Spatiotemporal Periodical Pattern Mining in Traffic Data

Tanvi Jindal, Prasanna Giridhar, Lu-An Tang, Jun Li, Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign
{tjindal2, giridha2, tang18, junli87, hanj}@illinois.edu

ABSTRACT

The widespread use of road sensors has generated huge amount of traffic data, which can be mined and put to various different uses. Finding frequent trajectories from the road network of a big city helps in summarizing the way the traffic behaves in the city. It can be very useful in city planning and traffic routing mechanisms, and may be used to suggest the best routes given the region, road, time of day, day of week, season, weather, and events etc. Other than the frequent patterns, even the events that are not so frequent, such as those observed when there is heavy snowfall, other extreme weather conditions, long traffic jams, accidents, etc. might actually follow a periodic occurrence, and hence might be useful to mine. This problem of mining the frequent patterns from road traffic data has been addressed in previous works using the context knowledge of the road network of the city. In this paper, we have developed a method to mine spatiotemporal periodic patterns in the traffic data and use these periodic behaviors to summarize the huge road network. The first step is to find periodic patterns from the speed data of individual road sensor stations, and use their periods to represent the station's periodic behavior using probability distribution matrices. Then, we use density-based clustering to cluster the sensors on the road network based on the similarities between their periodic behavior as well as their geographical distance, thus combining similar nodes to form a road network with larger but fewer nodes.

Categories and Subject Descriptors

H.2.8 [**Database Applications**] – Data Mining, Spatial Databases and GIS

General Terms

Algorithms, Experimentation

Keywords

Periodic Patterns, Spatiotemporal data, Traffic Data, KL-Divergence, Density-based clustering, Road network, Probability Distribution Matrices

1. INTRODUCTION

With the developments in technology, there are different ways in which moving objects are being monitored, generating huge amounts of mobility data. We have movement data for individuals, with the help of GPS systems; for animals and birds, with animal scientists studying their movement patterns; and other moving objects. Traffic sensors have been installed at a large number of monitoring stations on different highways, expressways, freeways and road intersections to monitor the flow of traffic through them. Such stations usually monitor the speed of the vehicles crossing through them, and keep a track of the average speeds and flow of the traffic they monitor over small periods of time, generating huge amounts of data that needs to be analyzed. The main aim of this paper is to find periodic behaviors from the speed data of traffic in the California region, and use them to cluster the stations in the road network that have similar behaviors to be able to summarize the huge road network.

There are two major different types of mobility data – individual and aggregate. In the individual or object-based mobility data, the identity of each object can be traced and the movement trajectories of each of the objects are analyzed separately to find patterns. In the aggregate or sensor-based mobility data, the identity of each object cannot be traced and the analysis is done on the collective behavior of the moving objects. The data from traffic sensors falls into the second category, where we have only averages over a large number of individual moving objects.

There are a number of different applications of mining road traffic data. It can help us in better management of traffic in a city and can help in identifying where new roads need to be developed. Another very important application is to be able to summarize the huge, and complex traffic data into actionable knowledge, which can even be used to discover best routes between any two points on the map. In addition to the basic route discovery, we can even use this knowledge to develop comprehensive methods that take into account the frequently traveled roads, the weather conditions, the traffic conditions, and so on. There are many challenges in being able to extract such information from the raw data. Firstly, the data is very huge, with speed values being collected every few minutes, and complex, owing to the large number of factors that can affect the behavior of traffic on the roads, introducing large variable noises in the data. Secondly, apart from the data, the patterns in the real world data are also complex as there are many patterns that are interacting with each other, making it very difficult to identify and model them.

One of the most important and frequently occurring patterns in moving object data is periodicity. Finding periodic behaviors is very important in understanding the object movements and in summarizing those movements. These periodic behaviors can very effectively describe the entire motion of the object, and hence, it is possible to store only these behaviors rather than storing the entire data. The main challenges in identifying periods is that with traffic data, the patterns may not repeat at the exact same times every time, or the speed that is repeating may not be the exact same. Also, there may be multiple different periods in the same series and the overlay and interaction of them makes them difficult to identify. For example, let us consider a station that is located near the San Jose exit on highway 101. Every morning, a large number of vehicles will take that exit to go to work to their offices located in San Jose. Now, there will be periodic patterns in the speeds of vehicles, but the actual speed at

UrbComp’13, August 11–14, 2013, Chicago, Illinois, USA.

a given time may vary on different cycles of the period, as the traffic at a given time on the exit might vary given the fact that the drivers may be running a little late or early. Also, there will be daily patterns and weekly patterns in the speed values, as people do not go to work on weekends, but there may be other vehicles entering San Jose on weekends for other purposes. But both these patterns are intertwined in the speed data, making it more difficult to mine them.

There have been many works ([4], [5], [13], [15]) that have tried to identify the periodic patterns from moving object data. Fourier transform and autocorrelation have been used for period detection in signal processing field for period detection but it is not feasible to use them directly for the traffic data because of the challenges mentioned above. We have modified the period detection approach from the KDD'10 paper to fit the needs of our dataset, to find the multiple periods in the traffic data.

Apart from periodicity finding methods, there have been efforts to develop much more comprehensive systems to summarize traffic data and use them for various applications. One such system proposed by Xiaolei Li et al [1], is FlowScan, which uses traffic-density based approach to find hot routes in traffic data. They use a hybrid between object-based and sensor-based analyses of mobility data and try to cluster road segments based on the density of traffic they share. Another work by Hector Gonzales et al [2], adaptively computes the fastest path between two points, based on the frequent traffic patterns mined from the traffic data. They use the hierarchy of roads to partition the road network into areas, and different path pre-computation strategies can be used at the area level. It also incorporates knowledge from driving patterns and speed patterns to take care of the weather conditions and the knowledge of the local people.

