entropy

MDPI

Article

Trajectory Shape Analysis and Anomaly Detection Utilizing Information Theory Tools †

Yuejun Guo [1,2], Qing Xu [1,*], Peng Li [1], Mateu Sbert [1,2,*] and Yu Yang [1]

1 School of Computer Science and Technology, Tianjin University, Yaguan Road #135, Tianjin 300350, China; guoyuejun13@gmail.com (Y.G.); liptju@gmail.com (P.L.); yang691@usc.edu (Y.Y.)

2 Graphics and Imaging Lab, Universitat de Girona, Campus Montilivi, 17071 Girona, Spain

* Correspondence: qingxu@tju.edu.cn or qingxu.itcn@gmail.com (Q.X.); mateu@ima.udg.edu (M.S.); Tel.: +86-22-27406538 (Q.X.); +34-972-418419 (M.S.)

† This paper is an extended version of our paper published in 22nd International Conference on Neural Information Processing, Istanbul, Turkey, 9–12 November 2015, pp. 423–431.

Received: 15 February 2017; Accepted: 27 June 2017; Published: 30 June 2017

Abstract: In this paper, we propose to improve trajectory shape analysis by explicitly considering the speed attribute of trajectory data, and to successfully achieve anomaly detection. The shape of object motion trajectory is modeled using Kernel Density Estimation (KDE), making use of both the angle attribute of the trajectory and the speed of the moving object. An unsupervised clustering algorithm, based on the Information Bottleneck (IB) method, is employed for trajectory learning to obtain an adaptive number of trajectory clusters through maximizing the Mutual Information (MI) between the clustering result and a feature set of the trajectory data. Furthermore, we propose to effectively enhance the performance of IB by taking into account the clustering quality in each iteration of the clustering procedure. The trajectories are determined as either abnormal (infrequently observed) or normal by a measure based on Shannon entropy. Extensive tests on real-world and synthetic data show that the proposed technique behaves very well and outperforms the state-of-the-art methods.

Keywords: trajectory shape analysis; trajectory clustering; anomaly detection; Kernel Density Estimation; Mutual Information; Shannon entropy

1. Introduction

In the present society, trajectory data, embodying motion characteristics of moving objects, are being largely used in many application fields, such as video surveillance for the sake of safety and security [1,2], and anomaly detection for air traffic [3]. The pattern analysis of motion trajectory data is a central task in practical scenarios, and clustering is a main technique of trajectory analysis [4,5]. With the patterns learned, the testing trajectories are recognized as either normal or abnormal, namely, either frequently observed or not [6].

Notably, due to its simplicity and effectiveness, the shape of a trajectory is one of the commonly used descriptions for trajectory analysis and anomaly detection [7–9], and our paper focuses on shape analysis and anomaly detection. The shape of a motion trajectory is classically characterized by a series of tangential angles at the positions of the corresponding moving object (for the sake of easy description, in this paper, we call these positions as the sample points of the trajectory), and these angles are usually modeled by using the von Mises distribution to represent the trajectory [7,8,10]. However, despite the fact that the angle attribute is largely helpful for a lot of practical applications, the use of only this single attribute cannot discriminate, for instance, whether a person is walking or running along a same path in an automated surveillance system. This means that the speed attribute is also fundamentally important for the shape modeling of trajectory data [11]. Even when both the angle and speed attributes are considered, a trajectory is defined by using some classical distributions, such as Approximated

Wrapped and Linear Gaussian (AWLG) [12], relying on the assumption that these distributions fit the real-world trajectory data fairly well. However, in practice it is not clear whether this fitting is close or not to the true distribution. As for the shape clustering for trajectory data, *k*-medoids plays a major role [7,12], but this kind of solution resorts to a predefined number of the clusters and to an explicit distance measure between trajectories, which are usually very difficult for complex trajectory data [13]. The Information Bottleneck (IB) theory [14] has been introduced to handle these issues very well [15], while unfortunately the practical execution of IB is based on an iterative procedure resulting in the local optima problem [16]. Lastly, but also importantly, abnormal detection happening in the context of trajectory shape analysis usually uses a crisp threshold to determine whether a trajectory is abnormal or not [7,12], but this obviously does not work for varied kinds of trajectory datasets.

To deal with the problems mentioned above, this paper proposes a novel effective technique, based on shape modeling and clustering, to analyze whether a trajectory is either recurring or not, namely, either normal or anomalous. The work presented here is an improvement and extension of our preliminary version [15] with the following new contributions:

1. For the trajectory modeling, we additionally take into account the speed attribute of the moving object, leading to a richer representation than relying solely upon the angle description for trajectory shape analysis. In order to avoid the difficult fitting problem for parametric models such as von Mises and AWLG, the general Kernel Density Estimation (KDE) is developed to obtain the probability distribution function (PDF) as the model of the trajectory shape.
2. In practical implementation, probability bins are taken to approximate the shape PDF. For the sake of reducing the computational cost, we propose to take advantage of the local probability extremas to adaptively determine the probability bins used for trajectory modeling.
3. The Information Bottleneck (IB) principle [14] partitions trajectories into clusters via minimizing the objective function of the loss of Mutual Information (MI) [17] between the trajectory dataset and a feature set of these trajectories (for conciseness we call this the MI based clustering in our paper). To alleviate the problem of local optima, we propose to improve the clustering performance by introducing an item of clustering quality into the objective function of IB.
4. Instead of using a manually defined threshold, for the evaluation of the differences between a trajectory being tested and the medoids of the learned clusters, we employ an adaptive decision, by deploying the Shannon entropy concept [17], to look at all the differences under consideration as a whole and to make the trajectory analysis more discriminative.

The rest of this paper is organized as follows: Section 2 covers the related work. The core components of our approach, namely KDE modeling, MI based clustering and Shannon entropy based detection are depicted in Sections 3–5, respectively. Section 6 presents and thoroughly discusses the experimental results on real-world and synthetic testing data. Finally, concluding remarks and future work are given in Section 7.

2. Related Work

In general, modeling is performed as the first step to deal with trajectory data. Due to the noise in trajectory data during their extraction procedure, it is reasonable to model the motion trajectories in a probabilistic way [7,8,12]. Basically, the angle attribute is used to represent the shape of a trajectory and, the shape modeling is done in a probabilistic way by using some classic parameter function, such as von Mises [7,8]. That is, for each trajectory, its corresponding sequence of angles are represented by a linear combination of weighted parameter functions. The expectation-maximization (EM) method [18] is employed to estimate the parameters and weights needed. In this way, the so-called Mixture of von Mises (MovM) [7,8] is obtained for the model of trajectory shape. Similarly, both the speed and angle of a trajectory have been used based on another parameter function, Approximated Wrapped and Linear Gaussian (AWLG), to get the Mixture of Approximated Wrapped and Linear Gaussian (MoAWLG) for shape modeling [12]. However, the problem here is that the parametric

functions cannot fit the complex real-world trajectory data well. It is worthwhile to indicate that, some other relevant attributes, including spatial position, acceleration, size of the moving object and the curvature have been considered to model a trajectory for analysis and anomaly detection to meet the application requirements [19,20]. However, multiple features are not treated simultaneously [19]. Alternatively, multifeatures are processed based on the fuzzy membership of a trajectory belonging to a cluster, then a merging has to be used to obtain final crisp clusters; and this complicated procedure may result in systematic errors [20]. Also interestingly, the idea of KDE is applied in the context of clustering for multifeature video trajectories [20], but KDE is not for the modeling of a trajectory. Note that, some dimension reduction techniques, such as the Piecewise Linear Segmentation (PLS) [21,22], have been used to preprocess the motion trajectories for relieving the computational burden of trajectory analysis. PLS removes redundant sample points based on a predefined distance threshold to compressedly represent a trajectory based on a few salient points. In [11], Sieranoja et al. use the trajectory data collected from Global Position System (GPS) based on the speed and angle attributes. The speed and angle values of each trajectory are respectively transformed into discrete Fourier transform (DFT) coefficients. Then, for the sake of biometric identification a GMM classifier is trained using both the coefficient and the corresponding user label. For the purpose of dimensionality reduction, discrete cosine transform (DCT) is carried on the coefficients without the phase part, and only the first several DCT coefficients are used for the representation. DFT has been commonly used in trajectory representation [23], while the simplified trajectory representation may lose useful fine detail features needed for further processing.

For the clustering on trajectory shape descriptors, k-medoids algorithm has been proved to be satisfactory [7,8,12]. Based on the similarities between each two trajectory shapes, trajectories are divided into non-overlapping groups. The medoid trajectory of a group is chosen minimizing the distances between it and all the other trajectories in this group. The algorithm runs iteratively until all the medoids are converged. Actually, numerous clustering methods for trajectory data have been developed so far; for the purpose of brevity, the interested readers are directed to the two surveys for the detailed review [4,24]. Basically, clustering is an unsupervised technique to learn patterns from datasets and has become the most widely used nowadays because it does not need any prior knowledge, and this is important in many real scenarios [4]. Most of these relevant clustering algorithms either require an explicit measure of distance between trajectories or need a predetermined number of clusters, or even both; but usually, either a good definition of the distance measure, or a predetermination of an optimum number of clusters, is not easy to obtain. In [9], a similarity measure, derived from Jaccard index, is proposed for GPS trajectories. The trajectory data is recorded with cell codes, it is straightforward to obtain the distance by computing the matched parts from two trajectories being processed. This metric implicitly includes both angle and speed attributes of trajectories, also is usable for complex trajectory data. It is worthy to point out that, Normalized Compression Distance (NCD) [25], which is an universal distance metric between strings based on the notion of Kolmogorov complexity, could be used for the distance between trajectories. Unfortunately, NCD has to use a given compressor in its computation because the Kolmogorov complexity is not computable. Additionally trajectories are complex in the sense that these data include many attributes. So, in reality, it is not clear to employ which kind of compressor suitable for the complex trajectory data [26,27]. We have made use of the IB principle [14] to perform the clustering on trajectory data, avoiding the use of an explicit distance measure between two trajectories, and an adaptive number of the trajectory clusters are achieved based on the minimization of the loss of Mutual Information (MI) [17] between the trajectory dataset and a feature set of these trajectories [15]. Nevertheless, the practical implementation of IB has to use an iterative hierarchical merging of two candidate clusters, which may result in the local optima problem in some iterations. Actually information theoretic clustering plays an important role for data clustering, because this kind of approach does not need to make any assumed distribution on the data to be clustered. Recently, Steeg et al. [28] present an impressive work for data clustering by the minimization of an objective function defined

with the conditional entropy, which is obtained by using a non-parametric estimator based on the idea of k-nearest neighbor, showing very promising results.

