Persistent Homology on Time Series

by

Yi Zhou

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Statistical Machine Learning

Department of Mathematical and Statistical Sciences
University of Alberta

Abstract

Topology is a useful tool of mathematics studying how objects are related to one another by investigating their qualitative structural properties, such as connectivity and shape. In this thesis, we applied the method of topological data analysis (TDA) on sequence data and adopt the theory of persistent homology for time series, based on topological features computed over the persistence diagram. Aiming to analyze sequence data from diverse views, we investigate topological features (in a persistent homology perspective) of both traditional statistical tools (i.e. time series) and machine learning methods (i.e. random forest). Combining the advantages of three different ideas, we finally have a way to solve clustering (unsupervised learning) and predicting problems (supervised learning) for our two datasets respectively.

There are two main contributions in this thesis. In Chapter 2, we applied persistent homology on the cross correlation matrices and partial correlation matrices of time series, and obtain topological features from the persistence diagrams and barcodes. With this information, we generated consistent clusters and loops from our data and this solution for unsupervised learning problems of unlabeled datasets constitutes my first contribution in this thesis. The second contribution lies in considering landscape as an important covariate for supervised learning problems. In Chapter 3, we applied persistent homology on polysomnography (PSG) time series and took the integrals of landscapes

as covariates generated from time series. A random forest model is built with these covariates to predict Obstructive Apnea-Hypopnea (3% desaturation) Index of new incoming patient.

Acknowledgements

First of all, I would take this chance to express my sincerest gratitude to my supervisor Dr. Giseon Heo. She opened a new world of topological data analysis to me, which is a field full of challenges and worth exploration. I learned a lot from her. She went through every detail in my thesis and guided me with great patience. Words failed me to express my gratitude to her. Without her support, I would not have finished this thesis.

I would also like to thank Dr. Ivan Mizera, Dr. Ivor Cribben and Dr. Bei Jiang, for accepting to act as members of my defense committee.

It is such treasurable experience to work with Matthew Pietrosanu and Steven Luoma. Matthew is experienced in coding and application of persistent homology. He gave me timely help whenever I met with trouble. Steven Luoma sacrificed his weekend to go over my thesis, correcting grammar and writing mistakes in my article word to word. I am really grateful to have the chance working with them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction of the tools we used

1.1 Time Series

In this study, we adopt two correlations between time series. We used cross-correlations with different lags as well as partial-correlations. The pairwise correlations are calculated and made into correlation matrices, then transformed into dissimilarity matrices, to which persistent homology analysis could be well applied. Also a moving window-size strategy is used together with partial correlation analysis to investigate large-scale time series data problems. The idea for this part is cutting long-scale time series data into overlapping pieces to extract piecewise information between them. The consistent performance of each piece under persistent homology confirms that the method works well for our data sets.

In time series analysis, there are two major parts, namely time domain analysis and frequency domain analysis. They reveal different aspects of data

structure, and provide complementary information. Time domain reveals how the signals change over time. While frequency domain reveals how often signals lie in the frequency range. Theoretically, signals are composed of many sinusoidal signals with different frequencies (Fourier series), but more accurately it is composed of infinite sinusoidal signals (fundamental and odd harmonics frequencies). Both time domain and frequency domain analysis are meaningful in our study. Time domain refers to variation of amplitude of signals with time. For example, in our PSG datasets, one signal called Electro cardiogram (ECG) maps the heartbeat with time. The recording is done every 30 minutes, and it is a typical time domain signal. However, frequency domain records the number of times each event has occurred during total period of observation. As in ECG, a number of peaks of different types exist. For example, in one heartbeat, 6 types of peaks or variation in amplitude occurs. In frequency domain, over the entire time period of recording, the number of times each peak comes is recorded. By distinguishing different types of peaks, frequency domain analysis tells us how often each event occurs or how many key points there are during the entire time interval. Up to now, we mainly focus on time domain analysis, thus correlations between time series are considered, while frequency domain would hopefully be investigated later on.

We now consider the situation where we have a number of time series and wish to explore the relations between them. We first look at the cross-correlation. Correlation is a linear measure of similarity between two signals. Cross-correlation could be seen as a generalization of the correlation measure since it takes into account the lag of one signal relative to the other.

Cross-correlation, also known as lagged correlation, of two time series is the product-moment correlation as a function of lags, between the series, which is

particularly important to assess the relationship between two signals in time. Say we have two time series y_t and x_t, The cross-covariance function (ccf) at a particular lag could be defined:

$$C_{x,y}(k) = \frac{1}{N-1}\sum_{t=1}^{N}(x_{t-k} - \mu_x)(y_t - \mu_y)$$

Where μ_x and μ_y are the means of each time series and there are N samples in each time series. The function $C_{x,y}(k)$ is the cross-covariance function. The cross-correlation is a normalized version:

$$r_{x,y}(k) = \frac{C_{x,y}(k)}{\sqrt{C_{x,x}(0) * C_{y,y}(0)}}$$

Where we denote $C_{x,x}(0)$ and $C_{y,y}(0)$ as the variances of each signal.

We use R to calculate cross-correlations. It tries different lags to calculate cross correlations between x_t and y_t, which is helpful for identifying the specific lags of the x-variable that may be useful to predict y_t. Regarded as a correlation coefficient between two time series, one of which just happens to be shifted some number of time units, a negative value for k is a correlation between the x-variable at a time before t and the y-variable at time t while a positive lag means there is an x-variable at a time after t and the y-variable at time t . If the largest cross-correlations come at negative lags, we could suppose x lags y and if they come at positive lags, we could predict x leads y, By investigating cross-correlations, we could get information about which variable causes another. Cross-correlations tell us the lead-lag relationship between time series. While cross-correlation is asymmetric, that is

$$r_{x,y}(k) = r_{y,x}(-k)$$

As a result, the cross correlation matrix for time series is not a symmetric matrix. It requires some strategy to transform it into a symmetric matrix as we have done in this study, which would be discussed in detail in chapter 2. Each nominal price is computed as

In a multivariate time series scenario, partial correlations between time series are often used. We could hold the other series as constant and investigate the relationship between two specific series. Thus we could find the unique relationship between two series while eliminating the effects from the other series. This is the idea of partial correlation. Basically, the advantage of using partial correlation is that it allows us to estimate networks for multivariate time series. Among n time series, we say that series i and j are partially correlated or partially linked if their partial correlation is not 0. The value of the partial correlation measures the strength of the link. And all the linked time series form a network.

Partial Correlation measures linear conditional dependence between series x_t and y_t, given on all other series held as constants. We have n time series x_1, x_2 ,....., x_n , the partial correlation for x_1 and x_2 describes the behavior of the two series when x_3,.....,x_n are held fixed. The partial correlation is denoted by $p_{1,2|3,\ldots,n}$

$$p_{1,2|3,\ldots,n} = \frac{R_{1,2} - R_{1,3,\ldots,n} * R_{2,3,\ldots,n}}{\sqrt{(1 - R^2_{1,3,\ldots,n}) * (1 - R^2_{2,3,\ldots,n})}}$$

Where R represents correlation coefficient in corresponding regression model.

We calculate all the pairwise partial correlations and obtain partial correlation matrix, which represents the network of the multivariate time series.

There are two equivalent approaches to obtain a partial correlation matrix. The first approach is by taking the inversion of the covariance matrix. Let $X = (x_1, x_2, ..., x_n) \in \mathbb{R}^{n \times p}$ denoting the $n \times p$ column-centered data matrix with rows corresponding to observations and columns corresponding to variables. The standard unbiased estimate of the $p \times p$ covariance matrix S is then given as

$$S = \frac{1}{n-1} X^T X$$

If the estimate S is invertible, an unbiased estimate of the partial correlation between x_i and x_j is obtained as $p_{i,j} = -\frac{w_{i,j}}{\sqrt{w_{i,i} * w_{j,j}}}$, where we denote the inverse of estimated covariance matrix $W = S^{-1} = (w_{i,j})$.

The other way to calculate partial correlation is by fitting a linear model. Consider a linear model of correlating measurement at series j to all other series,

$$x_j = \sum k \neq j(\beta_{jk} x_k + \epsilon_k)$$

, where ϵ stands for noise

The parameters β_{jk} are estimated by minimizing the sum of squared residuals of

$$L(\beta) = \sum j = 1^p \left\| x_j - \sum_{k \neq j} (\beta_{jk} x_k) \right\|$$

in a least squares regression. If we denote the least squares estimator by β_{jk}, by which the residuals are given, the partial correlation $r_{ij} = x_j - \sum_{k \neq j} (\beta_{jk} x_k)$ is then obtained by computing the correlation between the residuals of the model fit.

This partial correlation measures the mutual information between x_i and x_i themselves alone that could not be predicted by the other observations, telling us how strongly x_i and x_i are linked or correlated. After calculating all the partial correlations and finally obtaining the partial correlation matrix, we can see the whole picture of pairwise links between multivariate time series and their network would be quite clear.

Several R packages have been developed specially for the partial correlation like "*corpcor*", *ppcor* and *parcor*. In this article, we use R package *parcor* to generate the partial correlation matrix between time series. The package *parcor* can be used for regularized estimation of partial correlation matrices based on *LASSO*.

In $p >> n$ settings, like in our PSG case where $n = 28$ and p is around 4 millions, we also use moving window or rolling window procedure to obtain partial correlation matrices of different pieces of time series. This gives us the ability to solve the problem of long scale time series analysis and look at the consistency and stability of time series over a long scale time period. The moving window size method helps us to measure persistence in time series.

When choosing a rolling window size, denoted by w, i.e., the number of consecutive observation per rolling window, the size of the rolling window will

depend on the sample size, n, and periodicity of the data. In general, we can use a short rolling window size for data collected in short intervals, and a larger size for data collected in longer intervals. Generally, longer rolling window sizes tend to yield smoother rolling window estimates than shorter sizes. Additionally, we have to decide a step length s, which is how long each step moves forwards. The requirement is $\frac{s}{w} > 0.2$, to ensure that each pair of the slices overlap. Then the entire data set is partitioned into several overlapping subsamples. The first rolling window contains observations for period 1 through w, the second rolling window contains observations for period $1 + s$ through $w + s$, and so on. Then the model can be estimated using each of the rolling window subsamples.

There could be variations on the partitions so we could choose different combinations of w and s to check the performance of their persistent homology analysis. The setting of a proper value of s (which makes all pieces perform consistently) might help us to detect periods of time series and make better clusters and predictions.

1.2 Persistent Homology

Topology is the branch of mathematics that studies how objects relate to one another for their qualitative structural properties, such as connectivity and shape. It could be applied to problems of feature detection and shape recognition in high-dimensional sequence data. Specifically we use a primary mathematical tool considered as a homology theory for point cloud data sets, persistent homology, to solve our clustering and prediction problems. The topological properties we extract from the correlation matrices of time series

are Betti numbers, i.e., the number of n-dimensional holes in the discretized data space. The tools we use include persistence landscape, persistence diagram and barcode.

Topology was the branch of mathematics created to study basic properties such as loops, voids and holes. The basis of our topological analysis lies in the theory of persistent homology, which allows the constructions of persistence diagrams. Such diagrams can be viewed as summary statistics, which capture multi-scale topological features (Fasy et al., 2014).

To start, we could generate p-simplex from point clouds or distance matrices. A p-simplex is the convex hull of p+1 geometrically independent points $V = \{v_0, v_1, ..., v_p\} in \mathbb{R}^d$, to be specific, a 0-simplex could be deemed as a dot, a 1-simplex as line segment, a 2-simplex as a solid triangle, a 3-simplex as a solid pyramid and so on. Given a p-simplex $\sigma = [v_0, v_1, ..., v_p]$, any simplex spanned by a subset of $V = \{v_0, v_1, ..., v_p\}$ is called a face of σ. And a simplicial complex K in $\mathbb{R}^d$ is defined as a collection of simplices such that the intersection of any two simplices in K is a face of each of them and at the same time, every face of a simplex in K also belongs to K. Based on a simplicial complex K, we count its *Betti* numbers. *Betti* numbers are well defined by homology group and topological theorems. Here we only focus on their intuitive meanings. By definition, *Betti* numbers of a simplicial complex count different topological features. β_0 counts the number of connected components of a complex K; β_1 counts the number of loops of a complex K; β_2 counts the number of voids of a complex K; and if the topological features come up to dimension k, β_k counts the number of k-dimensional holes of a complex K. Especially, the value of β_0 is equivalent to the number of clusters in a point cloud. This allows the approach of persistent homology to be used

for clustering problems. β_1 shows the number of loops for a complex K. Furthermore, we can figure out which points or components form the loops and the components that form the same loop could share some feature in common.

Topological features may help to identify interesting patterns in the data clustering of time series. To make these topological features clear and straightforward, we use persistence diagrams and barcodes to illustrate. A persistence diagram is one way to represent p-dimensional holes. Each point in the diagram indicates the birth (x axis) and death (y axis) of a p-dimensional hole. Because we are concentrating on the most persistent features of a data set and ignore the noise, in persistence diagrams, we look for holes that persist for a long time, or possibly for the entire filtration. Persistence diagrams allow us to study how long those features persist when a filtration parameter is varying. Points lying on or near the diagonal in a persistence diagram are associated with short-lived p-dimensional holes that appear and die quickly. Therefore, they could be considered as noise while points lying far from the diagonal represent long-lived and important topological features.

The barcode is another way to illustrate life time of p-dimensional holes. A barcode is a graphical representation of complex K as a collection of horizontal line segments in a plane. Its x axis corresponds to the filtration parameter *epsilon*. Its y axis represents homology group generators. For each dimension p we have a sequence of horizontal lines with different lengths (the bars). In barcode, the persistence interval is represented as a horizontal line associated to the p-dimensional homology generators birth and death filtration stage. Therefore, short bars in barcode could be considered as topological noise and long bars represent a class which has longer life. The long bars might be the persistent topological features we are looking for.

By persistent homology, we can extract consistent clusters from persistence diagram and barcode. The main idea is that persistent homology allows for analysis of the most prominent features of a data set through filtration. The most persistent features (p-dimensional holes), are indicated by points that lie far away from the diagonal in the persistence diagram or those that have long length in the barcode.

In order to build a predictive model, we need to refine numerical features from a persistence diagram or barcode. This could be done through persistence landscape. Persistence landscape builds landscape-like structures based on persistence diagrams or barcodes and it contains the information in a persistence diagram or barcode, which could be considered as a functional summary of p-dimensional topological features. For a p-dimensional persistence barcode with an interval (a, b), or equivalently, a birth point a and death point b in the corresponding persistence diagram, we define the piecewise landscape function $L_{(a,b)} : \mathbb{R} \to \mathbb{R}$ by

$$L_{(a,b)(t)=max\{min(t-a,b-t)\}}$$

The persistence landscape of $\{(a_i, b_i)\}_{i=1}^n$ is the set of functions $\lambda_k : \mathbb{R} \to \mathbb{R}$ defined by λ_k is the kth largest value of $\{L_{(a_i,b_i)(t)}\}_{i=1}^n$ and ${}_k(t) = 0$ for all $k > n$. The graph of persistence landscape functions would be crossover triangles and λ_k traces the kth outermost outline of these overlapping triangles. For each dimension p, after generating persistence landscape functions, we calculate the integrals for each λ_k, and sum up these integrals, that is $L_p = \sum_{k=1}^n \int \lambda_k(t)d_t$, where n is the largest order of existing landscape. We compute features based

on information from the persistence diagram over various dimensions. These would be the numerical topological features we derive from the persistent homology and are used as input for prediction.

Persistent homology analysis is well defined and could be applied to point clouds or distance (dissimilarity) matrices. In our study, we first extract correlation matrices from time series and transform them into distance matrices. Our work starts from distance matrices that are derived from our time series.

1.3 Random Forest

Random Forest is a well defined machine learning model which could be applied to high dimensional data for both supervised learning (regression and classification) and unsupervised learning (clustering). In the second part of our study, we build a random forest model to do prediction for PSG data based on the covariates and the persistent homology features we extract from the data set.

Random forest (RF) is a novel machine learning model developed based on the idea ofensemble learning. Ensemble learning is a method that generates many classifiers and aggregates their results. It works well for both classification and regression problems. Random Forest takes advantage of two powerful machine-learning techniques: bagging and random feature selection and at the same time adds an additional layer of randomness to bagging. In addition to constructing each tree using a different bootstrap sample of the data, Random Forest randomly selects a subset of features to split at each node when growing a tree, instead of using all features. This strategy works out to perform very well compared to many other classifiers like discriminant analysis,

support vector machines and neural networks. Moreover, Random Forest has the advantage of being robust against overfitting (Breiman, 2001). Moreover, when building a Random Forest model, it requires only two parameters: *ntree* (the number of trees in the forest) and *mtry* (the number of features in the subset to split at each node), and the performance of the model is usually not so sensitive to their values.

Once the random forest model is trained, we could predict new data by aggregating the predictions. For classification, take the majority of the *ntree* votes and for regression, take the average or weighted average of the *ntree* results.

