软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(3):465−475 [doi: 10.3724/SP.J.1001.2013.04248]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

计算机兵棋作战实体轨迹聚类算法*

石崇林[1], 淦文燕[2], 吴 琳[3], 张茂军[1], 唐宇波[3]

[1](国防科学技术大学 信息系统与管理学院,湖南 长沙 410073)

[2](中国人民解放军理工大学 指挥自动化学院,江苏 南京 210007)

[3](国防大学 信息作战与指挥训练教研部,北京 100091)

通讯作者: 石崇林, E-mail: chonglinshi1982@163.com

摘 要: 针对计算机兵棋系统的实际应用,提出计算机兵棋实体轨迹聚类算法——CTECW(clustering trajectories of entities in computer wargames).算法分为 3 部分:轨迹预处理、轨迹分段聚类以及可视化表现.轨迹预处理将实体原始轨迹转化成实体简化轨迹,再进一步处理成轨迹分段;在 DBSCAN 算法的基本框架下引入 DENCLUE 算法中密度函数的概念,并基于提出的相似性度量函数对轨迹分段进行聚类;可视化表现将轨迹分段聚类的结果以赋有军事涵义的形式展现给参与兵棋推演的受训指挥员,体现出算法的实际应用价值.理论分析与实验结果表明,CTECW 算法能够得到与 TRACLUS 算法比较接近的聚类结果,但计算效率却比 TRACLUS 算法要高,并且聚类结果不依赖于用户参数的仔细选择.

关键词: 计算机兵棋;数据挖掘;轨迹聚类;相似性度量;密度估计熵

中图法分类号: TP181 **文献标识码**: A

中文引用格式: 石崇林,淦文燕,吴琳,张茂军,唐宇波.计算机兵棋作战实体轨迹聚类算法.软件学报,2013,24(3):465−475. http://www.jos.org.cn/1000-9825/4248.htm

英文引用格式: Shi CL, Gan WY, Wu L, Zhang MJ, Tang YB. Clustering trajectories of entities in computer wargames. Ruanjian Xuebao/Journal of Software, 2013,24(3):465−475 (in Chinese). http://www.jos.org.cn/1000-9825/4248.htm

Clustering Trajectories of Entities in Computer Wargames

SHI Chong-Lin[1], GAN Wen-Yan[2], WU Lin[3], ZHANG Mao-Jun[1], TANG Yu-Bo[3]

[1](School of Information System and Management, National University of Defense Technology, Changsha 410073, China)

[2](Institute of Command and Automation, PLA University of Science and Technology, Nanjing 210007, China)

[3](Department of Information Operation & Command Training, National Defense University, Beijing 100091, China)

Corresponding author: SHI Chong-Lin, E-mail: chonglinshi1982@163.com

Abstract: Under the background of a computer wargame system, a trajectory clustering algorithm named CTECW (clustering trajectories of entities in computer wargames) is proposed. The algorithm is composed of three parts: trajectory pretreatment, trajectory segments clustering, and visual presentation. Trajectory pretreatment transforms original trajectories into simplified ones which are ulteriorly processed into linear segments. In the second part, the concept of density function derived from DENCLUE is introduced and trajectory segments are clustered based on similarity measure under the framework of DBSCAN. The visual presentation exhibits clusters of trajectory segments with martial meanings to trainees, which embodies practical values of CTECW. Both theoretical analysis and experimental results indicate that CTECW could acquire approximate clusters more efficiently compared with TRACLUS and requires no input parameters.

Key words: computer wargame; data mining; trajectory clustering; similarity measure; density estimation entropy

* 基金项目: 中国博士后科学基金(201003746)

收稿时间: 2011-08-02; 定稿时间: 2012-04-01

运动对象的时空轨迹蕴含着丰富的信息,对大量轨迹数据的分析和理解,对于诸如交通控制、气象监测、商业决策、军事分析等领域都具有十分重要的意义,但同时也是一项富有挑战性的工作.

计算机兵棋系统作为一种特殊的训练模拟系统,由于具备节省训练经费、不受场地限制、接近真实体验等优点[1],很多国家都用其作为部队作战指挥能力训练的有效工具.计算机兵棋系统通常采用棋子军标代表各种作战实体,以棋盘地图(一般采用六角格地图)代表战场空间,推演过程中没有形象直观的行动过程和交战场面,给受训指挥员"读棋"带来了一定的难度.随着计算机技术的不断发展,现代计算机兵棋系统可以实时记录并存储每个作战实体(以下简称实体)的运动轨迹.这些轨迹除了能够描述每个实体自身的运动信息之外,还能反映出实体之间的整体行为模式.直接将大量杂乱无章的实体轨迹呈现给受训指挥员是没有意义的,必须对这些轨迹进行约简,轨迹聚类是一种行之有效的约简方法.

目前,轨迹聚类方法主要有对整条轨迹进行聚类和先对轨迹进行分段后再对轨迹分段进行聚类.由于在兵棋推演的初始阶段实体分布比较广,对绝大多数实体的整条轨迹而言并不具有相似性.基于以上考虑,本文提出了计算机兵棋实体轨迹聚类算法:CTECW(clustering trajectories of entities in computer wargames).该算法采用了先对轨迹进行分段再对轨迹分段进行聚类的基本思想,目的是通过对轨迹的有效聚类,挖掘出描述战场态势并能够直接被受训指挥员接收的信息,以辅助指导训练,辅助事后讲评,充分发挥计算机兵棋系统在指挥能力培养和作战理论探索上的优势.

本文第 1 节回顾轨迹聚类的相关工作.第 2 节详细描述 CTECW 算法.第 3 节通过实验数据和结果对比说明 CTECW 算法的有效性.第 4 节是对全文工作的总结以及对未来工作的展望.

1 相关研究工作

对轨迹聚类的研究通常涉及到 3 个方面的内容:轨迹描述、轨迹相似性度量以及聚类算法.