Considering the abnormal detection based on trajectory shape analysis, a general way is based on a predefined threshold for deciding whether the trajectory being tested is anomalous or not [7,8,12]. However, apparently, a widely usable threshold dealing with different kinds of trajectory data is difficult to obtain. Notably, a very recent state-of-the-art algorithm of general trajectory-based anomaly detection, Sequential Hausdorff Nearest-Neighbor Conformal Anomaly Detector (SHNN-CAD) [29], has been presented showing excellent results. A trajectory is determined as normal or not depending on the conformity between it and all the N trajectories in normal dataset. Given a testing trajectory under consideration C, its k nearest neighboring trajectories, in terms of the Hausdorff distance, are located to calculate a sum of k distances, D_C. Analogously, for each normal trajectory T, the corresponding sum of k distances, D_T can be computed. Then the number of normal trajectories with $D_T > D_C$, n, is obtained. $\frac{n}{N}$ is the conformity of C, if this conformity is smaller than a pre-determined threshold then C is abnormal. Obviously, good k and threshold are difficult to be set for the different datasets with varied complexities. The idea of anomaly detection for trajectory data has been applied for various purposes. For instance, Zhang et al. [30] propose the so-called IBAT (Isolation Based Anomalous Trajectory detection) to identify the anomalousness of GPS trajectories. iBAT uses a group of historical trajectories has been labeled as normal to discriminate whether a test trajectory is abnormal. A random sample point is picked from the test trajectory and the trajectories of this group are deleted if they do not involve this point. This picking and deleting procedure repeats until there does not exist trajectory left, and in this case the test trajectory is abnormal.

3. Adaptive Trajectory Modeling Based on Nonparametric KDE

This section interprets how to model trajectory data by utilizing KDE and illustrates its performance by illustrations, then a technique for an adaptive setting of probability bins is proposed.

3.1. Trajectory Modeling on Speed and Angle

We argue, as has been pointed out in Section 1, that the angle and speed are two major factors for mathematically representing the trajectory: the former is to globally characterize the motion trajectory and the latter is to locally describe the moving object. Considering the fact that the trajectory data are usually uncertain and noisy in real-world contexts, we model the trajectories statistically, as usually done in the field of anomaly detection [6]. That is, we believe that normal trajectory instances happen with high probabilities while anomalies occur with small probabilities. In this paper, we utilize KDE, a non-parametric data modeling technique, by using the speed and angle values of the trajectory sample points to establish the PDF, for the shape distribution model of this trajectory. In this way, our model representation describes the trajectory data very well, and we are going to demonstrate the advantage of deploying KDE for modeling the trajectories in Section 3.2. Notice that, the combination of speed and angle for shape modeling based on exploiting the non-parametric and general KDE is new, though these two attributes and KDE have been used for trajectory data.

Let $x = \{s_1, s_2, \ldots, s_n\}$ be a trajectory with n sample points, where $s_i = (v_i, \theta_i)^T$ is a sample point presented by a pair of speed v_i and angle θ_i. Here v_i relies on the distance and time interval between this sample point and its previous adjacent one, and θ_i is determined by the tangential direction at this sample point. The speed for a sample point is discretized against the largest speed of all the sample points of the trajectories in the dataset, in this case the speed value used is discrete. All s_i for a trajectory are considered as the probabilistic observations taken from a single continuous PDF $z(y)$ for a generic continuous variable $y = (v, \theta)^T$, here v and θ respectively represent continuous variables for speed and angle values happening to this trajectory, and y denotes the feature value (vector) taken by a general single point on this trajectory. It is noted that a PDF of a trajectory is a continuous function, representing a general shape feature distribution for this trajectory. The PDF is estimated according to the multivariate kernel density estimator [31]

$$z(y) = n^{-1} \sum_{i=1}^{n} \left(|\mathbf{H}|^{-1/2} K\left(\mathbf{H}^{-1/2} (y - s_i) \right) \right), \tag{1}$$

and $\mathbf{H}$ is a 2×2 diagonal bandwidth matrix and is set as $\begin{pmatrix} \hat{\sigma_v}^2 n^{-2/5} & 0 \\ 0 & \hat{\sigma_\theta}^2 n^{-2/5} \end{pmatrix}$, where $\hat{\sigma_v}^2$ and $\hat{\sigma_\theta}^2$ are respectively the sample variances for $\{v_i\}$ and $\{\theta_i\}$. Here K is a kernel function, which is the widely used two dimensional Gaussian function [31,32]:

$$K(y) = \frac{1}{2\pi} exp\left(-\frac{1}{2} y^T y \right). \tag{2}$$

Finally, the PDF is obtained as

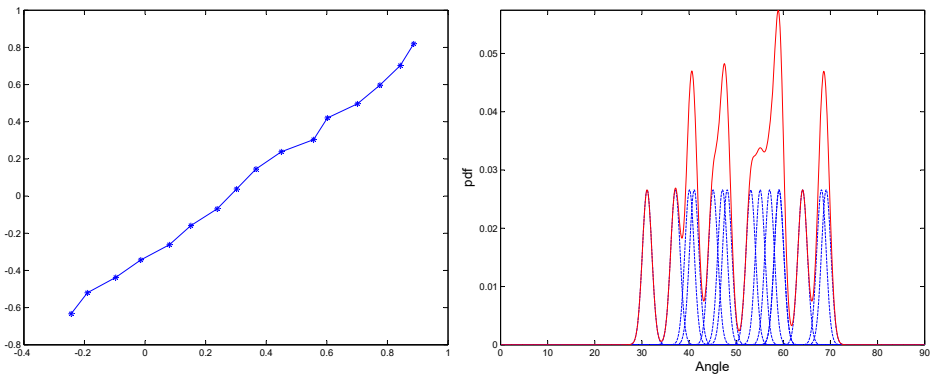
$$z(y) = \frac{1}{2\pi n^{3/5} \hat{\sigma_v} \hat{\sigma_\theta}} \sum_{i=1}^{n} exp\left(-\frac{n^{2/5}}{2} \left(\frac{(v - v_i)^2}{\hat{\sigma_v}^2} + \frac{(\theta - \theta_i)^2}{\hat{\sigma_\theta}^2} \right) \right). \tag{3}$$

Obviously, the PDF value at any point y on a trajectory is a weighted combination of the n values for Gaussian kernels centered on n observation sample points s_i. Notice, the logic underlying here is that the estimated probability density should have a large value at a point with many observations in its neighborhood, and vice versa [31,32].

For simple but clear illustration, we show an example of one dimensional (1D) PDF based on one dimensional KDE [32]

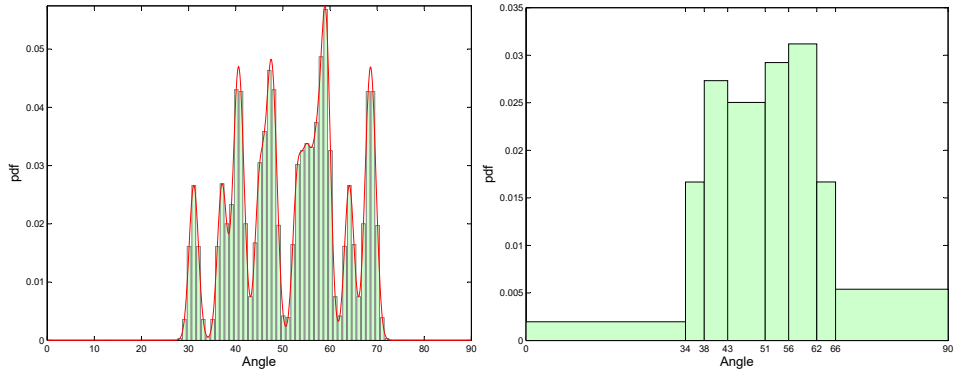
$$\begin{aligned} f(\theta) &= \frac{1}{nh} \sum_{i=1}^{n} k\left(\frac{\theta - \theta_i}{h} \right) \\ &= \frac{1}{nh\sqrt{2\pi}} \sum_{i=1}^{n} exp\left(-\frac{1}{2} \left(\frac{\theta - \theta_i}{h} \right)^2 \right), \end{aligned} \tag{4}$$

using angle attributes of a trajectory with 16 sample points (Figure 1a) in Figure 1b. Here k is the one dimensional Gaussian $k(u) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{1}{2} u^2 \right)$, and an optimal bandwidth is $h = 1.06 \hat{\sigma_\theta}^2 n^{-1/5}$ [32]. The continuous plot (red color) presents the continuous PDF for the trajectory, which is obtained by a weighted average of 15 Gaussian kernels (blue color). Note that each angle sample is calculated based on two consecutive sample points of the trajectory.