To assess the prediction accuracy of the random forest model, Random Forest conducts cross-validation in parallel with the training step by using the out-of-bag (OOB) samples. In the process of training, each tree is grown using a bootstrap sample with replacement from the training data Some of the data would be not used while others will be repeated in the sample. The left out data constitute the out-of-bag sample. As out-of-bag samples are used in the tree construction, we can safely use them to estimate the prediction performance. By aggregating the OOB predictions, an estimate of the error rate of Random Forest could be obtained. Given that enough trees have been grown, the OOB estimate of error rate is quite accurate and could be used to assess the performance of RF built on our topological features. The Random Forest algorithm was implemented by the R package *randomForest* and we use this package to establish our Random Forest model.

Chapter 2

Persistent Homology on time series of well water level data

2.1 Data description

We focus on environmental time series data. Our data is retrieved from the website of Alberta government (`http://environment.alberta.ca/`). There is a vast collection of information about wells located in different areas of Alberta. The wells names, latitudes, longitudes, depths, river basins and other station details are recorded, among which, their water levels (WL, measured by daily mean water level and unit in meters) are typically time series data. There are two types of wells, namely active wells and inactive wells, as well as two types of time series, namely "All Data" (from the year 1990 to 2016) and "Recent 3 Years Data" (from the year 2014 to 2016). Additionally, the website keeps recording the water levels, keeping them up to date. We consider 46 active wells and use their near real time recent 3 years (up to May 10th, 2016) daily mean water levels as the time series to be analyzed in our study,

which consequently comes to a 1094 by 46 matrix since we have 1094 days and 46 well locations in total. The idea is to apply Topological Data Analysis to time series correlation matrices and extract consistent clusters and loops for these 46 wells. Meanwhile, we evaluate the time and space relationships between the wells by investigating their performance of persistent homology on correlation matrices.

We draw the map illustrating the locations of all the 46 wells and their corresponding codes. As some of them have very similar longitudes and latitudes, they are drawn closely in the map. The map and the table demonstrating the codes and corresponding information of each well are as below. We also illustrate the "All Data" and "Recent 3 Years Data" of well "Aden˙0100" to show how the water level in this well fluctuated during the past years. And in the last figure for this part, we illustrate the recent 3 years statistics for well "Aden˙0100", including the Day mean, Historical Day Max, Historical Day Mean and Historical Day Min.

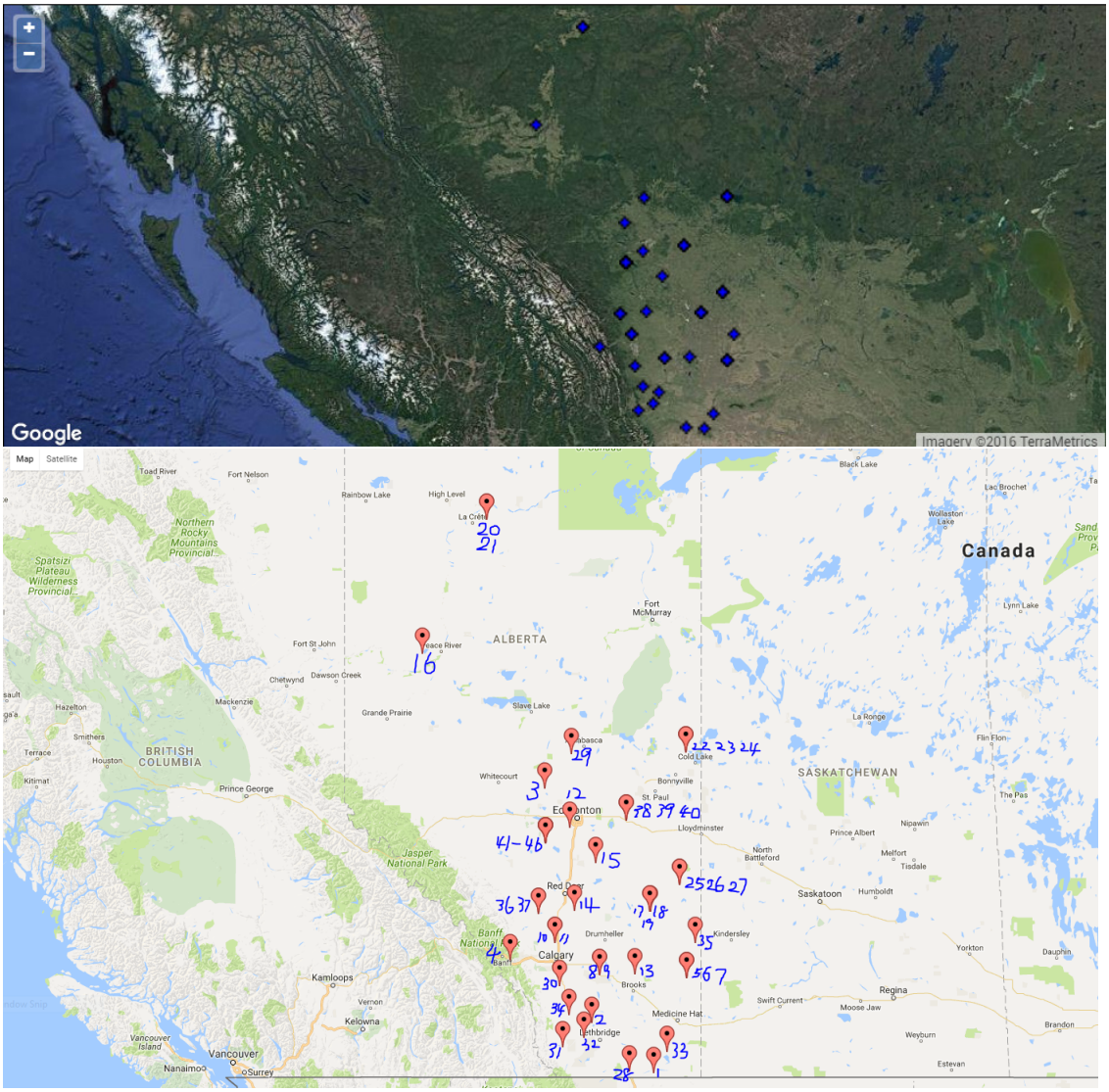


Figure 2.1: The locations of 46 wells are illustrated in the top figure and the bottom figure shows the wells with their codes. Some wells are located very close and they are separated by different codes

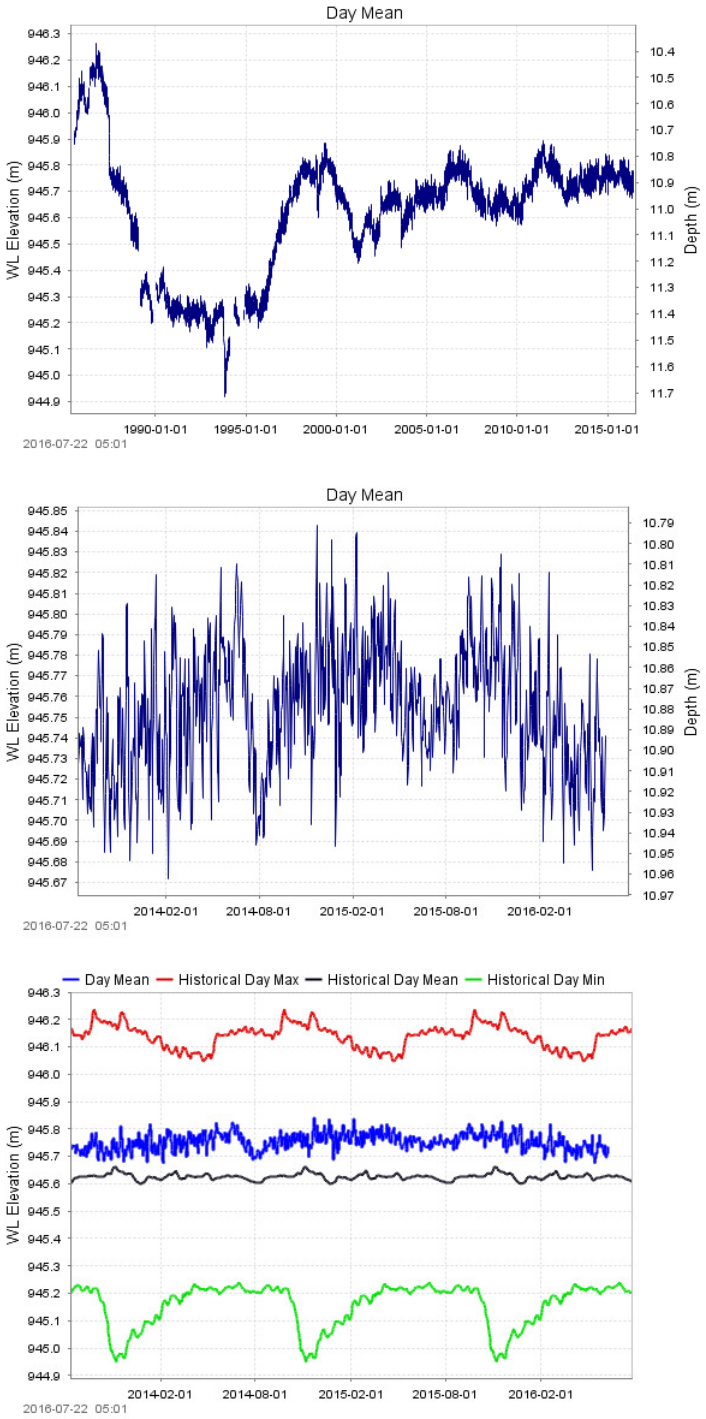


Figure 2.2: The two types of time series for well "Aden˙0100" are illustrated. top is the day means for all years from 1990 to 2015 and middle is the day means for the recent 3 years from 2014 to 2016. Bottom shows its statistics for the recent 3 years

Code	Station name	River Basin	Latitude	Longitude	Well Depth (m)
1	Aden_0100	Milk River	49.0787	-111.333	180
2	Barons 615E_0117	Oldman River	49.9929	-113.0766	19.81
3	Barrhead_0333	Athabasca River	54.0359	-114.3977	87.17
4	Canmore Tourist	Bow River	51.1073	-115.3664	85.3
5	Cavendish 2529E North_0276	Red Deer River	50.7973	-110.4181	94.5
6	Cavendish 2529E Middle_0277	Red Deer River	50.7974	-110.4181	51.80
7	Cavendish 2563E South_0281	Red Deer River	50.7972	-110.4181	21.9
8	Cluny 85-1_0218	Bow River	50.8543	-112.8428	72.50
9	Cluny 85-2 South_0219	Bow River	50.8543	-112.8428	14.30
10	Crossfield East_970	Bow River	51.4164	-114.1143	30.5
11	Crossfield West_969	Bow River	51.4164	-114.1143	57.3
12	Devon #2 North_0159	North Saskatchewan River	53.3881	-113.6911	7.62
13	Duchess 2564E_0289	Red Deer River	50.8662	-111.8694	7.62
14	Elnora #5_0129	Red Deer River	51.9668	-113.5511	13.70
15	Ferintosh Regional Landfill 85-1_0147	Battle River	52.7854	-112.9546	35.1
16	Grimshaw Kerndale_0339	Peace River	56.1891	-117.8211	53.34
17	Kirkpatrick Lake 86-1 West_0228	Sounding Creek	51.9527	-111.4301	84.7
18	Kirkpatrick Lake 86-2 Middle_0229	Sounding Creek	51.9527	-111.4431	33.5
19	Kirkpatrick Lake 86-3 East_0230	Sounding Creek	51.9527	-111.4431	11
20	La Crete 2358E South_0371	Peace River	58.2244	-116.0186	20.7
21	La Crete 2445E North_0370	Peace River	58.2244	-116.0186	83.5
22	Marie Lake Esso Seismic 2360E West_0249	Beaver River	54.6208	-110.4316	118.3
23	Marie Lake Esso Seismic 2361E Middle_0250	Beaver River	54.6209	-110.4314	95.4
24	Marie Lake Esso Seismic 2362E East_0251	Beaver River	54.6209	-110.4312	9.4
25	Metiskow 88-1_0265	Battle River	52.4212	-110.6072	128.90
26	Metiskow 88-2_0266	Battle River	52.4215	-110.6068	37.50
27	Metiskow 88-3_0267	Battle River	52.4215	-110.6068	6.04
28	Milk River 2479E_0260	Milk River	49.1153	-112.0111	25.9
29	Narrow Lake 2229E_0252	Athabasca River	54.6005	-113.6358	26.8
30	Okotoks Land Fill 2378E_0217	Bow River	50.6504	-113.9767	42.7
31	Oldman Dam Site #3 Obs 5_0263	Oldman River	49.5581	-113.8771	18.55
32	Orton 1514E_0111	Oldman River	49.7278	-113.2987	50.3
33	Pakowki 85-1_0104	South Saskatchewan River	49.4722	-110.9686	69.00
34	Pine Coulee 23D_0793	Oldman River	50.136	-113.6976	44.2
35	Sibbald 85-2_0123	Sounding Creek	51.4146	-110.1687	34.5
36	Sundre North Deep_0984	NA	51.9194	-114.5609	52.73
37	Sundre South Shallow_0983	NA	51.9194	-114.5609	28.04
38	Vegreville Enviroment Center 85-2 Middle_0165	North Saskatchewan River	53.5038	-112.1126	21.3
39	Vegreville Environment Center 85-1 East A_0164	North Saskatchewan River	53.5038	-112.1126	39.6
40	Vegreville Environment Center 85-3 West C_0166	North Saskatchewan River	53.5038	-112.1126	82.3
41	Warburg 2177E_0343	North Saskatchewan River	53.1268	-114.3613	243.90
42	Warburg 2178E_0310	North Saskatchewan River	53.1269	-114.3611	158.5
43	Warburg 2179E_0311	North Saskatchewan River	53.1269	-114.361	85.4
44	Warburg 2180E_0312	North Saskatchewan River	53.1269	-114.3613	21.3
45	Warburg 2181E_0313	North Saskatchewan River	53.1268	-114.3613	5.2
46	Warburg 2190E_0314	North Saskatchewan River	53.1269	-114.3613	64.90

Table 2.1: The table shows corresponding codes for 46 wells as well as their river basins, latitudes, longitudes and well depths.

2.2 Persistent Homology for cross correlation matrices

We now consider the time relationship for 46 wells in different locations of Edmonton. This is meaningful due to the fact that there could be underground connections among the wells. The change in water level of a certain well could reasonably result in the increase or decrease of water level in another well connected to it after a certain period. Theoretically, the cross-correlation between two time series is a function of lags, that is, the units we shift one time series left or right (in other words, before or after a certain period) to the other time series. By comparing the values of cross correlations, we take the lag corresponding to the largest value as the time period by which one series would affect (cause or delay) the other.

As the cross correlations between time series are functions of lags, for different lags, the cross correlations could be different. We demonstrate cross correlations with changing lags for 4 pairs of wells, namely Aden and Baron, Aden and Warburg, Cluny south and Elnora, Elnora and Duchess in the following figures.

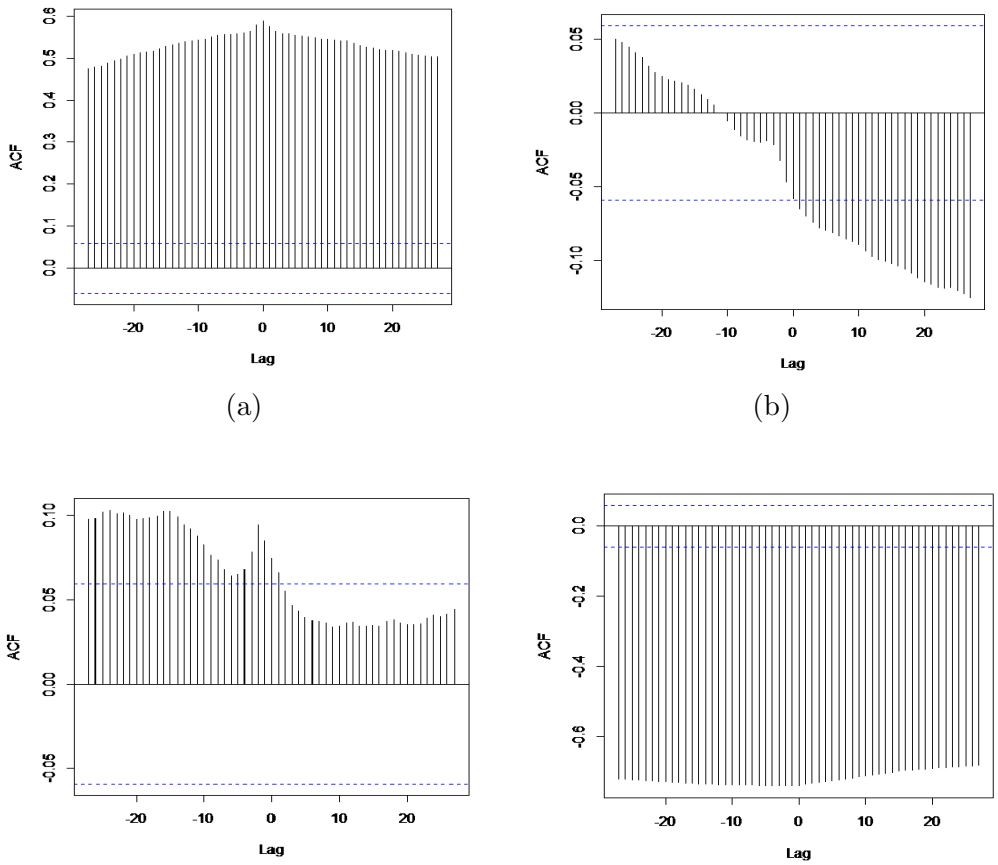


Figure 2.3: (a) shows the cross correlations for Aden and Baronto; (b) shows the cross correlations for Aden and Warburg; (c) shows the cross correlations for Cluny South and Elnora; (d) shows the cross correlations for cross correlations for Elnora and Duchess. So (a) to (d) show 4 pairs of wells with lags changing form -27 to 27 are illustrated respectively. It is obvious that with lags changing, the cross correlations change.

