Spatiotemporal Neighborhood Discovery for Sensor Data

Michael P. McGuire[1,2], Vandana P. Janeja[2], and Aryya Gangopadhyay[2]

[1] Center for Urban Environmental Research and Education, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA

[2] Information Systems Department, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA

mcguire1@umbc.edu, vjaneja@umbc.edu, gangopad@umbc.edu

Abstract. The focus of this paper is the discovery of spatiotemporal neighborhoods in sensor datasets where a time series of data is collected at many spatial locations. The purpose of the spatiotemporal neighborhoods is to provide regions in the data where knowledge discovery tasks such as outlier detection, can be focused. As building blocks for the spatiotemporal neighborhoods, we have developed a method to generate spatial neighborhoods and a method to discretize temporal intervals. These methods were tested on real life datasets including (a) sea surface temperature data from the Tropical Atmospheric Ocean Project (TAO) array in the Equatorial Pacific Ocean and (b)highway sensor network data archive. We have found encouraging results which are validated by real life phenomenon.

1 Introduction

Sensors are typically used to measure a phenomenon in terms of readings associated with a specific location for example: i) environmental sensors - to monitor quality, temperature etc. in air, water or land, ii) traffic monitoring sensors - to monitor congestion on highways, iii) comparative vacuum monitoring sensors - to monitor the structural stability of bridges, to name a few. Such sensors can be considered as *spatial* objects generating measurements over a period of time (*temporal*). A key to effective knowledge discovery tasks (such as outlier detection, pattern discovery etc.) is to first identify a group of sensors which may be characterized similarly based on their spatial proximity and temporal measurements. For instance, an outlier sensor in a set of traffic sensors is one which is unusual with respect to its nearby sensors. This characterization of similar sensors where the data is spatiotemporal in nature is termed as the *spatiotemporal neighborhood*. In this paper our focus is the discovery of spatiotemporal neighborhoods. Our approach consists of three components:

M.M. Gaber et al. (Eds.): Sensor-KDD 2008, LNCS 5840, pp. 203–225, 2010.

- Defining spatial neighborhoods
- Discretizing temporal intervals
- Combining spatial neighborhoods with temporal intervals to generate spatiotemporal neighborhoods.

Our notion of a spatiotemporal neighborhood is distinct from the traditional notions since we consider both a spatial characterization as well as a temporal characterization in the formation of neighborhoods.

Traditionally, spatial neighborhoods are defined as a group of objects that are in spatial proximity to each other that have similar non-spatial attributes [21] [6]. A particular challenge in this research is to extend this definition to include non-spatial attribute values in the formation of the neighborhoods and to account for neighborhood boundaries that are not crisp.

If there is a vast number of measurements over a period of time associated with each spatial object it is not feasible to analyze every value in such a complex time series. Thus, a temporal characterization must discretize [13] a time series in such a way that the resulting intervals represent distinct temporal features within which knowledge discovery can be focused. Therefore, we define a temporal interval as a segment of time that has similar measurement characteristics. The method to generate temporal intervals must be able to handle the complexity that is often found in real world datasets. This is particularly a challenge in situations where divisions between intervals are not easily deduced and the number of temporal intervals is not known before hand.

The individual challenges of generating spatial neighborhoods and temporal intervals are compounded when combined to form spatiotemporal neighborhoods. A particular challenge is to be able to track spatial change over time. Just as it is not feasible to analyze every value in a complex time series, it is even more problematic to analyze spatial patterns at every time step in a dataset. Because of this, a major challenge of the spatiotemporal neighborhood approach will be to find the temporal intervals where changes in spatial patterns occur.

This research is applicable to a number of domains including transportation planning, climatology, meteorology, hydrology, and others. We next present two motivating examples:

Example 1. Climatology: The TAO/TRITON array [19] consists of sensors installed on buoys positioned in the equatorial region of the Pacific Ocean. The sensors collect a wide range of meteorological and oceanographic measurements.Sea Surface Temperature (SST) measurements are reported every five minutes. Over time, this results in a massive dynamic spatiotemporal dataset. This data played an integral part of characterizing the 1997-98 El Nino [17] and is currently being used to initialize models for El Nino prediction. There have been a number of studies which assimilate meteorological and oceanographic data to offer a description of the phenomena associated with the events of the 1982-83 El Nino [4] [20] and the 1997-1998 El Nino [17]. These analyses show a particular importance in the spatiotemporal patterns of meteorological variables and SST anomalies that characterize El Nino events.

El Nino events are most often characterized by anomalously high values of SST in the eastern Pacific from 160 degrees west eastward to the coast of South America. Daily anomalies are typically calculated using a combination of in situ and satellite measurements where the degree of the anomaly is based on the difference between the current SST analysis value and SST monthly climatology. This method finds global outliers at a relatively high spatial resolution. However, if a scientist would like to see outliers at higher temporal resolutions than the daily average, a dataset with a higher temporal frequency, such as data from the TAO / TRITON network, is needed. This data consists of a vast time series collected at 44 sensors across the equatorial Pacific Ocean. The challenge from the scientist's perspective is first to find the sensors in the TAO network that are proximal and have similar SST measurements. To make the analysis more efficient, the scientist would like to automatically find areas in the data where changes to the spatial patterns are most likely to occur and focus the analysis on finding anomalies in these areas.

Example 2. Traffic Monitoring: Traffic congestion is a common problem in urban areas. The duration and intensity of congestion has grown over the last 20 years [2]. Because of this, transportation planners are continually devising strategies to combat congestion. Many highway systems are now employing Intelligent Transportation Systems (ITS) and have sensors which monitor traffic conditions. These sensors allow traffic engineers to understand the dynamics of traffic in multiple locations on the highway network and in turn offer insight into the spatiotemporal patterns of congestion. There are a number of traffic control measures that can be employed to reduce congestion. But to arrive at an optimal solution, traffic engineers must understand where congestion exists in order to determine locations to introduce traffic control measures. In this situation, knowing the spatiotemporal pattern of congestion would be extremely useful. Furthermore knowing the spatiotemporal characterization would allow the traffic engineer to identify anomalies that occur during peak period and off peak period hours and provide a better understanding of the dynamics that cause congestion and result in new strategies to deal with congestion problems.

Key Contributions: From these motivating examples we can identify the following key contributions of our work in discovering the spatiotemporal characterization which we refer to as the spatiotemporal neighborhood for complex sensor data.

Spatial Neighborhoods: While generating spatial neighborhoods it is essential to find the spatial distribution of measurements at individual locations in combination to the spatial relationships between locations. One important challenge in identifying the spatial neighborhoods in such application domains is that they do not have crisp boundaries. Thus a key contribution of this work is to accommodate for overlapping neighborhoods.

Temporal Intervals: These intervals embody the concept of neighborhoods in time (similar to spatial neighborhoods in space). A major contribution of this work is to create unequal width or unequal frequency intervals that are robust in the presence of outliers.

