Dynamic State Model for Analysis of Traffic States using Dirichlet Process Mixture Model

Kelathodi Kumaran Santhosh, Student Member, IEEE, Debi Prosad Dogra, Member, IEEE, Partha Pratim Roy and Bidyut Baran Chaudhuri, Life Fellow, IEEE

Abstract—Appropriate modeling of a surveillance scene is essential for detection of anomalies in road traffic. Learning usual paths can provide valuable insight of road traffic conditions and thus can help in identifying abnormal routes taken by commuters/vehicles. If usual traffic paths are learned in a nonparametric way, manual interventions for marking roads can be avoided. We propose an unsupervised and nonparametric method to learn frequently used paths from the tracks of moving objects in $\Theta(kn)$ time, where $k$ is the number of paths and $n$ represents the number of tracks. In the proposed method, temporal dependencies of the moving objects are taken into consideration to make the clustering meaningful using Temporally Incremental Gravity Model (TIGM)-Dynamic State Model (DSM). In addition, the distance-based scene learning makes it realistically intuitive to estimate the model parameters. Experimental validation reveals that the proposed method can learn a scene quickly without prior knowledge about the number of paths ($k$). We have compared the results with state-of-the-art methods. We also highlight the advantages of the proposed method over these techniques, especially for traffic monitoring applications. Further, we extend the model to represent notable traffic dynamics of a scene, that can be used for administrative decision making to control traffic at junctions or crowded places. We have also applied the proposed model to understand its effectiveness in clustering network traffic data.

Index Terms—Dirichlet process mixture model, Gibbs sampling, Bayesian inference, Incremental trajectory clustering, Intelligent transportation system, Nonparametric model, Unsupervised learning.

I. INTRODUCTION

In the event of increasing number of traffic on roads, it is difficult for the human operators to monitor and understand traffic behavior in real-time. Developing Intelligent transportation systems (ITS) can help the traffic administrations to learn the scene and detect anomalies efficiently.

In order to develop such systems, it is important to learn/model the traffic scenes from the data captured through sensors. With the widespread use of surveillance cameras, traffic video feeds are most useful sources to understand the scene. With the advances in machine learning algorithms, it is possible to infer hidden patterns using unsupervised learning techniques. Most of the existing unsupervised learning techniques require the number of patterns to be specified apriori for the learning of traffic patterns [40]. Nonparametric modeling of data is one of the key features when the number of hidden patterns are not known a priori. Learning the normal trajectory patterns is the key to understand the traffic behavior at roadways. Some of the typical traffic situations such as detection of congestion, illegal stoppages of vehicles, reverse driving, crashes, lane breaking, illegal U-turns, pedestrian movement on road can be categorized as anomaly [39] subjective to the context and application. If model of normal patterns of data can be learned, it will be easy to classify a new data as normal or abnormal. Moreover, the traffic behavior changes over time during the course of the day. Hence it is important for the model to be adaptive so as to discriminate the traffic behavior correctly.

Thus, an ideal system should be able to learn the scene in an unsupervised and nonparametric way. We propose an unsupervised learning method for the scene using video trajectories with non-parametric modeling based on Dirichlet Process Mixture Model (DPMM). We utilize the sequential dependencies among the trajectories that are typically ignored by the traditional clustering schemes [10], [14] to achieve better performance without compromising the accuracy. The proposed model, we refer to as Temporally Incremental Gravity Model (TIGM) is temporally extended as a Dynamic State Model (DSM) to understand the dynamics of different traffic paths. This can help the traffic authorities to understand how the traffic changes during the course of time such that better transportation strategies can be devised. Knowing the traffic state of roadways can also help the authorities to do efficient monitoring, congestion management, traffic flow analysis, traffic signal management, traffic planning for road construction, etc.

A. Related Work

Trajectory learning and classification are two important components of video analytic methods. Trajectory-based methods have been used for flow analysis [21], trajectory classification [4], [31], activity recognition [32], interaction analysis [12], abnormal object detection [8] and object classification and tracking [26]. The aforementioned research works use supervised approaches where labeled datasets are required. Unsupervised approaches have been used for vehicle behavior understanding [29], trajectory classification [3] and trajectory clustering [7], [16], [28], [33]. These methods employ unlabeled dataset to cluster similar trajectories and use clustered data to train the models for classification. Incremental clus-
tering has been used in trajectory modeling [16] and anomaly detection [35], where data is processed sequentially.

Typically trajectory learning methods use extracted trajectory features for clustering. Meanshift clustering is used in [33] and [2] with Discrete Wavelet Transform (DWT) coefficients and multi-feature vector, respectively. Particle swarm optimization based clustering used in [18] with Dynamic Time Wrapping (DTW) uses all the trajectory points. The work proposed in [41] performs trajectory clustering using instantaneous position as observation and trajectory as a mixture of topics to extract semantic regions in a scene.

The work presented in [16] performs trajectory clustering using Time-Sensitive Dirichlet Process Mixture Model (tDPMM) [46]. Authors of [43] have used adaptive multi-kernel-based estimation together with K-means to produce accurate clusters. The work in [13] uses Dirichlet Process Mixture model (DPMM) [36] to extract repeating patterns (Motifs) and their occurrence time in a scene.

A learned model representing trajectory patterns can be used for different purposes irrespective of the training methods employed. Such models can be used for abnormality detection [19], [23], [35], or classification and abnormality detection together [8], [21]. Trajectory retrieval [5], [16] is another possible application. Learned models can be used for online classification and abnormality detection even with partially observed tracks. The problem has been addressed in [4], [8], [21]. This is important when timely actions are to be taken in response to an observed event.

However, conventional convergence-based methods [9], [27], [36] suffer from lower accuracy, especially for clustering sequential data. Thus, we propose a new clustering method that is accurate and fast as compared to the state-of-the-art techniques when applied on trajectory as well as other domain data.

**B. Motivation and Research Contributions**

![Fig. 1. (a) Shows the tracks of objects in a typical traffic scene taken from QML dataset. (b) Represents the start (blue) and end points (green) of the tracks.](image)

A careful observation of start and end points of tracks shown in Fig. 1 reveals that they follow typical distributions and each of the distributions are distance apart. If the likelihood function can model the points in close proximity as a cluster, grouping of tracks can be done easily in a nonparametric way. Conventional DPMM, as expressed in (1 - 4) can be an ideal model for nonparametric modeling of data to learn distributions.

\[
z_i | \pi \sim \text{Discrete}(\pi) \quad (1)
\]

\[
x_i | z_i, \theta_k \sim F(\theta_{z_i}) \quad (2)
\]

\[
\pi = (\pi_1, \cdots, \pi_K) | \alpha \sim \text{Dirichlet}(\alpha/K, \cdots, \alpha/K) \quad (3)
\]

\[
\theta_k | H \sim H \quad (4)
\]

