

A survey on trajectory clustering analysis

Jiang Bian^a, Dayong Tian^{b,*}, Yuanyan Tang^c, Dacheng Tao^d

^a*Centre for Artificial Intelligence, FEIT, University of Technology Sydney, Australia*

^b*School of Electronics and Information, Northwestern Polytechnical University, China*

^c*Faculty of Science and Technology, University of Macau, China*

^d*UBTECH Sydney AI Centre, FEIT, University of Sydney, Australia*

Abstract

This paper comprehensively surveys the development of trajectory clustering. Considering the critical role of trajectory data mining in modern intelligent systems for surveillance security, abnormal behavior detection, crowd behavior analysis, and traffic control, trajectory clustering has attracted growing attention. Existing trajectory clustering methods can be grouped into three categories: unsupervised, supervised and semi-supervised algorithms. In spite of achieving a certain level of development, trajectory clustering is limited in its success by complex conditions such as application scenarios and data dimensions. This paper provides a holistic understanding and deep insight into trajectory clustering, and presents a comprehensive analysis of representative methods and promising future directions.

Keywords: trajectory clustering, object movement, pattern recognition, survey, review

1. Introduction

With the development of tracking and surveillance devices, tremendous numbers of object trajectory data were collected, which makes extracting useful information imperative and challenging. Trajectory clustering is an efficient

*Corresponding author

Email addresses: bianjiang22@gmail.com (Jiang Bian), dayongt@gmail.com (Dayong Tian), yytang@umac.mo (Yuanyan Tang), dacheng.tao@sydney.edu.au (Dacheng Tao)

method to analyze trajectory data and it has been applied in pattern recognition, data analysis and machine learning, etc. Furthermore, trajectory clustering aims to gain space, time or even potential information inside trajectory data, so it is ubiquitous in some application fields such as object motion prediction [1], traffic monitoring [2][3][4], activity understanding [5][6][7], abnormal detection [8][9][10][11], 3 dimensional reconstruction[12], weather forecasting [13] and geography [14].

Trajectory data are recorded in different formats according to device types, object movements and even purposes. For instance, GPS tracking devices generate a trajectory by tracking object movement as $Trajectory = (Tr_1, Tr_2, \dots, Tr_n)$, 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 Fig.1. In some specific circumstances, other properties relevant to object movement are added, such as velocity, direction or acceleration. As shown in Fig.2, for image data, a sequence of pixels in consecutive frames forms up a trajectory, which is similar to optical flow [15][16].

In order to measure similarities among different types of trajectory data,

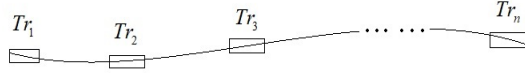


Figure 1: Trajectory generated by GPS tracking devices

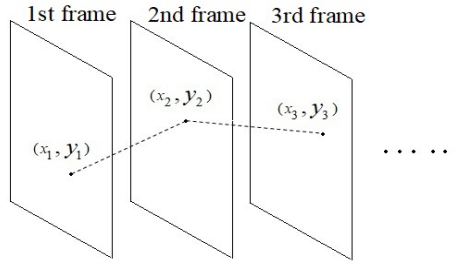


Figure 2: Trajectory generated from camera device

data representation, feature extraction and distance metric selection are crit-

ical preliminary works of trajectory clustering. For example, trajectories can be represented as a vector and downsampled to an unified length, so Euclidean distance is used [17]. Trajectories also can be treated as samples of a probabilistic distribution. Hence, Bhattacharyya Distance [18] is used to measure the distance between two distributions.

According to the availability of labeled data, trajectory clustering methods are divided in three categories: unsupervised, supervised, semi-supervised. Unsupervised models aim at clustering data without human experts supervision or labeled data. An inference function has been drawn by analyzing unlabeled data sets [19][13][20][21]. Supervised models are learned prior to trajectory clustering. Generally, labeled data are used to learn a function mapping data to their labels, i.e. clusters. The clusters of unlabeled data are predicted by this function, then [22][9][23][24]. Labeling data need a heavy burden of manual works by human experts. It is unfeasible for large data sets. Semi-supervised compromises the previous two types of models. It is trained by labeled data and tuned by unlabeled data [4][8][9].

The rest of this paper is organized as follows. Preliminary works are introduced in Sect.2, and the models based on unsupervised algorithms are described in Sect.3. A description of the models under supervised algorithms are presented in Sect.4. Sect.5 discusses some models based on semi-supervised algorithms. Finally, promising future directions are given in Sect.6, and conclusions are made in Sect.7.

2. Preliminaries

2.1. Trajectory Clustering Preparation

In some clustering models [25][26][27][28], trajectory data are required to be set as a unified length so that they could be measured. However, as shown in Fig.3, for two arbitrary trajectories, their lengths maybe largely different from each other. **Therefore, representing trajectories in an unified length with little loss of information is a major preliminary work of these models.** This procedure

is called clustering preparation.

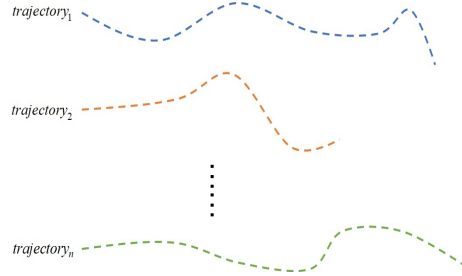


Figure 3: Trajectory data with varying lengths

2.1.1. Trajectory Transformation Algorithms

For some methods, original data are represented in other space with a same length. For instance, trajectory data are projected into a subspace [29]. Linear transformation algorithm aims at representing trajectory as a combination of basis trajectories [30]. Curve fitting is another method to approximate trajectories by a parameterized quadratic curve [25]. In order to distinguish similar curves, the direction of the last trajectory point is chosen as an additional parameter. In [26], 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 [13]. Vector fields give trajectory a smooth streamline and induce a notion of similarity of trajectories. Principle Component Analysis (PCA) is a statistical procedure to compute a set of linearly uncorrelated variables called principle components by orthogonal transformation. To avoid partially extracted information, a number of organized segmentations substitute for the corresponding trajectory in [5] and [31]. 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 comprised of Fourier coefficients in [27] and [28]. In [32], 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 aligned as matrix [31]. In a complex scene such as hand writing data set, Equidistant sampling fixes the problem that two same characters are recorded in different temporal sequence because of different writing speeds [33]. Since re-sampled trajectory points are discontinuous, it is critical that normalization should be involved after re-sampling [34]. It has been widely acknowledged that re-sampling method causes information loss [35]. Therefore, sparsity regularization is used in [36][37][38][39].