关于轨迹描述:很多研究人员针对不同的应用背景提出了各种模型,总结起来可分为几何描述[2,3]与符号描述[4,5]两种方法.几何描述是指将轨迹采用一系列带时间戳的坐标点来表示,这一方式可以较精确地反映轨迹的几何形状,精确性取决于采样点的数量,但实际应用中需要在精确性和简化性之间折衷考虑.符号描述是指将轨迹表示成区域符号的序列,这一方式在数据存储和计算效率上都有优势,但前提是需要对空间进行划分,因此存在划分粒度问题:粒度过大会导致某些轨迹模式丢失;过小会导致原本相似的模式无法被提取出来.

关于轨迹相似性度量:Vlachos 等人[6]使用基于最长公共序列(longest common subsequence,简称 LCSS)的相似性函数来发现相似的多维轨迹.在他们的实验中,基于 LCSS 的聚类方法表现出了比基于欧氏距离和动态时间调整(dynamic time warping,简称 DTW)更好的性能.Yanagiswa 等人[7]专注于从时空轨迹中提取个体的运动模式,他们定义了一个基于轨迹形状的相似性距离函数,该函数在轨迹平移、旋转以及缩放情况下保持不变.Zhang 等人[8]总结了多种常用的轨迹相似性度量方法,包括欧氏距离、主成分分析(principal components analysis,简称 PCA)加欧氏距离、豪斯多夫距离(Hausdroff)距离、LCSS、DTW 和隐马尔科夫模型(hidden Markov models,简称 HMM),并从识别率和时间消耗上对 PCA 加欧氏距离的方法给予了肯定的评价.Hwang 等人[9]研究了路网空间中的轨迹相似性问题,提出了基于关注点(points of interest,简称 POI)的相似性函数,但前提是必须预先定义这些关注点,如果一个错误的点被选中,将会严重影响聚类的结果.Lee 等人[2]提出的距离函数由轨迹段之间的垂直距离、平行距离和角度距离这 3 部分组成,不受轨迹段长短的限制,较全面地度量轨迹段之间的相似度.

关于轨迹聚类算法:Gaffney 等人[10]发现,基于向量的轨迹在某些情况下是不充分的.因此,他们引入了概率衰退混合模型,并说明了如何在轨迹聚类中使用 EM 算法.不过,他们仅考虑了整条轨迹的相似,而忽略了相似的分段轨迹.Nanni 等人[11]提出的基于密度的轨迹聚类算法 TF-OPTICS 能够支持交互式搜索以发现最佳的聚类结果,不过他们也仅考虑了整条轨迹的相似性.Lee 等人[2]围绕分段轨迹的相似问题提出了 TRACLUS 算法,他们首先将轨迹分割成线段,然后采用基于密度的聚类算法对轨迹分段进行聚类.Pelekis 等人[12]利用模糊集理论对轨迹的不确定性进行建模,并提出了 CenTR-I-FCM 算法对轨迹进行聚类.Chen 等人[13]提出了 DENTRAC 算法,

该算法的特别之处在于它是对无参数的轨迹密度函数进行操作的.

Morris 等人[14]在 6 个不同的数据集上比较了 6 种轨迹相似度函数和 7 种聚类算法的性能,实验结果表明,在没有任何先验知识的情况下,没有哪一种相似性函数和聚类算法能够显示出绝对的优势.因此,采用哪种相似性函数和聚类算法应结合具体的应用背景,以取得最佳的聚类效果[15].从所查阅的文献来看,与本文工作比较接近的是具有分段-组合结构的 TRACLUS 算法.但该算法存在以下不足:

1) TRACLUS 算法中采用 MDL 方法对轨迹进行分段,但此方法并不能每次都找出最优的分段,并且计算较复杂;
2) TRACLUS 算法采用的距离函数不满足三角不等式关系,使得传统的空间索引技术无法直接应用,算法的复杂度也就无法通过采用高效的空间索引方法来降低;
3) TRACLUS 算法在对分段聚类并提取了每个聚类的代表轨迹后,没有对代表轨迹做进一步的处理和分析,没有考虑代表轨迹之间的联系.这在本文的应用背景下是无法满足要求的,因为代表轨迹依然不能直观地描述战场的整体态势.

针对以上的不足,本文提出的 CTECW 算法首先对原始的实体轨迹进行预处理,将轨迹处理成轨迹分段;再基于提出的相似性度量函数对轨迹分段进行聚类;最后对聚类结果进行可视化表现,以描绘出整体的战场态势.图 1 给出了 CTECW 算法的总体框架.

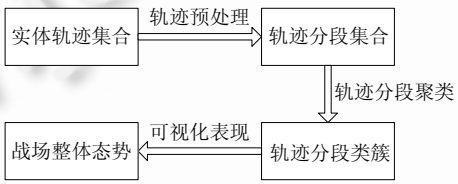


Fig.1 General framework of CTECW algorithm

图 1 CTECW 算法总体框架

2 CTECW 算法

2.1 轨迹预处理

定义 1(实体原始轨迹). 实体原始轨迹指在兵棋推演过程中,将系统记录下的实体经纬坐标点按推演时间的前后顺序组成的序列.用 PT 表示实体原始轨迹,$PT=p_1p_2\ldots p_kp_{k+1}\ldots p_n$,其中,$p_k=(long_k,lat_k)$称为原始轨迹点.

计算机兵棋系统通常将实际的地图量化成六角格或四方格,尤以六角格最为常见.这是因为在相同的采样密度下,六角格的逼近质量比四方格的一致性更好[16].在计算机兵棋系统中,实体只能沿着六角格的 6 个对边方向机动.基于这样的前提条件,我们定义了实体简化轨迹的概念.

定义 2(实体简化轨迹). 从实体原始轨迹的起点开始遍历整条轨迹,将同在一个六角格内且顺序上相邻的原始轨迹点用所在六角格的中心点代替,所得到的六角格中心点的序列即为实体简化轨迹.用 ST 表示实体简化轨迹,$ST = p_{h_1}p_{h_2}\ldots p_{h_k}\ldots p_{h_m}$,其中,$p_{h_k}=(x_{h_k},y_{h_k})$ 为六角格中心点,也称为简化轨迹点.

