文章编号：2095-2163(2020)09-0065-04 中图分类号：TP183 文献标志码：A

基于高斯核层次聚类的汽车工况构建

韩 鑫
（西安石油大学 计算机学院，西安 710065）

摘 要： 现有的车况构建主要采用 K-means 方法对运动学片段进行聚类，该方法需要通过经验确定聚类的个数，然而人工经验在数据量大和情况复杂时很容易带来误差。因此，本文在对不良数据进行处理、定义怠慢区并对运动学片段进行分割之后，构建基于高斯核的层次聚类算法，对片段进行聚类后确定构建工况的候选集，以解决这个难题。本文还引入统计特征、形状特征、熵特征等共 14 个运动学片段，作为聚类运动学片段的有效特征。根据运动学片段类别及时间比例，构建了 1 300 s 的工况图。实验结果表明，本文构建的工况图具一定的有效性和实用性。

关键词： 汽车工况构建；层次聚类；高斯核；核方法

Gaussian Kernel Based Hierarchical Clustering for Driving Cycle Construction

HAN Xin
（School of Computer Science, Xi′an Shiyou University, Xi′an 710065, China）

【Abstract】 The existing vehicle condition construction mainly uses the k-means method to cluster the kinematic segments. This method needs to determine the number of clusters through experience. However, manual experience is easy to bring errors when the amount of data is large and the situation is complex. Therefore, this paper constructs a hierarchical clustering algorithm based on Gaussian kernel after processing the bad data, defining the slack area and segmenting the kinematic segments, and determines the candidate set of construction conditions after clustering the segments to solve this problem. This paper also introduces 14 kinematic segments, such as statistical feature, shape feature and entropy feature, as the effective features of clustering kinematic segments. According to the category and time proportion of kinematic segments, the 1300 s working condition diagram is constructed. The experimental results show that the working condition diagram constructed in this paper is effective and practical.

【Key words】 Driving Cycle; Hierarchical clustering algorithm; Gaussian Kernel; Kernel Method

0 引 言

汽车行驶工况也被称为车辆测试循环，描述了汽车行驶速度的时间曲线（通常在 1 800 s 以内），反映了道路上汽车的运动特性[1]。其是汽车工业中重要且常见的基本技术、车辆的能耗排放测试方法和极限标准的基础，也是汽车各种性能指标的校准和优化的最重要基准。中国幅员辽阔，不同城市之间的发展程度、气候条件和交通条件的差异，使各个城市的驾驶条件特征明显不同。因此，作为车辆开发和评估的基础，越来越需要从城市自身的驾驶数据中进行汽车行驶工况构建的研究。

大多数已有研究在工况构建时选择 k-means 聚类方法作为聚类手段[2]，但由于 k-means 聚类需要提前确立数据中聚类的个数 k。根据已有的研究结果及经验，将类别分为 3 类或者 4 类[3]。然而，当数据量较大，采集数据情况复杂时，先验知识具有很大的局限性[4]。在不观察数据就确立类别数，势必会给聚类结果带来很大的误差。层次聚类算法可以返回一颗聚类树，从聚类树中可以得到所有的聚类结果供使用者选择，从而避免了选择聚类个数的问题。由于一般汽车工况特征比较复杂，极有可能导致数据在低维空间下不可分，而使用核方法特别是高斯核方法，可以将数据特征空间映射到高维甚至无限维的空间，从而更好地将数据分开[5]。因此，本文采用基于高斯核的层次聚类算法，对构建的车况特征进行聚类，提高聚类准确度。

1 特征定义

将收集的速度数据转换为特征参数数据的过程，可以视为数据转换。特性参数可以更好地表达短途行驶的情况，并且更有利于分析。在分割的短行程中只有速度和时间数据，但是仅使用速度和时间并不能完整地表征短行程运行的特征。因此，本文从统计信息、形状信息以及熵信息中共提取构建了 21 个特征。

