \left[\sum_{j=1}^{N} A_{ij}\left(\widetilde{\mathbf{\Phi}}_{t-1,m}^j(X^1) - id\right) + id\right](p)\Bigg)^2 dp \\
&= L^2\int_0^1 \mathbb{E}\sum_{i=1}^{N}\left(\sum_{j=1}^{N} A_{ij}(\widetilde{\mathbf{\Phi}}_{t-1,m}^j(X)(p) - \widetilde{\mathbf{\Phi}}_{t-1,m}^j(X^1)(p))\right)^2 dp \\
&\leqslant L^2\|A\|_2^2\mathbb{E}(d^2(\widetilde{\mathbf{\Phi}}_{t-1,m}(X), \widetilde{\mathbf{\Phi}}_{t-1,m}(X^1))) \overset{(c)}{\leqslant} r^{m+1}d^2(X, X^1).
\end{aligned}
$$

Inequality (c) comes from that $\widetilde{\mathbf{\Phi}}_{t,m}(X) \overset{d}{=} \widetilde{\mathbf{\Phi}}_{t-1,m}(X), \forall t \in \mathbb{Z}$ and the hypothesis (5.4.1). Thus, by induction, Equation (3.4.2) is verified for all $m \geqslant 1$, thus Condition C2 is checked by IRF System (5.1.8).

By Theorems 3.4.4, 3.4.5, we have proved Theorem 5.1.2. Note that since $\mathcal{X}$ is the set of quantile functions from $(0,1)$ to $[0,1]$, all the elements in $\mathcal{X}$ are bounded uniformly in the sense of Inequality (3.4.3) with d defined as (5.1.7). Thus Theorem 3.4.5 applies to all solution $\mathbf{S}_t$ in (X,d). Moreover, since α is taken as 2, by Theorem 3.4.8, we obtain furthermore the stationarity of the unique solution $\mathbf{X}_t$ in the underlying Hilbert space $(X,\langle,\rangle)$ with $\langle X,Y\rangle=\sum_i^N\langle X_i,Y_i\rangle_{Leb}$.

Lastly, applying Proposition 3.4.6, we have additionally that IRF system (5.1.8) is geometric moment contracting.

■

5.4.2 Proof of Proposition 5.1.4

Since $\mathbf{X}_t$ is a solution of IRF system (5.1.8) defined in $\mathcal{T}^N,d$, where $\mathcal{T}$ is a set of quantile functions over $[0,1]$. Thus point 1 is checked.

For any $p\in(0,1)$, from the definition of system (5.1.8), we have

$$\mathbf{X}_{i,t}(p)=\boldsymbol{\epsilon}_{i,t}\circ\left[\sum_{j=1}^{N}A_{ij}\left(\mathbf{X}_{j,t-1}-id\right)+id\right](p),\quad t\in\mathbb{Z},\ i=1,\dots,N.$$

Take expectation on both sides, gives

$$\begin{aligned}\mathbb{E}\mathbf{X}_{i,t}(p)&=\mathbb{E}\left[\boldsymbol{\epsilon}_{i,t}\circ\left[\sum_{j=1}^{N}A_{ij}\left(\mathbf{X}_{j,t-1}-id\right)+id\right](p)\right]\\&\overset{(a)}{=}\mathbb{E}\left[\sum_{j=1}^{N}A_{ij}\left(\mathbf{X}_{j,t-1}-id\right)+id\right](p)\\&=\sum_{j=1}^{N}A_{ij}\left(\mathbb{E}\mathbf{X}_{j,t-1}(p)-p\right)+p.\end{aligned}$$

Equation (a) comes from that $\boldsymbol{\epsilon}_{i,t}$ is independent of $\mathbf{X}_{t-1}$. Thus we have $\mathbb{E}\mathbf{X}_{i,t}(p)-p=\sum_{j=1}^{N}A_{ij}\left(\mathbb{E}\mathbf{X}_{j,t-1}(p)-p\right),\ i=1,\dots,N,p\in(0,1)$. When $\mathbf{X}_t,t\in\mathbb{Z}$ is a stationary solution, the equation becomes $\mathbb{E}\mathbf{X}_{i,t}(p)-p=\sum_{j=1}^{N}A_{ij}\left(\mathbb{E}\mathbf{X}_{j,t}(p)-p\right),\ i=1,\dots,N,p\in(0,1)$. Let $\vec{x}_p=\left(\mathbb{E}\mathbf{X}_{1,t}(p)-p,\dots,\mathbb{E}\mathbf{X}_{N,t}(p)-p\right)^\top$, we then have

$$\vec{x}_p=A\vec{x}_p\iff(I-A)\vec{x}_p=0.$$

Since $\|A\|_2<1$, $I-A$ is invertible, which implies $\vec{x}_p=0$ for any $p\in(0,1)$. Thus $\mathbb{E}\mathbf{X}_{1,t}(p)=p,\ i=1,\dots,N,p\in(0,1)$.

■

5.4.3 Proof of Proposition 5.1.5

Process (5.1.8) writes in terms of components $\mathbf{X}_{i,t}$ as

$$\mathbf{X}_{i,t}=\boldsymbol{\epsilon}_{i,t}\circ\left[\sum_{j=1}^{N}A_{ij}\left(\mathbf{X}_{j,t-1}-id\right)+id\right].$$

Subtract id from the both sides, we obtain

$$\mathbf{X}_{i,t} - id = \boldsymbol{\epsilon}_{i,t} \circ \left[\sum_{j=1}^{N} A_{ij} \left(\mathbf{X}_{j,t-1} - id \right) + id \right] - id.$$

Pass the both sides to the inner product with $\mathbf{X}_{l,t-1} - id$, then take the expectation, we get

$$\begin{aligned}
\mathbb{E}\langle \mathbf{X}_{i,t} - id, \mathbf{X}_{l,t-1} - id \rangle &= \mathbb{E} \left\langle \boldsymbol{\epsilon}_{i,t} \circ \left[\sum_{j=1}^{N} A_{ij} \left(\mathbf{X}_{j,t-1} - id \right) + id \right] - id \, , \mathbf{X}_{l,t-1} - id \right\rangle \\
&= \mathbb{E} \left(\mathbb{E} \left[\left\langle \boldsymbol{\epsilon}_{i,t} \circ \left[\sum_{j=1}^{N} A_{ij} \left(\mathbf{X}_{j,t-1} - id \right) + id \right] - id \, , \mathbf{X}_{l,t-1} - id \right\rangle \middle| \, \mathbf{X}_{\tau}, \tau \leqslant t-1 \right] \right) \\
&= \mathbb{E} \left(\left\langle \sum_{j=1}^{N} A_{ij} \left(\mathbf{X}_{j,t-1} - id \right) \, , \mathbf{X}_{l,t-1} - id \right\rangle \right) = \sum_{j=1}^{N} A_{ij} \mathbb{E} \langle \mathbf{X}_{j,t-1} - id \, , \mathbf{X}_{l,t-1} - id \rangle .
\end{aligned}$$

Compare the definitions of $\Gamma(0), \Gamma(1)$, we can then retrieve Representation (5.1.9). ■

5.4.4 Proof of Lemma 5.2.1

Since $\widetilde{\boldsymbol{F}}_0^{-1}$ follows the stationary distribution π defined in Theorem 3.4.4, the sequence $\widetilde{\boldsymbol{F}}_t^{-1}$, $t = 1, \dots T$ generated successively by the stationary model (5.1.8) follow π, furthermore, their auto-covariance is time-invariant and equal to the stationary solution $\mathbf{X}_t$, $t \in \mathbb{Z}$. Thus we can use the data $\widetilde{\boldsymbol{F}}_t^{-1}$ in Representation 5.1.5, which writes as

$$A = \Gamma(0) \left[\Gamma(1) \right]^{-1},$$

where $[\Gamma(0)]_{j,l} = \mathbb{E}\langle \widetilde{\boldsymbol{F}}_{j,t-1} - id \, , \widetilde{\boldsymbol{F}}_{l,t-1} - id \rangle_{Leb}$ and $[\Gamma(1)]_{j,l} = \mathbb{E}\langle \widetilde{\boldsymbol{F}}_{j,t} - id \, , \widetilde{\boldsymbol{F}}_{l,t-1} - id \rangle_{Leb}$. Since $\widetilde{\boldsymbol{A}} = \widetilde{\boldsymbol{\Gamma}}(0) \left[\widetilde{\boldsymbol{\Gamma}}(1) \right]^{-1}$. We first show $[\widetilde{\boldsymbol{\Gamma}}(0)]_{j,l} - [\Gamma(0)]_{j,l} = \mathcal{O}(\frac{1}{\sqrt{T}})$ by applying Theorem 3 in Wu and Shao (2004).

To this end, we define $g_{jl} : (\mathcal{X}, \langle \cdot, \cdot \rangle) \to \mathbb{R}$ as

$$g_{jl}(\mathbf{X}) = \langle \mathbf{X}_j - id \, , \, \mathbf{X}_l - id \rangle_{Leb} - \mathbb{E}\langle \mathbf{X}_j - id \, , \, \mathbf{X}_l - id \rangle_{Leb}.$$

The construction implies that for any random object $\mathbf{Y} = (Y_i)_{i=1}^N$ in $(\mathcal{X}, \langle \cdot, \cdot \rangle)$, $\mathbb{E} g_{jl}(\mathbf{Y}) = \mathbf{0}$. Moreover, since all Y_i are bounded function from $(0,1)$ to $[0,1]$, $|g_{jl}(\mathbf{Y})|^p < \infty$ for all $p > 2$, which leads to $\mathbb{E}|g_{jl}(\mathbf{Y})|^p < \infty$ for all $p > 2$. It is then left to show that g_{jl} is stochastic dini-continuous (Wu and Shao, 2004, Equation (9)).

For any $\mathbf{Y}$ and $\mathbf{Y}^1$ identically distributed in $(\mathcal{X}, \langle \cdot, \cdot \rangle)$, we have

$$\begin{aligned}
|g_{jl}(\mathbf{Y}) - g_{jl}(\mathbf{Y}^1)| &= |\langle \mathbf{Y}_j - id \, , \, \mathbf{Y}_l - id \rangle_{Leb} - \langle \mathbf{Y}_j^1 - id \, , \, \mathbf{Y}_l^1 - id \rangle_{Leb}| \\
&\leqslant |\langle \mathbf{Y}_j - id \, , \, \mathbf{Y}_l - id \rangle_{Leb} - \langle \mathbf{Y}_j - id \, , \, \mathbf{Y}_l^1 - id \rangle_{Leb}| \\
&\qquad + |\langle \mathbf{Y}_j - id \, , \, \mathbf{Y}_l^1 - id \rangle_{Leb} - \langle \mathbf{Y}_j^1 - id \, , \, \mathbf{Y}_l^1 - id \rangle_{Leb}| \\
&\leqslant |\langle \mathbf{Y}_j - id \, , \, \mathbf{Y}_l - \mathbf{Y}_l^1 \rangle_{Leb}| + |\langle \mathbf{Y}_j - \mathbf{Y}_j^1 \, , \, \mathbf{Y}_l^1 - id \rangle_{Leb}| \\
&\leqslant \|\mathbf{Y}_j - id\| \|\mathbf{Y}_l - \mathbf{Y}_l^1\| + \|\mathbf{Y}_j - \mathbf{Y}_j^1\| \|\mathbf{Y}_l^1 - id\|.
\end{aligned}$$

Since $\mathbf{Y}_j$ and $\mathbf{Y}_l^1$ are increasing function from $(0,1)$ to $[0,1]$, $\|\mathbf{Y}_j - id\| \leqslant \frac{1}{2}$ and $\|\mathbf{Y}_l^1 - id\| \leqslant \frac{1}{2}$. Thus, we have furthermore

$$\begin{aligned}|g_{jl}(\mathbf{Y}) - g_{jl}(\mathbf{Y}^1)| &\leqslant \frac{1}{2}\left(\|\mathbf{Y}_l - \mathbf{Y}_l^1\| + \|\mathbf{Y}_j - \mathbf{Y}_j^1\|\right) \\ &\leqslant \frac{\sqrt{2}}{2}\left(\|\mathbf{Y}_l - \mathbf{Y}_l^1\|^2 + \|\mathbf{Y}_j - \mathbf{Y}_j^1\|^2\right)^{\frac{1}{2}}.\end{aligned}$$

Then

$$\sup_{\mathbf{Y},\mathbf{Y}_1}\left\{|g_{jl}(\mathbf{Y}) - g_{jl}(\mathbf{Y}^1)|\mathbf{1}_{\left(\sqrt{\sum_{i=1}^N \|\mathbf{Y}_i - \mathbf{Y}_i^1\|^2} < \delta\right)}\right\} \leqslant \frac{\sqrt{2}}{2}\delta.$$

