

基于实时战略游戏重放记录数据编码和机器学习的 游戏获胜者预测

王伟清¹, 王全迪^{1,2}, 周杰^{3*}

(1. 华南理工大学 数学学院, 广州 510641; 2. 华南理工大学 继续教育学院, 广州 510641;

3. 广东省计算机网络重点实验室(华南理工大学), 广州 510006)

(* 通信作者电子邮箱 jiezhou@scut.edu.cn)

摘要:对实时战略(RTS)游戏仿真平台 μ RTS 自带的 RTS 游戏 AI 机器人之间进行游戏比赛产生的重放记录数据进行采样, 用独热编码对采样点数据中的游戏玩家在游戏状态和动作信息进行编码, 利用卷积神经网络、支持向量机和 K-近邻等机器学习算法对 RTS 游戏 AI 机器人在游戏比赛中的获胜者进行预测。实验结果表明, 结合给出的编码方法和机器学习算法预测 RTS 游戏获胜者的准确率与已有方法相比有显著提高, 预测结果 ROC 曲线的 AUC 值较高。

关键词:实时策略游戏; 游戏 AI 机器人; 重放记录数据编码; 机器学习; 游戏获胜者预测

中图分类号: TP183; **文献标志码:** A

Winner prediction based on RTS game replay data encoding and machine learning

WANG Weiqing¹, WANG Quandi^{1,2}, ZHOU Jie^{3*}

(1. School of Mathematics, South China University of Technology, Guangzhou Guangdong 510641, China;

2. School of Continuing Education, South China University of Technology, Guangzhou Guangdong 510641, China;

3. Guangdong Key Lab of Communication and Computer Network (South China University of Technology), Guangzhou Guangdong 510006, China)

Abstract: The replay data generated by μ RTS, a Real-Time strategy (RTS) game simulation platform, in a game match between build-in RTS game AI robots were sampled, the players' sampled state and action information were encoded by one-hot encoding method, and the winner of the RTS game AI robots in the game match was predicted by machine learning algorithms such as Convolutional Neural Network (CNN), Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). Combined with the encoding method and machine learning algorithm, the experimental results show that the accuracy of prediction and AUC (Area Under the ROC (Receiver Operating Characteristic) curve) are significantly improved compared with existing RTS game AI robot's winner prediction methods.

Key words: Real Time Strategy (RTS) game; game Artificial Intelligence robot; replay data encoding; machine learning; game winner prediction

0 引言

实时战略(Real-Time Strategy, RTS)游戏已广泛应用于人工智能(Artificial Intelligence, AI)研究的相关领域^[1-2]。最近 Vinyals 等^[3]基于一种包含深层神经网络的多主体强化学习算法开发了一个代理程序 AlphaStar。在没有任何游戏预设条件的情况下, AlphaStar 在 Battle.net 上的《星际争霸 II》(一款流行的 RTS 游戏)活跃玩家中排名超过 99.8%。该研究成果为利用 RTS 游戏研究 AI 技术提供了有益的借鉴。由于极大的状态空间、有限的决策时间和动态环境,为利用 RTS 游戏研究 AI 相关技术(如 AI 规划、AI 搜索中的状态评估和 AI 研究人员建模等)提供了一个理想的环境^[4-7]。其中,实时对抗计划和不确定决策是 AI 规划的重要研究领域^[2], 获胜者预测是 AI 规划的一个主要研究方向。AI, 特别是 AI 中的卷积神经网络

(Convolution Neural Network, CNN)、贝叶斯模型等已被证明可有效预测 RTS 游戏的获胜者^[4,6,8-9]。而基于机器学习预测 RTS 游戏获胜者的相关研究较少^[2]。采用机器学习算法对 RTS 游戏 AI 机器人(以下简称 AI 机器人)在游戏比赛中的获胜者进行预测涉及搜索算法、状态评估和动作抽象等领域,对评估机器学习算法的性能具有实际意义。利用 AI, 特别是 AI 中的机器学习算法研究 RTS 游戏获胜者预测的方法中,对游戏重放记录数据的编码是其中的重要环节,有效的编码方法对提高预测的准确性起到关键的作用。

鉴于 RTS 游戏的特性,有许多基于 RTS 游戏对 AI 相关技术的研究, Ontañón 等^[2]总结了一些已有的研究方法和结果,其中也包含利用 AI 研究 RTS 游戏获胜者预测的一些方法和结果。近年来利用 AI 研究 RTS 游戏获胜者预测的方法有了进一步的发展。Wu 等^[10]利用进化机器学习(Evolutionary

