Finding Dynamic Co-evolving Zones in Spatial-Temporal Time Series Data

Yun Cheng(✉), Xiucheng Li, and Yan Li

Air Scientific, Beijing, China
chengyun.hit@gmail.com, xiucheng90@gmail.com, yan.li@coilabs.com

Abstract. Co-evolving patterns exist in many Spatial-temporal time series Data, which shows invaluable information about evolving patterns of the data. However, due to the sensor readings' spatial and temporal heterogeneity, how to find the stable and dynamic co-evolving zones remains an unsolved issue. In this paper, we proposed a novel divide-and-conquer strategy to find the dynamic co-evolving zones that systematically leverages the heterogeneity challenges. The precision of spatial inference and temporal prediction improved by 7 % and 8 % respectively by using the found patterns, which shows the effectiveness of the found patterns. The system has also been deployed with the Haidian Ministry of Environmental Protection, Beijing, China, providing accurate spatial-temporal predictions and help the government make more scientific strategies for environment treatment.

Keywords: Air quality · Time series clustering · Co-evolving

1 Introduction

Spatio-temporal time series data has become ubiquitous thanks to affordable sensors and storage. Those invaluable data shows a potential to extract and understand complex spatio-temporal phenomena and their dynamics. Additionally, the ubiquitous sensor stations continuously measure several geophysical fields over large zones and long (potentially unbounded) periods of time, which highlights the importance of unsupervised methods in monitoring spatio-temporal dynamics with little or no human supervision.

Time series clustering are rapidly becoming popular data mining techniques. Lots of methods have been proposed to solve the problem [18]. Different dissimilarity measures for time series have been tested for various purposes. Yet, the ubiquitous sensor monitoring data is always spatio-temporal heterogenous, which means that different clustering structure may exist during the whole period. Furthermore, in the geo-sensory applications wherein a bundle of sensors are deployed at different locations to cooperatively monitor the target condition, groups of sensors are spatially correlated and co-evolve frequently in their readings and how to find those spatial co-evolving patterns is of great importance to various real-world applications [21]. When dealing with dense and continuous

B. Berendt et al. (Eds.): ECML PKDD 2016, Part III, LNAI 9853, pp. 129–144, 2016.
DOI: 10.1007/978-3-319-46131-1_20

spatio-temporal data, the co-evolving sensors (zones) may change their sizes, shape and statistical properties over time (see Fig. 1). The goal is to find those dynamic co-evolving zones and try to establish linkages between those found zones and give reasonable explanations.

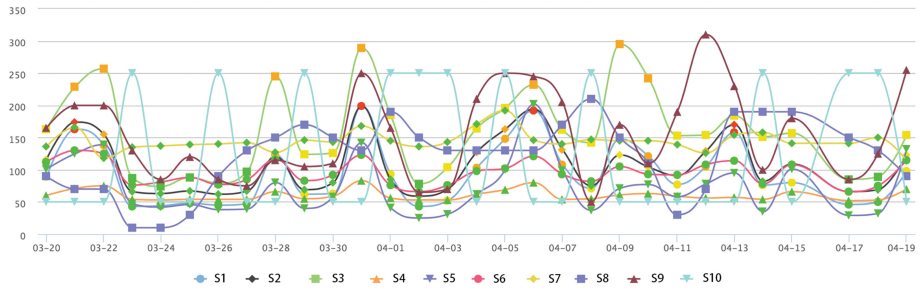

Fig. 1. The spatio-temporal air quality monitoring data (10 spatially adjacent sensor readings during one month).

In this paper, we propose a novel dynamic co-evolving zones discovery paradigm to identify co-evolving zones in continuous spatio-temporal field and establish linkages where the co-evolving zones may change their size, shape from time to time. Our paradigm first detects the overall breakout and divides the time series into uptrend and downtrend intervals. Then, we cluster the spatio-temporal time series data in each interval by using the specific dissimilarity measures. A hierarchical clustering method is used to deal with the found dynamic co-evolving zones in the previous step to give the final co-evolving structure. We evaluate our paradigm on a real world application of monitoring air quality which uses ubiquitous sensor stations on a regional scale (see our previous work [4] for details). The paradigm produced more stable and meaningful co-evolving zones and automatically found the segmentation intervals which helped better understand the evolving patterns of the pollution. We then use the found patterns to make spatial-temporal predictions and find an obvious improvement on the performance, which shows a potential usage area of the found dynamic co-evolving zones. Overall, our contribution has three parts:

- We proposed a novel paradigm to find the dynamic co-evolving zones and structures in the spatio-temporal time series data. The model uses three general key steps to deal with the spatio and temporal heterogenous to find co-evolving structures and patterns for future use.
- We use the patterns found in the co-evolving structures to increase the accuracy of spatial-temporal predictions and find a significant improvement compared with the original method.
- We use the proposed approach and result in a real world application, which has been used in the daily work of a environmental protection agency to help them make accurate predictions, do pollution causal analysis and make decisions or strategies.

