

# 基于瓶颈神经网络的轨迹嵌入技术及其在飞行目标轨迹分类中的应用

雷磊

中国电子科技集团公司第十研究所, 四川 成都

收稿日期: 2022年9月24日; 录用日期: 2022年10月20日; 发布日期: 2022年10月27日

## 摘要

轨迹分类是利用目标运动轨迹识别出目标类型的技术。如何从轨迹数据中提取出可分性好的轨迹特征一直是轨迹分类的研究重点。本论文提出了基于神经网络的轨迹嵌入方法, 从轨迹数据中提取可分性较好的轨迹特征。该方法首先从轨迹数据中提取局部和全局特征, 组成高维度的特征向量; 然后将这些高维度向量带入瓶颈神经网络(bottleneck NN, b-NN), 得到低维度的超向量, 称为t-vector。因为b-NN将高维度的特征向量投影到低维度的“通用坐标”空间中, 对特征向量进行了校准和压缩, 所以t-vector在超向量空间中具有较低的维度和较好的可分性。实验表明, t-vector能够提升分类模型5%以上的准确率, 并使其检测代价值(detection cost function, DCF)较低, 有效提高了飞行目标轨迹分类的性能。

## 关键词

轨迹嵌入, 轨迹分类, 轨迹特征提取, 瓶颈神经网络

## Trajectory Embedding Based on Bottleneck Neural Network and Its Application on Trajectory Classification of Aircrafts

Lei Lei

The 10th Research Institute of China Electronics Technology Corporation, Chengdu Sichuan

Received: Sep. 24<sup>th</sup>, 2022; accepted: Oct. 20<sup>th</sup>, 2022; published: Oct. 27<sup>th</sup>, 2022

## Abstract

Trajectory classification is a technique which classifies the objects based on their trajectory. The key of trajectory classification is to find the discriminative features that better discriminate the class. This

paper proposes a trajectory embedding method based on neural network to extract the discriminative features from trajectory data. This proposed method extracted the local and global features having high dimension from raw trajectory data at first, and then obtained the low-dimension supervector named t-vector using the bottleneck neural network (b-NN). Due to the b-NN maps the high-dimension vectors into a low-dimension "common coordinate" space, the t-vector had low dimension and good discrimination when doing similarity computations in the supervector space. The experimental results shown the classification model with t-vector obtained more than 5% accuracy, and obtained lower detection cost function (DCF), which improved the performance of the trajectory classification.

## Keywords

Trajectory Embedding, Trajectory Classification, Trajectory Feature Extraction, Bottleneck Neural Network

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

轨迹分类是一种利用目标轨迹识别目标类型的技术，应用于行为习惯挖掘、交通管理、船只航行监控等领域，为辅助完成疏导交通拥堵，优化交通调度、海上预警等任务[1] [2]。在空中管理中，轨迹分类技术用于区分不同飞行目标的类型，预警客机附近出现的鸟类、无人机等飞行目标，使得相关人员能够及时采取措施，避免相撞事故，保障民航班机安全飞行[3] [4]。

目标轨迹是轨迹分类任务的处理对象，通常由 GPS、北斗、雷达等定位系统产生，由一组和时间相关的目标空间位置数据组成。由于轨迹数据结构简单，因此如何从轨迹数据中提取出可分性好的轨迹特征，以支撑轨迹分类任务是轨迹分类研究的重点和难点[5]。

最早的轨迹特征提取算法由 Lee 提出，称为 TraClass [6]。该方法先将轨迹区域划分成网格，然后不断细化网格，直到同类目标的轨迹几乎处于同一网格中，那么这个网格就被当成该类轨迹的特征。但是，该方法仅考虑了轨迹的空间信息而没有考虑到轨迹的时间信息。

为了同时提取飞行器的空间信息和时间信息，研究者提出了两种基于统计学的轨迹特征：局部特征[7]和全局特征[8]。局部特征提取方法先将轨迹划分为子段。然后根据子段数据得到目标速度、加速度、转向角等运动参数的统计量，作为轨迹特征。和局部特征相比，全局特征也由运动参数的统计量组成。区别在于全局特征由完整轨迹统计得到，而局部特征由局部轨迹统计得到。当轨迹数据质量较好时，局部特征和全局特征能够准确地刻画目标的运动特性，满足轨迹分类的需求。然而，受到大气环境和设备状况等因素的影响，轨迹数据通常包含大量的飞点(错误的目标空间数据)，对局部特征和全局特征造成干扰。