1.1 统计特征

短行程的统计特征数据主要为速度、加速度的比例、均值、标准差、最大最小值等，速度与时间数据是直接采集的。由于采集频率为 1 HZ，所以对于任意

作者简介：韩 鑫（1997-），男，硕士研究生，主要研究方向：数据挖掘、聚类分析、异常检测。

收稿日期：2020-07-11

时刻 i，则有 $t_{i+1} - t_i = 1$，加速度计算如式(1)所示：

$$a_{i,i+1} = \frac{v_{i+1} - v_i}{t_{i+1} - t_i} \times \frac{1000}{3600} = \frac{v_{i+1} - v_i}{3.6}, \quad i = 1,2,\cdots. \tag{1}$$

其中，$a_{i,i+1}$ 为第 i 秒到第 $i+1$ 的加速度，m/s^2；v_i 为 i 秒的速度，km/h；t_i 为第 i 秒时刻，s。

(1)最大速度、平均速度、速度方差($v_{\max}, v_m, v_{me}, v_{sd}$) 的计算公式分别为：

$$\begin{aligned} &v_{\max} = \max\{v_1, v_2, \cdots v_k\}, \\ &v_m = mean\{v_1, v_2, \cdots v_k\} = S/T, \\ &v_{me} = mean\{v_j, v_{j+1}, \cdots v_k\} = S/(T - T_i), \\ &v_{sd} = std\{v_1, v_2, \cdots v_k\} = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(v_i - v_m)^2}. \end{aligned} \tag{2}$$

(2)最大加速度、最小加速度、平均加速度、平均减速度、加速度方差($a_{max}, a_{min}, a_a, a_d, a_{sd}$) 的计算公式分别为：

$$\begin{aligned} &a_{\max} = \max\{a_1, a_2, \cdots a_{k-1}\}, \\ &a_{\min} = \min\{a_1, a_2, \cdots a_{k-1}\}, \\ &a_a = \text{mean}\{a_i \mid a_i \geqslant 0.15\} = \sum\{a_i \mid a_i \geqslant 0.15\}/T_a, \\ &a_d = \text{mean}\{a_i \mid a_i \leqslant -0.15\} = \sum\{a_i \mid a_i \leqslant -0.15\}/T_d, \\ &a_{sd} = \text{std}\{a_1, a_2, \cdots a_{k-1}\} = \sqrt{\frac{1}{k-2}\sum_{i=1}^{k-1} a_i^2}. \end{aligned} \tag{3}$$

其中，T_a 为加速度大于 0.15 的时间；T_d 为减速度小于 0.15 的时间。

1.2 形状特征

除构建统计特征外，由于片段为时间序列，需要捕获速度在波形形状上的特征。最新研究表明，将偏度和峰度相结合是对序列相关性度量的有用特征。偏度是统计数据分布中偏斜方向和程度的度量，是统计数据分布中偏斜度的数值特征。峰度表示概率密度分布曲线的峰值在平均值处高度的数量特征。直觉上，峰度反映了峰的锐度[6]。

对于长度为 T 的时间序列 $X_T = \{x_1, \cdots, x_T\}$，其均值 μ 和方差 σ 分别为：

$$\mu = \frac{1}{T}\sum_{i=1}^{T} x_i, \tag{4}$$

$$\sigma^2 = \sum_{i=1}^{T} \frac{1}{T}(x_i - \mu)^2. \tag{5}$$

T 的偏度定义为其三阶标准化矩为：

$$\boldsymbol{Skew}(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{1}{T}\sum_{i=1}^{T}\frac{(x_i - \mu)^3}{\sigma^3}. \tag{6}$$

T 的峰度定义为其四阶中心矩与方差平方的比值：

$$Kurt(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] = \frac{E[(X-\mu)^4]}{(E[(X-\mu)^2])^2}. \tag{7}$$

1.3 序列熵特征

除构建片段统计特征和形状特征外，还需要描述片段的确定性或者稳定性。在本文中，对于速度片段的时间序列，加入 Binned 熵和 Approximate 熵用于分别度量速度片段的均匀性和稳定性。

Binned 熵考虑将时间序列 X_T 的取值进行分区操作。之后计算时间序列的取值分散在所有区域中的概率分布的熵。

$$\text{binned entropy}(X) = -\sum_{k=0}^{\min(\max bin, len))} p_k \ln(p_k) \cdot 1_{(p_k > 0)}. \tag{8}$$