2 Related Work

2.1 Problem Formulation

The goal of this work is to autonomously extract dynamic co-evolving zones from a continuous spatio-temporal field and give reasonable explanations. The dense deployment air quality monitoring data is an example of continuous spatio-temporal field, where each location has unique spatial coordinates and has co-evolving patterns with other sensors, which is changing over time. In the following subsection, we will first describe the existing approaches on the related topics, then gives our challenges.

2.2 Existing Approaches

Generally, our work is related to the following topics.

Time series change point detection. Sliding window, top-down, and bottom-up approaches [10] are popular methods to partition a time series into line segments. Wang et al. [17] proposed the pattern-based hidden Markov model that can segment a time series as well as learn the relationships between segments. Methods have also been proposed [9] to obtain piecewise polynomial approximations and/or perform on-line segmentation.

Change detection aims to find the time points where the statistical property of the time series changes significantly. It is closely related to time series segmentation as such points can be considered as the boundaries of different segments. Yamanishi et al. [20] unified the problems of change detection and outlier detection based on the on-line learning of an autoregressive model. Sharifzadeh et al. [15] used wavelet footprints to find the points where the polynomial curve fitting coefficients show discontinuities. Kawahara et al. [8] judged whether a point is a change by computing the probability density ratio of the reference and test intervals.

Our work uses the bottom-up segmentation approach due to its simplicity and practical effectiveness, it can be easily adapted to other segmentation algorithms. It is also worth mentioning that, the segmentation of this work is performed on short evolving intervals instead of the original long time series, which renders the segmentation process really fast.

Time series clustering. A crucial question in time series cluster analysis is establishing what we mean by similar data objects, i.e., determining a suitable similarity/dissimilarity measure between two time series objects. There exist a broad range of measures to compare time series and the choice of the proper dissimilarity measure depends largely on the nature of the clustering, i.e., on determining what the purpose of the grouping is. Current dissimilarity measures are grouped into four categories: model-free measures, model-based measures, complexity-based measures and prediction-based measures [13]. Considering the unsupervised feature of the problem and temporal heterogenous properties, we choose the model-free approaches.

The *Minkowski distance* is typically used to measure the proximity of two time series. This metric is very sensitive to signal transformations as shifting or time scaling. *Frechet distance* was introduced by Frechet [7] to measure the proximity between continuous curves, but it has been extensively used on the discrete case (see [6]) and in the time series framework. The dynamic time warping (DTW) distance was studied in depth by [14] and proposed to find patterns in time series by [2]. [5] introduce a dissimilarity measure addressed to cover both conventional measures for the proximity on observations and temporal correlation for the behavior proximity estimation, which includes both behavior and values proximity estimation.

In our scenario, we need to cluster the time series with both behavior and values similarity, we use an extension of the adaptive dissimilarity index covering both proximity on values and on behavior.

Co-evolving Zones. [16] studied the problem of finding regions that show similar deviations in population density using mobile phone data. They assume that the condition has periodicity, i.e., the daily population densities in a region are similar in different days. While this assumption is reasonable for population density, it does not hold in many geo-sensory applications like air quality monitoring. Moreover, they extract vertical changes in population density by comparing the same hour of different days. In contrast, we extract the horizontal changes, i.e., comparing the condition in current time interval with the previous time interval.

[21] studied problem of mining spatial co-evolving patterns from geo-sensory data, due to the sparse data they used, the paper only mines the spatial coevolving patterns (SCPs), i.e., groups of sensors that are spatially correlated and coevolve frequently in their readings. In our situation, we first find the co-evolving zones, then give the causal explanations of the phenomenon, which can be used to further improve the accuracy of spatial inference and temporal prediction.

2.3 Challenges

In addition to the technical limitations, finding the co-evolving zones faces significant challenges in many real world applications. One significant challenge is the heterogeneity in space and time (see Fig. 1). Space heterogeneity refers to the case where data belonging to different clusters may have the same feature values. While heterogeneity in time refers to the instance where the sensor cluster membership may change over time, which all lead to one much debated question [12]: How long should the time series be? If too short, the clusters found can be spurious; if too long, dynamics can be smoothed out. Those heterogeneity challenges caused us to propose a novel paradigm to eliminate the limitations.

Another challenge is how to find the physical meaning of the found co-evolving zones, i.e., how to give the causal explanation, and find associate relationship between those zones to improve the performance of other application domains, e.g., space inference and temporal prediction.

3 Overview

To address the above-mentioned challenges, we propose a novel dynamic co-evolving zones data mining paradigm that systematically leverages the very challenges. Our paradigm consists of two main steps: finding the co-evolving zones under the spatial and temporal heterogeneity constraint; mining the association between the found co-evolving zones and give reasonable explanations. Figure 2-A outlines three key steps. The first step is to do the changepoint detection, which acts on the average value of the monitoring region and gets the uptrend or downtrend change intervals for the use of next step. The second step is to cluster the time series data in every change interval, the key here is to choose an appropriate dissimilarity measure under the space constraint. The final step is to mine the relationship between the previous found zones, which in the best case will give us the inner relationship between those co-evolving zones and causal explanations of the phenomenon, which also gives us the appropriate time series segementation length for clustering analysis.