随着机器学习技术的发展，神经网络结构不仅用于完成识别、分类任务，也用于生成数据特征。这种利用神经网络结构从数据中提取特征的技术称为嵌入技术[9]。目前，嵌入技术已经应用于各个领域。在文本处理领域，著名的 word2vector 模型[10]就是利用嵌入技术将文本转化词向量。和传统的 one-hot 编码相比，词向量具有语义相关和低维度的优点。在语音处理领域，google 工程师利用瓶颈神经网络 (bottleneck neural network, b-NN)将高维度的语音倒谱特征映射成低维度的 d-vector [11]、s-vector [12]，从

而大幅度提高了语音识别和声纹识别的性能。在图像处理领域, Tang 等人也利用 CNN 结构对图像数据进行嵌入, 提高了视频识别的性能[13]。另外, 嵌入技术还应用于水文数据挖掘[14]、雷达信号识别[15]、医学数据处理[16]等领域, 并获得了成功。这些研究成果说明嵌入技术能够从数据中获取性能较好的特征。

为了减弱飞点对轨迹局部和全局特征的影响, 本论文提出了一种基于神经网络的轨迹嵌入方法。该方法首先从轨迹中提取出局部和全局特征, 组成高维度的特征向量; 然后将这些高维特征向量带入 b-NN 得到低维度的超向量, 称为 t-vector。由于 b-NN 将高维特征向量投影到低维度“通用坐标”空间, 对特征向量值进行校对和压缩[17][18], 所以 t-vector 具有维度较低且可分性较好的优点。本论文还将 Xgboost、SVM 和随机森林组成三人表决器, 对 t-vector 分类, 提高了目标轨迹分类的鲁棒性。在实际试验中, t-vector 提高了三人表决器 5% 以上的准确率, 并使其获得较低的 DCF 值。

## 2. 相关方法

### 2.1. 轨迹特征提取

为了从飞行器轨迹中提取有效的轨迹特征, 学者提出了两种基于统计的特征提取方式: 局部特征提取和全局特征提取。

局部特征提取分为分段和特征计算两个部分。最常用轨迹分段方法由 Dodge 提出[19]。该方法根据运动物理学原理, 将轨迹数据分为“低偏离低蜿蜒”、“低偏离高蜿蜒”、“高偏离低蜿蜒”、“高偏离高蜿蜒”四种类型, 然后根据这些类型将轨迹分成若干轨迹段。得到轨迹段后, 求出这些轨迹段的运动参数统计量, 就得到局部特征。常用的统计量包括均值、期望、标准差、最值、百分占比、偏斜率、峰度、变化系数、自相关系数等[20]。全局特征提取方法从完整的轨迹中计算运动参数的统计量。另外, Yang 等人还提出了朝向变化率、停止率、速度变化率、航迹长度等全局特征[21]。

### 2.2. 瓶颈神经网络

瓶颈神经网络(bottleneck neural network, b-NN), 如其名所示, 是一种带有瓶颈层的神经网络, 用于完成特征提取任务, 最早由 google 工程师们提出, 用于从语音倒谱中生成 d-vector 和 s-vector, 提高了语音识别、声纹识别的性能。后来, b-NN 又应用于水文数据、雷达信号和医学数据的特征提取任务中[22]。

b-NN 由输入层, 输出层, 若干隐层和瓶颈层构成。隐层结构由具体任务决定, 可以为全连接层和可以为卷积层。输出层为 softmax 后验概率, 代表输入数据的类型。瓶颈层的节点比隐层节点少得多, 用于维度压缩和生成特征。它的位置不固定, 若靠近输入层, 那么特征表征数据个体特性; 若靠近输出层, 那么特征表征数据类型特性[23]。

## 3. 轨迹嵌入方法

在实际运用中, 受到定位环境和定位设备的影响, 得到的轨迹中含有大量的飞点(定位错误点), 从而干扰特征值。因此, 本论文提出基于嵌入技术的特征提取法, 以得到鲁棒性较好的轨迹特征。该方法流程如图 1 所示。

### 3.1. 预处理

预处理的作用是取出轨迹中的冗余点和异常点。假设一条航迹定义为  $T = \{p_1, p_2, \dots, p_N\}$ , 其中  $p_i = (g_i, t_i)$  为轨迹中的点,  $g_i$  为轨迹点的空间信息如经度、纬度和高度等;  $t_i$  为轨迹点的时间。在预处理中, 首先检查  $g_i$ 。比如经度和纬度必须要满足经度、纬度和高度要在合理范围之内。然后计算相邻点的速度  $v_i$  和距离  $s_i$ :

$$s_i = |p_{i+1} - p_i| \quad (1)$$

$$v_i = \frac{s_i}{t_{i+1} - t_i} \quad (2)$$

如果  $s_i$  小于阈值则认为  $p_i$  冗余点, 需要剔除掉。如果  $v_i$  不再合理范围内(如 10 m/s 或者 1000 m/s)则认为  $p_i$  为异常点, 需要剔除。