In this sense, we have good reason to interpret that the cross correlation matrices with different lags bear the information of time relationship among the 46 wells. By applying persistent homology on these matrices, we could obtain the topological features in respect of time relationship and study patterns of the clusters and high dimension holes.

In order to accomplish this, we proceed in a specific way. Given the 46 well level time series and a fixed lag k, calculate the cross correlation between each pair of them. This makes a 46 by 46 matrix, and the element $r_{i,j}$ represents the cross correlation between well i and well j. Noting that the sequence is of significance here, because we have pointed out that the cross correlation function is not symmetric, that is, for a certain lag k, $r_{i,j}(k) \neq r_{i,j}(-k)$.The original matrices generated by cross correlations are not symmetric matrices. As persistent homology requires distance matrices to be applied, we redefine each element in the matrix to make it symmetric, $\gamma_{i,j} = r_{i,j} + r_{j,i}, for\ i \neq j\ and\ \gamma_{i,i} = r_{i,i}$, thus making the matrix symmetric. And each row is then divided by its leading element (that is the element lying on the diagonal in this row). After this step, we obtain a matrix with its diagonal being all ones but once again asymmetric. With its upper triangle containing all the information we extract from pairwise cross correlations, we keep its upper triangle and make its lower triangle the same as its upper triangle. At this time the matrix is once again symmetric. With these steps, we obtained a symmetric matrix with its diagonal all ones. This symmetric matrix subtracted from a same size matrix containing all ones makes a symmetric transformed cross correlation matrix with its diagonal all zeroes and this could be deemed as a distance matrix we need for persistent homology. This procedure is illustrated below by an example of 3 by 3 cross correlation

matrix.

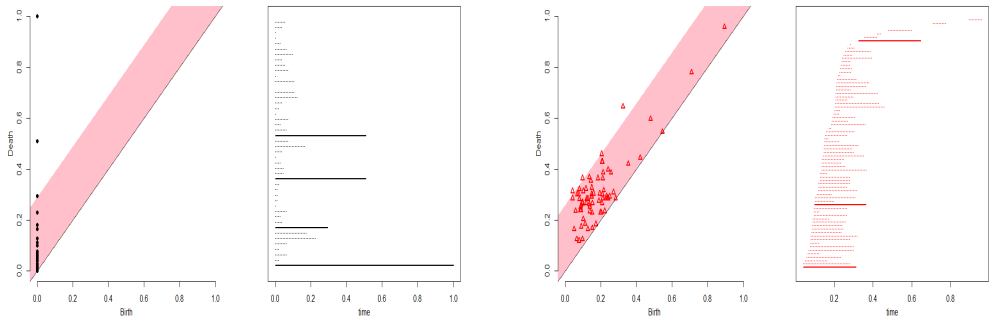
$$\begin{bmatrix} r_{11}\ r_{12}\ r_{13} \\ r_{21}\ r_{22}\ r_{23} \\ r_{31}\ r_{32}\ r_{33} \end{bmatrix} \rightarrow \begin{bmatrix} r_{11} & r_{12}+r_{21} & r_{13}+r_{31} \\ r_{21}+r_{21} & r_{22} & r_{23}+r_{32} \\ r_{31}+r_{13} & r_{32}+r_{23} & r_{33} \end{bmatrix} \rightarrow \begin{bmatrix} 1 & \frac{r_{12}+r_{21}}{r_{11}} & \frac{r_{13}+r_{31}}{r_{11}} \\ \frac{r_{12}+r_{21}}{r_{22}} & 1 & \frac{r_{23}+r_{32}}{r_{22}} \\ \frac{r_{13}+r_{31}}{r_{33}} & \frac{r_{23}+r_{32}}{r_{33}} & 1 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} 1 & \frac{r_{12}+r_{21}}{r_{11}} & \frac{r_{13}+r_{31}}{r_{11}} \\ \frac{r_{12}+r_{21}}{r_{11}} & 1 & \frac{r_{23}+r_{32}}{r_{22}} \\ \frac{r_{13}+r_{31}}{r_{11}} & \frac{r_{23}+r_{32}}{r_{22}} & 1 \end{bmatrix}$$

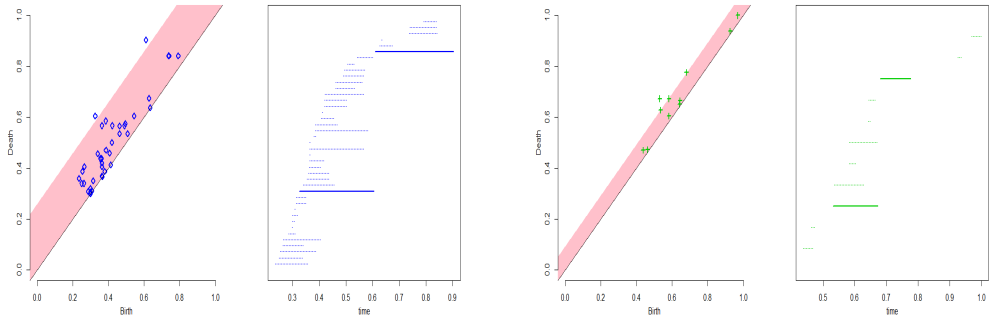
We repeat the previous steps for different lags, here we take 55 lags ranging from -27, -26,-25,,to 25, 26, 27. Thus, we have 55 distance matrices in total. Persistent homology is ready to be applied on these distance matrices. We denote them as $S_k, k = -27, -26, -25, ..., 25, 26, 27$. We prove that $S_k = S_{-k}$, which means for the lags with opposite sign and the same magnitudes the corresponding distance matrices should be identical. For each matrix, we draw a persistence diagram and barcode, and obtain dimension 0 to dimension 3 persistent homology features (we indeed have higher dimensional features for cross correlation matrices analysis and here we only consider dimension 0 to 3). In each pair of persistence diagram and barcode, we calculate the 95% confidence band. The points lying beyond the confidence band in the persistence diagram and the solid bars in the barcode indicate significant holes. For dimension 0, it indicates consistent clusters and dimension 1, it indicates consistent loops. For dimension 2 and 3, it points out consistent voids and holes

We illustrate persistent homology graphs for transformed cross correlation

matrix when taking lag=27 below and the corresponding confidence bands for each dimension. Based on these topological features, we could further investigate which wells consist each significant clusters, loops, voids and holes.



(a) (b)



(c) (d)

Figure 2.4: (a) to (d) show the persistence diagram and barcode for dimension 0, 1, 2, and 3 respectively for lag=27 cross correlation matrix respectively. For dimension 0, we see there are 4 points lying out of the confidence band and its corresponding solid bars in the barcode. And in dimension 1 diagram we see 3 points lying beyond the confidence band and corresponding solid bars. This means with 95% confidence, there are 4 consistent clusters and 3 consistent loops formed by 46 wells.

In this way, we first focus on cluster analysis based on the dimension 0 persistence diagrams and barcodes derived from the 55 cross correlation matrices with lags ranging from -27 to 27. The matrices for $lag = k$ and $lag = -k$ are identical, that is S_k and S_{-k} are the same. In consequence, the clusters generated from both barcodes would be identical too. And we summarize the cluster results for the 55 dimension 0 barcodes in the table below:

Lag	Cluster 1	Cluster 2	Cluster 3
27,21,24,5,4,3,2,1,0	well 17	other wells	none
26,16,13	well 25	other wells	none
25	well 6	well 17	other wells
23	well 25	other wells	none
22,20,18	well 17	well 25	other wells
19,6	well 8	well 17	other wells
17	well 8	well 25	other wells
15,14,8	well 8	other wells	none
12	well 6	other wells	none
11,10,9	well 7	well 8	other wells
7	well 6	well 8	other wells

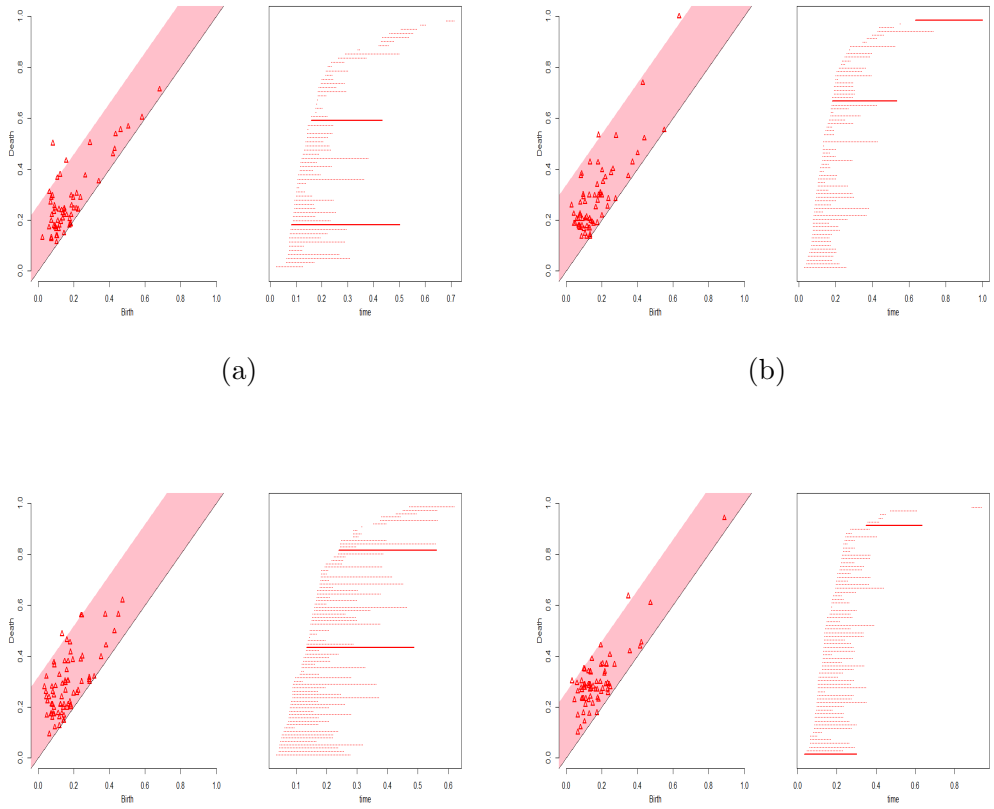
Table 2.2: The table shows the three most consistent clusters we derived from cross correlation matrices with lags ranging form 0 to 27 and which wells form each cluster.

Form this table, we see that the well No.6, No.8, No.17 and No.25 are most irregular ones and they are quite often clustered separately. Among them, the clusters formed by well No.7 against the other wells show up most often, which means this well might possess some special features compared with the others.

Secondly, we look at β_1 for each persistence diagram and barcode to analyze loops. As we have found out in the previous part, the cluster analysis would always yield the result that one well forming a certain cluster and the other wells forming a second cluster. If we figure out which wells compose a loop, this would give us more information about the wells that have the same feature in the sense of topological feature.

For each cross correlation matrix, we calculate 95% confidence band on its persistence diagram and the corresponding loops are indicated by solid bars

in the barcode. We illustrate this by the Figure 2-5 showing the persistence diagrams and barcodes for cross correlation matrix with lag=0, 7, 14 and 25 respectively and summarize the consistent loops and also the wells forming these loops in Table 2-3. As the table shows, for some lags, there is only one significant loop indicated by 95% confidence band.



(a) (b)

(c) (d)

Figure 2.5: (a) to (d) show the persistence diagrams and barcodes for dimension 1 with lag= 0, 7, 14 and 25 respectively.

For lag =0, we see that there are 2 points out of the confidence band and thus 2 solid bars in the barcode, indicating that there are 2 consistent loops formed by the wells. And refer to Table 2-3, we know that the first consistent loop is formed by well 1, 2, 5, 9, 10, 20, 29, 30, 32, 33, 36, 44 and 45. The second consistent loop is formed by well 1, 2, 5, 10, 37 and 45. From Table 2-1, we find the corresponding codes for the wells and know which wells form the consistent loops immediately. For example, with lag=0, the second consistent loop is formed by well Aden˙0100, Barons 615E˙0117, Cavendish 2529E North˙0276, Crossfield East˙970, Sundre South Shallow˙0983

and Warburg 2181E˙0313. And for lag=7, we see that there are 2 points out of the confidence band and thus 2 solid bars in the barcode, indicating that there are 2 consistent loops formed by the wells. And refer to Table 2-3, we know that the first consistent loop is formed by well 2, 5, 22, 26, 27, 28, 30, 38, 40 and 45. The second consistent loop is formed by well 1, 3, 5, 9, 11, 16, 20, 30, 33, 44 and 45. For the other lags, we adopt the same procedure and obtain most consistent loops by 95% confidence bands in the persistence diagrams or equivalently, by the solid bars in the barcodes. We summarize all the consistent loops indicated by the confidence bands in the following table.

Lag	The wells forming consistent Loop 1	The wells forming consistent Loop 2
27,26,25,24.23	5 7 22 23 26 32 33 38 40 42	2 29 36 39
22	5 11 20 30 40	none
21	1 5 10 11 28 30 34 40 41	2 29 36 39
20	5 6 7 22 23 26 27 32 33 38 40 42	2 5 14 27 30 31 38 40
19	1 2 4 10 13 14 20 29 39 43 46	none
18	1 5 10 22 28 34	none
17	1 3 5 9 16 19 22 28 44 45	1 24 43 45
16	1 3 5 7 17 25 45	1 5 22 28 43 45
15	3 11 17 21 30 33	none
14	2 6 10 14 15 17 19 20 27 34 36 46	1 13 17 24 27 31 35
13	1 5 22 28 34	1 5 11 20 24 30 33
12	1 5 11 13 15 20 24 30 33 34 45	1 5 22 28 34 43 45
11	1 5 22 28 34 41 43 45	none
10	1 5 9 22 28 34 41 43 44 45	9 11 20 24 29 30 33 36 44 45
9	1 5 9 22 28 34 45	none
8	5 11 22 24 28 30 33 41	none
7	2 5 22 26 27 28 30 38 40 45	1 3 5 9 11 16 20 30 33 44 45
6	6 10 13 14 19 20 27 37	1 6 9 19 20 45
5	1 2 5 6 9 10 13 15 20 36 43 44 45	1 2 9 13 24 27 30 33 38 40 42
4	1 5 9 11 24 30 31 33 45	1 2 9 24 27 30 33 38 40 42
3	2 6 10 14 20 22 27 29 30 33 38 40 42	2 6 10 14 20 27 29 38 40
2	2 9 22 24 27 30 33 36 38 40 42 44 45	1 2 5 9 11 14 24 30 31 32 33 38
1	1 2 5 9 14 15 29 30 32 33 36 44 45	1 2 5 9 10 34 37 44 45
0	1 2 5 9 10 20 29 30 32 33 36 44 45	1 2 5 10 37 45

Table 2.3: The table shows the most consistent loops we derived from cross correlation matrices with lags ranging form 0 to 27 and which wells form each cluster. The numbers represent the codes of corresponding wells. And for some lags, there is only one significant loop indicated by 95% confidence band.

From the summary table, we can see that the loops containing well 5, 7, 22, 23, 26, 32, 33, 38, 40 and 42, and the loop containing well 1, 5, 9, 22, 28, 34, 41, 43, 44 and 45 show up quite often and this indicates that these wells may consist significant loops. Now we have obtained 55 cross correlation matrices, and some persistent clusters and loops do show up regularly all through these 55 matrices. It confirms that some properties of the well water levels are retained by persistent homology features, not only in the cross correlation matrices throughout different lags.We need to investigate further what features these wells possess in common.

2.3 Persistent Homology for partial correlation matrix

We investigate the relationship for the 46 wells located in diverse areas of Alberta by applying persistent homology to the partial correlation matrix. Besides the association in time, the water levels in different wells are also affected significantly by their spatial connections. In order to figure out spatial relationship, we use partial correlations between time series and partial correlation matrix.

Partial correlation is the measure of association between two variables, while controlling the effect of the additional variables. Partial correlations can be used in many cases that assess for relationship, as long as we have the need to decide how two variables are related given the other variables are held as constants.

By means of partial correlation, we deem the 46 wells as a network. Each

well is a node of a network and the partial correlation estimates their pairwise association. Here, each well could be treated as a node in the network. Well i is presented with a vector $x_i, (i = 1, 2, 3, ..., 46)$ of n observed attributes, $x_i = [x_{i1}, x_{i2}, ..., x_{in}]$ and $n = 1094$ in this case because for each well, we have obtained 1094 time points for recent 3 years. And the partial correlation denoted as p_{ij} measures the level of association or interaction between two such nodes i and j.