收稿日期: 2020-08-21; 修回日期: 2020-12-14; 录用日期: 2020-12-23。

作者简介: 王伟清(1996—), 女, 山东潍坊人, 硕士研究生, 主要研究方向: 微分方程与动力系统、机器学习; 王全迪(1963—), 女, 吉林集安人, 教授, 博士, 主要研究方向: 微分方程与动力系统; 周杰(1964—), 男, 吉林双辽人, 教授, 博士, 主要研究方向: 组合优化、计算机网络、生物信息学、人工智能。

Machine Learning, EML)研究《星际争霸》游戏,提出一种抽象和简化的模型,并设计了一种动态编程算法以解决基于特定目标的游戏获胜者预测花费时间过多的问题。Stanescu等^[11-12]应用Lanchester损耗定律评估游戏玩家的实力以预测RTS游戏的获胜者,通过对RTS游戏中的单元和状态信息进行抽象,结合《星际争霸》预测游戏的获胜者。Barriga等^[1]通过对RTS中的游戏状态的抽象,基于游戏树搜索算法预测游戏的获胜者。上述研究都是通过对RTS游戏重放记录数据的状态信息进行抽象预测游戏的获胜者,但由于RTS游戏具有极大的状态空间,这种方法很难涵盖RTS游戏的所有状态信息,因而预测的准确率偏低。对RTS游戏的状态信息进行抽象实质上也是一种对游戏重放记录数据的编码方法。Li等^[8]利用《魔兽争霸Ⅲ》真实玩家的游戏数据,通过极限学习机(Extreme Learning Machine, ELM)模型评估RTS游戏策略,以获得不同类型单元的组合效率,通过处理单元产生和交互的顺序来预测游戏的获胜者。Stanescu等^[13]提出了一种基于贝叶斯模型来预测两个游戏玩家获胜者的方法,该研究主要通过通过对游戏玩家获胜者预测来指导游戏玩家建立单位以赢得《星际争霸》游戏。Ravari等^[7]开发了一个通用模型来预测《星际争霸》中游戏玩家的获胜者,该研究的目的主要是找出对预测结果产生影响的特征。Lin等^[9]提出神经过程方法,结合神经网络和概率推断,用于对《星际争霸Ⅱ》中游戏获胜者的预测。上述方法利用对游戏重放记录数据进行编码来预测游戏的获胜者,但这些方法都仅对游戏重放记录数据的状态信息进行编码,没有对其中的动作信息进行编码,对游戏重放记录数据信息的利用程度不足,预测的准确率偏低。另外,上面所有研究主要是针对《星际争霸》或《魔兽争霸》这两款特定的RTS游戏,具有一定的局限性。 μ RTS^[6]是一种基于代理和AI的开源RTS游戏仿真平台,为利用RTS游戏研究AI、验证新算法提供了有用的工具。Huang等^[4]使用多尺度卷积神经网络(Multi-Size Convolution Neural Network, MSCNN),基于 μ RTS仿真平台中AI机器人之间游戏比赛产生的游戏重放记录数据预测两个AI机器人在游戏比赛中获胜者的概率。Yang等^[14]利用 μ RTS仿真平台的游戏地图信息、AI机器人的特征和CNN预测RTS游戏的获胜者。Yang等^[6]将Huang等^[4]的方法扩展到RTS游戏的时间序列数据集上预测游戏获胜者。上面研究虽然针对的是一般的RTS游戏,而且也是利用对游戏重放记录数据进行编码来预测游戏的获胜者,但同样都只对RTS游戏重放记录数据的状态信息进行编码,使得预测的准确率偏低。通过对相关数据库和搜索引擎的检索,目前没有利用支持向量机(Support Vector Machine, SVM)和K-近邻(K-Nearest Neighbor, KNN)等机器学习算法预测游戏(特别是RTS游戏)获胜者的相关研究。

由于基于具体一款RTS游戏对AI相关技术(特别是游戏获胜者预测)的研究具有局限性,而且通过对RTS游戏状态信息进行抽象预测获胜者的研究中使用的游戏信息具有不完整性和同时为了弥补利用机器学习算法预测游戏获胜者相关研究的不足,鉴于游戏重放记录数据能够完全反映游戏的状态和动作信息,本文选择RTS游戏仿真平台 μ RTS自带的AI机

