Trajectory Data Classification: A Review

JIANG BIAN, University of Technology Sydney, Australia
DAYONG TIAN, Northwestern Polytechnical University, China
YUANYAN TANG, University of Macau, China
DACHENG TAO, University of Sydney, Australia

This article comprehensively surveys the development of trajectory data classification. Considering the critical role of trajectory data classification in modern intelligent systems for surveillance security, abnormal behavior detection, crowd behavior analysis, and traffic control, trajectory data classification has attracted growing attention. According to the availability of manual labels, which is critical to the classification performances, the methods can be classified into three categories, i.e., unsupervised, semi-supervised, and supervised. Furthermore, classification methods are divided into some sub-categories according to what extracted features are used. We provide a holistic understanding and deep insight into three types of trajectory data classification methods and present some promising future directions.

CCS Concepts: • **Computing methodologies → Machine learning approaches**; *Classification and regression trees*;

Additional Key Words and Phrases: Trajectory classification, object movement, classification algorithms, review

ACM Reference format:
Jiang Bian, Dayong Tian, Yuanyan Tang, and Dacheng Tao. 2019. Trajectory Data Classification: A Review. *ACM Trans. Intell. Syst. Technol.* 10, 4, Article 33 (August 2019), 34 pages.
https://doi.org/10.1145/3330138

1 INTRODUCTION

With the development of tracking and surveillance devices, tremendous numbers of object trajectory data were collected, which make extracting useful information imperative and challenging. Trajectory classification is an efficient way to analyze trajectory data and it has been applied in pattern recognition, data analysis, machine learning, and so on. Furthermore, trajectory classification is used to obtain spatiotemporal information inside trajectory data, so it is ubiquitous in some application fields, such as object motion prediction [26], traffic monitoring [7, 50, 75], activity

This work was supported by the China Scholarship Council (CSC): Grant No. 201406960011. This work was partially supported by National Science Foundation: Grant No. 61806166.
Authors' addresses: J. Bian, University of Technology Sydney, 15 Broadway, Sydney, Australia; email: bianjiang22@gmail.com; D. Tian, Northwestern Polytechnical University, Xi'an, China; email: dayongt@gmail.com; Y. Tang, University of Macau, Avenida da Universidade, Macau, China; email: yytang@umac.mo; D. Tao, University of Sydney, 1 Cleveland St, Sydney, Australia; email: dacheng.tao@sydney.edu.au.

2157-6904/2019/08-ART33 $15.00
https://doi.org/10.1145/3330138

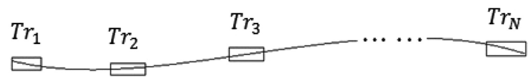


Fig. 1. Trajectory generated by GPS tracking devices.

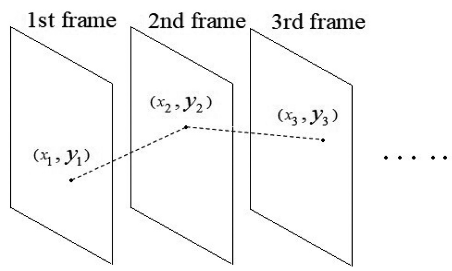


Fig. 2. Trajectory generated from camera device.

understanding [10, 132, 146], abnormal detection [22, 135, 154, 159], three-dimensional reconstruction [68], weather forecasting [39], and geography [86].

Trajectory data are recorded in different formats according to device types, object motion, or even purposes. For instance, Global Positioning System (GPS) tracking devices generate a trajectory by tracking object movement as $Trajectory = (Tr_1, Tr_2, \ldots, Tr_N)$, which is a consecutive sequence of points in geographical space, and Tr_i denotes a combination of coordinates and time stamp like $Tr_i = (x_i, y_i, t_i)$, as shown in Figure 1. In some specific circumstances, other properties relevant to object movement are added, such as velocity, direction, acceleration, or geographic information [149, 150].

Different from GPS devices, which record position information of trajectory data only, trajectory data also can be generated from image data or video data. In some papers, the interest points are located as initial points of trajectories, and models are used to track the interest points in the following images. As shown in Figure 2, for image data, a sequence of pixels in consecutive frames form up a trajectory, which is similar to optical flow [20, 131]. Furthermore, scale-invariant feature descriptors are employed to track the points in a video data set as well [58, 123]. With the trajectory data generated from images or videos, spatiotemporal and image information, including pixels or scale, is employed. However, the semantic trajectory data have attracted more attention recently [29, 160], because they contain more information to improve classification accuracy and can be used directly and hence save more time.

Semi-supervised and supervised trajectory classifications aim at assigning a label to test data from a predefined set of labeled data set, so the algorithms are trained by labeled trajectory data. With the development of devices, semantic information involving classification may reduce the total number of training data and improve the performance. However, the training data should still be labeled by a human expert in advance. In some circumstances, labeling data are impossible, such as grouping pedestrian trajectories in crowds and action recognition in surveillance, which need to recognize objects in a crowd scene. Thus, trajectory data are required to be classified without labels, and these unsupervised trajectory classification methods are called trajectory clustering methods in this review.

Recently, some papers have been published to review the topic of trajectories classification [66, 84, 88, 153]. Reference [66] gives a brief review on trajectory data and relevant usage in the reality including application of trajectories, service of trajectories and challenge of trajectories. It mainly talks about trajectory data generation and usage. Reference [84] focuses on trajectory data mining technologies based on different application fields. It gives a high-level view on trajectory data mining, but only brief introduction is given. Furthermore, it lacks of the comparison between different methods and technical details analysis. Reference [88] reviews vision-based trajectory data classification methods. It lists the advantages/disadvantages of methods, and different application scenes have also been considered. Reference [88] discusses the papers by following the clustering procedure, including data preprocessing, clustering, and path modeling, which is used for further inferring. However, only limited models have been reviewed and it mainly discusses about the trajectory data generated from surveillance. Reference [153] reviews the corresponding field as four aspects including measurement of trajectories computation, trajectories classification, clustering validation and application scenes. It mainly focuses on the papers classifying trajectories by spatiotemporal information. However, spatiotemporal information cannot achieve satisfactory robustness, effectiveness and accuracy alone. Therefore, methods using different features and data formats are reviewed in this article.

This survey discusses the application problems that all reviewed papers working on, the preparation methods and technical details of each papers are also listed. Furthermore, the performance and the differences are compared in the following tables at the end of each section and they reveal the weak/strong points of the reviewed models. Finally, state-of-the-art models are listed after introducing the experiment data sets and papers. For brief details, this survey divides trajectory classification methods into three categories, unsupervised, supervised, semi-supervised. In each section, the methods are further classified into different subsections according to their different feature usage methods. Unsupervised models aim at clustering data without human experts' supervision or training data. An inference function has been drawn by analyzing unlabeled data sets [37, 39, 139, 143]. Supervised models are learned prior to trajectory classification. Generally, labeled data are used to learn a function mapping data to their labels or groups. The categorizations of unlabeled data are predicted by this function [27, 45, 141, 154]. Labeling data need a lot of manual works by human experts. It is unfeasible for large data sets. Semi-supervised methods compromise the previous two types of models. They are trained by partial labeled data [50, 135, 154].

To measure similarities among different types of trajectory data, data representation, feature extraction and distance metric selection are critical preliminary works of trajectory classification. For example, trajectories can be represented as a vector and downsampled to a unified length, so Euclidean distance is used [91]. Trajectories also can be treated as samples of a probabilistic distribution. Hence, Bhattacharyya Distance [76] is used to measure the distance between two distributions.

Generally, trajectory data are collected by location recording devices and trajectory classification are divided into three steps: trajectory preparation, feature extraction and classification. Due to their different lengths, trajectories are unified into a fixed length in preparation step [11, 53, 54, 155]. However, in recent years, some algorithms are proposed to prepare trajectories without unifying the lengths, such as Reference [154], which segments trajectory into sub-trajectories with a fixed length. Furthermore, trajectory classification in computer version field needs one more procedure, which is tracking objects through all images to generate trajectory data before all steps [58, 129]. In feature extraction, spatial information is extracted to characterize trajectory data. Then, trajectory data are classified directly by measuring the spatial distance in previous works [76, 91]. However, in some recent papers, probabilistic inference models involve to solve the problem by classifying trajectory data directly without feature extraction step [12, 32]. As Figure 3 shows, the procedure of trajectory data classification is given.

Thus, the rest of this article is organized as follows. Preliminary works and popular experiment data sets are introduced in Section 2, and the models based on unsupervised algorithms are

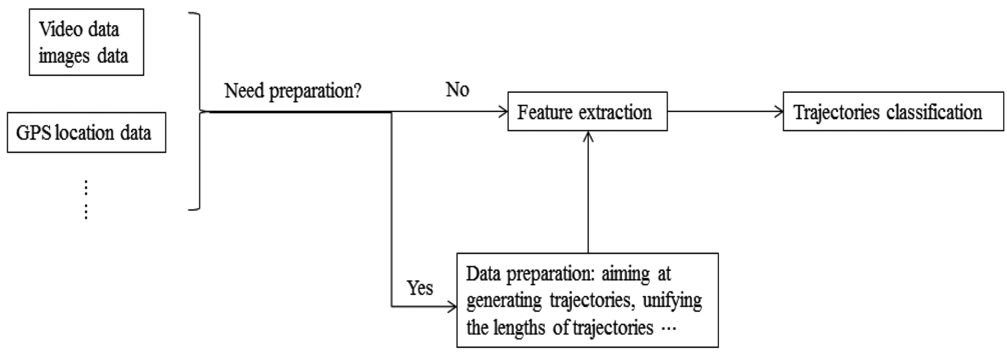


Fig. 3. The procedure of trajectory data classification.

described in Section 3. Description of the supervised models are presented in Section 4. Section 5 discusses the models based on semi-supervised algorithms. Finally, promising future directions are given in Section 6, and conclusions are made in Section 7.

2 PRELIMINARIES

2.1 Trajectory Classification Preparation

In some classification models [54, 90, 110, 155], trajectory data are required to be set as a unified length so that they could be measured. However, as shown in Figure 4, for two arbitrary trajectories, their lengths may be largely different from each other. Therefore, representing trajectories in a unified length with little loss of information is a major preliminary work of these models. This procedure is called classification preparation.

2.1.1 Trajectory Transformation Algorithms. For some methods, original data are represented in other space with the same length. For instance, trajectory data are projected into a subspace [53]. Linear transformation algorithms aim at representing trajectory as a combination of basis trajectories [1]. Curve fitting is another method to approximate trajectories by a parameterized quadratic curve [155]. To distinguish similar curves, the direction of the last trajectory point is chosen as an additional parameter. In Reference [110], trajectory data are approximated by a uniform cubic B-spline curve, so that a representation capable of encoding both the shape and the spatiotemporal profile of trajectory data is obtained. In addition, the lengths of trajectories are added to distinguish the trajectories with similar shapes. According to the fact that trajectory data contain a lot kinds of positional information, such as coordinates, speed and directions, vector fields are employed to represent trajectory data [39]. Vector fields give trajectory a smooth streamline and induce a notion of similarity of trajectories. Principal Component Analysis (PCA) is a statistical procedure to compute a set of linearly uncorrelated variables called principal components by orthogonal transformation. To avoid partially extracted information, a number of organized segmentations substitute for the corresponding trajectory in References [10] and [11]. The time ordering data are transformed and represented in frequency domain by Discrete Fourier Transformation (DFT), so a trajectory can be represented as a fixed length vector composed of Fourier coefficients in References [54] and [90]. In Reference [58], the interaction of trajectories are encoded and set as elements of codebook, so camera motion is ignored and the model's robustness is improved.

2.1.2 Re-sampling Methods. Re-sampling methods choose trajectory points by sampling rule to unify trajectory lengths. Trajectory data are segmented as sub-trajectories, and all of them are re-sampled to a fixed length so that sub-trajectories are arranged as a matrix [11]. In some complex

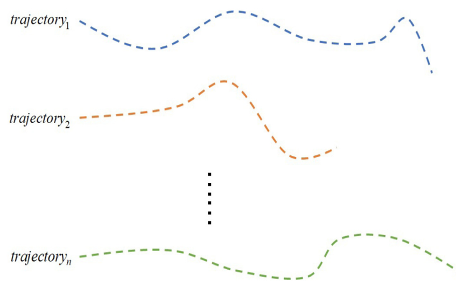


Fig. 4. Trajectory data with varying lengths.

scenes, for example, hand writing data set, Equidistant sampling fixes the problem that two same characters are recorded in different temporal sequence because of different writing speeds [105]. Since re-sampled trajectory points are discontinuous, it is critical that normalization should be involved after re-sampling [79]. It has been widely acknowledged that re-sampling method causes information loss [101]. Therefore, sparsity regularization is used in References [21, 35, 93, 134].

2.1.3 Trajectory Substitute. Sub-trajectories hold partial and hidden information of original trajectory data [54, 73], so they put together and describe trajectory with more flexibility. For instance, the latent motion rule beneath hurricane trajectories is figured out and a certain hurricane trend chart is printed by analyzing sub-trajectories of past hurricane trajectories in Reference [39]. Sub-trajectories also lead to simplified trajectories, which represent trajectory data as some smaller, less complex primitives suitable for storage and retrieval purposes [3]. In Reference [145], sub-trajectories are generated by well-defined policies based on facility performance, time range or distance range. For example, sub-trajectories can be segmented if they stay in different time windows. In References [10] and [11], trajectory is segmented at the so-called changing points at which direction or speed changes dramatically. Curvature describes direction information, and it could be extracted if a trajectory is treated as a curve by connecting consecutive trajectory points. Curvatures are computed by transforming three-dimensional position coordinates of points into spherical system and quantized as *up, down, left, right* [38], then a trajectory is segmented at the points where curvature changes. In addition, Minimum Description Length (MDL) principle traces the sub-trajectories to estimate trajectory motion by minimizing the differences between sub-trajectories and the corresponding trajectories in Reference [73]. Minimum Bounding Rectangles (MBR) is proposed to separate trajectories under occlusion and optimize the separability of inter-object in Reference [3]. It optimizes the bounding rectangles containing sub-trajectories to ensure that the distance between two rectangles are closer than the distance of trajectories.

Some specific regions of surveillance area hold special semantic information and attract more attentions, so Regional Segmenting method is implemented. The whole scene is split into several regions, and boundaries of the regions segment trajectories [157]. As independent motion pattern, sub-trajectories characterize more information while original trajectory presents limited information.