If we use partial correlation for characterizing pairwise spatial association between wells, it is required that a technique is used to organize these correlations together while also still being feasible to apply persistent homology to. It is evident that partial correlation matrices, when estimated with appropriate regularization, could provide a useful characterization of spatial association or connectivity (Eugene Duff et al., 2013) between different locations.

Given a set of m random variables, the partial correlation matrix is a symmetric matrix in which each off-diagonal element is the correlation coefficient between a pair of variables after ruling out (conditioning under normality) the contributions to the pairwise correlation of all other variables included in the dataset. In our case, the partial correlation between any two wells partials out the effects of the 44 other wells in our network.

since we have obtained a matrix that contains the spatial information of the 46 wells, which is a 46 by 46 partial correlation matrix. This is a symmetric matrix with the diagonal being all ones. We take the absolute value of this matrix then subtracted by a same size (46 by 46) matrix containing all ones. Thus we transform the partial correlation matrix into a dissimilarity matrix and therefore, persistent homology could be applied to this matrix.

Still, we obtain dimension 0, 1, 2 and 3 persistence diagram and barcode

for this single partial correlation matrix and based on the pattern of bars in dimension 0 and dimension 1 barcodes, study most persistent clusters and loops. Ultimately, the goal is to figure out the wells that form the corresponding consistent clusters and loops.

We illustrate persistence diagram and barcode generated by partial correlation matrix for the 46 wells in Figure 2.6 and the 95% confidence bands are indicated by pink bands respectively. For dimension 0, there are two significant points beyond the confidence bands showing that two consistent clusters are retained. In dimension 1 persistence diagram, there are five consistent loops. And in dimension 2 and 3 persistence diagrams, there are two consistent voids and two consistent 3-dimension holes formed by the wells. Specifically, we focus on dimension 1 topological features. We have investigated the loops further and figured out which wells form these consistent loops. We illustrate the results in Figure 2.6 to Figure 2.8. Figure 2.6 show the consistent clusters, loops, voids and 3-dimension holes indicating by 95% confidence bands and solid bars. For dimension 1, we figure out the wells that form the consistent loops and connect them on the map in Figure 2.7, as well as on the Multi-dimensional Scaling 2-D plots in Figure 2.8. And the wells forming the five consistent loops in dimension 1, forming the two consistent voids in dimension 2, and forming the two consistent 3-D holes in dimension 3 are summarized respectively in Table 2.4 to Table 2.6

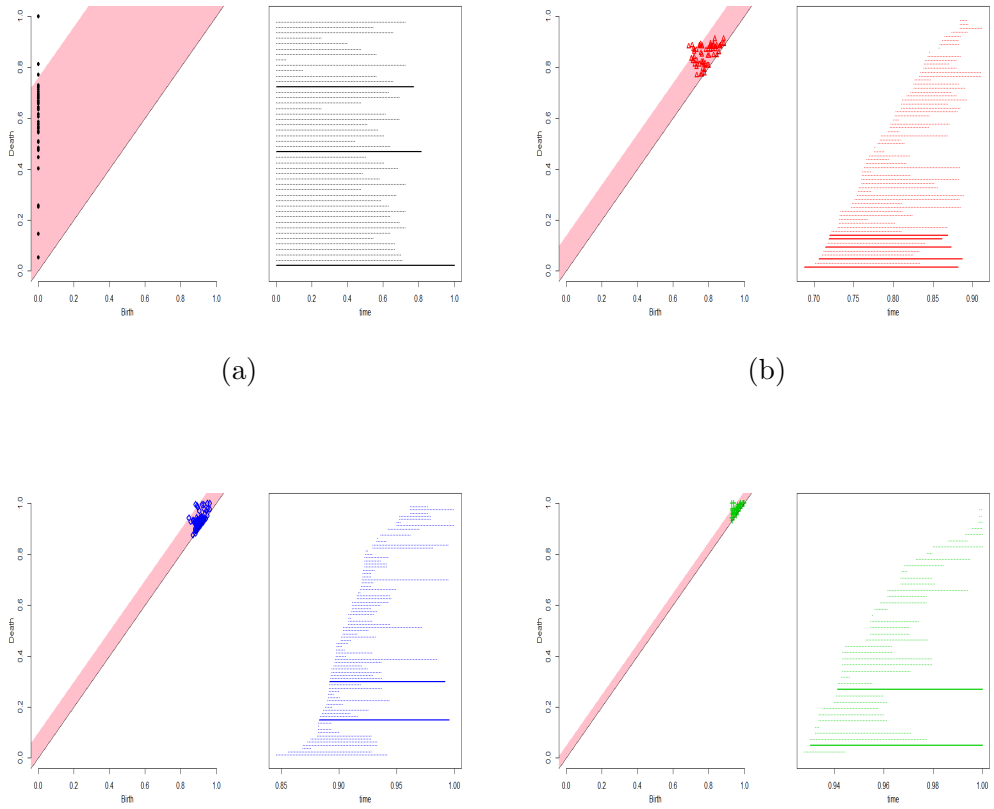
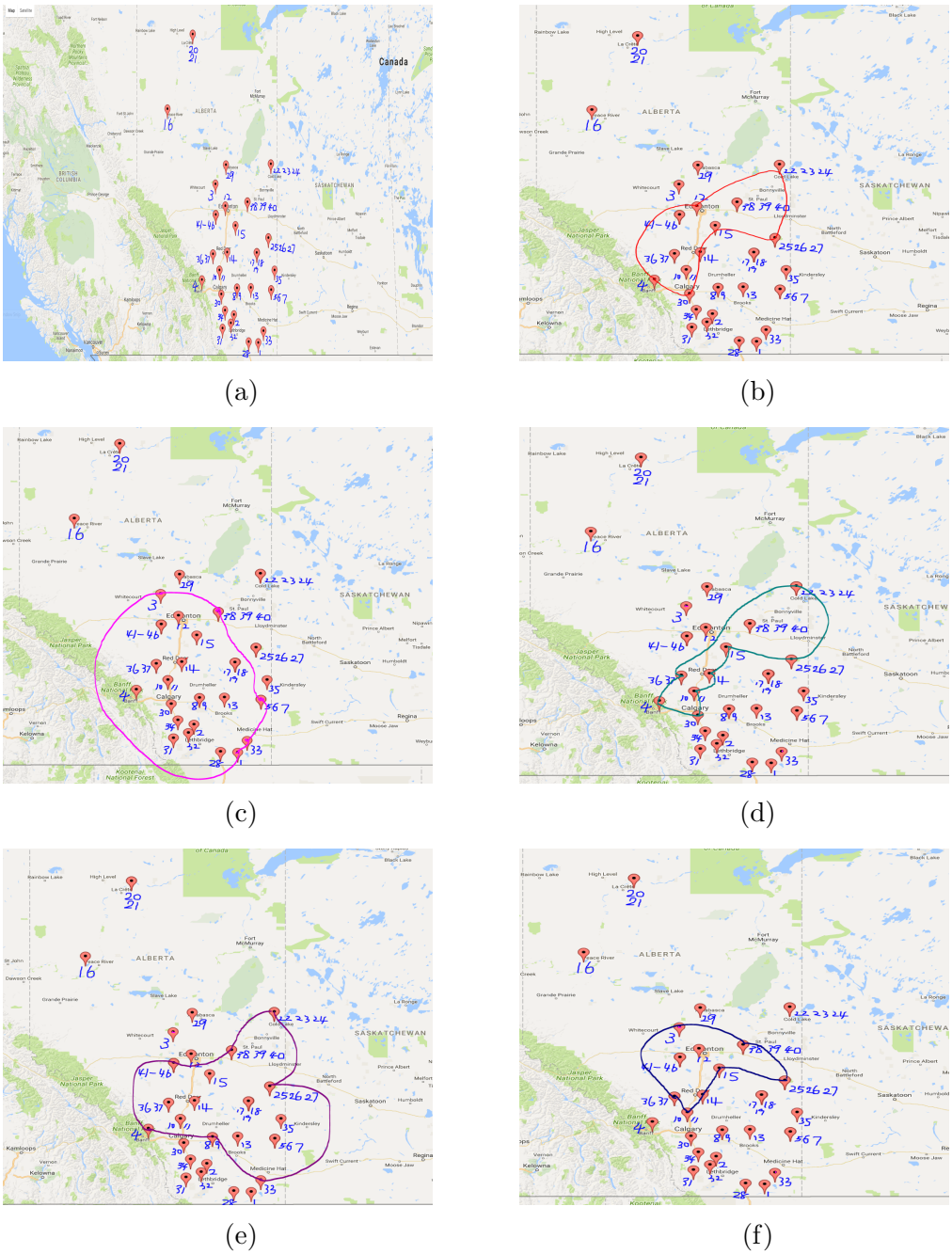


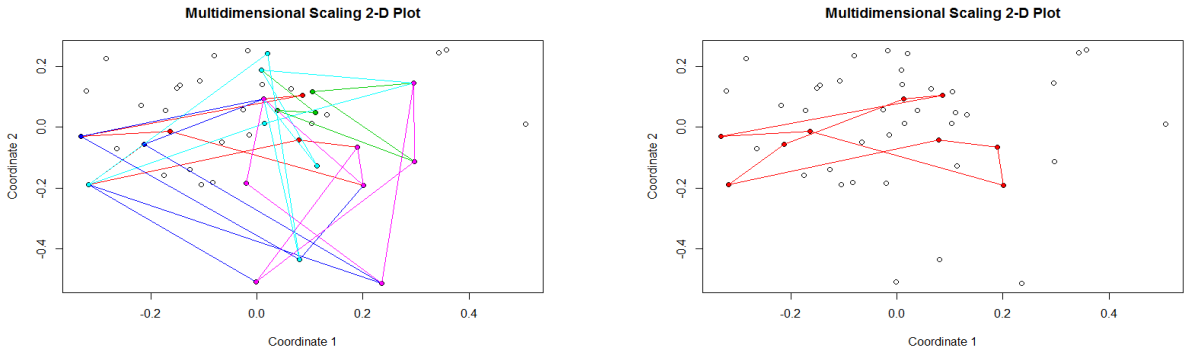
Figure 2.6: (a) to Bottom (d) show the persistence diagrams (with 95% confidence bands indicating significant features) and barcodes (with solid bars indicating significant features) for dimension 0, 1, 2 and 3 derived from partial correlation matrix of 46 wells respectively.



(a) (b) (c) (d) (e) (f)

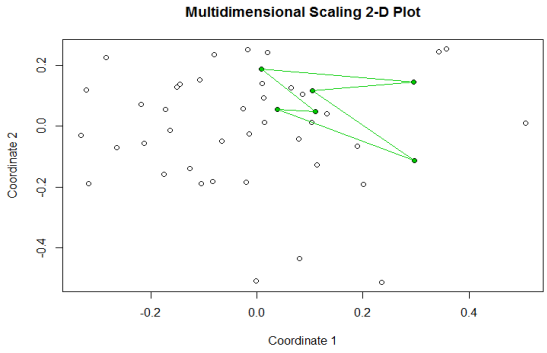
Figure 2.7: (a) shows the locations of all the 46 wells in Alberta. (b) to (f) show the wells forming the five consistent loops and how they are connected in the map respectively.

There are five consistent loops indicating by 95% confidence bands in dimension 1 persistence diagram and barcode. After we figured them out and connected them on the map, we found that some wells lying far away from each other might form a consistent loop. We need to study further what these wells have in common and why they are connected and retained in the same loop by their topological features.

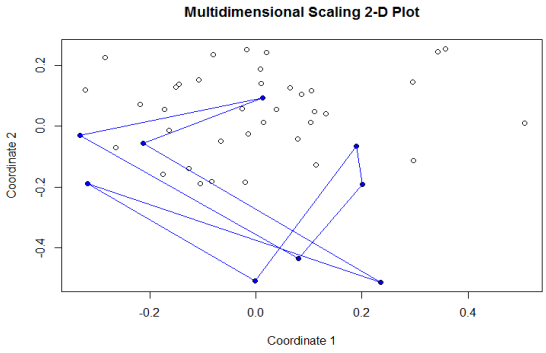


(a)

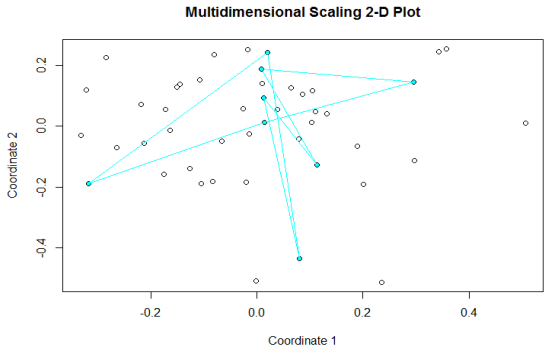
(b)



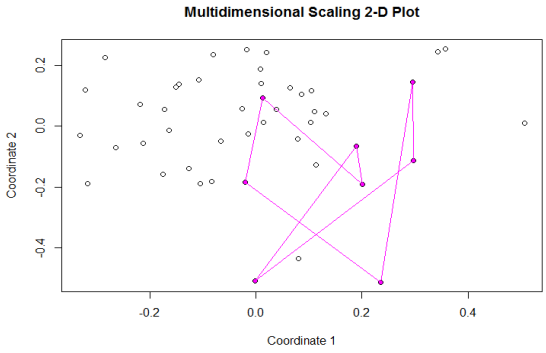
(c)



(d)



(e)



(f)

Figure 2.8: (a) shows the Multidimensional Scaling 2-D plots of all the 46 wells in the city of Edmonton. (b) to (f) connect the wells forming corresponding first, second, third, forth and fifth consistent loops respectively.

(b) connect well 4, 12, 14, 15, 23, 25, 26, 27 and 30, illustrating these 9 wells forming the first consistent loops and these 9 wells compose the largest bar in Figure 2.6 (b); (c) connect well 1, 3, 5, 6, 33 and 40, with these 5 wells forming the second consistent loops and they compose the second largest bar in Figure 2.6 (b). With the same idea, we connect the wells forming the other 3 consistent loops and these wells compose the other solid bars in Figure 2.6 (b). In total, we have five consistent loops and we figure out which wells form these loops.

Loops	Wells
1	4 12 14 15 23 25 26 27 30
2	1 3 5 6 33 40
3	4 10 14 15 24 25 27 30 37
4	4 9 24 27 29 33 40 44
5	3 10 14 15 27 36 37 40

Table 2.4: The table shows the five most consistent loops we derived from partial correlation matrix with dimension 1 persistence diagram and barcode, as well as the codes of the wells that form these loops.

Voids	Wells
1	1 2 4 7 12 13 23 25 26 30 32 33 40 41
2	3 5 10 12 14 15 20 24 26 30 36 37 40 44

Table 2.5: The table shows the two most consistent voids we derived from partial correlation matrix with dimension 1 persistence diagram and barcode, as well as the codes of the wells that form these voids.

Holes	Wells
1	12 14 15 20 23 25 26 30 40
2	1 2 3 4 5 7 8 11 13 14 25 27 29 32 43 44 46

Table 2.6: The table shows the two most consistent 3-D holes we derived from partial correlation matrix with dimension 1 persistence diagram and barcode, as well as the codes of the wells that form these holes.

At the end of this section, we desire to compare the performance of persistent homology on cross correlation matrices with diverse lags and the single partial correlation matrix. We hope to know at which lag of cross correlation matrix, the diagram and barcode are similar to those of the partial correlation matrix. A quick answer to this can be found by calculating Wasserstein distance between persistence diagrams for each dimension (dimension 0,1,2 and 3 here). The result is listed in Table 2.7

lag	dim=0	dim=1	dim=2	dim=3
27	13.09035	6.612717	2.743431	0.636713
26	13.04896	6.849057	2.800447	0.63978
25	12.86476	6.480896	2.82712	0.629331
24	13.05016	6.528012	2.862274	0.692991
23	12.68969	6.22202	2.948773	0.647767
22	13.10129	6.269431	2.914874	0.694965
21	13.32851	6.202196	2.964283	0.715616
20	12.98606	6.200113	3.220152	0.707556
19	13.16184	6.084648	3.051039	0.725045
18	13.00858	6.189613	3.089883	0.746044
17	13.26052	6.1835	3.197274	0.753645
16	13.22754	6.106437	3.206043	0.719598
15	13.57113	7.046607	3.145191	0.687796
14	13.63367	6.7975	3.270187	0.719512
13	13.52856	6.260584	3.365226	0.766263
12	13.27963	6.230544	3.096397	0.812083
11	13.57203	5.957931	3.073575	0.808883
10	13.51688	5.905016	2.968953	0.791809
9	13.5742	5.896544	2.979896	0.795903
8	13.5345	6.078822	2.989841	0.782666
7	13.0645	5.805182	2.860508	0.76873
6	13.45359	6.089922	2.783338	0.774376
5	13.37768	5.89724	2.741202	0.784516
4	13.43207	5.782444	2.645278	0.905936
3	13.36513	5.429051	2.616063	0.866756
2	13.39271	5.426729	2.691784	0.835533
1	13.34924	5.271626	2.375027	0.81925
0	13.39258	5.044525	2.329506	0.730845

Table 2.7: The table shows the Wasserstein distance between persistence diagrams of partial correlation matrix and cross correlation matrices with lags changing from 0 to 27, for each dimension (dimension 0,1,2 and 3). It compares how close two persistence diagrams are. The smallest value of Wasserstein distance means the two diagrams are very similar, thus the topological features derived from these two diagrams would be similar too. For dimension 0 , the partial correlation matrix has the smallest Wasserstein distance with cross correlation matrix when lag = 23; And for dimension 1, 2 and 3, the cross correlation matrices with lag = 0, 0 and 25 have the smallest Wasserstein distance with partial correlation matrix respectively.