Most of these works require an understanding of the underlying road network, to be able to identify the traffic patterns from the huge collection of data. In this paper, we try to come up with a method to summarize the road traffic data with a minimal amount of knowledge about the road network.

The main contributions of this paper are: (1) identifying the periodic behavior in the speed data of an individual station, which can tell us the distribution of speeds at different times of days, and on different days of week; (2) developing a method to find similarities between the periodic behaviors of different stations, and making clusters of stations on a particular route based on this similarity and the geographical distance between the stations.

The rest of the paper is organized as follows: Section 2 gives an overview of the problem that the paper is trying to solve, the data that is used and a high-level description of how the algorithm works. Section 3 discusses in detail the technique used to detect the periods and periodic behaviors from the traffic data. Section 4 discusses the clustering of stations to form nodes using the periodic behaviors discovered in the previous step. The subsequent sections then describe the experimental results, and conclusions.

2. OVERVIEW

To be able to understand the methods developed in this paper, it is first important to understand the data that has been used for conducting the various experiments. We use the well-established Performance Management System (PeMS) [8] to obtain the speed and flow data for the traffic on a large number of roads in the state of California. PeMS collects and stores data from California loop detectors, which record the occupancy and flow of vehicles on a freeway section. Each detector makes a recording every thirty seconds, where the data values are the average speeds across all the vehicles that cross the detector in that time span. The entire data is huge, given that it collects this data every 30 seconds over years for a large number of stations. To make it much more feasible to use the data effectively, PeMS makes available these data entries averaged over five minute intervals. For the purposes of this paper, we reduce the scope to considering only the stations that are on one single highway over the time span of only one month at a time. Even then, the data remains huge and has too fine granularity, and thus, we further average these readings over periods of one hour.

Fig. 1 shows a plot of the original speed data for a station for the time period of one month, that is, 720 hours, where x-axis shows the time in hours. The increments on x-axis are multiples of 24 to show the progression of the data with days.

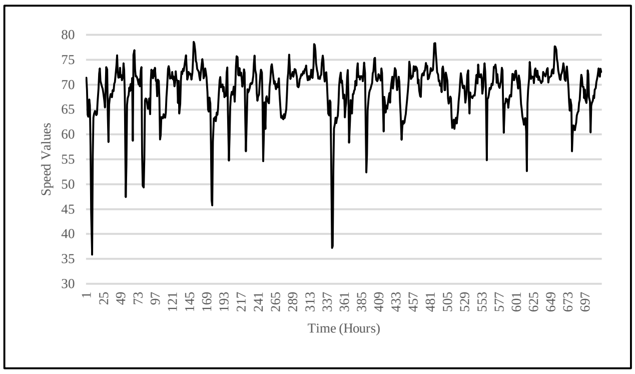
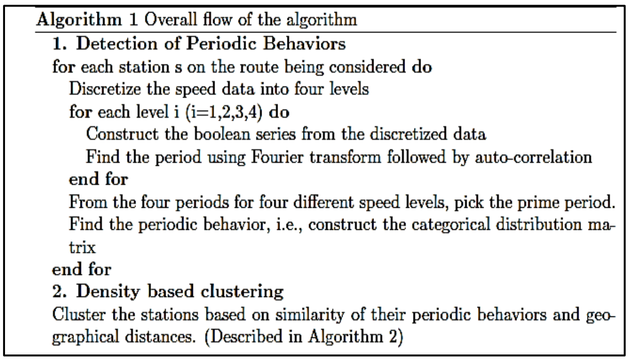


Figure 1. Original data for one station

Our algorithm takes this data as its input and analyzes it to find spatiotemporal periodic patterns from the data and summarize the road network. Figure 2 gives an overall flow of the algorithm and gives an idea about how the algorithm progresses. The whole algorithm works in two major stages:

1. *Detection of periodic behaviors* – In this stage, the algorithm analyses the speed data for an individual station to find the periodic behavior of the station. This in itself involves first finding the periods from the data and then constructing the categorical distribution matrices for the station.
2. *Clustering the stations* – Once we have constructed the categorical distribution matrices for all the stations, we use the similarity between these matrices, i.e., the periodic behavior of the stations, and the physical distance between the stations to form clusters.



```
Algorithm 1 Overall flow of the algorithm
1. Detection of Periodic Behaviors
for each station s on the route being considered do
  Discretize the speed data into four levels
  for each level i (i=1,2,3,4) do
    Construct the boolean series from the discretized data
    Find the period using Fourier transform followed by auto-correlation
  end for
  From the four periods for four different speed levels, pick the prime period.
  Find the periodic behavior, i.e., construct the categorical distribution matrix
end for
2. Density based clustering
Cluster the stations based on similarity of their periodic behaviors and geographical distances. (Described in Algorithm 2)
```

Figure 2. The flow of the whole algorithm

The next two sections discuss both these steps in much more detail along with appropriate examples.

