International Journal of Cognitive Informatics and Natural Intelligence
Volume 15 • Issue 2 • April-June 2021

Learning Trajectory Patterns via Canonical Correlation Analysis

Ping Huang, School of Mathematical Sciences, South China University of Technology, Guangzhou, China

Jinliang Lu, School of Mathematical Sciences, South China University of Technology, Guangzhou, China

ABSTRACT

A substantial body of research has been devoted to the analysis of motion trajectories. Usually, a motion trajectory consists of a set of coordinates, which is called a raw trajectory. In this paper, the authors first use vectors for some artificially constructed global features, such as the mean discrete curvature and standard deviation of acceleration, to represent the raw trajectory data, and then apply a multiset canonical correlation analysis method to extract latent features from the artificially constructed features. The performance of the latent features is then measured by evaluating the accuracy and F1 score of a gradient boosting decision tree model for different datasets, which include paired sample datasets and unpaired sample datasets. The experimental results show that the classifier performance for MCCA features is much better than that obtained for the artificially constructed features, such as that for the motion distance or mean velocity.

KEYWORDS

Discrete Curvature, Multiset Canonical Correlation Analysis, Raw Trajectory Classification, Unpaired Sample

1. INTRODUCTION

At present, it is important to analyze motion trajectories. By analyzing motion trajectories, different types of trajectories can be identified from a large number of trajectory data, such as distinguishing the choice of transportation - car or bus, classifying the traffic level-congested or smooth.

Trajectory data are generally a sequence of spatial location information data recorded alongside time that can be collected by an algorithm (K Li, Wang, & S Li (2019)). As given by Definition 1 in (Silva, Petry, & Bogorny (2019)), the raw trajectory data are the latitude and longitude information or the location of pixels in the image recorded alongside the time.

Trajectory classification is a very important part of trajectory analysis. Unlike other fields such as image recognition, trajectory classification focuses on how to extract effective features from the spatial position information sequence instead of constructing novel and efficient classifiers. Then, these features are used to train common classifiers such as Logistic Regression (LG), Gradient Boosting Decision Tree (GBDT), etc.

In this paper, we first extract the global features of a spatial position information sequence: the mean, standard deviation and entropy of the discrete curvature, the mean and standard deviation of the velocity and acceleration, as well as the length of the trajectory and motion duration. Second, we use multiset canonical correlation analysis (MCCA) to re-extract features from the artificially constructed features. Finally, we use the MCCA features to train a Gradient Boosting Decision Tree model, and conduct experiments using 7 different datasets to analyze the performance of the MCCA features.

DOI: 10.4018/IJCINI.20210401.oa1

The remainder of the paper is organized as follows: section 2 introduces the related research work for trajectory classification, the Gradient Boosting Decision Tree and canonical correlation analysis; section 3 introduces 9 features constructed in this work; section 4 introduces MCCA and how we can carry out MCCA for datasets with unpaired samples; section 5 includes experiments and results; section 6 includes the conclusion and future directions.

2. RELATED WORK

Recently, many studies have been carried out for trajectory classification. (Silva, Petry, & Bogorny (2019)) summarized three types of trajectory features: global features, local features, as well as global and local features, and presented an experimental comparison for several datasets with methods proposed by others. (Zheng, Li, Chen, Xie, & Ma (2008)) and (Sharma, Vyas, Schieder, & Akasapu (2010)) used global features to classify trajectories. The difference being that (Zheng, Li, Chen, Xie, & Ma (2008)) focuses on the transportation mode classification, so global features extracted by (Zheng, Li, Chen, Xie, & Ma (2008)) and (Sharma, Vyas, Schieder, & Akasapu (2010)) are different. Different studies show different choices for the classifiers. (Mlıch & Chmelar (2008)) and (Bashir, Khokhar, & Schonfeld (2007)) used Hidden Markov Models(HMM) to complete the classification task, while (Wang, Chu, Jiang & Li (2019)) used a Naive Bayesian Model (NBM) and (Liu & Lee (2017)) used BiLSTM. (Xiao, Wang, Fu, & Wu (2017)) used a Multilayer-Perceptron (MLP) and achieved a state-of-the-art accuracy of 75.56% and an F1 score of 73.83% for raw trajectories dataset “Geolife[1]” and achieved a state-of-the-art accuracy of 93.58% and an F1 score of 93.58% for the raw trajectories dataset “Animals[2]”, while (Ferrero, Alvares, Zalewski, & Bogorny (2018)) proposed a new method called MOVELETS to achieve the goal of robust trajectory classification. With a Support Vector Machine, the MOVELETS method achieved a state-of-the-art accuracy of 92.30% and an F1 score of 90.82% for the multiple-aspect trajectories dataset “Animals[2]”. The Gradient Boosting Decision Tree algorithm, which is widely used in many fields, and, in this paper, was proposed by (Friedman, (2001)). (Son, Jung, Park, & Han (2015)) applied the GBDT algorithm to object tracking tasks, and (Rao et al. (2019)) applied it to feature selection problems. At the same time, there are many optimizations for GBDT, such as the LightGBM algorithm proposed by (Ke et al. (2017)). Other optimization algorithms such as (Wang et al. (2018)) and (Wang et al. (2019)) can also be used. Canonical correlation analysis (CCA) was proposed by Hotelling, H., which is fully described in (Hotelling (1935)) and (Hotelling (1992)). There exists an expansion of CCA for a multiset called multiset canonical correlation analysis (MCCA). (Lisanti, Karaman, & Masi(2017)) used MCCA for person reidentification tasks, while (Kanatsoulis, Fu, Sidiropoulos, & Hong (2018)) used MCCA for largescale data, and (Nielsen, (2002)) used MCCA for the GIS system, with discussion of the different restrictions on the MCCA optimization problem.

3. FEATURES OF A TRAJECTORY

Assume that a raw trajectory is represented by a set of points $T = \left\{\left(x_i, y_i, t_i\right)\right\}_{i=0}^{n}$, where the set of times $\left\{t_i\right\}_{i=0}^{n}$ occur in ascending order. Although we can use the method in (K Li, Chen, W Li, He, & Xue (2018)) to model trajectory data that is ordered in time, similar to (Zheng, Li, Chen, Xie, & Ma (2008)) and (Sharma, Vyas, Schieder, & Akasapu (2010)), we will construct several global features in this section. Then, a vector of these features, which is called original features, is used in the experimental section to represent trajectories.