Here, \(x_i\) is a random variable representing the data and \(z_i\) corresponds to the latent variable representing cluster labels, where \(i = 1 \cdots N\) and \(N\) is the number of data points. \(z_i\) takes one of the values from \(k = 1 \cdots K\), where \(K\) is the number of clusters. \(\pi = (\pi_1, \cdots, \pi_K)\), referred to as mixing proportion, is a vector of length \(K\) representing the probabilities of \(z_i\) to be \(k\). \(\theta_k\) is the parameter of cluster \(k\) and \(F(\theta_{z_i})\) denotes the distribution defined by \(\theta_{z_i}\). \(\alpha\) denotes the concentration parameter and its value decides the number of clusters formed. Initially, we pick \(z_i\) from a Discrete distribution given in (1) and then generate data from a distribution parameterized by \(\theta_{z_i}\) as given in (2). Here, \(\pi\) is derived from a Dirichlet distribution as given in (3) and \(\theta_k\) is derived from distribution \(H\) of priors as represented in (4). The model [22] is graphically presented in Fig. 2(a). In our earlier work [37], we proposed a method to express the concentration parameter \((\alpha)\) in terms of distance measure using the deterministic relation \(\alpha = e^{-\beta}\), where \(\beta\) is referred to as concentration radius. We cannot directly use

![Fig. 2. (a) A conventional Dirichlet Process Mixture Model (DPMM). (b) Modified DPMM using parameter \(\beta\).](image)

the modified DPMM as shown in Fig. 2(b) for representing trajectory data as timing information is not considered in the model. In our proposed method, rather than representing the trajectories as a collection of unrelated data, we consider their temporal correlations. Moreover, as the concentration parameter is represented in terms of distance function, it can provide valuable intuition on how to choose the parameters for learning a traffic scene. It has been observed that, a single iteration of Gibbs sampling [34] is sufficient to associate tracks to clusters by learning the most frequently used paths in a given scene. In the proposed model DSM, learned parameters change temporally, thus allowing us to detect unusual events without the need to go through the entire data, unlike other statistical methods [10], [14], [20].
What can be categorized as normal events is highly contextual and it has a correlation in temporal domain. For example, during busy traffic, speed of a vehicle can be reasonably slow. If a person drives significantly faster as compared to other vehicles, that can be termed abnormal. However, in a sparse traffic, similar speed can be termed as normal. This is referred to as scene dynamics and it has been diligently addressed in our model. Another important characteristic is the temporal aspect, i.e. objects moving in a neighborhood around the same time have similar temporal correlations. To illustrate this, we assume traffic movements at a traffic junction signal. When the signal turns green, vehicles at the front of the queue starts moving followed by the vehicles behind. Similarly, there is a temporal relation between any moving objects entering and exiting the scene. These aspects are considered in the clustering, unlike other nonparametric, unsupervised methods such as DBSCAN [14] or mean shift [10].

Our main objective is to understand the spatio-temporal characteristics of the frequently present patterns from the trajectories obtained from a scene. This can be a possible building block for traffic behavior analysis. While accomplishing the above objective, we have made the following contributions:

(i) We propose a DPMM guided model that is referred to as Temporally Incremental Gravity Model (TIGM) and an inference scheme to create spatio-temporal clusters using conditional dependencies among the trajectories.

(ii) As typical DPMM clustering depends on the concentration parameter ($\alpha$) which is difficult to estimate, we have proposed a method to represent alpha using a distance-based method.

(iii) We also show how the proposed model can be used to estimate traffic state using Dynamic State Model (DSM) that has been built with the help of TIGM. We also discuss how DSM can be used for designing real-time traffic analysis frameworks.

(iv) It has been shown in this work that the proposed TIGM-DSM model can be used for clustering sequential data of other domains.

Rest of the paper is organized as follows. In Section II, rationale for building the proposed methodology is discussed. In Section III, we discuss experimental setup, datasets, parameters and analysis of results. In Section IV, we discuss other relevant methods in comparison to the proposed method followed by complexity analysis. We also discuss a few limitations of the proposed methods. Section V concludes our work with peek into the future directions of our work.

II. PROPOSED METHODOLOGY

Firstly, we discuss the terminologies used in the paper. The words observation and data are used interchangeably. Here, they represent the tracks or trajectories. Similarly, we refer cluster or topic to denote distribution of data and is represented by a label. They indicate the most frequently used paths. A model is used to represent a real-world phenomenon. We use graphical model [22] for representing the mixture models. A graphical model represents the generative model of the data. A parametric model has a fixed number of parameters, while the number of parameters grows with the amount of training data in non-parametric models. We follow an unsupervised method for learning a scene and our model is nonparametric. This characteristic is essential to address the scene dynamics, which is defined later.

A. Temporally Incremental Gravity Model (TIGM)

We have the following observations/assumptions to apply the proposed model for traffic analysis:

(i) The start or end points of the objects in motion constitute a distribution of data in 2D.

(ii) If tracks can be clustered based on the proximity, frequently used paths can be found.

In order to explain the TIGM, we discuss the inference process. Here, $x_i$ represents $i^{th}$ trajectory and $z_i$ be the corresponding cluster label. Our goal is to find $z_i$ for all tracks. $z_i$ corresponds to a discrete distribution and each $k \in \{1 \cdots K\}$ has a set of observations associated with it. The $k^{th}$ cluster has a proportion of observations. It can be observed that, $x_i$ is surely associated with one cluster of unknown distribution parameterized by $\theta_k = \{\mu_k, \Sigma_k\}$. Here, $\mu_k$ and $\Sigma_k$ represent mean and covariance of the distribution.

The inference method can be written using (5-6) [34], where $()_{-i}$ corresponds to the parameters excluding $i^{th}$ data. Here, (5) represents the probability of an observation becoming a new cluster while (6) represents the probability of an observation belonging to one of the K existing clusters. $F(x_i|\theta_k)$ corresponds to the likelihood of $x_i$ in $\theta_k$. $K = 0$, when no observation is considered. $n_{-i}$ denotes the number of observations for which clustering label is already assigned, excluding the observation $x_i$. $\alpha$ represents the concentration parameter of the Dirichlet distribution. These equations can be viewed as the posterior in terms of likelihood and prior. In other words, the probability of an observation belonging to a cluster $k$ (given other parameters) is proportional to the probability of the observation being generated from the cluster $k$ multiplied by the proportion of the observations present in the cluster.

$$p(z_i = k| z_{-i}, x_i, \theta_{k_{-i}}, \alpha) \propto F(x_i|\theta_{K+1}) \times \frac{\alpha}{n_{-i} + \alpha} \quad (5)$$

$$p(z_i = k| z_{-i}, x_i, \theta_{k_{-i}}, \alpha) \propto F(x_i|\theta_k) \times \frac{n_{k_{-i}}}{n_{-i} + \alpha} \quad (6)$$

We select a likelihood function in such a way that the probability of an observation to be associated with a cluster increases when the observation is nearer to the cluster mean. The association probability reduces as the observation goes away from the cluster mean. The exponential decay function $e^{-x}$ follows the above characteristics [37]. Here, $x$ represents the distance of $x_i$ from $\mu_k$. The inference equation can be written as (7). The condition, $k = K + 1$ denotes the probability of the observation forming a new cluster and the later part signifies the probability of $x_i$ going to one of the $K$ existing clusters.

