

文章编号:1671-6833(2023)05-0053-09

基于有向图的强化学习自动驾驶轨迹预测

崔建明¹, 蔺繁荣¹, 张迪¹, 张路宁¹, 刘铭²

(1. 长安大学 信息工程学院, 陕西 西安 710018; 2. 国家计算机网络应急技术处理协调中心, 北京 100029)

摘要: 轨迹预测作为自动驾驶中的重要组成部分,旨在对车辆进行行驶估计,以便车辆根据行驶估计进行路径规划,从而做出安全准确的决策。首先,为提升车辆轨迹预测精度,采用有向图方法构建高清驾驶场景地图,有向图方法将地图信息矢量化,以便有效提取地图拓扑结构;其次,采用生成对抗模仿学习(GAIL)通过生成器与判别器的对抗博弈学习数据集驾驶策略,从而根据当前状态采取对应驾驶行为;最后,通过采样遍历得到多模态预测轨迹方案。在 nuScenes 运动预测数据集上进行仿真,量化结果显示相比于其他方法, $K=5$ 时,最小最终位移误差 $MinFDE_5$ 提高了 10.8%; $K=10$ 时,最小最终位移误差 $MinFDE_{10}$ 提高了 17.53%,最小平均位移误差 $MinADE_{10}$ 提高了 9.52%,失误差率 $MissRate_{10}$ 减少了 28.26%。评估结果表明:生成的轨迹多模态符合场景基本结构,且准确度得到提高。

关键词: 自动驾驶; 轨迹预测; 有向图; 强化学习; GAIL; 注意力机制; 多模态预测

中图分类号: O211.62; TP183

文献标志码: A

doi: 10.13705/j.issn.1671-6833.2023.05.002

随着深度学习和计算机视觉技术的兴起,自动驾驶为解决交通出行安全、提高通行效率提供了新的解决方案。2020年,中国11个部委联合发布的《智能汽车创新发展战略》明确指出,自动驾驶汽车已经成为全球汽车产业发展的战略方向^[1]。技术进步、市场吸引、政策推动使自动驾驶成为未来智能交通发展的重要内容。

在自动驾驶中,轨迹预测能根据交通场景复杂的地图信息和周围车辆运动趋势为汽车选择更优的行车路线,提高通行效率。但是,随着我国机动化进程加快,交通事故发生率逐年上升,为更好地保障行车安全,对轨迹预测精度提出了更高的要求。高清驾驶场景地图(high-definition driving scene map, HD map)为提高车辆行驶轨迹预测精度提供更为全面、精准的车辆状态信息及道路结构信息,因此,HD map成为目前轨迹预测模型和公共自动驾驶数据集依赖的重要基础。

过去的工作使用机器学习将地图元素(车道、人行道等)栅格化为图层,并用不同的颜色对车道方向、红绿灯和限速信息进行编码^[2]。在空间网络

中逐层对静态实体、动态实体和语义地图信息进行编码。然后将这些编码用卷积神经网络(CNN)进行处理,学习到的地图特征为轨迹预测提供了有用的上下文信息。但是这种方法存在很多问题,如计算效率低下、容易丢失地图信息、很难捕获地图的复杂拓扑结构等。

此外,模拟人类驾驶行为对预测至关重要。模仿学习在自动驾驶中得到广泛应用,模仿学习需要采集一些示例数据,从状态空间 S 、动作空间 A 中提取状态动作对 (s, a) ,每个状态 s 对应其采取的动作 a ,将数据集状态动作对作为训练数据训练策略网络,从而能够让策略网络模仿专家策略。行为克隆是最简单的模仿学习,它将模仿学习视为有监督的学习问题。这种方法模拟学习自动驾驶策略,可以应用到停车场、高速公路^[2-3]。但这种方法会产生分布偏移,导致在序列决策问题中,一旦遇到没有在示例轨迹中出现的情况,就会产生很大的误差以及累计误差,从而使预测轨迹逐渐偏离正确轨迹。

本文提出一种基于有向图和生成对抗模仿学习(generative adversarial imitation learning, GAIL)^[4]结

收稿日期:2023-02-10;修订日期:2023-04-06

基金项目:国家自然科学基金资助项目(62106060)

作者简介:崔建明(1974—),男,河南洛阳人,长安大学副教授,博士,主要从事人工智能与车联网技术研究, E-mail: cjianming@chd.edu.cn。

通信作者:刘铭(1986—),男,河南郑州人,国家计算机网络应急技术处理协调中心高级工程师,博士,主要从事人工智能与网络安全技术研究, E-mail: liuming@cert.org.cn。

引用本文:崔建明,蔺繁荣,张迪,等.基于有向图的强化学习自动驾驶轨迹预测[J].郑州大学学报(工学版),2023,44(5):53-61.(CUI J M, LIN F R, ZHANG D, et al. Reinforcement learning autonomous driving trajectory prediction based on directed graph)[J]. Journal of Zhengzhou University (Engineering Science), 2023, 44(5): 53-61.)