(**a**) (**b**)

Figure 1. *Cont.*

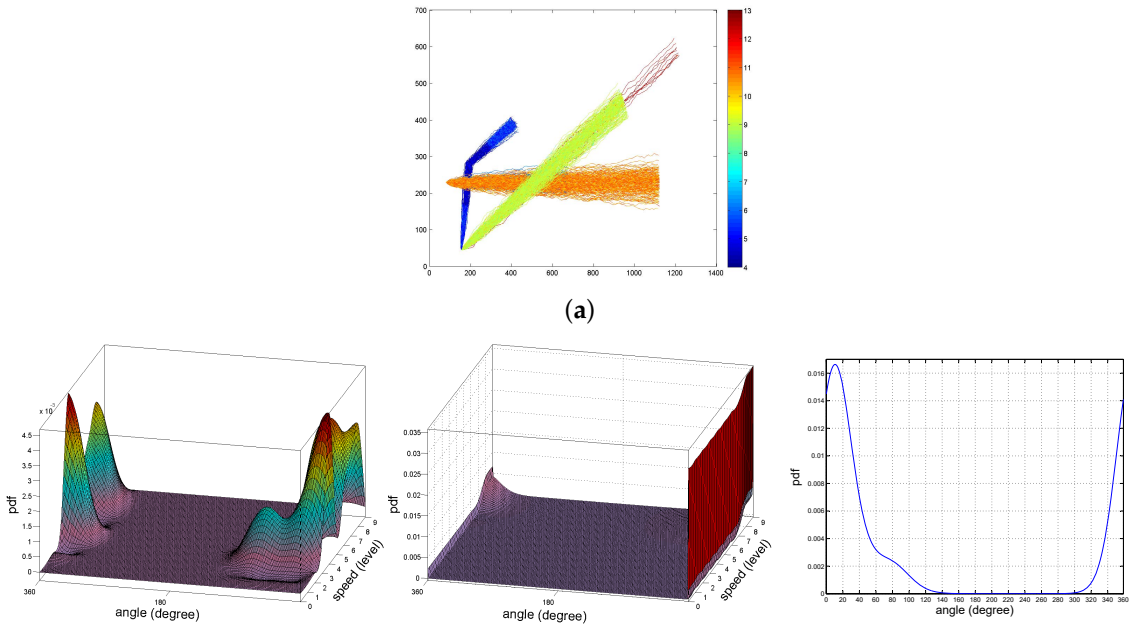


(**c**) (**d**)

Figure 1. The three kinds of probability distribution function (PDF) representations for a sample trajectory. For clearness, the domain for PDF is partly shown. (**a**) A sample trajectory; (**b**) Continuous PDF by KDE; (**c**) Highly approximated PDF; (**d**) Adaptive PDF Bins.

3.2. Illustration Comparing the Estimated PDFs by Different Modeling Schemes

We compare KDE with two commonly used parametric modeling approaches, von Mises [7,8] and AWLG [12]. A set of simulated object trajectories are designed to exemplify the different capabilities of KDE, MoAWLG and MovM for trajectory modeling. This dataset has 790 trajectories and each of them has 100 sample points. All the angles at the trajectory sample points are between 0° and 359°, and most of them are around 0°, 359°, 40° and 80°. Here the speed levels of this trajectory in all are 0–9, the speeds of the moving object vary around levels 4, 5, 7 and 8, as shown in four different colors in Figure 2a. In fact, each trajectory has a same number of sample points taken in a same duration. The different lengths for different trajectories in different colors indicate the different speed levels they have, leading to the different behavior of these trajectories.



(**a**)

(**b**) (**c**) (**d**)

Figure 2. Seven hundred and ninety trajectories and the plots of PDFs by three modeling schemes. The trajectories are visualized by different colors according to the speed values. (**a**) 790 trajectories; (**b**) Kernel Density Estimation (KDE); (**c**) Mixture of Approximated Wrapped and Linear Gaussian (MoAWLG); (**d**) Mixture of von Mises (MovM).

Figure 2 also gives the PDFs estimated by KDE, MoAWLG and MovM. Here for better illustration, all the 79,000 sample points for all the 790 trajectories in this specially designed dataset are used together for obtaining the PDFs. Obviously, the PDF obtained by KDE closely reflects the distribution characteristics of the speeds and angles of the trajectory sample points. By contrast, the PDF evaluated by MoAWLG cannot faithfully represent the speed characteristic of the designed trajectory. For example, two peaks occur on the PDF by KDE for the speed values around levels 4 and 5 and for the angles around 359° but, the PDF by MoAWLG for these speed and angle attributes cannot achieve this. The reason here is that KDE takes the nonparametric statistical way to better model a motion trajectory, MoAWLG has to assume that AWLG fits the true PDF, but actually this fitting is not usually true. As for MovM, it just focuses on a single univariate distribution, namely the angle distribution, of the trajectory sample points. Based on using MovM, we can basically obtain the angle distribution for the trajectory sample points, as shown in Figure 2d.

3.3. Adaptive Probability Bins

In practical implementation, a discrete probability distribution based on the estimated PDF by KDE is used for the later trajectory processing. Basically, a number of equally sized probability bins are taken to approximate a PDF. Here the height of a bin is obtained directly from the ordinate of the continuous PDF by using Equation (3) at the point of the bin edge, and it is worthwhile to know that the use of probability bins here is different from the probabilistic modeling by statistical histogram [31,32]. Importantly, an adaptive positioning of the bin edges, together with an adaptive number of probability bins, can be exploited to highlight "the principal components" of the original PDF to present a trajectory.

Given a trajectory dataset, a global PDF of the speed and angle values for all the sample points of all the trajectories fundamentally characterize this dataset. A large enough number of equally sized probability bins can be considered to closely approximate the global PDF. For example, 360 equally sized intervals are used for speed and also for angle levels, and in this case the two dimensional (2D) 360×360 equally sized probability bins are the highly approximated discrete probability distribution for this dataset. The position of an adaptive probability bin can be approximately obtained by locating a maximum (either global or local) and its neighboring minima (either global or local) on this 360×360 bins, then the domain values corresponding to the minima are set as the bin edges. The number of bins can also be determined adaptively. For easier and better illustration, Figure 3a,b respectively present the 360 equally sized 1D probability bins (here, the continuous PDF plot is used to have a better display), as the high-approximation, for the angle and speed values of a real aircraft trajectory dataset (see Section 6.1). Figure 3a shows 6 angle values, 34, 90, 133, 175, 228 and 274, corresponding to the minima on the high-approximation, and these angle values are taken as the edges of the adaptive bins. As a result, in all seven 1D adaptive probability bins are finally positioned for the high-approximation discrete probability distribution of angle values. Obviously, adaptive probability bins emphasize the high values of high-approximation discrete probability distribution, resulting in a condensed representation of the trajectory. Undoubtedly, the number of adaptively positioned bins, needed for modeling a trajectory, is smaller than that of the equally sized probability bins. This benefits the efficiency of trajectory computing. Analogously, Figure 3b gives two 1D adaptive bins with edges at the level 114 for the speed distribution. Finally, we use the 1D bin edges of angle and speed to effectively position the adaptive probability bins of joint angle and speed, as shown in Figure 3c. In practice, when the number and positions of adaptive bins suitable for all the trajectories in a dataset are obtained, and then these number and positions are used for every trajectory.

For any trajectory x_i in a given trajectory dataset $X = \{x_i\}\,(i = 1, 2, \ldots, m)$, a set of 2D bins located at a set of adaptive positions $Y = \{y_j\}\,(j = 1, 2, \ldots, n)$ can be obtained as the approximation of PDF given x_i. Obviously each y_j corresponds to a certain value (vector) of shape feature happening to trajectories in X. For the purpose of this paper, y_j is called feature point, and Y is called the feature set of these trajectories. The height of the bin located at y_j is actually the conditional probability

$p\left(y_j|x_i\right)$. Thus $p\left(Y|x_i\right)$ is a discrete probability distribution of shape feature of x_i, namely the discrete version of PDF of shape feature given x_i.

As an illustration example, Figure 2c, based on the green bins, shows a high approximation of the continuous PDF by KDE for the example trajectory, here 360 equally sized intervals is employed for angle values. Figure 2d gives 8 adaptive bins.

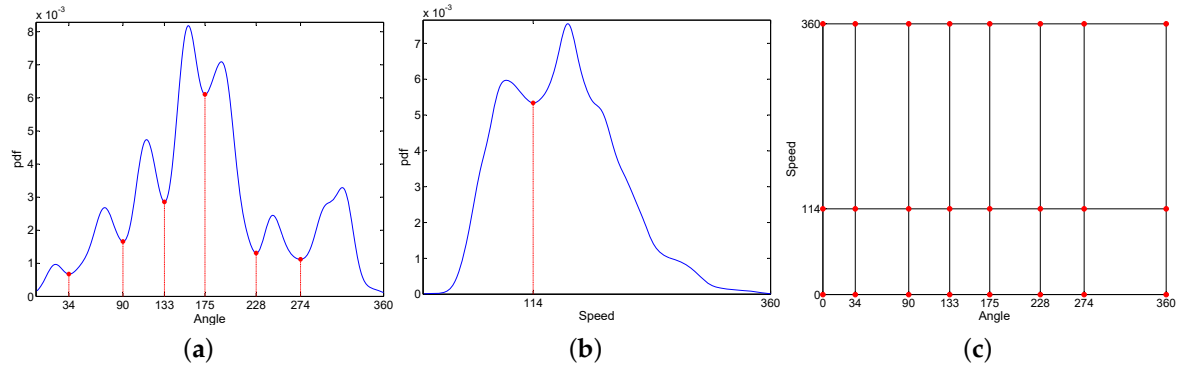


Figure 3. (**a**,**b**) PDF with adaptive probability bins. The angle and speed values corresponding to global/local minima on PDF are with red color, and these values are taken as the edges of adaptive bins. (**c**) Each small rectangle positions an adaptive bin of the joint distribution of angle and speed attributes. (**a**) PDF of angle distribution; (**b**) PDF of speed distribution; (**c**) adaptive probability bins.

Table 1 demonstrates an example of the advantage of using the adaptive probability bins. MI based trajectory clustering is one of the purposes for this paper, and the JSD computation for two PDFs is a fundamental step included in the clustering. Quite a lot of JSD computations are needed, which plays an important role for the clustering efficiency. In this table, the smallest numbers of bins, which are necessary to obtain the correct clustering results for test trajectory datasets (see Section 6), respectively for adaptively and equally sized probability models, are listed. The corresponding average runtime results (in seconds), needed for each JSD computation are also given. Clearly the computing efficiency is improved with the use of adaptive bins, and approximately at least 30% time saving can be achieved. Consequently, the proposed adaptive probability bins are effective and efficient for the trajectory modeling.

Table 1. Probability bins: Adaptive versus Equal.