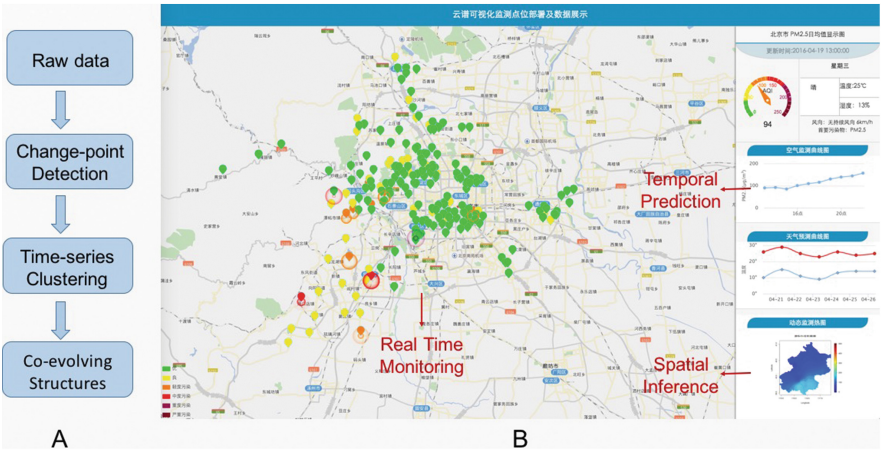

Fig. 2. A: dynamic co-evolving structure mining paradigm. B: web user interface of the deployed system.

The first step is essential. If we cluster the time series using the whole period, we will get bad and meaningless result for the space and temporal heterogeneity, which will be illustrated in the following experiment section. We cluster the segmented time series using an extension version of the adaptive dissimilarity index covering both proximity on values and on behavior in the second step. In the final step, we define a dynamic co-evolving zones' dissimilarity measure index and use the hierarchical clustering method to get the final co-evolving structure and give the dynamic segementation length used in clustering analysis.

Figure 2-B shows the real deployed web user interface in Haidian Ministry of Environmental Protection, where we can see the real time monitoring station

readings and accurate spatial-temporal prediction results. The above proposed paradigm helps us improve the prediction precision significantly and makes the scientific environment treatment possible.

4 Proposed Model

The proposed model takes an divide-and-conquer strategy to find the dynamic co-evolving structures and give final causal explanations. It first breaks down the problem into multiple sub-problems of the same (or related) type (divide), until these become simple enough (uptrend/downtrend intervals) to be solved directly (conquer). The solutions to the sub-problems are then combined to give a solution or explanation to the original problem. The approach eliminates the effect of space and temporal heterogeneity on the original problem and help produce more reliable and reasonable result for future use in related domains.

In this section, we will first describe our change point detection algorithm, which is simple and efficient, then follows the co-evolving time series clustering algorithm under the space constraint. Lastly, the co-evolving structure learning framework is proposed to build the relationship between the found co-evolving zones. The above steps belong to the divide-and-conquer approach, which divide the whole time series clustering problem into sub-problems and then cluster segmented co-evolving zones respectively to produce more stable and meaningful co-evolving zones.

4.1 Change Point Detection

Definition 1. *Uptrend/Downtrend Interval. Given a sensor reading s, an uptrend (downtrend) interval is a consecutive subsequence of measurement $\mathcal{I} = \langle s[t_i], s[t_{i+1}], \ldots, s[t_{i+m-1}] \rangle$ and $\forall j \in \{i, i+1, \ldots, i+m-2\}, s[t_{j+1}] - s[t_j] > 0 (< 0,$ for downtrend interval$)$, where m denotes the length of the subsequence and $t_i, t_{i+1}, \ldots, t_{i+m-1}$ are the timestamps of every measurement in $\mathcal{I}$.*

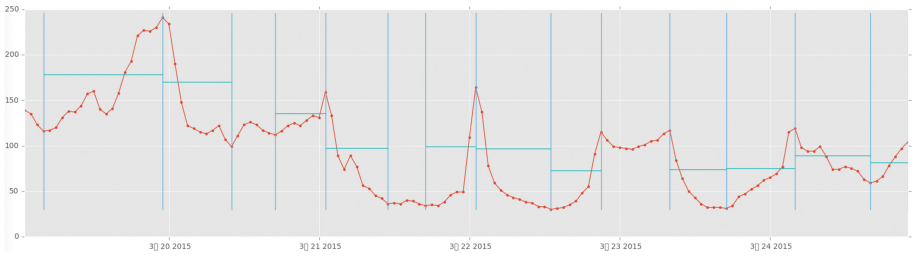

Fig. 3. The figure is the mean value of one monitoring region with almost 200 sensors; Blue lines is the segemented uptrend/downtrend intervals. (Color figure online)

Since the geo-sensory data is typically overwhelmed by various trivial fluctuations, we apply the wavelet transform to capture the multi-resolution evolving intervals by following the previous work [21]. Recall that we aim to discover the co-evolving sensor reading patterns, especially during the pollutant propagation period, which correspond to the uptrend or downtrend intervals of the geo-sensory data. Consequently, we adopt their method as well as the break and segment strategy [10] to extract the uptrend and downtrend intervals. Note that the uptrend and downtrend intervals we extract do not exactly follow Definition 1. Instead, we allow small fluctuations. Figure 3 shows the extracted uptrend and downtrend intervals in blue lines.