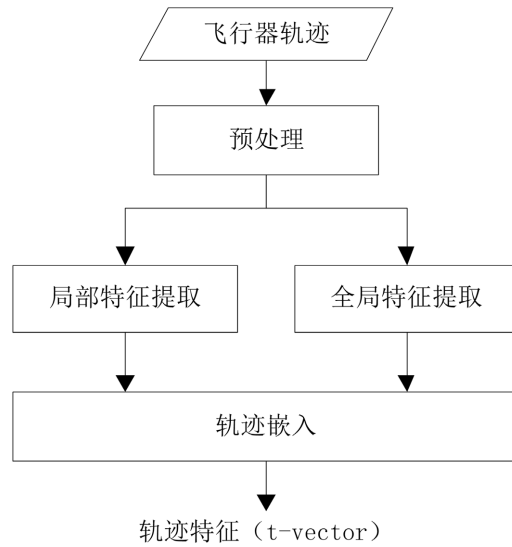


Figure 1. The flow of trajectory embedding  
图 1. 轨迹嵌入流程

### 3.2. 局部特征和全局特征提取

轨迹经过预处理后, 就可以用于提取局部和全局特征。提取方式如图 2 所示:

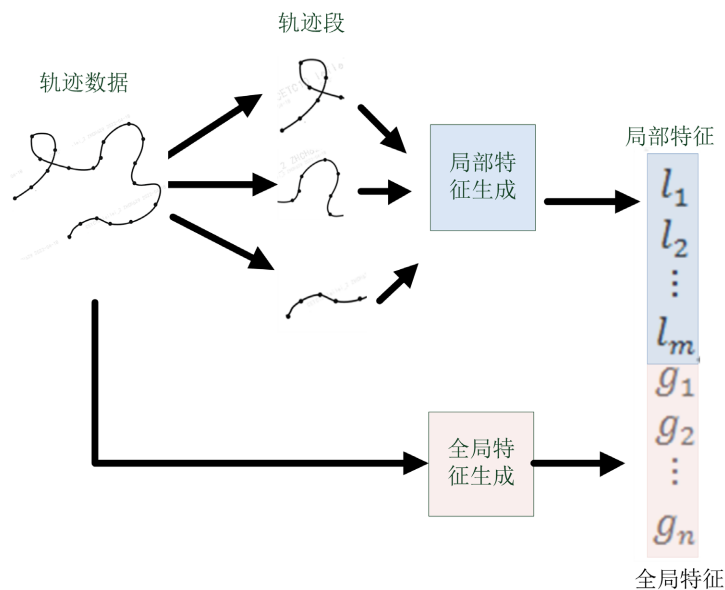


Figure 2. Local and global features extraction  
图 2. 局部和全局特征提取

首先计算速度、加速度、爬升量、转弯角四种运动参数用于划分轨迹，于是得到 16 种轨迹类型。根据每种类型将轨迹划分成若干轨迹段，然后基于这些轨迹段计算每种轨迹类型的段长度均值、标准差，类型数量和类型百分比作为局部特征(图中表示为  $l_i, i=1, 2, \dots, m$ )，得到  $m=64$  个局部特征。

全局特征(图中表示为  $g_i, i=1, 2, \dots, n$ )从整段轨迹中提取得到。本论文计算速度、加速度、爬升值、转弯角、蜿蜒度 5 种运动系数用于生成全局特征。每种运动系统对应的全局特征为：

均值；

标准差；

最大值、最小值；

模：即出现平率最多的取值范围；

四分位：下四分位和上四分位；

四分位距离。

偏斜度  $S_k$ ：

$$S_k = E \left[ \left( \frac{X - \mu}{\sigma} \right)^3 \right] \quad (3)$$

其中， $X$  为运动参数， $\mu$  为运动参数的均值， $\sigma$  为运动参数的方差。

峰度  $K_u$ ：

$$K_u = E \left[ \left( \frac{X - \mu}{\sigma} \right)^4 \right] - 3 \quad (4)$$

其中， $X$  为运动参数， $\mu$  为运动参数的均值， $\sigma$  为运动参数的方差。

变化系数  $CV$ ：

$$CV = \frac{\mu}{\sigma} \quad (5)$$

自相关系数  $A$ ：

$$A = \frac{C_1}{C_0} \quad (6)$$

其中，

$$C_k = \frac{1}{N} \sum_{i=1}^{N-k} (x_i - \mu)(x_{i+k} - \mu) \quad (7)$$

其中， $x_i$  为第  $i$  运动参数， $\mu$  为运动参数的均值， $\sigma$  为运动参数的方差。另外，还采用了 Wang 等人[24]提出的四种全局特征：朝向变化率、停止率、速度变化率和航迹长度。因此，得到 54 个全局特征。将 64 个局部特征和 54 个全局特征组合到一起，就构成了 118 维的特征向量，用于 b-NN 的输入以完成轨迹嵌入。