合的轨迹预测方法。首先,将车道构建为有向图;其次,用 GAIL 学习驾驶策略,将驾驶策略与轨迹生成器相结合输出预测轨迹;最后,在数据集 nuScenes^[5]上进行仿真,验证通过驾驶策略得到的预测轨迹结果是否可靠。

1 轨迹预测相关工作

最近的轨迹预测方法使用矢量化的地图要素,将地图编码为有向图。高清地图提供的车道信息、道路设施等静态信息中大多数都是样条线(如车道)、闭合形状(如十字路口区域)和点(如红绿灯)的形式,这些信息便于矢量化。矢量化地图相比于栅格地图能更好地捕获地图的复杂拓扑结构,并且用更少的参数就能实现更好的效果。车道图卷积网络(LaneGCN)^[6]将车道与车辆都视为图的节点,使用一维卷积和 LaneGCN 分别提取车辆和车道节点的特征,然后利用图卷积网络(GCN)^[7]和空间注意力机制更新车道和车辆节点的特征。本文使用有向图构建稀疏连接的车道图,由近及远分步融合需要关注的周围车辆和地图信息来更新节点,使用缩放点积注意力将重点关注的车辆信息更新到节点,分别使用 GCN 和图注意力网络(GAT)^[8]融合周围节点信息。

在序列决策问题中行为克隆会导致分布偏移,不会获得较好的效果。为了解决行为克隆分布偏移的问题,采用 GAIL 学习数据集的驾驶策略。

早期轨迹预测采用动力学^[9]模型输出单一的未來轨迹。由于交通参与者的未来轨迹是不确定的,通常有多种合理的轨迹,因此最近的轨迹预测方法开始使用深层生成模型来模拟轨迹分布。为了生成更多的轨迹,使用采样遍历生成多模态轨迹,并通过添加潜变量模拟速度等连续变化调节轨迹路线。

2 基于强化学习的自动驾驶轨迹预测模型构建

2.1 模仿学习

Caesar 等^[5]提出了 GAIL 方法并在许多连续控制任务中都取得了很好的表现。使用 GAIL 可以获得近似示例数据的分布,从而使其对扰动更加鲁棒。

本文使用 GAIL 模拟示例数据分布,其中策略网络对应于 GANs 中的生成器,用于根据状态来得到动作,判别网络用于判别一个状态动作对来自示例数据还是策略网络。2 个网络相互对抗,在示例样本较多的情况下,该方法可以最大限度地抽取示例数据中的特征,学习到示例数据中的内在分布,得

到效果非常好的策略网络和判别网络。

2.2 多模态轨迹预测

目标车辆在有向车道图中可以有任意条路径,并且每一条路径都可能是它的合理路径,但是采用这些路径的可能性各不相同。例如,接近路口缓慢行驶且车头偏离车道中心线的车辆,更可能改变车道。

为了得到多模态轨迹,本文首先通过循环输入当前状态得到预测轨迹,再采样得到预测轨迹对应的节点序列,然后使用多头注意力机制在节点序列上聚合地图和周围车辆信息,最后输出一个上下文向量 C 来编码采样路径。使用门控循环单元(GRU)编码目标车辆的轨迹,输出运动编码。除了模拟轨迹路线的变化,还使用抽样的潜在变量 z 模拟车辆的速度、加速度等连续变化。将得到的上下文向量、运动编码和抽样的潜变量通过 MLP 输出预测轨迹,再使用 K -means 聚类输出最有可能的 K 个轨迹集。

2.3 输入输出表示

假设可以从车载传感器获得交通场景中所有交通参与者的过去轨迹,在所研究轨迹地图的鸟瞰视图中矢量化描绘场景并预测轨迹,使用以目标车辆为中心的参考系,该参考系 y 轴与目标车辆的瞬时运动方向对齐,横向为预测 $[-50\text{ m}, 50\text{ m}]$,纵向 $[-20\text{ m}, 80\text{ m}]$ 。将车辆的过去轨迹和未来轨迹连接成一条关于时间的样条线,近似为向量序列,假设一个场景中有 N 个交通参与者,第 i 个交通参与者的过去的时间步长 t_h 的轨迹可以表示为一个向量序列 $\alpha_{-t_h:0}^i = [\alpha_{-t_h}^i, \alpha_{-t_h-1}^i, \dots, \alpha_0^i]$ 。其中, $-t_h$ 指过去轨迹开始时间 $-t_h$ 到预测开始时间 0。每个 $\alpha_t^i = [x_t^i, y_t^i, v_t^i, a_t^i, \delta_t^i, l_i]$, x_t^i, y_t^i 为时刻 t 时交通参与者 i 在鸟瞰图中的位置坐标; v_t^i, a_t^i 和 δ_t^i 分别为此时的速度、加速度和偏航角度; l_i 代表交通参与者 i 的类别。