4.2 Time Series Clustering

From the previous step, we can get mounts of time series uptrend/downtrend intervals. For each uptrend (downtrend) interval, we adopt the selected time series clustering method to get the co-evolving zones. We will first describe the dissimilarity measures used for time series clustering, then follows the description of the clustering method.

Dissimilarity Measures. The key in the dissimilarity measures, namely, how to define the similarity of two time series object. In our scenario, the spatial constraint also need to be considered to define the spatio-temporal distance function. Two time series objects are similar if they are spatially adjacent and have similar temporal characteristic. It is a function as below.

$$d_{st}(x, y) = \begin{cases} d_t(x, y) & \text{if } x \textit{ and } y \textit{ are spatial neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where, $d_t(x, y)$, is a time-series distance function.

The choice of time series distance function is related to the application. Commonly used time-series distance functions include proximity on value, proximity on behavior or both in the view of what the purpose of the grouping is. The conventional measures ignore the interdependence relationship between measurements, characterizing the time series behavior. The proximity is only based on the closeness of the values, while the proximity on behavior measure the growth behavior of the time series without considering the closeness of the values.

Previous work [5] introduced an adaptive dissimilarity index covering both proximity on values and on behavior, which is able to cover both conventional measures for the proximity on observations and temporal correlation for the behavior proximity estimation. These characteristics make it an ideal dissimilarity measures in our scenario.

First of all, temporal correlation for the behavior proximity estimation has been given. The proximity between the dynamic behaviors of the series is evaluated by means of the first order temporal correlation coefficient, which is defined by

$$CORT(\mathbf{X}_T, \mathbf{Y}_T) = \frac{\sum_{t=1}^{T-1}(X_{t+1} - X_t)(Y_{t+1} - Y_t)}{\sqrt{\sum_{t=1}^{T-1}(X_{t+1} - X_t)^2}\sqrt{\sum_{t=1}^{T-1}(Y_{t+1} - Y_t)^2}} \tag{2}$$

In the above equation, $CORT(\mathbf{X}_T, \mathbf{Y}_T)$ belongs to the interval $[-1, 1]$, The value $CORT(\mathbf{X}_T, \mathbf{Y}_T) = 1$ means that both series show a similar dynamic behavior, i.e., their growths (positive or negative) at any instant of time are similar in direction and rate, while $CORT(\mathbf{X}_T, \mathbf{Y}_T) = -1$ implies a similar growth in rate but opposite in direction (opposite behavior). Finally, $CORT(\mathbf{X}_T, \mathbf{Y}_T) = 0$ expresses that there is no monotonicity between X_T and Y_T, and the growth rates are stochastically linearly independent (different behaviors). In all, $CORT(\mathbf{X}_T, \mathbf{Y}_T)$ gives the similarity measures of time series.

The dissimilarity index proposed by [5] modulates the proximity between the raw-values of two time-series X_T and Y_T using the coefficient $CORT(X_T, Y_T)$. Specifically, it is defined as follows.

$$d_{CORT}(\mathbf{X}_T, \mathbf{Y}_T) = \phi_k[CORT(\mathbf{X}_T, \mathbf{Y}_T)] \cdot d(\mathbf{X}_T, \mathbf{Y}_T) \tag{3}$$

where $\phi_k(\cdot)$ is an adaptive tuning function to automatically modulate a conventional raw-data distance $d(X_T, Y_T)$ according to the temporal correlation. The modulating function should work increasing (decreasing) the weight of the dissimilarity between observations as the temporal correlation decrease from 0 to -1 (increase from 0 to $+1$). In addition, $d_{CORT}(\mathbf{X}_T, \mathbf{Y}_T)$ should approach the raw-data discrepancy as the temporal correlation is zero. In our scenario, we choose an exponential adaptive function given by

$$\phi_k(u) = \frac{2}{1 + exp(ku)}, k \geqslant 0. \tag{4}$$

The above exponential tuning function will cover both proximity on values and behavior, which is an appropriate choice in our situation.

Hierarchical Clustering Groups. Partitioning clustering methods meet the basic clustering requirement of organizing a set of objects into a number of exclusive groups [19], while in our situations we want to partition our data into groups at different levels such as in a hierarchy, which works by grouping data objects into a hierarchy or tree of clusters.

We use the agglomerative hierarchical clustering method based on the bottom-up strategy. It typically starts by letting each object form its own cluster and iteratively merges clusters into larger and larger clusters, until all the objects are in a single cluster or certain termination conditions are satisfied. The single cluster becomes the hierarchys root. For the merging step, it finds the two clusters that are closest to each other (according to some similarity measure), and combines the two to form one cluster. Because two clusters are merged per iteration, where each cluster contains at least one object, an agglomerative method requires at most n iterations.