3. PERIODICITY DETECTION

3.1 Finding periods

The first step in our approach is to find periods in the traffic data from a single station so as to be able to represent the station using its periodicity. The speed data obtained from PeMS has continuous values of speed, and has multiple periods interleaved and is very noisy. Hence, typical period detection methods like Fourier transform cannot be directly used on this speed data. To be able to find the periods, we first discretize the speed data into four discrete levels. To find the speed limits that will divide the entire possible speed range into four different discrete levels, we perform k-means clustering on the set of all possible speeds for all the stations under consideration. Setting the value of k to 4, the following speed ranges were obtained, <38, 38-57, 57-67, and >67, and we used 0, 1, 2, and 3 to represent these speed ranges in the discretized version of the data. Fig. 3 shows the same data after it has been discretized into 4 levels; 1, 2, 3, and 4, using the speed ranges.

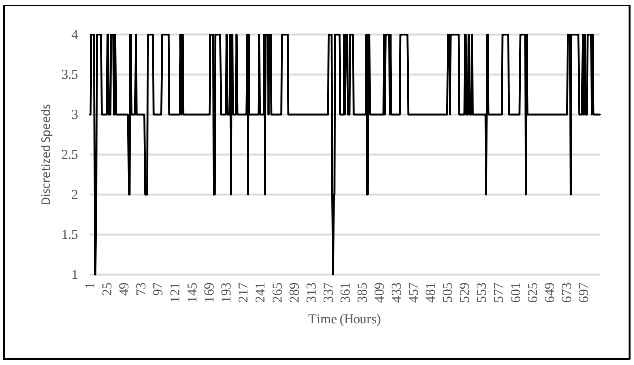


Figure 3 Discretized Data for the station

It can be seen clearly from these figures that there is some periodicity in the data, but it is not feasible to apply periodicity detection methods directly even on the discretized data, because of the presence of multiple periods. Thus to effectively detect the periods, we try to find the periodicities of the different speed levels individually. To do so, we convert the discretized speed time series into Boolean time series from the reference point of each of the speed levels. This simply entails the following step:

If the series is of the form 112344232133212…

Then we convert it to the following Boolean series by taking all the 3's as 1's and the rest of the labels as 0's 000100010011000…

This is when we are trying to find the periods in the occurrences of speed level 3.

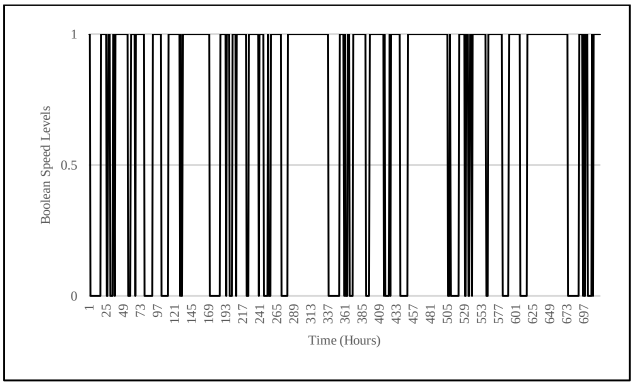


Figure 4 Station Data in Boolean form

Fig. 4 shows how the time series looks after it has been converted to a Boolean time series. Once we have this Boolean series, we find periods in it using Fourier Transform followed by circular autocorrelation. Vlachos et al [13] discuss how the two most common periodicity detection methods in the signal processing area, Fourier transform and auto-correlation, complement each other. On one hand, Fourier transform often suffers from the low resolution problem in the low frequency region, hence provides poor estimation of large periods. Also, it tends to generate a lot of false positives in the periodogram due to its spectral leakage problem. On the other hand, autocorrelation offers accurate estimation for both short and large periods, but is more difficult to set the significance threshold for important periods. To overcome these shortcomings of both the methods, they proposed to combine them to find periods from time series. In our paper, we use a similar approach that is a modification of the periodicity detection technique used by Zhenhui et al in [4].

The first step is to take the Discrete Fourier Transform (DFT) of the Boolean sequence B = $b_1, b_2, . . ., b_n$ (b_i is the Boolean speed value for the present analysis at the i^{th} time interval), to transform it into the sequence of n complex numbers $X_1, X_2, . . . ,X_n$. A periodogram is then constructed for the series, which is a plot of the power spectral density of each of these complex numbers. The periodogram helps in identifying the possible periods in the frequency domain, which is done by setting a threshold and taking all the frequencies that have power densities above a threshold as period hints.

The threshold is determined using the idea that any random permutation B′ of B should not exhibit any periodicities, and hence even the maximum power in B' will not indicate the period in the sequence. Therefore, the threshold is set as the maximum power for B', and to get a 99% confidence level on what frequencies are important, we repeat the above random permutation experiment 100 times and record the maximum power of each permutated sequence. The 99-th largest value of these maximum powers is then taken as the threshold.

The frequencies thus identified might still not indicate the true period of the data, as a single value k in frequency domain corresponds to a range of periods [n/k, n/k−1) in time domain. The next step is thus to use circular auto-correlation to identify the true periods from among the period hints provided by the Fourier Transform. The intuition is that only if the candidate period from the periodogram lies on a hill of the auto-correlation function, then we can consider it as a valid period, otherwise it is a false alarm.

Thus, for each period range [l, r) given by the periodogram, we test whether there is a peak in {R(l), R(l+1), . . . , R(r−1)} by fitting the data with a quadratic function. We return the point in this range which has the maximum value in case the function is concave, as that indicates the presence of a peak.