Spatiotemporal Neighborhood: There have been a number of approaches in the literature which model spatiotemporal patterns using a graph-based approach [14] [8] [5]. However, no existing approach includes finding temporal intervals in time series data to pinpoint temporal intervals where the spatial pattern changes. In this case, it becomes critical to accommodate the individual properties of spatial and temporal neighborhoods to identify points in time where the spatial pattern is most likely to change and identify temporal patterns at many spatial locations.

In this paper we propose a method to generate spatiotemporal neighborhoods. This is accomplished by first performing an initial spatial characterization of the data; then defining distinct temporal intervals; and finally by defining spatial neighborhoods at each interval. We discuss experiments on real world datasets on SST and traffic data with promising results in finding spatial neighborhoods and distinct temporal intervals in both datasets.

The rest of the paper is organized as follows. In section 2 we discuss our approach. In section 3 we outline our experimental results. Section 4 discusses related work and finally in section 5 we conclude and discuss some challenges for future research.

2 Approach

The overall approach is outlined in figure 1, which comprises of the following distinct steps:

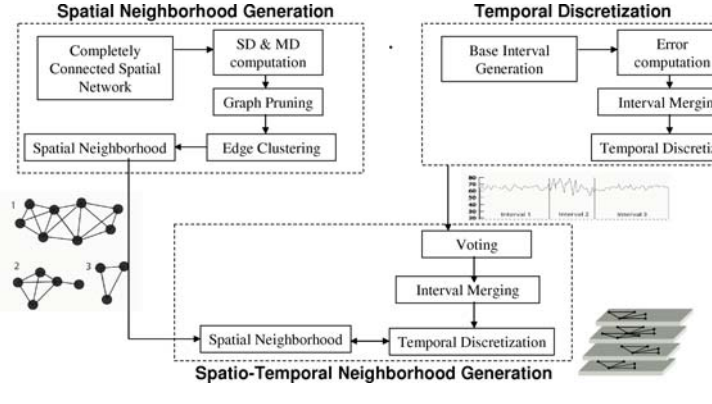


Fig. 1. Spatiotemporal Neighborhood Generation

1) Spatial Neighborhood Generation: We begin by creating the spatial neighborhoods using a graph-based structure derived from the relationships between the spatial nodes in terms of their spatial proximity and measurement similarity.

2) Temporal Interval Generation: We use agglomerative clustering to generate temporal intervals in a time series dataset comprised of measurements collected at a spatial node. For this we start with temporal intervals of a preset small size and merge contiguous intervals with similar within-interval statistics resulting in a set of unequal width intervals representing distinct sections of the time series.

3) Spatiotemporal Neighborhood Generation: Spatial neighborhoods and temporal intervals are then used as building blocks to generate the spatiotemporal neighborhoods.

2.1 Spatial Neighborhood Generation

A spatial neighborhood is defined as a group of spatial nodes that are within proximal distance of each other and exhibit similar characteristics. Before we formally define our concept of spatial neighborhood we define some spatial primitives:

Definition 1 (Spatial Node). *Let S be a set of spatial nodes $S = \{s_1, ..., s_n\}$ where each $s_i \in S$ has a set of coordinates in 2D Euclidean space (s_{ix}, s_{iy}) and a set of attributes $A_i = \{s_{ia1}, \ldots, s_{iam}\}$.*

To define a spatial neighborhood we first consider the spatial proximity as defined by spatial relationships:

Definition 2 (Spatial Relationship). *Given two spatial nodes $(s_p, s_q) \in S$ a spatial relationship $sr(s_p, s_q)$ exists if there exists a distance, direction or topological relationship between them.*

For instance the spatial relationships may be qualified using a distance relationship based on the following concept of Spatial distance:

Definition 3 (Spatial Distance). *The spatial distance $sd(s_p, s_q)$ is calculated as the Euclidean distance between two spatial coordinates such that*

$$sd = \sqrt{(s_{px} - s_{qx})^2 + (s_{py} - s_{qy})^2}$$

In addition to the spatial relationship we also quantify the similarity between nodes based on the distance between the measurement values(or the non-spatial attributes) of the spatial nodes as follows:

Definition 4 (Measurement Distance). *The measurement distance* $md(s_p, s_q)$ *is the Euclidean distance between the set of normalized numerical attributes* A_p *and* A_q *at* s_p *and* s_q *such that*

$$md = \sqrt{\sum_{1}^{m}(s_{pam} - s_{qam})^2}$$

for m *attributes measured at each spatial node.*

Thus a spatial neighborhood is defined as follows:

Definition 5 (Spatial neighborhood). *Given a set of spatial nodes* $S = \{s_1, \ldots, s_n\}$ *a spatial neighborhood* $spn = \{sp_1, \ldots, sp_l\}$ *such that* $spn \subset S$ *where* $\forall sp_i \in spn$ *exhibits* $sd(sp_i, sp_j) < d$*, where* d *is a spatial distance threshold and* $md(sp_i, sp_j) < \delta$ *where* δ *is a measurement distance threshold.*

Our spatial neighborhood method uses a graph-based structure to model the data such that a spatial neighborhood graph $SG = sg, < e >$ where sg is a set of nodes $\in spn$ such that for all pair of nodes $(s_i, s_j) \in sg$ there exists an edge $< e_i, e_j > \in e$.

In this neighborhood graph, the edges form relationships between the spatial nodes such as the spatial distance between two nodes or the distance between measurements taken at two nodes. For example, this could be the distance between SST measurements taken at two neighboring sensors. Figure 2 shows an illustrative example of graph-based spatial neighborhoods.

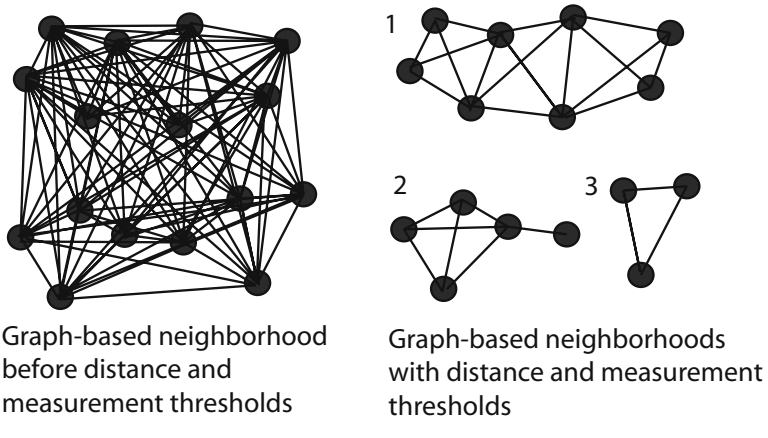


Fig. 2. Graph-based Spatial Neighborhoods

On the left, the measurement of spatial nodes is shown and all possible relationships between the nodes are shown as edges. The right shows three neighborhoods that are formed after applying the distance and measurement thresholds.

Neighborhood 1 shows a contiguous group of sensors that are connected by being close in proximity and having similar measurement values. Neighborhoods 2 and 3, while proximal to each other are divided by the measurement threshold. Notice the sensor that falls in the middle of neighborhood 2 and 3. This sensor is close in proximity to nodes in both neighborhoods however, because of the measurement threshold, it is more similar to the nodes in neighborhood 2.