Thus g_{jl} is stochastic dini-continuous. Since IRF system (5.1.8) is geometric moment contracting indicated in the proof of Theorem (5.1.2), then by Theorem 3 in Wu and Shao (2004),

$$\frac{S_T(g_{jl})}{\sqrt{T}} := \sqrt{T}\left(\left[\widetilde{\mathbf{\Gamma}}(0)\right]_{j,l} - [\Gamma(0)]_{j,l}\right) \xrightarrow{d} \sigma_{g_{jl}}\mathcal{N}(0,1),$$

which is followed

$$\left[\widetilde{\mathbf{\Gamma}}(0)\right]_{j,l} - [\Gamma(0)]_{j,l} = \mathcal{O}_p(\frac{1}{\sqrt{T}}).$$

The proof above holds valid for any $j,l = 1,\dots N$. Analogously, for any $j,l = 1,\dots N$, we define $g_{jl}^1 : (\mathcal{X}, \langle\cdot,\cdot\rangle) \times (\mathcal{X}, \langle\cdot,\cdot\rangle) \to \mathbb{R}$ as

$$g_{jl}(\mathbf{X},\mathbf{X}') = \langle \mathbf{X}_j - id\,,\, \mathbf{X}_l' - id\rangle_{Leb} - \mathbb{E}\langle \mathbf{X}_j - id\,,\, \mathbf{X}_l' - id\rangle_{Leb}.$$

It can be shown by the similar proof as before that $g_{jl}(\mathbf{X},\mathbf{X}')$ is stochastic dini-continuous with respect to the product metric

$$\rho((\mathbf{X},\mathbf{X}'),(\mathbb{Z},\mathbb{Z}')) = \sqrt{\sum_{i=1}^N \|\mathbf{X}_i - \mathbb{Z}_i\|^2 + \sum_{i=1}^N \|\mathbf{X}_i' - \mathbb{Z}_i'\|^2}.$$

Therefore by Theorem 3 in Wu and Shao (2004), we have

$$\left[\widetilde{\mathbf{\Gamma}}(1)\right]_{j,l} - [\Gamma(1)]_{j,l} = \mathcal{O}_p(\frac{1}{\sqrt{T}}), \quad j,l = 1,\dots N.$$

Since matrix inversion is a continuous mapping, thus by continuous mapping theorem

$$\left[\widetilde{\mathbf{\Gamma}}(0)^{-1}\right]_{j,l} - \left[\Gamma(0)^{-1}\right]_{j,l} = \mathcal{O}_p(\frac{1}{\sqrt{T}}), \quad j,l = 1,\dots N.$$

Representations of $\widetilde{\boldsymbol{A}}$ and A then bring to $\widetilde{\boldsymbol{A}}_{j,l} - A_{j,l} = \mathcal{O}_p(\frac{1}{\sqrt{T}})$, $j,l = 1,\dots N$.

■

5.4.5 Proof of Lemma 5.2.2

To prove Equation (5.2.6), we define, for any fixed $p \in (0,1)$, $i = 1, \dots N$, $g_i^p : (\mathcal{X}, \langle \cdot, \cdot \rangle) \to \mathbb{R}$ defined as

$$g_i^p(\mathbf{X}) = \mathbf{X}_i(p) - \mathbb{E}\,\mathbf{X}_i(p).$$

Follow the similar steps in the proof of Lemma 5.2.1, we have

$$\frac{S_T(g_i^p)}{T} := \frac{1}{T}\sum_{t=1}^{T} \widetilde{\boldsymbol{F}}_{i,t-1}^{-1}(p) - \mathbb{E}\,\widetilde{\boldsymbol{F}}_{i,t}^{-1}(p) \overset{Equation 5.1.3}{=} \frac{1}{T}\sum_{t=1}^{T} \widetilde{\boldsymbol{F}}_{i,t-1}^{-1}(p) - p = \mathcal{O}(\frac{1}{\sqrt{T}}).$$

Then plug the transformation relation in $\widetilde{\boldsymbol{F}}_{i,t-1}$, we have

$$\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{F}_{i,t}^{-1}[(F_{i,\oplus}^{-1})^{-1}(p)] - p = \mathcal{O}(\frac{1}{\sqrt{T}}),$$

where $(F_{i,\oplus}^{-1})^{-1}$ is the left continuous inverse of the quantile function $F_{i,\oplus}^{-1}$. By the statement under Assumption A3, $F_{i,\oplus}^{-1}$ is continuous, thus $(F_{i,\oplus}^{-1})^{-1}$ is strictly increasing. We therefore substitute q for $(F_{i,\oplus}^{-1})^{-1}(p)$, it follows $p = F_{i,\oplus}^{-1}(q)$, which brings to Equation (5.2.7).

■

5.4.6 Proof of Theorem 5.2.3

We first show a convergence result in Lemma 5.4.1 which will be required by the demonstration of the theorem.

Lemma 5.4.1. *Under the conditions of Lemma 5.2.1, we have*

$$\frac{1}{T}\sum_{t=1}^{T} \langle id, \widetilde{\boldsymbol{F}}_{j,t-1}^{-1} - id \rangle = \mathcal{O}(\frac{1}{\sqrt{T}}), \quad j = 1, \dots N.$$

The proof of Lemma 5.2.2 follows the same steps in the one of Lemma 5.2.1, by considering $g_j : (\mathcal{X}, \langle \cdot, \cdot \rangle) \to \mathbb{R}$ defined as

$$g_j(\mathbf{X}) = \langle id, \mathbf{X}_j - id \rangle_{Leb} - \mathbb{E}\langle id, \mathbf{X}_j - id \rangle_{Leb}.$$

Then use $\mathbb{E}\,\widetilde{\boldsymbol{F}}_{i,t}(p) = p, \ \forall p \in (0,1)$, we obtain the result.

■

We are now to prove $\widehat{\boldsymbol{A}}_o - \widetilde{\boldsymbol{A}} \xrightarrow{p} 0$. Note that analogous to A and $\widetilde{\boldsymbol{A}}$,

$$\widehat{\boldsymbol{A}}_o = \widehat{\boldsymbol{\Gamma}}(0)\left[\widehat{\boldsymbol{\Gamma}}(1)\right]^{-1},$$

$\widehat{\mathbf{\Gamma}}(0), \widehat{\mathbf{\Gamma}}(1)$ are given in Equation (5.2.4). Thus, we first show $[\widehat{\mathbf{\Gamma}}(0)]_{j,l} - [\widetilde{\mathbf{\Gamma}}(0)]_{j,l} \xrightarrow{p} 0$. By calculation, we have

$$
\begin{aligned}
[\widehat{\mathbf{\Gamma}}(0)]_{j,l} &= \frac{1}{T}\sum_{t=1}^{T}\langle \widehat{\boldsymbol{F}}_{j,t-1}^{-1} - id\,,\, \widehat{\boldsymbol{F}}_{l,t-1}^{-1} - id\rangle_{Leb} \overset{(a)}{=} \frac{1}{T}\sum_{t=1}^{T}\langle \widehat{\boldsymbol{F}}_{j,t}^{-1} - id\,,\, \widehat{\boldsymbol{F}}_{l,t}^{-1} - id\rangle_{Leb} + \mathcal{O}(\frac{1}{T}) \\
&= \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1} - id\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1} - id\rangle_{Leb} + \mathcal{O}(\frac{1}{T}) \\
&= \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1} - id\rangle_{Leb} - \frac{1}{T}\sum_{t=1}^{T}\langle id\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1} - id\rangle_{Leb} + \mathcal{O}(\frac{1}{T}) \\
&= \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1} - id\rangle_{Leb} + \mathcal{O}(\frac{1}{T}) \\
&= \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1}\rangle_{Leb} - \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1}\,,\, id\rangle_{Leb} + \mathcal{O}(\frac{1}{T}) \\
&= \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1}\rangle_{Leb} - \langle id\,,\, id\rangle_{Leb} + \mathcal{O}(\frac{1}{T}),
\end{aligned}
\tag{5.4.2}
$$

and

$$
\begin{aligned}
[\widetilde{\mathbf{\Gamma}}(0)]_{j,l} &= \frac{1}{T}\sum_{t=1}^{T}\langle \widetilde{\boldsymbol{F}}_{j,t-1}^{-1} - id\,,\, \widetilde{\boldsymbol{F}}_{l,t-1}^{-1} - id\rangle_{Leb} \overset{(b)}{=} \frac{1}{T}\sum_{t=1}^{T}\langle \widetilde{\boldsymbol{F}}_{j,t}^{-1} - id\,,\, \widetilde{\boldsymbol{F}}_{l,t}^{-1} - id\rangle_{Leb} + \mathcal{O}(\frac{1}{T}) \\
&= \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1} - id\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1} - id\rangle_{Leb} + \mathcal{O}(\frac{1}{T}) \\
&= \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{l,\oplus}^{-1} - id\rangle_{Leb} - \frac{1}{T}\sum_{t=1}^{T}\langle id\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{l,\oplus}^{-1} - id\rangle_{Leb} + \mathcal{O}(\frac{1}{T}) \\
&\overset{Lemma 5.4.1}{=} \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{l,\oplus}^{-1} - id\rangle_{Leb} + \mathcal{O}_p(\frac{1}{\sqrt{T}}) \\
&= \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{l,\oplus}^{-1}\rangle_{Leb} - \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, id\rangle_{Leb} + \mathcal{O}_p(\frac{1}{\sqrt{T}}) \\
&\overset{Lemma 5.4.1}{=} \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{l,\oplus}^{-1}\rangle_{Leb} - \langle id\,,\, id\rangle_{Leb} + \mathcal{O}_p(\frac{1}{\sqrt{T}}).
\end{aligned}
$$

Equations (a) and (b) come from $\widehat{\boldsymbol{F}}_{i,t}^{-1}$ and $\widetilde{\boldsymbol{F}}_{i,t-1}^{-1}$ are bounded between 0 and 1 for

all $i = 1, ..., N$ and $t \in \mathbb{N}$. Thus,

$$
\begin{aligned}
[\widehat{\mathbf{\Gamma}}(0)]_{j,l} - [\widetilde{\mathbf{\Gamma}}(0)]_{j,l} &= \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1}\rangle_{Leb} \\
&\quad - \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{l,\oplus}^{-1}\rangle_{Leb} + \mathcal{O}_p(\frac{1}{\sqrt{T}}) \\
&= \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1}\rangle_{Leb} - \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1}\rangle_{Leb} \\
&\quad + \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1}\rangle_{Leb} - \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{l,\oplus}^{-1}\rangle_{Leb} + \mathcal{O}_p(\frac{1}{\sqrt{T}}) \\
&= \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1} - \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1}\rangle_{Leb} \\
&\quad + \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1} - \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{l,\oplus}^{-1}\rangle_{Leb} + \mathcal{O}_p(\frac{1}{\sqrt{T}}).
\end{aligned}
\tag{5.4.3}
$$

We are now to show $\frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1} - \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1}\rangle_{Leb} \xrightarrow{p} 0$. We bound the absolute value

$$
\begin{aligned}
&\left|\frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1} - \boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1}\,,\, \boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1}\rangle_{Leb}\right| \\
&\leqslant \frac{1}{T}\sum_{t=1}^{T}\int_0^1 \left|(\boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1})(p) - (\boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1})(p)\right|\left|(\boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1})(p)\right| dp \\
&\overset{(a)}{\leqslant} \frac{1}{T}\sum_{t=1}^{T}\int_0^1 \left|(\boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1})(p) - (\boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1})(p)\right| dp.
\end{aligned}
$$

Inequality (a) is because $\boldsymbol{F}_{l,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_l}^{-1}$ is bounded between 0 and 1. Note that for any fixed $p \in (0,1)$, we have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\left|(\boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{\bar{\mu}_j}^{-1})(p) - (\boldsymbol{F}_{j,t}^{-1} \ominus \boldsymbol{F}_{j,\oplus}^{-1})(p)\right| = \frac{1}{T}\sum_{t=1}^{T}\left|\boldsymbol{F}_{j,t}^{-1} \circ (\boldsymbol{F}_{\bar{\mu}_j}^{-1})^{-1}(p) - \boldsymbol{F}_{j,t}^{-1} \circ (\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p)\right| \\
&= \begin{cases} \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{F}_{j,t}^{-1} \circ (\boldsymbol{F}_{\bar{\mu}_j}^{-1})^{-1}(p) - \boldsymbol{F}_{j,t}^{-1} \circ (\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p), & \text{if } (\boldsymbol{F}_{\bar{\mu}_j}^{-1})^{-1}(p) > (\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p), \\ \frac{1}{T}\sum_{t=1}^{T} -\boldsymbol{F}_{j,t}^{-1} \circ (\boldsymbol{F}_{\bar{\mu}_j}^{-1})^{-1}(p) + \boldsymbol{F}_{j,t}^{-1} \circ (\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p), & \text{otherwise,} \end{cases} \\
&= \begin{cases} p - (\boldsymbol{F}_{\bar{\mu}_j}^{-1}) \circ (\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p), & \text{if } (\boldsymbol{F}_{\bar{\mu}_j}^{-1})^{-1}(p) > (\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p), \\ -p + (\boldsymbol{F}_{\bar{\mu}_j}^{-1}) \circ (\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p), & \text{otherwise,} \end{cases} \\
&= \left|p - (\boldsymbol{F}_{\bar{\mu}_j}^{-1}) \circ (\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p)\right|.
\end{aligned}
$$