In our scenario, using the method in [11], we can divide the sensor readings, which have similar proximity on values and behavior, in each change interval into k different co-evolving groups.

4.3 Co-evolving Structures

From the previous step, we get k co-evolving zones in each evolving interval, as shown in the left of Fig. 4. Our problem is to find the relationship between the found co-evolving zones and build the tree structure to show the inner causal associations among the co-evolving zones of different time period. In our situation, we also use hierarchical clustering method to build the co-evolving tree, which is illustrated in the right of Fig. 4.

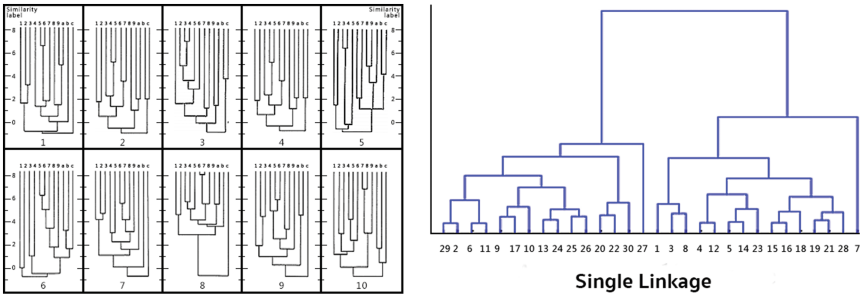

Fig. 4. The left figure is the co-evolving zones found in each co-evolving intervals, while the right figure shows how we restruct the relationship between those zones.

The key here is to define the similarity measures of the co-evolving zones in different time period. Suppose $A = \{A_1, ..., A_k\}$ and $B = \{B_1, ..., B_k\}$ are two co-evolving zones at two time interval of k clusters, the similarity measures of the two co-evolving zones is defined as follows:

Definition 2. *Cluster Similarity Measures. The Cluster Similarity Measures* $Sim(A, B)$ *of two co-evolving zones,* $A = \{A_1, ..., A_k\}$ *and* $B = \{B_1, ..., B_k\}$*, is defined by:*

$$Sim(A, B) = \frac{1}{k} \sum_{i=1}^{k} \max_{1 \leqslant j \leqslant k} Sim(A_i, B_j) \tag{5}$$

where

$$Sim(A_i, B_j) = \frac{|A_i \cap B_j|}{|A_i| + |B_j|} \tag{6}$$

in which $|\cdot|$ denoting the cardinality of the elements in the set.

In the merging step of the Hierarchical Clustering, we use the above similarity measure as the closeness index of two clusters (in our situation, we use single similarity linkage of two clusters) to combine the two to form one cluster.

The Hierarchical Clustering method works by grouping data objects (in our case, the co-evolving zones of different time intervals) into a hierarchy or tree of clusters, which reflects the relationship and inner association between those

co-evolving zones. The groups at different levels of the hierarchy can give us more valuable information about the co-evolving zones, such as the relationship, causal association etc. This would help us have a better understanding of the evolving of the co-evolving patterns and casual association, which can help us make better spatial inference and temporal predictions.

5 Experiment

In this section, we will give the experiment and evaluation of our proposed approach. At first, we give the data and features used in the evaluation Sect. 5.1, which contains all the feature description, then follows the evaluation using real world data Sect. 5.2, we compare the found co-evolving zones with the clustering result using the whole data to test and verify the effectiveness of our method. The found co-evolving structure provides a clear picture of the pollution evolving patterns and has the potential to improve the accuracy of spatial inference and temporal prediction result, even give the recommendation of new air quality stations' locations, which is illustrated in Sect. 5.3.

5.1 Data and Features

We utilize real air quality monitoring datasets collected from Haidian district of Beijing, China. The datasets consist of three parts, as elaborated in the following.

- **Air Quality Records.** The data contains the real-valued AQI of two kinds of pollutions, $PM_{2.5}$ and PM_{10}, measured by almost 200 air quality monitoring stations every 30 min. This dataset is collected over 11 months (from March 1, 2015 to February 1, 2016).
- **Meteorological Data.** Previous study has shown that the concentration of air pollutants is influenced by meteorology. Especially, wind speed, wind power, humidity and barometer pressure all have a big influence on the concentration of the air quality. We choose the four aspects and the weather condition as the five features to evaluate the co-evolving structure result. The fine-grained meteorological data is collected hourly from a public website [1].
- **Point-Of-Interests (POIs).** In the urban area, the land use and the function of the region is well reflected by the category and density of POIs in the area, which is valuable in making accurate spatial inference. In our setting, we extract 8 POI features by using a POI database of Baidu Maps of Beijing (see Table 1).