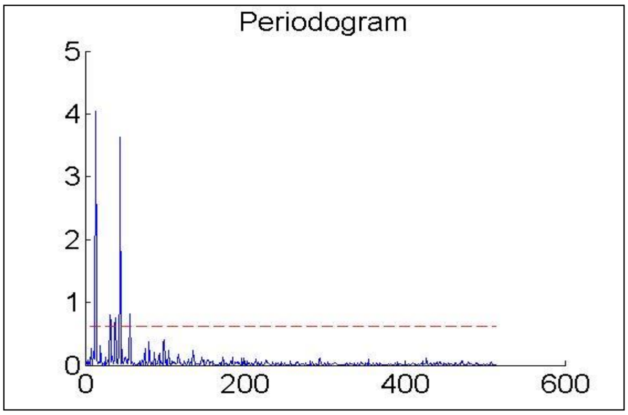


Figure 5. Periodogram of the Boolean series

To illustrate this technique of finding the periods of the datamore clearly, we will work on the Boolean series obtained from the speed data of one of the stations for the month of June

2009. Figures 1, 3 and 4 show how the plots of the original, discretized, and the Boolean data look for any station for one month. For one of these stations, Fig. 5 shows the periodogram obtained after taking the Fourier transform of the Boolean series. The x-axis shows the frequencies and the y-axis is a measure of the power spectral density. For a given frequency, higher power spectral density indicates a stronger sense of periodicity with that frequency in the data. The red dashed line shows the threshold identified using the method described above. We can see that there are many frequencies which pass the threshold test, but not all of them are true periods of the data. Fig. 6 then shows the auto-correlation function for all the candidate periods, where and we can easily see that only one of those lies on a peak, and hence, is the true period of the Boolean series.

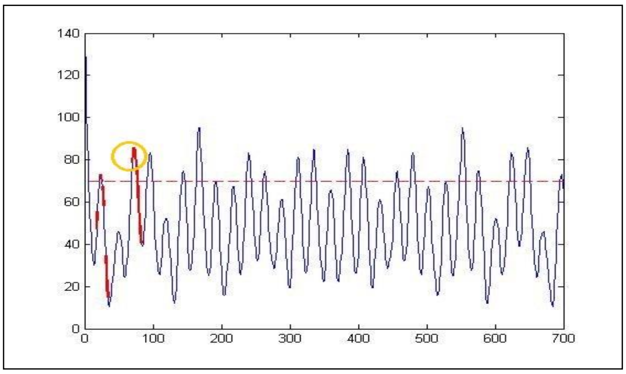


Figure 6 Autocorrelation function fitted to a quadratic

In this way, we do period analysis for each station individually and we limited our work to one route at a time, and considered only the stations on that highway. For each station, we generate four different Boolean series from the perspectives of the four speed levels, and then find out the periods for all these four series, giving us up to four different periods for each station.

For each station, the set of observed periods contained periods corresponding to approximately one day, i.e., 24 hours, or multiples of 24 hours. Thus, we identified the principal period for each station and associated each station to a single period. This also accommodates noises, treating periods of 23 or 25 hours as roughly daily periods, and if a station has both 24 hours and 24*k as its periods, then 24 hours is identified as its principal period**.**

3.2 Periodic Behavior of Stations

The main aim behind finding the periods for the stations was to determine a better way to represent the station data. Now, once we have the periods of the stations, we can use the periods to summarize the speed distribution at a station. This is done by constructing a categorical distribution matrix for each station.

DEFINITION 1 (Categorical Distribution Matrix)

Let T = {t_1, t_2, . . . , t_T } be a set of relative timestamps, and x_k be the categorical random variable indicating the speed level at the station at timestamp t_k. A categorical distribution matrix M for the station is a size s*T matrix where T is the period and s is the number of possible speed levels. Here, M(i,j) is the probability that the speed level at timestamp j is i.

In other words, M = [p_1, . . . , p_T], where each column p_k = [p(x_k =0), p(x_k = 1), . . . , p(x_k = s)]^T is an independent categorical distribution vector with $\sum_{i=0}^{s} p(x_k = s) = 1$.

Now, suppose that the time series for a station is generated by some distribution matrix P. To estimate P, we use the maximum likelihood estimation method. Let the whole series be denoted by S. We divide the series into T-sized intervals to get T= {T_1, T_2, …,T_n} , where n = floor(length(S)/T).

Then, the probability that S is generated using P is given by:

$$P(S|P) = \prod_{i=0}^{n}\prod_{j=0}^{T} P(x_k = S_i^j)$$

Where S_i^j is the speed level at jth timestamp in ith segment.

According to maximum likelihood estimation (MLE), to find the best generative model, the following log likelihood maximization problem needs to be solved.

$$max_p\{L(P|S) = \log P(S|P)\}$$

The well-known solution to this problem is given by:

$$p(x_k = m) = \sum_{i=0}^{n} (In(S_i^k == m))/n$$

where, In(b) is an indicator function which returns 1 if b evaluates to true, and 0 otherwise. In words, we say that p (x_k = m) is the ratio of the number of times the speed level is m at relative timestamp t_k with the total number of intervals, n.

To avoid the probabilities from being zero, in case, a particular relative timestamp never sees a particular speed level, we use a background prior probability u, and add it to p(x_k=i).

$$p(x_k = i) = (1 - \lambda)p(x_k = i) + \lambda\mu$$

where, λ is a small smoothing parameter 0 < λ <1.

In this way, the categorical distribution matrix is constructed for each station, representing the periodic behavior of the station.