2.1.3. Trajectory Substitute

Sub-trajectories hold partial and hidden information of original trajectory data [28][40], so they are put together and describe trajectory more flexible. 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 [13]. Sub-trajectories also lead to simplified trajectories which represent trajectory data as some smaller, less complex primitives suitable for storage and retrieval purposes [41]. In [42], sub-trajectories are generated by well-defined policies based on facility performance, time range or distance range. In [5] and [31], trajectory is segmented at the so-called significant changing points at which direction or speed changes dramatically. Curvature describes direction information, and it could be extracted if a tra-

jectory is treated as a curve by connecting consecutive trajectory points. Curvatures are computed by transforming 3-dimensional position coordinates of points into spherical system and quantized as *up*, *down*, *left*, *right* [43], then a trajectory is segmented at the points where curvature changes. In addition, Minimum Description Length (MDL) principle traces the sub-trajectories holding primary movement of trajectory data by minimizing the differences between sub-trajectories and the corresponding trajectories in [40]. Minimum Bounding Rectangles (MBR) is proposed to separate trajectories under occlusion and optimize the inter-object separability in [41]. 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 attention so Regional Segmenting method is implemented. The whole scene is split into several regions and boundaries of the regions segment trajectories [44]. 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 special regions are used to represent trajectory or scene in [45] 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 [46], 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 [46] where the first one is the points in entry/exit zones and the second one is 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 [47] and [48]. The importance of points can be measured and high-scored ones are selected in [49]. For video data, POI are obtained by Pyramid Representation [50]. In addition, optical flow is another popular implementation by estimating trajectory motion in [16] and [51].

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 [32] and [50]. In [50], 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 [32][52][53][54][55][56]. Furthermore, Scale-invariant Feature Transform (SIFT) descriptor represents image patch around tracking point [57][58][59][60][61], and computes scale and orientation information of image patches to localize tracking object in consecutive frames. In [57], Kanade Lucas Tomasi (KLT) tracker is used to find trajectory points and SIFT is applied to represent them. In [59], Difference-of-Gaussian (DOG) detector is used to detecting trajectory points instead of KLT in [57].

2.2. Common Distance Measurements

Essentially, trajectory are allocated into cohesive groups according to their mutual similarities. An appropriate metric is necessary [62][63][64].

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

$$D(A, B) = \frac{1}{N} \sum [(a_n^x - b_n^x)^2 + (a_n^y - b_n^y)^2]^{\frac{1}{2}}, \quad (1)$$

where a_n^x and a_n^y indicate the n th point of trajectory A on Cartesian coordinate. N is the total number of points. In [17], Euclidean distance is used to measure the contemporary instantiations of trajectories.

Hausdorff Distance: Hausdorff distance measures the similarities by considering how close every point of one trajectory to some points of the other one, and it measures trajectories A and B without unifying the lengths in [65][66],

$$D(A, B) = \max\{d(A, B), d(B, A)\}, \quad (2)$$

$$\begin{cases} d(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \\ d(B, A) = \max_{b \in B} \min_{a \in A} \|b - a\|, \end{cases} \quad (3)$$

Bhattacharyya Distance: Bhattacharyya distance measures how closely of two probability distributions. In [18], it is employed to measures similarities of quantized directions of points,

$$D(A, B) = -\ln(BC(A, B)), \quad (4)$$

where $BC(A, B) = \sum_{t=1}^T \sqrt{a_t \cdot b_t}$ and it is used to measure the separability of A and B . a_t and b_t 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 trajectories A and B , 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

$$D(A, B) = \inf_{\alpha, \beta} \max_{t \in [0, 1]} \{d(A(\alpha(t)), B(\beta(t)))\}, \quad (5)$$

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 [67][68].

$$W(A, B) = \min_f \frac{1}{n} \sum_{i=1}^n \|a_i - b_{f(i)}\|_2, \quad (6)$$

where A has n points and B 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 $LCSS_{\epsilon, \delta}(A, B)$ is written as

$$LCSS_{\epsilon, \delta}(A, B) = \begin{cases} 0, & \text{if } A \text{ or } B \text{ is empty} \\ 1 + LCSS_{\epsilon, \delta}(Head(A), Head(B)), & \\ \text{if } \|a_N - b_M\| < \epsilon \text{ and } |N - M| < \delta & \\ \max(LCSS_{\epsilon, \delta}(Head(A), B), & \\ LCSS_{\epsilon, \delta}(A, Head(B))), & \text{otherwise,} \end{cases} \quad (7)$$

where $Head(A)$ indicates first $N - 1$ points belonging to A and $Head(B)$ denotes first $M - 1$ points of B . Finally, $D(A, B) = 1 - \frac{LCSS_{\epsilon, \delta}(A, B)}{\max(N, M)}$.

In [40][69][70], more other distance types are proposed to consider more properties such as angle distance, center distance and parallel distance, which are defined as

$$d_{angle}(L_i, L_j) = \begin{cases} \|L_j\| \times \sin(\theta), & 0^\circ \leq \theta \leq 90^\circ \\ \|L_j\|, & 90^\circ \leq \theta \leq 180^\circ, \end{cases} \quad (8)$$

where θ is the smaller intersecting angle between L_i and L_j .

$$d_{center}(L_i, L_j) = \|center_i - center_j\|, \quad (9)$$

where $d_{center}(L_i, L_j)$ is the Euclidean distance between center points of L_i and L_j .

$$d_{parallel}(L_i, L_j) = \min(l_1, l_2), \quad (10)$$

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 L_i respectively.

Distance metrics are used in much more fields relating to trajectories clustering, e.g., density clustering [40][69][71][72][73]. It is critical to choose an optimal distance according to the scene. For instance, LCSS distance is proved to provide outperforming performance without concerning trajectories length [62]. Hausdorff distance aims at finding the minimum distance between two trajectories and ignore time-order in data. A comparison of distance is listed in Table I,

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	$O(mn)$
other distance types	No	$O(1)$

3. Unsupervised Algorithms of 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

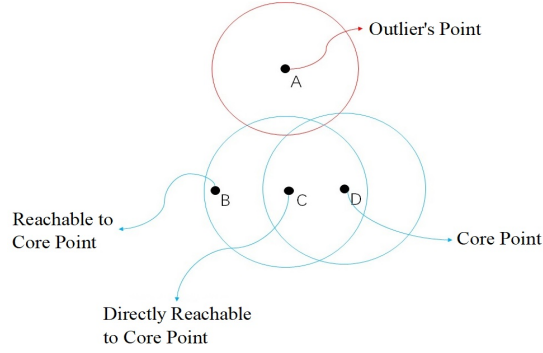


Figure 4: DBSCAN

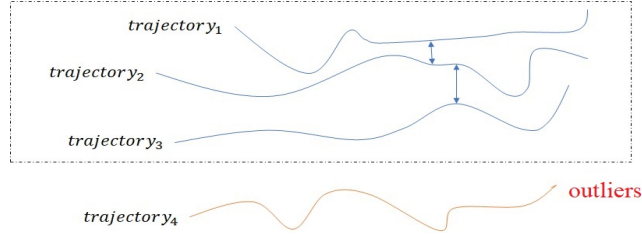


Figure 5: DBSCAN for trajectory clustering

Given the centroids, the closely points are packed together and this procedure is called densely clustering. Inspired by this idea, Density-based spatial clustering of applications with noise (DBSCAN) which has been widely applied to trajectory clustering is proposed in [19]. In DBSCAN, 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 $\{q_1, q_2, \dots, q_n\}$ are *reachable* to p if there is a path that q_1 is *directly reachable* to p and each q_{i+1} is *directly reachable* q_i [74][75]. 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 [40][69][76], 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 [40], the distance is measured by a combination of angle distance, center distance, parallel distance with equal weight in [69] and [76]. The core trajectories are computed from the clusters and used for classifying new coming trajectory in [69], [77], [78] and [79], e.g., all trajectories points belonging to same cluster are averaged as a new point at each time, and all averaged points form the representations of clusters [69]. In an adaptive multi-kernel-based method, shrunk clusters represent all groups by considering the attributes including positions, speeds and points, which retains much more discriminative messages in [80].