5.2 Evaluation Using Real World Data

To illustrate the effectiveness of the proposed approprach, we use almost 11 months $PM_{2.5}$ sensor data to evaluate the algoritm. Figure 5 shows the result of almost 2 months data. Figure 5-A shows the mean value of the time series data,

Table 1. Category of POIs

C1: Culture & education	C5: Shopping malls and Supermarkets
C2: Parks	C6: Entertainment
C3: Sports	C7: Decoration and furniture markets
C4: Hotels	C8: Vehicle Services (gas station, repair)

the blue line is the segment result of the uptrend/downtrend detection algorithm, the algorithm get 89 intervals in total. For each change interval, we use the dissimilarity measures defined above to cluster all the sensor readings and divide them into 10 different classes. Then, using the Cluster Similarity Measures, we get the final co-evolving structures, as shown in Fig. 5-B. The co-evolving structure has four obvious sub-clusters: 1, 2, 3, 4. When mapping the sub-clusters into the time dimension, we found a clear temporal correlation, which can be seen in Fig. 5-A. This result shows the heterogeneity of the dynamic co-evolving zones and may provide a novel way to get the appropriate segementation length for future clustering analysis.

Using the above found co-evolving structure and the new segment interval: 1, 2, 3, 4. We get the co-evolving zones for each of the time interval, which is shown in Fig. 5-D, E, F, G. Compared with the clustering result using the whole two months data, as shown in Fig. 5-C, Fig. 5-D, E, F, G show more meaningful and stable results. In this scenario, two month data is too long for the clustering algorithm and the dynamics is smoothed out. While in Fig. 5-D, E, F, G, we can see clear co-evolving zones (the sensors in the same zone show similar patterns in both behavior and value) and the zones are dynamic between two different time intervals, which shows the necessity and effectiveness of the proposed paradigm. In the following section, we will use the found patterns to help improve the accuracy of spatial inference and temporal prediction result and show the effectiveness of the found patterns.

In our experiment settings, we set significant_delta, significant_length in change point algorithm is 35 and 3, and get 508 uptrend/downtrend time intervals. For each time interval, the distance between x, y is below 2 km if they are spatial neighbours, and k is set to 10, which means that there are 10 different sub-clusters for each co-evolving time interval. Using the co-evolving structure clustering algorithm, we get 28 different co-evolving intervals, which all shows an obvious co-evolving zones structure, in the following section, we will use the above found results to evaluate the effect on spatial-temporal prediction result.

5.3 Effect on Prediction and Inference

Spatial Inference. In the previous work [3], we compared the spatial inference accuracy using linear, cubic spline and gaussian process regression method, which shows the effectiveness of Gaussian Process regression in spatial $PM_{2.5}$ concentration inference. However, one big disadvantage of the GP method is the

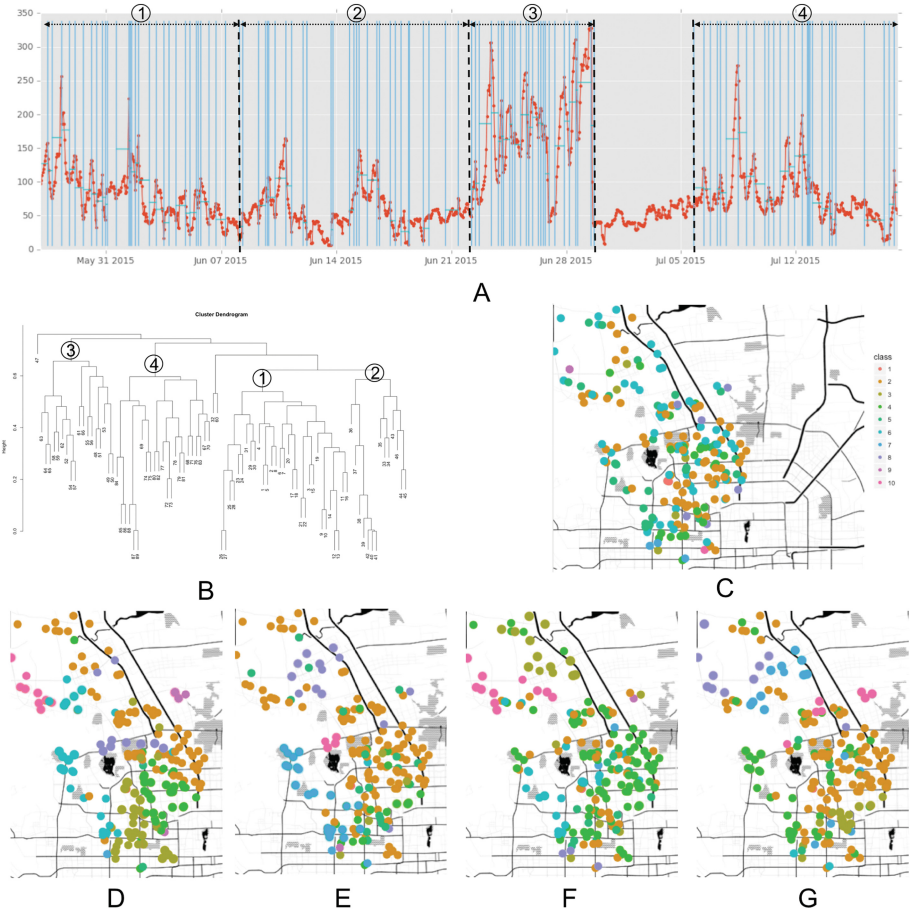

Fig. 5. Evaluation using the real world data (2 months). A: mean value of almost 200 sensors and segementation result; B: final co-evolving structure; C: co-evolving zones clustering result using whole data; D, E, F, G: co-evolving zones clustering result using segmentation intervals 1, 2, 3, 4. (Color figure online)

time complexity. Since an exact inference in Gaussian Process involves computing K^{-1}, the computation cost is $O(n^3)$ (n is the number of the training cases), when the deployment is large (in our situation, almost 200 devices), the compution cost is a big challenge in real time online systems. In this section, we try to decrease the number of devices (n) used for Gaussian Process algorithm with the help of the found co-evolving zones (C-zones).