Thus

$$\frac{1}{T}\sum_{t=1}^{T}\int_0^1 \left|(\boldsymbol{F}_{j,t}^{-1}\ominus\boldsymbol{F}_{\bar{\mu}_j}^{-1})(p)-(\boldsymbol{F}_{j,t}^{-1}\ominus\boldsymbol{F}_{j,\oplus}^{-1})(p)\right|dp$$

$$=\int_0^1\frac{1}{T}\sum_{t=1}^{T}\left|(\boldsymbol{F}_{j,t}^{-1}\ominus\boldsymbol{F}_{\bar{\mu}_j}^{-1})(p)-(\boldsymbol{F}_{j,t}^{-1}\ominus\boldsymbol{F}_{j,\oplus}^{-1})(p)\right|dp=\int_0^1\left|p-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p)\right|dp.$$

By Lemma 5.2.2, we have for any fixed $p_0\in(0,1)$, $\left|p_0-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p_0)\right|=\mathcal{O}(\frac{1}{\sqrt{T}})$. Since $\left|p_0-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p_0)\right|$ is a bounded sequence uniformly over T, $\mathbb{E}\left|p_0-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p_0)\right|$ converges to 0. Thus, $\mathbb{E}\left|p-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p)\right|$ converges pointwise to 0. On the other hand, note that for any fixed $\epsilon>0$, we have

$$\begin{aligned}
&\lim_{T\to\infty}\mathbb{P}\left(\int_0^1\left|p-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p)\right|dp>\epsilon\right)\\
&\leqslant\frac{\lim_{T\to\infty}\mathbb{E}\int_0^1\left|p-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p)\right|dp}{\epsilon}\\
&\overset{(b)}{=}\frac{\lim_{T\to\infty}\int_0^1\mathbb{E}\left|p-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p)\right|dp}{\epsilon}\\
&\overset{(c)}{=}\frac{\int_0^1\lim_{T\to\infty}\mathbb{E}\left|p-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p)\right|dp}{\epsilon}=0.
\end{aligned}$$

Exchange (b) comes from that $\left|p-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p)\right|$ is bounded for all $p\in(0,1)$, thus $\mathbb{E}\left|p-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p)\right|$ exists for all $p\in(0,1)$. Moreover, $\mathbb{E}\left|p-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p)\right|$ is bounded uniformly over $p\in(0,1)$, thus we can furthermore exchange the integral and the limit by bounded convergence theorem, which brings to Equation (c). Therefore,

$$\int_0^1\left|p-(\boldsymbol{F}_{\bar{\mu}_j}^{-1})\circ(\boldsymbol{F}_{j,\oplus}^{-1})^{-1}(p)\right|dp\xrightarrow{p}0.$$

This implies $\frac{1}{T}\sum_{t=1}^{T}\langle\boldsymbol{F}_{j,t}^{-1}\ominus\boldsymbol{F}_{\bar{\mu}_j}^{-1}-\boldsymbol{F}_{j,t}^{-1}\ominus\boldsymbol{F}_{j,\oplus}^{-1},\boldsymbol{F}_{l,t}^{-1}\ominus\boldsymbol{F}_{\bar{\mu}_l}^{-1}\rangle_{Leb}\xrightarrow{p}0$. Since the proof above is valid for all $j,l=1,...,N$, we have immediately $\frac{1}{T}\sum_{t=1}^{T}\langle\boldsymbol{F}_{j,t}^{-1}\ominus\boldsymbol{F}_{j,\oplus}^{-1},\boldsymbol{F}_{l,t}^{-1}\ominus\boldsymbol{F}_{\bar{\mu}_l}^{-1}-\boldsymbol{F}_{l,t}^{-1}\ominus\boldsymbol{F}_{l,\oplus}^{-1}\rangle_{Leb}\xrightarrow{p}0$. Combined the two convergence results in Equation (5.4.3) gives $[\widehat{\boldsymbol{\Gamma}}(0)]_{j,l}-[\widetilde{\boldsymbol{\Gamma}}(0)]_{j,l}\xrightarrow{p}0$, for all $j,l=1,...,N$.

Analogously, we can prove $[\widehat{\boldsymbol{\Gamma}}(1)]_{j,l}-[\widetilde{\boldsymbol{\Gamma}}(1)]_{j,l}\xrightarrow{p}0$, for all $j,l=1,...,N$. Since matrix inversion is a continuous mapping, thus by continuous mapping theorem

$$[\widehat{\boldsymbol{\Gamma}}(0)^{-1}]_{j,l}-[\widetilde{\boldsymbol{\Gamma}}(0)^{-1}]_{j,l}\xrightarrow{p}0,\quad j,l=1,...,N.$$

Representations of $\widehat{\boldsymbol{A}}_o$, $\widetilde{\boldsymbol{A}}$ bring to $\widehat{\boldsymbol{A}}_o-\widetilde{\boldsymbol{A}}\xrightarrow{p}0$.

■

The last term in Equation (5.4.2) implies that it is sufficient to obtain

$$\frac{1}{T}\sum_{t=1}^{T}\langle\boldsymbol{F}_{j,t}^{-1}\ominus\boldsymbol{F}_{j,\oplus}^{-1},\boldsymbol{F}_{l,t}^{-1}\ominus\boldsymbol{F}_{l,\oplus}^{-1}\rangle_{Leb}-\mathbb{E}\langle\boldsymbol{F}_{j,t}^{-1}\ominus\boldsymbol{F}_{j,\oplus}^{-1},\boldsymbol{F}_{l,t}^{-1}\ominus\boldsymbol{F}_{l,\oplus}^{-1}\rangle_{Leb}\to0,$$

to prove $\widehat{\boldsymbol{A}}_o - A \to 0$. Thus we do not have to achieve the primary result until $\widetilde{\boldsymbol{A}} - A \to 0$. However, to show clearly the logic of the proof of Theorem 5.2.3, and the difference of the random objects involved, especially $\widetilde{\boldsymbol{F}}_t$ and $\widehat{F}_t$, we complete the result for $\widetilde{\boldsymbol{A}}$ and emphasize it in Theorem 5.2.3.

5.4.7 Proof of Theorem 5.2.4

The Lagrangian of Problem (5.2.8) is given by

$$L(\boldsymbol{\beta}, \Lambda) = \boldsymbol{f}_T(\boldsymbol{\beta}) + \sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{\beta}) + \sum_{j=1}^{p} \nu_j h_j(\boldsymbol{\beta}).$$

Let Λ^* be a dual solution, then the strong duality implies the primal solution $\hat{\boldsymbol{\beta}}$ minimizes $L(\boldsymbol{\beta}, \Lambda^*)$, with $\sum_{i=1}^{m} \lambda_i^* f_i(\hat{\boldsymbol{\beta}}) + \sum_{j=1}^{p} \nu_j^* h_j(\hat{\boldsymbol{\beta}}) = 0$, see Boyd et al. (2004, Section 5.5.2). Therefore we have

$$L(\hat{\boldsymbol{\beta}}, \Lambda^*) \leqslant L(\beta^*, \Lambda^*).$$

Furthermore,

$$\boldsymbol{f}_T(\hat{\boldsymbol{\beta}}) \leqslant \boldsymbol{f}_T(\boldsymbol{\beta}^*) + \sum_{i=1}^{m} \lambda_i^* f_i(\beta^*) + \sum_{j=1}^{p} \nu_j^* h_j(\beta^*).$$

We subtract $\boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o)$ from both sides of the inequality above, which gives

$$0 \leqslant \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}) - \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o) \leqslant \boldsymbol{f}_T(\boldsymbol{\beta}^*) - \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o) + \sum_{i=1}^{m} \lambda_i^* f_i(\beta^*) + \sum_{j=1}^{p} \nu_j^* h_j(\beta^*). \tag{5.4.4}$$

Note that the non-negativity comes from that $\hat{\boldsymbol{\beta}}_o$ is the minimizer of $\boldsymbol{f}_T$.

On the other hand, because $\boldsymbol{f}_T$ is strongly convex with the constant $\boldsymbol{\mu}_T$, thus for any $\boldsymbol{s} \in \partial \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o)$, there is (Zhou, 2018, Lemma 3 (iii)),

$$\boldsymbol{f}_T(\beta^*) - \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o) \leqslant \langle \boldsymbol{s}, \hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_o \rangle + \frac{1}{2\boldsymbol{\mu}_T} \|\beta^* - \hat{\boldsymbol{\beta}}_o\|_{\ell_2}^2. \tag{5.4.5}$$

Because $\hat{\boldsymbol{\beta}}_o$ is the minimizer of $\boldsymbol{f}_T$, thus $0 \leqslant \boldsymbol{f}_T(\beta^*) - \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o)$. Moreover, $\mathbf{0} \in \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o)$, thus, we pick $\boldsymbol{s} = \mathbf{0}$, Equation (5.4.5) becomes

$$0 \leqslant \boldsymbol{f}_T(\beta^*) - \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o) \leqslant \frac{1}{2\boldsymbol{\mu}_T} \|\beta^* - \hat{\boldsymbol{\beta}}_o\|_{\ell_2}^2.$$

Since $\|\beta^* - \hat{\boldsymbol{\beta}}_o\|_{\ell_2}^2 = \mathcal{O}_p(r_T)$, and $\frac{1}{2\boldsymbol{\mu}_T}$ is stochastically bounded, thus $\boldsymbol{f}_T(\beta^*) - \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o) = \mathcal{O}_p(r_T)$.

Furthermore, β^* satisfies the constraints, thus $\sum_{i=1}^{m} \lambda_i^* f_i(\beta^*) + \sum_{j=1}^{p} \nu_j^* h_j(\beta^*) \leqslant 0$, combined with the previous convergence result in Inequality (5.4.4), we have

$$\boldsymbol{f}_T(\hat{\boldsymbol{\beta}}) - \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o) = \mathcal{O}_p(r_T).$$

Use once again the strong convexity of $\boldsymbol{f}_T$, we obtain (Zhou, 2018, Lemma 2 (iii))

$$\boldsymbol{f}_T(\hat{\boldsymbol{\beta}}) - \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o) \geqslant \langle \boldsymbol{s}, \hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_o \rangle + \frac{\boldsymbol{\mu}_T}{2} \left\| \hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_o \right\|_{\ell_2}^2, \quad \boldsymbol{s} \in \partial \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o). \tag{5.4.6}$$

Plug $\boldsymbol{s} = \mathbf{0}$ in Equation (5.4.6), we get

$$\frac{2}{\boldsymbol{\mu}_T} \left(\boldsymbol{f}_T(\hat{\boldsymbol{\beta}}) - \boldsymbol{f}_T(\hat{\boldsymbol{\beta}}_o) \right) \geqslant \left\| \hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_o \right\|_{\ell_2}^2.$$

The stochastic boundedness and the convergence finally leads to $\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_o = \mathcal{O}_p(r_T)$. ■

Chapter 6

Learning node predictability

In this chapter, we rely on the reconstruction problem over a network to measure the predictability of nodes. In Section 6.1, we develop the introduction in Section 2.2.2.1 of the adaptation of kernel ridge regression to the reconstruction problem and the devised ranking procedure. We also present the design of kernel on the domain $\mathcal{N} \times \mathbb{Z}$. In Section 6.2, we evaluate the node predictability with neural networks and obtain a finer ranking with scores. We leverage the model aggregation effect of dropout technique to assess the overall reconstruction accuracy of each node. Additionally, in Section 6.3, we consider the linear regression for the case when the connectivity of nodes are not available. Lastly, numerical experiments are carried out in Section 6.4 on the bike-sharing data set that we have seen in the experiment of Chapter 5.

We recall the reconstruction problem. Let $(\boldsymbol{x}_{it})_t$, $i \in \mathcal{N} := \{1, \dots, N\}$ be the observations over the network of interest $\mathcal{N}$. Hypothetically, we would like to reconstruct the observations at time t over a subset I of nodes, $\boldsymbol{x}_{it}$, $i \in I \subset \mathcal{N}$, from the past and present recordings on the rest of nodes $(\boldsymbol{x}_{j\tau})_{j\tau}$, $j \in I^c, t - H \leqslant \tau \leqslant t$. In this chapter, we apply the classical practice in multivariate time series domain to center or detrend the raw series $(\boldsymbol{x}_{it})_t$, $i \in \mathcal{N}$ in a preprocessing step. Thus we do not explicitly deal with non-zero process mean in the proposed models.

6.1 Kernel ridge regression of time and node predictor

As introduced in Section 2.2.2.1, to reconstruct a real value indexed by (thus mapped from) $i \in \mathcal{N}, t \in \mathbb{Z}$, we approximate the regression operator by the functions in RKHS $\mathcal{H}_k$ induced by a spatio-temporal kernel k on $\mathcal{N} \times \mathbb{Z}$. We furthermore require the kernel value to only depend on the time lag h rather than specific time stamps, namely $k\left[(i,t),(j,t-h)\right] = k\left[(i,t'),(j,t'-h)\right]$. Hence, we can denote kernel value $k\left[(i,t),(j,t-h)\right]$ by $k(i,j,h)$, analogous to the notations of auto-covariance for stationary process. Since the rest of kernel design does not impact the development of the ranking method, we present the proposed kernel design at the end of this section. We introduce the following notations.