3.1. Curvature

In curve theory, curvature is one of the invariants of a parametric curve. Although (Ujiie, Kato, Sato, & Matsuoka (2012)) use discrete curvature for curved profile generation, we use it as a feature describing the degree of bending of a curve on a point. Then, we calculate the mean, standard deviation and entropy for the discrete curvatures as the global features for a whole curve. Since a raw trajectory is represented by a set of discrete points, we first apply the interpolation method: (1) Use linear interpolation when the set contains 2 points; (2) Using quadratic spline interpolation when the set contains 3 points; (3) Use cubic spline interpolation when the set contains ≤ 3 points, before calculating the discrete curvature on the point $\left(x_i, y_i, t_i\right)$. For the set that contains only 1 point, we just simply set the discrete curvature to 0. Now, we can calculate the discrete curvature on the point $\left(x_i, y_i, t_i\right)$ through the formula (1):

$$\kappa_i = \left| \frac{s_x''(t)s_y'(t) - s_x'(t)s_y''(t)}{\left(s_x'(t)^2 + s_y'(t)^2\right)^{3/2}} \right|_{t=t_i}, i = 0, 1, \ldots, n\# \tag{1}$$

Then, we calculate the mean and standard deviation f discrete curvatures through formulas (2) and (3):

$$E\left(\kappa\right) = \frac{1}{n+1}\sum_{i=0}^{n}\kappa_i \# \tag{2}$$

$$S\left(\kappa\right) = \sqrt{\frac{1}{n+1}\sum_{i=0}^{n}\left(\kappa_i - E(\kappa)\right)^2} \# \tag{3}$$

Because calculation of the entropy of the discrete curvatures requires probability distribution information, we first divide the interval $\left[0, +\infty\right)$ into n parts (4):

$$\begin{aligned} A &= \left[0, \frac{k}{n-1}\right) \cup \left[\frac{k}{n-1}, \frac{2k}{n-1}\right) \cup \cdots \cup \left[\frac{(n-2)k}{n-1}, k\right) \cup \left[k, +\infty\right) \\ &= A_1 \cup A_2 \cup \cdots \cup A_{n-1} \cup A_n \# \\ &= \bigcup_{i=1}^{n} A_i \end{aligned} \tag{4}$$

where we use $k = 1$ and $n = 101$ in the experimenal section. Then, we count the number of discrete curvatures falling into interval A_i, and obtain the frequency information (5):

$$p_i = \frac{N_i}{n} \ and \ \sum_{i=1}^{n} p_i = 1\# \tag{5}$$

With the frequency information, the entropy can be calculated through formula (6):

$$Ent\left(\kappa\right) = -\sum_{i=1, p_i \neq 0}^{n} p_i \log_2\left(p_i\right)\# \tag{6}$$

3.2. Velocity and Acceleration

Velocity and acceleration are important characteristics of moving targets, especially vehicles and pedestrians. Additionally, due to the type of datasets we use to conduct experiments, extraction of global features from velocity and acceleration may become necessary.

Ignoring the number of points in the set of a trajectory, the velocity between 2 adjacent points can be calculated through formula (7):

$$v_i = \sqrt{v_{x,i}^2 + v_{y,i}^2}, i = 0,1,\ldots,n-1 where v_{k,i} = \frac{k_{i+1} - k_i}{t_{i+1} - ti}, k \in \left\{x, y\right\}\# \tag{7}$$

According to the sequence of velocity, we can calculate the acceleration $a_i, i = 0,1,\ldots,n-2$, and, by using the same formulas (2) and (3), we obtain $\mathrm{E}\left(v\right)$, $\mathrm{S}\left(v\right)$, $\mathrm{E}\left(a\right)$ and $\mathrm{S}\left(a\right)$.

3.3. Length and Motion Duration

The final global features we use are the trajectory length and motion duration. Instead of integrating the interpolation function to approximate the local arc length, we just calculate the Euclidean distance 2 adjacent points and sum them to obtain the trajectory length, see formula (8). The motion duration is easy to calculate from difference between the end time and start time $T = t_n - t_0$:

$$\tilde{s}_i = \sqrt{\left(x_{i+1} - x_i\right)^2 + \left(y_{i+1} - y_i\right)^2}, i = 0,1,\ldots,n-1 \ and \ \tilde{s} = \sum_{i=0}^{n-1} \tilde{s}_i\# \tag{8}$$

Finally, vector (9), which consists of these artificially constructed features or original features, is used to represent the trajectory:

$$\left(Ent\left(\kappa\right), E\left(\kappa\right), S\left(\kappa\right), E\left(v\right), S\left(v\right), E\left(a\right), S\left(a\right), \tilde{s}, T\right)\# \tag{9}$$

4. CANONICAL CORRELATION ANALYSIS

Canonical correlation analysis aims to find the linear combinations $U = a^T X^{(1)}$ and $V = b^T X^{(2)}$ with unit variances of two random vectors $X^{(1)}$ and $X^{(2)}$ that maximize the correlation $Corr\left(U,V\right) = \frac{a^T \Sigma_{12} b}{\sqrt{a^T \Sigma_{11} a}\sqrt{b^T \Sigma_{22} b}}$. This is equivalent to the optimization problem (10):

$$\begin{aligned} \max_{a,b} \quad & Cov\left(U,V\right) = a^T \Sigma_{12} b \\ s.t. \quad & a^T \Sigma_{11} a = 1\# \\ & b^T \Sigma_{22} b = 1 \end{aligned} \tag{10}$$

To solve the optimization problem (10), we obtain the i-th canonical variable pair having unit variances that maximize $\mathrm{Corr}\left(U_i, V_i\right)$ among all choices uncorrelated with the previous $\left(i-1\right)$-th canonical variable pair, where $i = 1, 2, \ldots, p$. Finally, we use the random vectors $U = \left[U_1, U_2, \ldots, U_p\right]$ and $V = \left[V_1, V_2, \ldots, V_p\right]$ to represent $X^{(1)}$ and $X^{(2)}$.