	Adaptive		**Equal**	
	Number of Bins	**Runtime**	**Number of Bins**	**Runtime**
Aircraft	14	0.000012	21	0.000017
MIT	40	0.000026	50	0.000036
Synthetic	4	0.000004	9	0.000009

4. MI Based Trajectory Clustering

This section gives the trajectory clustering based on IB principle [14], in addition we further improve the performance of clustering by introducing an item on the evaluation of clustering quality into the objective function of IB.

4.1. Basic Concepts of Information Theory

Before discussing the IB principle, we firstly introduce three necessary basic concepts of information theory.

The Shannon entropy of a probability distribution Q_1 is defined as [33]

$$H(Q_1) = \sum_{q_i \in Q_1} q_i \log q_i \tag{5}$$

The Jensen-Shannon divergence (JSD) between two probability distributions Q_1 and Q_2 with respective positive weights π_1 and π_2 ($\pi_1 + \pi_2 = 1$) is defined by [33]

$$JSD_{\pi_1,\pi_2}(Q_1, Q_2) = H\left(\sum_{i=1,2} \pi_i Q_i\right) - \sum_{i=1,2} \pi_i H(Q_i) \tag{6}$$

The Mutual Information (MI) between two probability distributions Q_1 and Q_2 gives the meaningful information shared by Q_1 and Q_2. The Mutual Information between Q_1 and Q_2 is defined by [17].

$$\begin{aligned} I(Q_1; Q_2) &= \sum_{p_i \in Q_1} \sum_{q_j \in Q_2} p(p_i, q_j) \log \frac{p(p_i, q_j)}{p(p_i) p(q_j)} \\ &= \sum_{p_i \in Q_1} \sum_{q_j \in Q_2} p(q_j|p_i)\, p(p_i) \log \frac{p(q_j|p_i)}{p(q_j)} \end{aligned} \tag{7}$$

4.2. IB Principle for Trajectory Data

Here the IB principle [14], which gives a suitable number of clusters for a trajectory dataset is presented.

As have been described in Section 3.3, given two random variables X and Y, $X = \{x_i\}$ $(i = 1, 2, \ldots, m)$ is a trajectory dataset and $Y = \{y_j\}$ $(j = 1, 2, \ldots, n)$ is the corresponding feature set, and an information channel between X and Y is shown as follows:

1. Input probability $p(x_i)$. This is the probability of a single trajectory. We simply use $p(x_i) = \frac{1}{n_T}$ to assign the uniform "importance" for all the trajectories under consideration, here $n_T = |X|$.
2. Conditional probability $p(y_j|x_i)$. This probability of the feature point y_j given the trajectory x_i can be obtained from the discrete probability distribution of shape feature of x_i, $p(Y|x_i)$. That is, this probability is the height of the bin located at y_j.
3. Output probability $p(y_j)$. This is the probability of the feature point y_j, given all the trajectories in the dataset. This can be obtained by the full probability formula

$$p(y_j) = \sum_{i=1}^{m} p(x_i)\, p(y_j|x_i). \tag{8}$$

Clustering based on IB principle obtains the most possible compacted X, $\tilde{X} = \{\tilde{x}_k\}$ $(k = 1, 2, \ldots, t)$, which is actually a set of trajectory clusters, by minimizing $I(\tilde{X}; X)$ and meanwhile preserving as much as possible the relevant information about Y, provided by $I(\tilde{X}; Y)$ [14]. Note that the proper and adaptive number of clusters, t, can be directly determined in the procedure of clustering. It is widely accepted that $\tilde{X}$ is found by using Lagrange multiplication method through minimizing the following functional [14]

$$\{I(\tilde{X}; X) - \beta I(\tilde{X}; Y)\}, \tag{9}$$

where the parameter β is a positive Lagrange multiplier, providing a trade-off between compacting data and preserving relevant information [16]. Equivalently, $\tilde{X}$ can be given by maximizing the functional

$$L_{max} = \left\{ I(\tilde{X}; Y) - \frac{1}{\beta} I(\tilde{X}; X) \right\}, \tag{10}$$

which is yielded by dividing Equation (9) with $-\beta$.

There exists an important extreme value for the Lagrange multiplier β. That is, in the limit $\beta \to \infty$, it is obviously observed from Equation (10) that the compacted clusters $\tilde{X}$ is obtained by maximizing the functional

$$L_{max} = \left\{ I\left(\tilde{X};Y\right) \right\}, \tag{11}$$

and namely, by only preserving relevant information about Y. Interestingly enough, in practice, a direct and intuitive way for getting a compacted set of clusters on X, $\tilde{X}$, is to merge elements in X [16]. Considering that $\tilde{X}$ is actually a lossy X, the MI between $\tilde{X}$ and Y, $I\left(\tilde{X};Y\right)$, is upper bounded by $I\left(X;Y\right)$ [34]. In this case, a hierarchical clustering can be organized as a bottom-up and iterative merging to generate the final clusters. In the initialization of iterative merging, X is copied as the initial $\tilde{X}\left(\tilde{X}_0\right)$, and this means that each trajectory $x_i \in X$ corresponds to a single cluster, as an element in $\tilde{X}_0$, $\tilde{X}_0 = \{\{x_i\}\}$. In each iteration, two elements of the current round $\tilde{X}\left(\tilde{X}_r\right)$ are merged to produce the next round $\tilde{X}\left(\tilde{X}_{r+1}\right)$, satisfying the minimization of the loss of MI, $\Delta L_{max} = L^b_{max} - L^a_{max}$, where $L^b_{max} = I\left(\tilde{X}_r;Y\right)$ and $L^a_{max} = I\left(\tilde{X}_{r+1};Y\right)$ respectively denote the MI before and after the merging. Obviously $I\left(\tilde{X}_r;Y\right) \geq I\left(\tilde{X}_{r+1};Y\right)$ holds because $\tilde{X}_{r+1}$ is a lossy version of $\tilde{X}_r$, and this essentially indicates that the bottom-up merging achieves the maximization of Equation (11) in an iterative manner. It is noted that the hierarchical clustering described just now is called agglomerative Information Bottleneck (aIB) [34]. In fact, aIB is simple but effective and, is very largely used in practice [16]. As a result, in this paper, we use the Lagrange multiplier $\beta \to \infty$ and exploit aIB for trajectory clustering.

In the following, we present the calculation details in aIB. Suppose in a certain iteration, $\tilde{x}_1$ and $\tilde{x}_2$ are the two candidate clusters to be merged, and $\tilde{x}^*$ is a new cluster obtained by this merging. The corresponding MI loss caused is [34]

$$\begin{aligned}
\Delta L_{max}\left(\tilde{x}_1,\tilde{x}_2\right) &= I\left(\tilde{X}^b;Y\right) - I\left(\tilde{X}^a;Y\right) \\
&= \sum_{j=1}^{n}\sum_{i=1}^{2} p\left(\tilde{x}_i,y_j\right)\log\frac{p\left(\tilde{x}_i,y_j\right)}{p\left(\tilde{x}_i\right)p\left(y_j\right)} - \sum_{j=1}^{n} p\left(\tilde{x}^*,y_j\right)\log\frac{p\left(\tilde{x}^*,y_j\right)}{p\left(\tilde{x}^*\right)p\left(y_j\right)} \\
&= p\left(\tilde{x}^*\right)\left(H\left(\sum_{i=1}^{2}\frac{p\left(\tilde{x}_i\right)}{p\left(\tilde{x}^*\right)}p\left(Y|\tilde{x}_i\right)\right) - \sum_{i=1}^{2}\frac{p\left(\tilde{x}_i\right)}{p\left(\tilde{x}^*\right)}H\left(p\left(Y|\tilde{x}_i\right)\right)\right) \\
&= p\left(\tilde{x}^*\right) JSD\left(\frac{p\left(\tilde{x}_1\right)}{p\left(\tilde{x}^*\right)},\frac{p\left(\tilde{x}_2\right)}{p\left(\tilde{x}^*\right)};p\left(Y|\tilde{x}_1\right),p\left(Y|\tilde{x}_2\right)\right),
\end{aligned} \tag{12}$$

where $\tilde{X}^b$ and $\tilde{X}^a$ in fact denote the random variables respectively corresponding to $\tilde{X}$ before and after the merge of $\tilde{x}_1$ and $\tilde{x}_2$. The merge of two candidate clusters is done according to minimizing the loss of MI induced by this merging, and obviously aIB only achieves a local optimum for each iteration [16]. When the merging is done, the conditional probability is updated by [16]

$$p\left(y_j|\tilde{x}^*\right) = \frac{p\left(\tilde{x}_1\right)}{p\left(\tilde{x}^*\right)}p\left(y_j|\tilde{x}_1\right) + \frac{p\left(\tilde{x}_2\right)}{p\left(\tilde{x}^*\right)}p\left(y_j|\tilde{x}_2\right) \tag{13}$$

where

$$p\left(\tilde{x}^*\right) = p\left(\tilde{x}_1\right) + p\left(\tilde{x}_2\right). \tag{14}$$

At the beginning of the clustering algorithm, $p\left(\tilde{x}_1\right) = 1/n_T$ and $p\left(\tilde{x}_2\right) = 1/n_T$ because $\tilde{x}_1$ and $\tilde{x}_2$ correspond to two single trajectories in X.

We take advantage of the MI ratio $\delta_{MI} = \frac{I\left(\tilde{X};Y\right)}{I\left(X;Y\right)}$ to terminate the iterative process in this paper. By $\delta_{MI} = 95\%$, the IB scheme behaves very well in trajectory clustering. In this case, the proper number of clusters can be adaptively and directly determined in the procedure of clustering.

For the sake of trajectory anomaly detection (see Section 5), the medoid trajectory for each cluster is determined as one of all the trajectories in this group to minimize the sum of distances between it and all the others.