实体简化轨迹能够表现兵棋推演模型实际计算所采用的运动轨迹,而且由于每个六角格都是大小相等的规则区域,六角格中心点坐标可以用建立的六角格索引坐标来代替,避免了直接使用经纬度坐标计算距离、角度等几何特性所需的复杂运算.因此,实体简化轨迹的概念结合了两种轨迹描述方式的优点,既体现了几何描述方式的精确性,又有区域符号的简明性.图 2 给出了实体原始轨迹 $PT=p_1p_2\ldots p_{10}$ 的实体简化轨迹是 $ST = p_{h_1}p_{h_2}p_{h_3}p_{h_4}p_{h_5}p_{h_6}$,$ST$ 相比 PT 减少了 40%的数据量.下文所说的实体轨迹如果没有特别说明,均指实体简化轨迹.

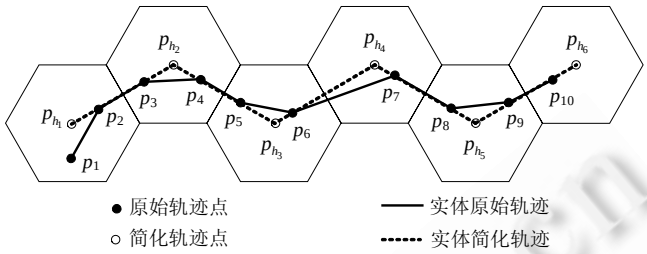


Fig.2 Original trajectory vs. simplified trajectory

图 2 实体原始轨迹 vs.实体简化轨迹

观察图 2 可以发现,由于兵棋推演模型中机动方向的限制,可能导致部分实体轨迹段成锯齿状,而这样的锯齿状轨迹段在描述实体真实机动意图上意义不大.若将这些锯齿状轨迹段简化成直线段,不仅不会影响轨迹的行为描述,还可以进一步减小数据量,更利于提高聚类算法的计算效率.此外,之前也提到,整条轨迹可能无法揭示实体共同的行为模式,而轨迹的局部却能表现出相似性.因此,我们提出了以下两个定义.

定义 3(轨迹特征点). 对某实体轨迹 $ST = p_{h_1} p_{h_2} \cdots p_{h_k} \cdots p_{h_m}$, p_{h_1} 和 p_{h_m} 为固定的轨迹特征点,除此之外,对给定的距离阈值 d,若 p_{h_i} ($h_1 \leqslant h_i < h_m$)为轨迹特征点,则 p_{h_j} ($h_j > h_i$)成为下一个轨迹特征点的条件是:

(1) $\forall p_{h_k}$, $h_i < h_k < h_j$,令 $dist_{\perp}(p_{h_k}, p_{h_i} p_{h_j})$ 表示为 p_{h_k} 到线段 $p_{h_i} p_{h_j}$ 的距离,则 $dist_{\perp}(p_{h_k}, p_{h_i} p_{h_j}) \leqslant d$;

(2) $\forall p_{h_k}$, $h_i < h_k < h_{j+1}$,使得 $dist_{\perp}(p_{h_k}, p_{h_i} p_{h_{j+1}}) > d$; 否则, $p_{h_{j-1}} = p_{h_{j+1}}$.

定义 4(轨迹分段). 对实体轨迹 $ST = p_{h_1} p_{h_2} \cdots p_{h_k} \cdots p_{h_m}$,若 $\{p_{c_1}, p_{c_2}, \ldots, p_{c_l}\}$ ($c_1 < c_2 < \ldots < c_l$)为 ST 上特征点集合,则轨迹分段是指每相邻两个特征点所构成的有向线段,记为 $\overrightarrow{p_{c_i} p_{c_{i+1}}}$ ($c_1 \leqslant c_i < c_{l-1}$).

定义 3 和定义 4 告诉我们如何将整条轨迹处理成轨迹分段:根据定义 3 找到实体轨迹的所有特征轨迹点,再按照定义 4 组织成轨迹分段.图 3 给出了一个示例:如果将 d 设为六角格中心点到边的距离值,那么实体轨迹 $ST = p_{h_1} p_{h_2} \cdots p_{h_{13}}$ 的所有轨迹特征点为 $\{p_{c_1}, p_{c_2}, \ldots, p_{c_6}\}$,$ST$ 将被处理成 5 个轨迹分段(如 $\overrightarrow{p_{c_1} p_{c_2}}$),减少了超过一半的数据量.

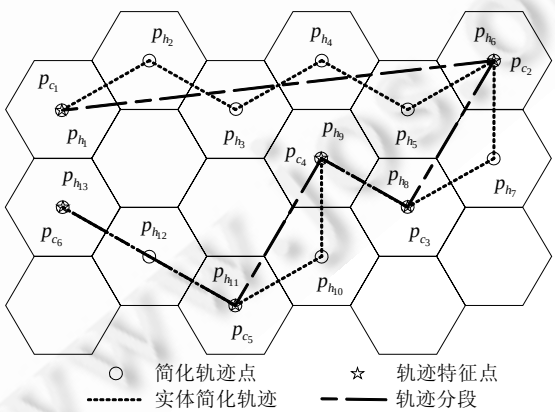


Fig.3 Simplified trajectory vs. trajectory segments

图 3 实体简化轨迹 vs.轨迹分段

轨迹预处理的过程就是将实体原始轨迹转化成实体简化轨迹,再进一步处理成轨迹分段的过程.实际上,这两个步骤可以合二为一,在将实体原始轨迹转化成实体简化轨迹的过程中,当得到某一简化轨迹点后,就可以判断前一个简化轨迹点是否是特征轨迹点.

2.2 相似性度量

相似度是定义一个聚类的基础,聚类的质量取决于对度量标准的选择,轨迹聚类算法一般采用样本间的距离作为度量标准.前面已经提到,TRACLUS 算法采用的距离函数虽然充分考虑了两个有向线段的长度、夹角以及空间位置等因素,能够较全面地度量其相似度,但是不完全满足度量(metric)性质(非负性、对称性以及三角不等式关系).因此,本文针对这一不足对轨迹分段距离重新做了定义,定义的理解可以结合图 4 的直观描述.假设有 3 个轨迹分段 $L_i=s_ie_i$,$L_j=s_je_j$ 和 $L_k=s_ke_k$,它们的中点分别是 c_i,c_j 和 c_k.