器人之间进行游戏比赛产生的重放记录数据,对RTS游戏重放记录数据进行采样,通过对采样点数据的状态和动作信息以及游戏比赛的获胜者信息统一编码,利用CNN、SVM和KNN等机器学习算法对AI机器人在游戏比赛中的获胜者进行预测,以此验证所设计方法的有效性,进而说明机器学习算法也可有效应用在对实时策略游戏的研究中,为研究机器学习算法提供了一个新的途径。

1 本文方法

RTS游戏仿真平台 μ RTS包含了近50个基于不同搜索策略的AI机器人,这些AI机器人之间的游戏比赛策略具有RTS游戏的基本特征,游戏比赛的重放记录数据完整且涵盖了游戏过程中所有的状态和动作信息。为此,本文利用 μ RTS仿真平台中具有不同策略的10个AI机器人在特定地图内进行游戏比赛生成重放记录数据集,对每条重放记录数据进行采样。

本文首先介绍了利用 μ RTS仿真平台的AI机器人生成重放记录数据集以及对重放记录数据采样的方法,详细描述了采样点数据的数据结构。在预测游戏获胜者时需要将采样点数据中的信息进行编码,但由于对采样点数据中的动作元素进行编码比较困难,所以本文设计了一种新的编码方法,不仅对采样点数据中的状态信息进行编码,也对动作信息进行编码。由于RTS游戏的动作信息中每个操作都包含主动和被动单位,在对采样点数据中的动作信息编码时,添加被动单位动作属性,其属性值表示单位的被动动作。利用设计的编码方法,将采样点数据中的状态和动作信息中的每个属性根据其属性值编码为一个 $8 \times 8 \times k$ 的3维数组。

接着详细描述了对采样点数据中的状态和动作信息以及游戏比赛的获胜者信息的编码方法和具体实现。鉴于目前没有利用SVM和KNN等机器学习算法预测游戏(特别是RTS游戏)获胜者的相关研究,本文通过对采样点数据中的状态和动作信息、重放记录数据中的获胜者信息统一编码后产生的数据集,利用CNN、SVM和KNN等机器学习算法对AI机器人在游戏比赛中的获胜者进行预测。最后在实验中计算了不同机器学习算法预测RTS游戏获胜的准确率、ROC曲线和AUC值。

2 数据的获取、处理和描述

鉴于 μ RTS仿真平台自带的AI机器人之间的游戏比赛具有RTS游戏的基本特征,本文选用 μ RTS仿真平台自带的RTS游戏AI机器人之间进行游戏比赛产生的重放记录数据,通过对重放记录数据进行采样生成游戏比赛记录数据集来预测游戏的获胜者。为了说明本文设计的对重放记录数据的状态和动作信息的编码方法,下面对游戏比赛记录数据进行详细的描述。

在 μ RTS包含的近50个基于不同搜索策略的AI机器人中,有近20个使用蒙特卡罗树搜索(Monte Carlo Tree Search, MCTS)算法,其余的使用其他搜索策略(如硬编码策略、alpha-beta搜索和minimax树搜索等)。 μ RTS仿真平台支持多种尺

寸的地图,范围从 8×8 到 24×24 。每种尺寸的地图包含许多初始模板,每个模板是一个包含游戏状态元素的可扩展标记语言(Extensible Markup Language, XML)文件。本文在 μ RTS 仿真平台中选择 10 个使用不同策略的 AI 机器人:5 个不使用 MCTS 算法的 AI 机器人和 5 个使用 MCTS 算法的 AI 机器人(表 1)。10 个 AI 机器人在 μ RTS 仿真平台上以 4 种不同模式(表 2),在 24 个不同初始模板的 8×8 地图中进行循环比赛,生成重放记录数据集,其中包含 9 600 ($10 \times 10 \times 4 \times 24$) 条重放记录数据。每对 AI 机器人的每场游戏比赛生成的重放记录数据是