Besides DBSCAN, there are some other models belonging to Densely Clustering models cluster trajectory data. K-means clusters trajectories by searching centroids of clusters repeatedly [13][48][81][82][83][84]. For improving the performance of K-means, Expectation Maximization (EM) algorithm is implemented to solve optimization problem iteratively [85]. Due to the issues such as data imprecision and complexity of large data sets, a trajectory may belong to multiple clusters so EM is used to classify them [86]. Fuzzy C-Means (FCM) algorithm employs parameters to measure the level of cluster fuzziness for each trajectory, called fuzzifier. The algorithm searches correct direction in each iteration for cluster trajectories [87][88][89].

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 Fig.6, two hierarchical types are also known as “bottom-up” and “top-down” approaches.

In Agglomerative frameworks, trajectories are grouped and the similar clus-

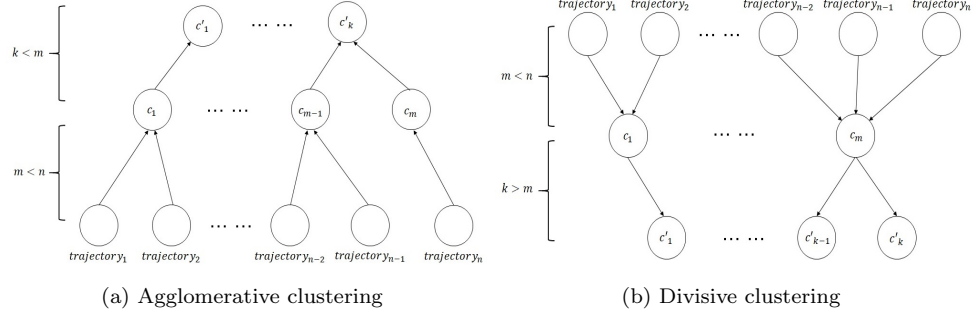


Figure 6: Hierarchical clustering models

ters 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, Agglomerative clustering models were explored in [90] 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. Following this framework, trajectory data are characterized by direction feature and clustered by Dominant-set embedded Bhattacharyya distance in initial clustering stage [18]. In each cluster, trajectories are split further except for the ones holding similar positions. Because of the good performance of iterative models, Test-and-Divide (TAD) model is proposed [91]. 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 [20]. For instance, trajectory $A = \{a_1, a_2, \dots, a_n\}$ where $a_i = \langle x_i, y_i, \beta_i \rangle$. It is comprised of 2-dimensional position and an additional attribute β such as velocity or object size. In the coarse clustering step, the distance measurement between trajectory

A and its nearest observation trajectory B are shown as follows,

$$f(A, B) = \frac{1}{N_A} \sum_{a_i \in A} \|(x_i^a - x_{\psi(i)}^b, y_i^a - y_{\psi(i)}^b + \gamma d(\beta_i^a, \beta_{\psi(i)}^b))\|, \quad (11)$$

where $\psi(i) = \arg \min_{j \in B} \|(x_i^a - x_j^b, y_i^a - y_j^b)\|$ and the minimum distance value is counted as the distance between A and B . N_A is the total number of points belonging to A , $d(\beta_i^a, \beta_{\psi(i)}^b)$ indicates the dissimilarity of A and B , and γ is weight parameter. In the fine-clustering stage, the model aims at distinguishing distortions by considering directed similarity $S_{A \rightarrow B}$ and confidence $C_{A \rightarrow B}$

$$S_{A \rightarrow B} = \frac{\sum_{a_i \in A} c(a_i, b_{\psi(i)}) s(a_i, b_{\psi(i)})}{\sum_{a_i \in A} c(a_i, b_{\psi(i)})}, \quad (12)$$

$$C_{A \rightarrow B} = \frac{\sum_{a_i \in A} c(a_i, b_{\psi(i)})^2}{\sum_{a_i \in A} c(a_i, b_{\psi(i)})}, \quad (13)$$

where $c(a_i, b_{\psi(i)}) = \exp(\frac{-\|(x_i^a - x_{\psi(i)}^b, y_i^a - y_{\psi(i)}^b)\|}{\sigma_1})$ and $s(a_i, b_{\psi(i)}) = \frac{\exp(-d(\beta_i^a, \beta_{\psi(i)}^b))}{\sigma_2}$.

Furthermore, a similar hierarchical framework is explored to group videos by constructing the trajectories of video [92] 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 [93] and [94]. For two trajectories in video, $a = \{x_1^a, x_2^a, \dots, x_{t_a}^a\}$ and $b = \{x_1^b, x_2^b, \dots, x_{t_b}^b\}$, the distance is computed as follows,

$$d(a, b) = \max_{t \in [\tau_1, \tau_2]} d_{spatial}[t] \cdot \frac{1}{\tau_2 - \tau_1} \sum_{t=\tau_1}^{\tau_2} d_{velocity}[t], \quad (14)$$

where $d_{spatial}[t]$ is the positional distance at time stamp t , and $d_{velocity}[t]$ is the similarity measurement of velocity. An affinity matrix $w(a, b) = \exp(-d(a, b))$ is calculated and trajectories are clustered by greedy agglomerative hierarchical models [93][94]. The clusters are overlapped because of similar parts, so every trajectory is weighted and optimized to classify in [95]. Since one motion object may generate several trajectories, it is critical to employ as much features as possible to ensure object recognition, and a multi-layer classifier is invented in [18][96].

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 [21]. In addition, affinity matrix characterizes videos [97] and represents the relationships. In [98], affinity matrix A is constructed as follows,

$$A_{ij} = \exp\left[\frac{-\bar{d}_{ij}}{2\sigma^2}\right], \quad (15)$$

where $\bar{d}_{ij} = \frac{1}{n} \sum_{k=1}^n \|x_{i,k} - x_{j,k}\|$, and $x_{i,k}$ indicates the k th point of trajectory i . Considering different lengths of trajectories, some novel models are explored to construct affinity matrix [99][100] and it is constructed as

$$A_{ij} = \begin{cases} e^{(-\frac{1}{\sigma_i \sigma_j} \|v_i - v_j\|^2)}, & \text{for } i \neq j \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

where v_i and v_j are points, σ_i and σ_j indicates scale invariance computed by the median of the l nearest neighbors. In order to increase the separation of points belonging to different groups, SVD decomposition is used to construct the affinity matrix [101]. In addition, a novel distance method is explored to compute trajectories P and Q [2] so that spatial distinction can be considered.

$$s(P, Q) = e^{-\frac{1}{2} h_\alpha(P, Q) h_\alpha(Q, P) / (\sigma_P \sigma_Q)}, \quad (17)$$

$$h_\alpha(P, Q) = \text{ord}_{p \in P}^\alpha \left(\min_{q \in N(C(p))} d(p, q) \right), \quad (18)$$

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

For clustering high dimensional trajectory data by Spectral Clustering models, several novel methods are explored in [102], [103] and [104]. For example, a new similarity metric captures causal relationships between time series in

[103] and a mixture of affinity subspaces is applied to approximate trajectory in [104]. Trajectory data are represented by considering covariance features of trajectories in [105], so it avoids considering different lengths of trajectory data. Spectral clustering works with multiple-instance learning frameworks to achieve human action recognition in [106].