其中，p_k 表示时间序列 X_T 的取值落在第 k 个桶的比例(概率)；maxbin 表示区域的个数；$len(X_T) = T$ 表示时间序列 X_T 的长度。

片段速度序列的 Binned 熵越大，说明这一段时间内速度取值的分布，在 $[\min(X_T), \max(X_T)]$ 之间越均匀。如果一个片段的速度序列的 Binned 熵值较小，说明这一段时间序列的取值是集中在某一段上。

Approximate 熵是为了判断一个序列是随机出现还是具有某种趋势。其基本思想是，把一维空间的时间序列映射到高维空间中，并通过高维空间向量之间的相似度判断，推导出一维空间的时间序列是否存在某种趋势或者确定性。

$$ApEn(m,r) = \Phi^m(r) - \Phi^{m+1}(r). \tag{9}$$

其中，$\Phi^m(r)$ 为一个 m 维的函数。

2 基于高斯核的层次聚类

层次聚类是一种常见的聚类算法，该算法能在不同的层次上对数据样本进行划分归类，而不需要提前确定聚类的类别的数量。同样，该算法适用于对样本不确定或缺乏领域知识时使用。通常，层次聚类可分为两种特定的策略。一是：将样本(小类)从底部到顶部(大类)进行分组的策略；二是拆分型层次聚类：将大类从顶部进行划分。根据研究对象及数据的具体情况，本文采用第一种凝聚型层次聚类策略。

凝聚型层次聚类的具体步骤，是将每个样本视为具有单个元素的单个聚类，然后计算类之间的距离(相异性)，合并具有最短距离的类(即最大的相

似性)，并遍历整个过程，逐步将小类合并，直到所有样本都在同一类中为止。设给定 n 个样本点 x_1，x_2，…，x_n，具体流程如下：

(1)将每个样本点视为一个类，并计算两个样本之间的距离 $dist(x_i,x_j)$；

(2)将两个最接近的类，合并为一个新类；

(3)更新类间的距离；

(4)重复(2)和(3)步骤，直到所有样本都被合并到一个类中/达到结束条件为止。

从层次聚类算法流程中可以看出，凝聚型层次聚算法的关键问题是，确立对象(样本)间，以及簇与簇之间的距离。而类与类之间的距离是根据不同的连接函数(如单连接、全连接)从样本间的距离产生。因此，两两样本之间的距离在算法中发挥着重要作用。在计算两个样本之间的距离时，传统的层次聚类法往往采用欧式距离。对于样本 x_i 和 x_j，其距离度量如式(10)所示。

$$dist(x_i,x_j) = \| x_i - x_j \|^2. \tag{10}$$

然而，基于欧式距离的凝聚型层次聚算法受噪声点的影响较大。当两个类的距离较近时，会由于少量距离较近的点优先合成一个簇，而实际两个类的大多数样本并不接近，从而造成聚类误差。基于欧式距离的凝聚型层次聚类算法，可看做是使用线性模型学习决策边界，由于它只能学习非常简单的线性决策边界，因此造成该算法对噪声点非常敏感，从而无法将类别有效的分开。对于在线性空间中无法分开的情况，可以将数据提高维度，在高维空间中找到分类边界，进而避免噪声点在原始空间的影响[5]。

本文采用高斯核度量的方法实现维数的增加，其定义如下：

$$K(x_i,x_j) = \exp\left(\frac{- \| x_i - x_j \|^2}{2\sigma^2}\right). \tag{11}$$

如式(12)所示，高斯核函数的特性是把低维空间转化为无限维空间，同时又实现了在低维计算高维点积。

$$k(x,y) = \langle \varphi(x),\varphi(y) \rangle = e^{-\sigma \| x-y \|^2} = e^{-\sigma(x^2+y^2)} e^{\sigma 2xy} = e^{-\sigma(x^2+y^2)} \left(1 + \frac{2\sigma xy}{1!} + \frac{(2\sigma xy)^2}{2!} + \frac{(3xy)^2}{3} + \cdots\right) = e^{-\sigma x^2} \left[\sqrt{\frac{2\sigma}{1!}} x^2\right] \left[\sqrt{\frac{(2\sigma)^2}{2}} y^2\right]. \tag{12}$$