表 1 AI 机器人和策略描述

AI 机器人	策略描述
WorkerRushPlusPlus	使用硬编码策略,训练工人、建立基础,收获资源、将多余的工人派遣到最近的敌军基地和营房进行进攻
WorkerRush	使用硬编码策略,不断产生工人并将其派遣送到攻击中
CRush_V1	建造兵营,跟踪构建命令,制造军事单位并控制其攻击最近的目标
AHTNAI	对抗性分层任务网络,将 minimax 游戏树搜索与 HTN 规划相结合
POWorkerRush	使用硬编码的策略,不断产生工人攻击最近的目标;如果没有敌人,则移动到最近的位置;只需要一个工人来开采资源
MonteCarlo	标准的蒙特卡洛树搜索算法;对每个合法参与者的操作,都会运行尽可能多的模拟以估计其预期收益
NaiveMCTS	具有采样策略的蒙特卡洛树搜索算法,专门为具有组合分支因子的游戏设计,组合多个武器的特殊树形结构
MLPSMCTS	使用 MLPS(匹配学习与多项式存储)采样策略的蒙特卡洛算法
LSI	带有线性边信息的蒙特卡洛搜索算法
PuppetSearchMCTS	基于脚本对抗性搜索的蒙特卡洛算法,它可以将选择点暴露给预先搜索程序,为选择点选择脚本和决策的组合表示接下来要应用的动作

表 2 循环赛中使用的四种模式

模式	时间预算	迭代次数 预算	模式	时间预算	迭代次数 预算
模式 1	100	100	模式 3	100	200
模式 2	200	100	模式 4	200	200

本文在每条重放记录数据中随机选取 3 个采样时间点。每个采样时间点包含 1 个轨迹条目。将每个采样时间点的轨迹条目和该场比赛的获胜者元素一起组成一个 XML 文件。对重放记录数据集中的每条重放记录数据同样处理,结果构成游戏比赛预测记录数据集,其中包含 27 856 个 XML 文件。图 1 是一个采用点的轨迹条目的数据结构图;图 2 是一个轨迹条目在 μ RTS 仿真平台中的示意图。

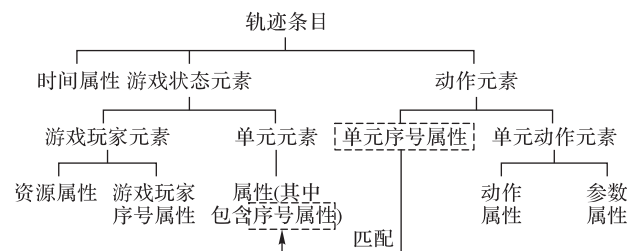
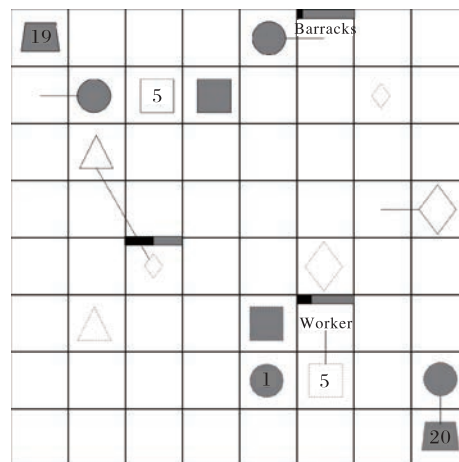


图 1 轨迹条目数据结构

每个轨迹条目包含一个时间属性,表示该轨迹条目的采样时间点。游戏状态元素包含游戏玩家元素和单元元素。游戏玩家元素包含游戏玩家序号属性和资源属性,游戏玩家序号属性值分别为 0 和 1,代表比赛中的两个 AI 机器人(在图 2 中分别以实和虚的轮廓线表示);资源属性值表示相应 AI 机器人在采样时间点的基地单位的初始资源。单元元素包含一系列子元素,每个子元素都包含序号、单元类型、资源、抗打击度(也称血量)、 x 、 y 和游戏玩家等属性。其中,序号属性的值是该单元元素的标识;单元类型属性的值分别是资源(图 2 中由灰色梯形表示)、基地(图 2 中由白色正方形表示)、营房(图

一个 XML 文件,由一系列记录游戏状态和动作信息的轨迹条目以及记录该场比赛获胜者的获胜者元素组成。每个轨迹条目主要包含游戏状态元素和动作元素。获胜者元素的属性值为 1 表示胜方,为 0 表示负方;获胜者元素的属性值为-1 时,表示比赛是平局或未在最大游戏周期内结束。本文设置每场比赛的最大游戏周期为 3 000。选定 AI 机器人游戏比赛的重放记录数据的 XML 文件的一些轨迹条目和相应的获胜者元素,使用机器学习算法预测游戏的获胜者,旨在提高 AI 游戏预测的性能,也是与 AI 计划、AI 搜索中的状态评估和人类行为建模相关的重要研究课题。

