Optik 127 (2016) 3905–3911

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Identification of flight state under different simulator modes using improved diffusion maps

B. Jia[a], C.F. Wei[b], J.F. Mao[c], R. Law[d], S. Fu[a], Q. Wu[a,e,*]

[a] School of Aeronautics and Astronautics, Shanghai Jiao Tong University, 800 Dongchuan Road, Minhang District, Shanghai 200240, China
[b] School of Aeronautic Science and Technology, Beihang University, Beijing 100083, China
[c] School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore
[d] School of Hotel and Tourism Management, Hong Kong Polytechnic University, Hong Kong
[e] School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, China

ARTICLE INFO

Article history:
Received 11 September 2015
Accepted 28 December 2015

Keywords:
Flight simulator
Simulator motion
Diffusion maps
Bacterial foraging
Kernel fuzzy c-means algorithm

ABSTRACT

To identify the difference between dynamic and static simulator modes, a novel data analyzing method was presented in this paper using flight data sampled from manual flight task. The proposed method combined diffusion maps and kernel fuzzy c-means algorithm (KFCM) to identify types of flight data. Hybrid bacterial foraging (BF) and particle swarm optimization (PSO) algorithm (BF-PSO) was also introduced to optimize unknown parameters of the KFCM. This algorithm increased the possibility to find the optimal values avoided being trapped in local minima. The clustering accuracy of the proposed method applied in flight dataset demonstrated this method had the ability to recognize the types of flight state. The results of the paper indicated that the pilots movement sensing influenced pilot performance under the manual departure task.

1. Introduction

The goal of flight simulator is to offer pilots actual sensing which will aid in pilot training and aircraft designing. The fidelity of simulator is an essential part to make sure the training or designing is reliable.

The first flight simulator in the world was invented in 1932 [1]. As the technique of industry developed, many kinds of flight simulators have been used [2]. In 1979, Caro described the interaction of simulator motion and pilot sensing system [3]. After that, the simulator has changed to consider more about the reality of motion feelings. In 2005, Brki-Cohen proved that simulator motion cues had little effect on initial training of pilots by experiment [4]. He introduced a simulator with fixed cockpit and dynamic seat [5]. Gu also designed a low-cost flight simulator [6] for pilot training. On contrary, there were some researches emphasized that the simulator motion is necessary for a reliable simulation. Groen explored the perception model in moving simulator in 2007 [7]. Hess proposed a pilot model with motion sensing system to evaluate simulator fidelity in 2009 [8]. And the motion-visual phase-error of a dynamic simulator was calculated by Grant [9].

By the above literature review, researches were rarely done to find relationship between simulator motion and pilot perception using flight data. Clustering methods had not been reported to deal with simulator fidelity. And then, this paper focused on the novel classification method based on hybrid diffusion maps and KFCM using simulator flight data.

Diffusion maps (DMs) is proposed by Coifman and Lafon [10]. The basic part of diffusion maps is its metric which represents the probability that one spot transfers from one place to another. This probability follows a Markov chain which is updated as the diffusion time extends. Unlike the physical distance, the diffusion metric reflects not only physical distance but also the density of the dataset. Therefore small random perturbation from the dataset barely influences diffusion metric [11]. Diffusion maps help us explore properties of the integrated dataset. It maximizes the eigenvector approximation, meanwhile minimizes the distortion in the embedding space. This technique has been used successfully in several applications. Xu used this technique to analyze high dimensional gene expression data in 2010 [12]. Ferguson extracted fewer variables to study molecular simulation trajectories using it in 2011 [13]. At the same time, Farbman made use of diffusion distance to replace Euclidean distance in edge-aware operations [14]. The correlations among the variables of data are generally nonlinear, so the diffusion maps also provide a nonlinear reduction to discuss high-dimensional data, while the reduction from Principal Component Analysis (PCA) is often unsuitable [10].

* Corresponding author. Tel.: +86 2134003429.
E-mail addresses: icesea137@163.com (B. Jia), wuqi7812@sjtu.edu.cn (Q. Wu).

http://dx.doi.org/10.1016/j.ijleo.2015.12.162
0030-4026/

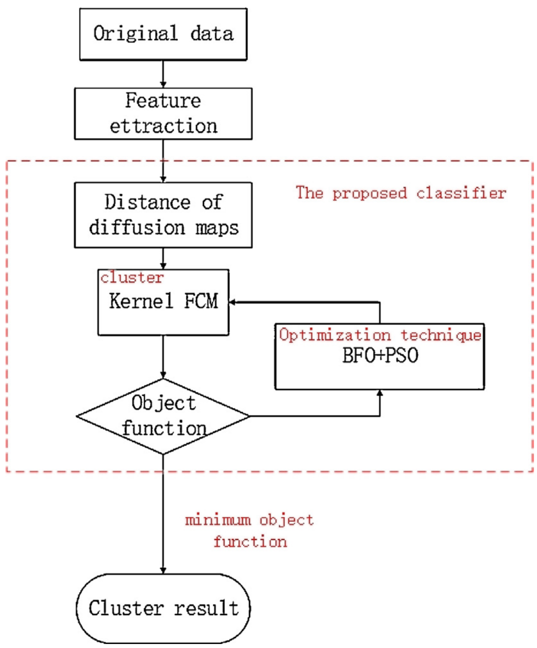

Fig. 1. The process of the hybrid cluster algorithm.