The ultimate goal of this approach is to find spatial groups in the data that are also based on non-spatial attributes. To do this we apply clustering to the non-spatial attributes of the remaining edges. Clustering is also used because in some cases, the edges that remain after applying the d and md thresholds do not form discrete neighborhood divisions. For example, if a node is within d of two neighborhoods and has a md that is less than δ from a node in each neighborhood, this node will connect the two neighborhoods and therefore finds non-crisp neighborhood boundaries. Clustering can address this if crisp boundaries are required because it will assign edges to neighborhoods based on the mean measurement value between the two nodes. The nodes of the resulting clusters are then extracted to form the spatial neighborhoods. The neighborhood quality is then measured where the measurement values of the nodes are compared to the mean measurement value of the spatial neighborhood.

Definition 6 (Neighborhood Quality). *We use a within-neighborhood sum of squared error (SSE)function applied to the set of attributes A_i for each s_i to measure the spatial neighborhood quality nq such that:*

$$nq = \sum_{i=1}^{n} (s_{iam} - \mu^{spn}))^2 / n$$

where s_{iam} are the attribute values for each s_i and μ^{spn} is the mean measurement value for the entire spatial neighborhood.

The nq is divided by n to normalize the value so that it can be compared across neighborhoods of varying sizes.

The Spatial Neighborhood generation is outlined in Algorithm 1. The algorithm requires a set of spatial nodes and corresponding attributes and threshold values for the spatial and measurement distance between spatial nodes. These thresholds are used as heuristics to control the relationships between spatial nodes. For example if two spatial nodes are too far apart but have similar measurement values, the edge would be removed from the clustering.

2.2 Temporal Interval Generation

In this section, we present an agglomerative approach to generate temporal intervals from a set of temporal measurements. Figure 3 gives an illustrative example of this approach.

The agglomerative approach first divides the time series into a set of base equal frequency temporal intervals. In general, a temporal interval is defined as:

Algorithm 1. Procedure: Graph-based Spatial Neighborhood Generation

Require: A set of spatial nodes $S = \{s_1, \ldots, s_n\}$
Require: A spatial distance threshold d
Require: A measurement distance threshold δ
Require: Number of clusters C
Ensure: A set of spatial neighborhoods $spn = [NodeID, NeighborhoodID]$
//Initialize the graph and calculate pairwise euclidean distance
for i = 1 to n **do**
COUNT = n-i
for j = 1 to COUNT **do**
Add edges i,$i + j$ to create $n(n-1)/2$ edge matrix
CALCULATE sd, md, and mean measurement between node i and $i + j$ and add to edge matrix
end for
end for
//Apply distance and measurement thresholds to graph
SelectedEdges = edges($sd < d$ AND $md < \delta$)
//Cluster edges based on measurement values
CIndex = K-Means(mean measurement,C)
EdgeCluster = CONCATENATE(SelectedEdges,CIndex)
//Assign nodes to neighborhoods based on CIndex
for each selected edge s **do**
for each cluster C **do**
if EdgeCluster(s) = C **then**
Membership(s) = Nodes in EdgeCluster(s)
Remove duplicate Node IDs from Membership(s)
end if
end for
end for
for each neighborhood N **do**
CALCULATE nq //Calculate neighborhood quality
end for

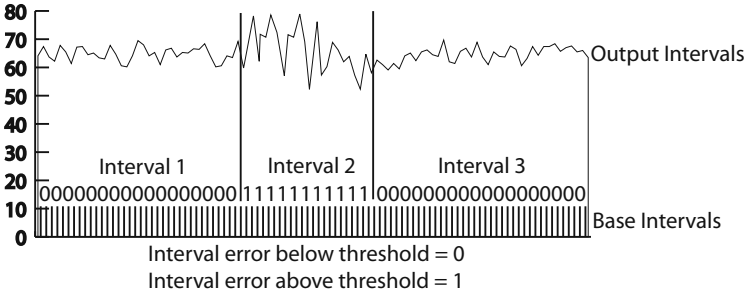


Fig. 3. Agglomerative Temporal Discretization

Definition 7 (Temporal Interval). *Given a set of temporal measurements* $T = \{t_1, \ldots, t_n\}$ *a temporal interval* $int = \{t_1, \ldots, t_m\}$ *is a division of* T *such that* $int \subset T$ *and* $int_1 < int_2, \ldots, < int_k$, *where each* $int^i =< int^i_{start}, int^i_{end} >$ *such that the size* $int^i_{size} = (int_{start} - int_{end})$.

We would like to create intervals where the size of the interval is variable (unequal width intervals). In order to create such intervals we begin first with base intervals int^{base} where the size int^{base}_{size} is fixed to begin with and is a user defined

parameter which largely depends on the domain and granularity of the analysis. We calculate SSE for each base interval as follows:

Definition 8 (SSE). *The SSE is the sum of the squared differences of each value within* int^{base} *from the mean of all values in* int^{base} *such that:*

$$SSE = \sum_{bn=1}^{BN} dist(int_{bn}^{base} - \mu_{int}^{base})^2$$

Here bn is each temporal reading in the total BN readings for the base interval. Then for each base interval, the SSE value is given a binary classification which assigns base intervals as having either a high or low within-interval error. The binary interval error is defined as follows:

Definition 9 (Binary Interval Error). *A binary interval error* $\epsilon = (1, 0)$ *such that if* $SSE(int) > \lambda$ *then* $\epsilon = 1$ *else* $\epsilon = 0$

Here $\epsilon = 1$ is a high error and $\epsilon = 0$ is a low error. This error is applied to each int^{base} by using an error threshold λ such that if $SSE(int) > \lambda$ the interval is classified as 1 and if $SSE(int) < \lambda$ the interval is classified as 0.

Based on the binary interval error we merge the base intervals into larger intervals such that consecutive groups have similar error classification. This results in a set of variable width temporal intervals defined by the within-interval error. This method is flexible in that any statistical measure can be used for within-interval error. Currently as an example, we have used SSE.

Algorithm 2. Procedure:Temporal Interval Generation

Require: Time series measurements ts and its instances $t_1, t_2, \ldots, t_n$
where $t \in ts$ and $t_1 < t_2 < t_n$
Require: base temporal interval size I
Require: error threshold λ
Ensure: Set of variable width temporal intervals $I = i_1, \ldots, i_n$
where each $i = start, end, error$
//Create base temporal intervals and calculate SSE
Interval Start = 1
Interval End = Interval Start + I
while Interval Start < length(ts) **do**
CALCULATE SSE for interval
end while
//Apply Binary Error Classification
for each i in I **do**
if interval SSE ¡ λ **then**
ErrorGroup(t) = 0
else
ErrorGroup(t) = 1
end if
end for
//Merge binary classification to create temporal intervals
for each i in I **do**
if ErrorGroup(t) ≠ ErrorGroup(t+1) **then**
Add Interval Start and Interval End to output
end if
end for

The agglomerative method is formalized in Algorithm 2. The algorithm requires as input a time series ts, a base temporal interval size, and a minimum

error threshold λ that is used to merge intervals. The output of the algorithm is a set of variable width temporal intervals defined by columns representing the interval start, interval end, and interval error.