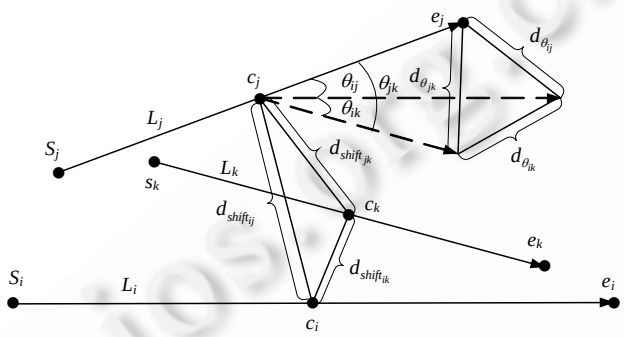


Fig.4 Distance description of trajectory segments

图 4 轨迹分段距离描述

定义 5(平移距离). 轨迹分段 L_i 和 L_j 之间的平移距离是指 L_i 和 L_j 中点之间的距离,即: $d_{shift_{ij}}=\|c_ic_j\|$.其中,$\|c_ic_j\|$代表 c_i 和 c_j 之间的欧氏距离.

定义 6(角度距离). 轨迹分段 L_i 和 L_j 之间的角度距离是指将 L_i 平移至 c_i 和 c_j 重合时,e_i 和 e_j 之间的距离,即: $d_{\theta_{ij}}=\sqrt{\|c_ie_i\|^2+\|c_je_j\|^2-2\|c_ie_i\|\times\|c_je_j\|\times\cos(\theta_{ij})}$.其中,θ_{ij} 为 L_i 和 L_j 之间的夹角,$0\leqslant\theta_{ij}\leqslant\pi$.

定义 7(轨迹分段距离). 轨迹分段距离由平移距离和角度距离两部分组成,形式为

$$dist(L_i,L_j)=\omega_{shift}\cdot d_{shift_{ij}}+\omega_\theta\cdot d_{\theta_{ij}},$$

其中,ω_{shift} 和 ω_θ 的值取决于不同的应用背景,默认情况下,ω_{shift} 取 1,ω_θ 取 2.

从图 3 的直观描述中可以看出,轨迹分段距离同样综合考虑了有向线段的长度、夹角以及空间位置等因素,相比 TRACLUS 算法中的距离函数,轨迹分段距离还满足度量性质.

定理 1. 轨迹分段距离满足度量性质.

证明:显然,$dist(L_i,L_j)$是满足非负性和对称性的,我们主要证明它还满足三角不等式关系.如图 4 所示,对任意的有向线段 L_i,L_j 和 L_k,有

$$dist(L_i,L_k)=\omega_{shift}\cdot d_{shift_{ik}}+\omega_\theta\cdot d_{\theta_{ik}}\leqslant\omega_{shift}\cdot(d_{shift_{ij}}+d_{shift_{jk}})+\omega_\theta\cdot(d_{\theta_{ij}}+d_{\theta_{jk}}),$$

而

$$\omega_{shift}\cdot(d_{shift_{ij}}+d_{shift_{jk}})+\omega_\theta\cdot(d_{\theta_{ij}}+d_{\theta_{jk}})=(\omega_{shift}\cdot d_{shift_{ij}}+\omega_\theta\cdot d_{\theta_{ij}})+(\omega_{shift}\cdot d_{shift_{jk}}+\omega_\theta\cdot d_{\theta_{jk}})=dist(L_i,L_j)+dist(L_j,L_k).$$

即 $dist(L_i,L_k)\leqslant dist(L_i,L_j)+dist(L_j,L_k)$.故三角不等式关系也满足. □

定理 1 可以保证在下一步的轨迹分段聚类中直接采用高效的空间索引技术来降低算法的复杂度,以提高聚类的效率.

2.3 轨迹分段聚类

对轨迹分段进行聚类比较适合采用基于密度的聚类方法,因为此类方法无须事先知道类簇的个数,可以发现任意形状和大小且含有噪声的类簇.考虑到轨迹中含有大量的噪声,本文基于 DBSCAN 算法[17]的基本框架来完成轨迹分段聚类的工作,这与 TRACLUS 算法是类似的.不同之处在于,本文借鉴了 DENCLUE 算法[18]中密度

函数的概念,核密度估计方法不利用任何有关数据分布的先验知识,对数据分布不附加任何假定,是一种从数据样本本身出发的研究数据分布特征的方法,因而在统计学理论和其应用领域如聚类分析中受到高度重视.

首先明确算法中的相关定义,设 $D=\{L_1,L_2,\ldots,L_n\}$为所有轨迹分段的集合.

定义 8(密度函数). 轨迹分段 $L\in D$ 的密度函数为 $f^D(L)=\sum_{L_i\in near(L)}K\left(\frac{dist(L,L_i)}{\sigma}\right)$,其中,$K(\cdot)$为核函数,如高斯函数;$near(L)=\{L_i|L_i\in D,dist(L,L_i)\leqslant k\sigma\}$;常数$\sigma$为核函数的窗宽.

定义 9(核心轨迹分段). 给定密度门限值ξ,若轨迹分段 $L\in D$ 的密度估计 $f^D(L)\geqslant\xi$,并且$\exists L_i\in near(L)$,L_i 与 L 不属于同一条轨迹,则 L 称为核心轨迹分段.

定义 10(直接密度可达). 若 $L\in D$ 为核心轨迹分段,则$\forall L_i\in near(L)$,称 L_i 从 L 出发直接密度可达.

定义 11(密度可达). 对轨迹分段链 $L_j,L_{j-1},\ldots,L_{i+1},L_i\in D$,若 L_k 从 L_{k+1} 出发直接密度可达,则称 L_i 从 L_j 密度可达.