Spectral Clustering models are derived from Graph Theory in which an undirected graph represents the relationships and constructs a symmetric adjacency matrix presenting them [107]. By constructing a graph, both explicit and implicit intentions inside trajectory data are mined [108]. The graph is cut into sub-graphs to classify trajectories, and each sub-graph represents its own cluster [25][109]. Hierarchical layers search sub-clusters in each cluster by treating trajectories points as graph nodes and this procedure is called *Hierarchical graph partitioning* [110]. For considering more variables, a novel measurement function comprised of the entropy rate of a random walk on a graph is presented in [65]. From the idea that an undirected graph can be represented as an adjacent matrix, a directed graph also can be involved [3]. Trajectory Binary Partition Tree (BPT) represents video in [111] 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 [109]. An object creates several trajectories if different parts of the object are tracked, so a model is invented to describe trajectories by feature patches [112]. 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 [113]. Since the great performance of PageRank, it is used to score the edges in [114], too.

3.4. Discussion

Densely Clustering models classify trajectories by distance metrics mostly, which may result in classifying trajectory data by spatial information. Hierar-

chical Clustering models fix this problem by considering more attributes in each level. However, this operation cost much more time in computation. Spectral Clustering models compute internal relationships by analyzing the affinity matrix, and it saves much more computational resource by processing all trajectory data together. But [115] 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.

4. Supervised Algorithms of Trajectory Clustering

Supervised algorithms aims at learning a function which determines the labels of testing data after analyzing labeled training data. Therefore, supervised algorithms outperform on scene understanding and utilization of ground truth, and the algorithms could save much more computation resource. In some supervised algorithms, trajectory data are classified by unsupervised algorithms and the representations 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 clustered quickly in [5] and [87]. Trajectory data are classified and organized in a tree-construction and new coming trajectories are clustered by searching the tree in [4] and [116].

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 clustering, the distances from an inquiry trajectory to all labeled trajectory data are computed, and the label of the inquiry trajectory is voted by its k nearest neighbors. Shown

in Fig.7, the inquiry trajectory is assigned as blue cluster if $k = 1$ and assigned as red one if $k = 3$.

In the implementation, it is important to choose a suitable distance metric

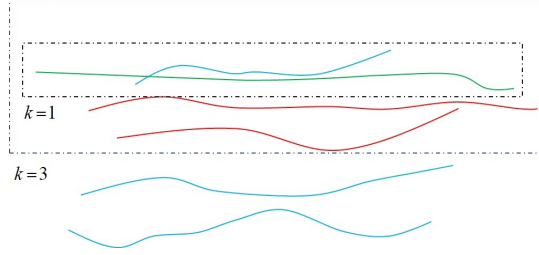


Figure 7: k -NN for trajectory clustering. Inquiry trajectory is the green one, the labeled data are the red and the blue ones which means two clusters.

according to the scenario, occlusion, trajectory data sources and feature types. Therefore, trajectory data are represented by MBR and classified by k -NN in [22]. It avoids occlusion and increasing inter-object separability. Furthermore, trajectory data are represented in Riemannian manifold [117] so their shapes can be modeled and compared by using an elastic metric. For accessing k -NN faster, fast nearest neighbor (fastNN) algorithm organizes trajectory data in an Octree [118]. 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 is considered in [119]. Support Vector Machine (SVM) is trained to generate the hypervolume, and the inquiry trajectory is determined as outliers if it falls outside the hypervolume [120]. Structural Support Vector Machine (Structural SVM) is explored to detect social groups in crowds in [121]. Furthermore, SVM works with Graph Theory to cluster trajectories in [122].

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 which 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 at describing the sample from $\{x_1, x_2, \dots, x_n\}$ in a component of GMM as

$$P(x_j) = \sum_{i=1}^K \pi_i N(x_j; \mu_i, \Sigma_i), \quad (19)$$

where $N(x_j; \mu_i, \Sigma_i)$ is the probability density of the i th 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 [9]. 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 [68].

Bayesian inference classifies new coming data, and the classified ones update the probability functions of Bayesian inference. For samples $\{x_1, x_2, \dots, x_n\}$, 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 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 sample space. The clusters can be parameterized for classifying new inquiry data, e.g., Dirichlet Process Mixture model (DPMM) is used to represent all m clusters with parameterized indexes $\{\Theta_1, \Theta_2, \dots, \Theta_m\}$ in [28]. 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. In order to learn coupled spatial and temporal patterns, Hierarchical Dirichlet Process (HDP) algorithm is applied in [6]. Bayesian model is used to segment object by classifying trajectories, so that human motion is also detected [123].

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 such as deep generative model. It is trained to represent multivariate time series if trajectory data are generated as a vector [124], and a deep fully-connected Neural Network with weight decay and sparsity constraint transfers trajectory data from different viewpoints to a fixed viewpoint in compact representation [125].

In most cases, Neural Network is used to classify data. It can be viewed as

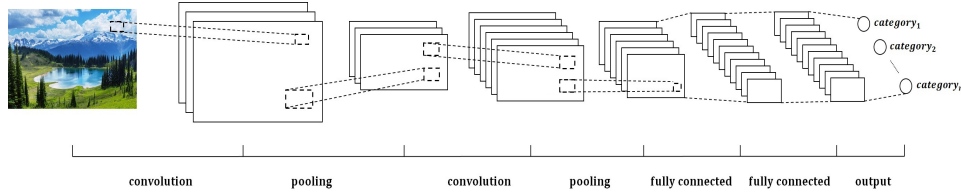


Figure 8: CNN is one of the classical models of Neural Network and widely used in images classification

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 clustering issue of computer vision. As Fig.8 shown, CNN is comprised by two convolutional layers, two pooling layers, two fully connected layers and one output layer which acts as an image classifier. CNN is employed for trajectory clustering in [23] and [24]. Furthermore, CNN also ranks the trajectory clustering results in [51]. 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 [126]. In order 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 which learns a more compact and powerful representation of trajectories [127]. Furthermore, DNN keeps the structural relationships between trajectories in [128], and mines the relationship between multiple features including spatiotemporal features, audio features and inter-class relationship to classify videos in [129] and [130]. Self-Organizing Map learns the similarities between trajectories in a 2-dimensional grid and each element of the grid indicates a specific prototype in [27] and [131]. In training steps, each training trajectory is trying to find the most suitable prototype in network, and adjust the neighbors of the matched one accordingly.

4.4. Discussion

Nearest Neighbor algorithm only considers the spatial relationships between a pair of trajectory data but ignores local characters. Statistical model makes up for this imperfection by combining them in a mixture model or inferring the relationships in Bayesian models. Neural Network considers the differences of trajectory data and requires a huge number of data to train it. Though the supervised methods obtain the classifiers by observing a number of training data,

overfitting problem may happen when the model overreacts training data.

5. Semi-supervised Algorithms of Trajectory Clustering

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 firstly, then inquiry data are kept sending to the trained model to make sure that it can be updated to outperform the previous model. Semi-supervised procedure needs only a small cost in terms of human classification efforts. This procedure not only avoids 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 firstly and the new inquiry ones are clustered to update the classifier automatically [4][8][9][132]. Detected anomaly trajectory data are used to recalculate the representation of anomaly trajectory clusters in [132]. Trajectory data of video are modeled as the combination of normal and abnormal patterns, and probabilistic terms characterize the patterns in [9]. From this modeling, the terms can be updated by the detected inquiry trajectory. In order to detect abnormal trajectories faster in complex scene, low-rank approximation is employed to describe trajectory data and the new detected abnormal ones update the threshold in [8].