2 中由灰色正方形表示)、工兵(图 2 中由灰色圆形表示)、轻型工兵(图 2 中由小的菱形表示)、远程工兵(图 2 中由三角形表示)和重型工兵(图 2 中由大的菱形表示),它们均为游戏中的单位(表 3)。图 2 中代表各属性值的图形中的数字表示相应的 AI 机器人在该单位中剩余的资源数量,由资源属性值表示;抗打击度属性值表示相应单元类型属性值对应的单位的抗打击度(血量)(在图 2 中由黑色长条表示);属性 x 和 y 的值表示相应单元类型属性值对应的单位在地图中的位置;游戏玩家属性值代表单元类型属性中属性值所属的 AI 机器人,特别地,当游戏玩家属性的值为-1 时表示相应单元类型属性值对应的单位是公共单位。

图 2 一个轨迹条目在 μ RTS 仿真平台中的示意图

轨迹条目中的动作元素包含一系列动作子元素。每个动作子元素包含一个单元序号属性和一个单元动作元素。单元序号的属性值与单元元素中的序号属性值匹配,即每个动作元素在游戏状态元素中都有一个单元元素与之对应。单元动

作元素中的动作类型属性和参数属性指明单元元素的属性值对应的单位(主动执行单位和被动执行单位)在游戏比赛过程中执行的动作。单元动作元素中包含的动作类型属性值为0、1、2、3、4和5,分别表示“等待”“移动”“收获”“返回”“生产”和“攻击”等动作(表3)。根据动作类型属性值的不同,单元动作元素可分为4种情况:1)当动作类型属性值为0、1、2、3和4时,单元动作元素还包含参数属性,其值为0、1、2、3和10;其中参数属性值为0、1、2、3分别表示上、右、下、左,表示被执行

单位动作的方向(从主动到被动,在图2中由黑色线条表示)。2)当动作类型属性值为0,单元动作元素的参数属性值为10时表示被执行单位等待10个周期。3)当动作类型属性值为4时,单元动作元素还包含产生单元类型属性,其值表示被执行单位在相应方向上产生的单位。4)当动作类型属性值为5时,单元动作元素仅包含 x 和 y 属性,其值表示被执行单位攻击地图的 (x,y) 方格内的单位。下一节将根据如上对采样时间点的轨迹条目中的属性值的描述设计的编码方法。

表3 轨迹条目中的元素、属性、对应的平面数和属性值

游戏比赛采样时间点的状态和动作	属性	平面数	属性值
游戏状态元素	单元类型属性	7	资源, 基地, 营房, 工兵, 轻型工兵, 远程工兵, 重型工兵
	游戏玩家序号属性	2	0, 1
	资源属性	7	1, 2, 3, 4, 5, 6-9, 或 ≥ 10
	抗打击度属性	5	1, 2, 3, 4, 或 ≥ 5
动作元素	动作类型属性 [*]	6	0, 1, 2, 3, 4, 5
	被动单元动作	6	被等待, 被移动, 被收获, 被返还, 被生产, 被攻击
	产生的单元类型属性	6	基地, 营房, 工兵, 轻型工兵, 远程工兵, 重型工兵

注:表示动作类型属性的值为0~5分别代表等待、移动、收获、返还、生产和攻击。

3 属性值的编码

对采样时间点的轨迹条目中的属性值和获胜者元素的属性值进行编码是利用游戏回放记录数据和机器学习算法预测游戏获胜者的重要环节。由于已有的RTS游戏比赛获胜者预测方法仅对游戏状态元素中的属性值进行编码,对游戏回放记录数据信息的利用程度不足,本文设计了一种新的编码方法,使用独热编码不仅对轨迹条目中的游戏状态元素的属性值进行编码,而且也对动作元素的属性值进行编码。在第4章中用实验验证了所设计编码方法的有效性。

对单元元素中的单元类型、资源、抗打击度和游戏玩家等属性,根据其属性值分别编码为一个 $8 \times 8 \times k$ 的3维数组(图3),其中的每个 8×8 的2维数组称为一个平面, k 为表3中相应属性对应的平面数。具体地,对单元类型、资源、抗打击度和游戏玩家等属性中的每个不同属性值,根据其所在单元元素中的 x 和 y 的属性值,在相应平面的 (x,y) 位置为1,其他平面的 (x,y) 位置为0;当游戏玩家属性值为-1时,图3中相应平面的 (x,y) 位置均为0;当资源属性值为0或抗打击度属性值为0时,图3中相应平面的 (x,y) 位置均为0。根据图2,在编码时,若单位类型为基地单位时,其中的资源属性值由相应游戏玩家的资源属性值(单位的初始资源属性值)替换。