为了显示未来轨迹的多模态,输出车辆预测时间步长 t_f 内的 K 个轨迹 $\beta_{1:t_f}, \beta_{1:t_f} = [\beta_{1:t_f}^1, \beta_{1:t_f}^2, \dots, \beta_{1:t_f}^K]$ 。每个 $\beta_{1:t_f}^K$ 都代表了预测的一种模式,理想情况下对应于不同的合理路线或同一路线上的不同运动轨迹。

2.4 模型构建

图 1 为轨迹预测模型结构图,它由 3 个模块组成,分别为车道有向图、策略网络和轨迹生成器。其中,车道有向图结合高清地图和车辆轨迹,输出学习到的车道图的每个节点表示,形成了模型的主干;策略网络通过生成器与判别器的对抗生成符合车辆运

动的策略;轨迹生成器用来输出基于策略和采样潜在变量所遍历路径的轨迹。

图 2 为模型网络结构图,网络结构图由 3 个模块组成。车道有向图分别将节点、周围车辆和目标车辆的特征通过大小为 16 的线性层输入,再通过深度为 1、大小为 32 的 GRU 生成编码特征,然后使用缩放点积注意力机制为节点融合周围车辆信息,最后通过 GNN 层融合周围节点信息,获得最新节点编码。策略网络的生成器 G 融合车辆和节点特征,使

用 MLP 输出采取动作的得分;专家数据通过同样的结构得到真实轨迹的动作;判别器网络 D 通过 MLP 判断 G 生成的动作是否来自专家数据动作,采用 Adam 对判别器进行梯度上升,采用 TRPO 对生成器进行梯度下降。轨迹生成器由目标车辆编码和节点编码循环输入策略网络得到未来轨迹的节点序列,通过多头注意力机制得到一个上下文向量 C ,结合潜变量 z 和目标车辆编码,输入 MLP 得到未来轨迹。

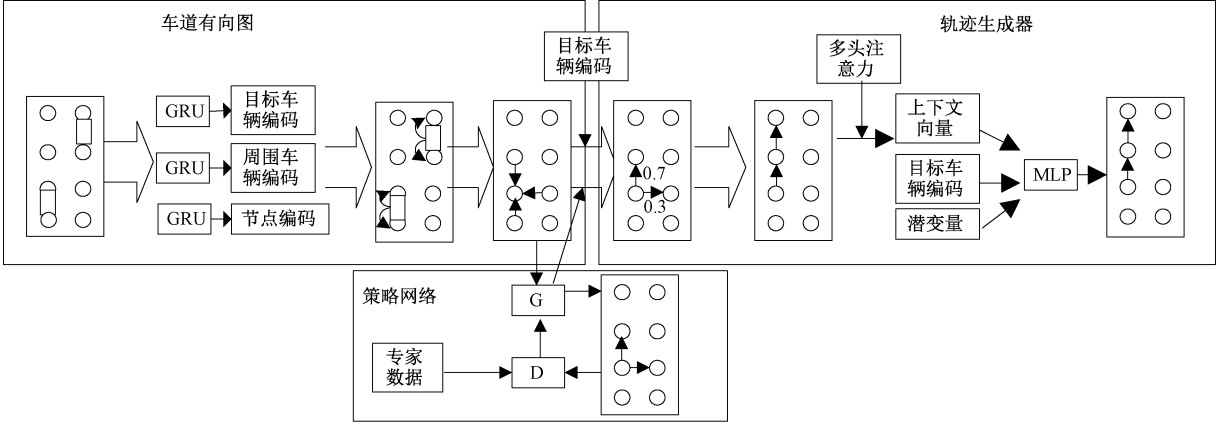


图 1 基于有向图的强化学习自动驾驶轨迹预测模型结构图
Figure 1 Structure diagram of reinforcement learning autonomous driving trajectory prediction model based on directed graph