Let $K_{\mathcal{N}}(h)$ denote the Gram matrix of $\mathcal{N} \times \{h\}$ for $h \in \mathbb{Z}$, that is

$$K_{\mathcal{N}}(h) \in \mathbb{R}^{N \times N} \text{ with } [K_{\mathcal{N}}(h)]_{ij} = k(i, j, h), \quad i, j \in \mathcal{N}.$$

We then denote the matrix extraction $[K_{\mathcal{N}}(h)]_{i,I^c}$ by $K_{iI^c}(h)$. Analogously, we have $K_{I^c}(h) = [K_{\mathcal{N}}(h)]_{I^c}$. The extractions are defined the same way as in Chapter 4. We moreover define the block matrix

$$K_{I^c}^H = \begin{pmatrix} K_{I^c}(0) & K_{I^c}(1) & \cdots & K_{I^c}(H) \\ K_{I^c}(-1) & K_{I^c}(0) & \cdots & K_{I^c}(H-1) \\ \vdots & \vdots & \ddots & \vdots \\ K_{I^c}(-H) & K_{I^c}(-H+1) & \cdots & K_{I^c}(0) \end{pmatrix} \in \mathbb{R}^{q \times q}.$$

We now recall the best approximation namely the estimator in kernel ridge regression is given by

$$\widehat{f}_t = \min_{f \in \mathcal{H}_k} \sum_{i \in I^c, t-H \leqslant \tau \leqslant t} \|x_{i\tau} - f(i, \tau)\|_{\ell^2}^2 + \lambda \|f\|_{\mathcal{H}_k}^2.$$

Representer theorem indicates that $\widehat{f}_t$ has the form

$$\widehat{f}_t(i, \tau) = \begin{pmatrix} K_{iI^c}(\tau - t) & \cdots & K_{iI^c}(\tau - t + H) \end{pmatrix} \alpha_t^*, \quad i \in \mathcal{N},\ \tau \in \mathbb{Z},$$

where

$$\alpha_t^* = (K_{I^c}^H + \lambda I_q)^{-1} x_{I^c t}^H \quad \text{and} \quad x_{I^c t}^H = (x_{I^c t}, x_{I^c, t-1}, \ldots, x_{I^c, t-H}).$$

Note that $x_{I^c t}^H$ is the observations on nodes I^c from time t back to time $t - H$. Thus we have the prediction of x_{it} as $K_{iI^c}^H \alpha_t^*$ for all $i \in I$, where

$$K_{iI^c}^H = \begin{pmatrix} K_{iI^c}(0) & K_{iI^c}(1) & \cdots & K_{iI^c}(H). \end{pmatrix}$$

The total prediction error given the history of network observations $(x_{it})_{i,t}, i \in \mathcal{N}, 1 \leqslant t \leqslant T$ can then be measured by the residual sum of squares

$$\begin{aligned} \sum_{i \in I} RSS(i | I^c) &= \sum_{i \in I} \frac{1}{T} \sum_{t=H+1}^{T} (x_{it} - \widehat{\Theta}_{i\lambda} x_{I^c t}^H)^2 \\ &= \sum_{i \in I} \left(\widehat{\sigma}_i - 2 \widehat{\beta}_{iI^c}^H \widehat{\Theta}_{i\lambda}^\top + \widehat{\Theta}_{i\lambda} \widehat{\alpha}_{I^c}^H (\widehat{\Theta}_{i\lambda})^\top \right), \end{aligned} \tag{6.1.1}$$

where $\widehat{\Theta}_{i\lambda}$ denotes $K_{iI^c}^H (K_{I^c}^H + \lambda I_q)^{-1}$,

$$\widehat{\alpha}_{I^c}^H = \begin{pmatrix} \widehat{\Gamma}_{I^c}(0) & \widehat{\Gamma}_{I^c}(1) & \cdots & \widehat{\Gamma}_{I^c}(H) \\ \widehat{\Gamma}_{I^c}(-1) & \widehat{\Gamma}_{I^c}(0) & \cdots & \widehat{\Gamma}_{I^c}(H-1) \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{\Gamma}_{I^c}(-H) & \widehat{\Gamma}_{I^c}(-H+1) & \cdots & \widehat{\Gamma}_{I^c}(0) \end{pmatrix} \in \mathbb{R}^{q \times q}, \quad \widehat{\Gamma}_{I^c}(h) = [\widehat{\Gamma}(h)]_{I^c}, \tag{6.1.2}$$

$$\widehat{\beta}_{iI^c}^H = \begin{pmatrix} \widehat{\Gamma}_{iI^c}(0) & \widehat{\Gamma}_{iI^c}(1) & \cdots & \widehat{\Gamma}_{II^c}(H) \end{pmatrix} \in \mathbb{R}^{p \times q}, \quad \widehat{\Gamma}_{iI^c}(h) = [\widehat{\Gamma}(h)]_{iI^c}, \tag{6.1.3}$$

and $\widehat{\Gamma}$ are the sample auto-covariance matrices. Likewise, the extractions in the definition above are defined the same way as in Chapter 4.

To rank the predictability of all nodes in the network, we propose to rely on the greedy adaptation of a sensor selection problem to take different I into account. We consider such sensor selection problem where for a given cardinality p, we search the optimal set $I^*(p)$ which minimizes the criteria $\sum_{i\in I} RSS(i|I^c)$. The greedy strategy of this selection problem makes up the first p steps of the proposed ranking algorithm 7, and it extends compatibly to $p = N - 1$.

Algorithm 7 Ranking of predictability evaluated by the kernel predictor.

Input: $(\widehat{\Gamma}(j))_{j=0}^{H}$, $K_{\mathcal{N}}^{H}$, λ.
Initialize: $n = 1$, $I_0^c = \mathcal{N}$. Compute $(K_{I_0^c}^H + \lambda I_q)^{-1}$.
for $n < N$ **do**
  for $i \in I_{n-1}^c$ **do**
    $S_i \leftarrow I_{n-1}^c \backslash i$.
    Compute $(K_S^H + \lambda I_q)^{-1}$ from $(K_{I_{n-1}^c}^H + \lambda I_q)^{-1}$ (block matrix inversion formula).
    Compute $RSS(i|S_i)$ as Equation (6.1.1).
  end for
  $i_n \leftarrow \arg\min_{i\in I_{n-1}^c} RSS(i|S_i)$.
  $I_n^c \leftarrow I_{n-1}^c \backslash i_n$.
  Compute $(K_{I_n^c}^H + \lambda I_q)^{-1}$ from $(K_{I_{n-1}^c}^H + \lambda I_q)^{-1}$ (block matrix inversion formula).
end for
Output: $i_1, i_2, \ldots, i_N$.

We recall the interpretation of the ranking derived from the greedy adaptation: for every given cardinality p, we assign the nodes in the optimal set $I^*(p)$ one score, the rest nodes zero score. The total scores for all p's from 1 to N forms the ranking.

Note that, in Algorithm 7, we still rely on the block matrix inversion formula to treat $(K_S^H + \lambda I_q)^{-1}$ when S slides over I_n^c to gain $\mathcal{O}(H|I_n^c|)$ speed-up with respect to the naive matrix inversion. The inner loop of Algorithm 7 will cost $\mathcal{O}(H^3|I_n^c|^3)$, therefore the whole ranking algorithm at $\mathcal{O}(H^3N^4)$. The algorithm requires to be moreover optimized to handle large scale networks in the future work.

Lastly, for the time series over a network, we propose the following kernel design to describe the similarity of data points. Assuming that we have the knowledge of the connectivity of nodes $\mathcal{N}$, characterized by the edge set $\mathcal{E}$, we can construct the corresponding adjacency matrix A, and the graph Laplacian L by the definitions in Section 3.1. The Laplacian defines a popular kernel on the vertex domain as below.

Definition 6.1.1. Given Laplacian matrix of a graph, denoted by L. If L admits orthonormal eigendecomposition denoted as $\Phi\Lambda\Phi^\top$, we can define the corresponding graph Laplacian kernel $k_{\mathcal{G}} : \mathcal{N} \times \mathcal{N} \to \mathbb{R}$ as $k_{\mathcal{G}}(i,j) = K_{ij}$, where

$$K = \Phi r(\Lambda)\Phi^t,$$

with $r : \mathbb{R}_+ \to \mathbb{R}_+$ a pre-given mapping. A typical example of r is

$$r(\lambda) = \begin{cases} \lambda^{-1}, & \lambda \neq 0, \\ 0, & \text{otherwise}, \end{cases}$$

for which the matrix $K = L^{-}$ is the Moore-Penrose inverse of L.

We then propose kernel (6.1.4) for the domain $\mathcal{X} = \mathcal{N} \times \mathbb{Z}$.

Definition 6.1.2. The kernel $k_{\mathcal{G}T} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is defined as

$$k_{\mathcal{G}T}\left[(i,t),(j,t')\right] = k_{\mathcal{G}}(i,j) \cdot k_T(i,j,t-t'), \tag{6.1.4}$$

where $k_{\mathcal{G}}$ is the graph kernel, $k_T(i,j,t-t') = [\widehat{\Gamma}(t-t')]_{i,j}$.

By Definition 3.5.1, k_T is a valid kernel. Thus according to Remark 3.5.2, $k_{\mathcal{G}T}$ is a valid kernel as well.

6.2 Dropout in predictability evaluation for neural network predictors

We firstly adapt a given network to the reconstruction problem introduced for the predictability evaluation. For the network predictor, we only use the present observations $(x_{jt})_{j\in I^c}$ to reconstruct $(x_{it})_{i\in I}$. Since in experiment, we do not see the gain from regressing on further history. In order to facilitate the application of dropout, we do not directly take $(x_{jt})_{j\in I^c}$ as input of network. Instead, we compose a complete graph signal $\widetilde{\mathrm{x}}_t^{I^c}$ by inserting zeros to the places of $(x_{it})_{i\in I}$, that is

$$\left[\widetilde{\mathrm{x}}_t^{I^c}\right]_i = \begin{cases} x_{it}, & i \in I^c, \\ 0, & i \in I. \end{cases}$$

$\widetilde{\mathrm{x}}_t^{I^c}$ is then an input sample. Its output is denoted by $\widehat{\mathrm{x}}_{It} = (\widehat{x}_{it})_{i\in I}$, which is compared to the target $\mathrm{x}_{It} = (x_{it})_{i\in I}$ through the loss function J. We define

$$J(\widehat{\mathrm{x}}_{It}; \mathrm{x}_{It}) = \|\widehat{\mathrm{x}}_{It} - \mathrm{x}_{It}\|_{\ell_2}^2 .$$

When no dropout is applied, the gradient used to update a network parameter v at the n-th updating step is then computed as the average

$$\frac{1}{|B_n|} \sum_{t\in B_n} \frac{\partial J(\widehat{\mathrm{x}}_{It}(v); \mathrm{x}_{It})}{\partial v},$$

where B_n is the batch of the step.

Now in order to vary the set I within the same training, we apply dropout on the input layer. Instead of $\widetilde{\mathrm{x}}_t^{I^c}$, we input the full graph signal x_t and multiply it with the dropout vector $\mathbf{w}^{(n)}$ to randomly choose a missing set I_n. We recall that we re-sample the dropout at each batch. Thus the actual inputs to the prediction network from batch B_n are $\mathrm{x}_t \odot \mathbf{w}^{(n)}$, $t \in B_n$. On the output layer, we change the dimension from $|I|$ to N, we denote this output by $\widehat{\mathrm{x}}_t$. We apply the reverse dropout on the batch loss as

$$\frac{1}{|B_n|} \sum_{t\in B_n} \left\| \widehat{\mathrm{x}}_t \odot (1 - \mathbf{w}^{(n)}) - \mathrm{x}_t \odot (1 - \mathbf{w}^{(n)}) \right\|_{\ell_2}^2 .$$

Thus the computation of the gradient does not take into account the prediction errors of observations on I^c. Because the missing set I_n remains unchanged for all samples in batch B_n, at each optimization step, the network is trained as a specific prediction model with input $\widetilde{\mathrm{x}}_t^{I_n^c}$ and target $\mathrm{x}_{I_n t}$. Sampling the dropout vectors then varies missing set I across the optimization steps in a training.

Therefore, by the same spirit of model aggregation, we propose to test the trained model on the validation set V without dropout. We input the complete graph signal x_t to the network and obtain the output $\widehat{\mathrm{x}}_t$. For each node i, we calculate the total prediction error on the validation set as

$$R^2\left((\widehat{x}_{it})_{t\in V}, (x_{it})_{t\in V}\right) := 1 - \frac{\sum_{t\in V}(x_{it} - \hat{x}_{it})^2}{\sum_{t\in V}(x_{it} - \bar{x}_i)^2}, \tag{6.2.1}$$

where $\bar{x}_i = \sum_{t\in V} x_{it}$. Note that, we use R^2 score to measure the prediction error instead of the ℓ_2 norm as before. Because we observe in the experiment that the selected nodes by R^2 have better reconstruction performance. The prediction error (6.2.1) from the aggregated network represents the overall predictability of node i when it is predicted within different I's. We take this prediction error as the score of node, which derives furthermore the rank. The higher scores are endowed with the higher ranks. Figure 6.1 reviews the predictability learning by a network.