2.1.4 Points of Interest. Some specific regions of surveillance area hold special semantic information. Thus, the points inside the regions are used to represent trajectory or scene in Reference [122] and all these points are called Points of Interest (POI). The points outside the regions are ignored, because they are short of useful information. For instance, activity analysis is a key part in surveillance application to seek low-level situational awareness by understanding and characterizing behaviors of objects in the scene [88], so it is critical to extract POI in the special regions. In topographical map, POI inside the special regions are represented as a single node. For example, two types of POI are introduced in Reference [88] where the first ones are the points in entry/exit zones and the second one are the points at the scene landmarks that objects intend to approach, move away, or stay for a long time. Except for the special areas, points are represented by a node if their speed are less than a threshold in References [13] and [89]. The importance of points can be measured and the high-scored ones are selected in Reference [165]. For video data, POI are obtained by Pyramid Representation [129]. In addition, optical flow is another popular implementation by estimating trajectory motion in References [40] and [131].

2.1.5 Scale-invariant Features. In image frames, more robust and representative features are needed rather than only positional information of trajectory points in References [58] and [129]. In Reference [129], histograms of oriented gradients (HOG) and histograms of optical flow (HOF) features are used to describe static appearance information and local motion information of trajectories, respectively. HOG feature computes orientation information to keep scale-invariant property of tracking point and it is fast to implement [58, 65, 69, 82, 83, 140]. Furthermore, the Scale-invariant Feature Transform (SIFT) descriptor represents an image patch around a tracking point [115, 123, 126, 128, 130], and it computes scale and orientation information of image patches to localize the tracking object in consecutive frames. In Reference [123], the Kanade-Lucas-Tomasi (KLT) tracker is used to find trajectory points and SIFT is applied to represent them. In Reference [128], the Difference-of-Gaussian (DOG) detector is used to detect trajectory points instead of KLT in Reference [123].

2.2 Common Distance Measurements

Essentially, trajectories are allocated into cohesive groups according to their mutual similarities. For example, Reference [151] allocates objects to the closest event by measuring the distance between objects and events directly. An appropriate metric is necessary [8, 87, 156].

Euclidean Distance: Euclidean distance requires that lengths of trajectories should be unified and the distances between the corresponding trajectories points should be summed up,

$$D(Trajectory_i, Trajectory_j) = \frac{1}{N}\sum_n [(x_n^i - x_n^j)^2 + (y_n^i - y_n^j)^2]^{\frac{1}{2}}, \tag{1}$$

where x_n^i and y_n^i indicate the nth point of trajectory $Trajectory_i$ in the Cartesian coordinate. N is the total number of points. Euclidean distance is used to measure the distance of trajectories in Reference [91].

Hausdorff Distance: Hausdorff distance measures the similarities by considering how close every point of one trajectory to some points of the other one. For example, it measures $Trajectory_i$ and $Trajectory_j$ without unifying the lengths in References [24, 80],

$$D(Trajectory_i, Trajectory_j) = \max\{d(Trajectory_i, Trajectory_j), d(Trajectory_j, Trajectory_i)\}, \tag{2}$$

$$\begin{cases} d(Trajectory_i, Trajectory_j) = \max\limits_{Tr_i \in Trajectory_i} \min\limits_{Tr_j \in Trajectory_j} ||Tr_i - Tr_j||, \\ d(Trajectory_j, Trajectory_i) = \max\limits_{Tr_j \in Trajectory_j} \min\limits_{Tr_i \in Trajectory_i} ||Tr_j - Tr_i||. \end{cases} \tag{3}$$

Bhattacharyya Distance: Bhattacharyya distance measures how closely of two probability distributions. In Reference [76], it is employed to measures similarities of quantized directions of points,

$$D(Trajectory_i, Trajectory_j) = -\ln(BC(Trajectory_i, Trajectory_j)), \tag{4}$$

where $BC(Trajectory_i, Trajectory_j) = \sum_{n=1}^{N} \sqrt{Dir_n^i \cdot Dir_n^j}$, and it is used to measure the separability of $Trajectory_i$ and $Trajectory_j$. Dir_n^i and Dir_n^j are quantized directions.

Frechet distance: Frechet distance measures similarity between two curves by taking into account location and time ordering. After obtaining the curve approximations of $Trajectory_i$ and $Trajectory_j$, their curves map unit interval into metric space S, and a re-parameterization is added to make sure t cannot be backtracked. Frechet distance is defined as

$$D(Trajectory_i, Trajectory_i) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \{d(Trajectory_i(\alpha(t)), Trajectory_j(\beta(t)))\}, \tag{5}$$

where d is distance function of S, α, β are continuous and non-decreasing re-parameterization.

Dynamic Time Warping (DTW) Distance: DTW is a sequence alignment method to find an optimal matching between two trajectories and measure the similarity without considering lengths and time ordering [12, 107].

$$W(Trajectory_i, Trajectory_j) = \min_{f} \frac{1}{N} \sum_{n=1}^{N} ||Tr_n^i - Tr_{f(n)}^j||_2, \tag{6}$$

where $Trajectory_i$ has N points and $Trajectory_j$ has M points, all mappings $f : [1, N] \rightarrow [1, M]$ should satisfy the requirements that $f(1) = 1,\ f(N) = M$ and $f(i) \leq f(j),\ for\ all\ 1 \leq i \leq j \leq N$.

Longest Common Subsequence (LCSS) Distance: LCSS aims at finding the longest common subsequence in all sequences, and the length of the longest subsequence could be the similarity between two arbitrary trajectories with different lengths. The distance $LCSS_{\epsilon,\delta}(Trajectory_i, Trajectory_j)$ is written as

$$LCSS_{\epsilon,\delta}(Trajectory_i, Trajectory_j) = \begin{cases} 0,\ if Trajectory_i\ or\ Trajectory_j\ is\ empty, then \\ 1 + LCSS_{\epsilon,\delta}(Head(Trajectory_i), Head(Trajectory_j)), \\ \quad if ||Tr_N^i - Tr_M^j|| < \epsilon\ and\ |N - M| < \delta, then \\ max(LCSS_{\epsilon,\delta}(Head(Trajectory_i), Trajectory_j), \\ \quad LCSS_{\epsilon,\delta}(Trajectory_i, Head(Trajectory_j))),\ otherwise, \end{cases} \tag{7}$$

where $Head(Trajectory_i)$ indicates first $N-1$ points belonging to $Trajectory_i$ and $Head(Trajectory_j)$ denotes first $M-1$ points of $Trajectory_j$. Finally, $D(Trajectory_i, Trajectory_j) = 1 - \frac{F}{\max(N,M)}$ where F indicates $LCSS_{\epsilon,\delta}(Trajectory_i, Trajectory_j)$. In References [149, 150], a modified LCSS distance is employed to classified semantic trajectory data.

Spatial Network Distance: Much like a transportation network, geometric objects are indicated as vertices of graph, and the edges between two arbitrary vertices are computed by a specific method. This graph is called spatial network or geometric graph. For example, the Euclidean distance between two vertices can be set as the edge when the corresponding distance between two vertices are smaller than a fixed value. In Reference [120], trajectories are presented in a road map, because objects cannot move freely on the road, and Reference [120] only considers spatial information and time series case is not taken into account due to it not always meaningful in spatial networks. For instance, Spatial Network distance in Reference [120] is formulated as

$$d(Tr_n^i, Tr_m^j) = \begin{cases} 0,\ c(Tr_n^i, Tr_m^j) \wedge c(Tr_m^j, Tr_n^i) = 0 \\ \dfrac{\min c(Tr_n^i, Tr_m^j), c(Tr_m^j, Tr_n^i)}{\max c(Tr_n^i, Tr_m^j), c(Tr_m^j, Tr_n^i)},\ otherwise, \end{cases} \tag{8}$$

Table 1. Summary of Common Distance Measurements

Measurement types	Unifying lengths	Computational complexity
Euclidean distance	Yes	$O(n)$
Hausdorff distance	No	$O(mn)$
Bhattacharyya distance	Yes	$O(n)$
Frechet distance	No	$O(mn)$
LCSS distance	No	$O(mn)$
DTW distance	No	$O(mn)$
Spatial Network distance	Yes	$O(n)$
other distance types	No	$O(1)$

where $d(Tr_n^i, Tr_m^j)$ indicates the distance between the nth point of trajectory i and mth point of trajectory j, and $c(Tr_n^i, Tr_m^j)$ denotes the cost from the nth point of trajectory i to mth point of trajectory j. Then, the network distance can be obtained,

$$D_{net}(trajectory_i, trajectory_j) = \frac{1}{N} \cdot \sum_{n=1}^{N} (d(Tr_n^i, Tr_n^j)), \tag{9}$$

where trajectory i and j should have a fixed length.

Other distance types: In References [72, 73, 77], more other distance types are proposed to consider more properties such as angle distance, center distance and parallel distance. Angle distance is defined as

$$d_{angle}(Trajectory_i, Trajectory_j) = \begin{cases} ||Trajectory_j|| \times \sin(\theta), \ 0^o \leq \theta \leq 90^o, \\ ||Trajectory_j||, \ 90^o \leq \theta \leq 180^o, \end{cases} \tag{10}$$

where θ is the smaller intersecting angle between $Trajectory_i$ and $Trajectory_j$. For center distance,

$$d_{center}(Trajectory_i, Trajectory_j) = ||Tr_{centre}^i - Tr_{centre}^j||, \tag{11}$$

where $d_{center}(Trajectory_i, Trajectory_j)$ is the Euclidean distance between center points of $Trajectory_i$ and $Trajectory_j$. And parallel distance is

$$d_{parallel}(Trajectory_i, Trajectory_j) = \min(l_1, l_2), \tag{12}$$

where l_1 is the Euclidean distances of p_s to s_i and l_2 is that of p_e to e_i. p_s and p_e are the projection points of s_j and e_j onto $Trajectory_i$, respectively.

Distance metrics are used in much more fields relating to trajectories classification, e.g., density clustering [4, 18, 72, 73, 94]. It is critical to choose an optimal distance according to the application. For instance, LCSS distance is proved to provide outperforming performance without concerning trajectories length [87]. Hausdorff distance aims at finding the minimum distance between two trajectories and ignores time-order in data. Furthermore, a comparison of distance is listed in Table 1.

2.3 Data Sets

In the case that different trajectory data sets are collected from different scenes or contain different type data, so this section briefly introduces popular trajectory data sets or methods relating to generating trajectory data. Furthermore, trajectories classification has different application problems, including action recognition, discovery of social events, discovery of hot route, and objects detection and recognition. Thus, there are a lot of data sets available to test trajectories classification models, and some popular ones are reviewed in the following. After reviewing the data sets, it should be clearer to figure out the state-of-the-art trajectories classification models.

(1) Hollywood2 data set: It contains 12 classes of human actions distributed over 3,669 video clips, which were collected from 69 movies, so it mainly used to test trajectories classification in action recognition [42, 92, 104, 133, 148]. However, it still has other application fields with this data set, such as video categorization [59, 142].
(2) HMDB: It mainly collected from movies and small part of data set is come from public data sets such as YouTube. HMDB contains 51 human actions distributed over 6,849 video clips, and each action has 101 clips. Furthermore, not only body movements are included, general facial actions and facial actions with objects are recorded as well. Therefore, it mainly applied to action recognition [25, 42, 92, 133].
(3) OlympicSports: It contains sports videos of athletes participating 16 different sports, and all videos are collected from YouTube. Comparing to the previous two data sets, the scenarios of OlympicSports are simpler, and limited objects are used. Thus, it is mainly used for action recognition [42, 92, 133].
(4) UCF: UCF data sets are separated into more than 10 data sets, which are categorized by the amount number of groups, applying purposes and recording devices. UCF50 and UCF Sport are the main data sets involved in trajectories classification [25, 33, 51, 103, 104, 109, 121, 133]. UCF50 contains 50 classes of human actions, which are challenging because of large variations. Furthermore, 50 classes are clustered as 25 groups standing for 25 different persons, backgrounds and viewpoints. UCF Sports collected the human actions relating to sports activities.
(5) TRECVID: It contains videos with detailed ground truth data and more than 30 semantic features are required in the data set. Thus, not only is action recognition [51] the main application problem, which this data set concentrates on, but other applications are involved as well, such as discovery of social events [44].
(6) UCSD: UCSD data set records video data from a mounted camera watching over pedestrian walkways. It mainly records pedestrians in different crowd density from sparse to crowd. Furthermore, normal and abnormal events are the top categorizations, and different scenes are further criterions to separate each category. Thus, this data set mainly works for action recognition [36] and discovery of social events [113, 154].

In detail, Hollywood2 and HMDB data sets are more complicated than others, because they involve in general scenes such as meeting, public scenes, interactions with persons or objects. Furthermore, they all contain much more video clips. Thus, complicated body movements and facial actions in different backgrounds and illumination conditions are the features of these data sets. Sport events are the main scenes collecting as OlympicSports and UCF data set. They contain a lot of videos involving many sporting events, and UCF also contains a few other activities, such as playing musical instruments and fitness, but backgrounds and illuminations of these data sets are simpler comparing to the previous. For example, cycling track is the main scene except for players in cycling matches. TRECVID is also a video data set containing different actions covering different scenes, but it has fewer actions than the others. For UCSD data set, only pedestrian walkways are recorded, so the data set is appropriate for social events discovery and persons interaction analysis. Therefore, the classification results and even trajectories generation are affected by the complexity and diversity of scenarios, for example, in Reference [133], feature descriptor is working better on UCF50 and OlympicSports due to better objects tracking performance on more simple scenarios. Furthermore, because of limited amount of data, Hollywood2, HMDB, UCF, and OlympicSports mainly involve in unsupervised models, and the authors prefer to collect data in some fixed scenes by themselves, so the situation affects the performance because of the diversity of data sets lost.

There still have other papers generating trajectory data by themselves. For example, Reference [57] obtains trajectory data from vehicles and animals by mounted GPS device. Reference [6] applies Reference [5] to track pedestrians in the surveillance videos and generate trajectory data; Reference [41] employs Reference [119] to generate trajectory date as well. Furthermore, some papers take videos as the data set like UCSD, such as References [56] and [139].