由于动作元素中的每个操作都包含主动和被动单位,在编码动作元素中的属性时,去除其中的参数属性,添加被动单元动作属性(表3),表示被动动作。

对动作元素下的属性值的编码方法具体描述如下:首先,找到与动作元素中的单元序号属性值匹配的序号属性值,继而找到它所在的游戏状态元素的单元元素;然后,根据单元元素中 x 和 y 的属性值,对单元动作元素中的动作类型属性进行编码。对于单元动作元素的4种不同情况的编码方法如下:1)当动作类型属性值为0、1、2、3和4时,根据参数属性值,对被动单元动作属性进行编码;2)当动作类型属性值为0,单元动作元素的参数属性值为10时,根据参数属性值,对被动单元动作属性进行编码;3)当动作类型属性值为4时,根据单元元素的 x 和 y 的属性值以及参数属性值,对产生的单位类型

属性编码;4)当动作类型属性值为5时,则根据单元动作元素中 x 和 y 的属性值,对被动单元动作属性进行编码。

对动作类型、被动单元动作和产生的单位类型等属性,根据其属性值分别编码为一个 $8 \times 8 \times k$ 的3维数组(图3)。对动作类型属性,根据其所在的动作元素中的单元序号属性值对应的单元元素中的 x 和 y 属性值,在相应平面的 (x,y) 位置为1,其他平面的 (x,y) 位置为0;对被动单元动作属性,根据其所在的动作元素中的动作类型属性值、参数属性值和单元序号属性值对应的单元元素中 x 和 y 属性值在相应平面的 $(x \pm 1, y \pm 1)$ 位置为1,其他平面的 $(x \pm 1, y \pm 1)$ 位置为0,其中的 ± 1 由参数属性值确定;对产生的单位类型属性,根据其所在的动作元素中的动作类型属性值、 x 和 y 的属性值在相应平面的 (x,y) 位置为1,其他平面的 (x,y) 位置为0(表3和图3)。最后,对于每个获胜者元素,若属性值为0,则编码为10,属性值为1,则编码为01。

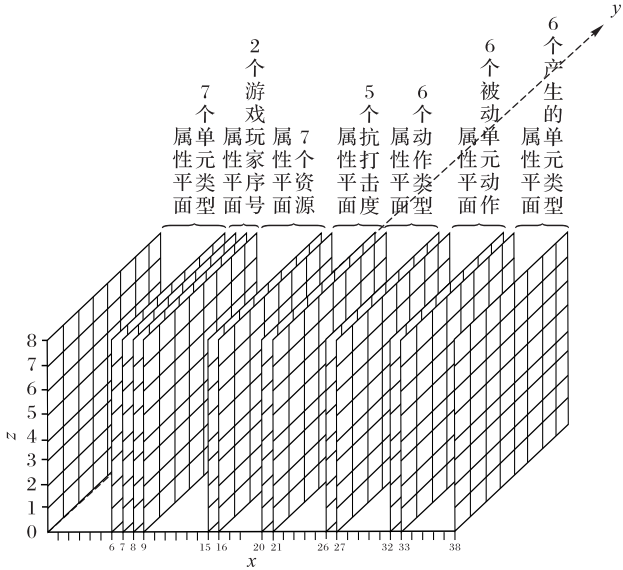


图3 轨迹条目中属性值独热编码的属性平面

通过上面的编码方法,将每个轨迹条目中的游戏状态元素和动作元素中的属性统一编码为一个 $8 \times 8 \times 39$ 的三维数组 A

(表 3 和图 3)。对获胜者元素中的属性编码为 2 个元素的一维数组。对于每场比赛的预测记录数据集的每个 XML 文件编码后得到的数据集称为编码数据集,用于利用机器学习算法预测 AI 机器人游戏比赛获胜者的输入数据集。