4.1. Multi-Set Canonical Correlation Analysis

Multiset canonical correlation analysis is a natural expansion from double sets to multiple sets. Similar to the optimization problem (10). The optimization problem for MCCA is shown as follows:

$$\begin{aligned} \max_{\alpha_1,\ldots,\alpha_n} \quad & \sum_{i,j} \alpha_i^T S_{ij} \alpha_j \\ s.t. \quad & \sum_{i,j} \alpha_i^T S_{ii} \alpha_i = 1 \end{aligned} \# \tag{11}$$

where S_{ij} represents the sample covariance matrix between $X^{(i)}$ and $X^{(j)}$.

$$\boldsymbol{S}_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} \left(\boldsymbol{x}_k^{(i)} - \bar{\boldsymbol{x}}^{(i)}\right)\left(\boldsymbol{x}_k^{(j)} - \bar{\boldsymbol{x}}^{(j)}\right)^{\mathsf{T}}$$

Using a Lagrange multiplier method, the optimization problem (11) changes into the Lagrange function (12)

$$L\left(\pm_1, \ldots, \pm_n, \lambda\right) = \sum_{i,j} \pm_i^{\mathsf{T}} \boldsymbol{S}_{ij} \pm_j - \lambda \left(\sum_{i,j} \pm_i^{\mathsf{T}} \boldsymbol{S}_{ii} \pm_i - 1\right) \# \tag{12}$$

with an optimal solution satisfying the KKT conditions (13):

$$\frac{\partial L\left(\alpha_1, \ldots, \alpha_n, \lambda\right)}{\partial \alpha_i} = \sum_{j=1}^{n} S_{ij} \alpha_j - \lambda S_{ii} \alpha_i = 0 \# \tag{13}$$

Finally, we obtain the generalized eigenvalue problem (14):

$$\begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1n} \\ S_{21} & S_{22} & \cdots & S_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ S_{n1} & S_{n2} & \cdots & S_{nn} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} = \lambda \begin{pmatrix} S_{11} & & & \\ & S_{22} & & \\ & & \ddots & \\ & & & S_{nn} \end{pmatrix} \begin{pmatrix} \pm_1 \\ \pm_2 \\ \vdots \\ \pm_n \end{pmatrix} \# \tag{14}$$

4.2. Analysis for MCCA

We assume that there are m random samples for each of variable $\boldsymbol{X}^{(i)}$; so, $\boldsymbol{X}^{(i)}$ can be assembled into a data matrix $\boldsymbol{D}^{(i)} = \left[\boldsymbol{x}_1^{(i)}, \boldsymbol{x}_2^{(i)}, \boldsymbol{x}_m^{(i)}\right]$. By solving problem (14), the eigenvectors matrix $\boldsymbol{A} = \left[\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_n\right]$, is obtained where $A_i = \left[\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,p}\right]$. Moreover, the eigenvector $\left[\alpha_{1,j}, \alpha_{2,j}, \alpha_{n,j}\right]^T$ corresponds to the eigenvalue λ_j, where λ_j satisfies $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$. Therefore, there will be a data matrix $\boldsymbol{D}_{mcca}^{(i)}$ for the canonical correlation variables (15):

$$D_{mcca}^{(i)} = A_i^T D^{(i)} = \begin{pmatrix} \alpha_{i,1} x_1^{(i)} & \alpha_{i,1} x_2^{(i)} & \cdots & \alpha_{i,1} x_m^{(i)} \\ \alpha_{i,2} x_1^{(i)} & \alpha_{i,2} x_2^{(i)} & \cdots & \alpha_{i,2} x_m^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{i,p} x_1^{(i)} & \alpha_{i,p} x_2^{(i)} & \cdots & \alpha_{i,p} x_m^{(i)} \end{pmatrix}, i = 1, 2, \ldots, n \# \tag{15}$$

Then, the errors between the covariance matrix and approximate covariance matrix are calculated through formula (16):

$$M^{errors} = \begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1n} \\ S_{21} & S_{22} & \cdots & S_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ S_{n1} & S_{n2} & \cdots & S_{nn} \end{pmatrix} - \begin{pmatrix} \tilde{S}_{11} & \tilde{S}_{12} & \cdots & \tilde{S}_{1n} \\ \tilde{S}_{21} & \tilde{S}_{22} & \cdots & \tilde{S}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{S}_{n1} & \tilde{S}_{n2} & \cdots & \tilde{S}_{nn} \end{pmatrix} \# \tag{16}$$

where

$$\tilde{S}_{ij} = Cov\left(D_{mcca}^{(i)}, D_{mcca}^{(j)}\right)$$

Finally, the canonical correlation variables can be explained by calculating the covariance matrix between $D^{(i)}$ and $D_{mcca}^{(i)}$.

$$Corr\left(D^{(i)}, D^{(i)}_{mcca}\right) = A_i^T Corr\left(D^{(i)}, D^{(i)}\right) = A_i^T S_{ii} \# \tag{17}$$

4.3. MCCA with Unpaired Samples

However, not every time does one find exactly m samples of each variable $\boldsymbol{X}^{(i)}$. This means that there are m_i random samples of $\boldsymbol{X}^{(i)}$, where $\exists i, j$ such that $m_i \neq m_j$. To apply multiset canonical correlation analysis for unpaired sample datasets, we simply use randomly sampled $M - m_i$ samples, which allow duplicates from $\boldsymbol{D}^{(i)}$ for each variable $\boldsymbol{X}^{(i)}$ where $M = \max_i \left\{m_i\right\}$ and add these samples to the data matrices so that there are exactly M samples in each data matrix. Multiset canonical correlation analysis is applied on all data matrices with M paired samples, and we obtain $\boldsymbol{A}_i, i = 1, 2, \ldots, n$. Finally, we calculate $\boldsymbol{D}^{(i)}_{mcca}, i = 1, 2, \ldots, n$ for the original data matrix $\boldsymbol{D}^{(i)}, i = 1, 2, \ldots, n$ without additional samples through formula (15).

5. EXPERIMENTS AND RESULTS

We conduct 4 main experiments for MCCA features and 4 additional experiments for the original features in the datasets of (Morris & Trivedi (2009)) or (Cruz, Macedo, & Guimaraes (2015)). In section 5.1, we conduct experiments for the errors and correlations between the original data and MCCA data through formula (16) and (17). In section 5.2, we train GBDT classifiers on the MCCA data. Like (Li et al. (2019)) who selected different metrics to discuss the differential evolution algorithm, we measure the classifier performance by calculating different indices: accuracy and F1 score. In section 5.3, we analyze the stability of the method in 4.3 applied to unpaired sample datasets. In section 5.4, we analyze the influence of curvature features on classifier performance. In section 5.5, we analyze the rationality of artificially constructed features.