3 UNSUPERVISED ALGORITHMS OF TRAJECTORY CLASSIFICATION

In recent years, more and more algorithms are required to analyze trajectory data in real time without any prior knowledge. Thus, no supervision is involved and it is called *clustering methods* in this article. For example, in a big shopping mall, abnormal behaviors are required to be recognized in real time, but few prior abnormal trajectories are recorded in surveillance. Therefore, unsupervised algorithms are invented to implement trajectory clustering. Unsupervised algorithms infer a function to describe internal relationships between unlabeled data. Clustering is the method to draw this hidden structure, and some models relating to trajectory clustering are reviewed such as Densely Clustering models, Hierarchical Clustering models, and Spectral Clustering models.

3.1 Densely Clustering Models

The Densely Clustering models measure how closely points are packed together after given the centroids. Inspired by this idea, Density-based spatial clustering of applications with noise (DBSCAN), which has been widely applied to trajectory clustering is proposed in Reference [37], shown in Figure 6. In DBCSAN, as shown in Figure 5 point p is chosen as the core point and distance threshold ϵ is given in advance. The points inside circle of which the radius is ϵ and the center is p are called *directly reachable* to p. Furthermore, points $\{Tr_1, Tr_2, \ldots, Tr_n\}$ are *reachable* to p if there is a path that Tr_i is *directly reachable* to p and each Tr_{i+1} is *directly reachable* to Tr_i [57, 133]. Other points are the outliers. Thus, the distance metric and the core parts selection are important. For solving the problem that DBSCAN cannot cluster the trajectories with large differences in densities [62, 72, 73], all trajectories are partitioned and substituted by sub-trajectories, then sub-trajectories are clustered and all clusters are grouped at the last step. However, different from measuring distance by Euclidean distance in Reference [73], the distance is measured by a combination of angle distance, center distance, parallel distance with equal weight in References [62] and [72]. The core trajectories are computed from the clusters and used for classifying new coming trajectory in References [31], [72], [162], and [163], e.g., all trajectories points belonging to the same cluster are averaged as a new point at each time, and all averaged points form the representations of clusters [72]. A similar model [118] proposes to find *traveling buddy* by computing density reachable and density connection between person and business places, which is similar to *directly reachable* and *reachable* definition in DBSCAN. In an adaptive multi-kernel-based method, shrunk clusters represent all groups by considering the attributes including positions, speeds and points, which retain much more discriminative messages in Reference [144].

Besides DBSCAN, there are other models belonging to Densely Clustering models cluster trajectory data. K-means clusters trajectories by searching centroids of clusters repeatedly [39, 43, 55, 85, 89, 116]. However, K-means is sensitive to the noisy data, because a small number of the outlier trajectory data can substantially influence the representations of clusters. Due to that reason and issues such as data imprecision and complexity of large data sets, a trajectory may belong to multiple clusters. Therefore, for improving the performance of K-means, Expectation Maximization (EM) algorithm is implemented to solve optimization problem iteratively [64, 166], because EM allocates a probability of the new trajectory assigning to each assignment. Fuzzy C-Means (FCM)

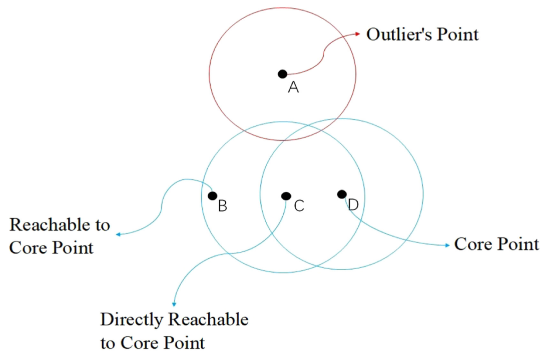


Fig. 5. DBSCAN.

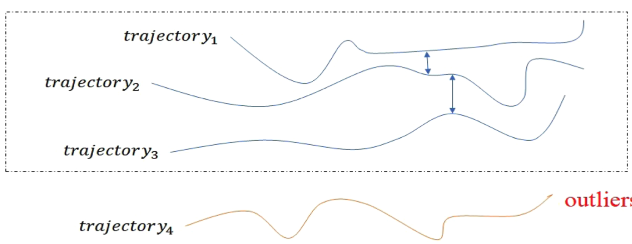


Fig. 6. DBSCAN for trajectory clustering.

algorithm employs parameters to measure the level of cluster fuzziness for each trajectory, called fuzzier. The algorithm searches correct direction in each iteration for cluster trajectories [96, 97, 111]. Furthermore, the algorithm assigns each element of trajectory a cluster centre, and this algorithm reduces intra-cluster variance to improve the trajectory clustering performance. FCM is similar to K-means that they all intend to minimize the optimization function iteratively, and the minimum they got is the local minimum. However, different from K-means, FCM assigns a membership value to each element of trajectory. In other papers, they propose to find flock patterns of trajectories [48], which are m trajectory segments inside a flock of I time interval and r radius called *flock* [47]. Reference [125] proposes to find disk p, the flock at arbitrary time $\triangle t_p$, in grid-based index. Furthermore, there should be enough points of different trajectories have been assigned to same disk. Reference [127] aims at tracking animals' movement trend by mining animals' spatial information. Similar to flock patterns, Reference [9] proposes to cluster trajectories recorded at different time together by applying spatial information. In detail, $trajectory_i$ and $trajectory_j$ are matched if they contain sub-trajectories $trajectory_i'$ and $trajectory_j'$, which are close enough. Table 2 shows the application field and the corresponding methods in Densely Clustering models.

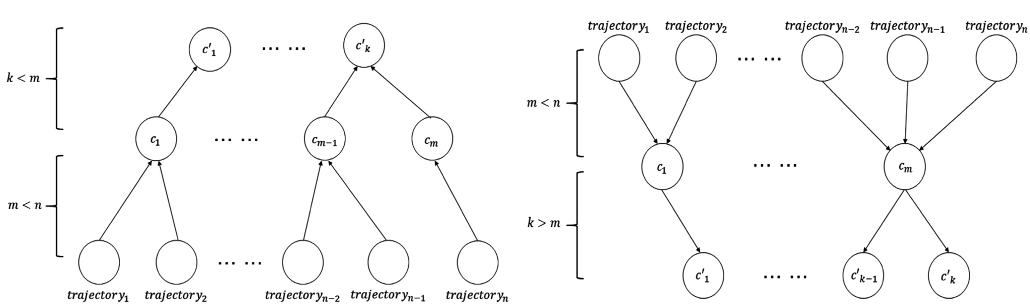
3.2 Hierarchical Clustering Models

Hierarchical Clustering models help to understand trajectory by multiple features, so this tree-type construction is proper to implement. Hierarchical Clustering models generally fall into two clustering types, Agglomerative and Divisive. As shown in Figure 7, two hierarchical types are also known as "bottom-up" and "top-down" approaches.

In Agglomerative frameworks, trajectories are grouped and the similar clusters are merged by searching their common properties. Optimal classifications are obtained by repeating representation computation and clusters merging until meeting the requirements. Inspired by this idea,

Table 2. Trajectory Data Clustering Application and Corresponding Works by Densely Clustering Models

Application problem	Method	Data preparation	Experiment data set
Discovery of groups of objects	[57], [125]	no	vehicles' GPS recording data set[57], four real datasets: *Truck, Buses, Cars, Caribous*[125]
	[72], [73]	partitioning trajectories as line segments and grouping the closing segments	synthetic data set
Extraction and categorization of moving objects	[144], [39]	no	the traffic data set[144], synthetic data set[39]
	[55]	detecting the foreground objects, tracking pixels of objects and generating trajectories	synthetic data set
Discovery of social events	[89]	detecting the foreground objects, tracking pixels of objects and generating trajectories	synthetic data set
Action recognition	[133]	no	Hollywood2, HMDB51, OlympicSport, UCF50
Characterization of regions or trajectories	[43], [97]	no	synthetic data set



(a) Agglomerative clustering (b) Divisive clustering

Fig. 7. Hierarchical clustering models.

Agglomerative clustering models were explored in Reference [164] to mine the locations that users are interested, Hypertext Induced Topic Search (HITS) model is proposed to achieve this goal and movement tracks of users are recorded as trajectories. Top n interesting trajectory clusters are obtained iteratively and the most popular locations are generated.

Different from Agglomerative, Divisive frameworks cluster trajectory data into groups and split them recursively to reach the requirements. For example, trajectory data are characterized by direction feature and clustered by Dominant-set embedded Bhattacharyya distance in initial clustering stage [76]. Furthermore, trajectories are split further except for the ones holding similar positions in each cluster. Because of the good performance of iterative models, Test-and-Divide (TAD) model is proposed [161]. It is a Divisive framework detecting all the closed trajectories firstly and splitting them recursively. More attributes of trajectory points are considered to improve the performance in [139]. For instance, trajectory $Trajectory_i = \{Tr_1, Tr_2, \ldots, Tr_N\}$, where $Tr_i = \langle x_n^i, y_n^i, \beta_n^i \rangle$. It is composed of two-dimensional position and an additional attribute β such as velocity or object size. In the coarse clustering step, the distance measurement between $Trajectory_i$ and its nearest observation $Trajectory_i$ are shown as follows:

$$f(Trajectory_i, Trajectory_j) = \frac{1}{N_i} \sum_{Tr_n^i \in Trajectory_i} ||(x_n^i - x_{\psi(n)}^j, y_n^i - y_{\psi(n)}^j + \gamma d(\beta_n^i, \beta_{\psi(n)}^j))||, \quad (13)$$

where $\psi(n) = \arg\min_{Tr_m^j \in Trajectory_j} ||(x_n^i - x_m^j, y_n^i - y_m^j)||$ and the minimum distance value is counted as the distance between $Trajectory_i$ and $Trajectory_j$. N is the total number of points belonging to $Trajectory_i$, $d(\beta_n^i, \beta_{\psi(n)}^j)$ indicates the dissimilarity of $Trajectory_i$ and $Trajectory_j$, and γ is weight parameter. In the fine-clustering stage, the model aims at distinguishing distortions by considering directed similarity $S_{i\to j}$ and confidence $C_{i\to j}$,

$$S_{i\to j} = \frac{\sum_{Tr_n^i \in Trajectory_i} c(Tr_n^i, Tr_{\psi(n)}^j) s(Tr_n^i, Tr_{\psi(n)}^j)}{\sum_{Tr_n^i \in Trajectory_i} c(Tr_n^i, Tr_{\psi(n)}^j)}, \quad (14)$$

$$C_{i\to j} = \frac{\sum_{Tr_n^i \in Trajectory_i} c(Tr_n^i, Tr_{\psi(n)}^j)^2}{\sum_{a_i \in A} c(Tr_n^i, Tr_{\psi(n)}^j)}, \quad (15)$$

where $c(Tr_n^i, Tr_{\psi(n)}^j) = \exp(\frac{-||(x_n^i - x_{\psi(n)}^j, y_n^i - y_{\psi(n)}^j)||}{\sigma_1})$ and $s(Tr_n^i, Tr_{\psi(n)}^j) = \frac{\exp(-d(\beta_n^i, \beta_{\psi(n)}^j))}{\sigma_2}$.

Furthermore, a similar hierarchical framework is explored to group videos by constructing the trajectories of video [42] as an unordered tree, and a kernel method recognizes videos by clustering the trees. In addition, Hierarchical Clustering models also recognize actions from video in References [104] and [117]. For two trajectories in video, $Trajectory_i$ and $Trajectory_j$, the distance is computed as follows:

$$d(Trajectory_i, Trajectory_j) = \max_{t \in [\tau_1, \tau_2]} d_{spatial}[t] \cdot \frac{1}{\tau_2 - \tau_1} \sum_{t=\tau_1}^{\tau_2} d_{velocity}[t], \quad (16)$$

where $d_{spatial}[t]$ is the positional distance at time stamp t, and $d_{velocity}[t]$ is the similarity measurement relative to velocity. An affinity matrix $w(Trajectory_i, Trajectory_j) = exp(-d(Trajectory_i, Trajectory_j))$ is calculated and trajectories are clustered by greedy agglomerative hierarchical models [104, 117]. The clusters are overlapped because of similar parts, so every trajectory is weighted and optimized to classify in Reference [92]. Since one motion object may generates several trajectories, it is critical to employ as much features as possible to ensure object recognition, and a multi-layer classifier is invented in References [6, 76]. Table 3 shows trajectory data application and corresponding works by Hierarchical clustering models.

3.3 Spectral Clustering Models

Trajectory data can be represented as a matrix called affinity matrix, and the relationships between them are extracted as the elements of matrix. The top K eigenvectors form clusters with distinctive gaps between them, which can be readily used to separate data into different groups [143]. In addition, affinity matrix characterizes videos [121] and represents the relationships. In Reference

Table 3. Trajectory Data Clustering Application and Corresponding Works by Hierarchical Clustering Models

Application problem	Method	Data preparation	Experiment data set
Characterization of places or regions	[164]	stay points computation	GPS collected data set
	[139]	a blob tracker recording data from videos	visual surveillance data set
Extraction and categorization of moving objects	[76]	trajectories smoothing	vehicles' GPS collected data set
	[42]	tracking pixels and generating trajectories	High Five, Hollywood2, OlympicSport, HMDB
Discovery of social events	[161]	closed-crowd checking	vehicles' GPS collected data set
Action recognition	[92], [104], [117]	tracking pixels and generating trajectories	{Hollywood2, HMDB51, OlymicSports}[92], {Hollywood1, UCF-Sports}[104], {LFR graph benchmark, club networks, football networks, political books network}[117]
Discovery of relationships or interaction	[6]	computing length-based clustering	pedestrians trajectories obtained by tracker system[5]

[56], affinity matrix A is constructed as follows:

$$A_{ij} = \exp\left[\frac{-\bar{d_{ij}}}{2\sigma^2}\right], \tag{17}$$

where $\bar{d_{ij}} = \frac{1}{N}\sum_{n=1}^{N}||x_n^i - x_n^j||$, and x_n^i indicates the nth point of $trajectory_i$. Considering different lengths of trajectories, some novel models are explored to construct affinity matrix [16, 17], and it is constructed as

$$A_{ij} = \begin{cases} e^{(-\frac{1}{\sigma_i\sigma_j}||Tr_n^i - Tr_m^j||^2)}, \ for\ i \neq j, \\ 0,\ otherwise, \end{cases} \tag{18}$$

where Tr_n^i and Tr_m^j are points, and σ_i and σ_j indicate scale invariance, which is computed by the median of the l nearest neighbors. To increase the separation of points belonging to different groups, SVD decomposition is used to construct the affinity matrix [70]. In addition, a novel distance method is explored to compute trajectories i and j [7], so spatial distinction can be considered.

$$s(i,j) = e^{-\frac{1}{2}h_\alpha(i,j)h_\alpha(j,i)/(\sigma_i\sigma_j)}, \tag{19}$$

$$h_\alpha(i,j) = ord^{\alpha}_{Tr_n^i \in Trajectory_i}\left(\min_{Tr_m^j \in N(C(Tr_n^i))} d(Tr_n^i, Tr_m^j)\right), \tag{20}$$

where $h_\alpha(i,j)$ is the directed Hausdorff distance, $ord^{\alpha}_{Tr_n^i \in Trajectory_i} f(Tr_n^i)$ indicates the value of $f(Tr_n^i)$, and $N(C(Tr_n^i))$ denotes the subset of points, which are the ones matching to the point Tr_n^i in trajectory $Trajectory_i$.