4 实验方法和结果

如引言中所述,目前还没有利用 SVM 和 KNN 等机器学习算法预测 RTS 游戏获胜者的相关研究,因而,本文选择 SVM 和 KNN 作为 RTS 游戏获胜者的预测算法;另外,为了与已有研究结果对比,还选择 CNN 算法预测 RTS 游戏获胜者。利用给出的编码方法得到的编码数据训练、验证和测试机器学习算法 CNN、SVM 和 KNN 算法,以此验证所提方法的预测性能。实验结果说明机器学习方法也可有效地应用在实时策略游戏中,为研究机器学习算法提供了一个新的途径。所用的机器学习算法通过 Python 的 Tensorflow 等包实现。在实验过程中,将编码数据集按照 2:1:1 的比例分为训练集、验证集和测试集。

为了评估所提出方法在预测 AI 机器人获胜者上的预测性能,实验中计算了 3 个机器学习算法对 AI 机器人获胜者预测结果的准确率(表 4)、ROC 曲线和相应的 AUC 值(图 4)。实验结果表明,使用本文的方法,3 个机器学习算法对 AI 机器人在游戏比赛中获胜者预测的准确率分别为 0.82 和 0.74(表 4)。而已有的方法预测的准确率都约为 0.6(参看参考文献[4]中的表 2、文献[11]中的表 3 和文献[7]中的表 3)。这说明利用机器学习算法预测 AI 机器人在游戏比赛中的获胜者,对游戏比赛产生的重放记录数据的编码时增加对游戏玩家在游戏中的动作信息的编码,能够明显提高预测的准确率。

表 4 三个机器学习算法对 AI 机器人获胜者预测的准确率

算法	准确率
CNN	0.821 250
SVM	0.825 313
KNN	0.743 958

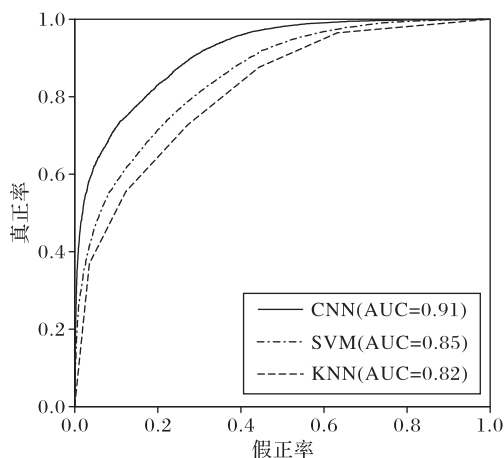


图 4 三个机器学习算法对 AI 机器人获胜者预测的 ROC 曲线和 AUC 值

ROC 曲线是反映敏感性和特异性的综合指标,曲线上的每个点代表对应于特定决策值的敏感度和特异性,可准确地反映所用分析方法特异性与敏感性的关系。ROC 曲线越靠近

左上角,测试效率越高。由于预测 AI 机器人在游戏比赛中的获胜者相当于多分类问题,所以本文采用微平均度量方法计算 ROC 曲线。AUC 值反映模型的评价预测精度,在用机器学习算法预测获胜者的实验中,AUC 值表示正确区分获胜者的能力。AUC 值越高,则模型可以更好地预测比赛的获胜者。由图 4 可以看出,在使用 3 个机器学习算法和游戏比赛重放记录数据集进行获胜者预测的实验中,所有 ROC 曲线均靠近左上角,并且相应的 AUC 值达到 0.82 以上。这表明本文方法具有可靠的性能和良好的预测能力。

5 结语

本文研究的目的是通过在 RTS 游戏的仿真平台 μ RTS 中选择 10 个具有不同搜索方法和策略的 AI 机器人之间进行循环比赛产生的重放记录数据(包括游戏玩家在游戏状态和动作信息),通过独热编码,利用 CNN、SVM 和 KNN 等机器学习算法对 RTS 游戏 AI 机器人在游戏比赛中的获胜者进行预测。通过对 RTS 游戏 AI 机器人之间游戏比赛产生的重放记录数据进行采样,分析采样点数据的动作元素中的属性组成特性,去除单元动作元素中的参数属性,添加被动单元动作属性。对采样时间点的轨迹条目中的属性值和获胜者元素的属性值性统一编码,不仅对每个采样点的轨迹条目中的游戏状态元素的属性进行编码,而且还对动作元素的属性进行编码,以此获取较全面的游戏信息。实验结果表明,本文给出的方法具有较高的预测准确率、可靠的预测能力,能为利用 RTS 游戏研究机器学习算法提供一种可行的方法。

参考文献 (References)