There are 6 datasets included in (Morris & Trivedi (2009)) with the names: “cross”, “i5”, “i5sim”, “i5sim2”, “i5sim3” and “labomni”, where dataset “i5” and dataset “labomni” are unpaired sample datasets while the remaining datasets are paired sample datasets. There is only one unpaired sample dataset, which is called “UCI GPS Trajectories”, as included in (Cruz, Macedo, & Guimaraes (2015)) but with 4 evaluation indices: bus crowdedness level (BCL), traffic jam level (TJL), transportation choice (TL) and weather condition (WC). The redraws for the trajectories in each datasets are shown in *Figure 7* in Appendix A and the sample distributions are shown in *Table 2* and *Table 3* in Appendix A. Moreover, the detailed description for each dataset is shown in *Table 4* and *Table 5* in Appendix A.

Moreover, we only use 8 artificially constructed features for the datasets in (Morris & Trivedi (2009)) except for T, because the time is unclear in the datasets of (Morris & Trivedi (2009)). We extract 1 to 8 dimensional MCCA features for each dataset.

5.1. Error and Correlation

In this section, we conduct experiments on the paired sample datasets “cross”, “i5sim”, “i5sim2” and “i5sim3”. We first extract the MCCA features, and then calculate the error matrices between the original data and MCCA data with 8 dimensional MCCA features through formula (16). We count the mean, median and mode for each error matrix and draw histograms for each error matrix. The results are shown in *Figure 1* and *Table 1*.

From *Figure 1*, we see that most of the errors in each error matrix except for “i5sim2” are concentrated in the interval $\left[0,1\right]$. Although the error distribution for “i5sim2” is more scattered than others, the number continues to decrease as the error value becomes larger. In Table 1, the mean

Figure 1. Error distributions for the datasets: “cross”, “i5sim”, “i5sim2” and “i5sim3”

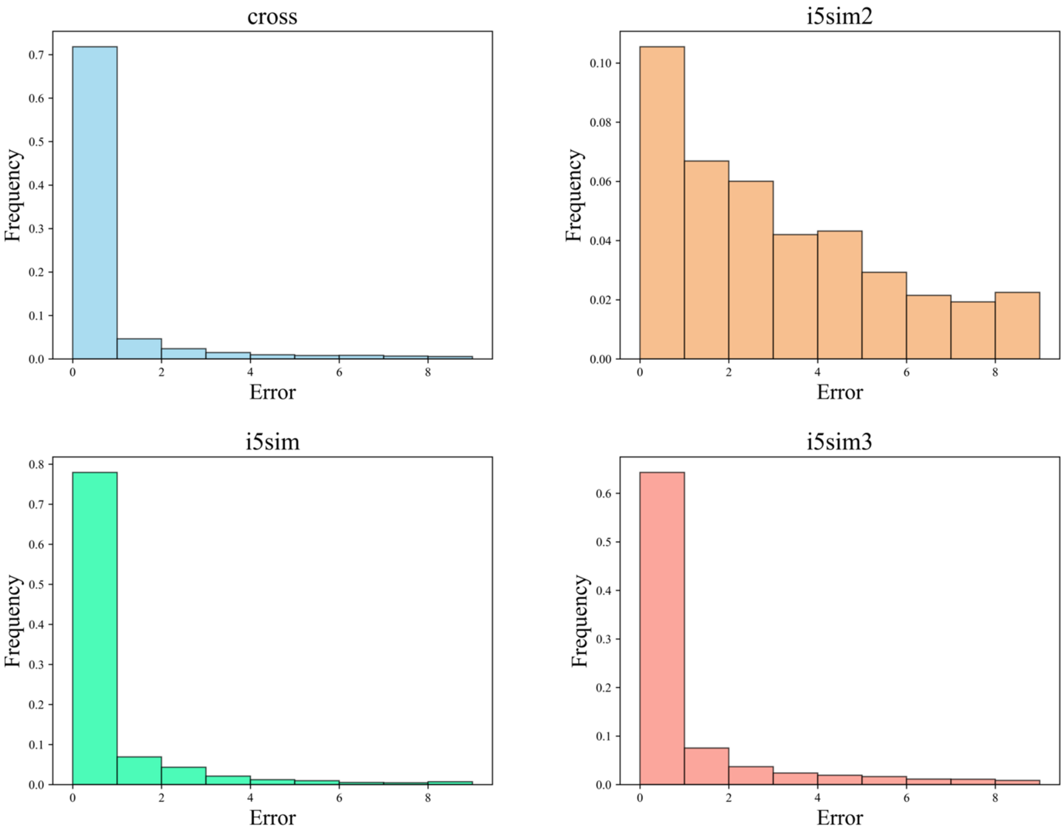


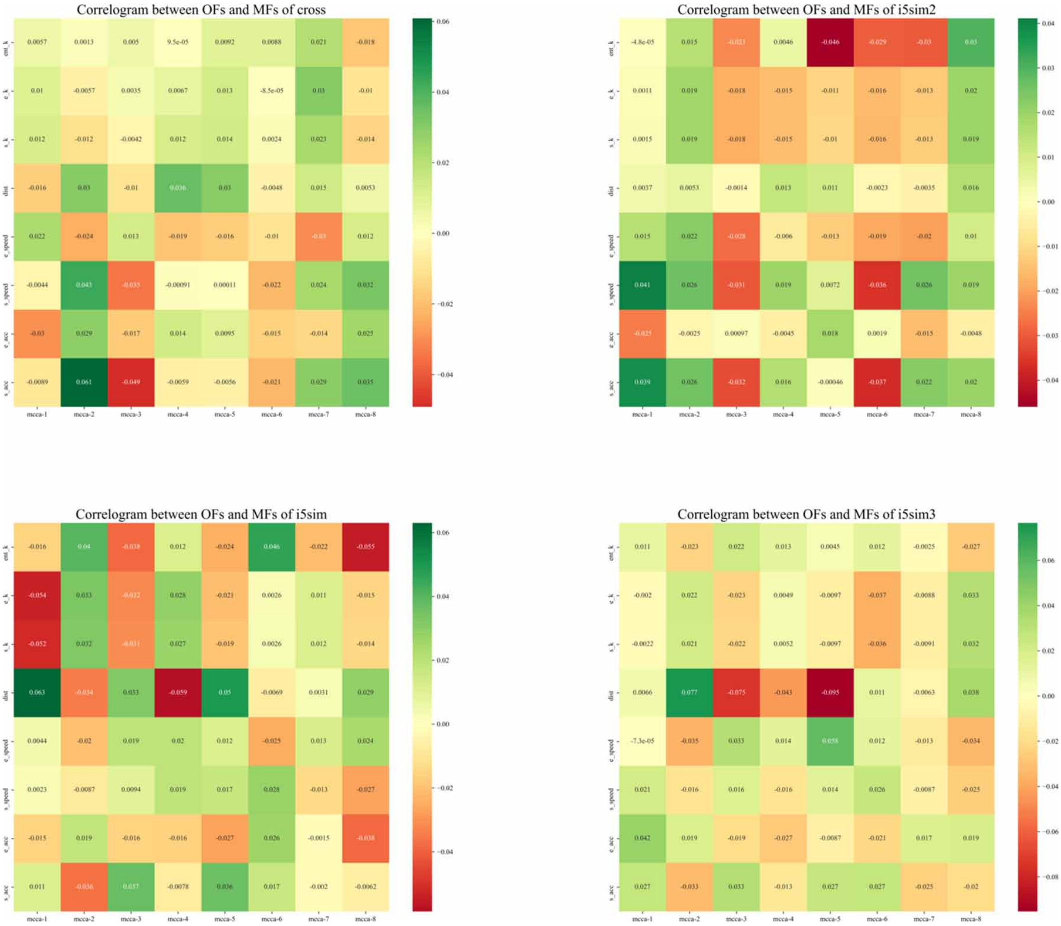
Table 1. Three indices for each error matrix