For clustering high dimensional trajectory data by Spectral Clustering models, several novel methods are explored in References [23], [52], and [158]. For example, a new similarity metric captures causal relationships between time series in Reference [52] and a mixture of affinity subspaces is applied to approximate trajectory in Reference [23]. Trajectory data are represented by considering covariance features of trajectories in Reference [36], so it avoids considering different lengths of trajectory data. Spectral clustering works with multiple-instance learning frameworks to achieve human action recognition in Reference [148].

Spectral Clustering models are derived from Graph Theory in which an undirected graph represents the relationships and constructs a symmetric adjacency matrix presenting them [15]. By constructing a graph, both explicit and implicit intentions inside trajectory data are mined [25]. The graph is cut into sub-graphs to classify trajectories, and each sub-graph represents its own cluster [78, 155]. Hierarchical layers search sub-clusters in each cluster by treating trajectories points as graph nodes and this procedure is called *Hierarchical graph partitioning* [49]. For considering more variables, a novel measurement function composed of the entropy rate of a random walk on a graph is presented in Reference [80]. From the idea that an undirected graph can be represented as an adjacent matrix, a directed graph also can be involved [75]. Trajectory Binary Partition Tree (BPT) represents video in Reference [95] by representing trajectories as nodes so the edges indicate relationships between a pair of trajectories, and graph cut method groups trajectory data. Because of the robustness of composite feature descriptors, the descriptors including Speeded Up Robust Features (SURF) and Maximally Stable Extremal Regions (MSER) are employed in Reference [78]. An object creates several trajectories if different parts of the object are tracked, so a model is invented to describe trajectories by feature patches [81]. The edges are computed by geometric distance and appearance distance. Hausdorff distance is utilized to measure the similarities and set as weights of edges in Reference [60]. Since the great performance of PageRank, it is used to score the edges as well [28]. Table 4 shows trajectory data application and corresponding works by Spectral clustering models.

3.4 Discussion

Densely Clustering models classify trajectories by distance metrics mostly, which may result in classifying trajectory data by spatial information. Hierarchical Clustering models consider more attributes in each level and group trajectories by different categories, so it is a suitable algorithm when different clustering criteria are considered. Spectral Clustering models compute internal relationships by analyzing the affinity matrix and group the trajectory data by the internal relationship.

As a state-of-the-art method in densely clustering algorithm, DBSCAN has a time complexity of $O(n \log n)$. It is a fast and efficient algorithm to implement when only spatial information is involved. However, with the development of location recording device, more information is recorded and required to contribute trajectory data analysis. Thus, Hierarchical Clustering model fix this problem, but this operation costs much more time in computation with $O(2^n)$ and $O(n^2)$ time complexity for divisive clustering and agglomerative clustering, respectively. Therefore, the algorithm is not a proper method when a huge number of trajectory data involve in. Spectral Clustering models reach the time complexity of $O(n)$ by processing all trajectory data together. However, [61] mentions that Spectral Clustering models have their own limitation that they are well defined only for the non-negative affinities between trajectories. Furthermore, that trajectory lengths are required to be unified is another issue of applying Spectral Clustering models.

For the performance, Hierarchical Clustering Densely Clustering models outperform Densely Clustering models. For example, Reference [133] reaches 57.2% and 91% accuracy experimenting for HMDB and OlymicSport data set, respectively. Reference [104] proves that the performance

Table 4. Trajectory Data Clustering Application and Corresponding Works by Spectral Clustering Models

Application problem	Method	Data preparation	Experiment data set
Extraction and categorization of moving objects	[56], [81], [143]	tracking pixels and generating trajectories	{collected crowd traffic surveillance videos}[56], {LHI, PETS, I-80}[81], {collected varying surveillance videos}[143]
	[155]	objects classified firstly, then clustering trajectories in each objects categories	collected traffic surveillance videos
Action recognition	[25], [36]	no	{UCF Sports, HMDB 51, MSR-II}[25], {UCSD anomaly detection, MIT Parking Lot}[36]
	[121]	tracking dense trajectories[130]	UCF Sports, MICC-SOCACT4, Volleyball Activity Dataset 2014
	[148]	affine motion model employed to extract trajectories	UCF Sports, Hollywood
Objects detection and recognition	[16], [70]	tracking pixels and generating trajectories	Hopkins 155 data set
Location prediction and objects clustering	[52]	no	synthetic data set
Discovery of hot place	[49]	extracting representing points in pre-defined circle to improve GPS measurement	collected truck trajectory data
Discovery of hot route	[75]	no	network-based data generator[14]
Discovery of hot events	[60]	no	collected videos

could be improved if more features are considered and reaches 65.5% and 92.3% accuracy. With more features involved, it costs more computation resource and not properly for a large amount of experiment data in reality such as objects movement pattern analysis in crowd scene. Thus, Spectral Clustering models are properly used for simplex application scenario. For example, References [121] and [148] are better than Reference [104]. However, Spectral Clustering models fall behind Hierarchical Clustering models on complex scenarios, such as the Hollywood and HMDB data sets. It is the case that Spectral Clustering models aim to project data into other space or extract the representation in sub-space and only primary component is extracted, so not all information is taken into account.

4 SUPERVISED ALGORITHMS OF TRAJECTORY CLASSIFICATION

Supervised algorithms aim at learning a function that determines the labels of testing data after analyzing labeled training data. Therefore, supervised algorithms outperform others and the supervised ones could save much more computation resource on classification part. In some supervised algorithms, trajectory data are classified by unsupervised algorithms and the representations

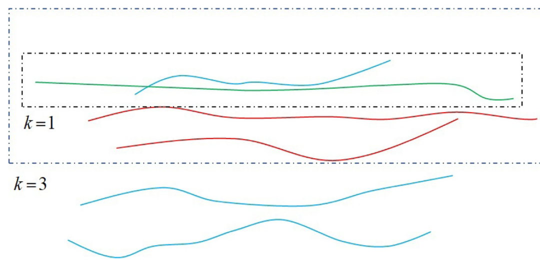


Fig. 8. k-NN for trajectory classification. Inquiry trajectory is the green one, are the labeled data are the red and the blue ones, which means two groups.

of clusters are obtained to classify new inquiry trajectories. For example, in Densely Clustering models, the representations can be computed from the grouped training trajectory data and new coming trajectories are classified quickly in References [10] and [96]. Trajectory data are classified and organized in a tree-construction and new coming trajectories are grouped by searching the tree in References [50] and [98].

4.1 Nearest-neighbor Algorithms

Nearest-neighbor algorithms, such as k-Nearest-neighbor algorithm (k-NN), are finding a voting system to determine the category of a new coming entity and all data are kept in the same feature space. In trajectory classification, the distances from an inquiry trajectory to all labeled training trajectory data are computed, and the label of the inquiry trajectory is voted by its k nearest neighbors. For example, as shown in Figure 8, the inquiry trajectory is assigned as blue one if $k = 1$ and assigned as red one if $k = 3$.

In the implementation, it is important to choose a suitable distance metric according to the scenario, occlusion, trajectory data format, and feature types. Therefore, trajectory data are represented by MBR and classified by k-NN in Reference [45]. It avoids occlusion and increasing separability of inter-object. Tree construction is employed in some papers to help searching, and training data set is also used as the representations of groups to improve the searching rate of k-NN. The spatiotemporal information of trajectory data are represented in Riemannian manifold [33] and matched to a shape space, then a function $F(X)$ is trained by training set $\{(X_i, y_i)\}$, where y_i is the labels, so the inquiry action can be recognized by k-NN classifier. A training set is also employed to recognize the action by computing the Maximum Cosine Similarity between training set and test set as the measurement in k-NN, $\hat{u} = arg\max_{\vec{u_i} \in U} \vec{Tr} \cdot \vec{u_i}$. Furthermore, for accessing k-NN faster, fast nearest-neighbor (fastNN) algorithm organizes trajectory data in an Octree to improve searching [102]. With the increasing inquiry trajectories, the trends of trajectory data in a fixed period are required instead of general representation, so a circumstance that dynamically searches the nearest neighbors in a fixed period or the ones belonging to some specific types are considered in Reference [41]. Support Vector Machine (SVM) is trained to obtain the optimal parameters and hyper-volume, and the inquiry trajectory is determined as outliers if it falls outside the hyper-volume [100]. For more complicated scenarios, such as detecting social groups in crowds, Structural Support Vector Machine (Structural SVM) is explored in Reference [114]. *Temporal Causality* is added to measure what pedestrians are mutually affecting the others motion paths, which means the affection between two objects are computed and added as features, one of *Social Identities* [114]. Spatial information is also employed to measure *physical distance*, *shape similarity* and *group activities*. Trajectories are represented as a graph that the points of trajectory data are set as the nodes, and SVM is trained to classify trajectories in Reference [113]. However,

Table 5. Trajectory Data Classification Application and Corresponding Works by Nearest-neighbor Algorithms

Application problem	Method	Data preparation	Experiment data set
Action recognition	[33]	no	MSR Action 3D data set, Florence 3D Action data set, UT Kinect, UCF-Kinect
	[102]	the points of trajectories are downsampled to achieve time duration invariance	2D Graffiti, Kinect, 6DMG
Extraction and categorization of moving objects	[41]	no	A fleet of trucks and a fleet of school buses real trajectory data set [119]
Discovery of social events	[100]	trajectories subsampling to a fixed dimensions	synthetic data
	[113]	only *space-time* interest points are extracted from video clips to generate trajectories	UCSDped1, UCSDped2, UMN
Discovery of social groups in crowds	[114]	time window is applied to obtain trajectories in fixed length	MPT-20x100, GVEII

spatiotemporal interest points should be extracted by computing matrix μ, which is composed of first-order spatial and temporal derivations and Gaussian weighting function. Table 5 shows trajectory data application and corresponding works by Nearest-neighbor Algorithms.

4.2 Statistical Models

Statistical models exploit a set of probability distributions to represent the data generating process such as Gaussian Mixture model (GMM) and Bayesian inference. GMM usually combines with EM algorithm to train each component, and Bayesian inference obtains a set of probability functions that determine the categories of inquiry trajectory data. Bayes' theorem is critical for Bayesian inference and written as $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$, where A and B indicate two events in event space.

GMM aims to describe the sample from $\{x_1, x_2, \ldots, x_n\}$ in a component of GMM as

$$P(x_j) = \sum_{i=1}^{K} \pi_i N(x_j; \mu_i, \Sigma_i), \tag{21}$$

where $N(x_j; \mu_i, \Sigma_i)$ is the probability density of the ith component belonging to a component with mean μ_i and variance Σ_i. π_i is the weight with a constraint that $\sum_{i=1}^{K} \pi_i = 1$, and they can be computed according to event frequency. Generally, EM algorithm iteratively optimizes the parameters of GMM, but Maximum Likelihood algorithm is implemented instead of EM if labeled trajectory data are available in training stage. For example, video events are treated as a linear combination of a set of event patterns, and two probabilistic terms are proposed to characterize video events in Reference [154]. Furthermore, the abnormal patterns are scored by summarizing the probabilities of trajectory data of the corresponding video. GMM models the variance caused by the environmental factors and embedded into DTW to recognize gestures [12].

Table 6. Trajectory Data Clustering Application and Corresponding Works by Statistical Models

Application problem	Method	Data preparation	Experiment data set
Discovery of social events	[132]	no	Edinburgh data set, the MIT Carpark data set, New York Central terminal data set
	[154]	tracking pixels and generating trajectory data	UCSDped1, Avenue data set, Subway surveillance
Action recognition	[12]	no	ADHD behavioral pattern data set
	[32]	continues trajectories segmented into sub-sequences to provide a deeper analysis	MSRC-12, Cornell Activity dataset 120, Multi-modal Action data set, Online RGB-D data set
Extraction and categorization of moving objects	[54]	segmenting trajectories to characterize time-varying information	synthetic trajectory data set, the hand sign data set, the vehicle motion trajectory data set

Bayesian inference classifies new coming data, and the classified ones update the probability functions of Bayesian inference. For samples $\{x_1, x_2, \ldots, x_n\}$, the probability of the corresponding labels $y_{1:n}$ is $p(y_{1:n}|x_{1:n})$. Derived from Markov Chain Monte Carlo (MCMC) algorithm, the distribution of variables can be approximated by a joint distribution, so Gibbs sampling is used to approximate $p(y_{1:n}|x_{1:n})$ by sampling $p(y_i|y_{-i}, x_{1:n})$ iteratively. According to Bayes' theorem, $p(y_i|y_{-i}, x_{1:n})$ is represented as $p(y_i|y_{-i}, x_{1:n}) \propto p(x_i|y_i)p(y_i|y_{-i})$ where $p(x_i|y_i)$ is the likelihood and $p(y_i|y_{-i})$ is the marginal distribution. In the Dirichlet Process (DP) model, which is one of the Bayesian inference frameworks, $p(y_i|y_{-i})$ is formulated as $p(y_i|y_{-i}) \propto \alpha G_0(y_i) + \sum_{j \in -i} \delta(y_i - y_j)$ where α is scale parameter and G_0 is base measure in sampling space. The clusters can be parameterized for classifying new inquiry data, e.g., Dirichlet Process Mixture model (DPMM) is used to represent all m categories with parameterized indexes $\{\Theta_1, \Theta_2, \ldots, \Theta_m\}$ in [54]. Finally, the new inquiry trajectory is classified by a trained DPMM as $p(\Theta_k|R) \propto p(R|\Theta_k)p(\Theta_k)$ where $p(R|\Theta_k)$ is the likelihood and $p(\Theta_k)$ is the prior probability. To learn coupled spatial and temporal patterns, Hierarchical Dirichlet Process (HDP) algorithm is applied in [132]. Bayesian model is used to segment objects by classifying trajectories, so that human motion is also detected [32]. Table 6 shows trajectory data application and corresponding works by Statistical Models.