定义 12(密度相连). 若$\exists L_k\in D$,使得 $L_i,L_j\in D$ 均从 L_k 密度可达,则称 L_i 和 L_j 是密度相连的.

定义 13 (任意形状的类簇). 对非空子集 $C\subseteq D$,给定窗宽σ和密度门限值ξ,若满足以下两个条件,则称 C 为由σ和ξ确定的任意形状的类簇:

(1) $\forall L_i,L_j\in C$,L_i 和 L_j 是密度相连的;

(2) $\forall L_i,L_j\in D$,若 $L_i\in C$ 且 L_j 从 L_i 密度可达,则 $L_j\in C$.

基于以上的几个定义,轨迹分段聚类算法的基本步骤如下:

输入:$D=\{L_1,L_2,\ldots,L_n\}$,σ,ξ;

输出:聚类集合 $O=\{C_1,C_2,\ldots,C_m\}$.

步骤 1. 初始化所有轨迹段.令 $i\leftarrow 1$,$k\leftarrow 1$.

步骤 2. 若 $L_i\in D$ 未被处理,采用高斯核密度函数 $f^D(L_i)=\sum_{L_j\in near(L_i)}e^{-\frac{dist(L_i,L_j)^2}{2\sigma^2}}$ 计算其核密度估计,根据高斯分布的 3σ规则,$near(L_i)$中 k 取 4;否则,跳至步骤 6.

步骤 3. 如果 L_i 是核心轨迹分段,则生成包含 $near(L_i)$中轨迹段的类簇 C_k;否则,跳至步骤 6.

步骤 4. 检测 C_k 中所有未被处理的轨迹段 L_p 是否为核心轨迹分段.若是,则将 $near(L_p)$内未被包含在 C_k 中的轨迹分段加入到 C_k 中.

步骤 5. 重复执行步骤 4 直到没有新的轨迹分段被加入到当前 C_k 中.令 $k\leftarrow k+1$.

步骤 6. 令 $i\leftarrow i+1$,若 $i\leqslant n$,转至步骤 2 执行;否则,算法结束.

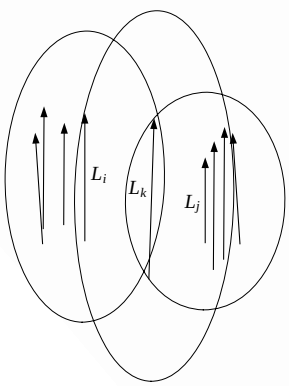


Fig.5 TRACLUS vs. CTECW

图 5 TRACLUS 算法 vs.本文算法

通过图 5 进一步解释本文算法与 TRACLUS 算法之间的区别.

如图 5 所示,在 TRACLUS 算法中,若 MinLns=5,L_i,L_j 和 L_k 密度相同且均为核心轨迹段,那么图 5 中的轨迹段将被聚在同一个类中;而在本文的算法中,由于 $f^D(L_j)>f^D(L_i)>f^D(L_k)$,若$\xi>f^D(L_k)$,则会因为 L_k 不是核心轨迹段而使得 L_i 和 L_j 不会被聚在同一类中.因此,本文提出的算法在密度的定义上较 TRACLUS 算法更加精细,从而聚类的结果也就比 TRACLUS 算法更加细腻.

2.4 参数值选择

算法中有两个重要参数:窗宽σ和密度门限值ξ.其中,σ值会影响密度函数的估计结果.在σ值确定的情况下,ξ值决定了核心轨迹分段的个数.

关于窗宽σ的选择,文献[19]将其看作是密度估计熵的最小化问题,是针对点数据的,我们针对轨迹分段对

密度估计熵的概念重新进行定义.

定义 14(密度估计熵). 设 $D=\{L_1,L_2,\ldots,L_n\}$为所有轨迹分段的集合,每个轨迹分段的密度函数值为 $f^D(L_i)$, $i=1,\ldots,n$,则密度估计熵可表示为 $Density\_Entropy=-\sum_{i=1}^{n}\frac{f^D(L_i)}{Z}\log\left(\frac{f^D(L_i)}{Z}\right)$,其中, $Z=\sum_{i=1}^{n}f^D(L_i)$ 为标准化因子.

对密度估计熵重新定义后,窗宽σ的值可以采用文献[19]中的方法进行确定.一旦确定了σ的取值,就可以得到每个轨迹分段的密度估计值,继而可以从所有轨迹分段密度估计值的分布结构上选择合适的ξ值.若将轨迹分段按密度估计值划分区间,绘制出区间与区间内轨迹分段数量之间的关系曲线图.理论上,曲线会在低密度区间和高密度区间之间形成波谷,合适的ξ值应在波谷偏右的区间内进行选择.在第 3 节会进一步说明如何选择合适的参数.

2.5 可视化表现

轨迹分段聚类完成后,其聚类结果必须以可视化的方式展现给受训指挥员,轨迹聚类工作才具有实际意义. TRACLUS 算法仅仅提取了每个类簇的代表轨迹,而没有考虑类簇之间的联系.本文如果仅仅将代表轨迹展示给受训指挥员,依然不能为他们提供整体直观的战场态势,轨迹聚类的军事意义也就不能得到很好的体现.因此,本文在得到聚类结果后,首先提取出每个类簇的统计特征,包括代表方向和代表位置;再检测每个类簇的后继关系,并根据所有的后继关系形成后继关系链;最后,根据后继关系链绘制出作战态势图.在描述可视化表现算法之前,先明确几个概念.