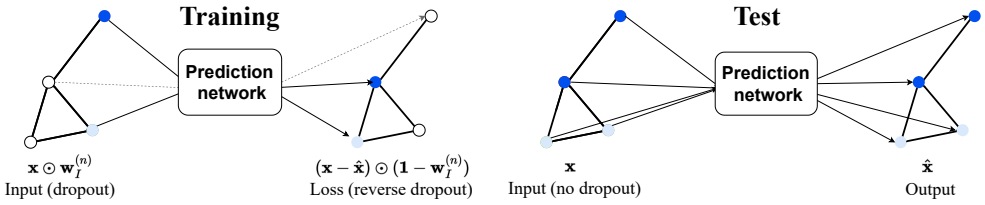

Figure 6.1: *Application of the dropout in predictability learning.* On the left represents the the optimization step on the n-th batch. I is sampled again at every batch n. After the training converges, the network is tested on the validation set as illustrated on the right, whose reconstruction errors of each node gives its score.

Lastly, we give the details of the training stage in the predictability learning. The training is terminated at the early stopping point. Namely, at the end of each epoch, we test the latest updated network on the validation set V without dropout, and calculate the validation loss

$$\frac{1}{|V|}\sum_{t\in V}\|\mathrm{x}_t - \widehat{\mathrm{x}}_t\|_{\ell_2}^2 .$$

We stop the training when the validation loss does not decrease with respect to the previous epoch. The key is the convergence of the validation loss. In the experiments, it shows when a small dropout rate ($\leqslant 0.2$, $|B_n| = 100$ for our data sets) is given, the validation loss will converge. By contrast, when the dropout rate is large, the loss is likely to become infinity in training, since the missing rate is relatively high for the reconstruction. Additionally, we use gradient descent as the optimization algorithm, because it does not use the previously computed gradients to update network parameters, so that each batch represents an independent model training.

6.3 Partial variance of multivariate time series

When no edge information is provided, to predict $(\boldsymbol{x}_{it})_{i\in I}$ given $(\boldsymbol{x}_{j\tau})_\tau$, $j \in I^c, t-H \leqslant \tau \leqslant t$, we consider the linear regression to introduce the notion of partial variance. We assume the multivariate process $\mathbf{x}_t = (\boldsymbol{x}_{it})_{i\in\mathcal{N}}$ is stationary under the definition of Section 3.2.1, with zero mean and auto-covariance matrices $\Gamma(h)$. Then the partial variance of $(\boldsymbol{x}_{it})_t$, $i \in I \subset \mathcal{N}$ given $(\boldsymbol{x}_{i\tau})_\tau$, $i \in I^c, t-H \leqslant \tau \leqslant t$ is defined by the variance of the regression residual as

$$\sigma^2_{i|I^c} = \min_{\Theta_i\in\mathbb{R}^{N-p}} \mathbb{E}\|\boldsymbol{x}_{it} - \Theta_i \mathbf{x}^H_{I^c t}\|_{l_2}^2 = \sigma_i - \beta^H_{iI^c}(\alpha^H_{I^c})^{-1}(\beta^H_{iI^c})^\top$$

where

$$\alpha^H_{I^c} = \begin{pmatrix} \Gamma_{I^c}(0) & \Gamma_{I^c}(1) & \cdots & \Gamma_{I^c}(H) \\ \Gamma_{I^c}(-1) & \Gamma_{I^c}(0) & \cdots & \Gamma_{I^c}(H-1) \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_{I^c}(-H) & \Gamma_{I^c}(-H+1) & \cdots & \Gamma_{I^c}(0) \end{pmatrix} \in \mathbb{R}^{q\times q}, \quad \Gamma_{I^c}(h) = [\Gamma(h)]_{I^c},$$

$$\beta^H_{iI^c} = \begin{pmatrix} \Gamma_{iI^c}(0) & \Gamma_{iI^c}(1) & \cdots & \Gamma_{iI^c}(H) \end{pmatrix} \in \mathbb{R}^{p\times q}, \quad \Gamma_{iI^c}(h) = [\Gamma(h)]_{iI^c}.$$

In practice, $\sigma^2_{i|I^c}$ can be estimated by the sample auto-covariance matrices as

$$\widehat{\boldsymbol{\sigma}}^2_{i|I^c} = \widehat{\boldsymbol{\sigma}}_i - \widehat{\boldsymbol{\beta}}^H_{iI^c}(\widehat{\boldsymbol{\alpha}}^H_{I^c})^{-1}(\widehat{\boldsymbol{\beta}}^H_{iI^c})^\top, \tag{6.3.1}$$

where $\widehat{\boldsymbol{\alpha}}^H_{I^c}$ and $\widehat{\boldsymbol{\beta}}^H_{iI^c}$ are defined in Equations (6.1.2) and (6.1.3) for the RSS formula of kernel predictor. It is easy to find sample partial variance $\widehat{\boldsymbol{\sigma}}^2_{i|I^c}$ is a special case of $RSS(i|I^c)$ with $\lambda = 0$ and the kernel defined totally as k_T in Definition 6.1.4. Thus by the same procedure as Section 6.1, we relate $\sum_{i\in I}\widehat{\boldsymbol{\sigma}}^2_{i|I^c}$ to the criteria of the sensor selection problem and derive Algorithm 8 from its greedy adaptation.

Algorithm 8 Ranking of predictability measured by partial variance.

Input: $(\widehat{\boldsymbol{\Gamma}}(j))_{j=0}^H$.
Initialize: $n = 1$, $I^c_0 = \mathcal{N}$. Compute $(\widehat{\boldsymbol{\alpha}}^H_{I^c_0})^{-1}$.
for $n < N$ **do**
  for $i \in I^c_{n-1}$ **do**
    $S_i \leftarrow I^c_{n-1}\backslash i$.
    Compute $(\widehat{\boldsymbol{\alpha}}^H_S)^{-1}$ from $(\widehat{\boldsymbol{\alpha}}^H_{I^c_{n-1}})^{-1}$ (block matrix inversion formula).
    Compute $\widehat{\boldsymbol{\sigma}}^2_{i|S_i}$ as Equation (6.3.1).
  end for
  $i_n \leftarrow \arg\min_{i\in I^c_{n-1}} \widehat{\boldsymbol{\sigma}}^2_{i|S_i}$.
  $I^c_n \leftarrow I^c_{n-1}\backslash i_n$.
  Compute $(\widehat{\boldsymbol{\alpha}}^H_{I^c_n})^{-1}$ from $(\widehat{\boldsymbol{\alpha}}^H_{I^c_{n-1}})^{-1}$ (block matrix inversion formula).
end for
Output: $i_1, i_2, \ldots, i_N$.

6.4 Experiments

In this section, we evaluate the proposed methods on the bike-sharing data set in Chapter 5. Especially, we apply the inferred rankings to the sensor selection problem and compare the reconstruction performance of the selected sensors with the random sampling method in Puy et al. (2018).

6.4.1 Settings

Training test split We split the whole data set into training and test sets by the ratio of $0.8 : 0.2$. The predictability learning for all three proposed methods use the training set. When comparing the reconstruction performance of the selected set with the literature method, we construct the corresponding predictors still with the training set data. We only use the test set for reconstructing the unseen observations on the selected nodes in order to compare the method performance. For the network method, we take furthermore the last 0.1 history of training set as validation set.

Network and training For the network predictor, we take *ChebNet* (Defferrard et al., 2016) as an example to illustrate the proposed method in Section 6.2. The ChebNet used in the experiments consists of the input layer, 1 convolutional layer ($K_1 = 50$, $F_1 = 16$) with no pooling, followed by 3 fully-connected layers (respectively, 128, 500, and 64 neurons, all with bias terms) and the output layer. When learning the predictability, the input and output dimensions are both 274. The hyper-parameters in its training stage are: dropout rate 0.05, batch size 100, and learning rate 0.05. When comparing the reconstruction performance, we re-train the network with the output dimension reduced to $|I|$. The training samples are constructed as explained in the beginning of Section 6.2. The batch size and the learning rate are reset as 1000 and 0.001, respectively. We use early stopping in all network training.

Graph construction For kernel ridge regression, the ChebNet, and the random sampling method of Puy et al. (2018), we build the graph by connecting each station with its k-nearest neighbors using the geographical coordinates. More specifically, we first compute the Euclidean distance between nodes (sensors) using the latitude and longitude, denoted as $d(\cdot, \cdot)$. Then, for each node i, we take its k_0 nearest neighbors, denoted as $i_1, i_2, ..., i_{k_0}$, with the associated distances $d(i, i_1), d(i, i_2), ..., d(i, i_{k_0})$. We add edges (i, i_s), $i = 1, ..., N, s = 1, ..., k_0$ into $\mathcal{E}$, with weights calculated as

$$a_{ii_s} = \exp\left(-\frac{d^2(i, i_s)}{\sigma_i \sigma_{i_s}}\right)$$

where σ_j is the local scale of node j, that is chosen as $\sigma_j = d(j, j_{k_1})$, with j_{k_1} the k_1-th nearest neighbour of j. This self-tuning approach is proposed in Zelnik-Manor and Perona (2005). In our experiments, we took $k_0 = 20$ and $k_1 = 7$. For the pairs $(i, j) \notin \mathcal{E}$, a_{ij} is defined as 0. Thus, we have built the adjacency matrix $A = (a_{ij})_{i,j}$. We use the combinatorial Laplacian for all the graph methods.

Detrend and scale The bike-sharing data set shows a significant periodic trend, thus in the preprocessing step we remove it from the raw series $(x_{it})_t$, $i \in \mathcal{N}$. We use the training set to estimate the trend, then use the estimate to detrend the whole data set. More specifically, let T denote the total hours of training set. Because the time series $(x_{it})_t$ of data set is recorded on an hour basis, we estimate the weekly profile of data set as

$$b_{m,i} = \sum_{\tau \in I_m} \frac{x_{i\tau}}{|I_m|}, \quad m = 0, 1, ..., 167,$$

where $I_m = \{\tau = 1, \ldots, T : \tau \bmod 168 = m\}$. We detrend as $x_{it} - b_{m,i}$, $t \bmod 168 = m$. Additionally, we divide each detrended series by its standard deviation. Figure 6.2 gives an example of a comparison between the original and pre-processed series.

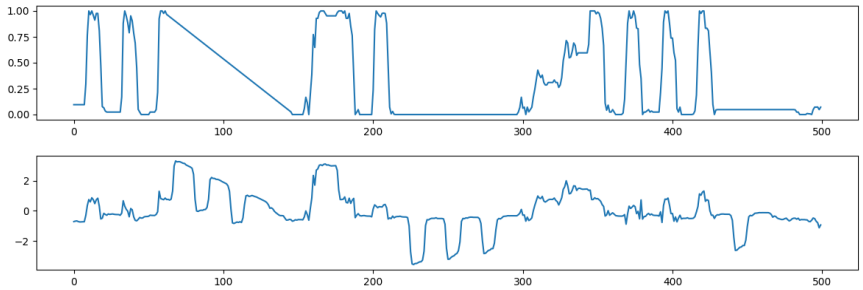

Figure 6.2: *Data preprocessing: detrend and scale.* Top is the original time series. Bottom is the processed time series.

Others H is given as 3 for the kernel and the linear methods. The regularization parameter for kernel method is set as $0.1\lambda_{\max}$, where $\lambda_{\max}$ is the largest eigenvalue of the Gram matrix $K_{\mathcal{N}}^H$ or $\widehat{\boldsymbol{\alpha}}_{\mathcal{N}}^H$.

6.4.2 Results

We have already shown in Section 2.2.2.2 the scores of node predictability learned through the dropout scheme. Since the scheme carries randomness, in Figure 6.3, we show the scores obtained from another run. Their training curves are given in Figure 6.4.

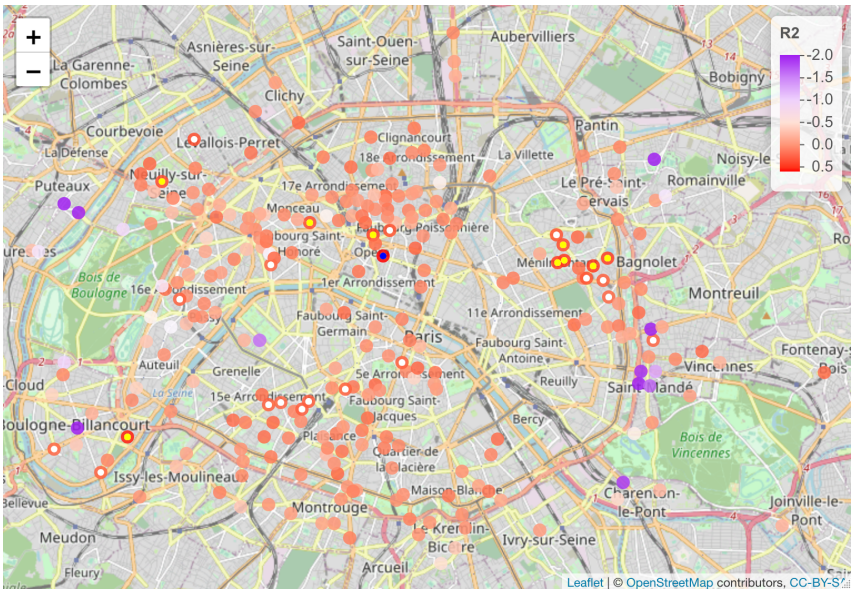

Figure 6.3: *Scores of node predictability for bike-sharing network in Paris.* The higher scores are shown in darker red and the lower in darker purple. The top 1, 10, and 27 nodes are in blue, yellow, and white respectively. For the nodes with lowest scores (lower than -2), we clipped their scores to -2 in this visualization to make the color change of other nodes visible. The same measure for Figure 2.6.