Concluding from the above table, for dimension 0, the cross correlation matrix with lag = 23 has the smallest Wasserstein distance with partial correlation matrix and for dimension 1, the cross correlation matrix with lag = 0 has the smallest Wasserstein distance with partial correlation matrix. For dimension 2, the cross correlation matrix with lag = 0 has the smallest Wasserstein distance with partial correlation matrix. For dimension 3, the cross correlation matrix with lag = 25 has the smallest Wasserstein distance with partial correlation matrix.

2.4 Persistent homology for partial correlation matrices generated by moving window method

In the previous section, we include partial correlation matrices into our work to study the spatial connection between wells in different locations. Desirably, it turns out to be one matrix which aggregates the useful information. However, what if we generate several partial correlation matrices and compare the results? Moving window analysis of time series enables us to do this.

Originally, moving window analysis of a time series model serves two purposes. Firstly it could assess the models stability over time, or rather, the stability of parameters in the time series. A common time-series model assumption is that the coefficients are constant with respect to time. Checking for instability equates to examining whether the coefficients are time-invariant. Secondly, this method is used to assess the predictive performance of the time series model, or rather, the forecast accuracy of the model. In other words, application of moving window method could measure the persistence in a time

series.

Inspired by this, we tried moving window method in the end of this section mainly to achieve two goals: 1. For large scale time series, we could cut the series into overlapping parts and apply persistent homology to each piece, analyzing whether the results yielded from different pieces are consistent; 2. Generate several partial correlation matrices from 46 locations and take it as a series of matrices combining both time and spatial information between the wells and analyze their persistent homology performance.

Two major steps of moving window method is to decide window size, which is denoted by w and step length, which is denoted by s.The choice of window size involves a balance between two opposing factors. A shorter window implies a smaller data set on which to perform the estimations. A longer window implies an increase in the chance that the data-generating process has changed over the time period covered by the window, so that the oldest data are no longer representative of the system's current behavior. Most people follow a rule for choosing window size and step length, in practice: the step length to window size ratio is larger than 0.2 . If the window size is too large and the step length is too small, then there would be only a few different points in each piece, meaning that the partial correlation matrices in successive pieces would not change much.

Since we have 1094 time points in each of the 46 well water level series, it is decided that window size should equal 94 and step length should equal 40. In this way, we segment multivariate time series of our well water level data into 26 overlapping pieces or blocks with equal sizes (w=96). Also, we calculate the partial correlation matrices for each block, transform them into dissimilarity matrices using the same procedures in part 2.3 and analyze their

persistent homology features by persistence diagrams and barcodes.

We obtained 26 partial correlation matrices and their corresponding persistence diagrams and barcodes. With 95% confidence bands on the dimension 1 persistence diagrams, we figure out the significant clusters for each partial correlation matrix and summarize the results in Table 2.8. The persistence diagrams and barcodes for the first 4 partial correlation matrices generated by moving window method are illustrated in Figure 2.9.

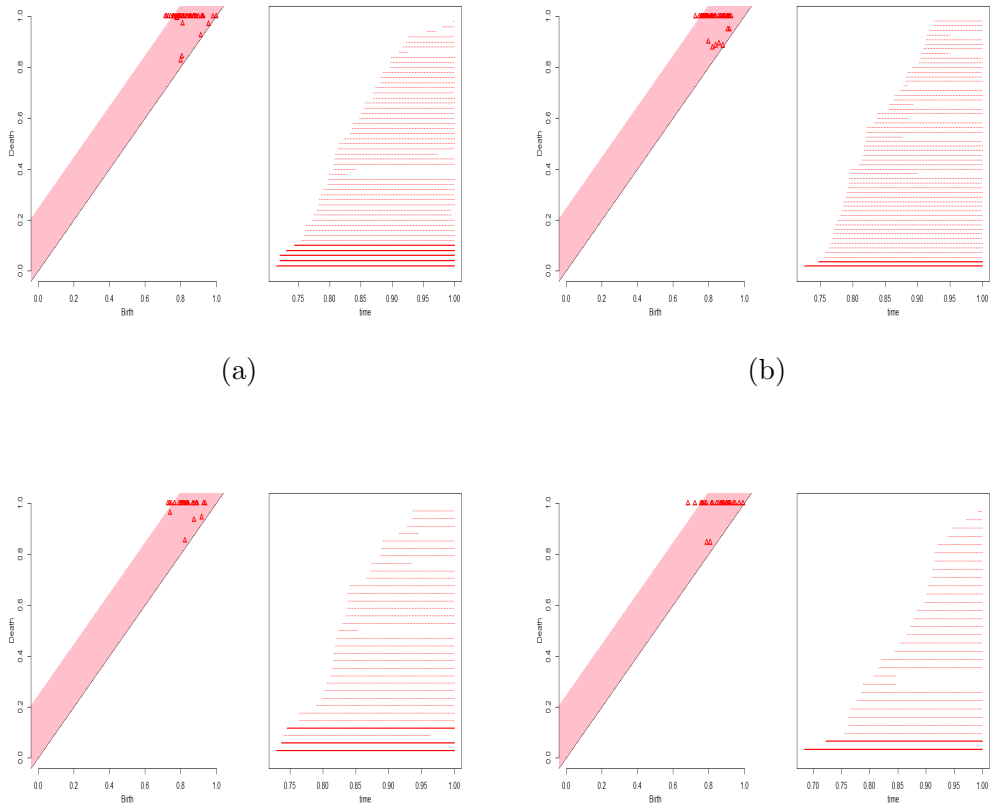


Figure 2.9: (a) to (d) show the persistence diagrams (with 95% confidence bands indicating significant features) and barcodes (with solid bars indicating significant features) for dimension 1 derived from the first 4 partial correlation matrices by moving window method.

We have 26 partial correlation matrices in total and for each matrix, we calculate the 95% confidence band (indicated by pink band in the graph) of dimension 1 persistence diagram, the points lying beyond the pink band represent significant loops. For each diagram we figure out the significant loops and the wells that form these loops. in the summary table below, we only include the most consistent loop for each diagram.

Matrix	Wells forming the most consistent loop
1,2,3,5	4 12 14 15 23 25 26 27 30
4,6,8,9,10,13,	4 10 14 15 24 25 27 30 37
7,12,13,14,15,18,20	4 9 24 27 29 33 40 44
11,16,19,21,22,	3 4 10 14 15 27 36 37 40
23,24,25,26	4 12 15 23 25 26 27 44

Table 2.8: The table displays the wells that form the most consistent loop for each of the 26 partial correlation matrices. There are five loops that show quite often. Well 4, 10, 14, 15, 23, 25, 26 are always connected together to form a consistent loop. We need further investigate the similarity between these wells and figure out what they have in common that endow them the similar topological features.

2.5 Discussion

As a summary of this chapter, we investigate both time and space relationship between 46 wells. For time association, we consider 55 cross correlation matrices with diverse lags and work out significant clusters and loops for each lag based on the 95% confidence band. Not surprisingly, several clusters and loops show up repeatedly through different lags and thus it is reasonable to investigate them further. Meanwhile, when we take partial correlations for the 46 locations, we obtain one single matrix and this partial correlation matrix enables us to measure the spatial connection between each two pairs of wells holding others as constants. From the persistence diagram and barcode and the 95% confidence band, we also figure out the most consistent clusters, loops, void, and 3-D holes as well as which well s form these consistent topological shapes.

Additionally, we use moving window size to get several partial correlation matrices for overlapping blocks of the time series and also measure the distance between each persistence diagram derived from cross correlation matrices and

persistence diagram from partial correlation matrix. It is convincing that time and spatial connections of 46 wells water level time series data are retained in their cross correlation matrices and partial correlation matrices, furthermore, revealed by their persistent homology features in the form of persistence diagrams and barcodes. And the significant k-dimensional shapes are indicated by confidence bands in the corresponding persistence diagrams.

Chapter 3

Persistent Homology and Random Forest on time series of PSG data

3.1 Description of dataset and study goal

In Chapter 2, we combined persistent homology and time series to solve unsupervised learning problems for the well water level data (clustering). In this chapter, we will investigate clinical time series data, to be specific, polysomnography (hereafter PSG) data and apply persistent homology to time series to settle supervised learning problems (prediction). Our dataset was retrieved from the website of National Sleep Research Resource (NSRR), which has an online documentation of rich sleep research data collected in children and adults across the U.S. The dataset we used was Cleveland Children's Sleep and Health Study, one of eight datasets available on this website.

The Cleveland Children's Sleep and Health Study (CCSHS) is a large

population-based pediatric cohort on objective sleep studies. A large minority representation is included in this study. The cohort in this study is a stratified random sample of full-term (FT) and preterm (PT) children, born during the period of 1988 to 1993, identified from the birth records of 3 Cleveland area hospitals. It includes 907 children, studied at ages 8-11 years with in-home sleep studies, acoustic reflectometry, anthropometry, spirometry, blood pressure (BP), and neuropschology (NP) and behavioral assessments. CCSHS had three (3) longitudinal visits. The most recent visit, Transdisciplinary Research on Energetics and Cancer (TREC), took place between 2006 and 2010. The data we used in this study is from the third or TREC visit because this visit included full, in-lab polysomnography. In total, it has 517 records. All of the subjects have their covariates recorded while only 100 of them have in-lab polysomnography records. We use these 100 records, each with two types of features, namely 131 covariates (numerical and categorical) and 28 PSG time series.

Under each participant ID, we have their bmi (Body Mass Index), htcm (Height), bpdias (Diastolic blood pressure), bphr (Heart Rate), bpsys (Systolic blood pressure) and other covariates (131 covariates in total). Each covariate has their details illustrated via summary table as well as by graph (histogram or barchart) in the website.

Obstructive sleep apnea (OSA) is the most common type of sleep apnea and is caused by complete or partial obstructions of the upper airway. It is characterized by repetitive episodes of shallow or paused breathing during sleep, despite the effort to breathe, and is usually associated with a reduction in blood oxygen saturation. The earlier OSA is detected, the better the cure would be and also the cost would be lower, so it is of great significance for us

to find a way to detect the severity of OSA for children.

The goal of our study is to build a model based on persistent homology and machine learning methods, to predict the value of variable oahi3 (or OAHI, ie. Obstructive Apnea-Hypopnea (3% desaturation) Index), which proves to be a most important factor in detecting the severity of OSA (Susan Redline et al., 2010). The problem lies in how to incorporate PSG time series data into the format of machine learning methods and find a way to extract useful information from participants' PSG records, transform them into covariates to be used in prediction. Table 3.1 displays the distribution of OAHI for the 517 participants

Numbers of participant	Interval of OAHI (units by Events per Hour)
506	0 to 8
6	8 to 16
2	16 to 25
2	25 to 45
0	46 to 70
0	71 to 89
1	90 to 99

Table 3.1: The table demonstrates the distribution of OAHI for the 517 participants. Most patients have OAHI ranging from 0 to 8 events per hour.6 of them range from 8 to 16. Only 1 of them has extremely high OAHI compared with others, with the value lying in the interval 90 to 99.

3.2 PSG time series and sleep study

Sleep specialists tend to look at selected variables for PSG and sleep questionnaires to predict OSA severity. Polysomnography is a test used to diagnose sleep disorders. It is a multi-parametric test used in the study of sleep and as a diagnostic tool in sleep medicine. The test result is called a polysomnogram. Polysomnography records brain waves, the oxygen level in blood, heart rate and breathing, as well as eye and leg movements during the study.

Polysomnography is often recorded at a sleep disorders unit at a sleep center or in a hospital. Polysomnography is a comprehensive recording of the biophysiological changes that occur during sleep. It is usually performed at night, and in some special cases, it could also be done during the day time. The PSG monitors many body functions including brain (Electroencephalography or EEG), eye movements (Electrooculography or EOG), muscle activity or skeletal muscle activation (Electromyography or EMG) and heart rhythm (Electrocardiography or ECG) during sleep. In the 1970s, the sleep efficiency and duration, sleep stages, apnea-hypopnea index, oxygen saturation, carbon

dioxide level, sleep stage changes, spontaneous arousal index breathing functions respiratory airflow and respiratory effort indicators were added to PSG records together with peripheral pulse oximetry. Basically, Polysomnography records a lot of time series associated with human sleep and provide rich information about the quality of sleep. Each channel is a time series. Figure 3.1 shows how typical PSG data looks like. There are several channels in the PSG and each channel is a time series recorded by the units of 10 seconds. During the whole sleeping period (9.5 to 10 hours often), there are millions of time points recorded and for each participant, their PSG data would be multivariate time series with millions of time points. And this figure is from the NSRR website.

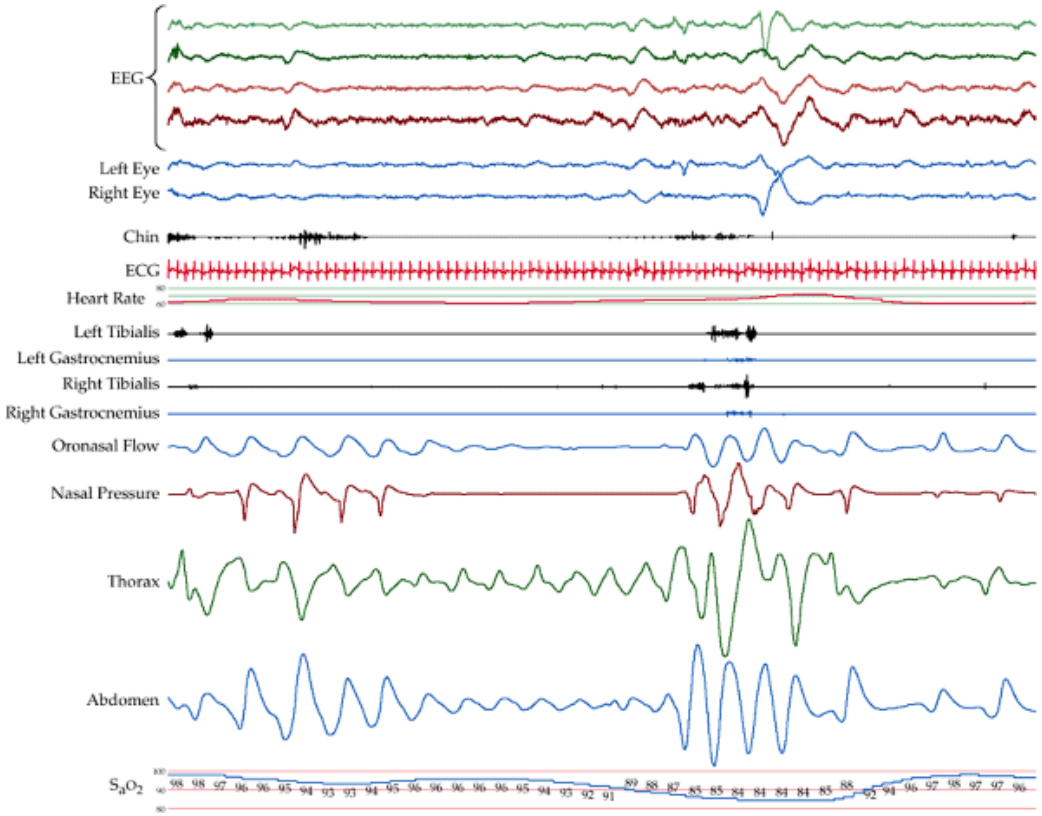


Figure 3.1: The figure displays the different channels in PSG.There are several channels in the PSG and each channel is a time series recorded by the units of 10 seconds. During the whole sleeping period (9.5 to 10 hours often), there are millions of time points recorded and for each participant, their PSG data would be multivariate time series with millions of time points.In reality, sleep specialists could record different signals based on different types of diseases or diverse goals.

Particularly, in our study, each of the 100 participants with their PSG recorded has 28 signals in their PSG, namely Electroencephalography (EEG, which has 4 channels of signals namely C3, C4, A1 and A2), left outer canthus (LOC), right outer canthus (ROC), electrocardiogram (which has two signals namely ECG1 and ECG2), LEFT LEG1, LEFT LEG2, RIGHT LEG1, RIGHT LEG2, electromyogram (which has three signals, namely EMG1, EMG2 and EMG3), Airflow via thin catheters placed in front of nostrils and mouth (AIR-FLOW), absence in the effort in the thoratic (THOR EFFORT), absence of effort in the abdominal (ABDO EFFORT), Snoring (SNORE), sum channels (SUM), Body position (POSITION), Oxygen saturation (OX STATUS), pulse oximetry (PULSE), Oxygen level (SpO2), Light, heart rate (HRate), plethy-mograpgy (Pleth WV), and nasal pressure (NASAL PRES). Each signal is a time series recorded every 10 seconds during a ten hour sleep period, as per signal we have millions of time points. We sample 3000 time points from these millions with an even interval, which means, we sample one time point every 30 seconds. Figure 3.2 illustrates the PSG record of Participant 1 in our study. This figure is downloaded from the NSRR website.