若给定 n 个样本点 x_1，x_2，…，x_n，基于高斯核的凝聚型层次聚算法如下：

(1)将每个样本点视为一个类，并基于式(11)计算两个样本之间的距离；

(2)将两个最接近的类合并为一个新类；

(3)更新类间距离；

(4)重复(2)和(3)，直到所有类都被合并到一个类中/达到结束条件为止。

从高斯核凝聚型层次聚类算法流程可以看出，该算法将样本间的距离计算修改为基于高斯核函数的度量，其它则保持了原始算法的步骤。该算法在保证了原始层次聚类算法简单性的同时，又可提高算法在克服线性不可分情况的缺陷。

3　实　验

原始采集数据经过运动学片段的划分、筛选，采用基于高斯核的层次聚类结果，使用 TSNE 在二维空间中可视化的展示如图 1 所示。所有运动学片段可被分为 3 个类别，但每个类别中仍然有数百个运动学片段，则可从每个类别中提取适当的片段，这些片段应该尽可能完整地反映每种类型的片段特征，从而使构造的车况曲线可以客观地反映车辆的实际驾驶情况。

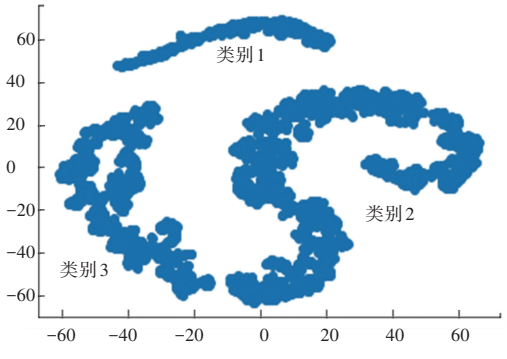

图 1　聚类结果图

Fig. 1　Result of clustering

通过分析每一类的运动学片段发现：第一类的加速、减速时间比例最低，怠速时间比例最高，说明汽车长时间怠速，但是起步加速与制动减速运行时间较短，第一类可代表汽车在拥堵的主干道上的交通特征；第二类的加速、怠速、减速时间比例均中平，匀速时间比例最高，表明汽车匀速行驶时间较长，同时也要经历一定的停车、怠速、起步，第二类可代表汽车在比较畅通的支干道上行驶的特征；第三类的匀速、怠速时间比例最低，加速、减速时间比例最高，代表汽车行驶中可以长时间加速、减速行驶，停车怠速时间很短，该类可代表汽车在通畅的城郊道路上行驶的特征。

从每个类中挑选运动学的一个片段，拼接成 1 300 s 的工况循环曲线，如图 2 所示。

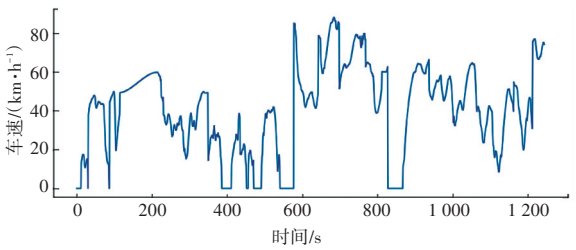

图 2 构建工况图

Fig. 2 Created Driving Cycle

由此可见,其结果完全符合汽车工况规律,具有有效性。

4 结束语

汽车行驶工况描述了汽车行驶速度的时间曲线,反映了道路上汽车的运动特性,是车辆的能耗排放测试方法和极限标准的基础,是汽车各种性能指标的校准和优化的最重要基准。本文在定义了包括统计特征、形状特征、熵特征等共计 14 个运动学片段的有效特征后,构建基于高斯核的层次聚类算法对片段进行聚类。根据运动学片段类别的比例及时间比例,从聚类结果的中抽取具有代表性的片段拼接成 1 300 s 的工况图。经试验结果表明,构建的工况图具有较大参考价值。

参考文献

[1] 徐小俊, 李君, 刘宇, 等. 电动汽车城市行驶工况构建[J]. 科学技术与工程, 2017, 17(35): 330-336.