$$p(z_i = k| z_{-i}, x_i, \theta_{k_{-i}}, \alpha) \propto e^{-x} \times \frac{\alpha_{n_{-i} + \alpha}}{n_{k_{-i} + \alpha}}; \text{if } k = K + 1;$$

$$e^{-x} \times \frac{\alpha_{n_{-i} + \alpha}}{n_{k_{-i} + \alpha}}, \text{otherwise.} \quad (7)$$
When an observation forms a new cluster, the likelihood function \( e^{-x} = 1 \) as the distance of an observation to itself is 0. The inference equation can be further simplified to (8) by removing the proportionality symbol and common denominator \( n_{-i} + \alpha \), where \( b \) is normalization constant and \( \alpha = e^{-\beta} \). We call \( \beta \) as the concentration radius. This inference process is derived without the assumption \( K \to \infty \), unlike [17], [34], [36].

\[
p(z_i = k | z_{-i}, x_i, \theta_{k_{-i}}, \beta) = \begin{cases} b \times e^{-\beta}, & \text{if } k = K + 1; \\ b \times e^{-x} \times n_{k_{-i}}, & \text{otherwise.} \end{cases}
\]

(8)

Above equation is the key to find the value of the concentration parameter (\( \alpha \)) as \( \beta \) can be expressed in terms of distance from the center to a point at periphery of the distribution. If an observation forms a new cluster, it has to be at an infinitesimal distance (\( \delta x \)) greater than that of the distribution’s periphery from the mean \( x = \text{max}(x) \). This implies, \( e^{-\beta} = e^{-(\text{max} + \delta x) \times n_{k_{-i}}} \). Therefore, the value of \( \beta \) can be estimated using (9).

\[
\beta = (\text{max} + \delta x) - \ln(n_{k_{-i}}).
\]

(9)

It may be observed that, \( \delta x \) can be ignored as \( \delta x << \text{max} \).
\( \ln(n_{k_{-i}}) \approx \ln(n_k) \), yields \( \beta = \text{max} - \ln(n_k) \). Once the initial clustering is done using an appropriate \( \text{max} \), actual \( \text{max} \) can be obtained and with some fine-tuning of \( n_k \), better clustering results can be produced.

The distance \( (x) \) can be euclidean distance, mahalanobis distance, dynamic time warping (DTW) [6], etc. We have used Euclidean distance even though mahalanobis also approximate elliptical distributions. This is done intentionally to limit distribution variance in \( XY \) plane. For real-time road traffic applications, DTW can be costly as the distance calculation itself is \( O(n_p^2) \), where \( n_p \) is the number of track points. However, DTW can be useful in applications such as handwriting matching, signature matching, etc. To simplify the model, we learn the posterior probability, not the likelihood function. The model is represented graphically as shown in Fig. 3(a).

We refer the model as Temporally Incremental Gravity Model (TIGM) for the following reasons: (i) The clusters are able to attract more observations from the neighborhood. This is not only due to the spatial proximity, but also for incremental temporal order. Thus, clustering does not strictly become bounded by distance. The gravitational strength (a large group of closely appearing observations attracts more nearby observations) is also increased with more observations getting added to the cluster. In order to illustrate this idea, let us consider a list of points equally apart in a straight line segment as shown in Fig. 3(b). Let us visualize how the mean of the points will shift if mean calculation is done incrementally. If we decide to cluster the points strictly based on a radius of half the length of the line segment, it may form more than one clusters based on the order of sampling. If done incrementally, it will form exactly one cluster. Moreover, even if we consider a radius less than half the line segment, our model can form a single cluster as mixing proportion factor attracts more observations to the cluster.

As explained earlier, when a signal turns green at a traffic junction, vehicles at the beginning usually move earlier. Unless front vehicles move, vehicles behind cannot move. In real life situations, this is exactly how many objects moves. This is the rationale behind modeling the conditional dependencies in TIGM. It means, clustering based on temporal order of their arrival can yield the best results as we consider the dependency of observations. In typical statistical approaches, where all observations are considered for clustering, multiple iterations of sampling may be required to achieve similar clustering results. (ii) Temporal information is maintained as inference is done in an incremental fashion. This model is different from DPMM as we represent the temporal correlation between observations and the corresponding latent variables using the conditional dependence. The generative process can be represented using (10-13).

\[
z_i | \pi, z_{i-1} \sim \text{Discrete}(\pi) \]

(10)

\[
x_i | z_i, x_{i-1}, \theta_k \sim F(\theta_{z_i}) \]

(11)

\[
\pi | e^{-\beta} \sim \text{Dirichlet}(e^{-\beta} / K, \cdots, e^{-\beta} / K) \]

(12)

\[
\theta_k | H \sim H \]

(13)

B. Dynamic State Model (DSM)

TIGM cannot give instantaneous state (for example highly crowded, moderately crowded, less crowded) of a scene as scene dynamics is not included in the model. We have extended the model to reflect the temporal states of a scene. Before proceeding further, we quantify what we referred to as scene dynamics. We denote it by \( \Phi^t \) subsequently. We
define $\Phi^t$ as a parameter vector of length $K^t$ such that $\Phi^t = \langle \Phi^t_1, \ldots, \Phi^t_{K^t} \rangle$. Here, $\Phi^t_k$ represents cluster dynamics in the $t^{th}$ temporal segment of length $\Delta t$ for the cluster $k^t$. It is defined as $\Phi^t_k = \langle \mu^t_k, \Sigma^t_k, n^t_k \rangle$, where $\mu^t_k$ denotes the mean of the cluster $k^t$, $\Sigma^t_k$ denotes its covariance and $n^t_k$ denotes the number of observations in cluster $k^t$. To complete the model, we have the following additional assumptions:

(i) When data is captured from a static camera, the cluster statistics do not change significantly in short duration.

(ii) If we take longer intervals, the traffic state can be completely different and clustering over a longer windows can be meaningful in real-time monitoring system.

(iii) If we fix the time segment over which the clustering is done, with prior information about the learned cluster, semi-supervised learning can be used for clustering and classification of trajectories in subsequent temporal segments.