dataset	mean	median	mode
i5sim	10.955	0.130	0.001
i5sim2	3310.074	13.845	0.034
i5sim3	857.056	0.322	0.000
cross	202.023	0.122	0.000

of the errors for each error matrix is very large, while the median and mode are close to zero, because, although there are large errors at some points, most of the errors are small. Consequently, we still think that $\tilde{\mathbf{S}}_{ij}$ shows a good performance for reappearing $\mathbf{S}_{ij}$ in four datasets.

There may be different explanations of the same MCCA features facing different classes in one dataset; therefore, in order to make a statistical explanation for each MCCA feature of each dataset through heatmaps, we calculate the correlation matrix using the whole dataset with origin features and MCCA features and draw the heatmaps. The results are shown in *Figure 2*. Taking dataset “i5sim2” as an example, when the fifth MCCA variable shows the largest negative correlation with $\mathrm{Ent}\left(\kappa\right)$,

Figure 2. Heatmaps for the correlations between the origin Features and MCCA features for the datasets: "cross", "i5sim", "i5sim2" and "i5sim3"


Correlogram between OFs and MFs of cross

Correlogram between OFs and MFs of i5sim2

Correlogram between OFs and MFs of i5sim

Correlogram between OFs and MFs of i5sim3

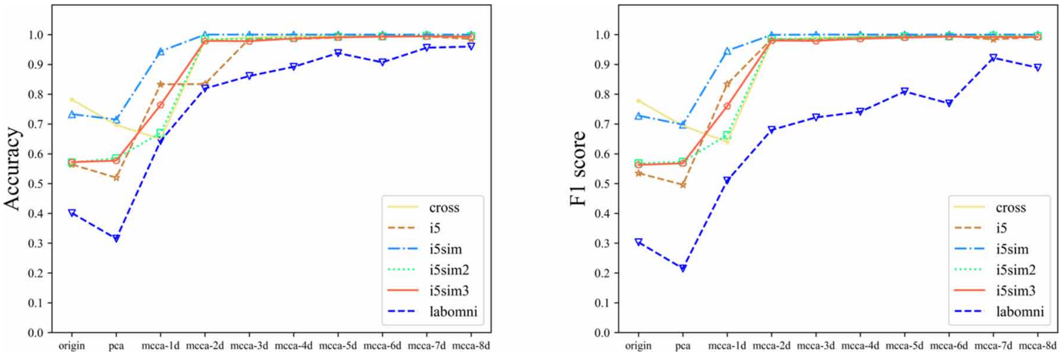
the first MCCA variable shows the largest positive correlation with $\mathrm{S}(v)$ and $\mathrm{S}(a)$. Therefore, we interpret the fifth MCCA variable as a negative synthesis of the original feature $\mathrm{Ent}(\kappa)$ and the first MCCA variable as a positive synthesis of the original features $\mathrm{S}(v)$ and $\mathrm{S}(a)$.

5.2. Classifier Performance

In this section, we conduct experiments on all datasets in (Morris & Trivedi (2009)). For the unpaired sample datasets "i5" and "labomni", we apply the method discussed in 4.3 to these datasets. We also first extract MCCA features, and then divide the dataset into a training set and test set in a $6:4$ ratio. We train a GBDT classifier on the training set and test on the test set. The results are shown in *Figure 3*. To enable comparison with the performance of the MCCA features, we also apply principal components analysis (PCA) to the original features to extract 3 dimensional PCA features.

From *Figure 3*, we see that as the dimension of the MCCA features increases, the accuracy also continues to increase. In contrast, with a 3-dimensional PCA feature, the accuracy and F1 score drop by at most 9% compared with the original feature. Moreover, compared with the 3-dimensional PCA features, the 3-dimensional MCCA features show an average improvement in the accuracy of 39%,

Figure 3. Classifier performance based on accuracy and F1 score for the datasets in (Morris & Trivedi (2009))



and an average improvement in the F1 score of 41%. When the MCCA features reach 8 dimensions, the accuracy is almost 100%. Although the F1 score may drop a little in the "labomni" dataset on the way as the dimension of the MCCA features increases, there is still a big improvement in the F1 score. Compared with the performance of the PCA features, 3-dimensional MCCA features have a huge advantage.

5.3. Stability

MCCA features extracted from the whole dataset show very good well performance in only one experiment. We can then ask the question can the MCCA features extracted from the training set show a similar performance. Moreover, we also question if the simple method in section 4.3 can be used to solve an unpaired sample problem that actually works. In this section, we conduct experiments using the datasets "labomni" and "UCI GPS Trajectories". Both of these datasets are unpaired sample datasets. Instead of extracting MCCA features at first, we first divide the dataset into a training set and test set in a $6:4$ ratio. Then, we apply the method in 4.3 only on the training set and solve the generalized eigenvalue problem (14) using the data in the training set. Here, we extract MCCA features from both the training data and test data. We then train a GBDT classifier on the training set and test on the test set. Finally, we repeat the experiment 100 times. The results are shown in *Figure 4*.