4.3 Neural Network

Neural network is an artificial system simulating the biological neural network in animal brains. The network is constructed by a number of mutually connected neurons, and each neuron is represented as a real number. Neural networks can represent data like the deep generative model. It is trained to represent multivariate time series if trajectory data are generated as a vector [152], and it replaces the factor analyzer's inner product operation with convolution so temporal dependency is involved in the model, shown in Equation (22). Then, a generative model is proposed to generate origin data from latent variables independently and repeatedly.

$$\mathbf{Tr} = \mathbf{A} * \mathbf{x} + \mathbf{b} + \epsilon, \tag{22}$$

where $\mathbf{A}, \mathbf{b}, \epsilon$ are parameters need to be optimized and $\mathbf{x}$ is latent variables. A deep fully connected Neural Network with weight decay and sparsity constraint transfers trajectory data from different

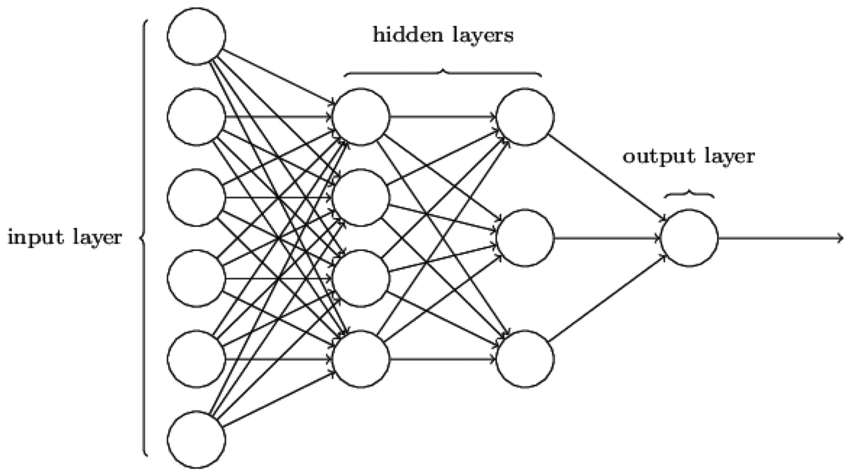


Fig. 9. CNN construction, input layer, hidden layers, and output layer.

viewpoints to a fixed viewpoint in compact representation [103]. The same action acted by same person from different viewpoints is recorded as a sequence of trajectory data, which are 108 view angles in the article. A deep fully connected network is used to train the model, called Robust Non-linear Knowledge Transfer Model (R-NKTM).

In most cases, Neural Network is used to classify data. It can be viewed as a mathematical function $f : X \rightarrow Y$, where X is the observation and Y indicates the corresponding label. For example, Convolutional Neural Network (CNN or ConvNet) consists of multiple layers including convolutional, pooling and fully connected layers. That layout tolerates the variations of the input data, avoids overfitting problem, and distinguishes data as similar as Multilayer perceptron (MLP). CNN has been proved efficient in classification issues of computer vision. As Figure 9 shows, CNN comprises input layer, hidden layers, and output layer. CNN is employed for trajectory classification in References [27] and [141]. In Reference [141], frame-based, object-based, and scene-based features are extracted, respectively, and they fuse together to complete video categorization. Furthermore, the semantic relationship between video and objects/scenes can be discovered as well. In fusion phase, the averaged representations of three feature sequences are computed separately, and fed into the first layer of fusion network. The score of video belonging to explicit label is the output of network. CNN also ranks segmentation by classifying dense trajectories of objects [40]. With the moving objects detector, the object in foreground and background are extracted. Then, the trajectories of objects in spatiotemporal tubes are classified, so the object can be scored as 0 or 1, which mean foreground or background in the article. By this method, multiscale trajectory classification solves the segmentation ambiguities caused by motion variations of objects. Reference [138] treats the dissimilarity of coordinates between two arbitrary frames as skeleton trajectory and projects the trajectories to three orthogonal planes. The projection method is called Joint Trajectory Maps (JTM). It is distinct to save the spatial information, and the motion difference and temporal evolution such as direction and speed are encoded as well. Other than CNN, Recurrent Neural Network (RNN) is another artificial neural network that can access memory of network. It is suitable to process temporal dynamic behaviors. However, the gradients are forgotten in back-propagation of RNN, so Long Short-term Memory (LSTM) is proposed to fix the problem. More complicated networks are involved in Reference [34], which applies CNN as a deep hierarchical feature extractor. It helps to capture complex trajectories. With LSTM, trajectories are transferred to *sentence* *mappings*, so large-scale images are reduced dimensions and represented. For further application, Reference [63] is the method employing multi-layer LSTM model to classify the vehicles surround ego-vehicle and predict the movements. Instead of fusing the scores of RNNs based on different projections, Reference [2] proposes a “social-pooling” layer representing the state by fusing the hidden states of neighbors’ LSTM, this framework considers the influence between persons and a bivariate Gaussian distribution is involved to describe the future positions. With training data, the loss function based on hidden state is minimized and the parameters of the bivariate Gaussian distribution are trained. Thus, Reference [2] considers the influences from neighbors and gives a proper function to simulate this situation, that is a state-of-the-art model on some data set.

Deep learning models not only classify trajectory data, but they also learn to construct a feature map of trajectory data or describe trajectory data for improving the performance of classification. For example, Reference [30] proposes to represent trajectory data as Gaussian Mixture Model (GMM) in unsupervised manner, and a fully-connected deep network is trained to represent trajectory. A flexible deep CNN called Deep Event Network (DevNet) is trained by ImageNet data set, and the trained DevNet is tuned to extract generic image-level features of trajectory data in Reference [44]. To figure out the differences between image classification and multimedia event detection, DevNet fine tunes parameters by a specific data set, and backward passing is employed to identify pixels in consecutive frames to recount events. Deep Neural Network (DNN) is another Neural Network that learns a more compact and powerful representation of trajectories [19, 51]. Especially in Reference [19], DNN not only be employed to represent body motion, but it can also predict the future body motion, which presents in skeletal trajectory data. In detail, each limb is connected to their own nodes in next layer until only one node is left in this tree construction. For storing the structural relationship between trajectories, a novel Deep Trajectory Descriptor (DTD) is proposed to describe trajectories. DTD is the projection of dense trajectories on a canvas, called *texture image*. In this method, the relative motions are characterized effectively [109]. Video contains diversity semantic information, and all information can be presented in different features. Thus, the relationships between multiple features including spatiotemporal features, audio features and inter-class relationship are discovered to classify videos in References [59] and [142]. Furthermore, local features such as HOG or HOF are properly to discover salient and informative regions in each frame of video data, but they lack the capacity of visual representation and not discriminative for recognizing action. Thus, Reference [112] proposes a two-stream ConvNets, which is composed by spatial nets and temporal nets; the first nets aim to understand discriminative appearance by taking each frame as input and the second nets capture the effective motion features by using multi-frame optical flow. This framework helps to reach or even outperform the local feature descriptors, which are not properly to apply on action recognition. To obtain the in-depth investigation, Reference [147] also conducts extensive experiments with different options, and it summarizes that combing predictions from spatial and temporal streams is useful. It is important and efficient for trajectories classification. Furthermore, Reference [136] proposes a method to combine two-stream ConvNets and improved dense trajectories to reach the goal. In detail, multi-scale convolutional feature maps can be learnt by two-stream ConvNets, and it is used as trained generic feature extractor. In addition, a homography matrix is estimated to help compute dense trajectories by taking camera motion into account, so improved dense trajectories are obtained. However, Reference [137] tries to explain that two-stream ConvNets still not get a state-of-the-art level, because it is relatively shallow and small training data involved. Other than two-stream ConvNets, Reference [108] proposes a three-stream framework called sequential Deep Trajectory Descriptor (sDTD), including CNN, LSTM, and a novel trajectory feature descriptor, which aims to project trajectories to text images, so this two-dimensional construction save computation resource in processing. However, two-stream or three-stream framework still has their limitation such as disregarding the long-term content of the video, so the model is confused when a complex scene involved. Thus, Reference [46] proposes a novel way to represent video across spatiotemporal space, and anchor points are going to represent trajectories by iterative training. This operation describes the whole trajectories more accuracy than max-pooling or average representation. Furthermore, this article is important that it gives a powerful result, the highest layer of convolutional layer is better to aggregate appearance and motion features together.

Self-Organizing Map (SOM) is an algorithm to produce low dimensional data from the high dimensions, which is efficient for trajectory clustering. Furthermore, SOM also makes it possible that creating low-dimensional views of high-dimensional data. Thus, SOM learns the similarities between trajectories in a two-dimensional grid and each element of the grid indicates a specific prototype in References [90] and [106]. Reference [90] aims at projecting trajectory data into a low-dimensional manifold and the parameters of the manifold are used as feature of trajectories, so the accuracy of classification and recognition results have been improved. Reference [106] visualizes trajectory data, and the classification can be monitored at arbitrary time. In training step, each training trajectory is trying to find the most suitable prototype in network and adjusts the neighbors of the matched one accordingly. Table 7 shows trajectory data application and corresponding works by Neural Network.

4.4 Discussion

Nearest-neighbor algorithms only consider the spatial relationship between a pair of trajectory data but ignores local characters. Statistical models make up for this imperfection by combining them in a mixture model or inferring the relationships in Bayesian models. Based on this procedure, labels can be obtained from data by generative process, and the model can also be optimized by inference process, so the whole model should be trained in iterative steps. In some circumstances that a huge number of data emerged, the samples have highly correlation. For example, they have similar motion tracks, the same start and finish point, or the same operator who produces trajectories. Therefore, the training step helps the model to recognize the different characteristics of data, and large amounts of data are required. Neural Network considers the differences of trajectory data and requires a huge number of data to improve itself.

The time complexity of Nearest-neighbor algorithms could reach $O(ndk)$ where d is the dimension of trajectory and k indicates iteration times. Although it reaches a considerable time complexity, the algorithm only uses spatial information. It is efficient to analysis trajectory data with coordinates or geographical information. For Neural Network, the training step takes a huge time complexity of $O(n^3)$; time complexity has been saved and reaches $O(n)$ in the classification step, which is an effect and state-of-the-art algorithm. Furthermore, Neural Network has been used in a lot of application scenarios and helps to reach the state-of-the-art performance. Although the supervised methods obtain the classifiers by observing a number of training data, overfitting problem may happen when the model overreacts training data.

Reference [113] reaches a state-of-the-art performance, 97.14% accuracy for abnormal detection, which is 93.7% accuracy reached by Reference [154]. However, as we can see from the tables above, Nearest-neighbor algorithms need to unify the lengths of trajectories in most papers, and Statistical models do not have to do this process. For Statistical models, each trajectory should be assumed not to be related to the others, but in most circumstances, including hand signs, pedestrians walking in crowds, and traffic scenes, objects are influenced by the others. Neural Network has been experimented on a lot of data sets, including ActivityNet, Hollywood, and UCF data set. Furthermore, this model has been used for some applications, such as action recognition, discovery of social events, video categorization, object segmentation, and object movement categorization. Thus, it proves that this model is proper for wide application fields. Neural Network reaches

Table 7. Trajectory Data Clustering Application and Corresponding Works by Neural Network

Application problem	Method	Data preparation	Experiment data set
Extraction and categorization of moving objects	[90], [141], [152]	no	{hand signs data set from ASL UCI, CAVIAR data set}[90], {ActivityNet, FCVID}[141], {UCI machine learning repository, Motion Capture data from CMU }[152]
Action recognition	[27], [46],[109], [112], [136], [141]	no	{HDM05}[27], {KTH, UCF50}[109], {ActivityNet, FCVID}[141], {HMDB, UCF101}[46], [112], [136]
	[19]	convert trajectories into Cartesian coordinate system	CMU mocap data set
	[51]	locating motion regions, then tracking the moving persons and generating trajectories	KTH, UCF50, VIRAT, TRECVID
	[103]	A realistic 3D human body required to fit a real MoCap sequence, then trajectories in multiple views are generated	INRIA Xmas Motion Acquisition Sequence, UWA 3D Multiview Activity, northwestern-UCLA Multiview Action-3D, UCF Sports
	[108]	dense trajectories generated from videos	KTH, HMDB, UCF101
	[138]	the skeleton trajectories between consecutive frames are projected on three orthogonal planes	MSRC-12, G3D, UTD-MHAD
Objects segmentation	[40]	no	VSB 100, Moseg
Discovery of social events	[44]	no	NIST TRECVID 2014, Multimedia Event Detection data set
Video categorization	[59], [142]	no	Hollywood2, Columbia Consumer Videos (CCV), CCV+
Location prediction and objects clustering	[2], [63]	no	{ETH, UCY}[2], {KITTI}[63]

state-of-the-art level. For example, Reference [103] has 90% accuracy for UCF Sports data set and Reference [142] reaches the accuracy of 65.7% for Hollywood data set. However, overfitting problem may happens when training data have a lot of noisy data, and appropriate data set and features are required.

5 SEMI-SUPERVISED ALGORITHMS OF TRAJECTORY CLASSIFICATION

Semi-supervised algorithms fall between unsupervised algorithms and supervised algorithms. The algorithms make use of a small number of labeled data and continuous inquiry data to complete tasks. The model is trained by labeled data first, then inquiry data are kept sending to the trained model to make sure that it can be updated to outperform the previous models when the amount of training data are too small. Semi-supervised procedure needs only a small cost regard to human classification efforts. This procedure not only avoids the overfitting problem but also is more accurate than the unsupervised ones.