Based on the above assumptions, new trajectories can be added and old trajectories that are not in the time segment can be removed from the cluster during learning. If Gibbs sampling is performed using $\theta^t_{k-1}$ as a prior for the $t^{th}$ frame, cluster labels can be maintained between consecutive frames. The inference can be done using (14) with exactly one iteration of the Gibbs sampling at a specified time interval. The rationale behind using a single iteration is to make sure that the temporal aspects are not lost in clustering. If we do re-sampling at specified intervals, stabilization of scene dynamics can be achieved. Here, $z_{t-i}$ is different from $z_{t-i}$ discussed earlier. It represents the set of all cluster assignments except for $x^t_i$, such that it excludes the observations and corresponding cluster assignments before $t-1$. $\theta^t_{k-i}$ is the parameter representing the distribution corresponding to cluster $k^t$ by excluding observation $x^t_i$. Here, $\mu^t_{k-i}$ is the mean of $\theta^t_{k-i}$ distribution. $n^t_{k-i}$ denotes the number of observations in $\theta^t_{k-i}$ and $b$ is the normalization constant.

$$p(z^t_i = k| z_{t-i}, x^t_i, \theta^t_{k-i}, \beta) = \begin{cases} \beta x^{-\beta}, & \text{if } k = K + 1; \\ \beta x^{-x \times n_{k^t_i}}, & \text{otherwise.} \end{cases}$$

Our proposed model is shown in Fig. 4. It can be represented using (15-18). Here, $x^t_i (i = 1 \cdots N_i)$ corresponds to the observation at time $t$ excluding the observations before $t-1$ and $z^t_i (i = 1 \cdots N_i)$ corresponds to the latent variable representing cluster labels, taking one of the values from $k = 1 \cdots K^t$. $N_i$ is the number of observations and $K^t$ is the number of clusters at $t$. $\pi^t$ is a vector of length $K^t$. $\pi^t_k = (\pi^t_1, \ldots, \pi^t_K)$ represents the mixing proportion of observations among the clusters. $\theta^t_k$ is the parameter of $k^{th}$ cluster and $F(\theta^t_k)$ denotes the distribution defined by $\theta^t_k$. First, we pick $z^t_i$ from a Discrete distribution given in (15). The data is then generated from a distribution parameterized by $\theta^t_k$, as given in (16), where $\pi^t$ is derived from a Dirichlet distribution as given in (17). $\theta^t_k$ is derived from another distribution $H$ of prior as represented in (18). The model is different from the original TIGM shown in Fig. 3. Here, $\theta^t_k \rightarrow \theta^t_{k-1}$ and $\pi^t \rightarrow \pi^{t-1}$. This is meaningful as the distributions do not change significantly within short intervals. For example, vehicle density usually does not vary significantly within short interval (in a span of 4-5 minutes) during peak hours. Thus, there is a conditional dependence of both $\theta^t_k$ and $\pi^t$ in between successive time slots.

$$z^t_i|\pi^t, z^t_{i-1} \sim \text{Discrete}(\pi^t)$$

$$x^t_i|z^t_i, z^t_{i-1}, \theta^t_k \sim F(\theta^t_k)$$

$$\pi^t|e^\beta, \pi^{t-1} \sim \text{Dirichlet}(e^\beta/K^t, \ldots, e^\beta/K^t)$$

$$\theta^t_k|H, \theta^t_{k-1} \sim H$$

The inference method for cluster assignment uses Gibbs sampling [34]. The process is described in Algorithm 1. Here, $x_i$ can be $<x_{start}, y_{start}, x_{end}, y_{end}, t_{dur}>$, where $<x_{start}, y_{start}>$ and $<x_{end}, y_{end}>$ represent start and end positions and $t_{dur}$ represents the length of the track. The rationale for using only these features (start and end points, and trajectory length) is that, in a road traffic scenario, most of the vehicles follow typical patterns. Therefore, those following similar patterns will be closely distributed in a multi-dimensional space. $x_i$ also can be $<x_{start}, y_{start}>, <x_{end}, y_{end}>$, and a combination of both. Then, it can be used to find the tracks originating at a particular position or/and ending at a particular regions.

### III. Experimental Results

In this section, we present results obtained using the proposed model applied on various public datasets.
Algorithm 1 Dynamic State Model (DSM)

**Input:** Input video, $\beta$, Length of time segment ($\Delta t$)

**Output:** Motifs

1: $K = 0$; $time = 0$; $prevtime = 0$;
2: Start a cyclic timer of length $\Delta t$ for updating time;
3: frame = Next Frame;
4: for !Empty (frame) do
5: Track the objects from previous frame;
6: Create $x_i$ for completed tracks;
7: for each newly created $x_i$ do
8: Find $z_i$ using (14);
9: end for
10: if ($time > pretime + 1$) then
11: un-assign $x_i$ arrived before $pretime$;
12: Resample and find $z_i$ using (14);
13: Display the frame with cluster labels;
14: $pretime = time$;
15: end if
16: frame = Next Frame;
17: end for

A. Experimental Setup and Datasets

We have used OpenCV to implement the proposed framework. Experiments have been conducted on four publicly available traffic video datasets, namely UCF [1], QML [24], MIT [41] and Grand Central Station, New York (GCS) [45]. UCF is a traffic junction video of 1 minute duration captured from the top of a building covering the vehicle movements across different roadways of a complex junction. QML is a video dataset of approximately 50 min duration of a busy traffic junction containing vehicles and pedestrian movement containing many anomalies. We have extracted the trajectories of the above datasets using context racker [11]. For TIGM experiments on QML, we have the trajectories of first 10 minutes duration. MIT is a video trajectory dataset of a parking lot captured from the top of a nearby building. GCS is also a trajectory dataset of containing the tracks of people motion at a busy underground metro station, namely Grand Central Station, New York. DSM experiments on QML dataset is conducted by extracting the trajectories for the entire video duration using [44].

B. $\beta$ Variation

Deciding a correct value of $\beta$ for clustering has to be more intuitive for a particular scene. From our analysis, we have observed that for start/end points-based clustering, initial $max$ value can be the approximate radius in the X-Y plane corresponding to the start/end point distribution. More than $n_k$, choosing correct $max$ value has bigger influence in clustering, as $\ln(n_k)$ is very small compared to $max$. Fine-tuning has been done using the term $\ln(n_k)$ depending on the number of trajectories in the clusters. For higher dimensions, initial $max$ value is set as two times (for additional two dimensions) or three times (for additional three dimensions) the fine-tuned $\beta$ value that has been obtained using previous step. Further fine-tuning is done similar to the method applied for lower dimensions.

As we have replaced the concentration parameter ($\alpha$) with the concentration radius ($\beta$), it is important to see how $\beta$ variation influences the clustering results. The results shown in Fig. 5, demonstrate that there is in inverse relation between $\beta$ and the number of clusters formed. However, the relation is more intuitive as compared to that of the relation between $\alpha$ and the number of clusters.

C. Convergence Vs. Correctness

The experiments on three datasets, namely QML, UCF and GCS reveal, in continuous monitoring applications, temporal correlation of the trajectories is more important than the convergence. To illustrate this point, start point based clustering is applied on QML dataset with 76 tracks, uses all five dimensions. Clustering on GCS dataset with 3500 tracks also uses all five dimensions. Fig. 6 shows the errors for QML, UCF, and GCS.

Fig. 5. Depiction of how $\beta$ influences the number of clusters formed. (a) Plot of $\beta$ vs. number of clusters. Plot also shows the $\alpha$ values to be used to get the same number of clusters. (b-d) The clustered trajectories for three different $\beta$ values.

Fig. 6. Depiction of Convergence process of three datasets. In QML, with 146 tracks, start point based clustering is used. Clustering on UCF dataset with 76 tracks, uses all five dimensions. Clustering on GCS dataset with 3500 tracks also uses all five dimensions.
this grouping can be acceptable due to the remoteness of track 38 from cluster 2, tracks of cluster 12 should logically belong to cluster 9. Such incorrect clustering happens due to the loss of temporal information with convergence. The convergence plots for QML, UCF, and GCS datasets are shown in Fig. 6. Though convergence using multiple iterations is faster, clustering results are not meaningful. Hence, single iteration of Gibbs sampling has been adopted.