From *Figure 4*, we see that for the case of 100 repeated experiments, there is no serious turbulence observed in the accuracy and F1 score as the dimension of the MCCA features increases, so the method in section 4.3 shows a certain stability. We believe that MCCA is stable for extracting effective latent features and that the method in section 4.3 can work stably with high probability.

Figure 4. Classifier performance based on accuracy and F1 score in stability experiments

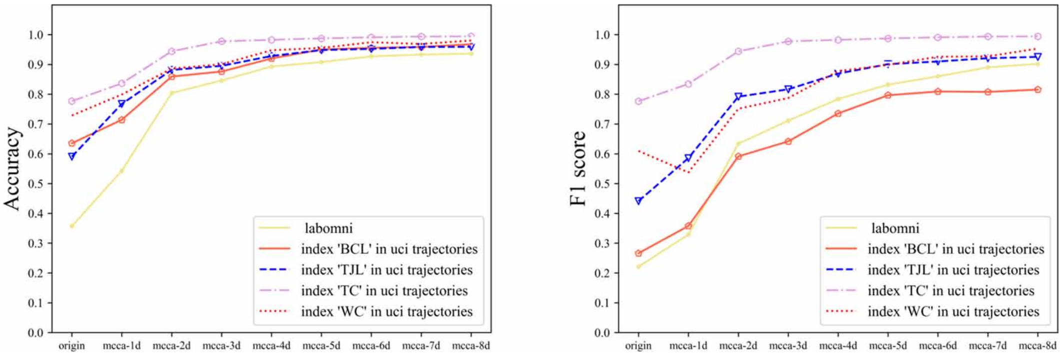
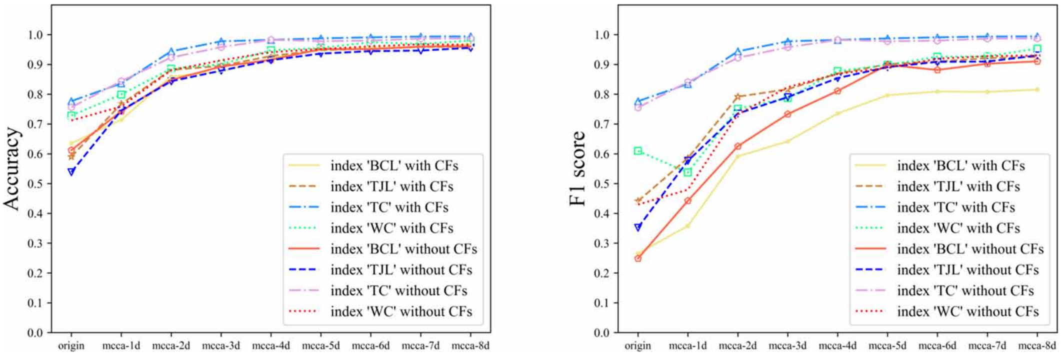


Figure 5. Classifier performance based on the accuracy and F1 score obtained for experiments with or without curvature features



5.4. Influence of Curvature Features

Calculation of the discrete curvature requires interpolation and differentiation, which may cost substantial computer resources and time. Since we cannot figure out the influence of curvature features just through Figure 2, in this section, we conduct experiments for determining the influence of curvature features (CFs) on the dataset “UCI GPS Trajectories”. Compared with the experimental results in 5.3, we apply the original features without CFs, which include $\mathrm{E}\left(\kappa\right)$, $\mathrm{S}\left(\kappa\right)$ and $\mathrm{Ent}\left(\kappa\right)$. Then, we repeat the experiments 100 times, so that we can determine the influence of the curvature features. The results are shown in *Figure 5*.

Unfortunately, from *Figure 5*, we see that although the classifier performance with CFs is a little better than that without CFs, CFs do not, in fact, make any difference, because the classifier performance is good enough whether there are CFs or not.

5.5. Rationality of Artificially Constructed Features

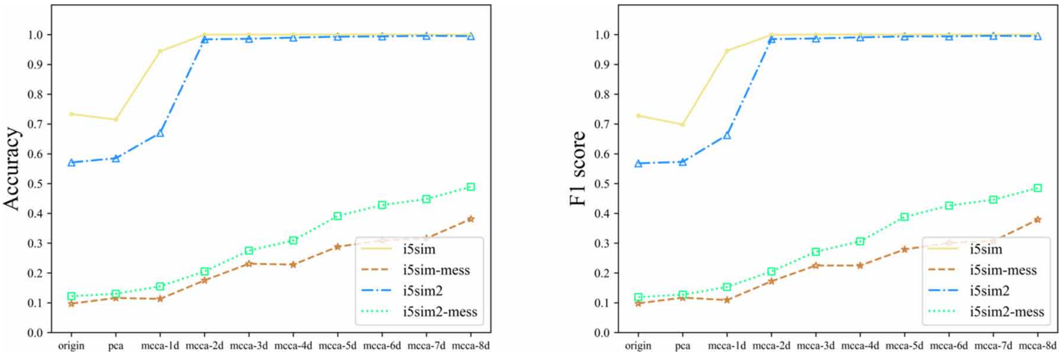
Finally, in this section, we conduct an additional experiment for the datasets “i5sim” and “i5sim2” to determine whether or not artificially constructed features lead to good classifier performance for MCCA features. Compared with the artificially constructed features, we randomly generate values using a standard normal distribution as features, which we call random features. The random features contain 8 dimensions. We also apply MCCA to the random features. The results are shown in *Figure 6*.

From Figure 6, we see that artificially constructed features lead to good classifier performance for the MCCA features, while random features only lead to bad classifier performance for MCCA features. Because we extract MCCA features from the whole dataset, an upward trend for classifier performance on the random features is observed when the dimensions of their MCCA features increase. Consequently, it is necessary to use artificially constructed features to represent trajectories before extracting MCCA features.

6. CONCLUSION

This paper mainly considers the role of multiset canonical correlation analysis in extracting latent features of trajectories. First, we find that multiset canonical correlation analysis can be used to extract more effective latent features, which greatly improves the classifier performance compared to principal components analysis. Second, we can see that the method of random sampling used in the original datasets to prepare the samples for each dataset paired is simple and practical for dealing with the unpaired sample problem. Third, according to the experimental results, we can see that

Figure 6. Classifier performance based on accuracy and F1 score for determining the influence of artificially constructed features and random features



curvature features may slightly improve the classifier performance. Finally, we find that compared with the features generated from a random normal distribution, it is very necessary to first construct artificial features to represent trajectory, so that we can extract effective latent features by multiset canonical correlation analysis.