4. CLUSTERING OF STATIONS

The main motivation for this paper is the need to be able to summarize the entire road network and the way traffic behaves in a city. The type of road, that is, if it is an inter-state highway, state highway, or a local road greatly affects the speeds of the vehicles travelling on that road. Apart from the road type and size, there are many other factors that affect the behavior of traffic. One of the major factors is the time of day at which the vehicle is traversing the road. For example, in the early hours of morning, there are very few vehicles on the road, and hence the traffic moves smoothly, with high average speeds. Whereas, during peak rush hours, around 9-10 am in the morning or 4-6 pm in the evenings, the average speeds are much lower and the possibility of occurrences of traffic jams becomes much higher. In addition for a given road at a given time, the speed at which vehicles move also depends on the position of the vehicles on the road. The speeds might be much lower on those segments of the road where the traffic from other roads merges, or other temporary factors, like construction work, or an accident.

To be able to effectively represent the traffic behavior on the road network, the roads have to be divided into smaller segments rather than being used as individual edges. In the scope of this paper, where we collect data from a large number of stations that are present along the highways, an obvious choice is to use each station as a node and the road segments between them as the edges in the network.

The problem with doing this is that this creates a very huge network, and also due to the proximity of the stations to each other, sometimes, it is not necessary to consider them as separate nodes. For example, let us consider two stations that are close to each other on the same road, then there is a possibility that the behavior of traffic at each of these stations is very similar, as there are no major changes that take place in the average speeds of vehicles in moving from one station to the other. Thus, it is more efficient and less redundant to combine these stations into one node, to reduce the size of the road network. The periodic behavior of the node then can be defined as the average of the periodic behaviors of all the stations that are a part of the node.

In this part of this paper, we develop a method to identify such stations and combine them into one node. An intuitive approach to do so is to cluster the stations that are close to each other based on a similarity measure for their periodic behavior representations. The main challenges in the clustering process are: (1) finding a similarity measure that can effectively identify the stations that should be put together in a node, and (2) finding an appropriate clustering algorithm for the stations.

4.1 Similarity Measures

For any clustering algorithm, it is very important to select the correct similarity measure. For our system, we have very limited information about each of the stations. The two major pieces of information we have are (1) the geographical location of the station and, (2) the categorical distribution matrix representing the periodic behavior of the speeds of vehicles crossing the station.

4.1.1 Physical Distance

For this work, since we are limited to analyzing only one highway at a time, we use the Manhattan distance between two stations along the road as the physical distance between them. PeMS system makes available details about all the stations as metadata. The metadata has the location of a station in the form of latitude and longitude coordinates, but it also has the mile position of the station on the highway it is located on. In this paper, we directly use the absolute value of the difference between the mile positions of two stations as the physical distance between them.

4.1.2 Similarity between the periodic behaviors

Since the periodic behaviors are represented using probability distribution matrices, a similarity measure between the two must be such that two stations similar to each other will have a high chance to be generated from the same periodic behavior. One of the most popular distance measures is Kullback-Leibler divergence (KL divergence), which is defined as follows:

$$KL(P,Q) = \sum_{k=1}^{T}\sum_{i=1}^{d} p(x_k = i).\log(\frac{p(x_k = i)}{q(x_k = i)})$$

Where T is the time period and d is the number of possible speed levels. KL divergence might become infinite if the probability values become 0, and hence, we already added a smoothened background variable u to $p(x_k$=i) to avoid this.

Although KL divergence is a good measure of similarity between two probability distribution matrices, it is not a metric as it is not symmetric with KL(P,Q) ≠ KL(Q,P). Dominik Endres et al suggested a metric which is derived from KL divergence, called the Jersen-Shannon divergence (JS divergence) defined as follows:

$$JS(P,Q) = \frac{1}{2}(KL(P,M) + KL(Q,M))$$

Where M =1/2 (P+Q) is the mid-point between the two matrices. JS divergence is a metric as it is both symmetric and bounded, and hence, it is a better similarity measure as compared to KL divergence.

Clearly,

$$JS(P,Q) = JS(Q,P)$$

Also, $0 \leq JS(P,Q) \leq 1$, if we take logarithms on base 2.

If we examine JS divergence from a statistical point of view, we can see that

$$JS(P,Q) = H(M) - \tfrac{1}{2}H(P) - \tfrac{1}{2}H(Q)$$

Let us use p_i to denote $p(x_k$=i), q_i to denote $q(x_k$=i) and m_i to denote $m(x_k$=i). Then,

$$JS(P,Q) = \frac{1}{2}(\sum_{k=1}^{T}\sum_{i=1}^{d} p_i.\log\left(\frac{p_i}{m_i}\right) + q_i.\log(\frac{q_i}{m_i}))$$

$$= \frac{1}{2}(\sum_{k=1}^{T}\sum_{i=1}^{d} p_i.\log p_i + q_i.\log q_i - (p_i + q_i)\log m_i)$$

$$= \frac{1}{2}(-H(P) - H(Q) + 2H(M))$$

Since the KL divergence, KL(P,Q) can be interpreted as the inefficiency of assuming that the true distribution is Q when it really is P, JS divergence, JS(P,Q) could be seen as a minimum inefficiency distance. Thus, we can see that this is an efficient and appropriate measure to find the distance between the periodic behaviors of two stations.

4.2 Clustering Algorithm

After the similarity measures have been finalized, an appropriate clustering algorithm needs to be developed which can effectively use both the distance measures to put similar stations into one node and non-similar stations into separate nodes. Another important thing is that the algorithm should not require the system to pre-specify a number of clusters in the clustering result. In this paper, we used an adaptation of density-based clustering to cluster the stations.

The main idea is that for each station, the algorithm first finds out the stations that are in physical proximity to the station. This set of stations is called the neighborhood of the station.

DEFINITION 2 (Neighborhood of a station)

The neighborhood of a station s N(s) is defined as: N(s) = {t| d(s,t) < d_0}, where d(s,t) is the physical distance between stations s and t and d_0 is a user-defined distance threshold that controls the size of the clusters formed.