4.3. New Objective Function

Given three clusters $\tilde{x}_1, \tilde{x}_2$ and $\tilde{x}_3$, assuming that the trajectories in $\tilde{x}_1$ and $\tilde{x}_2$ are similar, but different with those in $\tilde{x}_3$, then there will be unacceptable grouping if $\Delta L_{max}(\tilde{x}_1, \tilde{x}_3)$ or $\Delta L_{max}(\tilde{x}_2, \tilde{x}_3)$ is the minimum value. For example, Figure 4 shows three clusters visualized in red ($\tilde{x}_1$), green ($\tilde{x}_2$) and blue ($\tilde{x}_3$) in the 312-th iteration of clustering on aircraft trajectories (see Section 6.1). The trajectories in $\tilde{x}_1$ and $\tilde{x}_2$ are similar in angle and speed values, but very different from those in $\tilde{x}_3$ with respect to angle attribute. In fact, $\tilde{x}_1$ and $\tilde{x}_2$ present same airplane operations, landing on runway 27R, while $\tilde{x}_3$ include trajectories taking off from runway 33. Obviously, $\tilde{x}_1$ and $\tilde{x}_2$ should be grouped together, however, $\tilde{x}_1$ and $\tilde{x}_3$ are merged by IB since the merging induces the smallest loss of Mutual Information ($\Delta L_{max}(\tilde{x}_1, \tilde{x}_2) = 0.00871, \Delta L_{max}(\tilde{x}_1, \tilde{x}_3) = 0.00864, \Delta L_{max}(\tilde{x}_2, \tilde{x}_3) = 0.00911$). The unsatisfying clustering is due to that the greedy procedure of IB achieves the local optimum at each iteration.

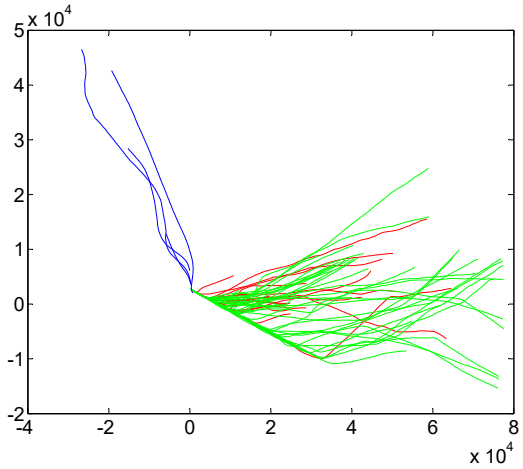


Figure 4. Three clusters in aircraft trajectory dataset. The numbers of trajectories in red, green and blue clusters are 16, 38 and 4, respectively.

In this paper, to alleviate the optimum problem of IB for improving its performance, we propose to add the clustering quality to the objective function of IB as follows:

$$\Delta = \Delta L_{max}\left(\tilde{x}_1, \tilde{x}_2\right) + \lambda \psi\left(\tilde{x}^*\right) \tag{15}$$

where $\psi\left(\tilde{x}^*\right)$ measures the intra-class similarity of the new cluster by merging $\tilde{x}_1$ and $\tilde{x}_2$, and we compute this similarity by

$$\psi\left(\tilde{x}^*\right) = \max_{x_i, x_j \in \tilde{x}^*} JSD\left(x_i, x_j\right) \tag{16}$$

where $JSD\left(x_i, x_j\right)$ is the Jensen-Shannon divergence between the PDFs of trajectories x_i and x_j. The unit of $\psi\left(\tilde{x}^*\right)$ is bit, which is the same as ΔL_{max}. The parameter λ controls a trade-off between loss of mutual information and clustering quality, we have observed by experimentation that $\lambda = \frac{1}{N}$ works well, here N is the number of trajectories in the dataset. Considering that λ is determined in a heuristic way, actually, this λ could be more adaptive to data, and will be included in our future work.

5. Shannon Entropy Based Trajectory Anomaly Detection

In previous work, if the differences between a testing trajectory and the cluster medoids are higher than a "hard" threshold, then this trajectory is classified as abnormal; otherwise the testing trajectory belongs to a labeled group, which is possibly an anomalous cluster [7,8]. However, in many and varied practical situations, it is infeasible to set the direct and "hard" threshold for the possible quite large ranges of trajectory differences. As a matter of fact, the key observation for an abnormal trajectory is that, among all the differences between this trajectory and all the cluster medoids, there is no one being significantly larger than the others. That is, the differences between the abnormal trajectory and the cluster medoids can be regarded as "approximately equal". In this paper, the PDFs of each testing trajectory and the cluster medoids are used to build a probability distribution by calculating the distances between the trajectory to be tested and those of the medoids. Then the Shannon entropy [17] of this probability distribution is exploited to fulfill the task of anomaly detection. We make use of the Shannon entropy to estimate the homogeneity of all the differences between the considered trajectory and the medoids of the clusters learned. In this case, we make an adaptive situation for all the differences under consideration: this is opposed to the approach proposed previously [7,8], in which these differences themselves are just separately evaluated as whether large or small, by using some straightforward threshold. Consequently, our anomaly detection is more discriminating.

Suppose we have a testing trajectory x_{test} and a set of the learned clusters represented with the trajectory medoids $\{m_i\}$ $(i = 1, 2, \ldots, t)$, here m_i is the medoid of the i-th cluster. First, we obtain the distances, $\{d_j\}$ $(j = 1, 2, \ldots, t)$, between x_{test} and $\{m_i\}$ by Bhattacharyya coefficient [35]. Further, a probability distribution, P, can be obtained as

$$p_i = \frac{d_i}{\sum_{j=1}^{t} d_j}, i = 1, \ldots, t \tag{17}$$

and then the Shannon entropy $H(P)$ of this distribution is computed by Equation (5). Basically, $H(P)$ measures the information used to identify whether the testing trajectory x_{test} is normal or not. $H(P)$ achieves the maximum $\log(t)$ when all the p_i are equal to $1/t$. With $H(P)$ becoming larger, it is more uncertain to determine that x_{test} belongs to a certain cluster, thus, x_{test} can more probably be deemed as abnormal. In practice, an anomaly is reported if there is no specific distance d_i $(i \in \{1, \ldots, t\})$ that is apparently bigger than the other d_j $(j \neq i, j \in \{1, \ldots, t\})$. The point here is to utilize the critical value of $H(P)$ for differentiating between normal and abnormal trajectories. For convenience, this critical value is denoted by H_c. If $H(P) \in (H_c, \log(t)]$ then x_{test} is identified as abnormal; otherwise x_{test} belongs to the cluster with the medoid m_{id} meeting $d_{id} = \min\{d_j\}$ $(j \in \{1, \ldots, t\})$. Notice that this cluster may be an anomaly having been labeled. In this paper, H_c corresponds to the Shannon entropy of a certain trajectory from the given dataset which has the most approximated distances to $\{m_i\}$ $(i = 1, 2, \ldots, t)$, and the three steps on how to obtain H_c are as follows. The first is to obtain t trajectories, $\left\{x^{(i)}\right\}$ $(i = 1, 2, \ldots, t)$, and $x^{(i)}$ is a trajectory from the i-th cluster and has the largest distance from m_i. Second, for each $x^{(i)}$, we calculate the distances to all the medoids and then compute the variance of these t distances. Finally, H_c takes the value of Shannon entropy of the trajectory with minimum variance.

As usually done, the number of trajectories in the anomalous cluster is set below a small percent of the total number of all the trajectories. Some dynamic change in the trajectory clusters may occur after the process of Shannon entropy based detection, and we call this the local update of the trajectory clusters. For practical use of our proposed approach to trajectory analysis based anomaly detection, all the clusters are globally updated by re-clustering all the trajectories periodically.

6. Experiments on Synthetic and Real Trajectory Data

Based on a collection of public real-world and synthetic data with different speed and angle attributes, extensive tests have been done to evaluate the performance of our proposed technique.

We compare the clustering performance of IB with and without adding the clustering quality to its objective function (for simplicity, we call IB with clustering quality as our proposed approach, and the other one as the original IB). In addition, based on the modeling representations resulted from using KDE, we utilize *k*-means, a typical and widely used clustering algorithm, to mine clusters, and the parameters employed are fine tuned to obtain the best possible output. In this paper, the distance between two PDFs applied in *k*-means is measured by Bhattacharyya coefficient [35]. All the experiments are carried on a Windows PC with Intel Core i7 3.40 GHZ CPU and 32 GB RAM. Further details on trajectory data, along with the experimental results, are presented in the following sections.

6.1. Aircraft Trajectory Dataset

We first make use of a real aircraft trajectory dataset [36] contributed by Gariel et al. [37] to evaluate the performance of clustering and anomaly detection. The training dataset consists of 320 aircraft trajectories, which have different numbers of sample points varying between 102 and 1023. Figure 5 displays this dataset and the clusters generated by different approaches.

In Figure 5d, based on the speed and angle attributes, our approach obtains 8 clusters that correctly divide the trajectory data with respect to the physical rationality in practical civil aviation scenario. Namely, considering a trajectory occurring at a certain runway with a certain operation (either taking off or landing), the operations of these clusters are (from left to right): landing on runway 27R, taking off from runway 28L, taking off from runway 28R, landing on runway 19L, landing on runway 11, taking off from runway 11, taking off from runway 01R and taking off from runway 33. The numbers of trajectories in these 8 clusters are 54, 43, 53, 59, 50, 44, 13 and 4, respectively.

In Figure 5e, the clustering result achieved by the original IB scheme is much inferior to that by our proposed approach, which takes the clustering quality into consideration. All the trajectories in the second to the seventh clusters are satisfactorily identified, while some problem occurs to the first and last clusters. That is, 16 trajectories landing on runway 27R are wrongly mixed with those taking off from runway 33. Obviously, these trajectories including two operations have rather different angles and speeds. The reason for the current issue is due to that the numbers of trajectories in these 2 clusters are quite uneven, which causes some degradation in the clustering performance of IB algorithm, as discussed in Section 4.3. In other words, our proposal of introducing new objective function for IB technique is pretty effective and helpful.