D. Experiments on Vehicle Trajectory Datasets

Clustering results are presented in Figs. 8 - 10. As expected, $\beta$ variation gives different groupings of tracks. However, acceptable $\beta$ values can be derived as discussed earlier. The tracks are incrementally clustered to make sure subsequent tracks following similar same path are grouped together. This happens due to the neighboring positions of the new tracks with that of previous one in the multi dimensional space. The clusters with more number of tracks form Motifs (frequently occurring trajectory patterns) [13]. Individual clusters are shown with color gradients indicating the motion direction. Tracks start with red, goes through a change of color and ends with pink to demonstrate temporal effect. This representation is used to give clear idea of the direction of the trajectory and are useful in representing Motifs as discussed in [13].

In Figs. 8 - 9, clustering has been performed considering all five dimensions of the tracks. The results clearly show that trajectories in a cluster are similar in nature. Though a few clusters in Fig. 9 look similar, they have been grouped separately due temporal variations. It has been observed that the number of clusters formed in MIT dataset is high. This is due to a large number of entry and exit points in the parking area, where many vehicles did not strictly follow the paths while moving in or out. If there is no restrictions on the paths to be taken by the drivers, clustering may be performed based on entry or exit points to find out most frequently used entry and exit points, as given in Fig. 11.

E. Experiments on Crowd Dataset

Applicability of the model on crowd datasets has been tested to understand the underlying patterns of crowd movements. Experiments on a difficult dataset like GCS reveal that the proposed method can produce interesting results, including detection of frequently used paths, unusual patterns of people movement, most frequently used entry and exit points, etc. We have used the tracks of length larger than 400 to avoid the truncated trajectories. Clustering results as demonstrated in Figs. 12 - 14 reveal interesting patterns including most
Fig. 10. Depiction of trajectory clustering applied on MIT dataset with TIGM ($\beta = 50$) using start and end positions. (a - i) Prominent clusters are shown in color gradient form. Unlike the UCF and QML datasets, number of clusters formed is large. However, we have shown only the top 9 clusters. (j - l) Less frequent or abnormal clusters.

Fig. 11. Depiction of trajectory clustering applied on MIT Parking dataset based on start point and end points using TIGM ($\beta = 70$). (a - f) Six prominent clusters based on start points. (g - l) Six prominent clusters based on end points.

Fig. 12. Depiction of trajectory clustering applied on GCS dataset using TIGM ($\beta = 170$). (a) Original trajectories. (b - l) Clusters representing the noticeable patterns. (m - p) Clusters representing less frequently occurring patterns. (q - t) Clusters representing unusual trajectories.

F. Experiments using DSM

In order to conduct experiments for detecting scene dynamics, we have used GCS and QML datasets. In order to depict the appearance of the tracks temporally, a plot of the frequency of new tracks vs. appearance time is shown in Figs. 15 - 16 for the GCS dataset. The inference is applied based on start points by taking a time segment ($\Delta t$) of 10K frames for the DSM model. The method reveals the scene dynamics representing different underlying states of the scene in temporal plane. This information can be used for crowd management at the station. For example, deciding the number of service personal for issuing tickets or better scheduling of trains can be done based on scene dynamics. The results of clustering, when observed in conjunction with the video, clearly reflect the scene dynamics. In order to illustrate the results, we have taken three clusters corresponding to three entry points in the scene. Fig. 17 shows the dynamics of different entry points, i.e. $\Phi_k$, at equally spaced four time segments. Figs. 17 (b) - (e) reveal that, the crowd density reduces by $t_3$ and again increases slightly by $t_4$ at the bottom-right entry point. However, the dynamics do not change significantly for the clusters originating at the middle-bottom entry point as can be seen from Figs. 17 (f) - (i). Density is observed to be highest among all the entry points. Clusters corresponding to the top-left entry point are lightly dense and the dynamics are not changing over time as can be seen from Figs. 17 (j) - (m). It reveals that, traffic inflow is lesser at this entry point irrespective of time. The above mentioned behavior can be verified from the histograms of arrival frequency, as shown in Fig 16. This means, our model is able to represent the scene dynamics correctly as seen in Fig. 17. In this experiment, we have considered entry point-based clustering on DSM. However, this can be extended by
employing a combination of all five dimensions to get the states of traffic.

For the QML dataset, we have used a $\Delta t$ of 15K frames. Noisy trajectories are removed so as to contain only the vehicle trajectories. The results of the experiments are presented in Figs. 18 - 19. The results reveal important information on how the traffic changes over time at the junction. This information can be used effectively for better scheduling of traffic signals apart from sharing the information to the subsequent signals in a sequential camera network.

Here, we want to highlight how cluster dynamics represented in terms of $\Phi_k^t$ can be useful for futuristic traffic applications as it represents a set of features. Hence it can be further grouped using different features such as traffic speed, traffic density, spread of trajectory distribution to get meaningful interpretation of traffic scene. For example, traffic can be classified as low density, medium density or high density. Traffic at a particular roadway can be observed on how the densities change over time. This can help the authorities for better planning of road traffic. This means, cluster dynamics can be mapped to states (low, medium or high), to reflect traffic speed in a path, traffic density, degree of chaos based on spread of distribution, etc. Clusters can be collected for all time segments ($t$) of the day based on the trajectories found from a scene. Using $\Phi_k^t$ as an observation, clustering can be done to learn different states either by using hard threshold or by using unsupervised techniques, such as K-means [20]. Once the state parameters are known, dynamics can be mapped to different states for meaningful interpretation of traffic scene. Such analysis can help the traffic authorities for better traffic situation assessment, transportation planning, scheduling of traffic at junctions, routing of traffic through different roadways, etc.

![Fig. 13](image1.png)

Fig. 13. Depiction of Trajectory clustering applied on GCS dataset based on destination of trajectories using TIGM ($\beta = 70$). (a - l) Prominent clusters. Clusters clearly indicate the entry paths as well as the place, where people generally stop mid-course to decide about further action.

![Fig. 14](image2.png)

Fig. 14. Depiction of Trajectory clustering applied on GCS dataset based on source of trajectories using TIGM ($\beta = 100$). (a - h) Prominent Clusters. Clusters clearly indicate the entry paths as well as the place, where people usually resume their movement.

![Fig. 15](image3.png)

Fig. 15. Histogram depicts the number of new tracks against their arrival time in GCS.

![Fig. 16](image4.png)

Fig. 16. Histograms of trajectory arrival frequency for different clusters from GCS. (a) Trajectory arrival frequency at bottom-right entry point. (b) Trajectory arrival frequency at middle-bottom entry point. (c) Trajectory arrival frequency at top-left entry point.

![Fig. 17](image5.png)

Fig. 17. Depiction of pattern dynamics as represented by clusters ($\beta = 100$) at different time segments for different entry points in GCS data. (a - d) Dynamic behavior of the scene for the bottom-right entry point. (e - h) Dynamic behavior of the scene for the middle-bottom entry point. (i - l) Dynamic behavior of the scene for the top-left entry point.