From the neighborhood of a station, we then isolate those stations that have JS divergence less than a threshold, and put them together in a cluster. The threshold for JS divergence is set locally relative to the neighbor set. Consider a neighborhood set N(s) for a station s. To set the threshold, we take all possible pairs in the set and compute the JS divergence between them. The threshold is then set to (mean + 0.5 standard deviation) for this set of distances. Setting the threshold locally is more effective than setting a global threshold as it helps to discover clusters from neighborhoods depending on the average divergence between the neighborhood stations. If the average divergence is very high in one neighborhood, setting a small global threshold will never be able to cluster them. Similarly, if the average divergence in a neighborhood is very small, and the global threshold is set to a large value, the algorithm will always put them in the same cluster.

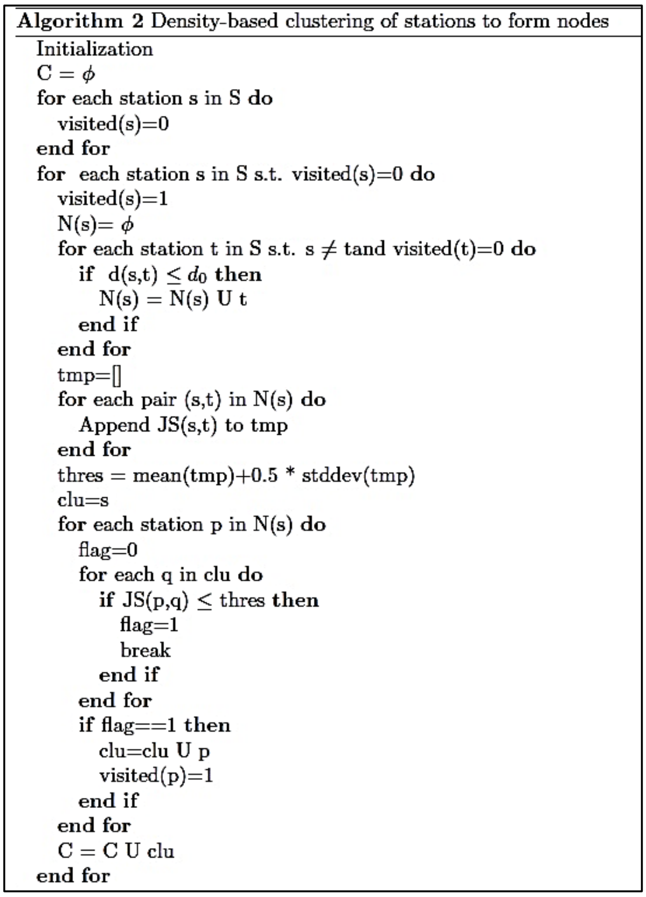
The algorithm iterates over all the stations and tries to form a cluster from the neighborhood of that station. It maintains a visited flag for each station and sets it when a station becomes a part of a cluster, to ensure that no station is part of two clusters. The algorithm is described in detail in Fig. 7.

5 EXPERIMENTAL RESULTS

For this study, we used stations on a single route only to form clusters to form the clusters in the target road network. We have included the results from the intermediate steps like periodicity detection along with the description of those steps. Fig. 8 shows the results from the final step when the algorithm is run on the highway I-405. It is clear from the image that the stations put in the same cluster are close to each other. The size of the clusters can be easily adjusted by adjusting the distance threshold parameter, d_0. In Fig. 8, d_0 was set to 2 miles, and in Fig. 9, d_0 was set to 1 mile. We can see clearly the difference in the clusters that are obtained.

6 RELATED WORK

A lot of different types of work have been done on different spatiotemporal data in the past. Finding the periods from time series data is a very important part of the work done in this paper. A number of periodic pattern mining techniques have been proposed in data mining literature. Han et al. [15, 16] propose the algorithms for mining frequent partial periodic patterns. There have been a number of works that are based on the definition of frequent periodic pattern mining with a strict minimum support threshold. They tend to output a large set of patterns, most of which are slightly different. Besides, frequent periodic patterns cannot capture the statistical information as the periodic behaviors. Indyk et al. [9] studies the problem of discovering the most representative trend that repeats itself every T timestamps. However, they can only discover one trend for a given period T and such trend covers the whole time span. Vlachos et al. [13] proposed a method to use a combination of Fourier transform and auto-correlation to find periods without having to face the problems of finding false periods along with the true periods.



```
Algorithm 2 Density-based clustering of stations to form nodes
  Initialization
  C = $\phi$
  for each station s in S do
    visited(s)=0
  end for
  for  each station s in S s.t. visited(s)=0 do
    visited(s)=1
    N(s)= $\phi$
    for each station t in S s.t. s $\neq$ tand visited(t)=0 do
      if  d(s,t) $\leq d_0$ then
        N(s) = N(s) U t
      end if
    end for
    tmp=[]
    for each pair (s,t) in N(s) do
      Append JS(s,t) to tmp
    end for
    thres = mean(tmp)+0.5 * stddev(tmp)
    clu=s
    for each station p in N(s) do
      flag=0
      for each q in clu do
        if JS(p,q) $\leq$ thres then
          flag=1
          break
        end if
      end for
      if flag==1 then
        clu=clu U p
        visited(p)=1
      end if
    end for
    C = C U clu
  end for
```