2.3 Spatiotemporal Neighborhood Generation

Space and time are most often analyzed separately rather than in concert. Many applications collect vast amounts of data at spatial locations with a very high temporal frequency. For example, in the case of SST, it would not be possible to comprehend 44 individual time series across the equatorial Pacific Ocean. Furthermore, to look at the change in spatial pattern at each time step would also be confusing because it would require a large number of map overlays. The challenge in this case is to find the temporal intervals where the spatial neighborhoods are likely to experience the most change in order to minimize the number of spatial configurations that need to be analyzed.

In our method for spatiotemporal neighborhoods we have incorporated both of the above approaches into an algorithm that generates the temporal intervals where spatial patterns are likely to change and for each interval generates spatial neighborhoods. The combined result of this algorithm is a characterization of the spatiotemporal patterns in the dataset.

Because of the addition of a time series to the spatial dataset, the spatiotemporal algorithm has a number of subtle differences from the above approaches. The first is that a long time series makes it less efficient to calculate the md and mean measurement value at the same time as sd. Therefore threshold d is applied first and the md and mean measurement values are calculated only for the proximal edges.

The spatiotemporal algorithm also requires an additional step to deal with time series at many spatial nodes. After the binary error classification is created for each time series at each spatial node, the time series has to be combined to form temporal intervals that can be applied to all spatial nodes. To accomplish this task, we have implemented a voting function to count for each base temporal interval, the number of spatial nodes that have an error classification. The voting function counts for each int the number of spatial nodes that have a binary error classification of 1. This results in the total number of base intervals that have high error values.

A threshold mv is then applied to the result of the voting algorithm where mv represents the minimum number of votes for a temporal interval to be considered a high error interval for all spatial nodes. The application of mv converts the result of the voting algorithm back to a binary matrix by giving each $int_{votes} > mv$ a value of 1 and each $int_{votes} < mv$ a value of 0. These intervals are then merged using the same method as in the agglomerative temporal interval algorithm. This results in a set of temporal intervals for which the md and measurement values for each edge are averaged. Once the temporal intervals are created, the δ threshold is applied to the mean md for each edge in each interval

resulting in a selected set of edges for each temporal interval. Then the edges are clustered for each interval and the spatial nodes are assigned to their respective spatial neighborhoods. The spatiotemporal neighborhood generation algorithm is presented in Algorithm 3.

Algorithm 3. Algorithm for Spatiotemporal Neighborhoods

Require: A set of spatial nodes $S = [s_1, \ldots; s_n]$ where
each s_i has a time series of measurements T and its instances $[t_1, t_2, \ldots, t_n]$
where $t \in T$ and $t_1 < t_2 < t_n$
Require: A spatial distance threshold d
Require: A measurement distance threshold δ
Require: A base temporal interval size I
Require: An interval error threshold λ
Require: A minimum number of votes threshold mv
Require: Number of clusters C
Ensure: A set of spatiotemporal neighborhoods STN = [IntervalStart,IntervalEnd,NodeID,NeighborhoodID] //Procedure: Graph-based Spatial Neighborhood Generation //Procedure: Temporal Interval Generation //Procedure: Create spatiotemporal graph
for each t in ts **do**
if SUM(ErrorGroup(t))¡mv **then**
IntervalError(t) = 0 //Apply voting function
else
IntervalError(t) = 1
end if
end for
for each interval i = 1 to number of intervals **do**
if IntervalError(i) $\neq$ IntervalError($i+1$) **then**
Add Interval Start and Interval End to output matrix IntInterest //Merge binary classification to create temporal intervals
end if
end for
//Form spatial neighborhoods for each interval
for each IntInterest I **do**
for each proximal edge p **do**
p_{md} = MEAN(md) //Calculate mean md for each interval
if $p_{md} < \delta$ **then**
SelectedEdges = ProximalEdges //Apply δ to mean md of edges at each temporal interval
end if
end for
end for
for each IntInterest I **do**
CIndex = K-Means(edge mean measurement value,C) //Cluster edges based on measurement values
EdgeCluster = CONCATENATE(SelectedEdges,CIndex)
end for
for each IntInterest$I$ **do**
for each selected edge s **do**
for each C **do**
if EdgeCluster(s) = C **then**
Membership(C) = Nodes in EdgeCluster(s) //Assign nodes to neighborhoods based on CIndex
Remove duplicate values from Membership(C)
end if
CALCULATE nq //Calculate neighborhood quality
end for
end for
end for

3 Experimental Results

Our experimental results are organized as follows:

- Spatial Neighborhood discovery
- Temporal Interval discovery
- Spatiotemporal Neighborhood discovery

We utilized two datasets Sea Surface Temperature Dataset(SST) and Maryland Highway Taffic Dataset. In the following section we outline these two datasets, discuss the results of the spatial, temporal, and spatiotemporal neighborhoods. Finally for each dataset we provide ground truth validations based on real-world phenomenon.

3.1 Datasets

SST Data: The algorithms were tested on sea surface temperature data from the Tropical Atmospheric Ocean Project (TAO) array in the Equatorial Pacific Ocean [19]. These data consisted of measurements of sea surface temperature (SST) for 44 sensors in the Pacific Ocean where each sensor had a time series of 1,440 data points. The format of the SST data shown in Table 1 has columns for latitude, longitude, data, time (GMT), and SST in degrees Celsius.

Table 1. Sea Surface Temperature Data Format

Latitude	Longitude	Date	Time	SST(degrees C)
0	-110	20040101	000001	24.430
0	-140	20040101	000001	25.548
0	-155	20040101	000001	25.863
...	...	...	...	...

The temporal frequency of the data is 15 minutes. The SST data was used to demonstrate methods for spatial neighborhoods, temporal intervals, and spatiotemporal neighborhoods.

Traffic Data: The algorithms were also tested using average traffic speed from a highway sensor network data archive operated by the Center for Advanced Transportation Technology Laboratory at the University of Maryland, College Park [7]. The format of the traffic data shown in Table 2 consists of columns for date and time, direction, location, and average speed in miles per hour. The temporal frequency of the data is 5 minutes and consisted of approximately 2,100 data points for each sensor. This data was used to test graph-based spatial neighborhood, agglomerative temporal interval, and spatiotemporal neighborhood algorithms.

Table 2. Average Traffic Speed Data Format

Date Time	Direction	Location	Speed(mph)
1/2/2007 0:01	East	US 50 @ Church Rd	79
1/2/2007 0:06	East	US 50 @ Church Rd	81
1/2/2007 0:11	East	US 50 @ Church Rd	61
...	...	...	...

3.2 Spatial Neighborhood Discovery

The graph-based spatial neighborhood algorithm was applied to both SST and traffic data. In this section the preliminary results of this analysis are presented.

SST Data: Figure 4 shows the edge clustering of the spatial neighborhood for the TAO array.