- [1] BARRIGA NA, STANESCU M, BESOAIN F, et al. Improving RTS game AI by supervised policy learning, tactical search, and deep reinforcement learning [J]. IEEE Computational Intelligence Magazine, 2019, 14(3): 8-18.
- [2] ONTAÑÓN S, SYNNAEVE G, URIARTE A, et al. A survey of real-time strategy game AI research and competition in StarCraft [J]. IEEE Transactions on Computational Intelligence and AI in Games, 2013, 5(4): 293-311.
- [3] VINYALS O, BABUSCHKIN I, CZARNECKI W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning[J]. Nature, 2019, 575(7782): 350-355.
- [4] HUANG J, YANG W. A multi-size convolution neural network for RTS games winner prediction [C/OL]// EITCE 2018: Proceedings of the 2018 Second International Conference on Electronic Information Technology and Computer Engineering [2020-05-06]. <https://doi.org/10.1051/mateconf/201823201054>.
- [5] STANESCU M, BARRIGA NA, HESS A, et al. Evaluating real-time strategy game states using convolutional neural networks [C]// Proceedings of the 2016 IEEE Computational Intelligence and Games. Piscataway: IEEE, 2016: 1-7.
- [6] YANG W, HUANG J, HU Y. A modified multi-size convolution neural network for winner prediction based on time serial datasets [C]// ICAI 2019: Proceedings of the 2019 Fourth International Conference on Mathematics and Artificial Intelligence. New York: ACM, 2019: 110-114.

- [7] RAVARI Y N, BAKKES S, SPRONCK P. StarCraft winner prediction, artificial intelligence in adversarial games [C]// AAAI 2016: Proceedings of the Fourth Workshop on Artificial Intelligence in Adversarial Games. Palo Alto: AAAI Press, 2016: 16-21.
- [8] LI Y, LI Y, ZHAI J, et al. RTS game strategy evaluation using extreme learning machine [J]. *Soft Computing*, 2012, 16: 1627-1637.
- [9] LIN M, WANG T, LI X B, et al. An uncertainty-incorporated approach to predict the winner in StarCraft II using neural processes [J]. *IEEE Access*, 2019, 7: 101609-101619.
- [10] WU L, MARKHAM A. Evolutionary machine learning for RTS game StarCraft [C]// AAAI 2017: Proceedings of the the 31st AAAI Conference on Artificial intelligence. Palo Alto: AAAI Press, 2017: 5007-5008.
- [11] STANESCU M, BARRIGA N, BURO M. Using Lanchester attrition laws for combat prediction in StarCraft [C]// AIIDE 2015: Proceedings of the Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference. Palo Alto: AAAI Press, 2015: 86-92.
- [12] STANESCU M, BARRIGA N A. Combat outcome prediction for RTS games [M]// *Game AI Pro3: Collected Wisdom of Game AI Professionals*. Steve Rabin: CRC Press, 2017: 1-8.
- [13] STANESCU M, HERNANDEZ S P, ERICKSON G, et al. Predicting army combat outcomes in StarCraft [C]// AIIDE 2013: Proceedings of the Ninth Artificial Intelligence and Interactive Digital Entertainment Conference. Palo Alto: AAAI Press, 2013: 86-92.
- [14] YANG Z, ONTAÑÓN S. Learning map-independent evaluation functions for real-time strategy games [C]// CIG 2018: Proceedings of the 2018 Conference on Computational Intelligence and Games. Piscataway: IEEE, 2018: 301-307.
- [15] Santionanon. microRTS [EB/OL]. [2020-08-06]. <https://github.com/santionanon/microrts>.
- [16] BURO M. Call for AI research in RTS games [C]// AAAI 2004: Proceedings of the 2004 Workshop on Challenges in Game AI. Palo Alto: AAAI Press, 2004: 139-142.
- [17] URIARTE A, ONTAÑÓN S. Game-tree search over high-level game states in RTS games [C]// AIIDE 2014: Proceedings of the Tenth Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence and Interactive Digital Entertainment. Palo Alto: AAAI Press, 2014: 73-79.
- [18] SILVER D, HUANG A, MADDISON C J, et al. Mastering the game of Go with deep neural networks and tree search [J]. *Nature*, 2016, 529(7587): 484-489.
- [19] SILVER D, SCHRITTWIESER J, SIMONYAN K, et al. Mastering the game of Go without human knowledge [J]. *Nature*, 2017, 550(7676): 354-359.