### 3.2. 轨迹嵌入

本论文采用 b-NN 将 118 维的向量映射成 32 维的 t-vector 完成轨迹嵌入。该 b-NN 的结构如图 3 所示。

该 b-NN 有 7 层全联接层隐层，每层有 128 个节点。为了得到分类性能更好的嵌入向量，瓶颈层位于输出层和最后一层隐层之间，包含 16 个节点。b-NN 的最后一层为分类层，包含四个节点，分别表示客机、无人机、鸟类和其他目标。训练时，采用所有支撑数据进行训练，学习率为 0.01。生成 t-vector

时, 去掉最后一层输出层, 将每条轨迹对应的局部和全局特征作为输入, 然后输出一个 t-vector。即每个 t-vector 对应一条轨迹样本。

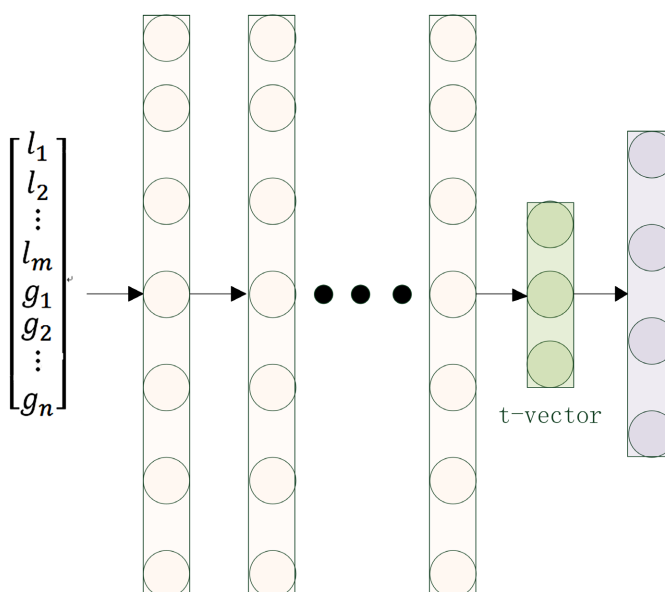


Figure 3. Structure of b-NN  
图 3. b-NN 网络结构

#### 4. 轨迹分类流程

论文将 t-vector 用于轨迹分类任务, 提出一种轨迹分类模型, 其流程如图 4 所示。

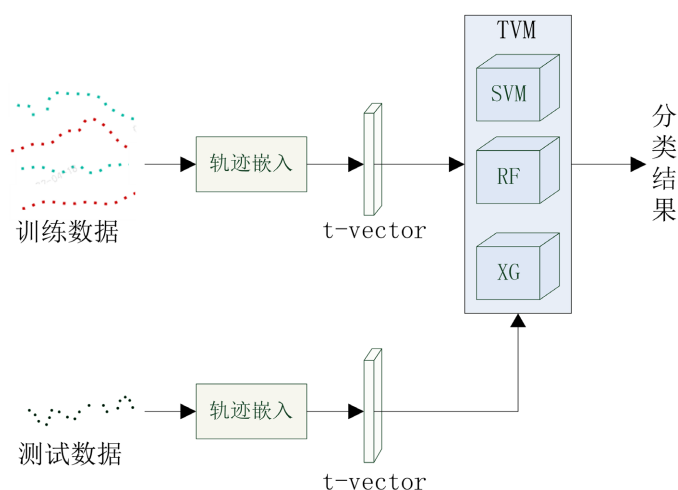


Figure 4. The flow of trajectory classification  
图 4. 轨迹分类流程

轨迹分类流程分为两个部分: 训练部分和评估部分。在训练部分, 训练数据(分类已知)代入轨迹嵌入过程得到 t-vector, 然后用 t-vector 训练由支撑向量机(support vector mechine, SVM), 随机森林(Random Forest, RF)和 XGBoost(XG)组成的三人表决器模型(Three-voter model, TVM)。在评估阶段, 测试数据(分类未知)输入轨迹嵌入过程得到 t-vector, 然后将 t-vector 代入训练好的 TVM 得到最后的分类结果。

## 5. 实验与结论

### 5.1. 实验数据集和实验过程

实验采用的数据集为公开领域收集的客机、鸟类、无人机和其他飞行器轨迹数据，各 2 万条航迹。该数据中，每行代表一个航迹点。每行包含 6 列，分别为 ID、Lat、Lon、Alt、T 和 Class。ID 为航迹标识，即同一 ID 的航迹点属于同一个飞行目标的同一航迹；Lat 和 lon 两项分别为经度和纬度，单位为度；Alt 为高度，单位为米；T 为系统时间即获取经纬度高度信息的时间，单位为秒。Label 为目标类型标识，为了方便训练，将客机、鸟类、无人机和其他飞行器打标为“1000”、“0100”、“0010”和“0001”。