Inspired by Hierarchical Frameworks, trajectories and the clusters are represented as a tree where children nodes indicate trajectories and roots denote the representations of the clusters in [133], [134] and [135]. A new cluster is created if no clusters close to the inquiry trajectory. Trajectory T is constructed as a vector of 2 dimensional coordinates $T = \{t_1, \dots, t_n\}$ where $t_j = \{x_j, y_j\}$. A representation of cluster is computed as $C_i = \{c_{i1}, \dots, c_{im}\}$, where $c_{ij} = \{x_{ij}, y_{ij}, \sigma_{ij}^2\}$ and σ_{ij}^2 is an approximation of the local variance of

the cluster i at time j . The inquiry trajectory is assigned to the nearest cluster and the corresponding cluster should be updated by the new one. For the nearest cluster point $c = \{x, y, \sigma^2\}$ to the point of trajectory $t = \{\hat{x}, \hat{y}\}$, c is updated as following

$$\begin{cases} x = (1 - \alpha)x + \alpha\hat{x} \\ y = (1 - \alpha)y + \alpha\hat{y} \\ \sigma^2 = (1 - \alpha)\sigma^2 + \alpha[dist(t_i, c_j)]^2, \end{cases} \quad (20)$$

where α is the update rate between 0 and 1.

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 [28]. Furthermore, in order to add new trajectory data, the previous samples and the new ones are sampled by Gibbs Sampling as

$$p(\eta_i | \eta_{-i}, y_{1:N+\phi}) = p(\eta_i | \eta_{1:N} = W_{1:N}, \eta_{-i}^{new}, y_{1:N+\phi}), \quad (21)$$

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 i th one. From Bayes' theorem, the cluster 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 need to be fixed is carrying out Gibbs Sampling on $\eta_{N+1:N+\phi}$ to compute $p(\eta_i | \eta_{1:N} = W_{1:N}, \eta_{-i}^{new})$.

6. Promising Future Directions and Tasks

In trajectory clustering, we saw that how to measure trajectories with different lengths is important. Thus, a representation method or feature descriptor is essential for trajectory clustering. In recent years, transforming trajectory data into other space are paid more attention, such as DFT which keeping data

information and unifying lengths of trajectory data [28]. For other preparation works, re-sampling is efficient for sparse scene [36], but it limits the robustness of model. Curve approximation fits the movement of trajectory [25][26]. 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 [136]. Besides grid-based DBSCAN, sub-trajectories are acted as the substitute for trajectory in [40], [69] and [76].

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 is fail or not. Thus, Spectral Clustering models need to handle the problem of constructing 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 have 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.

7. Conclusion

In this paper, we reviewed the methods of trajectory clustering. They are classified into three categories: unsupervised, supervised and semi-supervised algorithms. Unsupervised algorithms can be grouped into three sub-categories:

Densely Clustering models, Hierarchical Clustering models and Spectral Clustering models. Supervised algorithms are divided into Nearest Neighbor algorithms, statistical models and Neural Network. By means of a comprehensive analysis, we found that unsupervised algorithms have the disadvantages of high computation cost and heavy memory load, although there is no training data requirement and human experts supervising. Semi-supervised algorithms combine the advantages of both previous algorithms, and may result in more efficient methods. Finally, we proposed several promising future directions and tasks, and this paper could help readers to gain a thorough understanding of trajectory clustering.

References

References

- [1] Z. Chen, H. T. Shen, X. Zhou, Y. Zheng, X. Xie, Searching trajectories by locations: an efficiency study, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ACM, 2010, pp. 255–266.
- [2] S. Atev, O. Masoud, N. Papanikolopoulos, Learning traffic patterns at intersections by spectral clustering of motion trajectories, in: IROS, 2006, pp. 4851–4856.
- [3] X. Li, J. Han, J.-G. Lee, H. Gonzalez, Traffic density-based discovery of hot routes in road networks, in: International Symposium on Spatial and Temporal Databases, Springer, 2007, pp. 441–459.
- [4] S. Gurung, D. Lin, W. Jiang, A. Hurson, R. Zhang, Traffic information publication with privacy preservation, ACM Transactions on Intelligent Systems and Technology (TIST) 5 (3) (2014) 44.

- [5] F. I. Bashir, A. A. Khokhar, D. Schonfeld, Object trajectory-based activity classification and recognition using hidden markov models, *IEEE transactions on Image Processing* 16 (7) (2007) 1912–1919.
- [6] H. Wang, C. OSullivan, Globally continuous and non-markovian crowd activity analysis from videos, in: *European Conference on Computer Vision*, Springer, 2016, pp. 527–544.
- [7] T. Yao, Z. Wang, Z. Xie, J. Gao, D. D. Feng, Learning universal multiview dictionary for human action recognition, *Pattern Recognition* 64 (2017) 236–244.
- [8] L. Wang, M. Dong, Detection of abnormal human behavior using a matrix approximation-based approach, in: *Machine Learning and Applications (ICMLA)*, 2014 13th International Conference on, IEEE, 2014, pp. 324–329.
- [9] Y. Yuan, Y. Feng, X. Lu, Statistical hypothesis detector for abnormal event detection in crowded scenes, *IEEE transactions on cybernetics* 47 (11) (2017) 3597–3608.
- [10] W. Zhao, Z. Zhang, K. Huang, Gestalt laws based tracklets analysis for human crowd understanding, *Pattern Recognition* 75 (2018) 112–127.
- [11] R. Chaker, Z. Al Aghbari, I. N. Junejo, Social network model for crowd anomaly detection and localization, *Pattern Recognition* 61 (2017) 266–281.
- [12] S. Kumar, Y. Dai, H. Li, Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion, *Pattern Recognition* 71 (2017) 428–443.
- [13] N. Ferreira, J. T. Klosowski, C. E. Scheidegger, C. T. Silva, Vector field k-means: Clustering trajectories by fitting multiple vector fields, in: *Computer Graphics Forum*, Vol. 32, Wiley Online Library, 2013, pp. 201–210.

- [14] Y. Mo, D. Wu, Y. Du, Application of trajectory clustering and regionalization to ocean eddies in the south china sea, in: Spatial Data Mining and Geographical Knowledge Services (ICSDM), 2015 2nd IEEE International Conference on, IEEE, 2015, pp. 45–48.
- [15] G. Cai, K. Lee, I. Lee, A framework for mining semantic-level tourist movement behaviours from geo-tagged photos, in: Australasian Joint Conference on Artificial Intelligence, Springer, 2016, pp. 519–524.
- [16] H. Wang, D. Oneata, J. Verbeek, C. Schmid, A robust and efficient video representation for action recognition, *International Journal of Computer Vision* 119 (3) (2016) 219–238.
- [17] M. Nanni, D. Pedreschi, Time-focused clustering of trajectories of moving objects, *Journal of Intelligent Information Systems* 27 (3) (2006) 267–289.
- [18] X. Li, W. Hu, W. Hu, A coarse-to-fine strategy for vehicle motion trajectory clustering, in: 18th International Conference on Pattern Recognition (ICPR’06), Vol. 1, IEEE, 2006, pp. 591–594.
- [19] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise., in: *Kdd*, Vol. 96, 1996, pp. 226–231.
- [20] X. Wang, K. Tieu, E. Grimson, Learning semantic scene models by trajectory analysis, in: European conference on computer vision, Springer, 2006, pp. 110–123.
- [21] T. Xiang, S. Gong, Spectral clustering with eigenvector selection, *Pattern Recognition* 41 (3) (2008) 1012–1029.
- [22] Y.-J. Gao, C. Li, G.-C. Chen, L. Chen, X.-T. Jiang, C. Chen, Efficient k-nearest-neighbor search algorithms for historical moving object trajectories, *Journal of Computer Science and Technology* 22 (2) (2007) 232–244.