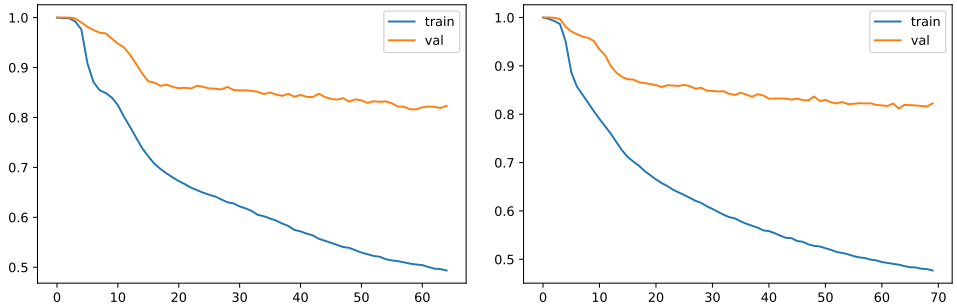

Figure 6.4: *Training curves of the dropout scheme.* Left corresponds to the result in Figure 2.6. The training stops at the 65-th epoch. Right corresponds to Figure 6.3. The training stops at the 70-th epoch. The number of I's explored in two trainings are 2015 and 2170 respectively. We normalized the loss curves in order to compare the training and the validation losses by the same scale.

We can see that the results do not change much across different runs, especially the nodes endowed the highest (in blue, yellow, and white) and lowest predictability (in dark purple) are very close. On the other hand, the curves in Figure 6.4 validate the designed training can converge. Both points support that the proposed learning method is valid and effective.

In Figure 6.5, we compare the derived node scores with the sampling probability of nodes proposed in Puy et al. (2018). In all the figures of visualization, we use the

same colors to indicate the comparable concepts. The nodes with higher scores and the nodes with lower sampling probability will both be removed from the observed set (on average for the latter since it is not a deterministic method). We can see that both strategies suggest to keep the nodes from suburb in the graph. From the graph point of view, the suburb nodes have less near neighbours in the graph. For the aspect of data, the suburb stations have less chance to be visited by users, thus their observations are noisier and harder to be reconstructed. Therefore, it is reasonable to remain their observations in the reconstruction.

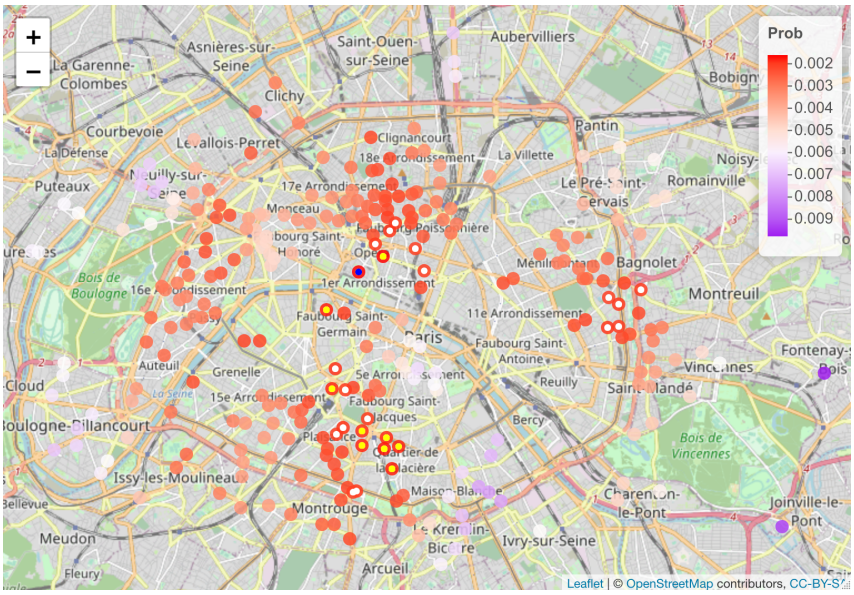

Figure 6.5: *Variable density sampling distribution from Puy et al. (2018) for Paris network.* Lower sampling probabilities (equivalent to higher scores) are shown in red. The top 1, 10, and 27 probabilities are in blue, yellow, and white respectively.

Figure 6.6 represents the rankings of predictability evaluated by the kernel predictor in Section 6.1 and the partial variance defined in Section 6.3. We can see that the two rankings show similar patterns. Especially, they both endow high predictability to stations in the transport towards La defense (upper left corner of the map). Besides, the kernel predictor indicates again the group of well-predictable stations around the train station Paris Montparnasse. Whereas, the partial variance suggests more predictable stations in the suburb areas.

Next, we evaluate the application of the predictability ranking to the sensor selection problem. We compare in Figure 6.7 the different methods their reconstruction errors of the selected set I. The reconstruction errors are measured by ℓ_2 norm for an increasing size of I. For the random sampling method, the underlying reconstruction method is given in (Puy et al., 2018, Equation (5)). We only report the results from the bandlimit $k = 10$ which has the best reconstruction performance. We can see that the data-driven strategies bring better reconstruction performance for all tested sampling rates. Lastly, we show in Figure 6.8 the representative reconstruction of highest rank node signals along time using the three prediction methods. The

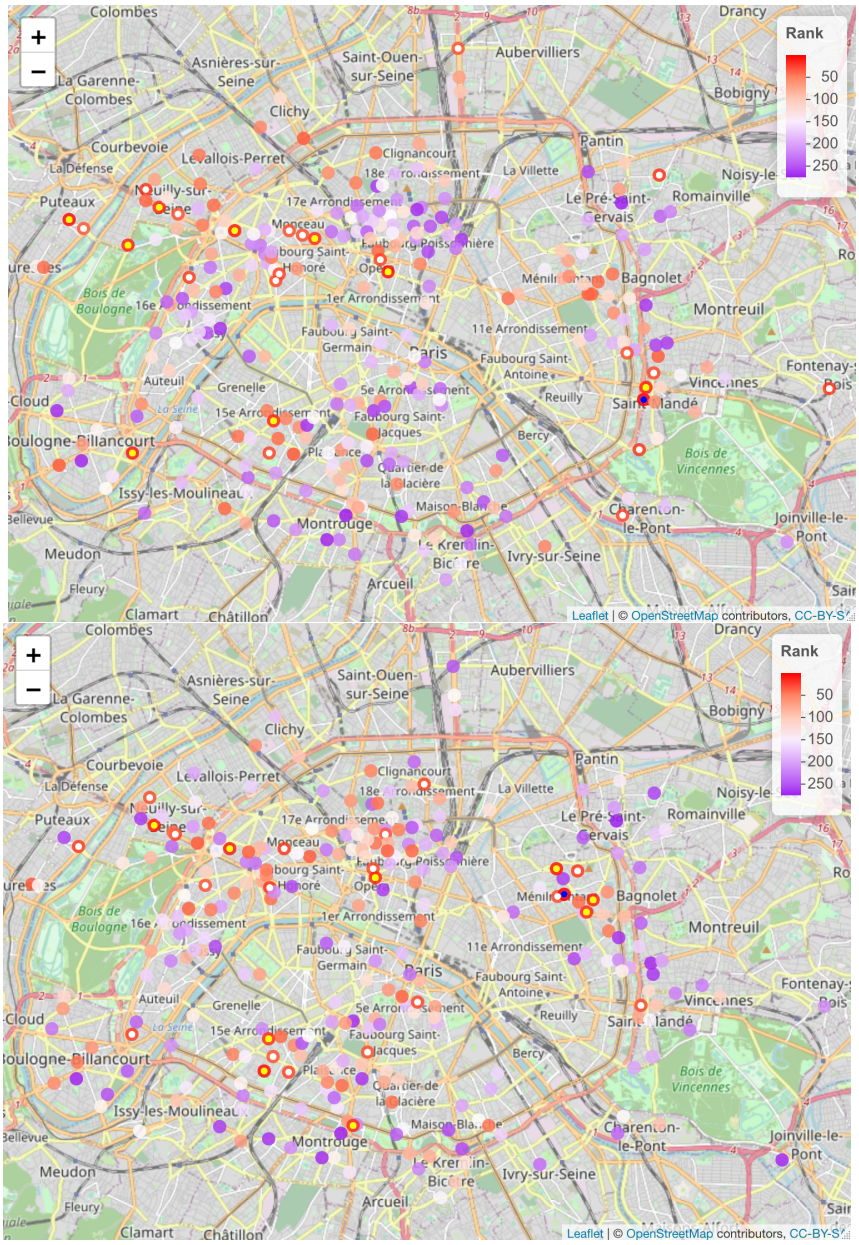

Figure 6.6: *Rankings by linear (top) and kernel method (bottom) for Paris network.* The ranks of nodes are represented by color, which is consistent with Figures 6.3 and 6.5.

results show the satisfactory reconstruction for both graph methods. However, for the linear method, even though it has good reconstruction error values, but it can not distinguish the regular signals from the noisy signals.

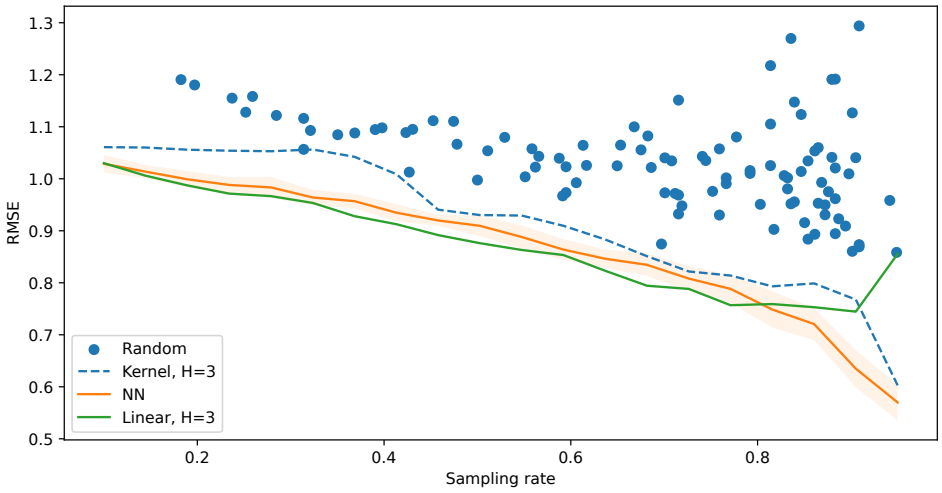

Figure 6.7: *Root mean squared error of graph signal reconstruction (normalized by the sampling rate).* Sampling rate is given by $1 - |I|/N$. The sampling rate for the method in Puy et al. (2018) is the effective sampling rate which only considers the unique nodes. For the network predictor, due to the randomness of training, we repeat the evaluation procedure (from the dropout scheme to the predictor re-training) 10 times. The orange curve represents the mean RMSE, while the shaded area around represents the 1 standard deviation.

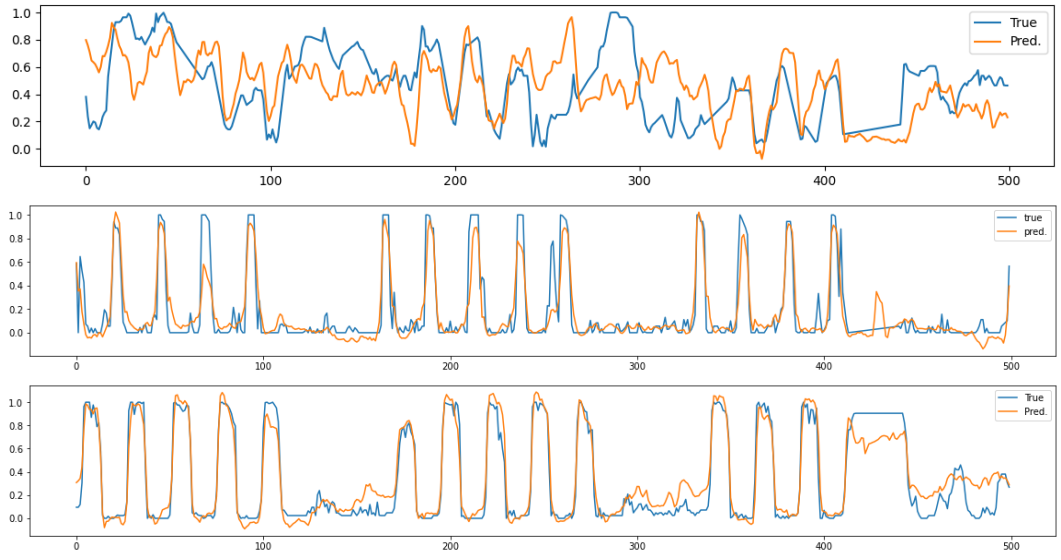

Figure 6.8: *Signal reconstruction of the respective highest rank nodes, from linear (top), kernel (middle), and network predictors (bottom).* The signals are predicted by all nodes after rank 13 (in a context of removing 5% nodes). The trends are added back to the predictions.