首先将所有数据作为支撑集训练 b-NN，然后将数据按照 8:2 的比例划分成训练集和测试集，用于训练和测试轨迹分类的性能。为了评估测试轨迹的分类性能，采用准确率(ACC)和检测代价函数值(DCF)最为评判标准。准确率为分类正确的样本量和总样本量的比值。DCF 定义为：

$$DCF = \alpha F_1 + (1 - \alpha) F_2 \quad (8)$$

其中， $\alpha$  为权重，设为 0.5。 $F_1$  为正确接收率，即正确分类正例样本量和正例样本量的比。 $F_2$  为正确拒绝率，即正确分类反例样本量和反例样本量的比。为了计算 DCF，本论文分别取 1 万条客机、飞鸟、无人机和其他飞行器航迹为正例，取剩下分类的 1 万条样本航迹作为反例。例如，当去 1 万条客机航迹为正例时，取飞鸟、无人机和其他飞行器共 1 万条航迹为反例。因此，我们将得到 4 个 DCF：DCF<sub>1</sub>、DCF<sub>2</sub>、DCF<sub>3</sub> 和 DCF<sub>4</sub>，分别表示客机、鸟、无人机和其他飞行器为正例时得到的 DCF。

### 5.2. 轨迹嵌入性能测试

实验测试用于轨迹嵌入的 b-NN 性能。实验中包含不同隐层的 b-NN 用于生成 t-vector，采用上述四种飞行目标的轨迹进行实验。实验中，b-NN 根据飞行目标的轨迹数据生成 t-vector，由 SVM、RF、XG 组成的三人表决模型(Tree-voters model, TVM)分析 t-vector 给出飞行目标的类型。实验结果如图 5 所示。

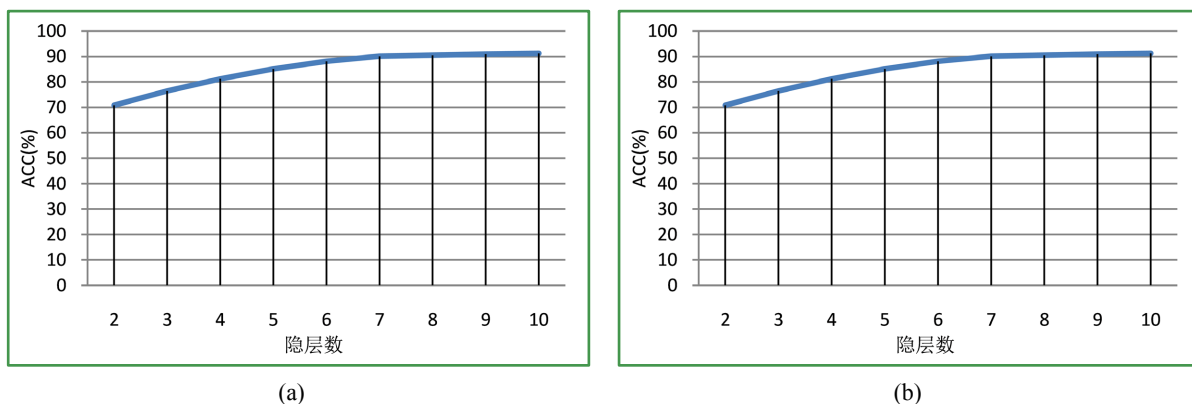
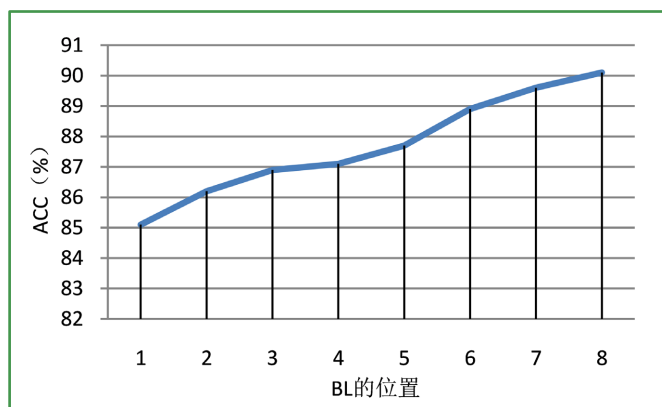


Figure 5. The performance of trajectory embedding with different hidden layers. (a)Accuracy; (b) Time cost  
图 5. 不同隐层对轨迹嵌入的影响。(a) 准确率；(b) 时耗