- [23] Z. Wu, Y. Fu, Y.-G. Jiang, L. Sigal, Harnessing object and scene semantics for large-scale video understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3112–3121.
- [24] K. Cho, X. Chen, Classifying and visualizing motion capture sequences using deep neural networks, in: Computer Vision Theory and Applications (VISAPP), 2014 International Conference on, Vol. 2, IEEE, 2014, pp. 122–130.
- [25] T. Zhang, H. Lu, S. Z. Li, Learning semantic scene models by object classification and trajectory clustering, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 1940–1947.
- [26] R. R. Sillito, R. B. Fisher, Semi-supervised learning for anomalous trajectory detection., in: BMVC, Vol. 1, 2008, pp. 035–1.
- [27] A. Naftel, S. Khalid, Motion trajectory learning in the dft-coefficient feature space, in: Fourth IEEE International Conference on Computer Vision Systems (ICVS'06), IEEE, 2006, pp. 47–47.
- [28] W. Hu, X. Li, G. Tian, S. Maybank, Z. Zhang, An incremental dpmm-based method for trajectory clustering, modeling, and retrieval, IEEE transactions on pattern analysis and machine intelligence 35 (5) (2013) 1051–1065.
- [29] H. Hu, J. Feng, J. Zhou, Exploiting unsupervised and supervised constraints for subspace clustering, IEEE transactions on pattern analysis and machine intelligence 37 (8) (2015) 1542–1557.
- [30] I. Akhter, Y. Sheikh, S. Khan, T. Kanade, Trajectory space: A dual representation for nonrigid structure from motion, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (7) (2011) 1442–1456.

- [31] F. I. Bashir, A. A. Khokhar, D. Schonfeld, Real-time motion trajectory-based indexing and retrieval of video sequences, *IEEE Transactions on Multimedia* 9 (1) (2007) 58–65.
- [32] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, C.-W. Ngo, Trajectory-based modeling of human actions with motion reference points, in: *European Conference on Computer Vision*, Springer, 2012, pp. 425–438.
- [33] J. Schenk, G. Rigoll, Novel hybrid nn/hmm modelling techniques for on-line handwriting recognition, in: *Tenth International Workshop on Frontiers in Handwriting Recognition*, Suvisoft, 2006.
- [34] W.-G. Liou, C.-Y. Hsieh, W.-Y. Lin, Trajectory-based sign language recognition using discriminant analysis in higher-dimensional feature space, in: *2011 IEEE International Conference on Multimedia and Expo*, IEEE, 2011, pp. 1–4.
- [35] N. Piotto, N. Conci, F. G. De Natale, Syntactic matching of trajectories for ambient intelligence applications, *IEEE Transactions on Multimedia* 11 (7) (2009) 1266–1275.
- [36] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: *BMVC 2009-British Machine Vision Conference*, BMVA Press, 2009, pp. 124–1.
- [37] P. Ochs, J. Malik, T. Brox, Segmentation of moving objects by long term video analysis, *IEEE transactions on pattern analysis and machine intelligence* 36 (6) (2014) 1187–1200.
- [38] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, pp. 2790–2797.
- [39] O. Cappé, S. J. Godsill, E. Moulines, An overview of existing methods and recent advances in sequential monte carlo, *Proceedings of the IEEE* 95 (5) (2007) 899–924.

- [40] J.-G. Lee, J. Han, K.-Y. Whang, Trajectory clustering: a partition-and-group framework, in: Proceedings of the 2007 ACM SIGMOD international conference on Management of data, ACM, 2007, pp. 593–604.
- [41] A. Anagnostopoulos, M. Vlachos, M. Hadjieleftheriou, E. Keogh, P. S. Yu, Global distance-based segmentation of trajectories, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2006, pp. 34–43.
- [42] Z. Yan, C. Parent, S. Spaccapietra, D. Chakraborty, A hybrid model and computing platform for spatio-semantic trajectories, in: Extended Semantic Web Conference, Springer, 2010, pp. 60–75.
- [43] D. R. Faria, J. Dias, 3d hand trajectory segmentation by curvatures and hand orientation for classification through a probabilistic approach, in: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2009, pp. 1284–1289.
- [44] Z. Zhang, K. Huang, T. Tan, L. Wang, Trajectory series analysis based event rule induction for visual surveillance, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.
- [45] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: a survey, *Foundations and trends® in computer graphics and vision* 3 (3) (2008) 177–280.
- [46] B. T. Morris, M. M. Trivedi, A survey of vision-based trajectory learning and analysis for surveillance, *IEEE transactions on circuits and systems for video technology* 18 (8) (2008) 1114–1127.
- [47] N. Brandle, D. Bauer, S. Seer, Track-based finding of stopping pedestrians-a practical approach for analyzing a public infrastructure, in: 2006 IEEE Intelligent Transportation Systems Conference, IEEE, 2006, pp. 115–120.

- [48] B. T. Morris, M. M. Trivedi, Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach, *IEEE transactions on pattern analysis and machine intelligence* 33 (11) (2011) 2287–2301.
- [49] Q.-Y. Zhou, V. Koltun, Dense scene reconstruction with points of interest, *ACM Transactions on Graphics (TOG)* 32 (4) (2013) 112.
- [50] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: *Computer Vision and Pattern Recognition (CVPR)*, 2011 IEEE Conference on, IEEE, 2011, pp. 3169–3176.
- [51] K. Fragkiadaki, P. Arbelaez, P. Felsen, J. Malik, Learning to segment moving objects in videos, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4083–4090.
- [52] A. Klaser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: *BMVC 2008-19th British Machine Vision Conference*, British Machine Vision Association, 2008, pp. 275–1.
- [53] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: *Computer Vision and Pattern Recognition*, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.
- [54] G. Willems, T. Tuytelaars, L. Van Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: *European conference on computer vision*, Springer, 2008, pp. 650–663.
- [55] P. Matikainen, M. Hebert, R. Sukthankar, Trajectons: Action recognition through the motion analysis of tracked features, in: *Computer Vision Workshops (ICCV Workshops)*, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 514–521.
- [56] P. Matikainen, M. Hebert, R. Sukthankar, Representing pairwise spatial and temporal relations for action recognition, in: *European Conference on Computer Vision*, Springer, 2010, pp. 508–521.