定义 15(类簇代表方向和代表位置). 假设 $C=\{L_p,L_{p+1},\ldots,L_{q-1},L_q\}$为某轨迹分段类簇,那么代表方向 $\vec{C}$ 可表示为 $\vec{C}=\sum_{i=0}^{q-p}\omega_{p+i}\overrightarrow{L_{p+i}}$;代表位置$(C_x,C_y)$可表示为 $C_x=\sum_{i=0}^{q-p}\omega_{p+i}s_x^{p+i}$, $C_y=\sum_{i=0}^{q-p}\omega_{p+i}s_y^{p+i}$.其中, $\overrightarrow{L_{p+i}}$ 代表轨迹分段 L_{p+i} 的方向,$\|L_{p+i}\|$代表 L_{p+i} 的长度, (s_x^{p+i},s_y^{p+i}) 代表 L_{p+i} 的起点坐标, $\omega_{p+i}=\frac{\|L_{p+i}\|}{\sum_{i=0}^{q-p}\|L_{p+i}\|}$.

定义 16(类簇最小外包矩形). 所谓类簇最小外包矩形是指能够包围类簇中所有轨迹分段的最小矩形.假设 $C=\{L_p,L_{p+1},\ldots,L_{q-1},L_q\}$为某轨迹分段类簇,则 C 的最小外包矩形的顶点坐标分别为

$$(\min\{\min(s_x^{p+i},e_x^{p+i})\mid L_{p+i}\in C\},\min\{\min(s_y^{p+i},e_y^{p+i})\mid L_{p+i}\in C\}),$$
$$(\min\{\min(s_x^{p+i},e_x^{p+i})\mid L_{p+i}\in C\},\max\{\max(s_y^{p+i},e_y^{p+i})\mid L_{p+i}\in C\}),$$
$$(\max\{\max(s_x^{p+i},e_x^{p+i})\mid L_{p+i}\in C\},\min\{\min(s_y^{p+i},e_y^{p+i})\mid L_{p+i}\in C\}),$$
$$(\max\{\max(s_x^{p+i},e_x^{p+i})\mid L_{p+i}\in C\},\max\{\max(s_y^{p+i},e_y^{p+i})\mid L_{p+i}\in C\}),$$

其中, (s_x^{p+i},s_y^{p+i}) 代表 L_{p+i} 的起点坐标, (e_x^{p+i},e_y^{p+i}) 代表 L_{p+i} 的终点坐标,$0\leqslant i\leqslant q-p$.

定义 17(后继关系与后继关系链). 对于类簇 C_i 和 C_j,若$\exists L_p\in C_j,L_q\in C_i$,使得 $(e_x^q,e_y^q)=(s_x^p,s_y^p)$,则称 C_j 为 C_i 的后继关系,记为 $C_i\to C_j$.若 $C_i\to C_{i+1}(p\leqslant i<q)$,则 $C_p\to C_{p+1}\to\ldots\to C_{q-1}\to C_q$ 称为后继关系链.

基于以上概念,我们给出可视化表现算法的基本步骤:

输入:聚类集合 $O=\{C_1,C_2,\ldots,C_m\}$;

输出:带箭头的战场态势图.

步骤 1. 按照定义 15 计算每个类簇的代表方向和代表位置.

步骤 2. 按照定义 16 计算每个类簇的最小外包矩形.

步骤 3. 根据最小外包矩形是否重叠对所有类簇进行分组,分组结果为$\{G_1,G_2,\ldots,G_r\}$.令 $i\leftarrow 1$.

步骤 4. 对于分组 G_i 中的每个类簇 C_j,按照定义 17 判断 G_i 中的其他类簇是否是 C_j 的后继关系,保存所有 C_j 的后继关系.

步骤 5. 令 $i\leftarrow i+1$,若 $i\leqslant r$,回到步骤 4.

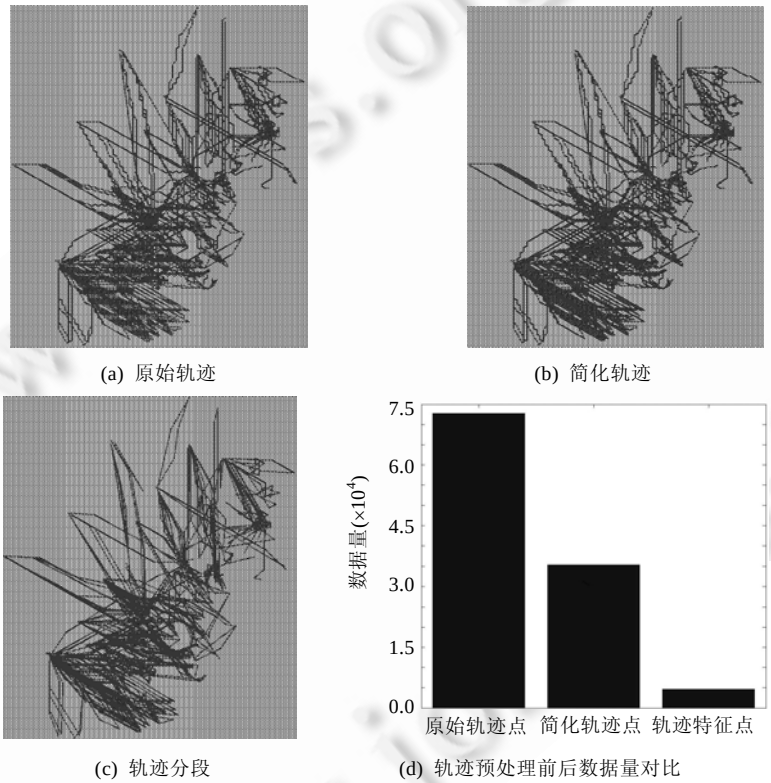
步骤 6. 将所有类簇的后继关系按照定义 17 形成后继关系链.

步骤 7. 根据类簇的代表方向和代表位置将每条后继关系链绘制成带箭头的战场态势图.

3 实验分析

为了验证CTECW算法的有效性,我们采集了某计算机兵棋系统某次推演过程中某一阶段红方的实体轨迹数据作为测试数据,其中包括了 738 个实体,共 72 741 个原始轨迹点数据.所有程序采用 C++编写,并集成在计算机兵棋系统中,在配置为 Pentium 4 2.0GHz,1GB 内存,160G 硬盘的计算机上运行.