图 5(a)中显示，随着隐层数量的增加，TVM 的准确率逐步提升。当隐层数量达到 7 层之前，准确率提升速度较快，而达到 7 层之后，准确率提升趋于平缓。这说明较多隐层有利于提升 t-vector 的可分性，从而提高分类准确率，但是隐层增加也会增加生成 t-vector 的时间，如图 5(b)所示，因此拥有 7 层隐层的 b-NN 可以兼顾准确率和时耗。

下面将 b-NN 的隐层固定为 7 层，然后移动瓶颈层(bottleneck layer, BL)的位置，实验结果如图 6 所示。





**Figure 6.** The performance of trajectory embedding with different locations of the bottle-neck layer  
**图 6.** 瓶颈层位置对轨迹嵌入的影响

图 6 中可以看出, 当 BL 位于前 5 层时, TVM 的准确率较低, 即 TVM 的分类性能较差; 当 BL 位于后 3 层时, TVM 获得最高的准确率, 即 TVM 的分类性能较好。这说明, BL 靠近 b-NN 尾部能得到可分性更好的 t-vector。这是因为当 BL 靠前(即隐层在第 1~5 层)时, 生成的 t-vector 带有更多的轨迹个体信息, 而 BL 靠后(即隐层在第 6~8 层)时, 生成的 t-vector 带有更多的轨迹类型信息。由于分类任务的目标是分类轨迹类型, 所以靠后的 BL 能够提升表决器的性能。

### 5.3. 轨迹特征性能测试

实验测试 t-vector 的性能。对比对象为 TraClass、局部特征(local feature, LF)、全局特征(global feature, GF)。为了评估特征性能, 采用 SVM、RF 和 XG 组成的 TVM 对上述 4 种轨迹特征分类。实验结果如表 1 所示。

**Table 1.** The performance of different trajectory features  
**表 1.** 轨迹特征的性能

特征	ACC(%)	DCF <sub>1</sub>	DCF <sub>2</sub>	DCF <sub>3</sub>	DCF <sub>4</sub>
TraClass	80.3	0.073	0.131	0.139	0.133
LF	84.7	0.081	0.086	0.089	0.091
GF	85.0	0.075	0.078	0.083	0.081
t-vector	90.1	0.051	0.054	0.058	0.062

表中显示, 采用 TraClass 作为特征时, TVM 的准确率最低而 DCF<sub>2</sub>、DCF<sub>3</sub> 和 DCF<sub>4</sub> 最高。这表明 TraClass 不适合用于分类飞行目标轨迹分类。另外, TraClass 在文献[1]中获得很好的性能, 而在本实验中的性能不佳。这是因为文献中使用 TraClass 表示车辆轨迹。车辆运动轨迹受道路限制, 使得 TraClass 提取过程能较容易“框住”同类目标的轨迹, 提取出较好的特征。最后一行中, 采用 t-vector 作为特征时, TVM 获得最高的准确率和最低的 DCF, 这说明 t-vector 的可分性最好。另外, LF 和 GF 对应的准确率低于 t-vector 对应的准确率, 同时它们对应的 DCF 高于 t-vector 对应的 DCF。这说明 LF 和 GF 的性能没有 t-vector 好。这是因为, LF 和 GF 为基于统计学的特征, 容易受到轨迹中飞点的干扰, 而 t-vector 由 b-NN 生成得到。b-NN 将统计特征映射到“通用坐标空间”, 对特征值进行了校准, 从而降低了飞点干扰。



## 5.4. 轨迹分类模型性能测试

本实验测试分类模型的性能, 将提出的 TVM 和 SVM、RF、决策树(decision tree, DT)、贝叶斯网络(Bayes network, BN)和多层感知机(multilayers perceptron, MLP)。这些分类模型常用于完成车辆、动物、台风、飞行目标等轨迹分类任务实验中, b-NN 用于根据轨迹生成 t-vector, 然后将 t-vector 带入上述 5 种分类模型, 完成飞行目标轨迹分类。实验结果如表 2 所示。

**Table 2.** The performance of different classification models  
**表 2.** 不同分类模型的性能

特征	ACC(%)	DCF <sub>1</sub>	DCF <sub>2</sub>	DCF <sub>3</sub>	DCF <sub>4</sub>
SVM	85.3	0.073	0.075	0.080	0.079
RF	83.9	0.081	0.090	0.089	0.092
XG	88.7	0.063	0.067	0.070	0.071
BN	82.6	0.091	0.010	0.016	0.012
MLP	90.4	0.050	0.053	0.060	0.057
TVM	90.1	0.051	0.054	0.058	0.062