Ground Truth Validation: The resulting edge clustering is validated by the satellite image of SST where the light regions represent cooler temperatures and the dark regions represent warmer temperatures. The edges in Figure 4(a) represent cooler water that extends from the southwestern Pacific shown in lower right part of the SST image and extends westward along the equator. The cluster shown in Figure 4(b) represents the warm waters of the southwestern Pacific shown in the lower left part of the image. The clusters in Figure 4(c) and (d) represent more moderate temperature regions that fall in between the extremes of clusters (a) and (b). A depiction of the nodes colored by neighborhood is shown in Figure 5.

The neighborhoods shown in Figure 5(a), (b), (c), and (d) directly reflected the result of the edge clustering and thus were also validated by the pattern of SST shown in the satellite image background. Figure 5(e) refers to nodes that had edges that are connected to nodes from multiple neighborhoods. These nodes represent locations where the neighborhoods overlap and, as would be expected, typically occur along neighborhood boundaries. This illustrates the continuous nature of SST data and a major challenge to defining spatial neighborhoods in

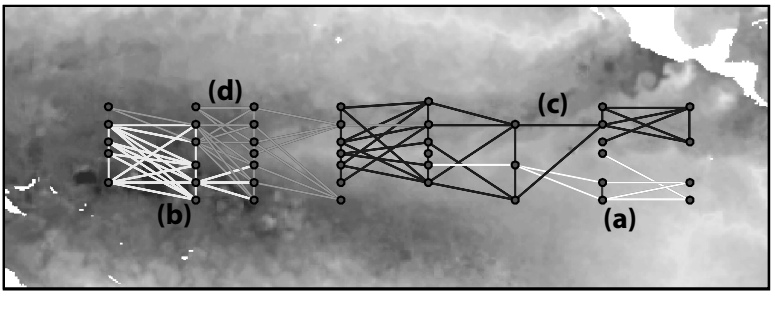


Fig. 4. Result of edge clustering for SST in the Equatorial Pacific

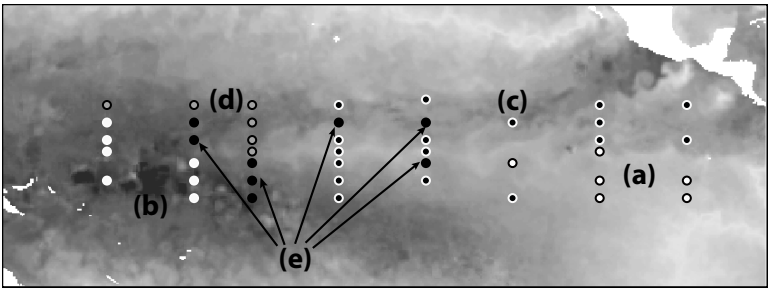


Fig. 5. Graph-based neighborhoods for SST in the Equatorial Pacific

that the spatial patterns are more represented by gradual changes in SST rather than well defined boundaries.

The last step in the algorithm was to calculate the neighborhood quality using the SSE/n of the measurements taken at the nodes within the neighborhood. The neighborhood quality for the above neighborhoods is shown in Table 3.

Table 3. Graph-based Neighborhood Quality for SST Data

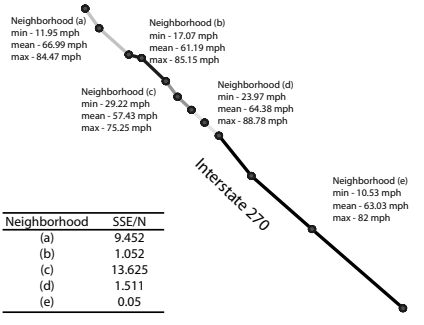
Neighborhood	SSE/n
(a)	0.338
(b)	0.169
(c)	0.286
(d)	0.116

The quality values show that the within-neighborhood error was relatively low and that neighborhoods (b) and (d) had less error than neighborhoods (a) and (c). This suggests that there is more variability in neighborhoods (a) and (c) and that the higher error values suggest that the inner spatial structure of the neighborhoods requires further investigation.

Traffic Data: The graph-based approach also lends itself well to data that is distributed along a directional network such as traffic data. A few modifications had to be made to the algorithm to find distinct neighborhoods in the network data. First, because the nodes and edges are predefined, only linear edges need to be created to successively connect the nodes. To do this, the edges are sorted by the order that they fall on the directional network so that the nodes are connected in sequential order. This removes the complexity of the first step in the algorithm in that a pairwise distance function is not needed to calculate the sd, md, and mean measurement value. Also, because the edges are predefined by a network, there is no need for thresholds to prune edges that have high spatial and measurement distances. Moreover, because the nodes are connected by only one segment, two similar neighborhoods that are separated by a neighborhood

that is not similar are not connected and thus should be represented as separate neighborhoods. Because of this, the result of the clustering algorithm had to be post-processed to assign a new neighborhood ID to similar but unconnected edges. To do this, we looped through the cluster index and assigned nodes to a new neighborhood each time the cluster ID changed.

The algorithm was run on traffic data from 12 sensors located on Interstate 270 South from Frederick, Maryland to the Washington D.C. Beltway (Interstate 495). A one month period of data was used. This consisted of approximately 3,000 records for each sensor. Weekends and holidays were excluded because we wanted the spatial neighborhoods to reflect the peak periods found in the data. Peak periods are typically absent during weekends and holidays. Because of the nature of traffic patterns in terms of periods of jams and free flow, the k-means clustering was run on the minimum, mean, and maximum speed along each edge. The result of the algorithm and the neighborhood quality is shown in Figure 6.



Neighborhood	SSE/N
(a)	9.452
(b)	1.052
(c)	13.625
(d)	1.511
(e)	0.05

Fig. 6. Graph-based neighborhoods for traffic data - I-270 south from Frederick to Washington Beltway

Ground Truth Validation: According to the results the I-270 corridor is characterized by five traffic neighborhoods. Starting in Frederick to the northwest, the first two neighborhoods appear to have a much lower minimum speed. This indicates the presence of at least one very severe traffic jam. As traffic moves to neighborhood (c), the minimum speed speeds up and continues into neighborhood (d) because the highway goes from two to four lanes in this area. Finally in neighborhood (e), the minimum speed indicates the presence of a severe traffic jam neighborhood which reflects congestion in this area caused by the Washington D.C. Beltway. The neighborhood quality is very interesting in this example. It shows that neighborhoods (a) and (c) are different in terms of their within-neighborhood error. This indicates that these neighborhoods need to be investigated further to determine the cause of this result.

3.3 Temporal Interval Discovery

The agglomerative temporal interval algorithm was tested on both the SST and traffic datasets. For the traffic and SST data we used an error threshold(λ) of

1 standard deviation from the mean SSE for all intervals and the base interval size was 20.

SST Data: The sea surface temperature data was collected at one sensor in the TAO array located at 0 degrees north latitude and 110 degrees west longitude. For this sensor, SST is measured every 15 minutes and in this demonstration, a 10 day period was used from 01/01/2004 to 01/10/2004. This consisted of approximately 1400 measurements. The result of the agglomerative algorithm for the SST data is shown in Figure 7.