Chapter 7

Conclusion and perspectives

In this thesis, we provided the new tools to analyze the data of diverse natures recorded over a network of sensors. These tools are especially designed for a purpose of revealing the information which is not evident to the elementary data analysis, but valuable for further analysis, prediction, and decision making.

Chapter 4 and 5 focus on identifying the links between the evolution of time series observed from different nodes in a network. We proposed the novel auto-regressive models to describe the data. Therefore, by estimating the model coefficients, we were able to infer the links.

In Chapter 4, we were concerned by the more practically involved issue, which is the online inference from the raw data. We proposed the data model for the matrix-variate time series with periodic trends, and introduce two learning frameworks respectively in both low- and high- dimensional regimes. Especially in the high dimensional learning, we introduce the novel Lasso type (4.2.4) and extend the classical Homotopy algorithms. Lasso (4.2.4) differs from the classical Lasso (4.2.5) not only by replacing the coefficient vector with the coefficient matrix A, but mainly by requiring A to have the desired structure indicated by $\mathcal{K}_{\mathcal{G}}$ and the partial sparsity penalized by $\|A_{\mathrm{N}}\|_{\ell_1}$. Thus the derivation of the Homotopy algorithms provides useful techniques to address the structure constraint. Moreover, this derivation does not rely on the specific structure, nor on the particular partial sparsity regularization. Therefore, they can be applied to other model designs. Other model extensions are possible, for example, from matrix-variate to tensor-variate time series by using the multi-way Kronecker sum notion, or considering more time lag terms in the matrix-variate AR model, and accordingly replacing Lasso penalty with group Lasso penalty in (4.2.4).

In Chapter 5, we extended the standard VAR models to the distributional multivariate AR models. The proposed model provides a way to model a collection of multiple time-dependent probability measures, and to represent their dependency structure by a directed weighted graph at the same time. Especially, the proposed data centering method for random measures allows the development of auto-regressive and regressive models with multiple predictors. Moreover, the empirical studies on the real data sets demonstrate that, the proposed approaches equipped with the distributional data representation are the efficient tools for analyzing and understanding the spatial-temporal data of distributional nature. For future research directions, this

paper provides a class of multivariate AR models for distributional time series which favors the graph learning. More classes which suit different data analysis purposes are expected to explore.

Lastly, in Chapter 6, we are interested in understanding the predictability of the data observed on different nodes in a network. We proposed the approaches to evaluate moreover rank the predictability of nodes with respect to the linear, kernel, and neural network predictors. The derived rankings allow to understand furthermore the spatial distribution of such node predictability, as illustrated in the numeric experiments. Additionally, the rankings can serve as data-driven strategies for sensor selection. The presence of historical data has significantly improved the reconstruction performance. In particular, the sensor selection based on the neural networks as a reconstruction method is innovative, which is far from the existing approaches.

Mid-term: Object data analysis. Except developing extension and variants for the models proposed in this thesis, a more general direction introduced by the works in Chapters 4 and 5 is the object data analysis, which is an emerging domain gaining popularity in recent statistics. The object data analysis deals with the statistical learning of the data, where a data point is not longer a scalar or a vector, instead, is an object, such as matrix or probability measure from our works. Other examples include data on the surface of spheres (Di Marzio et al., 2014; Zhu and Müller, 2022), and phylogenetic trees (Billera et al., 2001).

The challenge of analysing such data of complex type is that data points do not lie in a vector space, thus Euclidean methods will fail as mentioned in Chapter 5. However, the work there also demonstrates an effective treatment is view the data points as random objects in some metric space, such as Wasserstein space for distributional data. This treatment is also considered by other recent works in the domain of object data analysis. This is because of the widely availability of the notion of distance. Meanwhile, modeling in a metric space has its own advantages. More specifically, the weighted expectation and weighted integral can be easily established in a general metric space, following the idea of Fréchet mean. On noting that the weighted expectation $\mathbb{E}[w(\boldsymbol{x})\boldsymbol{y}]$ and weighted integral $\int w(t)f(t)dt$ in Euclidean setting minimize respectively the distance functionals

$$\arg\min_{c\in\mathbb{R}} \mathbb{E}[w(\boldsymbol{x})(c-\boldsymbol{y})^2], \text{ with } \mathbb{E}w(\boldsymbol{x}) = 1,$$

and

$$\arg\min_{x\in\mathbb{R}} \int w(t)\left[x - f(t)\right]^2 dt, \text{ with } \int w(t)dt = 1,$$

the weighted expectation and the weighted integral in a general metric space $(\mathcal{X}, d)$ can be defined by the minimizers of the analogue functionals in terms of d:

$$\arg\min_{C\in\mathcal{X}} \mathbb{E}[w(\boldsymbol{x})d[C,\boldsymbol{Y}]^2],$$

and

$$\int_{\oplus} w(t)F(t)dt := \arg\min_{X\in\mathcal{X}} \int w(t)d\left[X, F(t)\right]^2 dt.$$

This implies that all the classical notions which can be identified as the weighted expectation and weighted integral can be generalized immediately to metric space, leading to the extended models. Examples in literature include the extension of Nadaraya–Watson estimator (Hein, 2009), global/local linear regression (Petersen and Müller, 2016), and the principal components in functional PCA (Dubey and Müller, 2020).

So far, many tools of common tasks for object data are still undeveloped, such as very fundamentally regression (object-to-object), auto-regressive models, that are only developed for a very few special object cases. Thus developing more object-valued tools especially with the metric-space treatments will be a very promising direction of mid-term research.

Short-term: Graph-valued data analysis. As the object closely related to the works of this thesis, graph/network itself is an important subject to work on. A critical reason is that it is adopted to represent the brain functional connectivity. This motivates statisticians to perform analysis for populations of networks, and apply the developed tools to the related neuroscience research. For example Ginestet et al. (2017) proposed the one-, two-, k-sample tests to test if several populations of graphs present the same mean level (compared with a prespecified value in one sample case). They then applied the test to study the impact of genders/ages on the brain functional connectivity.

Associating each observation of graph with a Laplacian matrix, graph-valued data also live in a metric space consisting of graph Laplacians, endowed with some matrix metric. Thus the metric-space treatments can apply. Additionally, as a convex subset of the Euclidean space of general matrices, the Laplacian space has its own properties to use in modelling. These set the development of graph-valued models at a good starting position.

Bibliography

C. C. Aggarwal, A. Bar-Noy, and S. Shamoun. On sensor selection in linked information networks. *Computer Networks*, 126:100–113, 2017. doi: 10.1016/j.comnet.2017.05.024. URL https://doi.org/10.1016/j.comnet.2017.05.024.

L. Ambrosio, N. Gigli, and G. Savaré. Gradient flows with metric and differentiable structures, and applications to the wasserstein space. *Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti Lincei. Matematica e Applicazioni*, 15(3-4):327–343, 2004.

L. Ambrosio, N. Gigli, and G. Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

A. Anis, A. Gadde, and A. Ortega. Efficient sampling set selection for bandlimited graph signals using graph spectral proxies. *IEEE Trans. Signal Processing*, 64(14): 3775–3789, 2016a. doi: 10.1109/TSP.2016.2546233. URL https://doi.org/10.1109/TSP.2016.2546233.

A. Anis, A. Gadde, and A. Ortega. Efficient sampling set selection for bandlimited graph signals using graph spectral proxies. *IEEE Transactions on Signal Processing*, 64(14):3775–3789, 2016b.

V. Apidopoulos. *Inertial Gradient-Descent algorithms for convex minimization*. PhD thesis, Université de Bordeaux, 2019.

N. N. Author. Suppressed for anonymity, 2021.

K. Baba, R. Shibata, and M. Sibuya. Partial correlation and conditional correlation as measures of conditional independence. *Australian & New Zealand Journal of Statistics*, 46(4):657–664, 2004.

F. R. Bach and M. I. Jordan. Learning graphical models for stationary time series. *IEEE transactions on signal processing*, 52(8):2189–2199, 2004.

P. Bartlett. Lecture notes in statistical learning theory, 2008.

F. Bauer. Normalized graph laplacians for directed graphs. *Linear Algebra and its Applications*, 436(11):4193–4222, 2012.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

J. Bigot. Statistical data analysis in the wasserstein space. *ESAIM: ProcS*, 68:1–19, 2020.

J. Bigot, R. Gouet, T. Klein, and A. López. Geodesic pca in the wasserstein space by convex pca. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 53 (1):1–26, 2017.

L. J. Billera, S. P. Holmes, and K. Vogtmann. Geometry of the space of phylogenetic trees. *Advances in Applied Mathematics*, 27(4):733–767, 2001.

C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

S. Bobkov and M. Ledoux. One-dimensional empirical measures, order statistics and kantorovich transport distances. *preprint*, 7127347, 2014.

A. Bolstad, B. D. Van Veen, and R. Nowak. Causal network inference via group sparse regularization. *IEEE transactions on signal processing*, 59(6):2628–2641, 2011.

E. V. Bonilla, K. Chai, and C. Williams. Multi-task gaussian process prediction. *Advances in neural information processing systems*, 20, 2007.

D. Bosq. *Linear processes in function spaces: theory and applications*, volume 149. Springer Science & Business Media, 2000.

S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

P. J. Brockwell and R. A. Davis. *Time series: theory and methods*. Springer science & business media, 2009.

A. E. Brouwer and W. H. Haemers. Distance-regular graphs. In *Spectra of Graphs*, pages 177–185. Springer, 2012.

J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

E. Cazelles, V. Seguy, J. Bigot, M. Cuturi, and N. Papadakis. Log-pca versus geodesic pca of histograms in the wasserstein space. *arXiv preprint arXiv:1708.08143*, 2017.

Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin. Training and testing low-degree polynomial data mappings via linear svm. *Journal of Machine Learning Research*, 11(Apr):1471–1490, 2010.

R. Chen, H. Xiao, and D. Yang. Autoregressive models for matrix-valued time series. *Journal of Econometrics*, 222(1):539–560, 2021a.

S. Chen and X. Chen. Weak connectedness of tensor product of digraphs. *Discrete Applied Mathematics*, 185:52–58, 2015.

S. Chen, R. Varma, A. Sandryhaila, and J. Kovacevic. Discrete signal processing on graphs: Sampling theory. *IEEE Trans. Signal Processing*, 63(24):6510–6523, 2015. doi: 10.1109/TSP.2015.2469645. URL https://doi.org/10.1109/TSP.2015.2469645.

Y. Chen, Z. Lin, and H.-G. Müller. Wasserstein regression. *Journal of the American Statistical Association*, pages 1–14, 2021b.

Z. Chen, Y. Bao, H. Li, and B. F. Spencer Jr. Lqd-rkhs-based distribution-to-distribution regression methodology for restoring the probability distributions of missing shm data. *Mechanical Systems and Signal Processing*, 121:655–674, 2019.

S. P. Chepuri and G. Leus. Graph sampling for covariance estimation. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):451–466, 2017.

A. Christmann and I. Steinwart. Support vector machines. 2008.

N. A. C. Cressie. *Statistics for spatial data*. John Wiley & Sons, New York; Chichester, 1993.

M. Crovella and E. Kolaczyk. Graph wavelets for spatial traffic analysis. In *Proceedings of IEEE Infocom*, Apr. 2003. URL http://www.cs.bu.edu/faculty/crovella/paper-archive/infocom03-graph-wavelets.pdf.

A. P. Dawid. Some matrix-variate distribution theory: notational considerations and a bayesian application. *Biometrika*, 68(1):265–274, 1981.

M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

M. Di Marzio, A. Panzera, and C. C. Taylor. Nonparametric regression for spherical data. *Journal of the American Statistical Association*, 109(506):748–763, 2014.

X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst. Learning laplacian matrix in smooth graph signal representations. *IEEE Transactions on Signal Processing*, 64(23):6160–6173, 2016.

X. Dong, D. Thanou, M. Rabbat, and P. Frossard. Learning graphs from data: A signal representation perspective. *IEEE Signal Processing Magazine*, 36(3):44–63, 2019.

P. Dubey and H.-G. Müller. Functional models for time-varying random objects. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(2):275–327, 2020.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, 2nd edition, 2000.

B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *Annals of statistics*, 32(2):407–499, 2004.

J. Fan and I. Gijbels. *Local polynomial modelling and its applications*. Routledge, 2018.

S. Fang, Q. Zhang, G. Meng, S. Xiang, and C. Pan. Gstnet: Global spatial-temporal network for traffic flow prediction. In *IJCAI*, pages 2286–2293, 2019.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

P. Garrigues and L. Ghaoui. An homotopy algorithm for the lasso with online observations. *Advances in neural information processing systems*, 21:489–496, 2008.