- [57] H. Uemura, S. Ishikawa, K. Mikolajczyk, Feature tracking and motion compensation for action recognition., in: BMVC, 2008, pp. 1–10.
- [58] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, J. Li, Hierarchical spatio-temporal context modeling for action recognition, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 2004–2011.
- [59] F. Wang, Y.-G. Jiang, C.-W. Ngo, Video event detection using motion relativity and visual relatedness, in: Proceedings of the 16th ACM international conference on Multimedia, ACM, 2008, pp. 239–248.
- [60] E. Vig, M. Dorr, D. Cox, Space-variant descriptor sampling for action recognition based on saliency and eye movements, in: European conference on computer vision, Springer, 2012, pp. 84–97.
- [61] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, International journal of computer vision 103 (1) (2013) 60–79.
- [62] B. Morris, M. Trivedi, Learning trajectory patterns by clustering: Experimental studies and comparative evaluation, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 312–319.
- [63] Z. Zhang, K. Huang, T. Tan, Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes, in: 18th International Conference on Pattern Recognition (ICPR’06), Vol. 3, IEEE, 2006, pp. 1135–1138.
- [64] S. Atev, G. Miller, N. P. Papanikolopoulos, Clustering of vehicle trajectories, IEEE Transactions on Intelligent Transportation Systems 11 (3) (2010) 647–657.
- [65] M.-Y. Liu, O. Tuzel, S. Ramalingam, R. Chellappa, Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a

matroid constraint, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (1) (2014) 99–112.

- [66] J. Chen, R. Wang, L. Liu, J. Song, Clustering of trajectories based on hausdorff distance, in: *Electronics, Communications and Control (ICECC), 2011 International Conference on*, IEEE, 2011, pp. 1940–1944.
- [67] Z. Shao, Y. Li, On integral invariants for effective 3-d motion trajectory matching and recognition, *IEEE transactions on cybernetics* 46 (2) (2016) 511–523.
- [68] M. A. Bautista, A. Hernández-Vela, S. Escalera, L. Igual, O. Pujol, J. Moya, V. Violant, M. T. Anguera, A gesture recognition system for detecting behavioral patterns of adhd, *IEEE transactions on cybernetics* 46 (1) (2016) 136–147.
- [69] J.-G. Lee, J. Han, X. Li, H. Gonzalez, Traclass: trajectory classification using hierarchical region-based and trajectory-based clustering, *Proceedings of the VLDB Endowment* 1 (1) (2008) 1081–1094.
- [70] Z. Li, J.-G. Lee, X. Li, J. Han, Incremental clustering for trajectories, in: *International Conference on Database Systems for Advanced Applications*, Springer, 2010, pp. 32–46.
- [71] A. T. Palma, V. Bogorny, B. Kuijpers, L. O. Alvares, A clustering-based approach for discovering interesting places in trajectories, in: *Proceedings of the 2008 ACM symposium on Applied computing*, ACM, 2008, pp. 863–868.
- [72] Y. Bu, L. Chen, A. W.-C. Fu, D. Liu, Efficient anomaly monitoring over moving object trajectory streams, in: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2009, pp. 159–168.
- [73] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, F. Giannotti, Interactive visual clustering of large collections of trajectories, in:

Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on, IEEE, 2009, pp. 3–10.

- [74] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, H. T. Shen, Discovery of convoys in trajectory databases, *Proceedings of the VLDB Endowment* 1 (1) (2008) 1068–1080.
- [75] H. Wang, C. Schmid, Action recognition with improved trajectories, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.
- [76] H. S. Khaing, T. Thein, An efficient clustering algorithm for moving object trajectories, in: *3rd International Conference on Computational techniques and Artificial Intelligence (ICCTAI 2014)* February, 2014, pp. 11–12.
- [77] Y. Zheng, Trajectory data mining: an overview, *ACM Transactions on Intelligent Systems and Technology (TIST)* 6 (3) (2015) 29.
- [78] K. Deng, K. Xie, K. Zheng, X. Zhou, Trajectory indexing and retrieval, in: *Computing with spatial trajectories*, Springer, 2011, pp. 35–60.
- [79] Y. Zheng, X. Xie, W.-Y. Ma, Geolife: A collaborative social networking service among user, location and trajectory., *IEEE Data Eng. Bull.* 33 (2) (2010) 32–39.
- [80] H. Xu, Y. Zhou, W. Lin, H. Zha, Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4328–4336.
- [81] N. Suzuki, K. Hirasawa, K. Tanaka, Y. Kobayashi, Y. Sato, Y. Fujino, Learning motion patterns and anomaly detection by human trajectory analysis, in: *2007 IEEE International Conference on Systems, Man and Cybernetics*, IEEE, 2007, pp. 498–503.

- [82] L. Galluccio, O. Michel, P. Comon, A. O. Hero, Graph based k-means clustering, *Signal Processing* 92 (9) (2012) 1970–1984.
- [83] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, S. Maybank, A system for learning statistical motion patterns, *IEEE transactions on pattern analysis and machine intelligence* 28 (9) (2006) 1450–1464.
- [84] J. Melo, A. Naftel, A. Bernardino, J. Santos-Victor, Detection and classification of highway lanes using vehicle motion trajectories, *IEEE Transactions on intelligent transportation systems* 7 (2) (2006) 188–200.
- [85] Y. Zhou, S. Yan, T. S. Huang, Detecting anomaly in videos from trajectory similarity analysis, in: *2007 IEEE International Conference on Multimedia and Expo*, IEEE, 2007, pp. 1087–1090.
- [86] Z. Kim, Real time object tracking based on dynamic feature grouping with background subtraction, in: *Computer Vision and Pattern Recognition*, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.
- [87] N. Pelekis, I. Kopanakis, E. E. Kotsifakos, E. Frentzos, Y. Theodoridis, Clustering trajectories of moving objects in an uncertain world., in: *ICDM*, Vol. 9, 2009, pp. 417–427.
- [88] N. Pelekis, I. Kopanakis, E. E. Kotsifakos, E. Frentzos, Y. Theodoridis, Clustering uncertain trajectories, *Knowledge and Information Systems* 28 (1) (2011) 117–147.
- [89] D. Simonnet, E. Anquetil, M. Bouillon, Multi-criteria handwriting quality analysis with online fuzzy models, *Pattern Recognition* 69 (2017) 310–324.
- [90] Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma, Mining interesting locations and travel sequences from gps trajectories, in: *Proceedings of the 18th international conference on World wide web*, ACM, 2009, pp. 791–800.
- [91] K. Zheng, Y. Zheng, N. J. Yuan, S. Shang, On discovery of gathering patterns from trajectories, in: *Data Engineering (ICDE)*, 2013 IEEE 29th International Conference on, IEEE, 2013, pp. 242–253.

- [92] A. Gaidon, Z. Harchaoui, C. Schmid, Activity representation with motion hierarchies, *International journal of computer vision* 107 (3) (2014) 219–238.
- [93] M. Raptis, I. Kokkinos, S. Soatto, Discovering discriminative action parts from mid-level video representations, in: *Computer Vision and Pattern Recognition, 2012 IEEE Conference on*, IEEE, 2012, pp. 1242–1249.
- [94] S. S. Tabatabaei, M. Coates, M. Rabbat, Ganc: Greedy agglomerative normalized cut, *arXiv preprint arXiv:1105.0974*.
- [95] B. Ni, P. Moulin, X. Yang, S. Yan, Motion part regularization: Improving action recognition via trajectory selection, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3698–3706.
- [96] G. Antonini, J.-P. Thiran, Counting pedestrians in video sequences using trajectory clustering, *IEEE Transactions on Circuits and Systems for Video Technology* 16 (8) (2006) 1008–1020.
- [97] F. Turchini, L. Seidenari, A. Del Bimbo, Understanding sport activities from correspondences of clustered trajectories, in: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 43–50.
- [98] W. Hu, D. Xie, Z. Fu, W. Zeng, S. Maybank, Semantic-based surveillance video retrieval, *IEEE Transactions on image processing* 16 (4) (2007) 1168–1181.
- [99] E. Brunskill, T. Kollar, N. Roy, Topological mapping using spectral clustering and classification, in: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2007, pp. 3491–3496.
- [100] T. Brox, J. Malik, Object segmentation by long term analysis of point trajectories, in: *European conference on computer vision*, Springer, 2010, pp. 282–295.