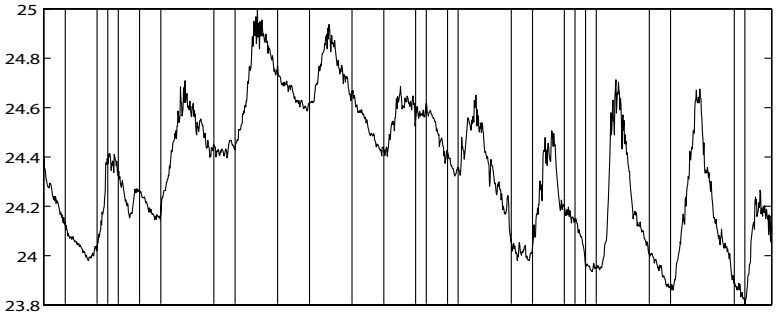


Fig. 7. Agglomerative temporal intervals for SST data

Ground Truth Validation: The temporal intervals are validated by the SST time series in the figure. It is evident that the algorithm was able to differentiate peak periods in the SST data from more stable periods. However, it is also evident that in some cases noise in the data causes a 1-0-1 pattern in the binary error classification whereby the base temporal intervals are exposed.

Traffic Data: The traffic data was taken from the intersection of east bound US Route 50 and Church Road in Maryland. This data consisted of average speed at 5 minute intervals for the period of 11/03/2007 to 11/10/2007. The size of the dataset was approximately 2100 measurements. The intervals for the traffic data are shown in Figure 8.

Ground Truth Validation: The algorithm was extremely effective in identifying periods of traffic jams and periods of free flowing traffic. However, the algorithm was not able to isolate the traffic jam in the interval shown in figure 8 (a). This is because this particular period is characterized by a slowly decreasing average speed and thus the SSE for each interval does not exceed λ.

3.4 Spatial-temporal Neighborhood Discovery

SST Data: We have employed the spatiotemporal neighborhood algorithm on a ten day time series of SST measurements for 44 sensors in the equatorial Pacific Ocean, totalling 63360 observations. The objective of the analysis is to

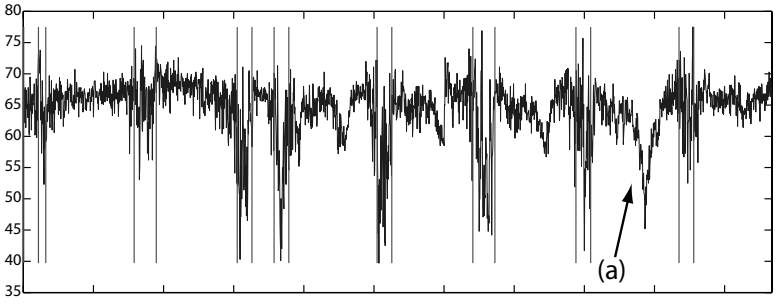


Fig. 8. Agglomerative temporal intervals for traffic data

determine if the algorithm can allow for the discovery of spatiotemporal patterns of sea surface temperature. In this section the preliminary results of this analysis are presented. We first discuss the temporal intervals, spatial neighborhoods and then the Spatiotemporal neighborhoods for some relevant intervals. The temporal intervals discovered by our approach are shown in Figure 9.

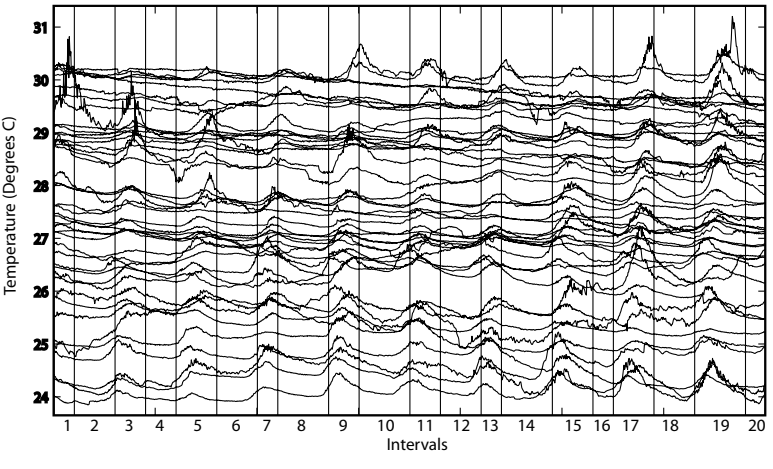


Fig. 9. Temporal Intervals for Time Series at all SST Measurement Locations

Ground Truth Validation: The algorithm divided the time series into 20 temporal intervals. In Figure 9 the intervals are plotted as vertical lines on top of the SST time series for all 44 sensors. The intervals show the ability to capture the diurnal pattern of the SST data by generally following the daily warming and cooling pattern that is evident in each time series. However, it can be noticed from the result that there are some sensors where there exists a lag in the diurnal pattern. This is likely a result of the locations being distributed across the Pacific

Ocean and time is reported in GMT and thus there exists a delay in the warming of the water based on the rotation of the earth from east to west. From a data mining standpoint, where the peak SST occurs during the interval could then be a predictor of the longitude of the sensor location.

The next part of the algorithm created spatial neighborhoods for each interval. Figure 10 shows the neighborhood quality for the four resulting neighborhoods at each temporal interval.

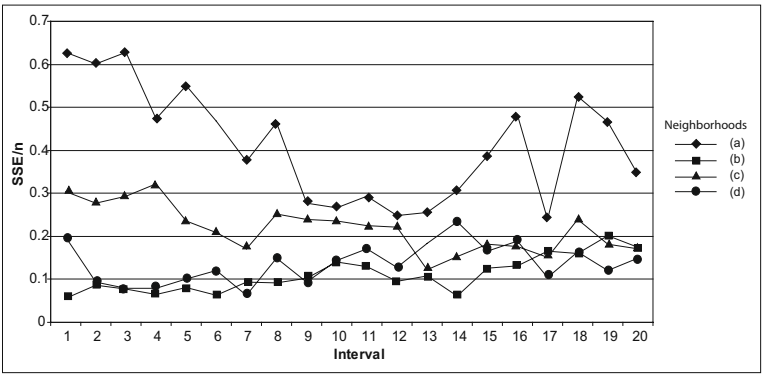


Fig. 10. Neighborhood Quality for each Interval

The neighborhood quality changes quite a bit for each interval with neighborhood (a) having the highest within-neighborhood error and neighborhood (b), (c), and (d) generally having a low within-neighborhood error. This indicates that there may be more than one natural grouping in neighborhood 1 during a number of intervals. However from intervals 9 to 13 the error in neighborhood (a) was comparable with neighborhoods (b), (c), and (d). This identifies a challenge in that there may not always be the same number of neighborhoods in a dataset and furthermore, the number of neighborhoods may not always be known a priori. One interesting pattern in the graph occurs between intervals 16 and 19 where the within-neighborhood error of neighborhood 1 goes from very high to low and back to very high. We will use these four intervals to demonstrate the results of the spatiotemporal neighborhoods. Figure 11 shows the neighborhoods formed for these intervals accompanied by a SST satellite image for the approximate time of the interval.