In experiment, the real deployment dataset of more than 11 month was used to evaluate the performances of the algorithm. There are totally 200 monitor stations deployed in an area with the size of 30 km × 30 km and each station reports its measurements every 30 min, the deployment map is shown in

Fig. 6. (**A**) The deployment map of the monitor stations; (**B**) The distribution of the deviation between station S_1 and S_2 over one month

Fig. 6-(**A**). We deliberately remove one station as ground truth and infer its value using the remaining stations' reading at each timestamp. The Fig. 6-(**B**) also shows the distribution of deviation between our two monitor stations, S_1 and S_2. The geospatial distance of the two stations is about 6 km shown in Fig. 6-(**A**), over 21 % cases have a deviation greater than 100, which also shows the need for an efficient and accurate spatial inference algorithm.

Table 2. Inference errors

Measure / Method	$\|x\|_1$	$\frac{1}{n}\|x\|_1$	$\|x\|_2$	RMSE	$\|x\|_\infty$
Gaussian Process	593429.56	25.12	4322.19	26.74	161.23
Gaussian Process + C-zones	569328.64	17.44	4011.73	20.14	145.08

Table 2 lists the inference errors of the two methods measured via different rules (assume that x is the absolute error vector). Gaussian Process uses all of the devices for training, while Gaussian Process + C-zones only uses the devices in the same co-evolving zones for training process, which only use almost 38 devices in average. From the comparison result, we can see that the inference accuracy has a significant increase by using the co-evolving zones, specially the Chebyshev norm $\|x\|_\infty$ achieved by Gaussian Process is 161.23 while the Gaussian Process + C-zones obtains a smaller value 145.08, which proves that the Gaussian Process + C-zones is more stable in the inference of $PM_{2.5}$ concentrations. The result also shows the efficiency of the found dynamic co-evolving zones.

Temporal Prediction. Over the past decades, some statistic models, like linear regression, regression tree and neural networks, have been employed in

atmospheric science to do a real-time prediction of air quality. However, these methods simply feed a variety of features about a location into a single model to predict the future air quality of the location [22]. In work [22], they use a *Temporal Predictor* to predict the air quality of a station in terms of the data about the station. Instead, the *Spatial Predictor* considers spatial neighbor data, such as the AQIs and the wind speed at other stations, to predict a station's future air quality, The two predictors generate their own predictions independently for a station, which are combined by the *Prediction Aggregator* dynamically according to the current weather conditions of the station. In this way, they improve the prediction accuracy significantly. However, the meteorological data is almost same for devices in dense deployment scenario, using the spatial partition method in work [22] equals to feed all the data from a station's neighbors into a machine learning model. In this way, there are too many inputs for an ANN, leading to too many parameters in the model. Consequently, we cannot learn a set of accurate parameters for the ANN based on the limited training data, which may lead to some problems and can not be used directly in practice (see details in [22]).

In this experiment, we use the devices in the same co-evolving zones as the selected "spatial partition devices" to evaluate the accuracy of the algorithm. Long period prediction may need more data in large scale, so we only evaluate the next 6 h $PM_{2.5}$ concentrations in this experiment, which can be extend to next 48 h prediction in the similar method.

For the next 1–6 h, we measure the prediction of each hour $\widehat{y_l}$ against its ground truth y_i, calculating the accuracy according to Eq. 7, We also calculate the absolute error of each time interval according to Eq. 8, where n is the number of instances measured for a time interval. We random select 30 devices for this evaluation for almost 5 months.

$$p = 1 - \frac{\sum_i |\widehat{y_l} - y_i|}{\sum_i y_i} \tag{7}$$

$$e = \frac{\sum_i |\widehat{y_l} - y_i|}{n} \tag{8}$$

Table 3 shows the prediction result using different methods, LR and ANN only use the local monitor station readings as the data source and make predictions. In general, LR has a similar performance in predicting normal instances but less effective than ANN in dealing with sudden drops. Also, the results presented in Table 3 justify the advantages of the *ANN + C-zones* which use local and devices in same co-evolving zones for prediction which acquires a big improvement in the performance of overall accuracy, especially in the sudden drops scenario.

6 Conclusion

In this paper, we propose a novel divide-and-conquer strategy to find the dynamic co-evolving zones that systematically leverages the sensor readings'

Table 3. Prediction Result of different methods.