Therefore, some semi-supervised algorithms are invented from unsupervised or supervised algorithms. For example, trajectory data are classified first, and the new inquiry ones are clustered to update the classifier automatically [50, 71, 135, 154]. In Reference [50], spatiotemporal information and object ID are all included to describe trajectory data, and object traveling information including object ID, road ID, object moving direction and time interval are given as anonymized data. Thus, the trajectories in same time interval are grouped as sub-dataset. In each sub-dataset, trajectories are classified and the representations are computed, so the model will gives a recommendation route when a new trajectory of new object comes. Reference [71] proposes a parameter-light model to solve the problem that it difficult to optimize many parameters to reach a great performance. Non-Conformity Measure (NCM) measures the difference between the new inquiry trajectory and every training trajectory. Depending on application, other information can be included to implement the goal including objects' speed, direction, third spatial location or temporal information. Reference [135] aims to represent trajectory as matrix, and a blob tracker obtains trajectories of objects from video clip and each element of trajectory is consisted by four corners location of blob area at current frame. Thus, a trajectory is represented as a $N \times 8$ matrix where N is the length of trajectory. After given motion matrix, its low-rank approximation matrix can be computed by $\tilde{M} = CUR$, where M is motion matrix, C indicates the matrix in representative motion subspace. U is computed by minimizing $||M - CUR||_F^2$, and $R = C^\top M$. To evaluate the approximation, the error is computed, $SSE = ||M - \tilde{M}||_F$. Under subspace, trajectories are assigned as abnormal or normal depending to SSE value. Trajectory data of video are modeled as the combination of normal and abnormal patterns, and probabilistic terms characterize the patterns in Reference [154]. From this modeling, the terms can be updated by the detected inquiry trajectory. To detect abnormal trajectories faster in a complex scene, low-rank approximation is employed to describe trajectory data and the new detected abnormal ones update the threshold in Reference [135].

Inspired by Hierarchical Frameworks, trajectories and the categories are represented as a tree construction where children nodes indicate trajectories and roots denote the representations of categories in References [67], [74], and [99]. Reference [67] proposes to train a factorial Hidden Markov model (FHMM) classifying trajectory data. It costs less time in training phase compared to the Hidden Markov model, and this model is an algorithm for incremental and autonomous acquisition and learning of motion primitives. Reference [99] setups a tree construction to improve the speed of clustering, and the construction divides into two phase including tree building and tree maintenance. In the first phase, trajectory T is constructed as a vector of two-dimensional coordinates $Trajectory = \{Tr_1, \ldots, Tr_N\}$, where $Tr_j = \{x_j, y_j\}$. A representation of category is computed as $C_i = \{c_{i1}, \ldots, c_{im}\}$, where $c_{ij} = \{x_{ij}, y_{ij}, \sigma_{ij}^2\}$ and σ_{ij}^2 is an approximation of the local variance of category i at time j. The inquiry trajectory is assigned to the nearest category and the corresponding category should be updated by the new one. For the nearest category point $c = \{x, y, \sigma^2\}$ to the point of trajectory $t = \{\hat{x}, \hat{y}\}$, c is updated as follows:

$$\begin{cases} x = (1-\alpha)x + \alpha\hat{x}, \\ y = (1-\alpha)y + \alpha\hat{y}, \\ \sigma^2 = (1-\alpha)\sigma^2 + \alpha[dist(t_i, c_j)]^2, \end{cases} \tag{23}$$

where α is the update rate between 0 and 1. A new category is created if no categories close to the inquiry trajectory. The second phase includes merging, concatenation, and pruning: Merging happens when incoming trajectory has been classified; concatenation happens after merging and a node in the tree has a single child; and the nodes will be pruned when it has not been updated for a long time.

Considering the fact that Bayesian model is derived from Bayes' theorem, the parameters are optimized by sampling training data, and it is feasible to update the model by classified new inquiry data [54]. Furthermore, to add new trajectory data, the previous samples and the new ones are sampled by Gibbs Sampling as

$$p(\eta_i|\eta_{-i}, y_{1:N+\phi}) = p(\eta_i|\eta_{1:N} = W_{1:N}, \eta_{-i}^{new}, y_{1:N+\phi}), \tag{24}$$

where y is trajectory data, $\eta_{1:N}$ indicate the known states of the previous samples, and $N + 1 < i < N + \phi$. η_{-i}^{new} denote the states of new inquiry trajectory data except for the ith one. From Bayes' theorem, the classifying process is rewritten as $p(y_i|\eta_i)p(\eta_i|\eta_{1:N} = W_{1:N}, \eta_{-i}^{new})$. $p(y_i|\eta_i)$ is estimated by the previous samples and it is assumed to be Gaussian distribution. The only issue that needs to be fixed is carrying out Gibbs Sampling on $\eta_{N+1:N+\phi}$ to compute $p(\eta_i|\eta_{1:N} = W_{1:N}, \eta_{-i}^{new})$. Table 8 shows trajectory data application and corresponding works by semi-supervised algorithms.

Discussion

From the above, semi-supervised algorithms almost derived from supervised algorithms. In hierarchical frameworks [50, 99], they cost a huge resource when a great number of data are involved due to representing trajectory computation after classifying each new inquiry data coming and finding the properly category in iterative steps. For statistic models [54, 67, 74, 154], a huge number of training data are required and sampling procedure is faster than hierarchical models because of the whole trajectory involving inferring processing in statistical model instead of computing the distance of each point of trajectory in hierarchical models. Reference [54] reaches the accuracy of 85% for traffic data and the accuracy of 74% for hand signs data set, and this article makes the new coming trajectory classification more convenient and faster, because only two parameters will be updated after the previous ones classified. Furthermore, they make the labeling work more inexpensive. However, the algorithms have to be iterated several times to reach the acquisitive accuracy, which means validation step and correction step are recommended as well, and the trajectories independent assumption influences the performance. As a Neural network model, Reference [51] reaches the accuracy of 98% for KTH data set and 53.8% for UCF50 data set. It outperforms some models, including Convolution Neural Network (CNN). That comes when a more compact and powerful representation of trajectories has been learned, so a better performance has been achieved and a few labeled training data are required. In most specific circumstances that a few labeled data are obtained, semi-supervised algorithms are proper to apply.

6 PROMISING FUTURE DIRECTIONS AND TASKS

In trajectory classification, we saw that measuring trajectories with different lengths is important. Thus, a representation method or feature descriptor is essential for trajectory classification. In

Table 8. Trajectory Data Clustering Application and Corresponding Works by Semi-supervised Algorithms

Application problem	Method	Data preparation	Experiment data set
Extraction and categorization of moving objects	[71], [99]	no	{self-produced trajectory data}[71], {collected vehicles' trajectories}[99]
	[54]	segmenting trajectories to characterize time-varying information	synthetic trajectory data set, the hand sign data set, the vehicle motion trajectory data set
Action recognition	[51]	locating motion regions, then tracking the moving persons and generating trajectories	KTH, UCF50, VIRAT, TRECVID
	[67], [135]	no	{nine different human movement observation sequences}[67], {the videos captured inside a room}[135]
Route recommendation	[50]	no	synthetic data set
Discovery of social events	[74]	no	trajectory data generated by GSTD generator
	[154]	tracking pixels and generating trajectory data	UCSDped1, Avenue data set, Subway surveillance

recent years, transforming trajectory data into other space has been paid more attention, such as DFT, which keeps data information and unifying lengths of trajectory data [54]. For other preparation works, re-sampling is efficient for sparse scene [134], but it limits the robustness of model. Curve approximation fits the movement of trajectory [110, 155]. Hence, trajectory data preparation may be a promising and helpful direction.

Recently, Densely Clustering models have achieved great progress in trajectory clustering. In particular, novel distance metrics have been proposed to measure trajectory data according to different properties. Furthermore, for the trajectory data with large difference in density, grid construction is employed to improve the performance [124]. Besides grid-based DBCSAN, sub-trajectories are substituted for trajectory in Reference [62], [72], and [73].

Though Spectral Clustering models and Graph method share a similar idea, they are intrinsically different. Spectral Clustering models are easy to implement and have no restriction on data dimensions, but the models require non-negative affinities and this limitation restricts the performance and the application. Therefore, a suitable affinity matrix construction method is needed. Furthermore, it is critical to determine scale value when the affinity matrix is being computed, because it determines the clustering results are efficient or not. Thus, Spectral Clustering models need to handle the problem of constructing an affinity matrix.

In supervised algorithms, a large number of training data are required to obtain an efficient model. However, such as in Neural Network, there may be an overfitting problem, and some special steps are needed, like pooling layers in CNN. In addition, it should be noticed that a meaningful distance metric is essential for Nearest-neighbor algorithms.

The performance of state-of-the-art models on UCF Sports data set is reviewed in Table 9.

Table 9. Performance of State-of-the-Art Models on UCF Sports Data Set

State-of-the-art methods	accuracy
Gaidon et al. [42]	88.6%
Jiang et al. [58]	59.1%
Jiang et al. [59]	66.4%
Laptev et al. [69]	62.8%
Ni et al. [92]	67.7%
Rahmani et al. [103]	76.2%
Raptis et al. [104]	79.2%
Shi et al. [109]	77.2%
Turchini et al. [121]	90.2%
Vig et al. [126]	58.9%
Wang et al. [129]	76.1%
Wang et al. [130]	80%
Wang et al. [133]	63.9%
Wu et al. [142]	77.9%
Yi et al. [148]	92.1%

7 CONCLUSION

In this article, we reviewed the methods of trajectory classification. According to the fact that trajectory data have a variety of characterizations and data forms, training data have been involved or motion information, such as speed value, are recorded. Then, different algorithms are required to reach the goal, so they are classified into three categories: unsupervised, supervised, and semi-supervised algorithms. Unsupervised algorithms are called clustering methods, and they can be grouped into three sub-categories: Densely Clustering models, Hierarchical Clustering models, and Spectral Clustering models, and Spectral algorithms take better performance in this category. By means of a comprehensive analysis, we found that unsupervised algorithms have the disadvantages of high computation cost and heavy memory load, though there is no training data requirement or human experts' supervision. Supervised algorithms are divided into Nearest-neighbor algorithms, Statistical models, and Neural Network. Furthermore, Neural Network is applied to solving a number of issues, including trajectories classification. Although a huge number of training data and plenty of time are needed to understand the construction inside trajectory data by training Neural Network, it is fast on classifying the new coming trajectory and shows robustness and accuracy in the real-time application. Semi-supervised algorithms combine the advantages of both previous algorithms, but validation and correction steps require several iterations. Therefore, it reduces computation time and it is suitable for the scenario that a small number of labeled data is involved. Finally, we proposed several promising future directions and tasks, and this article could help readers to gain a thorough understanding of trajectory classification.

REFERENCES

[1] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. 2011. Trajectory space: A dual representation for non-rigid structure from motion. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 7 (2011), 1442–1456.

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 961–971.

[3] Aris Anagnostopoulos, Michail Vlachos, Marios Hadjieleftheriou, Eamonn Keogh, and Philip S. Yu. 2006. Global distance-based segmentation of trajectories. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 34–43.
[4] Gennady Andrienko, Natalia Andrienko, Salvatore Rinzivillo, Mirco Nanni, Dino Pedreschi, and Fosca Giannotti. 2009. Interactive visual clustering of large collections of trajectories. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST'09)*. IEEE, 3–10.
[5] Gianluca Antonini, Santiago Venegas Martinez, Michel Bierlaire, and Jean Philippe Thiran. 2006. Behavioral priors for detection and tracking of pedestrians in video sequences. *Int. J. Comput. Vision* 69, 2 (2006), 159–180.
[6] Gianluca Antonini and Jean-Philippe Thiran. 2006. Counting pedestrians in video sequences using trajectory clustering. *IEEE Trans. Circ. Syst. Video Technol.* 16, 8 (2006), 1008–1020.
[7] Stefan Atev, Osama Masoud, and Nikolaos Papanikolopoulos. 2006. Learning traffic patterns at intersections by spectral clustering of motion trajectories. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS'06)*. 4851–4856.
[8] Stefan Atev, Grant Miller, and Nikolaos P. Papanikolopoulos. 2010. Clustering of vehicle trajectories. *IEEE Trans. Intell. Transport. Syst.* 11, 3 (2010), 647–657.
[9] Petko Bakalov, Marios Hadjieleftheriou, and Vassilis J. Tsotras. 2005. Time relaxed spatiotemporal trajectory joins. In *Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems*. ACM, 182–191.
[10] Faisal I. Bashir, Ashfaq A. Khokhar, and Dan Schonfeld. 2007. Object trajectory-based activity classification and recognition using hidden Markov models. *IEEE Trans. Image Process.* 16, 7 (2007), 1912–1919.
[11] Faisal I. Bashir, Ashfaq A. Khokhar, and Dan Schonfeld. 2007. Real-time motion trajectory-based indexing and retrieval of video sequences. *IEEE Trans. Multimedia* 9, 1 (2007), 58–65.
[12] Miguel Angel Bautista, Antonio Hernández-Vela, Sergio Escalera, Laura Igual, Oriol Pujol, Josep Moya, Verónica Violant, and María T. Anguera. 2016. A gesture recognition system for detecting behavioral patterns of ADHD. *IEEE Trans. Cybernet.* 46, 1 (2016), 136–147.
[13] Norbert Brandle, Dietmar Bauer, and Stefan Seer. 2006. Track-based finding of stopping pedestrians-a practical approach for analyzing a public infrastructure. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*. IEEE, 115–120.
[14] Thomas Brinkhoff. 2002. A framework for generating network-based moving objects. *GeoInformatica* 6, 2 (2002), 153–180.
[15] Andries E. Brouwer and Willem H. Haemers. 2011. *Spectra of Graphs*. Springer Science & Business Media.
[16] Thomas Brox and Jitendra Malik. 2010. Object segmentation by long term analysis of point trajectories. In *Proceedings of the European Conference on Computer Vision*. Springer, 282–295.
[17] Emma Brunskill, Thomas Kollar, and Nicholas Roy. 2007. Topological mapping using spectral clustering and classification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 3491–3496.
[18] Yingyi Bu, Lei Chen, Ada Wai-Chee Fu, and Dawei Liu. 2009. Efficient anomaly monitoring over moving object trajectory streams. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 159–168.
[19] Judith Butepage, Michael J. Black, Danica Kragic, and Hedvig Kjellstrom. 2017. Deep representation learning for human motion prediction and classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6158–6166.
[20] Guochen Cai, Kyungmi Lee, and Ickjai Lee. 2016. A framework for mining semantic-level tourist movement behaviours from geo-tagged photos. In *Proceedings of the Australasian Joint Conference on Artificial Intelligence*. Springer, 519–524.
[21] Olivier Cappé, Simon J. Godsill, and Eric Moulines. 2007. An overview of existing methods and recent advances in sequential Monte Carlo. *Proc. IEEE* 95, 5 (2007), 899–924.
[22] Rima Chaker, Zaher Al Aghbari, and Imran N. Junejo. 2017. Social network model for crowd anomaly detection and localization. *Pattern Recogn.* 61 (2017), 266–281.
[23] Guangliang Chen and Gilad Lerman. 2009. Spectral curvature clustering (SCC). *Int. J. Comput. Vision* 81, 3 (2009), 317–330.
[24] Jinyang Chen, Rangding Wang, Liangxu Liu, and Jiatao Song. 2011. Clustering of trajectories based on Hausdorff distance. In *Proceedings of the International Conference on Electronics, Communications and Control (ICECC'11)*. IEEE, 1940–1944.
[25] Wei Chen and Jason J. Corso. 2015. Action detection by implicit intentional motion clustering. In *Proceedings of the IEEE International Conference on Computer Vision*. 3298–3306.
[26] Zaiben Chen, Heng Tao Shen, Xiaofang Zhou, Yu Zheng, and Xing Xie. 2010. Searching trajectories by locations: An efficiency study. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 255–266.