表显示 MLP 和 TVM 获得最高的 ACC 和最低的 DCF。表明 MLP 和 TVM 的分类性能最优。然而, 相对于 TVM, MLP 训练时长较长, 且对硬件要求较高。另外, SVM、RF 和 XG 的 ACC 均比 TVM 的低而它们的 DCF 均比 TVM 高。这是因为, TVM 综合考虑 SVM、RF 和 XG 三个分类器的输出结果, 如果其中立个分类器的结果有误, 其余两个分类器可以其纠正错误使得最终结果正确。换句话说, TVM 的容错性比单独的分类模型要好。

## 6. 结束语

轨迹分类是一种利用目标轨迹识别目标类型的技术。飞行目标轨迹分类是轨迹分类的一种, 通常应用于机场附近空中交通管理领域, 及时识别出离客机较近的飞行目标, 辅助相关人员采取措施阻止相撞事故, 保障客机起飞和降落时的安全。从轨迹中提取出优质特征, 以支撑目标分类任务是轨迹分类的研究重点和热点。针对飞行目标轨迹, 研究者提出局部特征和全局特征提取方式, 在轨迹质量较好时能够从轨迹中提取出优质的特征。但是这两种方法基于数理统计, 容易受到飞点干扰。为了减弱飞点对轨迹特征的干扰, 本论文提出了一种基于瓶颈神经网络的轨迹嵌入方法。该方法先从目标轨迹中得到高维度的局部 - 全局特征向量, 然后利用 b-NN 结构根据高维度特征向量生成 t-vector, 完成轨迹嵌入。由于 b-NN 能将局部 - 全部特征向量映射到低维度“通用坐标”空间中, 对特征值进行了校准并降低了特征向量维度, 使得 t-vector 具有较低维度和较好可分性。为了实现轨迹分类, 本论文还将 SVB、RF 和 XG 构成 TVM, 和独立分类模型相比, TVM 具有较好容错性。