There remains much work for us to do in the future. First, in this paper, we only use seemingly reasonable artificially constructed features to represent raw trajectories without in-depth research of the original data. Actually, we can do some research into the original data for the raw trajectories to find more suitable features to represent the trajectories, which may lead to better performance for low-dimensional MCCA features. Second, we can apply kernel multiset canonical correlation or multiset canonical correlation based on neural networks to extract features more efficiently. Finally, we need to find a more reasonable and robust method to solve the unpaired sample problem instead of simply randomly generating samples.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China [grant number 11671149].

REFERENCES

Charilaos, I. (2018). Structured sum cor multiview canonical correlation analysis for largescale data. *IEEE Transactions on Signal Processing*, *67*(2), 306–319.

Cruz, Macedo, & Guimaraes. (2015). Grouping similar trajectories for car pooling purposes. In *2015 Brazilian Conference on Intelligent Systems (BRACIS)*, (pp. 234–239). IEEE.

Faisal, I. (2007). Object trajectory based activity classification and recognition using hidden markov models. *IEEE Transactions on Image Processing*, *16*(7), 1912–1919. PMID:17605388

Ferrero, C. A., Alvares, L. O., Zalewski, W., & Bogorny, V. (2018). Movelets: Exploring relevant subtrajectories for robust trajectory classification. *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 849–856.

Hotelling, H. (1935). The most predictable criterion. *Journal of Educational Psychology*, *26*(2), 139.

Hotelling, H. (1992). Relations between two sets of variates. In *Breakthroughs in statistics* (pp. 162–190). Springer.

Jerome, H. (2001). Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189–1232.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Tie, Y. L. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 3146–3154.

Leite da Silva, C., Petry, L. M., & Bogorny, V. (2019). A survey and comparison of trajectory classification methods. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, (pp. 788–793). IEEE. doi:10.1109/BRACIS.2019.00141

Li, K., Chen, Y., Li, W., He, J., & Xue, Y. (2018). Improved gene expression programming to solve the inverse problem for ordinary differential equations. *Swarm and Evolutionary Computation*, *38*, 231–239.

Li, K., Liang, Z., Yang, S., Chen, Z., Wang, H., & Lin, Z. (2019). Performance analyses of differential evolution algorithm based on dynamic fitness landscape. *International Journal of Cognitive Informatics and Natural Intelligence*, *13*(1), 36–61.

Li, K., Wang, H., & Li, S. (2019). A mobile node localization algorithm based on an overlapping selfadjustment mechanism. *Information Sciences*, *481*, 635–649.

Lisanti, G., Karaman, S., & Masi, I. (2017). Multichannelkernel canonical correlation analysis for crossview person reidentification. *ACM Transactions on Multimedia Computing Communications and Applications*, *13*(2), 1–19.

Liu, H., & Lee, I. (2017). Endtoend trajectory transportation mode classification using BiLSTM recurrent neural network. In *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, (pp. 1–5). IEEE.

Mlıch, J., & Chmelar, P. (2008). Trajectory classification based on hidden markov models. *Proceedings of 18th International Conference on Computer Graphics and Vision*, 101–105.

Morris, B., & Trivedi, M. (2009). Learning trajectory patterns by clustering: Experimental stud ies and comparative evaluation. In *2009 IEEE Conference on Computer Vision and Pattern Recog nition*, (pp. 312–319). IEEE.

Nielsen, A. A. (2002). Multiset canonical correlations analysis and multispectral, truly multitem poral remote sensing data. *IEEE Transactions on Image Processing*, *11*(3), 293–305. PMID:18244632

Rao, H., Shi, X., Rodrigue, A. K., Feng, J., Xia, Y., Elhoseny, M., Yuan, X., & Gu, L. (2019). Feature selection based on artificial bee colony and gradient boosting decision tree. *Applied Soft Computing*, *74*, 634–642.

Sharma, Vyas, Schieder, & Akasapu. (2010). Nearest neighbour classification for trajectory data. In *International Conference on Advances in Information and Communication Technologies*, (pp. 180–185). Springer.

Son, J., Jung, I., Park, K., & Han, B. (2015). Trackingbysegmentation with online gradient boosting decision treel. *Proceedings of the IEEE International Conference on Computer Vision*, 3056–3064.

Ujiie, Y., Kato, T., Sato, K., & Matsuoka, Y. (2012). Curvature entropy for curved profile generation. *Entropy (Basel, Switzerland)*, *14*(3), 533–558.

Wang, F., Li, Y., Zhang, H., Hu, T., & Shen, X. L. (2019). An adaptive weight vec tor guided evolutionary algorithm for preferencebased multiobjective optimization. *Swarm and Evolutionary Computation*, *49*, 220–233.

Wang, F., Zhang, H., Li, K., Lin, Z., Yang, J., & Shen, X. L. (2018). A hybrid particle swarm optimization algorithm using adaptive learning strategy. *Information Sciences*, *436*, 162–177.

Wang, W., Chu, X., Jiang, Z., & Liu, L. (2019). Classification of ship trajectories by using naive bayesian algorithm. In *2019 5th International Conference on Transportation Information and Safety (ICTIS)*, (pp. 466–470). IEEE. doi:10.1109/ICTIS.2019.8883562

Xiao, Z., Wang, Y., Fu, K., & Wu, F. (2017). Identifying different transportation modes from trajectory data using tree-based ensemble classifiers. *ISPRS International Journal of Geo-Information*, *6*(2), 57.

Zheng, Y., Li, Q., Chen, Y., Xie, X., & Ma, W. Y. (2008). Understanding mobility based on gps data. *Proceedings of the 10th international conference on Ubiquitous computing*, 312–321.