[27] Kyunghyun Cho and Xi Chen. 2014. Classifying and visualizing motion capture sequences using deep neural networks. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP'14)*, Vol. 2. IEEE, 122–130.

[28] Minsu Cho and Kyoung MuLee. 2010. Authority-shift clustering: Hierarchical clustering by authority seeking on graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*. IEEE, 3193–3200.

[29] Dong-Wan Choi, Jian Pei, and Thomas Heinis. 2017. Efficient mining of regional movement patterns in semantic trajectories. *Proc. VLDB Endow.* 10, 13 (2017), 2073–2084.

[30] César Roberto De Souza, Adrien Gaidon, Eleonora Vig, and Antonio Manuel López. 2016. Sympathy for the details: Dense trajectories and hybrid classification architectures for action recognition. In *Proceedings of the European Conference on Computer Vision*. Springer, 697–716.

[31] Ke Deng, Kexin Xie, Kevin Zheng, and Xiaofang Zhou. 2011. Trajectory indexing and retrieval. In *Computing with Spatial Trajectories*. Springer, 35–60.

[32] Maxime Devanne, Stefano Berretti, Pietro Pala, Hazem Wannous, Mohamed Daoudi, and Alberto Del Bimbo. 2017. Motion segment decomposition of RGB-D sequences for human behavior understanding. *Pattern Recogn.* 61 (2017), 222–233.

[33] Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo. 2015. 3D human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE Trans. Cybernet.* 45, 7 (2015), 1340–1352.

[34] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2625–2634.

[35] Ehsan Elhamifar and René Vidal. 2009. Sparse subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE, 2790–2797.

[36] Hamza Ergezer and Kemal Leblebicioğlu. 2016. Anomaly detection and activity perception using covariance descriptor for trajectories. In *Proceedings of the European Conference on Computer Vision*. Springer, 728–742.

[37] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD'96)*, Vol. 96. 226–231.

[38] Diego R. Faria and Jorge Dias. 2009. 3D hand trajectory segmentation by curvatures and hand orientation for classification through a probabilistic approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 1284–1289.

[39] Nivan Ferreira, James T. Klosowski, Carlos E. Scheidegger, and Cláudio T. Silva. 2013. Vector field k-means: Clustering trajectories by fitting multiple vector fields. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 201–210.

[40] Katerina Fragkiadaki, Pablo Arbelaez, Panna Felsen, and Jitendra Malik. 2015. Learning to segment moving objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4083–4090.

[41] Elias Frentzos, Kostas Gratsias, Nikos Pelekis, and Yannis Theodoridis. 2007. Algorithms for nearest neighbor search on moving object trajectories. *Geoinformatica* 11, 2 (2007), 159–193.

[42] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. 2014. Activity representation with motion hierarchies. *Int. J. Comput. Vision* 107, 3 (2014), 219–238.

[43] Laurent Galluccio, Olivier Michel, Pierre Comon, and Alfred O. Hero. 2012. Graph-based k-means clustering. *Signal Process.* 92, 9 (2012), 1970–1984.

[44] Chuang Gan, Naiyan Wang, Yi Yang, Dit-Yan Yeung, and Alex G. Hauptmann. 2015. Devnet: A deep event network for multimedia event detection and evidence recounting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2568–2577.

[45] Yun-Jun Gao, Chun Li, Gen-Cai Chen, Ling Chen, Xian-Ta Jiang, and Chun Chen. 2007. Efficient k-nearest-neighbor search algorithms for historical moving object trajectories. *J. Comput. Sci. Technol.* 22, 2 (2007), 232–244.

[46] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. 2017. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. IEEE, 3165–3174.

[47] Joachim Gudmundsson and Marc van Kreveld. 2006. Computing longest duration flocks in trajectory data. In *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*. ACM, 35–42.

[48] Joachim Gudmundsson, Marc van Kreveld, and Bettina Speckmann. 2004. Efficient detection of motion patterns in spatio-temporal data sets. In *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems*. ACM, 250–257.

[49] Diansheng Guo, Shufan Liu, and Hai Jin. 2010. A graph-based approach to vehicle trajectory analysis. *J. Locat. Based Serv.* 4, 3–4 (2010), 183–199.
[50] Sashi Gurung, Dan Lin, Wei Jiang, Ali Hurson, and Rui Zhang. 2014. Traffic information publication with privacy preservation. *ACM Trans. on Intell. Syst. Technol.* 5, 3 (2014), 44.
[51] Mahmudul Hasan and Amit K. Roy-Chowdhury. 2015. A continuous learning framework for activity recognition using deep hybrid feature models. *IEEE Trans. Multimedia* 17, 11 (2015), 1909–1922.
[52] Dezhi Hong, Quanquan Gu, and Kamin Whitehouse. 2017. High-dimensional time series clustering via cross-predictability. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'17).* 642–651.
[53] Han Hu, Jianjiang Feng, and Jie Zhou. 2015. Exploiting unsupervised and supervised constraints for subspace clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 8 (2015), 1542–1557.
[54] Weiming Hu, Xi Li, Guodong Tian, Stephen Maybank, and Zhongfei Zhang. 2013. An incremental DPMM-based method for trajectory clustering, modeling, and retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 5 (2013), 1051–1065.
[55] Weiming Hu, Xuejuan Xiao, Zhouyu Fu, Dan Xie, Tieniu Tan, and Steve Maybank. 2006. A system for learning statistical motion patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 9 (2006), 1450–1464.
[56] Weiming Hu, Dan Xie, Zhouyu Fu, Wenrong Zeng, and Steve Maybank. 2007. Semantic-based surveillance video retrieval. *IEEE Trans. Image Process.* 16, 4 (2007), 1168–1181.
[57] Hoyoung Jeung, Man Lung Yiu, Xiaofang Zhou, Christian S. Jensen, and Heng Tao Shen. 2008. Discovery of convoys in trajectory databases. *Proc. VLDB Endow.* 1, 1 (2008), 1068–1080.
[58] Yu-Gang Jiang, Qi Dai, Xiangyang Xue, Wei Liu, and Chong-Wah Ngo. 2012. Trajectory-based modeling of human actions with motion reference points. In *Proceedings of the European Conference on Computer Vision.* Springer, 425–438.
[59] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. 2015. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *arXiv preprint arXiv:1502.07209.*
[60] Imran N. Junejo and Hassan Foroosh. 2008. Euclidean path modeling for video surveillance. *Image Vision Comput.* 26, 4 (2008), 512–528.
[61] Margret Keuper, Bjoern Andres, and Thomas Brox. 2015. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the IEEE International Conference on Computer Vision.* 3271–3279.
[62] Hnin Su Khaing and Thandar Thein. 2014. An efficient clustering algorithm for moving object trajectories. In *Proceedings of the 3rd International Conference on Computational techniques and Artificial Intelligence (ICCTAI'14).* 11–12.
[63] Aida Khosroshahi, Eshed Ohn-Bar, and Mohan Manubhai Trivedi. 2016. Surround vehicles trajectory analysis with recurrent neural networks. In *Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC'16).* IEEE, 2267–2272.
[64] ZuWhan Kim. 2008. Real time object tracking based on dynamic feature grouping with background subtraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08).* IEEE, 1–8.
[65] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. 2008. A spatio-temporal descriptor based on 3D-gradients. In *Proceedings of the 19th British Machine Vision Conference (BMVC'08).* British Machine Vision Association, 275–1.
[66] Xiangjie Kong, Menglin Li, Kai Ma, Kaiqi Tian, Mengyuan Wang, Zhaolong Ning, and Feng Xia. 2018. Big trajectory data: A survey of applications and services. *IEEE Access* 6 (2018), 58295–58306.
[67] Dana Kulić, Wataru Takano, and Yoshihiko Nakamura. 2008. Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden Markov chains. *Int. J. Robot. Res.* 27, 7 (2008), 761–784.
[68] Suryansh Kumar, Yuchao Dai, and Hongdong Li. 2017. Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion. *Pattern Recogn.* 71 (2017), 428–443.
[69] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. 2008. Learning realistic human actions from movies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08).* IEEE, 1–8.
[70] Fabien Lauer and Christoph Schnörr. 2009. Spectral clustering of linear subspaces for motion segmentation. In *Proceedings of the IEEE 12th International Conference on Computer Vision.* IEEE, 678–685.
[71] Rikard Laxhammar and Göran Falkman. 2014. Online learning and sequential anomaly detection in trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 6 (2014), 1158–1173.
[72] Jae-Gil Lee, Jiawei Han, Xiaolei Li, and Hector Gonzalez. 2008. TraClass: Trajectory classification using hierarchical region-based and trajectory-based clustering. *Proc. VLDB Endow.* 1, 1 (2008), 1081–1094.
[73] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. 2007. Trajectory clustering: A partition-and-group framework. In *Proceedings of the ACM SIGMOD International Conference on Management of Data.* ACM, 593–604.
[74] Xiaohui Li, Vaida Ceikute, Christian S. Jensen, and Kian-Lee Tan. 2013. Effective online group discovery in trajectory databases. *IEEE Trans. Knowl. Data Eng.* 25, 12 (2013), 2752–2766.

[75] Xiaolei Li, Jiawei Han, Jae-Gil Lee, and Hector Gonzalez. 2007. Traffic density-based discovery of hot routes in road networks. In *Proceedings of the International Symposium on Spatial and Temporal Databases*. Springer, 441–459.

[76] Xi Li, Weiming Hu, and Wei Hu. 2006. A coarse-to-fine strategy for vehicle motion trajectory clustering. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, Vol. 1. IEEE, 591–594.

[77] Zhenhui Li, Jae-Gil Lee, Xiaolei Li, and Jiawei Han. 2010. Incremental clustering for trajectories. In *Proceedings of the International Conference on Database Systems for Advanced Applications*. Springer, 32–46.

[78] Liang Lin, Yongyi Lu, Yan Pan, and Xiaowu Chen. 2012. Integrating graph partitioning and matching for trajectory analysis in video surveillance. *IEEE Trans. Image Process.* 21, 12 (2012), 4844–4857.

[79] Wun-Guang Liou, Chung-Yang Hsieh, and Wei-Yang Lin. 2011. Trajectory-based sign language recognition using Discriminant Analysis in higher-dimensional feature space. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE, 1–4.

[80] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. 2014. Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 1 (2014), 99–112.

[81] Xiaobai Liu, Liang Lin, Song-Chun Zhu, and Hai Jin. 2009. Trajectory parsing by cluster sampling in spatio-temporal graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE, 739–746.

[82] Pyry Matikainen, Martial Hebert, and Rahul Sukthankar. 2009. Trajectons: Action recognition through the motion analysis of tracked features. In *Proceedings of the IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops'09)*. IEEE, 514–521.

[83] Pyry Matikainen, Martial Hebert, and Rahul Sukthankar. 2010. Representing pairwise spatial and temporal relations for action recognition. In *Proceedings of the European Conference on Computer Vision*. Springer, 508–521.

[84] Jean Damascène Mazimpaka and Sabine Timpf. 2016. Trajectory data mining: A review of methods and applications. *J. Spatial Info. Sci.* 2016, 13 (2016), 61–99.

[85] José Melo, Andrew Naftel, Alexandre Bernardino, and José Santos-Victor. 2006. Detection and classification of highway lanes using vehicle motion trajectories. *IEEE Trans. Intell. Transport. Syst.* 7, 2 (2006), 188–200.

[86] Yang Mo, Di Wu, and Yunyan Du. 2015. Application of trajectory clustering and regionalization to ocean eddies in the South China Sea. In *Proceedings of the 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM'15)*. IEEE, 45–48.

[87] Brendan Morris and Mohan Trivedi. 2009. Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE, 312–319.

[88] Brendan Tran Morris and Mohan Manubhai Trivedi. 2008. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Trans. Circ. Syst. Video Technol.* 18, 8 (2008), 1114–1127.

[89] Brendan Tran Morris and Mohan Manubhai Trivedi. 2011. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 11 (2011), 2287–2301.

[90] Andrew Naftel and Shehzad Khalid. 2006. Motion trajectory learning in the DFT-coefficient feature space. In *Proceedings of the 4th IEEE International Conference on Computer Vision Systems (ICVS'06)*. IEEE, 47–47.

[91] Mirco Nanni and Dino Pedreschi. 2006. Time-focused clustering of trajectories of moving objects. *J. Intell. Info. Syst.* 27, 3 (2006), 267–289.

[92] Bingbing Ni, Pierre Moulin, Xiaokang Yang, and Shuicheng Yan. 2015. Motion part regularization: Improving action recognition via trajectory selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3698–3706.

[93] Peter Ochs, Jitendra Malik, and Thomas Brox. 2014. Segmentation of moving objects by long term video analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 6 (2014), 1187–1200.

[94] Andrey Tietbohl Palma, Vania Bogorny, Bart Kuijpers, and Luis Otavio Alvares. 2008. A clustering-based approach for discovering interesting places in trajectories. In *Proceedings of the ACM Symposium on Applied Computing*. ACM, 863–868.

[95] Guillem Palou and Philippe Salembier. 2013. Hierarchical video representation with trajectory binary partition tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2099–2106.