**G. Model Evaluation and Comparison**

Next, we compare the clustering approach with two unsupervised and nonparametric approaches, namely mean shift [10] and DBSCAN [14]. We have applied clustering on QML dataset based on start point. The results are shown in Fig. 20. Our method outperforms these techniques in most cases. With proper tuning these methods can perform similarly.
Fig. 18. Depiction of DSM on QML dataset using start and end positions with a $\beta = 70$ on selected clusters. It may be observed that the traffic flow is almost uniform across different time segments for the clusters corresponding to first three rows. However, change of cluster dynamics can be observed in the last three clusters.

Fig. 19. Depiction of DSM on QML dataset using end positions with a $\beta = 50$ on selected clusters. Heavy traffic flow can be observed consistently for the north bound traffic (row 1) as compared to east-bound (row 2) or west-bound traffic (row 3). The information depicted through DSM as significant role as the changes in the traffic flow affects the subsequent signals.

However, in some cases, these techniques fail to include all trajectories in the right cluster. For example, as depicted in Fig. 20, track 164 is getting clustered into cluster 10. Our proposed method resolves such cases. However, we cannot claim it is incorrect clustering (done by DBSCAN) since the criteria used are different. Similarly, in mean shift clustering, tracks 89 and 99 are present in cluster 8. Even though track 89 can be logically grouped to cluster 8, in TIGM and DBSCAN, it forms a new cluster. It may be wrong to group track 99 with cluster 8. Ideally it can be part of cluster 9, as in TIGM and DBSCAN.

In order to evaluate these methods quantitatively, we have conducted experiments to calculate learning accuracy, precision and recall. Firstly, logical grouping of clusters has been done by an independent evaluator to prepare ground truths. Ground truths correspond to unique trajectory patterns that are possible in a typical traffic scene. For example, in the case of road ground truth represents the trajectory patterns on the road. While in GCS, they are the common movement patterns between a pair of source and destination. Wrong inclusion of a trajectory in the predicted group is categorized as False Positive (FP), and wrong exclusion from a coherent group is categorized as False Negative (FN). Rest of the trajectories are either True Positives (TP), or True Negatives (TN). The computational analysis on QML dataset using start point based clustering is depicted in Figs. 21 - 24. Fig. 21 reveals that TIGM takes linear time. The plots are shown to demonstrate the predictability of our model. It is observed that DBSCAN matches with TIGM in cluster prediction with respect to ground truth and their accuracies. Though the cluster prediction of mean shift gets better with more tracks, accuracies are found to be unpredictable.

Table I summarizes the experimental results in terms of its accuracy, precision and recall. For UCF (76 trajectories) and QML (146 trajectories), all five dimensions are used. In MIT (449 trajectories) and GCS (initial 1000 trajectories) datasets, we have omitted the duration ($t_{dur}$), as it was difficult to clearly discriminate between shorter and longer trajectories with same start and end positions. The parameter values are adjusted to create common patterns corresponding to source destination pairs such that the methods can be compared across various datasets. It may be observed that the performance of TIGM is close to that of DBSCAN when applied on road trajectories. However, in GCS it performs much better as compared to both DBSCAN and MeanShift. From the analysis, it has been found that DBSCAN performs poorly due to the spread of starting and ending points in the scene. When the trajectories are dense at a particular locality, DBSCAN typically groups the nearby trajectories to form bigger groups as illustrated in Fig. 25. By lowering $\epsilon$, such effect can be avoided. However, this leads to higher number of cluster formation causing accuracy reduction due to higher false negatives.

H. Discussion of Results

Trajectory based learning is one of the most common ways to analyze and interpret activities in a scene. From the
detection and classification can be done. Statistics correspond-
more specific features such as velocity or curvature, activity
behavior analysis can be done using the learned model. With
represented in (8). For a new trajectory, classification and
identified as normal or abnormal using posterior probabilities
with representative trajectories), the learned patterns can be
a semi-supervised approach (by preinitializing normal clusters
is able to learn unique trajectory patterns in the scene. Using
experiments, it has been observed that the proposed method
is able to learn unique trajectory patterns in the scene. Using
a semi-supervised approach (by preinitializing normal clusters
with representative trajectories), the learned patterns can be
identified as normal or abnormal using posterior probabilities
represented in (8). For a new trajectory, classification and
behavior analysis can be done using the learned model. With
more specific features such as velocity or curvature, activity
detection and classification can be done. Statistics correspond-
ing to unique trajectory patterns can be used by traffic au-
authorities for better planning of road networks, safety analysis,
risk assessment, etc. The Temporal extension of TIGM, i.e.,
the DSM reflects the traffic states in the temporal domain.
This is useful in traffic flow analysis, congestion study, traffic
density analysis, adaptive traffic signal scheduling, traffic re-
routing strategies as the cluster dynamics provide valuable
traffic information about the value at specific paths, entry and
exit points. For example, cluster density at a traffic path can
be used for better signal duration prediction for a particular
lane, while traffic volume information at the start or end
points can be used for better traffic routing at the preceding or
proceeding traffic signals with suitable communication through
centralized/ Internet of Things (IoT) devices.

I. Performance Comparison

In addition to nonparametric clustering algorithms, we com-
pared the performance of other unsupervised methods, namely
Minibatch K-means (MK-means) [38], Ward [30], Affinity
propagation (AP) [15], Spectral [25], Gaussian Mixture Model
(GMM) [27]. TIGM has been found to be very fast as
compared to other unsupervised clustering methods as shown in
Fig. 26.

| Dataset | Method | A (%) | P (%) | R (%) |
|---------|--------|-------|-------|-------|
| UCF     | TIGM (β = 240) | 97.37 | 98.67 | 98.67 |
|         | DBSCAN (ε = .83, m.s = 1) | 98.68 | 100 | 98.68 |
|         | Meanshift (q = 0.07) | 97.37 | 100 | 97.37 |
| QML     | TIGM (β = 170) | 92.47 | 95.74 | 96.43 |
|         | DBSCAN (ε = .83, m.s = 1) | 91.10 | 97.08 | 93.66 |
|         | Meanshift (q = 0.0459) | 89.73 | 97.04 | 92.25 |
| MTT     | TIGM (β = 50) | 90.00 | 96.43 | 93.10 |
|         | DBSCAN (ε = .27, m.s = 1) | 91.28 | 94.18 | 96.74 |
|         | Meanshift(q = 0.01) | 80.28 | 94.31 | 84.36 |
| GCS     | TIGM (β = 120) | 84.80 | 94.43 | 89.26 |
|         | DBSCAN (ε = .31, m.s = 1) | 47.20 | 80.55 | 53.27 |
|         | Meanshift(q = 0.01) | 67.80 | 85.39 | 76.70 |

Fig. 21. Plots showing the relationship of clustering time against
the number of observations based on QML dataset considering only start
point. Plot clearly shows the linear performance of TIGM as compared to
other methods.