图 6 分别从效果显示与数据量两个方面给出了轨迹预处理前后的对比.可以看出,经过预处理的轨迹基本保持了原始轨迹的形状,但数据量却显著减小,因此能够显著提高轨迹聚类工作的效率.适当提高距离阈值 d 可以使得轨迹分段长一些,这样处理的原因在文献[1]已做过讨论.实验中,我们将 d 设为六角格中心点到边的距离的 2 倍.



(a) 原始轨迹 (b) 简化轨迹

(c) 轨迹分段 (d) 轨迹预处理前后数据量对比

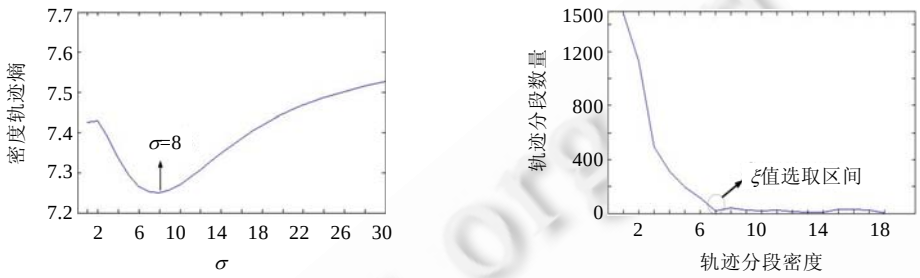
Fig.6 Trajectory pretreatment

图 6 轨迹预处理

图 7 给出的是 CTECW 算法中的两个重要参数σ和ξ的选择依据.其中,图 7(a)显示的是本文采用的测试数据中密度估计熵与窗宽σ的值之间的关系.可以发现,当σ=8 时,密度估计熵值达到极小值,我们选此最优值来设置窗宽σ;图 7(b)显示的是当σ=8 的情况下,轨迹分段密度与轨迹分段数量之间的关系(区间大小设为 1).从图 7(b)可以看出,在密度区间 7~8 附近,曲线出现了一个波谷,这也正是选择合适的ξ值的区间.实验中,我们设定ξ的值为 7.

图 8 显示的是分别采用 TRACLUS 算法和 CTECW 算法对测试数据最优聚类结果的对比.为了使得聚类结果尽量能在同一标准下进行对比,我们对聚类结果采用了同样的处理方式:对每个类簇,根据定义 15 计算其代表方向、代表位置以及类簇中所有轨迹分段的平均长度,可以得到一条代表类簇运动趋势的直线段,图 8 中用黑色粗线绘制的就是这样的直线段.观察图 8 可以看出,在各自最优聚类情况下,两种算法的聚类结果整体上比

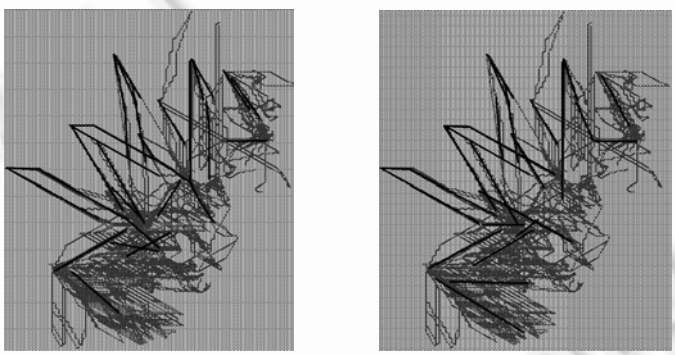
较接近,局部稍有不同.原因可以归结为两点:一是相似性度量不同;二是图 5 中所描述的算法上的区别.由于CTECW 算法采用的相似性度量函数满足度量性质,较 TRACLUS 算法而言可以直接使用高效的空间索引技术来降低算法的复杂度.此外,CTECW 算法在将轨迹处理成分段的过程较 TRACLUS 算法要简单很多.因此,CTECW 算法虽然得到的聚类结果与 TRACLUS 算法比较接近,但执行效率却提高很多.



(a) 优选窗宽σ的值 (b) 优选密度门限值ξ

Fig.7 Parameters choice

图 7 参数选择



(a) TRACLUS 算法聚类效果 (b) CTECW 算法的效果

Fig.8 Contrast of clustering results

图 8 聚类效果对比

图 9 显示的是最终的可视化表现效果,每个箭头代表一条完整的后继关系链.图 9 所示的聚类效果形象直观,容易被受训指挥员接受.若本文的测试数据是从红方角度观察到的,那么红方指挥员很容易就能从图 9 的结果看清楚当前的对抗重点在哪里,所属作战实体是否按照其作战意图如期展开行动;若测试数据是从蓝方角度观察到的,那么图 9 的结果能够辅助蓝方指挥员了解红方当前的作战意图.因此,无论是对哪一方, CTECW 算法在帮助受训指挥员掌握整个战场态势上都能起到辅助参考的作用.

4 结束语

计算机兵棋系统是和平时期作战指挥能力训练的有效途径,其发展越来越受到国内外的重视.本文在充分分析了计算机兵棋实体轨迹数据特点的基础上,提出计算机兵棋实体轨迹聚类算法(CTECW).该算法首先将轨迹处理成轨迹分段;再基于提出的相似性度量函数对轨迹分段进行聚类;最后对聚类结果进行可视化表现,以描绘出整体的战场态势.所提出的相似性度量函数满足度量性质,因此可以直接采用高效的空间索引技术来降低算法的复杂度.理论分析与实验结果表明,CTECW 算法得到的聚类结果与 TRACLUS 算法整体上比较接近,同

样能够发现任意形状和大小的类簇,并且聚类结果不依赖于用户参数的仔细选择,具有良好的鲁棒性.更为重要的是,CTECW 算法面向计算机兵棋系统,具有实际应用价值,使计算机兵棋系统能够充分发挥在指挥能力培养和作战理论探索上的优势.

目前,CTECW 算法中的可视化表现部分仅仅是将每一条后继关系链用一个箭头来表示.实际上我们可以发现,同一个类簇可能对应着多个后继关系链,这会导致多个箭头重叠在一起,画面显得不够简洁美观.因此,设计更加高效实用的可视化表现算法会是下一步工作的重点之一.此外,围绕计算机兵棋实体轨迹数据,挖掘出带有军事意义的运动模式(如集结、夹击、追击、包围等等),也是下一步值得深入研究的问题.