The formation of the spatiotemporal neighborhoods are validated by the pattern of sea surface temperature shown by the satellite image. Figure 11(a),(b),(c), and (d) show the neighborhood formation for each time step. Neighborhood (a) represents the cooler temperature water coming from the south east part of the image. Neighborhood (b) represents the area dominated by the very warm water in the south west part of the image, neighborhood (c) represents the moderate temperature water that is wrapped around neighborhood (a), and neighborhood (d)

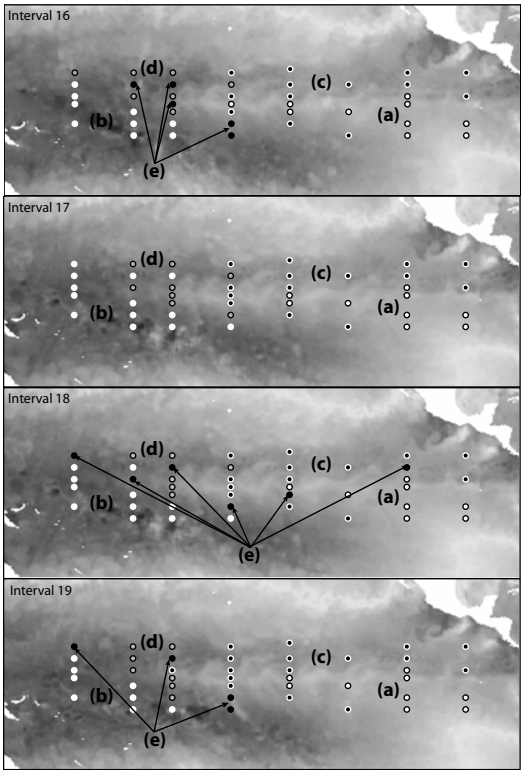


Fig. 11. Spatiotemporal Neighborhoods for Intervals 16 - 19 with AVHRR Satellite SST Image

represents the warmer temperatures that lie between neighborhoods (c) and (d). There are a number of locations where the neighborhoods overlap. Figure 11(e) points out the areas of overlap for each temporal interval. The overlapping areas typically take place along neighborhood boundaries where steep gradients of SST exist. The result also shows areas where change in SST occurs most. The most change occurs in the western four columns of sensors. This trend is validated by the satellite imagery in that it shows that this area is the boundary zone between warm water in the western Pacific and cooler water that travels along the equator.

Traffic Data: We have also demonstrated the spatiotemporal neighborhood algorithm on traffic data from the Interstate 270 cooridor, a heavily congested highway connecting Frederick, Maryland with the Washington DC Beltway. A one month period excluding weekends and holidays was taken from from 12 traffic sensors on south-bound Interstate 270. Measurements were taken every five minutes and this resulted in approximately 5000 values per sensor. The temporal intervals found in this data are shown in figure 12.

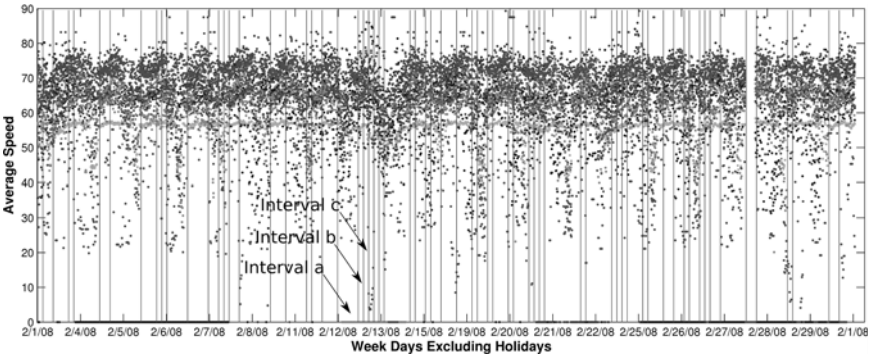


Fig. 12. Temporal Intervals for Traffic Data

Ground Truth Validation: The algorithm divided the time series into 67 distinct intervals. Each interval represents a change in the spatial pattern of traffic. For example during free flow traffic periods, the spatial pattern is typically represented by one spatial neighborhood for the entire section of highway. As traffic builds during peak periods, new spatial neighborhoods are formed where bottle necks occur. For example, intervals a, b, and c in figure 12 are characteristic of a period that goes from free traffic flow to a congested traffic flow in the morning of February 12, 2008. According to the Maryland Weather Blog (http://weblogs.marylandweather.com) freezing precipitation fell during this period. The spatial neighborhoods for intervals a, b, and c are shown in figure 13.

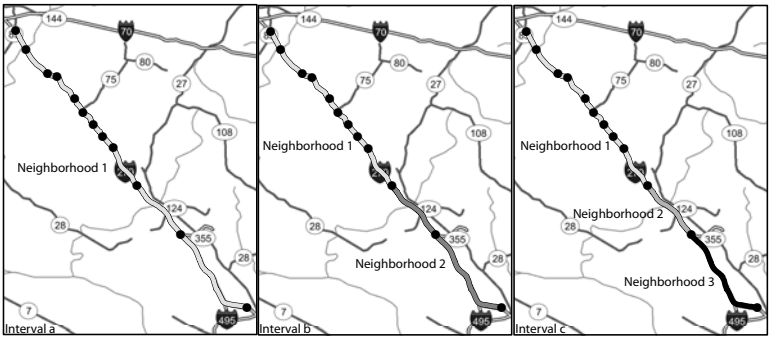


Fig. 13. Spatiotemporal Neighborhoods for Traffic Data: Intervals a, b, and c

Interval a is shown as one neighborhood of free flow traffic. Then in interval b, neighborhood 2 represents traffic slowing down as it approaches the Washington D.C. beltway (Interstate 495). Finally in interval c the traffic slows down more drastically to create neighborhood 3. The appearance of these distinct spatial

neighborhoods validates our approach in terms of the ability to find distinct temporal intervals where there is a change in the spatial neighborhood.

4 Related Work

Spatial neighborhood formation is a key aspect to any spatial data mining technique ([6, 11, 12, 16, 21, 22]etc.), especially outlier detection. The issue of graph based spatial outlier detection using a single attribute has been addressed in [21]. Their definition of a neighborhood is similar to the definition of neighborhood graph as in [6], which is primarily based on spatial relationships. However the process of selecting the spatial predicates and identifying the spatial relationship could be an intricate process in itself. Another approach generates neighborhoods using a combination of distance and semantic relationships [1]. In general these neighborhoods have crisp boundaries and do not take the measurements from the spatial objects into account for the generation of the neighborhoods.