Fig. 22. Plots comparing the relationship between number of trajectories
and number of clusters formed for the three unsupervised nonparametric
methods (mean shift, DBSCAN and TIGM). Plot of TIGM is overlapping
with DBSCAN at most of the points.

Fig. 23. Ground truth (GT) Vs. Actual number of clusters. Though the num-
ber of obtained clusters are different from ground truth as more trajectories
arrive, all methods are able to come closer to the ground truth. TIGM and
DBSCAN are able to learn them even with less number of trajectories.

Fig. 24. Accuracy of clustering plotted for different number of trajectories
for QML dataset based on start point. Accuracy of DBSCAN and TIGM
are comparable. However, mean shift gives unpredictable accuracy.

Fig. 25. Depiction of cluster merging in DBSCAN and meanshift as compared
to that of TIGM. (a) Three meaningful clusters using TIGM. (b) High
concentration of trajectories in a path tends to merge the trajectories into
a single cluster. (c) Measnsihft also does similar cluster merging.
J. Experiments in Other Domains

Due to its generic nature, DSM can be used in different domains. It can be applied where data points have temporal relations and the dynamics change over time. This includes grouping of people discussing similar ideas around the same time in social networks or over phone calls, mobile devices with GPS over a geographical area for giving better directions to the users to avoid crowd in roads; analyzing how network bandwidth is varied over time; calculating radiating power required for handling cell phones based on the number of people in cell, etc.

We have carried out an experiment on the data usage of different customers of an ISP over time using the proposed DSM. We have used a publicly available dataset in the form of pcap file [42]. We have mapped the source IP to a customer and the average data usage to be the value of random variable \( x_i \), as this information is readily available with the ISP. Fig. 27 shows a typical representation of the network considered. There are 41 source IPs corresponding to the customers. It may be observed that the data usage changes over time for different customers. We have used a random time 100s segment length to demonstrate the scene dynamics. Fig. 28 shows how the user behavior changes for a particular cluster over time. ISPs can use this for adaptive scheduling of network bandwidth for users when some customers are not using the bandwidth.

IV. DISCUSSION

A. Key Aspects: TIGM-DSM Vs. Relevant Methods

In spite of producing strikingly similar results in terms of coherent groups, proposed TIGM has other advantages over existing methods. Our method clusters the trajectories in \( \Theta(nk) \) time, while mean shift takes \( O(kn\log(n)) \) for lower dimensions the time complexity tends towards \( O(kn^2) \). Here, \( n \) is the number of observations and \( k \) the number of clusters. DBSCAN does clustering in \( O(n\log(n)) \) in the average case and \( O(n^2) \) in the worst case. DBSCAN and mean shift clustering algorithms are statistical in nature, i.e. they consider all data at a time. Such methods are useful for data mining related analysis. Though our method can be used in a statistical way by going for iteration convergence, temporal information may be lost in the clustering process. DBSCAN and mean shift have significance in clustering in a wide variety of applications. In traffic monitoring applications, trajectories arrive in temporal order, one at a time, making the incremental model ideal to represent scene dynamics. Moreover, by introducing the parameter \( \beta \) (concentration radius) as a distance, it is intuitive to estimate the parameter unlike DBSCAN or mean shift. Also, proposed clustering starts with no prior (in the inference), rather we build the prior incrementally during the clustering process. Inference mechanism allows us to unassign any observation without affecting the cluster labels. This is the key in building the DSM that can predict/describe states of the scene across different time frames. A model supporting different states is a key feature of the proposed method as there can be different rules at different time segments of the day. Thus, abnormal activity detection is more adaptive rather than following a strict threshold in traffic monitoring applications.

Moreover, there is no need to map the cluster labels as they are maintained in the temporal plane.

Like Dynamic-Dual-HDP [41], the proposed DSM is able to model a scene dynamically, thus catering to learning scenes dynamically. Moreover, only essential information need to be stored using the proposed method, rather than storing complete trajectory. However, Dynamic-Dual-HDP looks for convergence, thus losing some important temporal characteristics. Unlike [46], we handle temporal aspects inherently by using observations in temporal order. In the work [16], though a new model based on DFT features has been introduced, it does not...
explore much on trajectories from different contexts. We have done detailed semantic analysis using only five feature points. Moreover, the proposed method has the capability to handle scene dynamics as done in DSM with a simple extension of TIGM. Even without handling the shrunk trajectories, we are able to produce meaningful semantic regions on a difficult dataset like GCS as compared to [43].

B. Complexity Analysis

As discussed earlier, TIGM clustering complexity is $\Theta(nk)$. Even with DSM, the algorithm complexity does not change. Our algorithm takes one observation at a time and inference is applied to get clusters. At the time of unassigning the older observations from the cluster, exactly one more revisit is required on any observation. Each of the observations is checked against each of the $k$ clusters, exactly once. If resampling is involved, it is done at most twice. Thus, if there are $n$ trajectories and $k$ clusters, worst case complexity of the clustering is $\Theta(nk)$. Under normal circumstances, the value $k$ will be much smaller than $n$, hence the complexity can be approximated by $\Theta(n)$.

C. Limitations

For the simplicity of modeling, we have assumed a single parameter for the entire clustering. However, it may happen that, different clusters can have distributions of varying spread. In such cases, $\beta$ needs to be learned to avoid observations going to wrong clusters, even though they are spatially apart. This can happen to observations belonging to less frequent patterns with their means closer to a cluster of higher spread. Our method cannot be used in situations where trajectory patterns are random in nature, as accuracy may be affected due to the approximation of points to start and end positions along with duration. The proposed method is suitable for applications to monitor road traffic or crowd traffic that are supposed to follow patterns. By incorporating more dimensions like curvature, slope or direction, better accuracy can be achieved. The proposed model does not restrict to use other suitable distance measures. If noisy trajectories such as truncated tracks are present, the clusters formed may be the true representation of the scene as this method relies on the accuracy of tracking algorithms. Such noisy trajectories typically form new clusters as their characteristics (or features) are different from the normal ones. With the advances in machine learning, better tracking algorithms are available, thus such situations can be avoided.

V. CONCLUSION

The paper introduces a trajectory model with the help of DPMM through an intuitive perspective by using a distance measure and temporal correlation of the observations. The model is temporally extended to consider the dynamic behavior of a given scene to reflect the traffic states. An incremental approach has been used to build clusters without a prior and indicative number of clusters as required in unsupervised learning methods. The model has been validated on a wide range of video datasets. The proposed model is able to cluster trajectories meaningfully with high accuracy taking lesser computation time. Our model can be applied to videos for building real-time traffic analysis framework since it can learn and refine frequently occurring patterns in an unsupervised and nonparametric way. Though the primary focus of building the proposed model is for visual surveillance applications, it can be used in applications requiring representation of changing scene dynamics involving observations of temporal dependencies.

We foresee room for improvement at various levels. Firstly, we need to learn likelihood to cater distributions of different spreads. Secondly, Mahanolabis distance can be used for better approximation of elliptical distributions. Lastly, application for real-time monitoring of road traffic can be built.