Methods	All instances		Sudden drops	
	p	e	p	e
LR (Linear Regression)	0.684	27.5	0.298	103.2
ANN (Artifical Neural Network)	0.646	29.9	0.221	73.7
ANN + C-zones	**0.725**	**20.1**	**0.302**	**51.4**

spatial and temporal heterogeneity challenges. The paradigm produced more stable and meaningful co-evolving zones and automatically found the segmentation intervals which shows the inner pollution change paterns. We use the found result to evaluate the performance on spatial-temporal prediction result and found a significant improvement, which proves the effectiveness of the found patterns. What's more, the found zones and dynamic patterns may provide recommendation for new planned public monitoring stations and future city planning. The system has also been deployed with the Haidian Ministry of Environmental Protection (in Haidian district of Beijing, China) to make accurate spatial-temporal predictions and help the government better understand the pollution evolving patterns to make more scientific strategies for environment treatment. The current implementation still needs manual parameter tuning and has some limitations, for future work, we plan to eliminate those disadvantageous and make the algorithm more scaleable to use in the real production environment.

References

1. aqi.cn: Beijing air pollution. http://aqicn.org/city/beijing/
2. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD Workshop, Seattle, WA, vol. 10, pp. 359–370 (1994)
3. Cheng, Y., Li, X., Li, Z., Jiang, S., Jiang, X.: Fine-grained air quality monitoring based on gaussian process regression. In: Loo, C.K., Yap, K.S., Wong, K.W., Teoh, A., Huang, K. (eds.) ICONIP 2014, Part II. LNCS, vol. 8835, pp. 126–134. Springer, Heidelberg (2014)
4. Cheng, Y., Li, X., Li, Z., Jiang, S., Li, Y., Jia, J., Jiang, X.: Aircloud: a cloud-based air-quality monitoring system for everyone. In: Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems, pp. 251–265. ACM (2014)
5. Chouakria, A.D., Nagabhushan, P.N.: Adaptive dissimilarity index for measuring time series proximity. Adv. Data Anal. Classif. **1**(1), 5–21 (2007)
6. Eiter, T., Mannila, H.: Computing discrete fréchet distance. Technical report, Citeseer (1994)
7. Fréchet, M.M.: Sur quelques points du calcul fonctionnel. Rendiconti del Circolo Matematico di Palermo (1884–1940) **22**(1), 1–72 (1906)
8. Kawahara, Y., Sugiyama, M.: Change-point detection in time-series data by direct density-ratio estimation. In: SDM, vol. 9, pp. 389–400. SIAM (2009)
9. Keogh, E., Chu, S., Hart, D., Pazzani, M.: An online algorithm for segmenting time series. In: Proceedings IEEE International Conference on Data Mining, 2001, ICDM 2001, pp. 289–296. IEEE (2001)

10. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: a survey and novel approach. Data Min. Time Ser. Databases **57**, 1–22 (2004)
11. Langfelder, P., Zhang, B., Horvath, S.: Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for r. Bioinformatics **24**(5), 719–720 (2008)
12. Marti, G., Andler, S., Nielsen, F., Donnat, P.: Clustering financial time series: How long is enough? arXiv preprint arXiv:1603.04017 (2016)
13. Montero, P., Vilar, J.A.: Tsclust: an r package for time series clustering. J. Stat. Softw. **62**(1), 1–43 (2014)
14. Sankoff, D., Kruskal, J.B.: Time warps, string edits, and macromolecules: the theory and practice of sequence comparison. Addison-Wesley Publication, Reading (1983). Edited by Sankoff, D., Kruskal, J.B
15. Sharifzadeh, M., Azmoodeh, F., Shahabi, C.: Change detection in time series data using wavelet footprints. In: Medeiros, C.B., Egenhofer, M., Bertino, E. (eds.) SSTD 2005. LNCS, vol. 3633, pp. 127–144. Springer, Heidelberg (2005)
16. Trasarti, R., Olteanu-Raimond, A.M., Nanni, M., Couronné, T., Furletti, B., Giannotti, F., Smoreda, Z., Ziemlicki, C.: Discovering urban and country dynamics from mobile phone data with spatial correlation patterns. Telecommun. Policy **39**(3), 347–362 (2015)
17. Wang, P., Wang, H., Wang, W.: Finding semantics in time series. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, pp. 385–396. ACM (2011)
18. Warren Liao, T.: Clustering of time series data-a survey. Pattern Recogn. **38**(11), 1857–1874 (2005). http://dx.doi.org/10.1016/j.patcog.2005.01.025
19. Xu, R., Wunsch, D., et al.: Survey of clustering algorithms. IEEE Trans. Neural Netw. **16**(3), 645–678 (2005)
20. Yamanishi, K., Takeuchi, J.i.: A unifying framework for detecting outliers and change points from non-stationary time series data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 676–681. ACM (2002)
21. Zhang, C., Zheng, Y., Ma, X., Han, J.: Assembler: efficient discovery of spatial co-evolving patterns in massive geo-sensory data. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1415–1424. ACM (2015)
22. Zheng, Y., Yi, X., Li, M., Li, R., Shan, Z., Chang, E., Li, T.: Forecasting fine-grained air quality based on big data. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2267–2276. ACM (2015)