L. Ghodrati and V. M. Panaretos. Distribution-on-distribution regression via optimal transport maps. *Biometrika*, 01 2022. asac005.

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

K. Greenewald, S. Zhou, and A. Hero III. Tensor graphical lasso (teralasso). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(5):901–931, 2019.

A. K. Gupta and D. K. Nagar. *Matrix variate distributions*. Chapman and Hall/CRC, 2018.

R. H. Hammack, W. Imrich, S. Klavžar, W. Imrich, and S. Klavžar. *Handbook of product graphs*, volume 2. CRC press Boca Raton, 2011.

M. S. Handcock and J. R. Wallis. An approach to statistical spatial-temporal modeling of meteorological fields. *Journal of the American Statistical Association*, 89(426): 368–378, 1994.

T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

M. Hein. Robust nonparametric regression with metric-space valued output. *Advances in neural information processing systems*, 22, 2009.

L. Helmut. *New introduction to multiple time series analysis*. Springer Berlin Heidelberg, 2005.

M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

W. Huang, T. A. W. Bolton, J. D. Medaglia, D. S. Bassett, A. Ribeiro, and D. V. D. Ville. A graph signal processing perspective on functional brain imaging. *Proceedings of the IEEE*, 106(5):868–885, 2018. doi: 10.1109/JPROC.2018.2798928. URL https://doi.org/10.1109/JPROC.2018.2798928.

W. Imrich and I. Peterin. Cartesian products of directed graphs with loops. *Discrete Mathematics*, 341(5):1336–1343, 2018.

Y. Jiang. Wasserstein multivariate auto-regressive models for modeling distributional time series and its application in graph learning. *stat*, 1050:12, 2022.

Y. Jiang, J. Bigot, and S. Maabout. Sensor selection on graphs via data-driven node sub-sampling in network time series. *arXiv preprint arXiv:2004.11815*, 2020.

Y. Jiang, J. Bigot, and S. Maabout. Online graph topology learning from matrix-valued time series. *arXiv preprint arXiv:2107.08020*, 2021.

S. Joshi and S. P. Boyd. Sensor selection via convex optimization. *IEEE Trans. Signal Processing*, 57(2):451–462, 2009. doi: 10.1109/TSP.2008.2007095. URL https://doi.org/10.1109/TSP.2008.2007095.

H. Kadri, E. Duflos, P. Preux, S. Canu, A. Rakotomamonjy, and J. Audiffren. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17(20):1–54, 2016.

A. Kalaitzis, J. Lafferty, N. D. Lawrence, and S. Zhou. The bigraphical lasso. In *International Conference on Machine Learning*, pages 1229–1237. PMLR, 2013.

V. Kalofolias. How to learn a graph from smooth signals. In *Artificial Intelligence and Statistics*, pages 920–929. PMLR, 2016.

M. J. Kearns. *Computational Complexity of Machine Learning*. PhD thesis, Department of Computer Science, Harvard University, 1989.

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

P. Kokoszka, H. Miao, A. Petersen, and H. L. Shang. Forecasting of density functions with an application to cross-sectional and intraday returns. *International Journal of Forecasting*, 35(4):1304–1317, 2019.

E. D. Kolaczyk. *Statistical Analysis of Network Data: Methods and Models*. Springer Publishing Company, Incorporated, 1st edition, 2009. ISBN 038788145X, 9780387881454.

A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, June 2008. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1390681.1390689.

B. Lake and J. Tenenbaum. Discovering structure by learning sparse graphs. 2010.

P. Langley. Crafting papers on machine learning. In P. Langley, editor, *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pages 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

H. Liu, H. Kou, C. Yan, and L. Qi. Link prediction in paper citation network to construct paper correlation graph. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):1–12, 2019.

Y. Liu, A. Niculescu-Mizil, A. C. Lozano, and Y. Lu. Learning temporal causal graphs for relational time-series analysis. In *ICML*. Citeseer, 2010.

H. Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

X. Lyu, W. W. Sun, Z. Wang, H. Liu, J. Yang, and G. Cheng. Tensor graphical model: Non-convex optimization and statistical inference. *IEEE transactions on pattern analysis and machine intelligence*, 42(8):2024–2037, 2019.

J. Mairal and J.-P. Vert. Machine learning with kernel methods, 2018.

D. M. Malioutov, M. Cetin, and A. S. Willsky. Homotopy continuation for sparse signal representation. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v–733. IEEE, 2005.

S. Mazzuco and B. Scarpa. Fitting age-specific fertility rates by a flexible generalized skew normal probability density function. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 178(1):187–203, 2015.

J. Mei and J. M. Moura. Signal processing on graphs: Causal modeling of unstructured data. *IEEE Transactions on Signal Processing*, 65(8):2077–2092, 2016.

M. Meilă and T. Jaakkola. Tractable bayesian learning of tree belief networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 380–388. Morgan Kaufmann Publishers Inc., 2000.

N. Meinshausen, P. Bühlmann, et al. High-dimensional graphs and variable selection with the lasso. *Annals of statistics*, 34(3):1436–1462, 2006.

I. Melnyk and A. Banerjee. Estimating structured vector autoregressive models. In *International Conference on Machine Learning*, pages 830–839. PMLR, 2016.

C. A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural computation*, 17(1):177–204, 2005.

R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors. *Machine Learning: An Artificial Intelligence Approach, Vol. I.* Tioga, Palo Alto, CA, 1983.

A. Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

T. M. Mitchell. The need for biases in learning generalizations. Technical report, Computer Science Department, Rutgers University, New Brunswick, MA, 1980.

R. P. Monti, C. Anagnostopoulos, and G. Montana. Adaptive regularization for lasso models in the context of nonstationary data streams. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 11(5):237–247, 2018.

K. Neusser. *Time series econometrics*. Springer, 2016.

A. Newell and P. S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. In J. R. Anderson, editor, *Cognitive Skills and Their Acquisition*, chapter 1, pages 1–51. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1981.

A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018. URL http://dblp.uni-trier.de/db/journals/pieee/pieee106.html#OrtegaFKMV18.

M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA journal of numerical analysis*, 20(3):389–403, 2000.

H. Otneim and D. Tjøstheim. The Locally Gaussian Partial Correlation. *arXiv e-prints*, art. arXiv:1909.09681, Sep 2019.

V. M. Panaretos and Y. Zemel. Amplitude and phase variation of point processes. *The Annals of Statistics*, 44(2):771–812, 2016.

V. M. Panaretos and Y. Zemel. *An invitation to statistics in Wasserstein space*. Springer Nature, 2020.

N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

S. Park, K. Shedden, and S. Zhou. Non-separable covariance models for spatio-temporal data, with applications to neural encoding analysis. *arXiv preprint arXiv:1705.05265*, 2017.

N. Perraudin and P. Vandergheynst. Stationary signal processing on graphs. *IEEE Transactions on Signal Processing*, 65(13):3462–3477, 2017.

N. Perraudin, B. Ricaud, D. I Shuman, and P. Vandergheynst. Global and local uncertainty principles for signals on graphs. *APSIPA Trans. Signal Inf. Process.*, Apr. 2018.

A. Petersen and H.-G. Müller. Functional data analysis for density functions by transformation to a hilbert space. *The Annals of Statistics*, 44(1):183–218, 2016.

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

A. Petersen and H.-G. Müller. Wasserstein covariance for multiple random densities. *Biometrika*, 106(2):339–351, 2019b.

A. Petersen, C. Zhang, and P. Kokoszka. Modeling Probability Density Functions as Data Objects. *Econometrics and Statistics*, 21(C):159–178, 2022.

G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst. Random sampling of bandlimited signals on graphs. *Applied and Computational Harmonic Analysis*, 44 (2):446–475, 2018.

D. Romero, V. N. Ioannidis, and G. B. Giannakis. Kernel-based reconstruction of space-time functions on dynamic graphs. *IEEE Journal of Selected Topics in Signal Processing*, 11(6):856–869, 2017.

M. Rudelson and S. Zhou. Errors-in-variables models with dependent measurements. *Electronic Journal of Statistics*, 11(1):1699–1797, 2017.

A. Sakiyama, Y. Tanaka, T. Tanaka, and A. Ortega. Eigendecomposition-free sampling set selection for graph signals. *IEEE Trans. Signal Processing*, 67(10):2679–2692, 2019a. doi: 10.1109/TSP.2019.2908129. URL https://doi.org/10.1109/TSP.2019.2908129.

A. Sakiyama, Y. Tanaka, T. Tanaka, and A. Ortega. Eigendecomposition-free sampling set selection for graph signals. *IEEE Transactions on Signal Processing*, 67(10): 2679–2692, 2019b.

A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):211–229, 1959.

A. Sandryhaila and J. M. Moura. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Processing Magazine*, 31(5):80–90, 2014a.

A. Sandryhaila and J. M. F. Moura. Discrete signal processing on graphs: Frequency analysis. *Trans. Sig. Proc.*, 62(12):3042–3054, June 2014b. ISSN 1053-587X. doi: 10.1109/TSP.2014.2321121. URL https://doi.org/10.1109/TSP.2014.2321121.

S. Sarica, J. Luo, and K. L. Wood. Technet: Technology semantic network based on patent data. *Expert Systems with Applications*, 142:112995, 2020.

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*, pages 362–373. Springer, 2018.

H. L. Shang and S. Haberman. Forecasting age distribution of death counts: An application to annuity pricing. *Annals of Actuarial Science*, 14(1):150–169, 2020.

J. R. Shewchuk et al. An introduction to the conjugate gradient method without the agonizing pain, 1994.

M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 14(2):165–170, 1987.

D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.*, 30(3): 83–98, 2013. URL http://dblp.uni-trier.de/db/journals/spm/spm30.html#ShumanNFOV13.

J. Songsiri and L. Vandenberghe. Topology selection in graphical models of autoregressive processes. *The Journal of Machine Learning Research*, 11:2671–2705, 2010.

I. Spinelli, S. Scardapane, and A. Uncini. Missing data imputation with adversarially-trained graph convolutional networks. *arXiv preprint arXiv:1905.01907*, 2019.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

K.-T. Sturm, T. Coulhon, A. Grigoryan, et al. Probability measures on metric spaces of nonpositive. *Heat Kernels and Analysis on Manifolds, Graphs, and Metric Spaces: Lecture Notes from a Quarter Program on Heat Kernels, Random Walks, and Analysis on Manifolds and Graphs: April 16-July 13, 2002, Emile Borel Centre of the Henri Poincaré Institute, Paris, France*, 338:357, 2003.

S. Tabassum, F. S. Pereira, S. Fernandes, and J. Gama. Social network analysis: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(5):e1256, 2018.

J. Thai, C. Wu, A. Pozdnukhov, and A. Bayen. Projected sub-gradient with ℓ_1 or simplex constraints via isotonic regression. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 2031–2036. IEEE, 2015.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

C. Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

J.-L. Wang, J.-M. Chiou, and H.-G. Müller. Functional data analysis. *Annu. Rev. Statist*, 1:41, 2015.

Y. Wang, B. Jang, and A. Hero. The sylvester graphical lasso (syglasso). In *International Conference on Artificial Intelligence and Statistics*, pages 1943–1953. PMLR, 2020.

C. W. Wu. On rayleigh–ritz ratios of a generalized laplacian matrix of directed graphs. *Linear algebra and its applications*, 402:207–227, 2005.

W. B. Wu and X. Shao. Limit theorems for iterated random functions. *Journal of Applied Probability*, 41(2):425–436, 2004.

Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

M. Xu, Y. Yang, M. Han, T. Qiu, and H. Lin. Spatio-temporal interpolated echo state network for meteorological series prediction. *IEEE transactions on neural networks and learning systems*, 30(6):1621–1634, 2018.

S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

J. Yin and H. Li. Model selection and estimation in the matrix normal graphical model. *Journal of multivariate analysis*, 107:119–140, 2012.

M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

B. Zaman, L. M. L. Ramos, D. Romero, and B. Beferull-Lozano. Online topology identification from vector autoregressive time series. *IEEE Transactions on Signal Processing*, 69:210–225, 2020.

L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in neural information processing systems*, pages 1601–1608, 2005.

C. Zhang, P. Kokoszka, and A. Petersen. Wasserstein autoregressive models for density time series. *Journal of Time Series Analysis*, 2021.

M. Zhang, Z. Cui, M. Neumann, and Y. Chen. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

S. Zhou. Gemini: Graph estimation with matrix variate normal instances. *The Annals of Statistics*, 42(2):532–562, 2014.

X. Zhou. On the fenchel duality between strong convexity and lipschitz continuous gradient. *arXiv preprint arXiv:1803.06573*, 2018.

Y. Zhou and H.-G. Müller. Dynamic network regression. *arXiv preprint arXiv:2109.02981*, 2021.

C. Zhu and H.-G. Müller. Autoregressive optimal transport models. *arXiv preprint arXiv:2105.05439*, 2021.

C. Zhu and H.-G. Müller. Spherical autoregressive models, with application to distributional and compositional time series. *arXiv preprint arXiv:2203.12783*, 2022.