The concept of a temporal neighborhood is most closely related to the literature focused on time series segmentation. The purpose of which is to divide a temporal sequence into meaningful intervals. Numerous algorithms [3, 13, 10, 18, 15] have been written to segment time series. One of the most common solutions to this problem applies a piecewise linear approximation using dynamic programming [3]. Three common algorithms for time series segmentation are the bottom-up, top-down, and sliding window algorithms [13]. Another approach, Global Iterative Replacement (GIR), uses a greedy algorithm to gradually move break points to more optimal positions [10]. This approach starts with a k-segmentation that is either equally spaced or random. Then the algorithm randomly selects and removes one boundary point and searches for the best place to replace it. This is repeated until the error does not increase. Nemeth et al. (2003) [18] offer a method to segment time series based on fuzzy clustering. In this approach, PCA models are used to test the homogeneity of the resulting segments. Most recently Lemire [15] developed a method to segment time series using polynomial degrees with regressor-based costs. These approaches primarily focus on approximating a time series and do not result in a set of discrete temporal intervals. Furthermore, because the temporal intervals will be generated at many spatial locations, a more simplified approach is required.

There has been some work to discover spatiotemporal patterns in sensor data [21, 14, 9, 8, 5]. In [21] a simple definition of a spatiotemporal neighborhood is introduced as two or more nodes in a graph that are connected during a certain point in time.There have been a number of approaches that use graphs to represent spatiotemporal features for the purposes of data mining. Time-Expanded Graphs were developed for the purpose of road traffic control to model traffic flows and solve flow problems on a network over time [14]. Building on this approach, George and Shekhar devised the time-aggregated graph [9]. In this approach a time-aggregated graph is a graph where at each node, a time series exists that represents the presence of the node at any period in time. Spatio-Temporal Sensor Graphs (STSG) [8] extend the concept of time-aggregated graphs to model spatiotemporal patterns in sensor networks. The

STSG approach includes not only a time series for the representation of nodes but also for the representation of edges in the graph. This allows for the network which connects nodes to also be dynamic. Chan et al. [5] also use a graph representation to mine spatiotemporal patterns. In this approach, clustering for Spatial-Temporal Analysis of Graphs (cSTAG) is used to mine spatiotemporal patterns in emerging graphs.

Our method is the first approach to generate spatiotemporal neighborhoods in sensor data by combining temporal intervals with spatial neighborhoods. Also, there has yet to be an approach to spatial neighborhoods that is based on the ability to track relationships between spatial locations over time.

5 Conclusion and Future Work

In this paper we have proposed a novel method to identify spatiotemporal neighborhoods using spatial neighborhood and temporal discretization methods as building blocks. We have done several experiments in SST and Traffic data with promising results validated by real life phenomenon.

In the current work we have focused on the quality of the neighborhood which has led to a tradeoff in efficiency. In our future work we would like to extend this work to find high quality neighborhoods in an efficient manner. We will also perform extensive validation of our approach using spatial statistics as a measure of spatial autocorrelation and study the theoretical properties in the neighborhoods we identify. We also intend to use knowledge discovery tasks such as outlier detection to validate the efficacy of our neighborhoods. We will also explore the identification of *critical temporal intervals* where most dramatic changes occur in the spatial neighborhoods.

Acknowledgements

This work has been funded in part by the National Oceanic and Atmospheric Administration (Grants NA06OAR4310243 and NA07OAR4170518). The statatements, findings, conclustions, and recommendations are those of the authors and do not necessarily reflect the views of the National Oceanic and Atmospheric Administration or the Department of Commerce.

References

1. Adam, N.R., Janeja, V.P., Atluri, V.: Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets. In: Proc. ACM SAC, New York, pp. 576–583 (2004)
2. Administration, F.H.: Traffic bottlenecks: A primer focus on low-cost operational improvements. Technical report, United States Department of Transportation (2007)
3. Bellman, R., Roth, R.: Curve fitting by segmented straight lines. Journal of the American Statistical Association 64(327), 1079–1084 (1969)

4. Cane, M.: Oceanographic events during el nino. Science 222(4629), 1189–1195 (1983)
5. Chan, J., Bailey, J., Leckie, C.: Discovering and summarising regions of correlated spatio-temporal change in evolving graphs. In: Proc. 6th IEEE ICDM, pp. 361–365 (2006)
6. Ester, M., Kriegel, H., Sander, J.: Spatial data mining: A database approach. In: Scholl, M.O., Voisard, A. (eds.) SSD 1997. LNCS, vol. 1262, pp. 47–66. Springer, Heidelberg (1997)
7. C. for Advanced Transportation Technology Laboratory. Traffic data extraction software (web based)
8. George, B., Kang, J., Shekhar, S.: Spatio-temporal sensor graphs (stsg): A sensor model for the discovery of spatio-temporal patterns. In: ACM Sensor-KDD (August 2007)
9. George, B., Shekhar, S.: Time-aggregated graphs for modeling spatio-temporal networks. In: Roddick, J., Benjamins, V.R., Si-said Cherfi, S., Chiang, R., Claramunt, C., Elmasri, R.A., Grandi, F., Han, H., Hepp, M., Lytras, M.D., Mišić, V.B., Poels, G., Song, I.-Y., Trujillo, J., Vangenot, C. (eds.) ER Workshops 2006. LNCS, vol. 4231, pp. 85–99. Springer, Heidelberg (2006)
10. Himberg, J., Korpiaho, K., Mannila, H., Tikanmaki, J., Toivonen, H.: Time series segmentation for context recognition in mobile devices. In: ICDM, pp. 203–210 (2001)
11. Huang, Y., Pei, J., Xiong, H.: Co-location mining with rare spatial features. Journal of GeoInformatica 10(3) (2006)
12. Huang, Y., Shekhar, S., Xiong, H.: Discovering colocation patterns from spatial data sets: A general approach. IEEE TKDE 16(12), 1472–1485 (2004)
13. Keogh, E., Smyth, P.: A probabilistic approach to fast pattern matching in time series databases. In: Proc. 3rd ACM KDD, pp. 24–30 (1997)
14. Kohler, E., Langkau, K., Skutella, M.: Time-expanded graphs for flow-dependent transit times. In: Möhring, R.H., Raman, R. (eds.) ESA 2002. LNCS, vol. 2461, pp. 599–611. Springer, Heidelberg (2002)
15. Lemire, D.: A better alternative to piecewise linear time series segmentation. In: SIAM Data Mining 2007 (2007)
16. Lu, C., Chen, D., Kou, Y.: Detecting spatial outliers with multiple attributes. In: 15th IEEE International Conference on Tools with Artificial Intelligence, p. 122 (2003)
17. McPhaden, M.: Genesis and evolution of the 1997-98 el nino. Science 283, 950–954 (1999)
18. Nemeth, S., Abonyi, J., Feil, B., Arva, P.: Fuzzy clustering based segmentation of time-series (2003)
19. NOAA. Tropical atmosphere ocean project, `http://www.pmel.noaa.gov/tao/jsdisplay/`
20. Rasmusson, E., Wallace, J.: Meteorological aspects of the el nino/southern oscillation. Science 222(4629), 1195–1202 (1983)
21. Shekhar, S., Lu, C., Zhang, P.: Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In: 7th ACM SIG-KDD, pp. 371–376 (2001)
22. Sun, P., Chawla, S.: On local spatial outliers. In: 4th IEEE ICDM, pp. 209–216 (2004)