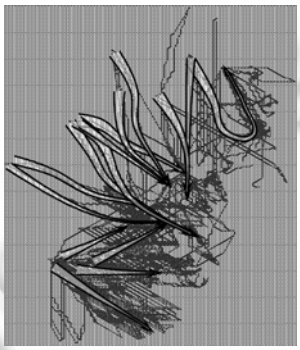


Fig.9 Visual presentation results

图 9 可视化表现效果

致谢 在此,我们向对本文轨迹聚类结果可视化表现部分的工作给予了支持和帮助的信息工程大学测绘学院的武志强教授以及国防大学信息作战与指挥训练教研部周成军讲师表示感谢.

References:

[1] Hu XF, Yang JY, Si GY, Zhang MZ. War Complex System Simulation Analysis and Experimentation. Beijing: Press of National Defence University, 2008 (in Chinese).

[2] Lee JG, Han JW, Whang KY. Trajectory clustering: A partition-and-group framework. In: Proc. of the SIGMOD 2007. 2007. 593−604. [doi: 10.1145/1247480.1247546]

[3] Lee JG, Han JW, Li XL. Trajectory outlier detection: A partition-and-detect framework. In: Proc. of the 24th Int'l Conf. on Data Engineering. 2007. 140−149. [doi: 10.110/ICDE.2008.4497422]

[4] Dodge S, Weibel R, Forootan E. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. Computers, Environment and Urban Systems, 2009,33(6):419−434. [doi: 10.1016/j.compenvurbsys.2009.07.008]

[5] Nguyen HT, Einoshin S. A symbolic representation for trajectory data. In: Proc. of the 24th Annual Conf. of the Japanese Society for Artificial Intelligence. 2010.

[6] Vlachos M, Kollios G, Gunopulos D. Discovering similar multidimensional trajectories. In: Proc. of the Int'l Conf. on Data Engineering. 2002. 673−684. [doi: 10.1109/ICDE.2002.994784]

[7] Yanagisawa Y, Akahani J, Satoch T. Shape-Based similarity query for trajectory of mobile objects. In: Proc. of the 4th Int'l Conf. on MDM. 2003. 63−77.

[8] Zhang Z, Huang K, Tan TN. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In: Proc. of the Int'l Conf. on Pattern Recognition. 2006. 1135−1138. [doi: 10.1109/ICPR.2006.392]

[9] Hwang JR, Kang HY, Li KJ. Searching for similar trajectories on road networks using spatio-temporal similarity. In: Proc. of the 10th East European Conf. on Advances in Databases and Information Systems (ADBIS). 2006. 282−295. [doi: 10.1007/11827252_22]

[10] Gaffney SJ, Robertson AW, Smyth P, Camargo SJ, Ghil M. Probabilistic clustering of extratropical cyclones using regression mixture models. Climate Dynamics, 2007,29(4):423−440. [doi: 10.1007/s00382-007-0235-z]

[11] Nanni M, Pedreschi D. Time-Focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems, 2006, 27(3):267−289. [doi: 10.1007/s10844-006-9953-7]

[12] Pelekis N, Kopanakis I, Kotsifakos E, Frentzos E, Theodoridis Y. Clustering trajectories of moving objects in an uncertain world. In: Proc. of the 9th IEEE Int'l Conf. on Data Mining (ICDM 2009). 2009. 417−427. [doi: 10.1109/ICDM.2009.57]

[13] Chen CS, Eick CF, Rizk NJ. Mining spatial trajectories using nonparametric density functions. In: Proc. of the 7th Int'l Conf. on Machine Learning and Data Mining in Pattern Recognition (MLDM 2011). 2011. 496−510. [doi: 10.1007/978-3-642-23199-5_37]

[14] Morris B, Trivedi M. Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2009). 2009. 312−319. [doi: 10.1109/CVPR.2009.5206559]

[15] Gelbard R, Goldman O, Spiegler I. Investigating diversity of clustering methods: An empirical comparison. Data and Knowledge Engineering, 2007,63(1):155−166. [doi: 10.1016/j.datak.2007.01.002]

[16] Condat L, Van DVD, Blu T. Hexagonal versus orthogonal lattices: A new comparison using approximation theory. In: Proc. of the IEEE Int'l Image Processing. 2005. 1116−1119. [doi: 10.1109/ICIP.2005.1530592]

[17] Ester M, Kriegel HP, Sander J. A density based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining. 1996. 226−231.

[18] Hinneburg A, Keim DA. An efficient approach to clustering in large multimedia databases with noise. In: Proc. of the 4th Int'l Conf. on Knowledge Discovery and Data Mining. 1998. 58−65.

[19] Gan WY, Li DY. Hierarchical clustering based on kernel density estimation. Journal of System Simulation, 2004,16(2):302−305 (in Chinese with English abstract).

附中文参考文献:

[1] 胡晓峰,杨镜宇,司光亚,张明智.战争复杂系统仿真分析与实验.北京:国防大学出版社,2008.

[19] 淦文燕,李德毅.基于核密度估计的层次聚类算法.系统仿真学报,2004,16(2):302−305.

石崇林(1982－),男,江苏如东人,博士生,主要研究领域为数据挖掘,战争模拟.
E-mail: chonglinshi1982@163.com

张茂军(1971－),男,博士,教授,博士生导师,主要研究领域为多媒体信息系统,虚拟现实技术.
E-mail: zmjbar@163.com

淦文燕(1971－),女,博士,副教授,主要研究领域为智能信息处理,复杂系统与复杂网络.
E-mail: wenyangan@163.com

唐宇波(1974－),男,博士,副教授,主要研究领域为作战模拟与仿真,通信网络分析.
E-mail: Tangttt1234@sina.com

吴琳(1974－),男,博士,教授,主要研究领域为作战模拟与仿真.
E-mail: Lieut_wu@hotmail.com