Figure 7. Density-based clustering of stations

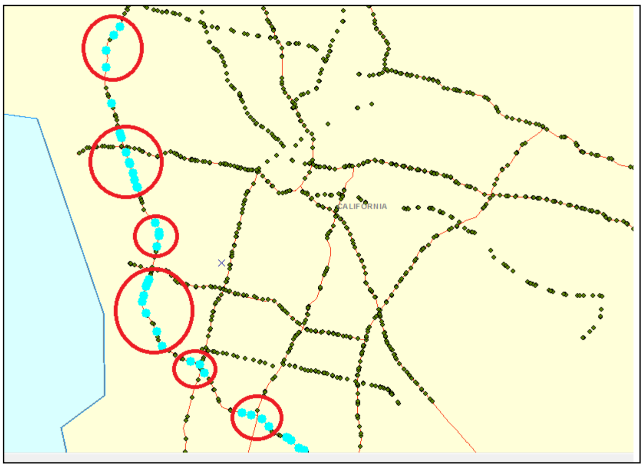


Figure 8. Clusters discovered on I-405

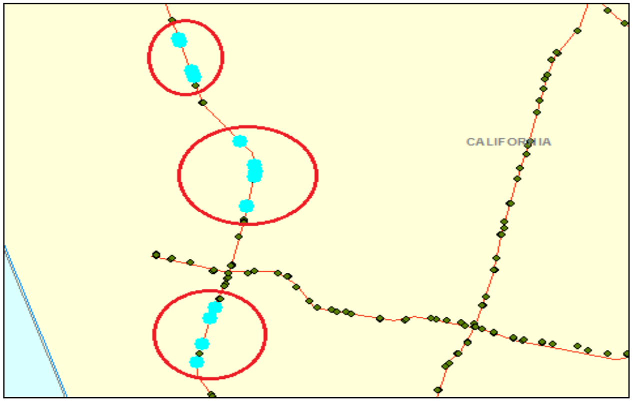


Figure 9. Clusters on I-405 with a smaller distance threshold

Zhenhui et al. [4] proposed an adaptation of this method so that it can be effectively used to find multiple periods from spatio-temporal data. The main intuition in this work is to separate the series into many Boolean series from different reference spots and then find periods. Another work by Zhenhui et al. [5] proposed a probabilistic method to find periods from incomplete observations.

Apart from work in finding periodicities, another set of works related to this paper includes work that try to summarize traffic data, finding frequent trajectories from moving object data, clustering trajectories, or route discovery from traffic data. Also, there is a large spectrum of works in this area, ranging from study of trajectories of individual objects to a collective study of the average behaviors of a large number of moving objects. Moving object clustering [18] discovers groups of objects that move together. Trajectory clustering [19] discovers groups of similar sub-trajectories from the whole trajectories of moving objects. Xiaolei Li et al. [1] proposes a method that is a hybrid of both these general techniques. The proposed method, FlowScan, that uses a density-based approach to discover hot routes from the trajectories of vehicular data. They try to cluster road segments based on the density of traffic that share, and hence even though they don’t consider individual moving objects, they use the identities of the objects to find how much of the traffic is really shared between two road segments. In another work, Hector Gonzales et al. [2] propose a method that tries to incorporate a large number of factors into the traffic analysis to find fastest paths between two points in an adaptive manner. The three main contributions of this work are to partition the road network into regions based on the hierarchy of roads, use of driving patterns and speed to incorporate the wisdom of the local people about the peculiar things that may affect the traffic. This work is closely related to the methods proposed in this work as the authors also strive to come up with an effective representation of the road network and consequently, to be able to discover edges between those nodes, based on speed patterns.

Another related area to our work is the work done in defining similarity measures between large time series. Felix Iglesias et al. [12] analyze the various different types of similarity measures that can be used while clustering time series. The unique thing about time series is that the shape of the input vectors entails features that are arranged in time, and hence, correlation becomes an important factor to consider. There are usually two types of methods used for clustering time series data: (a) feature-based or model-based, where raw data is pre- summarized or transformed by means of feature extraction or parametric models, e.g., dynamic regression, ARIMA, neural networks [20]; and, (b) raw-data-based, where clustering is directly applied over time series vectors without any space-transformation previous to the clustering phase. Several works concerning each kind of time series clustering are referred to in detail in [21]. The distance measures like simple Euclidian distance do not take into account the correlation between the different points in the time series. In this work, we first transform the time series into a probability distribution matrix and then try to find the distance between them. Kullback-Leibler divergence is the most popular similarity measures for probability distributions. Jersen-Shannon divergence [14] is based on KL-divergence and converts it into a metric, making it a more suitable similarity measure for clustering.

We would also like to briefly mention the works related to various different types of clustering methods. Xu Rui et al. [22] is a good compilation of all the clustering algorithms that are used to cluster different types of data. There are partitioning-based clustering algorithms like k-means and k-medoids that require the users to pre-specify the number of clusters that the objects should be divided into. Another set of clustering algorithms are those that do not require the user to pre-specify the number of clusters. Hierarchical clustering and density based clustering approaches are examples of clustering algorithms that aim to automatically discover the optimum number of clusters given the set of objects to be clustered.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we were able to develop an effective method to cluster stations on a road to effectively reduce the size of an otherwise huge road network. This is only the creation of nodes, and work still has to be done on defining the edges between these nodes. The main intuitions behind constructing the edges between these nodes would be (1) there can be many edges between two nodes, and (2) the edge weights will be based on the travel times between the nodes.