REFERENCES

[1] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In CVPR, 2007.
[2] N. Anjum and A. Cavallaro. Multifature object trajectory clustering for video analysis. IEEE Transactions on Circuits and Systems for Video Technology, 18(11):1555–1564, Nov 2008.
[3] V. Bastani, L. Marcenaro, and C. Regazzoni. Unsupervised trajectory pattern classification using hierarchical dirichlet process mixture hidden markov model. In MLSP, 2014.
[4] V. Bastani, L. Marcenaro, and C. Regazzoni. A particle filter based sequential trajectory classifier for behavior analysis in video surveillance. In ICIP, 2015.
[5] V. Bastani, L. Marcenaro, and C. S. Regazzoni. Online nonparametric bayesian activity mining and analysis from surveillance video. IEEE Transactions on Image Processing, 25(5):2089–2102, May 2016.
[6] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In KDD, 1994.
[7] B. Cancela, A. Iglesias, and M. G. Ortega. Contextualised trajectory modelling using temporal information via minimal paths. In CVPR, 2014.
[8] F. Castaldo, F. A. N. Palmieri, V. Bastani, L. Marcenaro, and C. Regazzoni. Abnormal vessel behavior detection in port areas based on dynamic bayesian networks. In ICF, 2014.
[9] D. Comaniciu and P. Meer. Distribution free decomposition of multivariate data. Pattern Anal. Appl., 2(1):22–30, April 1999.
[10] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence, 24(5):603–619, 2002.
[11] T. B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In CVPR, 2011.
[12] A. Dore and C. Regazzoni. Interaction analysis with a bayesian trajectory model. IEEE Intelligent Systems, 25(3):32–40, May 2010.
[13] R. Emonet, J. Varadarajan, and J. M. Odobez. Temporal analysis of motif mixtures using dirichlet processes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(1):140–156, Jan 2014.
[14] M. Ester, J. Kriegel, H. P. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In KDD, 1996.
[15] B. J. Frey and D. Dueck. Clustering by passing messages between data points. science, 315(5814):972–976, 2007.
[16] W. Hu, X. Li, G. Tian, S. Maybank, and Z. Zhang. An incremental dpmm-based method for trajectory clustering, modeling, and retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(5):1051–1065, May 2013.
[17] H. Ishwaran and M. Zarepour. Exact and approximate sum representations for the dirichlet process. Canadian Journal of Statistics, 30(2):269–283, 2002.
[18] Z. Izakian, M. S. Mesgari, and A. Abraham. Automated clustering of trajectory data using a particle swarm optimization. Computers, Environment and Urban Systems, 55:55–65, 2016.
[19] F. Jiang, Y. Wu, and A. K. Katsaggelos. A dynamic hierarchical clustering method for trajectory-based unusual video event detection. IEEE Transactions on Image Processing, 18(4):907–913, April 2009.
[20] X. Jin and J. Han. K-Means Clustering, pages 563–564. Springer US, Boston, MA, 2010.

[21] K. Kim, D. Lee, and I. Essa. Gaussian process regression flow for analysis of motion trajectories. In ICCV, 2011.

[22] D. Koller and N. Friedman. Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning. The MIT Press, 2009.

[23] R. Laxhammar and G. Fulkman. Online learning and sequential anomaly detection in trajectories. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(6):1158–1173, June 2014.

[24] C. C. Loy, T. Xiang, and S. Gong. From local temporal correlation to global anomaly detection. In ECCV, 2008.

[25] U. V. Luxburg. A tutorial on spectral clustering. Statistics and computing, 17(4):395–416, 2007.

[26] L. Marcenaro, L. Marchessotti, and C.S. Regazzoni. Self-organizing shape description for tracking and classifying multiple interacting objects. Image and Vision Computing, 24(11):1179 – 1191, 2006.

[27] T. K. Moon. The expectation-maximization algorithm. IEEE Signal Processing Magazine, 13(6):47–60, 1996.

[28] B. T. Morris and M. M. Trivedi. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(11):2287–2301, Nov 2011.

[29] B. T. Morris and M. M. Trivedi. Understanding vehicular traffic behavior from video: a survey of unsupervised approaches. Journal of Electronic Imaging, 22(4):044113–044113, 2013.

[30] F. Murtagh and P. Legendre. Ward’s hierarchical agglomerative clustering method: Which algorithms implement ward’s criterion? J. Classif., 3(3):274–295, October 2014.

[31] J. C. Nascimento, M. A. T. Figueiredo, and J. S. Marques. Trajectory classification using switched dynamical hidden markov models. IEEE Transactions on Image Processing, 19(5):1338–1348, May 2010.

[32] J. C. Nascimento, M. A. T. Figueiredo, and J. S. Marques. Activity recognition using a mixture of vector fields. IEEE Transactions on Image Processing, 22(5):1712–1725, May 2013.

[33] T. Nawaz, A. Cavallaro, and B. Rinner. Trajectory clustering for motion pattern extraction in aerial videos. In ICIP, 2014.

[34] R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. Journal of Computational and Graphical Statistics, 9(2):249–265, 2000.

[35] C. Picarelli, C. Micheloni, and G. L. Foresti. Trajectory-based anomalous event detection. IEEE Transactions on Circuits and Systems for Video Technology, 18(11):1544–1554, Nov 2008.

[36] C. E. Rasmussen. The infinite gaussian mixture model. In S. A. Solla, T. K. Leen, and K. Müller, editors, Advances in Neural Information Processing Systems 12, pages 554–560. MIT Press, 2000.

[37] K. K. Santhosh, D. P. Dogra, and P. P. Roy. Temporal unknown incremental clustering model for analysis of traffic surveillance videos. IEEE Transactions on Intelligent Transportation Systems, pages 1–12, 2018.

[38] D. Sculley. Web-scale k-means clustering. In WWW, 2010.

[39] A. A. Sodemann, M. P. Ross, and B. J. Borgohri. A review of anomaly detection in automated surveillance. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(6):1275–1281, Nov 2012.

[40] B. Tian, B. T. Morris, M. Tang, Y. Liu, Y. Yao, C. Gou, D. Shen, and S. Tang. Hierarchical and networked vehicle surveillance in: A survey. IEEE Transactions on Intelligent Transportation Systems, 18(1):25–48, Jan 2017.

[41] X. Wang, K. T. Ma, G. W. Ng, and W. E. L. Grimson. Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. International journal of computer vision, 95(5):287–312, 2011.

[42] J. Weber. Wireshark Layer 2-3 pcap Analysis w/ Challenges. Online-http://www.netsec.eu/, 2017.[Online https://blog.webernetz.net/2017/03/29/wireshark-layer-2-3-pcap-analysis-w-challenges-ccnp-switch/; accessed 25-May-2017].

[43] H. Xu, Y. Zhou, W. Lin, and H. Zha. Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage. In ICCV, 2015.

[44] B. Zhou, X. Tang, and X. Wang. Measuring crowd collectiveness. In CVPR, 2013.

[45] B. Zhou, X. Wang, and X. Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In CVPR, 2012.

[46] X. Zhu, Z. Ghabramani, and J. Lafferty. Time-sensitive dirichlet process mixture models. Technical report, DTIC Document, 2005.