In our study, each participant has 28 channels in their PSG records. We can see the 28 time series are of different types. Some have periods and some are fluctuating dramatically. Meanwhile, POSITION is almost a straight line. We sample from the 28 time series every 30 seconds and make the time series into matrix format. This results in a 3000 by 28 matrix for each participant.

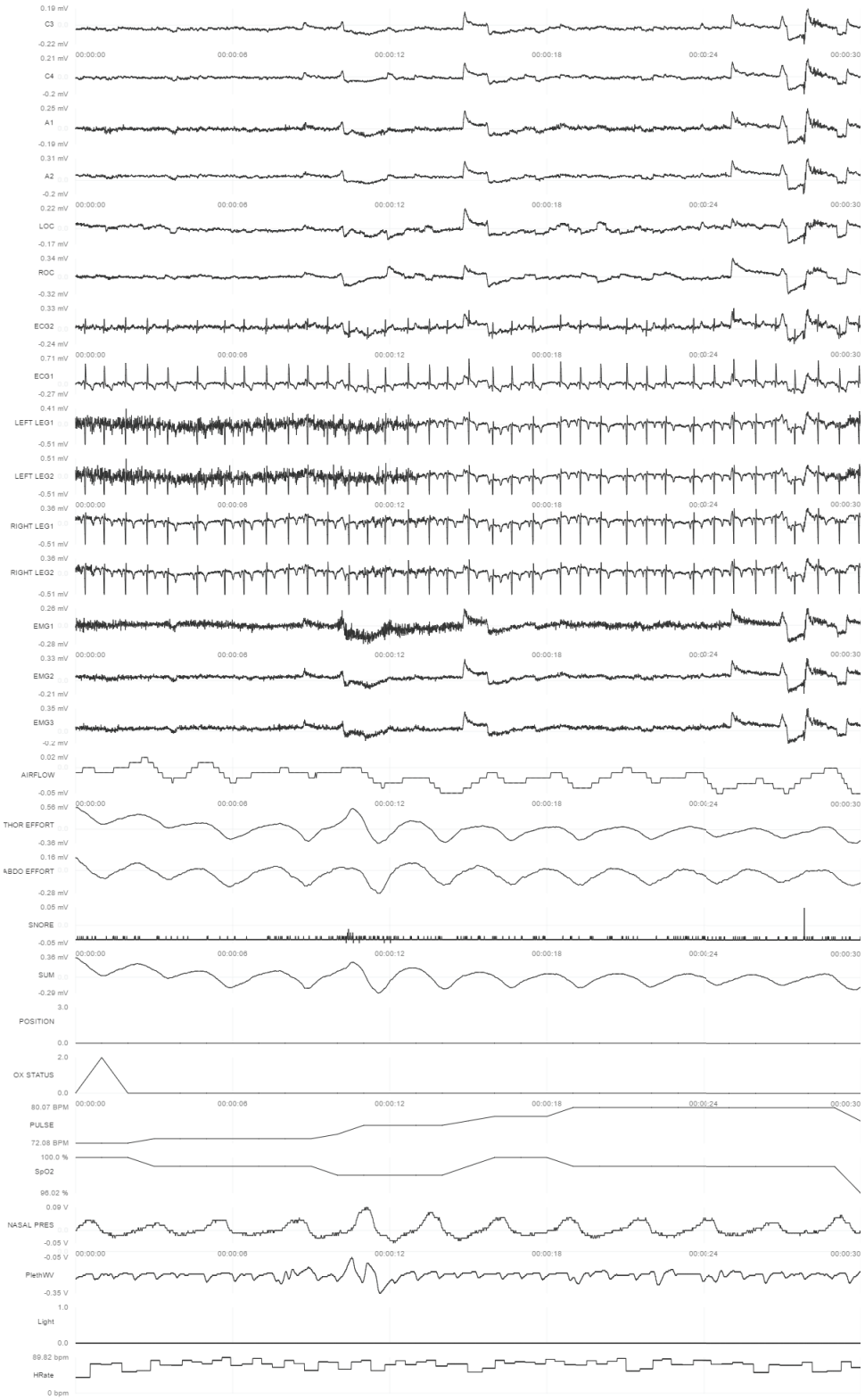


Figure 3.2: The figure displays the different channels in PSG for Participant 1.

3.3 Persistence landscapes

To summarize the information in PSG time series, we apply persistent homology to their partial correlation matrix. For each patient, we have 28 signals (time series), and each measures a certain part or function in brain. To estimate the connections between two signals while holding the others as constants, we calculate their pairwise partial correlations and organize them into a matrix (this work is done by R package *"parcor"*). This results in a 28 by 28 symmetric matrix with its diagonal all ones. We then transform it into a dissimilarity matrix by subtracted this matrix from a 28 by 28 matrix containing all ones.

Given this dissimilarity matrix, we could obtain its persistence diagram and barcode by *TDA* package in R and calculate integrals of its persistence landscape function for dimension 0, 1, 2 and 3 respectively. By applying persistent homology to the PSG time series, we could refine 4 numerical features for each participant, that is the integrals of its persistence landscape for dimension=0, 1, 2 and 3.

Figure 3.3 displays the persistence diagram and barcode, derived from the 28 by 28 partial correlation matrix of PSG time series, for Participant 1.

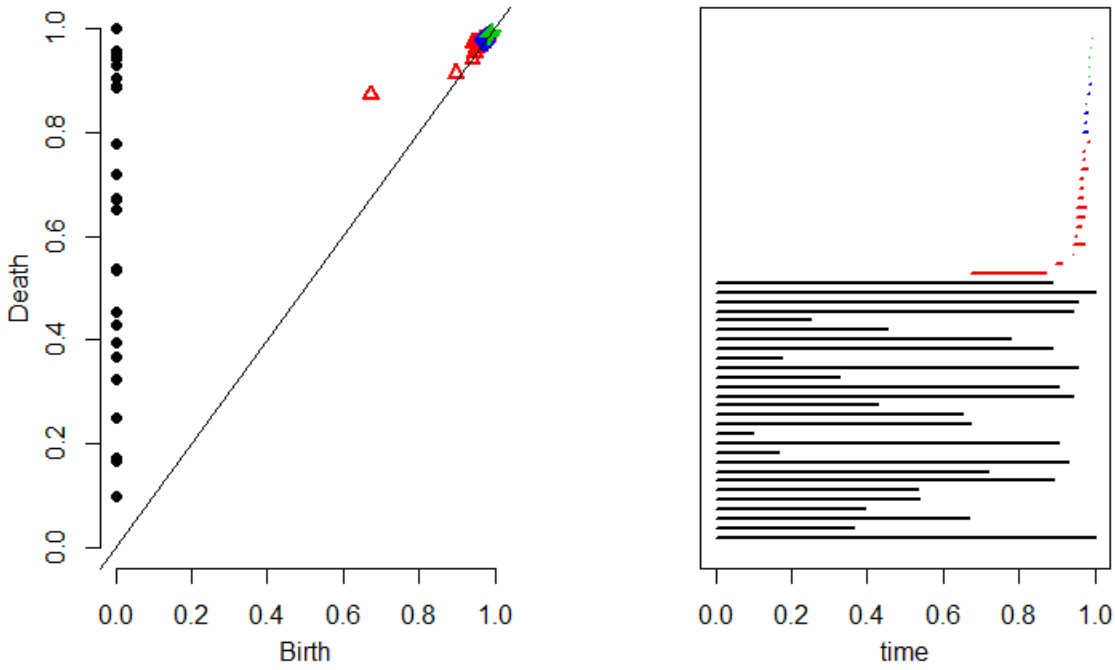


Figure 3.3: The persistence diagram and barcode of 28 signals from PSG of dimension 0, 1, 2 and 3 for participant 1

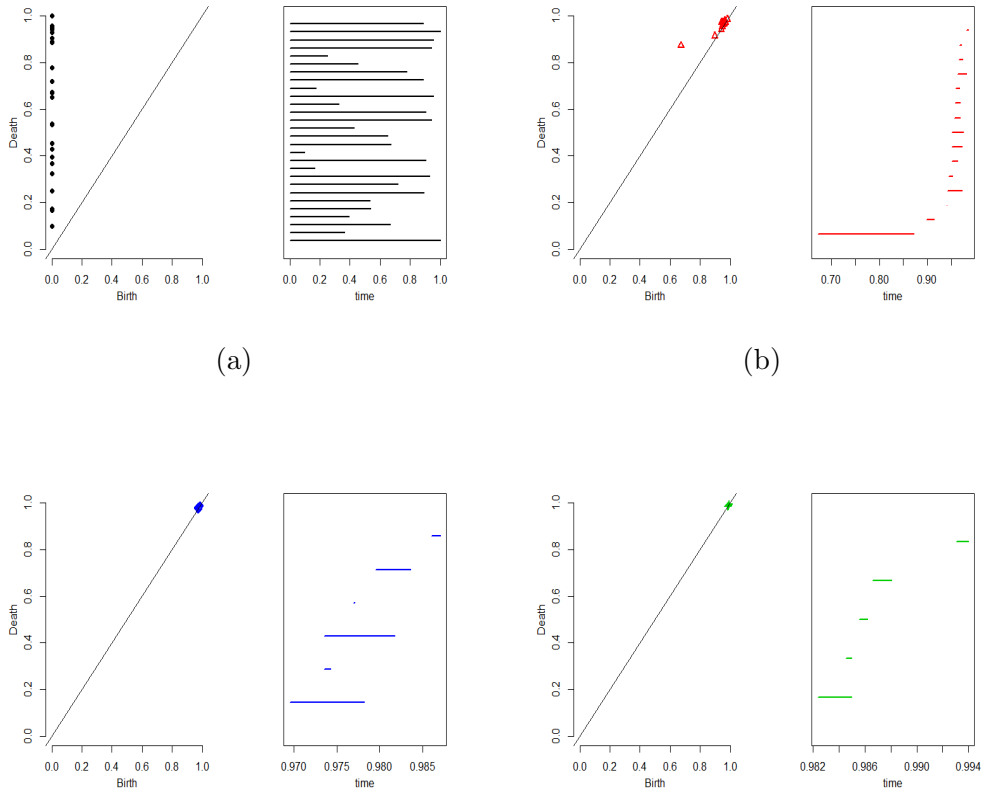


Figure 3.4: The figure displays the persistence diagram and barcode, derived from the 28 by 28 partial correlation matrix of PSG time series, for Participant 1. Top is the persistence diagram and barcode for dimension 0, 1, 2, 3 drawn together. (a) to (d) are persistence diagrams and barcodes for dimension 0, 1, 2, 3 respectively. From each barcode, we obtain its landscapes and calculate the integrals. This is the summary of information from the participant's PSG and take them as covariates to build a prediction model for their OAHI.

Persistence landscapes are real-valued functions that further summarize the information contained in a persistence diagram. The persistence landscape is a collection of continuous, piecewise linear functions $\lambda : \mathbb{Z}^+ \times \mathbb{R} \to \mathbb{R}$ that summarizes a persistence diagram. To define the landscape, consider the set of functions created by tenting each point $p = (x, y) = (\frac{b+d}{2}, \frac{d-b}{2})$, representing a birth-death pair (b, d) in the persistence diagram D as follows:

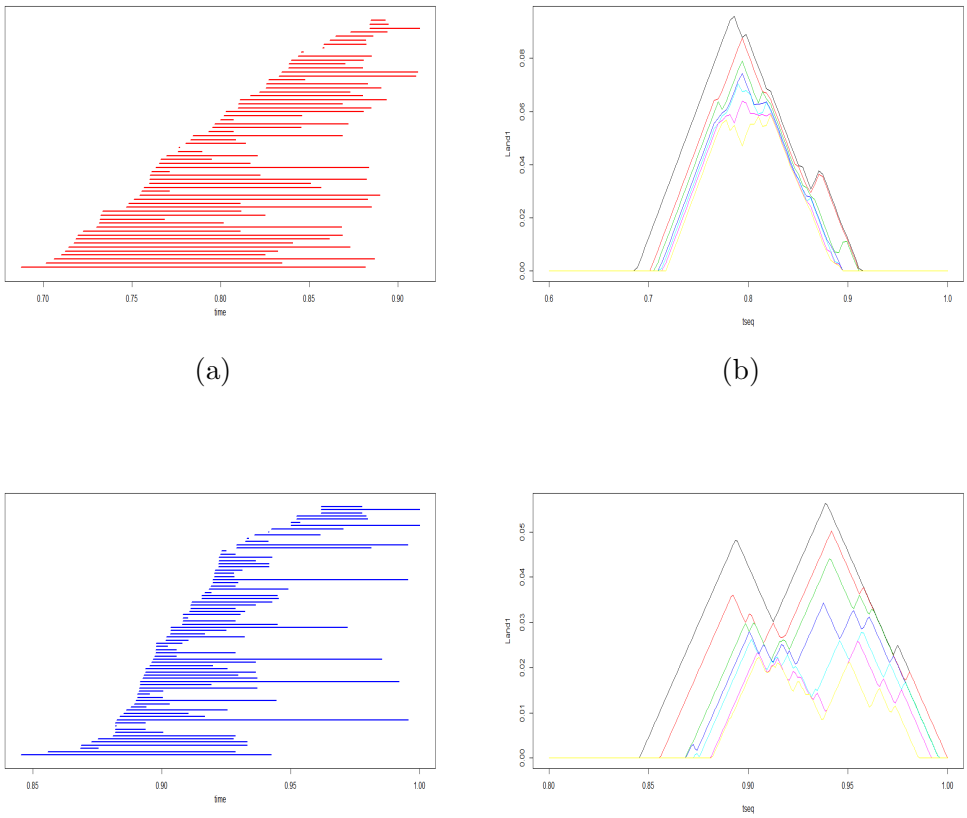
$$\Lambda_p(t) = \begin{cases} t - x + y & t \in [x - y, x] \\ x + y - t & t \in (x, x + y] \end{cases} = \begin{cases} t - b & t \in [b, \frac{b+d}{2}] \\ d - t & t \in (\frac{b+d}{2}, d] \\ 0 \end{cases}$$

We obtain an arrangement of piecewise linear curves by overlaying the graphs of the functions $\{\Lambda_p\}_p$; the persistence landscape of D is a summary of this arrangement. Formally, the persistence landscape of D is the collection of functions

$$\lambda(k, t) = k \ \underset{p}{max} \Lambda_p(t), t \in [0, T], k \in \mathbb{N}$$

where $kmax$ is the k th largest value in the set. In particular, 1 max is the usual maximum function

Figure 3.5 demonstrates the landscapes from the corresponding persistence brcodes. For each barcode, we can define a landscape function and for all the landscape functions, we take the k th largest value as the k th landscape.Then calculate the integrals of each landscape function, and sum them up.



(a) (b)

(c) (d)

Figure 3.5: The figure displays the persistence diagrams and barcodes, derived from the 28 by 28 partial correlation matrix of PSG time series, for Participant 1 and their corresponding persistence landscapes. (a) is dimension 1 peristence diagram and barcode, (b) is its corresponding landscapes. (c) is dimension 2 peristence diagram and barcode. (d) is its corresponding landscapes.

By this procedure, we obtained 4 landscapes (dimension 0, 1, 2 and 3) for each participant. Altogether we have 400 landscapes and among them the largest order is 28. Next, for each dimension, we calculate the sum of integrals of its every single order landscape. For dimension 0 landscapes, calculate:

$$\sum_{k=1}^{28} \int_0^1 \lambda_k(t) d_t = L_i, \ i = 1, 2, 3, 4$$

Likewise, for each participant, calculate the integrals of landscapes for dimension 0, 1, 2 and 3, denoting them by L_1, L_2, L_3 and L_4. Take them as the new covariates we generate from the view of persistent homology for every one of the 100 participants.

Since we have 100 participants and their OAHI values lie in different intervals. We look at Participant No.5, whose OAHI value is 0.24, Participant No.20, whose OAHI value is 3.81 and Participant No.23, whose OAHI value is 14.66. These are three typical participants with different levels of OAHI and we can investigate the difference of their barcodes and landscapes. Figure 3.6 to Figure 3.11 illustrate the above three participants' persistence diagram (dimension 0,1,2 and 3 together) and barcode, and the corresponding persistence landscapes.