[2]李洋. 基于聚类算法的汽车行驶工况研究[D]. 北京理工大学, 2016.

[3]石琴, 仇多洋, 周洁瑜, 等. 基于组合聚类法的行驶工况构建与精度分析[J]. 汽车工程, 2012, 34(2): 164 - 169.

[4] MURTAGH F. A survey of recent advances in hierarchical clustering algorithms [J]. The Computer Journal, Oxford University Press, 1983, 26(4): 354-359.

[5]STUETZLE W. Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample [J]. Journal of classification, Springer, 2003, 20(1): 25-47.

[6]VAN RIJSBERGEN C J. Information Retrieval. 2nd. Newton, MA [M]. USA: Butterworth-Heinemann, 1979.

[7] GOWER J C, ROSS G J. Minimum spanning trees and single linkage cluster analysis [J]. Journal of the Royal Statistical Society: Series C (Applied Statistics), Wiley Online Library, 1969, 18(1): 54-64.

(上接第 64 页)

发生了变化,但变化幅度并不大,方便铁路部门调整,也不会由于变化太大而给旅客带来不便。再结合客流 *OD* 表 6 和表 7 分析可知,优化调整后的列车开行方案更能满足日益增长的客流需求,为广大城际铁路列车乘客带来更多的便利。

表 9 优化后城际列车开行方案表

Tab. 9 Table of optimized intercity operation plan

列车种类	停站方案	开行数量
A	1→7	5
B	1→2→7	6
B	1→4→7	3
B	1→2→4→7	6
B	1→3→4→7	6
B	1→4→5→7	4
B	1→4→6→7	5
B	1→2→3→4→7	4
B	1→2→4→6→7	5
B	1→4→5→6→7	6
C	1→2→3→4→5→6→7	3

表 10 优化前后指标对比表

Tab. 10 Comparison table before and after optimization

	优化前	优化后
列车总开行对数/辆	49	50
一站直达列车数/辆	4	5
站站停列车数/辆	5	3
择站停列车数/辆	40	42
旅客出行成本/h	183 440.13	178 350.93
企业运营成本/元	2 559 000	2 580 000

5 结束语

从本文算例分析结果看,优化调整后的列车开行方案相较当前开行方案调整幅度不大,方便铁路部门调整,也不会由于变化太大而给旅客带来不便,更能满足日益增长的客流需求,因此该方法对长期的城际铁路列车开行方案周期性评价与优化具有很好的借鉴意义。但对于判断阈值的确定,还有待进一步的探索研究。

参考文献

[1] 于剑,张星臣,徐彬,等. 城市轨道交通过轨开行方案编制技术[J]. 城市轨道交通研究,2015,18(11):18-22,34.

[2] 邓连波,王峰,史峰,等. 城市直达列车开行方案研究[J]. 铁道科学与工程报,2013,10 (6):97-102.

[3] SCHÖBEL A, SCHOLL S. Line planning with minimal traveling time[C]//5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS´05). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2006.

[4]BUSSIECK M R,KREUAER P,ZIMMEMANN U T. Optimal lines for railway systems[J].European Journal of Operational Research, 1997,96(1):54-63.

[5]王正彬,马骐. 城际客运专线列车开行方案模型与算法[J]. 交通运输工程与信息学报,2017,15(1):28-33.

[6] 殷路,马彩雯. 城际客运专线列车开行方案评价方法研究[J]. 大连交通大学学报,2015,36(6):6-9,14.

[7] 雷英杰,张善文. MATLAB 遗传算法工具箱及应用[M]. 西安:西安电子科技大学出版社,2015.

[8] CHANG Y H, YEH C H, SHEN C C. A Muti-objiective model for passenger train services planning: application to Taiwan's High-speed Rail Line [J]. Transportation Research Part B Metodological,2000,34(2):91-106.