Let us consider two nodes A and B that are next to each other on a highway. If the edge between them were to be constructed based on the time that a vehicle takes to travel from A to B, it is evident that we cannot have a unique edge between A and B. This is because the behavior of traffic depends on a large number of factors, including and not limited to time of day, day of week, whether it’s a weekday or weekend and weather conditions. For example, during winters, if it snows, the time taken to go from one point to another can increase by manifolds as compared to the time taken during a warm summer day. Thus, construction of edges will entail defining many possible edges between any two nodes, and an edge would be a tuple of form $<n_1, n_2, c_1, c_2, \ldots, c_m>$, where n_1 is the starting node, n_2 is the end node, and $c_1, c_2, \ldots, c_m$ are the different factors that could possibly affect the vehicle speeds. One possible approach to discover edges can be to mine frequent patterns of traffic, and the defining of nodes beforehand will make mining such patterns much easier and more feasible.

Apart from the construction of edges, another important direction in which we are planning to work in the future is to find causal relationship patterns in the traffic propagation. To explain this, let us take the example of propagation of traffic jams. It is easy to observe that if there is a traffic jam on a road segment, it tends to propagate backwards, and the road segment just before the jammed segment also gets jammed in some time. The aim of discovering causal relationships between nodes is to be able to find models that can predict how such abnormal activity on a road segment propagates to other segments.

Once these basic extensions are made to the system, we can think of using the system to solve many real-life problems, such as discovery of the fastest routes between two given points. We could also think of using Data cubes to effectively store all these patterns and make it easier to manage them for real-time problem solving.

8 REFERENCES

[1] Xiaolei Li, Jiawei Han, Jae-Gil Lee, and Hector Gonzalez. "Traffic density-based discovery of hot routes in road networks." *Advances in Spatial and Temporal Databases (2007): 441-459*.

[2] Hector Gonzales, Jiawei Han, Xiaolei Li, Margaret Myslinska, and John Paul Sondag. "Adaptive fastest path computation on a road network: A traffic mining approach." *In Proceedings of the 33rd international conference on Very large data bases, pp. 794-805. VLDB Endowment, 2007.*

[3] Hector Gonzalez, Jiawei Han, Hong Cheng, Xiaolei Li, Diego Klabjan, and Tianyi Wu. "Modeling massive RFID data sets: a gateway-based movement graph approach." *Knowledge and Data Engineering, IEEE Transactions on 22, no. 1 (2010): 90-104.*

[4] Zhenhui Li, Bolin Ding, Jiawei Han, Roland Kays and P. Nye. Mining Periodic Behaviours for Moving Objects. *KDD ’10, 2010.*

[5] Zhenhui Li, Jingjing Wang, & Jiawei Han (2012, August). Mining event periodicity from incomplete observations. *In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 444-452). ACM.*

[6] Lu-An Tang, Yu Zheng, Jing Yuan, Jiawei Han, Alice Leung, Chih-Chieh Hung, and Wen-Chih Peng. "On discovery of traveling companions from streaming trajectories." *In Data Engineering (ICDE), 2012 IEEE 28th International Conference on, pp. 186-197. IEEE, 2012.*

[7] Robert L. Bertini, and A. Myton. "Using PeMS data to empirically diagnose freeway bottleneck locations in Orange County, California." Transportation Research Record: Journal of the Transportation Research Board 1925 (2005): 48-57.

[8] PeMS technology transfer overview http://www.techtransfer.berkeley.edu/newsletter/02-4/pems.php

[9] Gaffney, S., Smyth, P.: Trajectory clustering with mixtures of regression models. In *KDD’99 (1999)*

[10] J. Lee, Jiawei Han, K. Whang.: Trajectory clustering: A partition-and-group framework. In: *SIGMOD’07 (2007)*

[11] Lu-An Tang, Xiao Yu, Sangkyum Kim, Jiawei Han, Wen-Chih Peng, Yizhou Sun, Hector Gonzalez, and Sebastian Seith. "Multidimensional analysis of atypical events in cyber-physical data." *In Data Engineering (ICDE), 2012 IEEE 28th International Conference on, pp. 1025-1036. IEEE, 2012.*

[12] Félix Iglesias, and Wolfgang Kastner. "Analysis of Similarity Measures in Times Series Clustering for the Discovery of Building Energy Patterns." Energies 6, no. 2 (2013): 579-597

[13] M. Vlachos, P. S. Yu, and V. Castelli. On periodicity detection and structural periodic similarity. In *SDM, 2005.*

[14] Endres, Dominik M., and Johannes E. Schindelin. "A new metric for probability distributions." *Information Theory, IEEE Transactions on 49.7 (2003): 1858-1860.*

[15] J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. *In ICDE, 1999.*

[16] J. Han, W. Gong, and Y. Yin. Mining segment-wise periodic patterns in time-related databases. *In KDD, 1998.*

[17] P. Indyk, N. Koudas, and S. Muthukrishnan. Identifying representative trends in massive time series data sets using sketches. *In VLDB, 2000.*

[18] Gaffney, S., Smyth, P.: Trajectory clustering with mixtures of regression models. *In: KDD'99 (1999)*

[19] Lee, J., Han, J., Whang, K.: Trajectory clustering: A partition-and-group framework. *In: SIGMOD'07 (2007)*

[20] Hong, Y.Y.; Wu, C.P. Day-ahead electricity price forecasting using a hybrid principal component analysis network. *Energies 2012, 5, 4711–4725.*

[21] Liao, T.W. Clustering of time series data—*A survey. Pattern Recognit. 2005, 38, 1857–1874.*

[22] Xu, Rui, and Donald Wunsch. "Survey of clustering algorithms." *Neural Networks, IEEE Transactions on 16, no. 3 (2005): 645-678.*