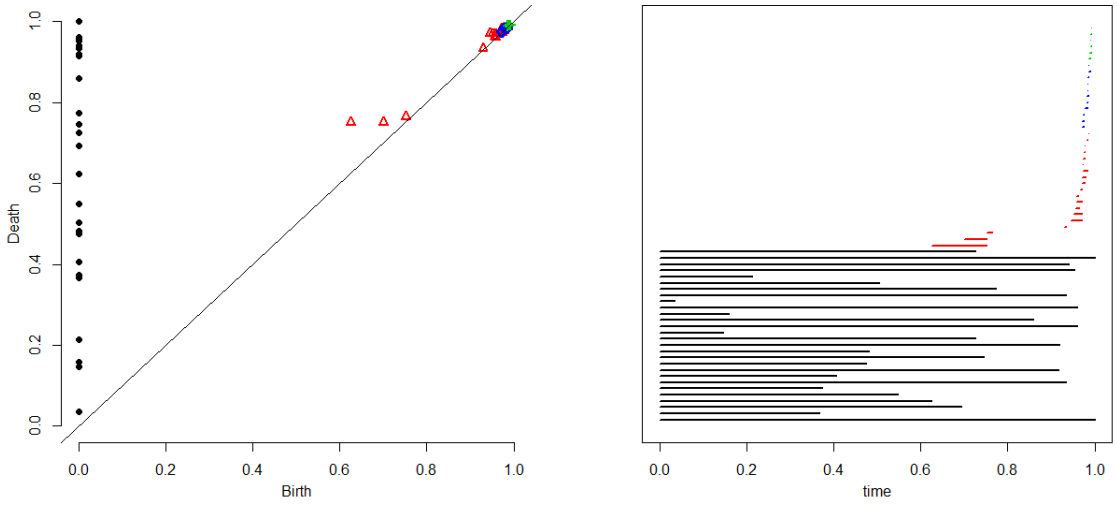
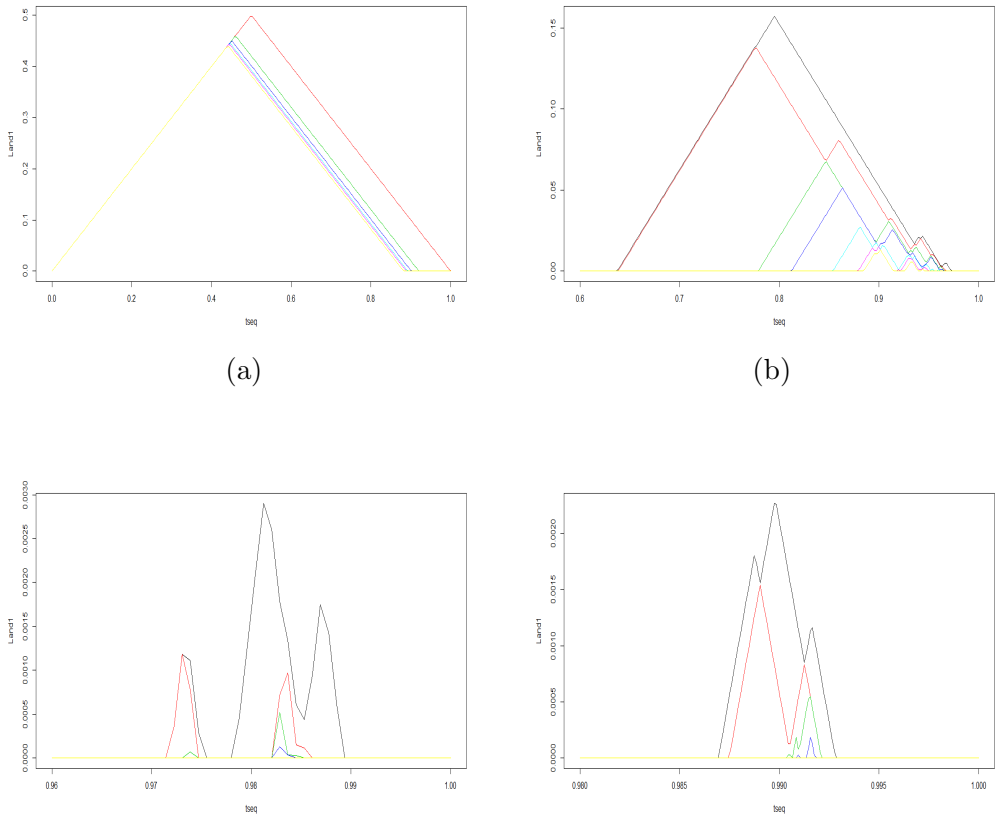


Figure 3.6: Participant No.5's persistence diagram and barcode of dimension 0,1,2 and 3 drawn together



(a) (b)

(c) (d)

Figure 3.7: The figure displays the persistence diagram and barcode, derived from the 28 by 28 partial correlation matrix of PSG time series, for Participant No.5. (a) to (d) are corresponding landscapes for dimension 0, 1, 2, 3 respectively. The landscapes are the summary of information from the participant's PSG. we take the integrals and sum the up for each dimension, taking the sum of dimension 0, 1, 2 and 3 as the covariates to build a prediction model for their OAHI.

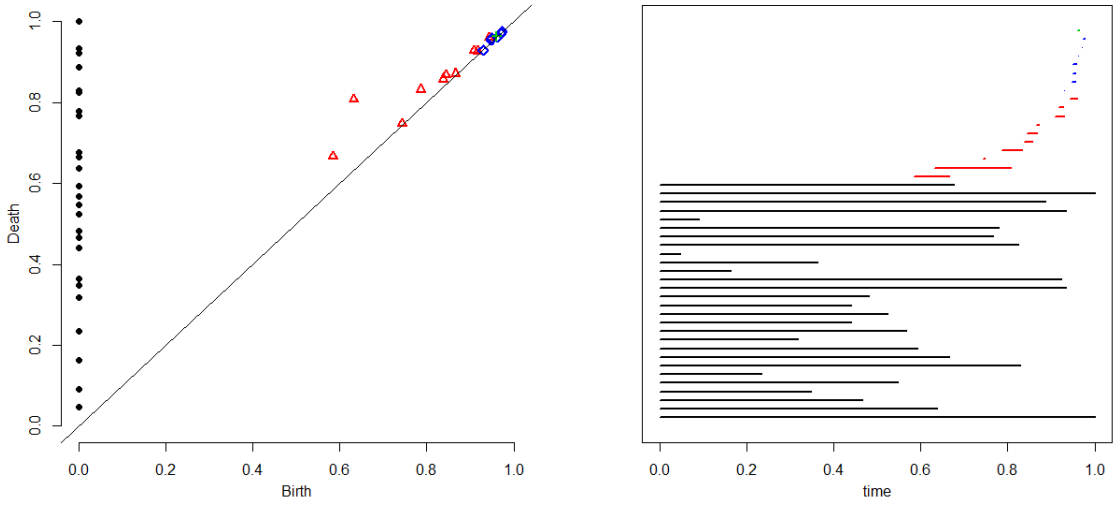
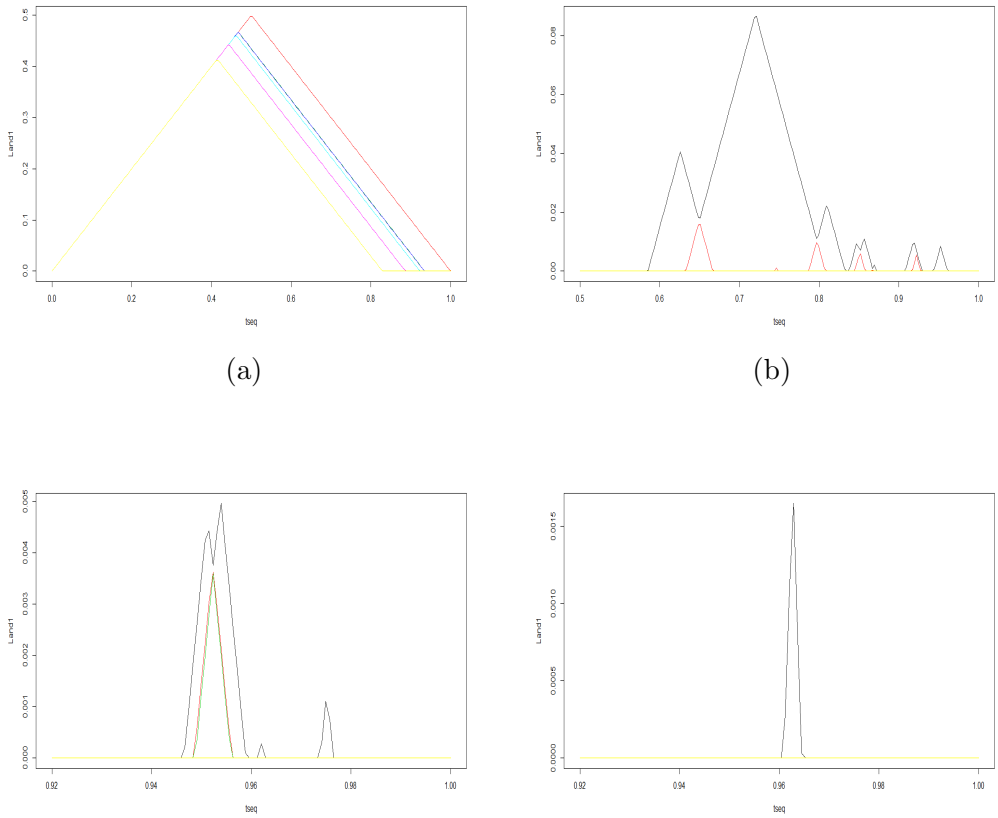


Figure 3.8: Participant No.20's persistence diagram and barcode of dimension 0,1,2 and 3 drawn together



(a) (b)

(c) (d)

Figure 3.9: The figure displays the persistence diagram and barcode, derived from the 28 by 28 partial correlation matrix of PSG time series, for Participant No.20. (a) to (d) are corresponding landscapes for dimension 0, 1, 2, 3 respectively. The landscapes are the summary of information from the participant's PSG. we take the integrals and sum the up for each dimension, taking the sum of dimension 0, 1, 2 and 3 as the covariates to build a prediction model for their OAHI.

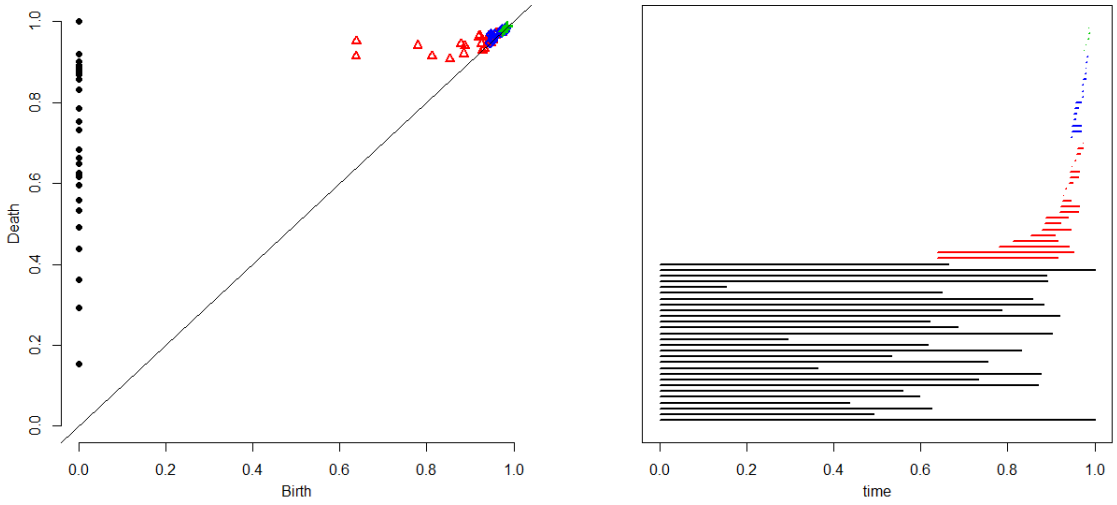
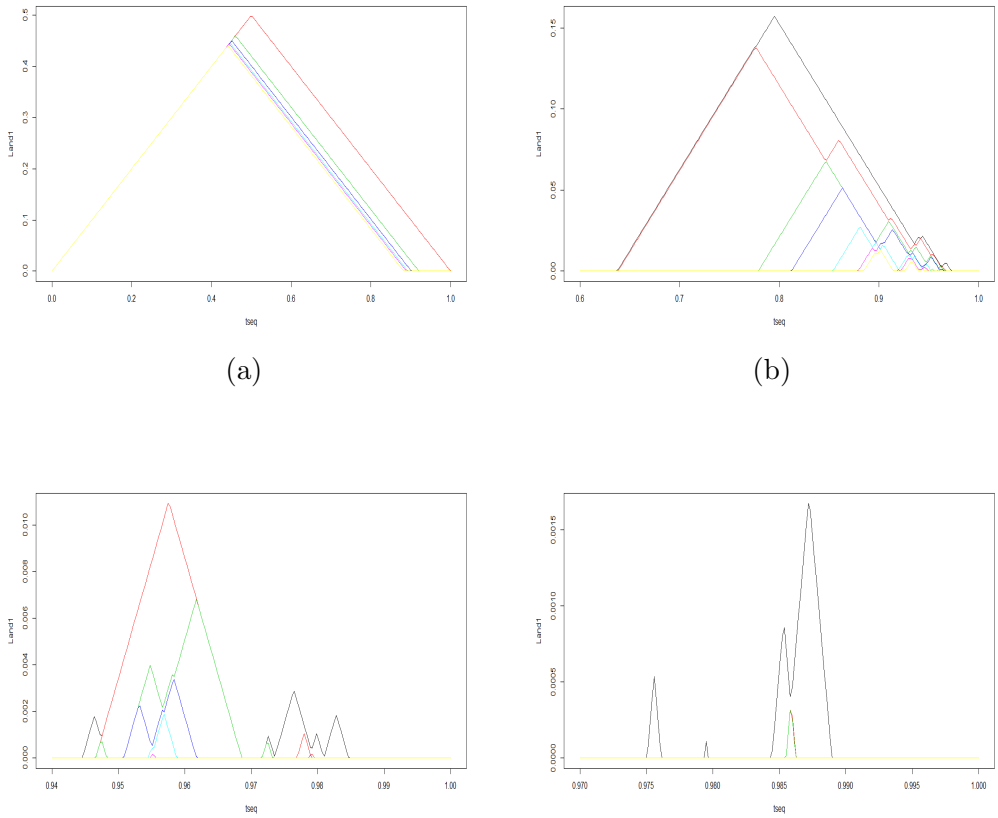


Figure 3.10: Participant No.23's persistence diagram and barcode of dimension 0,1,2 and 3 drawn together



(a) (b)

(c) (d)

Figure 3.11: The figure displays the persistence diagram and barcode, derived from the 28 by 28 partial correlation matrix of PSG time series, for Participant No.23. (a) to (d) are corresponding landscapes for dimension 0, 1, 2, 3 respectively. The landscapes are the summary of information from the participant's PSG. we take the integrals and sum the up for each dimension, taking the sum of dimension 0, 1, 2 and 3 as the covariates to build a prediction model for their OAHI.

3.4 Random Forest Model for PSG

Random forest is an ensemble learning-based classification and regression technique. It is one of the commonly used predictive modelling and machine learning methods. Random forest algorithm can be used for both classification and regression problems. In the scenario of our study, what we desire is the prediction of OAHI value, which is a supervised learning with labeled instances because OAHI is a continuous numerical variable. Consequently here we will use random forest for regression problems.

Random forest could be deemed as an ensemble model of decision trees. In a normal decision tree, one decision tree is built and in a random forest, a number of decision trees are built during the process. A vote from each of the decision trees is considered in deciding the final class of a case or an object and this is called ensemble process. This is a democratic process. Since many decision trees are built and used in a process of random forest algorithm, it is called a forest. It is believed that by averaging across the high variance, low bias trees, we will end up with a low bias, low variance estimator. In this sense, random forest will give out a better prediction than any single tree could.

Compared with single decision tree, random forests improve predictive accuracy by generating a large number of bootstrapped trees (based on random samples of variables), classifying a case using each tree in this new "forest", and deciding a final predicted outcome by combining the results across all of the trees (for regression, it takes the average of regression results from every tree and for classification, it adopts the majority vote of trees).

Random forest has many advantages which satisfy the needs in our study. Firstly, it possesses the strong power of handling large data sets with higher

dimensionality. Secondly, besides handling thousands of input variables, random forest could also identify the variables that contribute most to the model and this is what we need in this study, since we look forward to identifying important features in predicting OAHI values among all the 135 covariates. Furthermore, it has an effective method for estimating missing data and maintains accuracy and higher model performance when a large proportion of the data are missing. There are also disadvantages of random forest that should be taken into consideration carefully in this study. There are two disadvantages that would affect our model most. Firstly, random forest does not do as good at classification for regression problems since it does not give precise continuous nature predictions. This means in the case of regression, it would never predict beyond the range in the training data. Secondly, Random Forest may suffer from high chance of over-fitting especially when training data has a lot of noise.

When evaluating the model learnt from the training datasets, we do not have to conduct cross validation when using Random Forest, since random forest involves sampling of the input data with replacement called bootstrap sampling. Here one third of the data is not used for training and can be used for testing. These are called the out-of-bag samples. Error estimated on these out-of-bag samples is known as out-of-bag error. Study of error estimates for out-of-bag gives evidence to show that the out-of-bag estimate is as accurate as using a test set of the same size as the training set. Therefore, using the out-of-bag error estimate removes the need for a set-aside test set and saves us the work of cross validation.

For building a random forest predictor, we only need to set two parameters: *mtry* and *ntree*. *mtry* is the number of variables taken at each node to build

a tree and *ntree* is the number of trees to be grown in the forest. Once these two parameters are set, random forest will work in the following manner:

1. Assume the number of cases in the training set is N. Then, a sample of these N cases is taken at random with replacement. This sample will be the training set for growing the tree.