未来的研究分为两个方向进行。一方面着重研究轨迹嵌入, 将对抗网络、自动编码机等高性能生成网络运用到轨迹嵌入中。另一方面着力研究分类器, 性能跟好的分类网络模型。

## 致 谢

首先感谢编辑和评审专家对本论文的编辑和点评指导, 其次感谢杜晓彤对本论文提供的实验数据, 最后感谢妻子夏向玲对我科研工作的理解和支持。

## 基金项目

四川省科技重大专项(2018GZDZX0012)。

## 参考文献

- [1] Silva, C.L., Petry, L.M. and Bogomy, V. (2019) A Survey and Comparison of Trajectory Classification Methods. 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), Salvador, 15-18 October 2019, 788-793. <https://doi.org/10.1109/BRACIS.2019.00141>
- [2] Hammad, A.W.A., Rey, D., Bu-Qammaz, A., et al. (2020) Mathematical Optimization in Enhancing the Sustainability of Aircraft Trajectory: A Review. *International Journal of Sustainable Transportation*, **14**, 413-436. <https://doi.org/10.1080/15568318.2019.1570403>
- [3] Murca, M.C.R., Hansman, R.J., Li, L., et al. (2018) Flight Trajectory Data Analytics for Characterization of Air Traffic Flows: A Comparative Analysis of Terminal Area Operations between New York, Hong Kong and Sao Paulo. *Transportation Research Part C: Emerging Technologies*, **97**, 324-347. <https://doi.org/10.1016/j.trc.2018.10.021>
- [4] Gaiduchenko, N.E. and Gritsyk, P.A. (2019) Hypersonic Vehicle Trajectory Classification Using Convolutional Neural Network. *Proceeding of International Conference on Engineering and Telecommunication*, Dolgoprudny, 20-21 November 2019, 1-4. <https://doi.org/10.1109/EnT47717.2019.9030537>
- [5] May, P.L., Leite, D.S., Esuli, A., et al. (2020) MARC: A Robust Method for Multiple-Aspect Trajectory Classification via Space, Time and Semantic Embedding. *International Journal of Geographical Information Science*, **34**, 1428-1450. <https://doi.org/10.1080/13658816.2019.1707835>
- [6] Lee, J.G., Han, J. and Li, X., et al. (2008) TraClass: Trajectory Classification Using Hierarchical Region-Based and Trajectory-Based Clustering. *VLDB Endowment*, **1**, 1081-1094. <https://doi.org/10.14778/1453856.1453972>
- [7] Ferrero, C.A., Alvares, L.O., Zalewski, W., et al. (2018) MOVELETS: Exploring Relevant Subtrajectories for Robust Trajectory Classification. *Proceeding of the 33rd Annual ACM Symposium on Applied Computing*, Pau, 9-13 April 2018, 849-856. <https://doi.org/10.1145/3167132.3167225>
- [8] Kowalek, P., Loch-Olszewska, H. and Szwabirinski, J. (2019) Classification of Diffusion Modes in Single-Particle Tracking Data: Feature-Based versus Deep-Learning Approach. *Physical Review E*, **100**, Article ID: 032410. <https://doi.org/10.1103/PhysRevE.100.032410>
- [9] Cui, P., Wang, X., Pei, J., et al. (2019) A Survey on Network Embedding. *IEEE Transactions on Knowledge and Data Engineering*, **31**, 833-852. <https://doi.org/10.1109/TKDE.2018.2849727>
- [10] Pandey, S., Mathur, I. and Joshi, N. (2021) Hybrid Model with Word2vector in Information Retrieval Ranking. In: Khanna, A., Gupta, D., Pólkowski, Z., Bhattacharyya, S. and Castillo, O., Eds., *Data Analytics and Management*, Vol. 54, 761-773. [https://doi.org/10.1007/978-981-15-8335-3\\_58](https://doi.org/10.1007/978-981-15-8335-3_58)
- [11] Katsurada, K. and Richmond, K. (2020) Speaker-Independent Mel-Cepstrum Estimation from Articulator Movements Using D-Vector Input. *Proceeding of Interspeech 2020*, Shanghai, 25-29 October 2020, 3176-3180. <https://doi.org/10.21437/Interspeech.2020-1630>
- [12] Mary, S., Metilda, N.J., Umesh, S., et al. (2021) S-Vector and TESA: Speaker Embeddings and a Speaker Authenticator Based on Transformer Encoder. *IEEE Transactions on Audio, Speech and Language Processing*, **30**, 404-413. <https://doi.org/10.1109/TASLP.2021.3134566>
- [13] Tang, W., Li, B., Tian, S., et al. (2019) CNN-Based Adversarial Embedding for Image Steganography. *IEEE TRANSACTIONS on Information Forensics and Security*, **14**, 2074-2087. <https://doi.org/10.1109/TIFS.2019.2891237>
- [14] Mazher, A. (2020) Visualization Framework for High-Dimensional Spatio-Temporal Hydrological Gridded Datasets using Machine-Learning Techniques. *Water*, **12**, Article No. 590. <https://doi.org/10.3390/w12020590>
- [15] Gutman, Y., Qiao, T. and Tsukamoto, M. (2019) Application of Signal Analysis to the Embedding Problem of  $Z^k$ -Actions. *Geometric and Functional Analysis*, **29**, 1440-1502. <https://doi.org/10.1007/s00039-019-00499-z>
- [16] Lu, Q., Silva, N., Kafle, S., et al. (2019) Learning Electronic Health Records through Hyperbolic Embedding of Medical Ontologies. *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, Niagara Falls, 7-10 September 2019, 338-346. <https://doi.org/10.1145/3307339.3342148>
- [17] Kinnunen, T. and Li, H. (2010) An Overview of Text-Independent Speaker Recognition: From Features to Supervectors. *Speech Communication*, **52**, 12-40. <https://doi.org/10.1016/j.specom.2009.08.009>
- [18] Wang, S., Yang, Y., Wang, T., et al. (2019) Knowledge Distillation for Small Foot-Print Deep Speaker Embedding. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, Brighton, 12-17 May 2019, 6021-6025. <https://doi.org/10.1109/ICASSP.2019.8683443>

- 
- [19] Dodge, S., Weibel, R. and Forootan, E. (2009) Revealing the Physics of Movement: Comparing the Similarity of Movement Characteristics of Different Types of Moving Objects. *Computers, Environment and Urban System*, **33**, 419-434. <https://doi.org/10.1016/j.compenvurbsys.2009.07.008>
- [20] Zhibin, X., Yang, W., Kun, F., *et al.* (2017) Identify Different Transportation Models from Trajectory Data Using Tree-Based Ensemble Classifiers. *International Journal of Geo-Information*, **6**, Article No. 57. <https://doi.org/10.3390/ijgi6020057>
- [21] Yang, X., Stewart, K., Tang, L., *et al.* (2018) A Review of GPS Trajectories Classification Based on Transportation Mode. *Sensors*, **18**, Article No. 3741. <https://doi.org/10.3390/s18113741>
- [22] Amjad R A and Geiger, B.C. (2020) Learning Representations for Neural Network-Based Classification Using the Information Bottleneck Principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **42**, 2225-2239. <https://doi.org/10.1109/TPAMI.2019.2909031>
- [23] Variani, E., Lei, X., McDermott, E., *et al.* (2014) Deep Neural Networks for Small Footprint Text-Dependent Speaker Verification. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Florence, 4-9 May 2014, 4052-4056. <https://doi.org/10.1109/ICASSP.2014.6854363>
- [24] Wang, B., Chen, Z., Zhu, A.X., *et al.* (2019) Multi-Level Classification Based on Trajectory Features of Time Series for Monitoring Impervious Surface Expansions. *Remote Sensing*, **11**, Article No. 640. <https://doi.org/10.3390/rs11060640>