- [101] F. Lauer, C. Schnörr, Spectral clustering of linear subspaces for motion segmentation, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 678–685.
- [102] Z. Zhang, K. Huang, T. Tan, P. Yang, J. Li, Red-sfa: Relation discovery based slow feature analysis for trajectory clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 752–760.
- [103] D. Hong, Q. Gu, K. Whitehouse, High-dimensional time series clustering via cross-predictability, in: Artificial Intelligence and Statistics, 2017, pp. 642–651.
- [104] G. Chen, G. Lerman, Spectral curvature clustering (scc), International Journal of Computer Vision 81 (3) (2009) 317–330.
- [105] H. Ergezer, K. Leblebicioğlu, Anomaly detection and activity perception using covariance descriptor for trajectories, in: European Conference on Computer Vision, Springer, 2016, pp. 728–742.
- [106] Y. Yi, M. Lin, Human action recognition with graph-based multiple-instance learning, Pattern Recognition 53 (2016) 148–162.
- [107] A. E. Brouwer, W. H. Haemers, Spectra of graphs, Springer Science & Business Media, 2011.
- [108] W. Chen, J. J. Corso, Action detection by implicit intentional motion clustering, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 3298–3306.
- [109] L. Lin, Y. Lu, Y. Pan, X. Chen, Integrating graph partitioning and matching for trajectory analysis in video surveillance, IEEE Transactions on Image Processing 21 (12) (2012) 4844–4857.
- [110] D. Guo, S. Liu, H. Jin, A graph-based approach to vehicle trajectory analysis, Journal of Location Based Services 4 (3-4) (2010) 183–199.

- [111] G. Palou, P. Salembier, Hierarchical video representation with trajectory binary partition tree, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2099–2106.
- [112] X. Liu, L. Lin, S.-C. Zhu, H. Jin, Trajectory parsing by cluster sampling in spatio-temporal graph, in: *Computer Vision and Pattern Recognition*, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 739–746.
- [113] I. N. Junejo, H. Foroosh, Euclidean path modeling for video surveillance, *Image and Vision computing* 26 (4) (2008) 512–528.
- [114] M. Cho, K. MuLee, Authority-shift clustering: Hierarchical clustering by authority seeking on graphs, in: *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference on, IEEE, 2010, pp. 3193–3200.
- [115] M. Keuper, B. Andres, T. Brox, Motion trajectory segmentation via minimum cost multicut, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3271–3279.
- [116] N. Pelekis, P. Tampakis, M. Voudas, C. Doukeridis, Y. Theodoridis, On temporal-constrained sub-trajectory cluster analysis, *Data Mining and Knowledge Discovery* (2017) 1–37.
- [117] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, A. Del Bimbo, 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold, *IEEE transactions on cybernetics* 45 (7) (2015) 1340–1352.
- [118] S. Poularakis, I. Katsavounidis, Low-complexity hand gesture recognition system for continuous streams of digits and letters, *IEEE transactions on cybernetics* 46 (9) (2016) 2094–2108.
- [119] E. Frentzos, K. Gratsias, N. Pelekis, Y. Theodoridis, Algorithms for nearest neighbor search on moving object trajectories, *Geoinformatica* 11 (2) (2007) 159–193.

- [120] C. Piciarelli, C. Micheloni, G. L. Foresti, Trajectory-based anomalous event detection, *IEEE Transactions on Circuits and Systems for video Technology* 18 (11) (2008) 1544–1554.
- [121] F. Solera, S. Calderara, R. Cucchiara, Socially constrained structural learning for groups detection in crowd, *IEEE transactions on pattern analysis and machine intelligence* 38 (5) (2016) 995–1008.
- [122] D. Singh, C. K. Mohan, Graph formulation of video activities for abnormal activity recognition, *Pattern Recognition* 65 (2017) 265–272.
- [123] M. Devanne, S. Berretti, P. Pala, H. Wannous, M. Daoudi, A. Del Bimbo, Motion segment decomposition of rgb-d sequences for human behavior understanding, *Pattern Recognition* 61 (2017) 222–233.
- [124] C. Yuan, A. Chakraborty, Deep convolutional factor analyser for multivariate time series modeling, in: *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, IEEE, 2016, pp. 1323–1328.
- [125] H. Rahmani, A. Mian, M. Shah, Learning a deep model for human action recognition from novel viewpoints, *IEEE transactions on pattern analysis and machine intelligence* 40 (3) (2018) 667–681.
- [126] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, A. G. Hauptmann, Devnet: A deep event network for multimedia event detection and evidence recounting, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2568–2577.
- [127] M. Hasan, A. K. Roy-Chowdhury, A continuous learning framework for activity recognition using deep hybrid feature models, *IEEE Transactions on Multimedia* 17 (11) (2015) 1909–1922.
- [128] Y. Shi, W. Zeng, T. Huang, Y. Wang, Learning deep trajectory descriptor for action recognition in videos using deep neural networks, in: *2015 IEEE International Conference on Multimedia and Expo*, IEEE, 2015, pp. 1–6.

- [129] Z. Wu, Y.-G. Jiang, J. Wang, J. Pu, X. Xue, 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, 2014, pp. 167–176.
- [130] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, S.-F. Chang, Exploiting feature and class relationships in video categorization with regularized deep neural networks, arXiv preprint arXiv:1502.07209.
- [131] T. Schreck, J. Bernard, T. Von Landesberger, J. Kohlhammer, Visual cluster analysis of trajectory data with interactive kohonen maps, *Information Visualization* 8 (1) (2009) 14–29.
- [132] R. Laxhammar, G. Falkman, Online learning and sequential anomaly detection in trajectories, *IEEE transactions on pattern analysis and machine intelligence* 36 (6) (2014) 1158–1173.
- [133] C. Piciarelli, G. L. Foresti, On-line trajectory clustering for anomalous events detection, *Pattern Recognition Letters* 27 (15) (2006) 1835–1842.
- [134] D. Kulić, W. Takano, Y. Nakamura, Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains, *The International Journal of Robotics Research* 27 (7) (2008) 761–784.
- [135] X. Li, V. Ceikute, C. S. Jensen, K.-L. Tan, Effective online group discovery in trajectory databases, *IEEE Transactions on Knowledge and Data Engineering* 25 (12) (2013) 2752–2766.
- [136] O. Uncu, W. A. Gruver, D. B. Kotak, D. Sabaz, Z. Alibhai, C. Ng, Gridb-scan: Grid density-based spatial clustering of applications with noise, in: Systems, Man and Cybernetics, 2006. SMC’06. IEEE International Conference on, Vol. 4, IEEE, 2006, pp. 2976–2981.