Figure 5f illustrates that the clusters learnt by the common *k*-means cannot distinctly indicate different behaviors of the aircrafts. For example, the trajectories (third in Figure 5f) unsatisfyingly mix the trajectories respectively in the two separate clusters (third and fifth in Figure 5d), of which the respective trajectories have small speed differences but large angle distinctions. In fact, the two clusters include taking off from runway 28R and landing on runway 11, respectively. The reason is twofold: first, the clustering performance of *k*-means highly relies on the distance measure between trajectories, while the distance estimation is very sensitive to the number of sample points of trajectories [38]. In this case, these two clusters have very different numbers of sample points of the trajectories, and the respective min, max and mean points of the trajectories are $[276, 357, 298]$ and $[612, 915, 725]$. Second, it is hard to find an optimal distance method coping with all kinds of trajectory datasets [39]. Similarly, the last cluster by *k*-means also led to incorrect clustering by mixing several trajectories from the first cluster, although they have large difference in angle attribute.

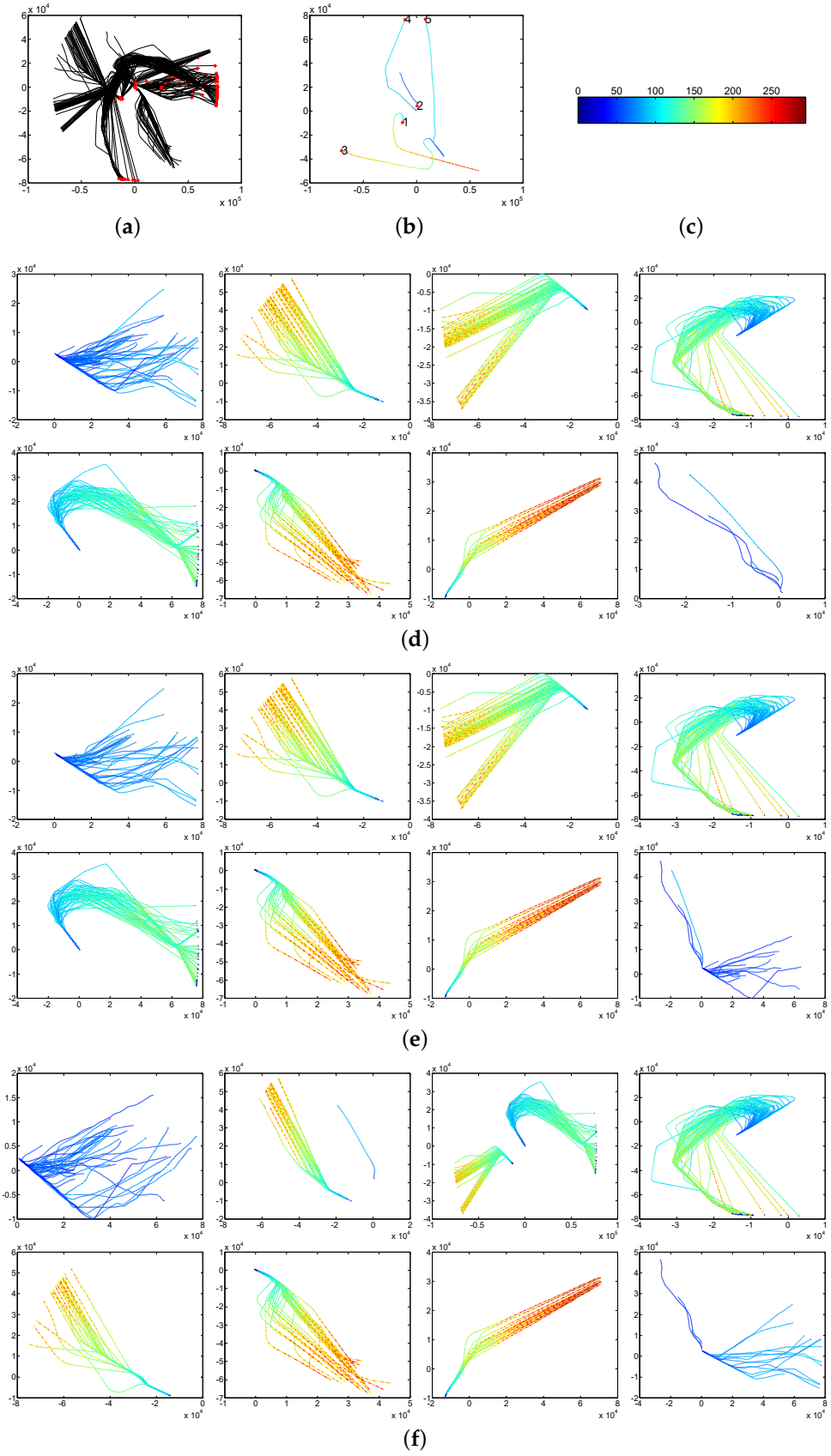


Figure 5. The aircraft trajectory dataset and comparison of clustering results. (**a**) A training set of aircraft trajectories with the beginning points highlighted in red color; (**b**) 5 numbered trajectories for anomaly test; (**c**) colorbar of speed values applied in visualizing testing trajectories and clusters; (**d**) clusters by our proposed approach; (**e**) clusters by original IB; (**f**) clusters by *k*-means.

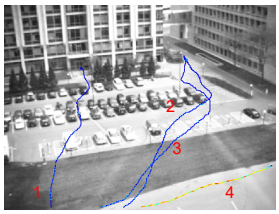
To evaluate the performance of anomaly detection, 5 trajectories are tested as shown in Figure 5b, in which the second one belongs to the corresponding cluster of taking off from runway 33 (the last cluster in Figure 5d), and all the others are outliers. All the testing trajectories are correctly recognized to be normal or abnormal by our Shannon entropy based anomaly detection, due to an adaptive differentiation between normal and abnormal trajectories. Besides, a recent and effective approach, Sequential Hausdorff Nearest-Neighbor Conformal Anomaly Detector (SHNN-CAD) [29], is implemented for comparison. SHNN-CAD detects whether a testing trajectory is abnormal or not based on a version of Hausdorff distance. Although being parameter-light, the number of nearest neighbors k still needs a good presetting for various kinds of trajectory datasets. In practice, it is not straightforward to determine the parameter without the background knowledge of trajectory data. In this experimentation, SHNN-CAD obtains 60% accuracy with incorrect identification of the last two trajectories as normal objects under different k values (k = 1–6, 10, 20, 30). Apparently, the computational cost grows fast with k increasing because of the distance measure between each two trajectories.

6.2. MIT Trajectory Dataset

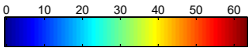
We choose 176 training trajectories with average 487 sample points in a parking lot [40] from the MIT dataset, provided by Wang et al. [41] for the clustering experiment. Figure 6a shows 4 anomaly trajectories used in the testing stage.

As shown in Figure 6c, the 8 clusters from our proposed approach clearly present different motion patterns of the moving objects. Apparently, the sixth and last clusters have very different angle attributes, since the trajectories in the sixth cluster move on a straight direction and then turn at about 45 degree angles, while the last cluster (very short trajectories between two buildings in the upper right area) has a single main direction. Unfortunately, the original IB mixes them together to the seventh cluster in Figure 6d. As a result, the trajectories, which are quite similar in speed and angle, are divided into the second and third clusters. For comparison, we give the numbers of trajectories in 8 clusters by the proposed approach and original IB, respectively, as follows: $(23, 25, 34, 23, 30, 21, 17, 3)$ and $(23, 11, 14, 34, 23, 30, 24, 17)$. The experimental results exhibit that our improvement on the IB objective function has the advantage in handling trajectories that rather unevenly belong to different clusters. Obviously, the result in Figure 6e given by the distance based k-means presents a messy clustering. For instance, the trajectories that have almost same speeds but largely different angles are grouped together in the fourth cluster (this problem also happens in the last cluster). On the other hand, similar trajectories are treated as discriminative, such as the first two clusters, and also the sixth and seventh clusters.

Both the proposed Shannon entropy based approach and the effective SHNN-CAD accurately recognize the 4 abnormal trajectories in Figure 6a. In addition, the processing time for anomaly detection is briefly reported here, considering that the trajectory learning is done in an off-line way. For a trajectory with an average of 634 sample points, the proposed technique costs 0.078 s, while SHNN-CAD takes 7.452 and 7.431 s when $k = 2$ and $k = 3$, respectively. The large difference of runtime shows our approach is more promising for applications which need fast response.



(**a**)



(**b**)

(**c**)

(**d**)

(**e**)

Figure 6. The MIT trajectory dataset and comparison of clustering results. (**a**) 4 numbered trajectories for anomaly test; (**b**) colorbar of speed values applied in visualizing testing trajectories and clusters; (**c**) clusters by our proposed approach; (**d**) clusters by original IB; (**e**) clusters by *k*-means.

6.3. Synthetic Trajectory Dataset

In this experiment, we investigate the clustering and anomaly detection performance of our proposed method on 100 synthetic trajectory datasets [42] (TS1-TS100) contributed by Piciarelli et al. [43]. Each dataset includes 250 normal and 10 abnormal trajectories with constant velocity, and the normal ones are designed to evenly group into 5 clusters. Considering that speed is an important attribute, we add different speed values to this dataset, as shown in Table 2. Especially, 5 anomalies have relatively the same speeds with these clusters. Additionally, random noise is involved in these trajectories to vividly simulate the cases in reality.

Table 2. Summary of speed attribute for the synthetic data.

	Minimum Speed	Maximum Speed	Accelerated Speed	Operation
clusters 1, 2	10	160	10	speed up
clusters 3, 4	200	350	−10	slow down
cluster 5	30	320	20	speed up

As discussed in Section 4.3, IB behaves unsatisfactorily because of the local optima problem, which could resulted from the outlier in the dataset, thus we take 250 normal trajectories and an anomaly as the training set. For the sake of objectively comparing different algorithms, the quantitative clustering qualities are given in this paper. In general, there are three types of measures on clustering quality, external criteria, internal criteria and relative criteria [44]. External criteria are based on the background knowledge (or groundtruth) of data. Internal criteria use the essential information included in the data. Relative criteria score by comparing results from different clustering algorithms. Considering that the external criteria are practically the most used, we make use of three popular external indexes, precision, recall and F-Measure [45] for clustering quality evaluation. Precision (P), recall (R) and F-Measure (F) focus on the testing accuracy based on the groundtruth. For each cluster, we calculate these three values by:

$$P = \frac{TP}{TP + FP} \tag{18}$$

$$R = \frac{TP}{TP + FN} \tag{19}$$

$$F = \frac{2PR}{P + R} \tag{20}$$

where TP, FP and FN mean the numbers of trajectories correctly grouped, wrongly grouped and belonging to but not grouped in this cluster, respectively. The final indexes for a clustering result is an average of all the clusters. Figure 7 shows the comparison results on 100 synthetic datasets. The average and standard error are detailed in Table 3. Apparently, both our proposed approach and the original IB algorithm work much better than the distance-driven k-means. Moreover, the proposal of introducing clustering quality to IB brings very promising improvement.