[96] Nikos Pelekis, Ioannis Kopanakis, Evangelos E. Kotsifakos, Elias Frentzos, and Yannis Theodoridis. 2009. Clustering trajectories of moving objects in an uncertain world. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'09)*, Vol. 9. 417–427.

[97] Nikos Pelekis, Ioannis Kopanakis, Evangelos E. Kotsifakos, Elias Frentzos, and Yannis Theodoridis. 2011. Clustering uncertain trajectories. *Knowl. Info. Syst.* 28, 1 (2011), 117–147.

[98] Nikos Pelekis, Panagiotis Tampakis, Marios Vodas, Christos Doulkeridis, and Yannis Theodoridis. 2017. On temporal-constrained sub-trajectory cluster analysis. *Data Min. Knowl. Discov.* (2017), 1–37.

[99] Claudio Piciarelli and Gian Luca Foresti. 2006. On-line trajectory clustering for anomalous events detection. *Pattern Recogn. Lett.* 27, 15 (2006), 1835–1842.

[100] Claudio Piciarelli, Christian Micheloni, and Gian Luca Foresti. 2008. Trajectory-based anomalous event detection. *IEEE Trans. Circ. Syst. Video Technol.* 18, 11 (2008), 1544–1554.

[101] Nicola Piotto, Nicola Conci, and Francesco G. B. De Natale. 2009. Syntactic matching of trajectories for ambient intelligence applications. *IEEE Trans. Multimedia* 11, 7 (2009), 1266–1275.

[102] Stergios Poularakis and Ioannis Katsavounidis. 2016. Low-complexity hand gesture recognition system for continuous streams of digits and letters. *IEEE Trans. Cybernet.* 46, 9 (2016), 2094–2108.

[103] Hossein Rahmani, Ajmal Mian, and Mubarak Shah. 2018. Learning a deep model for human action recognition from novel viewpoints. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 3 (2018), 667–681.

[104] Michalis Raptis, Iasonas Kokkinos, and Stefano Soatto. 2012. Discovering discriminative action parts from mid-level video representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 1242–1249.

[105] Joachim Schenk and Gerhard Rigoll. 2006. Novel hybrid NN/HMM modelling techniques for on-line handwriting recognition. In *Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition.* Suvisoft.

[106] Tobias Schreck, Jürgen Bernard, Tatiana Von Landesberger, and Jörn Kohlhammer. 2009. Visual cluster analysis of trajectory data with interactive Kohonen maps. *Info. Visual.* 8, 1 (2009), 14–29.

[107] Zhanpeng Shao and Youfu Li. 2016. On integral invariants for effective 3D motion trajectory matching and recognition. *IEEE Trans. Cybernet.* 46, 2 (2016), 511–523.

[108] Yemin Shi, Yonghong Tian, Yaowei Wang, and Tiejun Huang. 2017. Sequential deep trajectory descriptor for action recognition with three-stream CNN. *IEEE Trans. Multimedia* 19, 7 (2017), 1510–1520.

[109] Yemin Shi, Wei Zeng, Tiejun Huang, and Yaowei Wang. 2015. Learning deep trajectory descriptor for action recognition in videos using deep neural networks. In *Proceedings of the IEEE International Conference on Multimedia and Expo.* IEEE, 1–6.

[110] Rowland R. Sillito and Robert B. Fisher. 2008. Semi-supervised learning for anomalous trajectory detection. In *Proceedings of the British Machine Vision Conference (BMVC'08)*, vol. 1. 035–1.

[111] Damien Simonnet, Eric Anquetil, and Manuel Bouillon. 2017. Multi-criteria handwriting quality analysis with online fuzzy models. *Pattern Recogn.* 69 (2017), 310–324.

[112] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems.* MIT Press, 568–576.

[113] Dinesh Singh and C. Krishna Mohan. 2017. Graph formulation of video activities for abnormal activity recognition. *Pattern Recogn.* 65 (2017), 265–272.

[114] Francesco Solera, Simone Calderara, and Rita Cucchiara. 2016. Socially constrained structural learning for groups detection in crowd. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 5 (2016), 995–1008.

[115] Ju Sun, Xiao Wu, Shuicheng Yan, Loong-Fah Cheong, Tat-Seng Chua, and Jintao Li. 2009. Hierarchical spatio-temporal context modeling for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09).* IEEE, 2004–2011.

[116] Naohiko Suzuki, Kosuke Hirasawa, Kenichi Tanaka, Yoshinori Kobayashi, Yoichi Sato, and Yozo Fujino. 2007. Learning motion patterns and anomaly detection by human trajectory analysis. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics.* IEEE, 498–503.

[117] Seyed Salim Tabatabaei, Mark Coates, and Michael Rabbat. 2011. Ganc: Greedy agglomerative normalized cut. *arXiv preprint arXiv:1105.0974.*

[118] Lu-An Tang, Yu Zheng, Jing Yuan, Jiawei Han, Alice Leung, Chih-Chieh Hung, and Wen-Chih Peng. 2012. On discovery of traveling companions from streaming trajectories. In *Proceedings of the IEEE 28th International Conference on Data Engineering (ICDE'12).* IEEE, 186–197.

[119] Yannis Theodoridis, Jefferson RO Silva, and Mario A Nascimento. 1999. On the generation of spatiotemporal datasets. In *Proceedings of the International Symposium on Spatial Databases.* Springer, 147–164.

[120] Eleftherios Tiakas, Apostolos N. Papadopoulos, Alexandros Nanopoulos, Yannis Manolopoulos, Dragan Stojanovic, and Slobodanka Djordjevic-Kajan. 2006. Trajectory similarity search in spatial networks. In *Proceedings of the 10th International Database Engineering and Applications Symposium (IDEAS'06).* IEEE, 185–192.

[121] Francesco Turchini, Lorenzo Seidenari, and Alberto Del Bimbo. 2015. Understanding sport activities from correspondences of clustered trajectories. In *Proceedings of the IEEE International Conference on Computer Vision Workshops.* 43–50.

[122] Tinne Tuytelaars and Krystian Mikolajczyk. 2008. Local invariant feature detectors: A survey. *Found. Trends Comput. Graph. Vision* 3, 3 (2008), 177–280.

[123] Hirofumi Uemura, Seiji Ishikawa, and Krystian Mikolajczyk. 2008. Feature tracking and motion compensation for action recognition. In *Proceedings of the British Machine Vision Conference (BMVC'08).* 1–10.

[124] Ozge Uncu, William A. Gruver, Dilip B. Kotak, Dorian Sabaz, Zafeer Alibhai, and Colin Ng. 2006. Gridbscan: Grid density-based spatial clustering of applications with noise. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC'06)*, Vol. 4. IEEE, 2976–2981.

[125] Marcos R. Vieira, Petko Bakalov, and Vassilis J. Tsotras. 2009. On-line discovery of flock patterns in spatio-temporal data. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 286–295.

[126] Eleonora Vig, Michael Dorr, and David Cox. 2012. Space-variant descriptor sampling for action recognition based on saliency and eye movements. In *Proceedings of the European Conference on Computer Vision*. Springer, 84–97.

[127] Monica Wachowicz, Rebecca Ong, Chiara Renso, and Mirco Nanni. 2011. Finding moving flock patterns among pedestrians through collective coherence. *Int. J. Geogr. Info. Sci.* 25, 11 (2011), 1849–1864.

[128] Feng Wang, Yu-Gang Jiang, and Chong-Wah Ngo. 2008. Video event detection using motion relativity and visual relatedness. In *Proceedings of the 16th ACM International Conference on Multimedia*. ACM, 239–248.

[129] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. 2011. Action recognition by dense trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. IEEE, 3169–3176.

[130] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. 2013. Dense trajectories and motion boundary descriptors for action recognition. *Int. J. Comput. Vision* 103, 1 (2013), 60–79.

[131] Heng Wang, Dan Oneata, Jakob Verbeek, and Cordelia Schmid. 2016. A robust and efficient video representation for action recognition. *Int. J. Comput. Vision* 119, 3 (2016), 219–238.

[132] He Wang and Carol O'Sullivan. 2016. Globally continuous and non-Markovian crowd activity analysis from videos. In *Proceedings of the European Conference on Computer Vision*. Springer, 527–544.

[133] Heng Wang and Cordelia Schmid. 2013. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*. 3551–3558.

[134] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. 2009. Evaluation of local spatio-temporal features for action recognition. In *Proceedings of the British Machine Vision Conference (BMVC'09)*. BMVA Press, 124–1.

[135] Lijun Wang and Ming Dong. 2014. Detection of abnormal human behavior using a matrix approximation-based approach. In *Proceedings of the 13th International Conference on Machine Learning and Applications (ICMLA'14)*. IEEE, 324–329.

[136] Limin Wang, Yu Qiao, and Xiaoou Tang. 2015. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4305–4314.

[137] Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao. 2015. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*.

[138] Pichao Wang, Zhaoyang Li, Yonghong Hou, and Wanqing Li. 2016. Action recognition based on joint trajectory maps using convolutional neural networks. In *Proceedings of the ACM on Multimedia Conference*. ACM, 102–106.

[139] Xiaogang Wang, Kinh Tieu, and Eric Grimson. 2006. Learning semantic scene models by trajectory analysis. In *Proceedings of the European Conference on Computer Vision*. Springer, 110–123.

[140] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. 2008. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proceedings of the European Conference on Computer Vision*. Springer, 650–663.

[141] Zuxuan Wu, Yanwei Fu, Yu-Gang Jiang, and Leonid Sigal. 2016. Harnessing object and scene semantics for large-scale video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3112–3121.

[142] Zuxuan Wu, Yu-Gang Jiang, Jun Wang, Jian Pu, and Xiangyang Xue. 2014. Exploring inter-feature and inter-class relationships with deep neural networks for video classification. In *Proceedings of the 22nd ACM International Conference on Multimedia*. ACM, 167–176.

[143] Tao Xiang and Shaogang Gong. 2008. Spectral clustering with eigenvector selection. *Pattern Recogn.* 41, 3 (2008), 1012–1029.

[144] Hongteng Xu, Yang Zhou, Weiyao Lin, and Hongyuan Zha. 2015. Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage. In *Proceedings of the IEEE International Conference on Computer Vision*. 4328–4336.

[145] Zhixian Yan, Christine Parent, Stefano Spaccapietra, and Dipanjan Chakraborty. 2010. A hybrid model and computing platform for spatio-semantic trajectories. In *Proceedings of the Extended Semantic Web Conference*. Springer, 60–75.

[146] Tingting Yao, Zhiyong Wang, Zhao Xie, Jun Gao, and David Dagan Feng. 2017. Learning universal multiview dictionary for human action recognition. *Pattern Recogn.* 64 (2017), 236–244.

[147] Hao Ye, Zuxuan Wu, Rui-Wei Zhao, Xi Wang, Yu-Gang Jiang, and Xiangyang Xue. 2015. Evaluating two-stream CNN for video classification. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 435–442.

[148] Yang Yi and Maoqing Lin. 2016. Human action recognition with graph-based multiple-instance learning. *Pattern Recogn.* 53 (2016), 148–162.

[149] Josh Jia-Ching Ying, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S. Tseng. 2011. Semantic trajectory mining for location prediction. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 34–43.

[150] Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S. Tseng. 2010. Mining user similarity from semantic trajectories. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location-based Social Networks*. ACM, 19–26.

[151] Jin Soung Yoo, Shashi Shekhar, John Smith, and Julius P. Kumquat. 2004. A partial join approach for mining co-location patterns. In *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems*. ACM, 241–249.

[152] Chao Yuan and Amit Chakraborty. 2016. Deep convolutional factor analyser for multivariate time series modeling. In *Proceedings of the IEEE 16th International Conference on Data Mining (ICDM'16)*. IEEE, 1323–1328.

[153] Guan Yuan, Penghui Sun, Jie Zhao, Daxing Li, and Canwei Wang. 2017. A review of moving object trajectory clustering algorithms. *Artific. Intell. Rev.* 47, 1 (2017), 123–144.

[154] Yuan Yuan, Yachuang Feng, and Xiaoqiang Lu. 2017. Statistical hypothesis detector for abnormal event detection in crowded scenes. *IEEE Trans. Cybernet.* 47, 11 (2017), 3597–3608.

[155] Tianzhu Zhang, Hanqing Lu, and Stan Z. Li. 2009. Learning semantic scene models by object classification and trajectory clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE, 1940–1947.

[156] Zhang Zhang, Kaiqi Huang, and Tieniu Tan. 2006. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, Vol. 3. IEEE, 1135–1138.

[157] Zhang Zhang, Kaiqi Huang, Tieniu Tan, and Liangsheng Wang. 2007. Trajectory series analysis-based event rule induction for visual surveillance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

[158] Zhang Zhang, Kaiqi Huang, Tieniu Tan, Peipei Yang, and Jun Li. 2016. Red-sfa: Relation discovery-based slow feature analysis for trajectory clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 752–760.

[159] Weiqi Zhao, Zhang Zhang, and Kaiqi Huang. 2018. Gestalt laws-based tracklets analysis for human crowd understanding. *Pattern Recogn.* 75 (2018), 112–127.

[160] Bolong Zheng, Nicholas Jing Yuan, Kai Zheng, Xing Xie, Shazia Sadiq, and Xiaofang Zhou. 2015. Approximate keyword search in semantic trajectory database. In *Proceedings of the IEEE 31st International Conference on Data Engineering (ICDE'15)*. IEEE, 975–986.

[161] Kai Zheng, Yu Zheng, Nicholas Jing Yuan, and Shuo Shang. 2013. On discovery of gathering patterns from trajectories. In *Proceedings of the IEEE 29th International Conference on Data Engineering (ICDE'13)*. IEEE, 242–253.

[162] Yu Zheng. 2015. Trajectory data mining: An overview. *ACM Trans. Intell. Syst. Technol.* 6, 3 (2015), 29.

[163] Yu Zheng, Xing Xie, and Wei-Ying Ma. 2010. GeoLife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* 33, 2 (2010), 32–39.

[164] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. 2009. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, 791–800.

[165] Qian-Yi Zhou and Vladlen Koltun. 2013. Dense scene reconstruction with points of interest. *ACM Trans. Graph.* 32, 4 (2013), 112.

[166] Yue Zhou, Shuicheng Yan, and Thomas S. Huang. 2007. Detecting anomaly in videos from trajectory similarity analysis. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE, 1087–1090.

Received November 2018; revised January 2019; accepted May 2019