KFCM is a widely used clustering method. It is robust to noise, outliers and incomplete data [11]. And it suits to analyze unequal-size classes. Different from FCM, the KFCM is based on non-Euclidean distance by using kernel function. The kernel approach does not decide whether a data point belongs to a subset at the very first, but transforms the centroids and data points from the original space into a high-dimensional space with a kernel function. In this transformed space the nonlinear pattern from the original space is represented by linear one.

This paper aimed to find a reliable method to identify the difference between two simulator modes, dynamic and static. To identify pilot performances under two modes, signal features from manual-task flight data were extracted first. On the basis of suitable feature extraction, a novel classification method was also proposed to recognize the two modes of simulator. To identify the unknown parameters of KFCM, a novel hybrid BF and PSO algorithm was also used. This hybrid algorithm avoided from being trapped in local minima, and cost less time than some other optimizing algorithms.

This paper is organized as follows: Section 2 presented a methodology based on diffusion maps and kernel FCM. Section 3 proposed a hybrid BF and PSO algorithm to optimize the unknown parameters of KFCM. Section 4 analyzed flight signals for feature extraction. The experiments and discussion were presented in Section 5. Finally, the conclusions were arranged in Section 6.

2. Classifier based on improved diffusion maps

The procedure of flight state clustering method which is based on diffusion maps and KFCM is shown in Fig. 1. It can be summarized by the following steps:

Step 1 Feature extraction.
Step 2 Calculation of diffusion distances. Using diffusion maps, the complicated and high-dimensional features turned into low-dimensional and easy to cluster.
Step 3 Kernel FCM is used to cluster the set.
Step 4 If the object-function is not minimal, go to Step 5; else, go to the end.
Step 5 Optimization of KFCM parameters using hybrid BF-PSO algorithm. Then, go to Step 3.
Step 6 The end.

2.1. Diffusion maps

Diffusion distance is a helpful metric which could describe the connectivity of points and reflect the density information of the dataset. It supplies coordinates on the dataset that could be used to identify the points. There are three important parts, such as eigenmaps, dimensional reduction and Markov chain which is determined by the dataset distribution, constitute the diffusion framework. This framework was proposed in [15] as:

Let $G = (\Omega, W)$ be a finite set with n points. $W = \{w(x, y)\}_{x,y\in\Omega}$ is a symmetric and points positive matrix. The connection between any two points, x and y, could be measured as weight $w(x, y)$. This weight is usually calculated using Gaussian kernel $w = \exp(-\|x-y\|^2)/\sigma$.

In next step, a Markov chain is made according to $p_t(x, y) = \frac{w(x,y)}{d(x)}$, where$d(x) = \sum_{z\in\Omega} w(x, z)$ is the degree of node x. $p_t(x, y)$ means the possibility of travelling from x to y in time t. The parameter t indicates the power of influence from neighbors. If every two nodes in the dataset are connected [16]:

$$\lim_{t\to+\infty} p_t(x, y) = \phi_0(y)$$

where $\phi_0(y) = \frac{d(y)}{\sum_{z\in\Omega} d(z)}$. The quantity of $\phi_0(y)$ connects to the degree of x, which reflects the density in the graph.

The diffusion distance was defined in [17]:

$$D_t^2(x, z) = \left\| p_t(x, .) - p_t(z, .) \right\|^2_{1/\phi_0} = \sum_{y\in\Omega} \frac{(p_t(x, y) - p_t(z, y))^2}{\phi_0(y)} \quad (1)$$

This novel distance reflects intrinsic properties of the set. Because diffusion distance contains influences from neighbors, it is (unlike Euclidean distance, etc.) robust to noise.

The diffusion maps $\Psi_t : \Omega \mapsto E^{q(t)}$ makes the original space Ω map to a lower-dimensional space. In this space, traditional diffusion maps chose the Euclidean distance to represent diffusion distance in Ψ_t. This distance is the property we want to preserve during high-dimensional dataset reduction.

2.2. Kernel FCM

Clustering aims gathering data into subsets. This technique makes sure that the data in one subset have maximum similarities while the data in different subsets have rarely similarity [18–20].

Traditional FCM [21] divides a dataset into c fuzzy subsets. Following the division, the objective function $Q = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m \left\| x_j - o_i \right\|^2$ could be minimized. Here, n is the data number. $\|*\|$ means to calculate the distance. u_{ij} is the probability of x_j belonging to group i, and obeys the constraint $\sum_{i=1}^{c} u_{ij} = 1$. o_i is the centroids of subset i. m which controls the clustering fuzziness usually is set as 2.

As an improvement of classical FCM, the KFCM transforms the initial data space into a much higher dimensional space (a Hilbert space) using kernel functions. In the new space, clusters could be more obvious to be classified. Because in this space, relations between data points are more obvious [22–24]. This clustering method makes the probability u_{ij} be more robust.

3. The parameters optimization of KFCM

In KFCM, cluster centroids are selected randomly. How to place cluster centroids efficiently is a challenge. To conquer this,