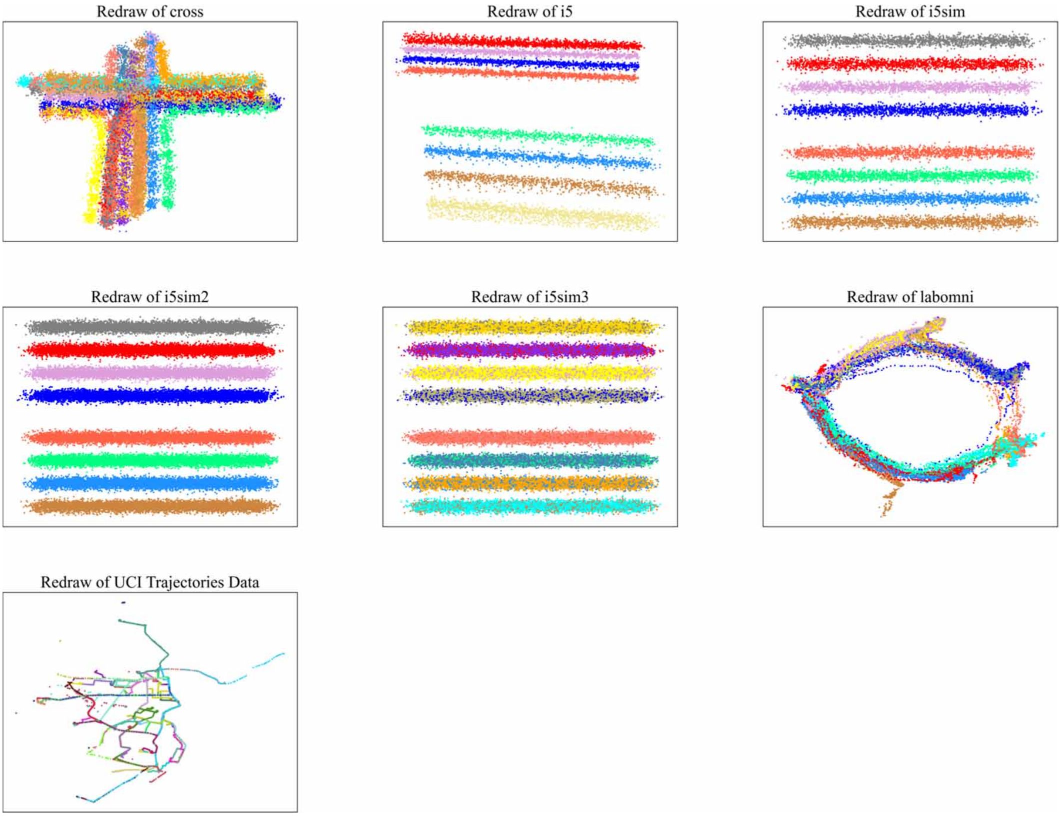
ENDNOTES

[1] https://www.microsoft.com/en-us/download/details.aspx?id=52367

[2] https://www.fs.fed.us/pnw/starkey/data/tables/

APPENDIX A. FIGURE 7 AND TABLES 2-5

Figure 7. Redraws for the trajectories



Redraw of cross

Redraw of i5

Redraw of i5sim

Redraw of i5sim2

Redraw of i5sim3

Redraw of labomni

Redraw of UCI Trajectories Data

Table 2. Sample distributions for the datasets in (Morris & Trivedi (2009))

dataset	sample distributions
cross	19 classes and 100 samples in each class
i5	01:137, 02:95, 03:107, 04:75, 05:114, 06:87, 07:93, 08:98
i5sim	8 classes and 100 samples in each class
i5sim2	8 classes and 200 samples in each class
i5sim3	8 classes and 100 samples in each class
labomni	01:8, 02:25, 03:8, 04:3, 05:30, 06:36, 07:28, 08:3 09:11, 10:4, 11:20, 12:3, 13:4, 14:22, 15:4

Table 3. Sample distribution for the dataset in (Cruz, Macedo, & Guimaraes (2015))

dataset	indices	sample distributions
UCI GPS Trajectories	bus crowdedness level	01:116, 02:34, 03:10, 04:3
	traffic jam level	01:17, 02:45, 03:101
	transportation choice	01:87, 02:76
	weather condition	01:116, 02:10, 03:37

Table 4. Detailed description of the datasets in (Morris & Trivedi (2009))

dataset	detailed description
cross	"Simulated four way traffic intersection with various through and turn patterns present. Units are pixels."
i5	"Highway trajectories in both direction of I5 outside of UCSD. Trajectories are obtained by a simple visual tracker. Units are pixels."
i5sim	"Simulated free flow highway scene. Noisy trajectory positions as might come from a GPS receiver. Units are meters."
i5sim2	"Simulated highway scene with bimodal speed distribution (slow and fast). Noisy trajectory positions as might come from a GPS receiver. Units are meters. The true cluster labeling only considers the lane of travel."
i5sim3	"Simulated highway scene with bimodal speed distribution (slow and fast). Noisy trajectory positions as might come from a GPS receiver. Units are meters. The true cluster labeling considers both the lane and speed. Therefore, the slow and fast trajectories in a single lane are considered as different clusters."
labomni	"Trajectories of humans walking through a lab captured using an omni-directional camera. Units are pixels."

Table 5. Detailed description of the datasets in (Cruz, Macedo, & Guimaraes (2015))

dataset	indices	detailed description
UCI GPS Trajectories	bus crowdedness level	An evaluation parameter that evaluates the amount of people inside the bus. (1-The amount of people inside the bus is little, 2-The bus is not crowded, 3-The bus is crowded.
	traffic jam level	An evaluation parameter that verify the volunteers perception about the traffic during the travel, in other words, if volunteers move to some place and face traffic jam, maybe they will evaluate 'bad'. (1-bad, 2-normal, 3-good).
	transportation choice	An evaluation parameter that evaluates people's choice of transportation. (1-car, 2-bus).
	weather condition	An evaluation parameter that evaluates the weather. (1-raining, 2-sunny).

Ping Huang, PhD, senior engineer, is currently the director of the Mathematics Experimental Center of South China University of Technology. He is mainly engaged in teaching and scientific research in the fields of data science, intelligent computing, machine learning and big data modeling. Presided over 3 scientific research projects above the provincial and ministerial level, more than 10 school-level research projects, and participated in 2 international natural science foundation projects. He has published more than 20 SCI, EI and Chinese core papers and edited and published 3 textbooks.

Jinliang Lu received his bachelor's degree in Mathematics and Applied Mathematics from the South China University of Technology in June 2019. In September 2019, he continues to study for a master's degree at South China University of Technology and is currently master student in computational mathematics. His research direction is object detection and tracking and abnormal behavior modeling. He once participated in "Huawei Cup" The 16th China Post-Graduate Mathematical Contest in Modeling and won the third prize.