Table 3. Average and standard error of precision, recall and F-Measure on clustering quality.

	Proposed Approach		Original IB		*k*-means	
	Average	Standard Error	Average	Standard Error	Average	Standard Error
precision	0.9791	0.0046	0.9792	0.0045	0.8683	0.0170
recall	0.9447	0.0096	0.9341	0.0103	0.7649	0.0129
F-Measure	0.9475	0.0096	0.9402	0.0097	0.7703	0.0155

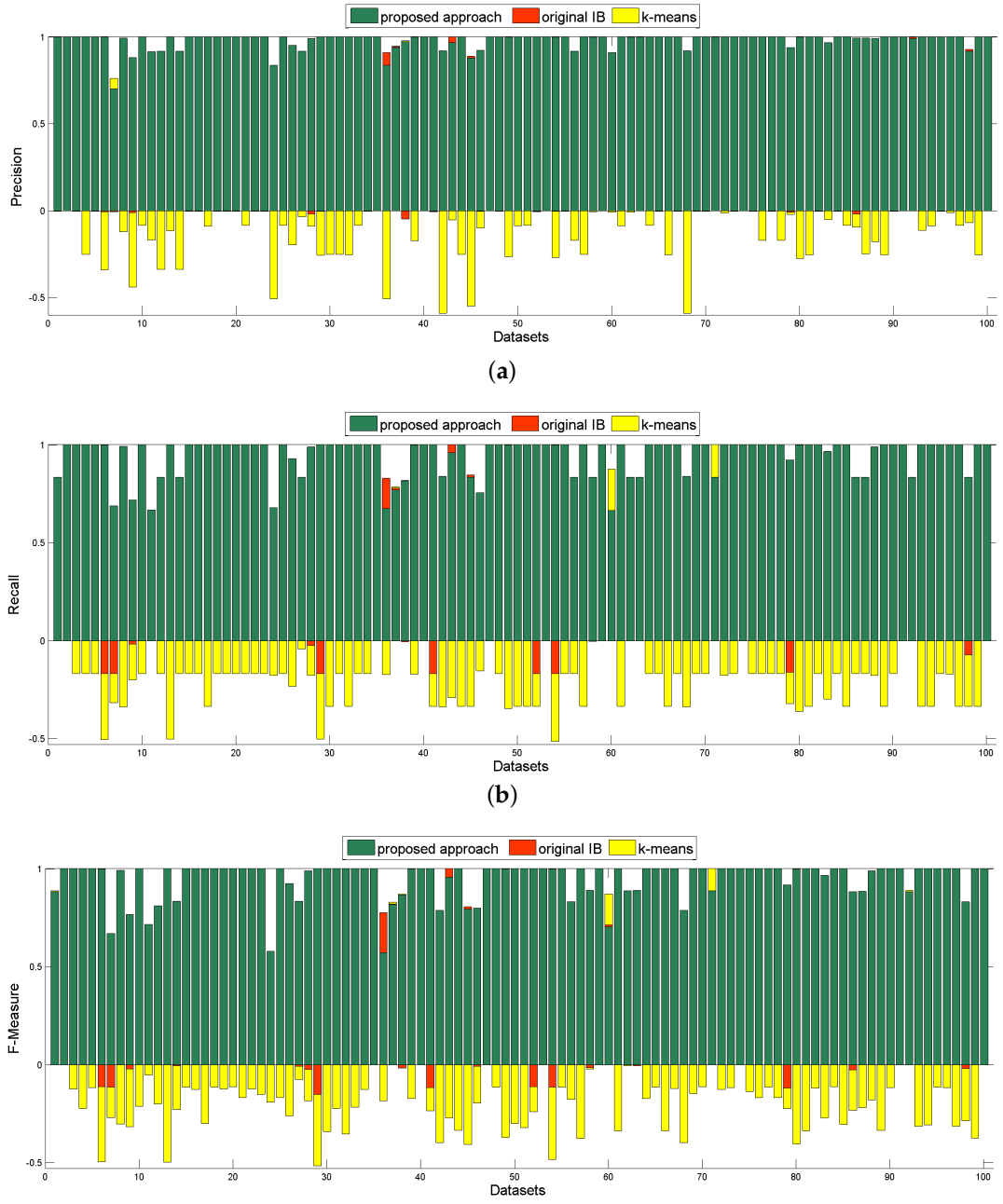


Figure 7. Clustering quality of 100 synthetic datasets. For each dataset, the real value of our proposed approach is presented by a bar, and the differences between another two methods and our approach are stacked on or under the same bar according to the corresponding difference is positive or negative. (**a**) Precision; (**b**) Recall; (**c**) F-Measure.

Besides, for anomaly detection, 10 anomalies in each cluster are taken as testing objects with 5 labeled clusters. We compare the proposed Shannon entropy based approach with SHNN-CAD ($k = 3, 4, 5$) on the accuracy and average runtime. As shown in Figure 8, the proposed Shannon entropy based approach performs slightly better than SHNN-CAD on most datasets in terms of identifying the abnormal trajectories. In some datasets, SHNN-CAD outperforms our proposed approach, that is because it uses the information of k nearest neighbors to identify the anomalous situation of a test trajectory. While in our Shannon entropy based approach, we simply use the medoids of learnt clusters. This experiment inspires us to make use of the k nearest neighbors in a cluster

instead of a single medoid for better performance, which will be attempted in our future work. In addition, the average runtimes for our approach and SHNN-CAD ($k = 3, 4, 5$) to identify a trajectory as normal or not are 0.031, 6.619, 6.740 and 7.219 s, respectively. Apparently, the computational cost of SHNN-CAD is higher, since for each test trajectory, SHNN-CAD needs to calculate the distance with each trajectory in the whole dataset to locate its k nearest neighbors. According to the average and standard error values in Table 4, our approach works the best among all the comparative methods from the overall perspective.

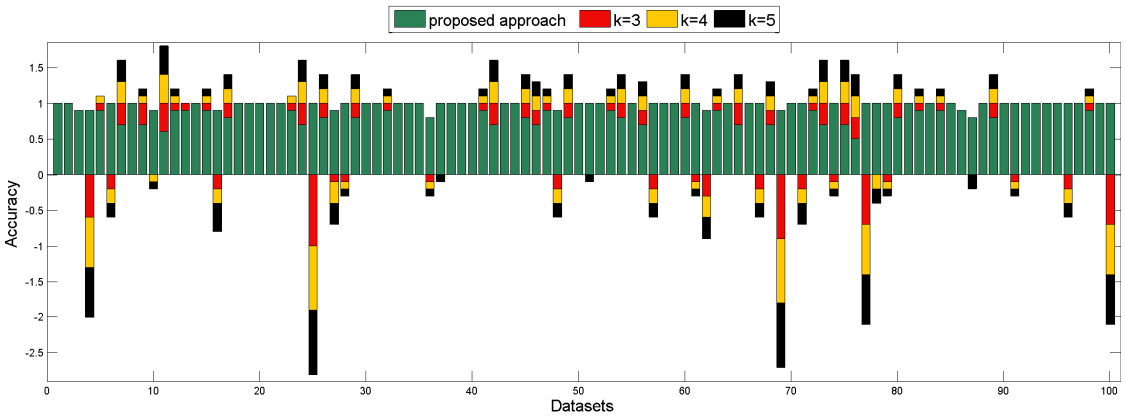


Figure 8. Accuracy of anomaly detection on 100 synthetic datasets. For each dataset, the real value of our proposed approach is presented by a bar, and the differences between Sequential Hausdorff Nearest-Neighbor Conformal Anomaly Detector (SHNN-CAD) ($k = 3, 4, 5$) and our approach are stacked on or under the same bar according to the corresponding difference is positive or negative.

Table 4. Average and standard error of accuracy on anomaly detection.

	Proposed Approach	**SHNN-CAD (k = 3)**	**SHNN-CAD (k = 4)**	**SHNN-CAD (k = 5)**
average	0.9160	0.9190	0.9100	0.9010
standard error	0.0114	0.0193	0.0194	0.0199

7. Conclusions and Future Work

In this paper, for the purpose of anomaly detection, we have established a new effective technique in which the speed attribute is explicitly utilized to improve the shape analysis of trajectory data. The 3D surface plot rendering of statistical models obtained by the nonparametric Kernel Density Estimation (KDE) demonstrates that the trajectory data can be depicted very well, since KDE does not need the difficult model fitting for some assumed parametric function which is not feasible in practice. Moreover, our proposal of using adaptive probability bins results in approximate 30% time saving on computing efficiency of further clustering process, and the underlying idea indeed works for complex trajectory data. A lot of experiments on both real-world and simulated trajectory data have revealed clearly that, the utilization of the Mutual Information (MI) and an extended version of Information Bottleneck (IB) for clustering can achieve more sound and promising performance than the typical distance-based k-means algorithm. What's more, our attempt of adding clustering quality to the objective function of IB attains effective and helpful results for alleviating the local optimum problem of IB. Actually, this could be a meaningful strategy for optimization problem. The Shannon entropy has been adopted for adaptively identifying whether a testing trajectory behaves as an anomaly or not. The accuracy and runtime comparison indicates that the Shannon entropy based anomaly detection approach outperforms the state-of-the-art method, Sequential Hausdorff Nearest-Neighbor Conformal Anomaly Detector (SHNN-CAD). The comparison of our technique between SHNN-CAD inspires us to take advantage of the k nearest neighbors for better trajectory processing.