2. If there are M input variables, a number $mtry < M$ is specified such that at each node, $mtry$ variables are selected at random out of the M. The best split on these $mtry$ is used to split the node. The value of $mtry$ is held constant while we grow the forest.

3. Each tree is grown to the largest extent possible and there is no pruning.

4. Predict new data by aggregating the predictions of the *ntree* trees for our predicting OAHI problem.

Random forest approach could be implement using *randomForest* package in R. Moreover, combined with packages *rpart*, *caret* and *e1071*, we are also able to find optimum value of model parameters *mtry* and *ntree*.

3.5 Model building and Evaluation

We use all of the 100 labeled data to train the random forest model. It is a supervised learning process and we set the variable oahi3 (OAHI) as a dependent variable and all the other covariates together with L_1, L_2, L_3 and L_4, which we obtained from applying persistent homology to partial correlation matrix of PSG time series, as predictors. In this way, we make use of information both from covariates and PSG, to estimate the value of OAHI.

In the building of the random forest, we confront the following problems: 1. How to choose the optimum value of parameters in the model; 2. How to

deal with missing values in the predictors.

In our datasets, there are quite a number of missing values in the predictors. For some participants, even ten out of 135 covariates could be missing. Random forests don't handle missing values in predictors automatically; we have to come up with a way to cope with it appropriately. Basically, if the predictors have missing values, we have two choices:

1. Use a different tool (method *rpart* could handle missing values nicely.)

2. Impute the missing values Not surprisingly, the *randomForest* package has a function for imputing the missing values, *rfImpute*. Here, we adopt the second way mentioned in order to deal with missing values on our datasets. The function *rfImpute* imputes missing values in predictor data using proximity from randomForest. It starts by imputing *NAs* using *na.roughfix*, then *randomForest* is called with the completed data. The proximity matrix from the *randomForest* is used to update the imputation of the *NAs*. For continuous predictors, the imputed value is the weighted average of the non-missing observations, where the weights are the proximities. For categorical predictors, the imputed value is the category with the largest average proximity. This process is iterated *iter* times.

When missing values in the predictors are well filled by the random forest, the model then is learnt from the imputed training dataset. To finally learn the model, we have to decide the value for parameters $mtry$ and $ntree$.

Firstly well try to find the optimal numbers of variables to try splitting on at each node ($mtry$).

We use *tuneRF* function to see the optimum value for $mtry$. The output shows that when $mtry = 16$, the model reaches its lowest mean square error so we choose 96 as the optimum value for parameter $mrty$. This means, at

Tree	Out-of-bag MSE	Out-of-bag %Variance
100	2.127	69.41
300	2.166	70.70
500	2.071	67.60
800	2.024	66.05
1000	2.009	65.55

Table 3.2: As the table shows, with the number of *ntree* (denoted by Tree in the table) increasing, the mean square out of bag error decreases, and when the tree number equals 1000, the Out-of-bag MSE is 2.009. So we take the parameter $ntree = 1000$.

each node when building a tree, we randomly choose 16 out of 135 predictors to split on.

As for *ntree*, we use package *rpart*, caret and e1071 and the output is displayed in Table 3.2.

When we take $ntree = 1000$, as it gives the smallest MSE for the model, we have built a model with participants covariates and their PSG records as features, to predict their OAHI values. In other words, given a new participant, as long as we have his PSG records and other covariates, we could reasonably use the trained model to predict his OAHI value.

To evaluate the performance of this model, we first look at the error rate of the forest with the number of trees, as 1000 decision trees (a forest) have been built using the random forest algorithm based learning. We can plot the error rate across decision trees. The figure indicates that after 200 decision trees there is not a significant reduction in error rate. Also, overall, the error rate for the model is relatively low (around 2.70). This confirms that the model we trained is quite effective. The error plots are illustrated in Figure 3.12

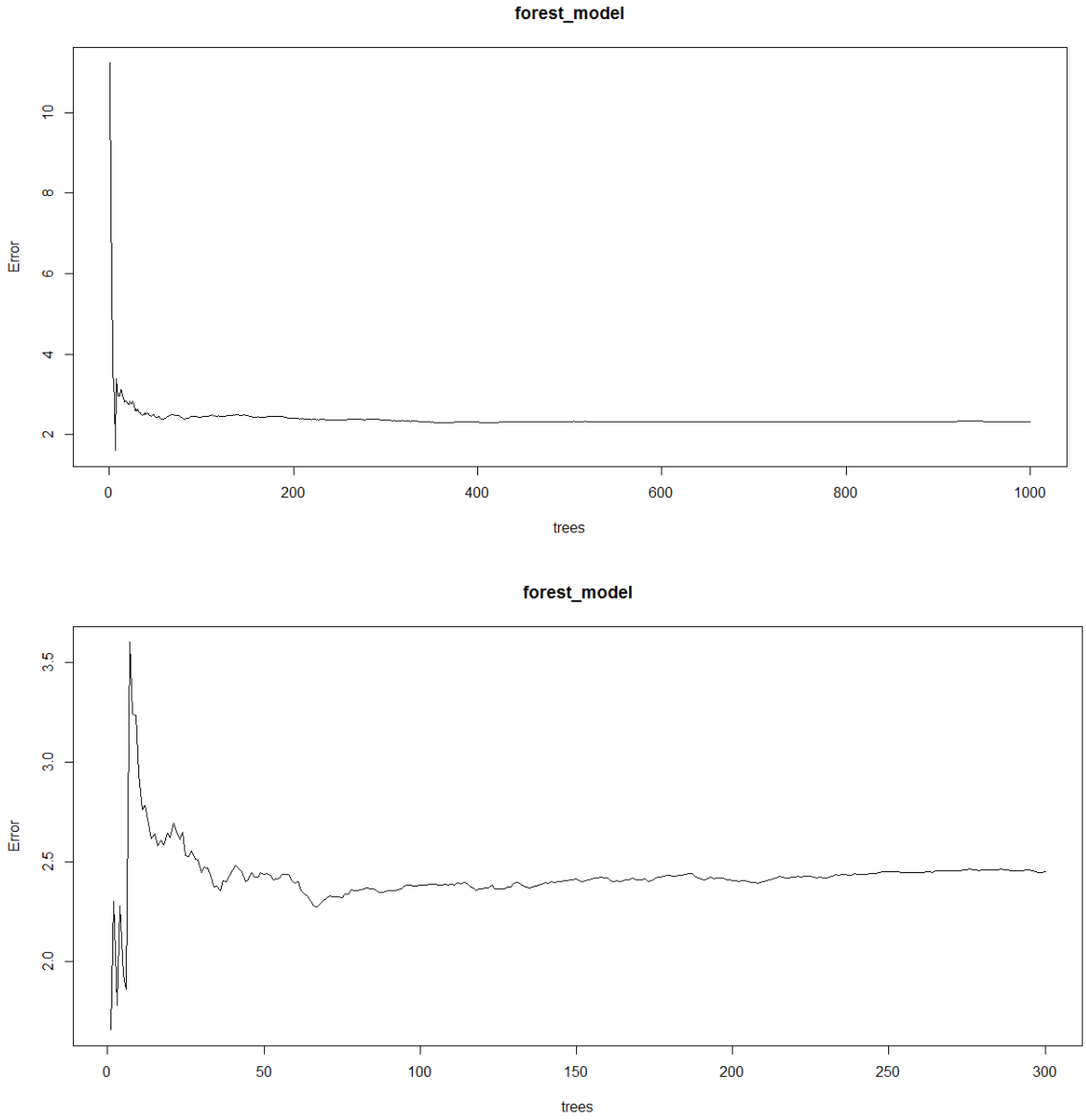


Figure 3.12: Top is the error plot of random forest model with the tree number ranging from 0 to 1000. Bottom is the error plot of random forest model with the tree number ranging from 0 to 300. When tree number is relatively small, the error of the model is large, and if the tree number is larger than 300, the error would be around 2.50 and not decrease obviously. Overall, after the tree number come to 300, the error rate for the model is relatively low (below 2.50). This confirms that the model we trained is quite effective.

Lastly, we discuss the contribution of covariates generated by applying persistent homology (L_1, L_2, L_3 and L_4) to the model. When building the model, random forest calculates the variable importance so we can see what contributed most to estimating OAHI. This evaluates what variables were most important in generating the forest.

The *importance()* function gives us a textual representation of how important the variables were in building the model, while the *varImpPlot()* function gives us a graphical representation of the importance of each predictor.

The variable importance measures is defined as the total decrease in node impurities from splitting on the variable, averaged over all trees. For classification, the node impurity is measured by the Gini index. For regression, it is measured by residual sum of squares. Figure 3.13 is the plot of variable importance. The y-axis corresponds to different covariates we have in building the model and their importance to the model are measured by *IncNodePurity* (*IncNodePurity* is the total decrease in node impurities, measured by the Gini Index from splitting on the variable, averaged over all trees). The large value of *IncNodePurity* means this variable contributes more to the model and thus is more important.

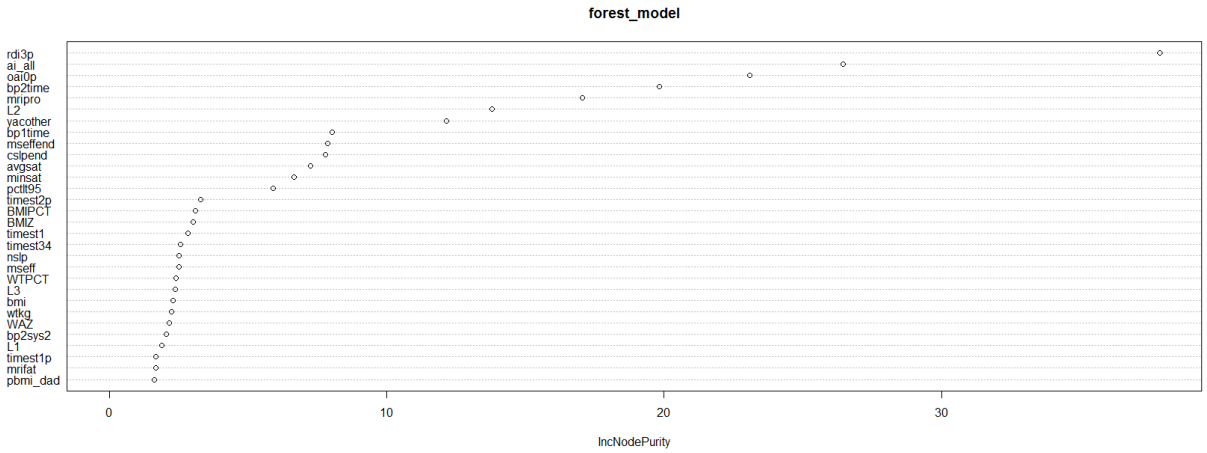


Figure 3.13: *IncNodePurity* for the covariates we used in building the model. *IncNodePurity* indicates how important the covariate is to the model. The important covariate would have a large *IncNodePurity* value. the plot displays only top 30 most important covariates. We can see L_1, L_2 and L_3 are all among them. As we have 137 covariates in total, this means the information from PSG contributes a lot to predicting participant's OAHI and the way of persistence landscape retain this important information. Especially, L_2 ranks very high among all the variables. The landscape from dimension 1 persistent diagram is especially important.

Covariate	IncNodePurity
L1	0.936609718
L2	16.235292518
L3	1.101331238
L4	0.910485854

Table 3.3: This table shows the *IncNodePurity* for L_1, L_2, L_3 and L_4. L_1, L_2 and L_3 all count for the most important variables in the model as is indicated by the importance plot. Among them, L_2 has especially large value, confirming that it is a very important covariate in the model.

To evaluate the importance of each covariate, we use the measure of *IncNodePurity*. *IncNodePurity* relates to the loss function, which by best splits are chosen. The loss function is MSE for regression and Gini-impurity for classification. More useful variables achieve higher increases in node purities. Table 3.3 shows the *IncNodePurity* for L_1, L_2, L_3 and L_4. among them, L_2 has especially large value, confirming that it is a very important covariate in the model.

We see from the Variable importance plot, that the most important variables included rdi3p (Overall Respiratory Disturbance (3% desaturation) Index), ai˙all (Overall arousal index), Oai0p (Obstructive Apnea (all desaturation) Index), bp2time (Time of Morning collection), mripro (Mean total protein per day) and L˙2. Furthermore, among all the 135 covariates, we pick out the 30 most important ones, including L˙1, L˙2, and L˙3. Especially, L˙2 (the calculus of 1st order persistence landscape) is of significantly importance compared with other covariates. This confirms that the polysomnography record does affect the value of OAHI and the method of obtaining barcode of partial correlation matrix of the PSG time series and then taking sum of integrals of dimension 0, 1, 2, 3 landscapes do retain these features. Persistent homology works well when applied to PSG time series data and their partial correlation matrices.

Chapter 4

Future Study

In our study, we made several contributions. Firstly, we realized unsupervised learning by persistent homology. We applied persistent homology to cross correlation matrices and partial correlation matrices generated from time series. By the persistence diagrams and barcodes, we extracted the most consistent clusters and loops for the 46 wells located in different areas in Edmonton. Secondly, we applied persistent homology to PSG time series and obtained its partial correlation matrices. Based on the barcodes, we generate their landscapes and take integrals of landscapes for dimension 0, 1, 2 and 3. Using these integrals of landscapes together with other covariates, we built a random forest to predict participants OAHI based on these topology features extracted from their PSG records. This work is done by means of persistent homology and supervised learning method.

The work we will accomplish in the future includes investigating further into the water level data and predicting the water level in the future by persistent homology and machine learning method. Additionally, as we have several parameters in chapter 2 and 3, take for example, the window width (w) and

step length (s), figuring out the optimal sets of these parameters would be also worth working on. As for the persistent homology of partial correlation matrices and cross correlation matrices, we need to do bi-filtration in topology data analysis. Lastly, as for the PSG time series, persistent homology for the same signals between different participants could also be investigated to obtain useful information about which signal is of significance to predict OSA and meanwhile, EEG spectral analysis on the 100 participants PSG EEG signals would be conducted to further throw light on the prediction problem of OSA.

Bibliography

[1] P. Bubenik and P. Kim, *A statistical approach to persistent homology.* (2007).

[2] Heo G, Gamble J, Kim PT, *Topological analysis of variance and the maxillary complex.* J Am Stat Assoc.

[3] G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian, *On the local behavior of spaces of natural images* preprint, (2006).

[4] Gander, Golub, Strebel, *Least Squares Fitting of Circles and Ellipses.* BIT (1994), 558-578.

[5] G. Carlsson, A. Zomorodian, A. Collins, and L. Guibas, *Persistence barcodes for shapes.* Intl. J. Shape Modeling, 11 (2005), 149-187.

[6] F. Chazal and A. Lieutier, *Weak feature size and persistent homology: computing homology of solids in Rn from noisy data samples.* in Proc. 21st Sympos. Comput. Geom. (2005).

[7] D. Cohen-Steiner, H. Edelsbrunner and J. Harer, *Stability of persistence diagrams.* in Proc.21st Sympos. Comput. Geom. (2005), 263-271.

[8] V. de Silva, *A weak definition of Delaunay triangulation.* preprint (2003).

[9] V. de Silva and G. Carlsson, *Topological estimation using witness complexes.* in SPBG04 Symposium on Point-Based Graphics (2004), 157-166.

[10] Jisu Kim, *Tutorial on the R package TDA.*

[11] L. Guibas and S. Oudot, *Reconstruction using witness complexes.* in Proc. 18th ACMSIAM Sympos. on Discrete Algorithms, (2007).

[12] Kelin Xia and Xin Feng, *Persistent Homology for The Quantitative Prediction of Fullerene Stability.* (2014).

[13] T. Kaczynski, K. Mischaikow, and M. Mrozek, *Computational Homology.* Applied Mathematical Sciences 157, Springer-Verlag, (2004).

[14] A. Zomorodian and G. Carlsson, *Computing Persistent Homology*. Discrete Comput. Geom., 33, (2005), 249-274.

[15] A. Zomorodian and G. Carlsson, *The theory of multidimensional persistence*. preprint (2006).

[16] R. Ghrist, *Barcodes: The persistent topology of data*. 2014.

[17] Adler, R. J., Bobrowksi, O., Borman, M. S., Subag, E., and Weinberger, *Persistent Homology for Random Fields and Complexes*. MS collections, 6, 124-143.

[18] H. Adams and A. Tausz, *Javaplex: A toolbox for persistent homology*. 2011.

[19] Edelsbrunner, H., Letscher, D., and Zomorodian, A, *Vines and Vineyards by Updating Persistence in Linear Time*. pp.119-126.

[20] Cohen-Steiner, D., Edelsbrunner, H., and Morozov, D, *Topological persistence and simplication*. Discrete and Computational Geometry, 28, 511-533.

[21] G. Heo, *Topological and Statistical Data Analysis*. Notes for Math 600 course, Summer 2013.

[22] G. Heo, J. Gamble, and P. Kim, *Topological analysis of variance and the maxillary complex*. Journal of the American Statistical Association, 107:477-492, 2012.

[23] B. Fasy, F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, and A. Singh, *Statistical Inference For Persistent Homology: Confidence Sets For Persis-tence Diagrams*. arXiv:1303.7117, 2013.