In our future work, the typical transformations, such as Discrete Fourier Transformation and Discrete Wavelet Transformation, will be used for the speeding up of the trajectory preprocessing. Some more trajectory attributes, such as spatial position, acceleration, etc., will be considered for the comprehensive modeling of trajectory data. In order to obtain the multivariate statistical modeling, Maximum Entropy Principle will be exploited. As for IB, some techniques such as stochastic optimization will possibly serve for the goal of its global optimization [16]. For the incremental updating of the clusters having been learned, some mathematical representation such as the Dirichlet process mixture model [46] can be potentially exploited to achieve this. In addition, the general and powerful visual analytics technique [47] will be possibly developed to propose attractive visualizations and effective user interactions to further enhance all the steps of the trajectory analysis.

Acknowledgments: This work has been funded by Natural Science Foundation of China (61471261, 61179067, U1333110) and Spanish ministry MINECO (TIN2016-75866-C3-3-R). First author acknowledges the support from Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement de la Generalitat de Catalunya and the European Social Fund.

Author Contributions: Yuejun Guo and Qing Xu proposed the computational mechanism and the use of Kernel Density Estimation and information theory tools for the trajectory anomaly detection. Peng Li and Yu Yang implemented the approach and the experiments. Mateu Sbert introduced the use of Shannon entropy in trajectory anomaly detection. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dee, H.M.; Velastin, S.A. How close are we to solving the problem of automated visual surveillance? *Mach. Vis. Appl.* **2008**, *19*, 329–343.
2. Ciccio, C.D.; van der Aa, H.; Cabanillas, C.; Mendling, J.; Prescher, J. Detecting flight trajectory anomalies and predicting diversions in freight transportation. *Decis. Support Syst.* **2016**, *88*, 1–17.
3. Smart, E.; Brown, D. A Two-Phase Method of Detecting Abnormalities in Aircraft Flight Data and Ranking Their Impact on Individual Flights. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1253–1265.
4. Morris, B.T.; Trivedi, M.M. Understanding vehicular traffic behavior from video: a survey of unsupervised approaches. *J. Electron. Imaging* **2013**, *22*, 041113.
5. Hu, W.; Xie, D.; Tan, T.; Maybank, S. Learning Activity Patterns Using Fuzzy Self-Organizing Neural Network. *IEEE Trans. Syst. Man Cybern. Part B* **2004**, *34*, 1618–1626.
6. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection: A Survey. *ACM Comput. Surv.* **2009**, *41*, 1–58.
7. Calderara, S.; Prati, A.; Cucchiara, R. Mixtures of von Mises Distributions for People Trajectory Shape Analysis. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 457–471.
8. Prati, A.; Calderara, S.; Cucchiara, R. Using Circular Statistics for Trajectory Shape Analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
9. Mariescu-Istodor, R.; Tabarcea, A.; Saeidi, R.; Fränti, P. Low Complexity Spatial Similarity Measure of GPS Trajectories. In Proceedings of the 10th International Conference on Web Information Systems and Technologies, Barcelona, Spain, 3–5 April 2014; pp. 62–69.
10. Markovic, I.; Cesic, J.; Petrovic, I. Von Mises Mixture PHD Filter. *IEEE Signal Process. Lett.* **2015**, *22*, 2229–2233.
11. Sieranoja, S.; Kinnunen, T.; Fränti, P. GPS Trajectory Biometrics: From Where You Were to How You Move. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*; Robles-Kelly, A., Loog, M., Biggio, B., Escolano, F., Wilson, R., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 450–460.
12. Calderara, S.; Prati, A.; Cucchiara, R. Learning People Trajectories using Semi-directional Statistics. In Proceedings of the Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, Genova, Italy, 2–4 September 2009; pp. 213–218.
13. Ding, H.; Trajcevski, G.; Scheuermann, P.; Wang, X.; Keogh, E. Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures. *Proc. VLDB Endow.* **2008**, *1*, 1542–1552.

14. Tishby, N.; Pereira, F.C.; Bialek, W. The information bottleneck method. In Proceedings of the 37th annual Allerton Conference on Communication, Control, and Computing, Citeseer, Chicago, IL, USA, 22–24 September 1999; pp. 368–377.
15. Guo, Y.; Xu, Q.; Liang, S.; Fan, Y.; Sbert, M. XaIBO: An Extension of aIB for Trajectory Clustering with Outlier. In *Neural Information Processing;* Lecture Notes in Computer Science; Arik, S., Huang, T., Lai, W.K., Liu, Q., Eds.; Springer: Cham, Switzerland, 2015; Volume 9490, pp. 423–431.
16. Slonim, N. The Information Bottleneck: Theory and Applications. Ph.D. Thesis, Hebrew University of Jerusalem, Jerusalem, Israel, 2002.
17. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*, 2nd ed.; Wiley-Interscience: San Francisco, CA, USA, 2006.
18. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 1–38.
19. Junejo, I.N.; Javed, O.; Shah, M. Multi Feature Path Modeling for Video Surveillance. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 26 August 2004; Volume 2, pp. 716–719.
20. Anjum, N.; Cavallaro, A. Multifeature Object Trajectory Clustering for Video Analysis. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 1555–1564.
21. De Vries, G.; van Someren, M. Clustering Vessel Trajectories with Alignment Kernels under Trajectory Compression. In *Machine Learning and Knowledge Discovery in Databases*; Springer, Berlin/Heidelberg, Germany, 2010; pp. 296–311.
22. De Vries, G.K.D.; Someren, M.V. Machine learning for vessel trajectories using compression, alignments and domain knowledge. *Exp. Syst. Appl.* **2012**, *39*, 13426–13439.
23. Annoni, R.; Forster, C.H.Q. Analysis of Aircraft Trajectories Using Fourier Descriptors and Kernel Density Estimation. In Proceedings of the15th International IEEE Conference on Intelligent Transportation Systems, Anchorage, AK, USA, 16–19 September 2012; pp. 1441–1446.
24. Zheng, Y. Trajectory Data Mining: An Overview. *ACM Trans. Intell. Syst. Technol.* **2015**, *6*, 1–41.
25. Cilibrasi, R.; Vitányi, P.M.B. Clustering by Compression. *IEEE Trans. Inf. Theory* **2005**, *51*, 1523–1545.
26. Izakian, H.; Pedrycz, W. Anomaly Detection and Characterization in Spatial Time Series Data: A Cluster-Centric Approach. *IEEE Trans. Fuzzy Syst.* **2014**, *22*, 1612–1624.
27. Ge, W.; Collins, R.T.; Ruback, R.B. Vision-based Analysis of Small Groups in Pedestrian Crowds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1003–1016.
28. Steeg, G.V.; Galstyan, A.; Sha, F.; DeDeo, S. Demystifying Information-Theoretic Clustering. *arXiv* **2013**, arXiv:1310.4210.
29. Laxhammar, R.; Falkman, G. Online Learning and Sequential Anomaly Detection in Trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1158–1173.
30. Zhang, D.; Li, N.; Zhou, Z.H.; Chen, C.; Sun, L.; Li, S. iBAT: Detecting Anomalous Taxi Trajectories from GPS Traces. In Proceedings of the 13th International Conference on Ubiquitous Computing, Beijing, China, 17–21 September 2011; pp. 99–108.
31. Wand, M.P.; Jones, M.C. *Kernel Smoothing, Monographs on Statistics and Applied Probability*; CRC Press: Boca Raton, FL, USA, 1995; Volume 60, p. 91.
32. Silverman, B.W. *Density Estimation for Statistics and Data Analysis*; Chapman and Hall: London, UK, 1986.
33. Lin, J. Divergence Measures Based on the Shannon Entropy. *IEEE Trans. Inf. Theory* **1991**, *37*, 145–151.
34. Slonim, N.; Tishby, N. Agglomerative Information Bottleneck. *Adcances Neural Inf. Process. Syst.* **1999**, *12*, 617–623.
35. Kailath, T. The Divergence and Bhattacharyya Distance Measures in Signal Selection. *IEEE Trans. Commun. Technol.* **1967**, *15*, 52–60.
36. Aircraft Trajectory Dataset. Available online: https://c3.nasa.gov/dashlink/resources/132/ (accessed on 9 June 2017).
37. Gariel, M.; Srivastava, A.N.; Feron, E. Trajectory Clustering and an Application to Airspace Monitoring. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1511–1524.
38. Guo, Y.; Xu, Q.; Fan, Y.; Liang, S.; Sbert, M. Fast Agglomerative Information Bottleneck Based Trajectory Clustering. In Proceedings of the 23rd International Conference on Neural Information Processing, Kyoto, Japan, 16–21 October 2016; pp. 425–433.

39. Morris, B.; Trivedi, M. Learning Trajectory Patterns by Clustering: Experimental Studies and Comparative Evaluation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 312–319.
40. MIT Trajectory Dataset. Available online: http://www.ee.cuhk.edu.hk/~xgwang/MITtrajsingle.html (accessed on 9 June 2017).
41. Wang, X.; Ma, K.T.; Ng, G.W.; Grimson, W.E.L. Trajectory Analysis and Semantic Region Modeling Using A Nnonparametric Bayesian Model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
42. Synthetic Trajectory Dataset. Available online: https://avires.dimi.uniud.it/papers/trclust/ (accessed on 9 June 2017).
43. Piciarelli, C.; Micheloni, C.; Foresti, G.L. Trajectory-Based Anomalous Event Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 1544–1554.
44. Halkidi, M.; Batistakis, Y.; Vazirgiannis, M. On Clustering Validation Techniques. *J. Intell. Inf. Syst.* **2001**, *17*, 107–145.
45. Larsen, B.; Aone, C. Fast and Effective Text Mining Using Linear-time Document Clustering. In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 15–18 August 1999; pp. 16–22.
46. Hu, W.; Li, X.; Tian, G.; Maybank, S.; Zhang, Z. An Incremental DPMM-Based Method for Trajectory Clustering, Modeling, and Retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1051–1065.
47. May, R.; Hanrahan, P.; Keim, D.A.; Shneiderman, B.; Card, S. The State of Visual Analytics: Views on what visual analytics is and where it is going. In Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, Salt Lake City, UT, USA, 25–26 October 2010; pp. 257–259.