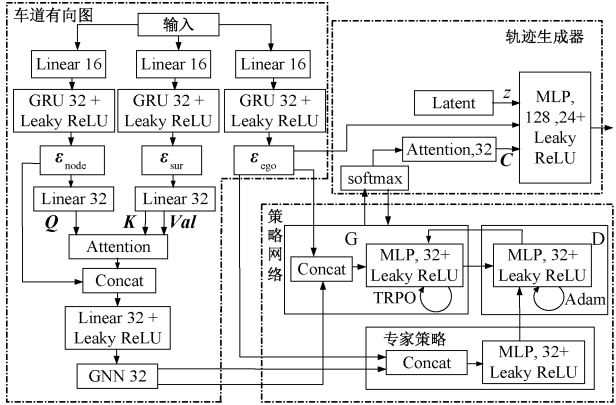


图 2 基于有向图的强化学习自动驾驶
轨迹预测模型网络结构图
Figure 2 Network structure diagram of reinforcement learning autonomous driving trajectory prediction model based on directed graph

2.4.1 基于有向图的高清地图构建

构建基于有向图的高清地图需要对每个车辆轨迹和车道进行编码。将地图中所有的车道片段作为车道有向图输入 $\vec{G}(V, E)$ 的节点,使用 GRU 编码器分别对目标车辆轨迹 $\alpha_{-t_h,0}^{\text{ego}}$ 、周围车辆轨迹 $\alpha_{-t_h,0}^i$ 和节点特征 $f_{1:N}^v$ 进行编码,输出目标车辆运动编码 ϵ_{ego} 、周围车辆运动编码 ϵ_{sur}^i 和初始节点编码 ϵ_{node}^v :

$$\epsilon_{\text{ego}} = \text{GRU}(\alpha_{-t_h,0}^{\text{ego}}); \quad (1)$$

$$\epsilon_{\text{sur}}^i = \text{GRU}(\alpha_{-t_h,0}^i); \quad (2)$$

$$\epsilon_{\text{node}}^v = \text{GRU}(f_{1:N}^v). \quad (3)$$

作为有向图中的节点 v ,每个车道节点代表一个固定长度的车道段,它的空间坐标为该车道段中心线的中点,节点的方向为车道中心线的方向。对于任意 2 个空间相邻的节点,给出 5 种属性:前置、后继、左邻、右邻和不可达。前置和后继分别是可直接到达 v 和从 v 出发可到达的节点;左、右邻指在不违反交通规则的情况下可以直接到达的左侧和右侧节点;不可达为违反交通规则时可以到达的节点。

使用特征向量表示节点,节点 v 以固定的时间间隔更新状态,将节点 v 离散为 N 个状态,由特征向量序列 $f_{1:N}^v = [f_1^v, f_2^v, \dots, f_N^v]$ 表示。这里每一个 $f_n^v = [x_n^v, y_n^v, \eta_n^v, I_n^v, \tau_n^v]$;其中 x_n^v, y_n^v 和 η_n^v 是 v 的第 n 个状态的位置和偏航角度; I_n^v 是一个二维二进制矢量,表示该姿态是否位于停车线或人行横道上; τ_n^v 对应于属性特征,即周围节点的连接属性和速度限制。

有向图中的边 E 均为 2 个连通节点间的连线,该设置便于捕获交通流的方向,使得场景中的车辆运动轨迹为合法路径。边具有 2 种属性,后继边

(E_{suc})和近边(E_c), E_{suc} 连接节点的后继节点,如图 3 中实线。一个节点可以有多个后继节点,多个节点也可以有相同的后继节点; E_c 连接距离在固定范围内、且运动方向在偏航阈值内的节点,如图 3 中虚线。

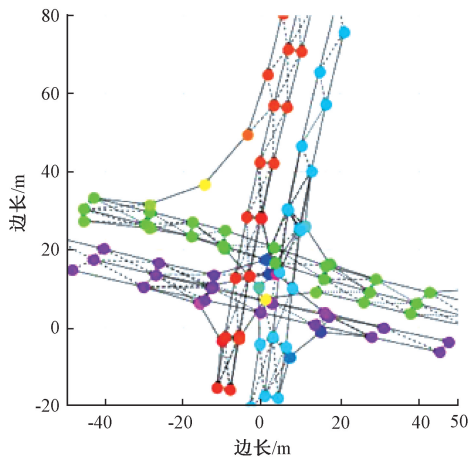


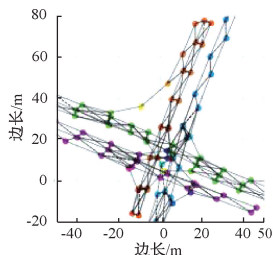
图 3 节点与边表示图

Figure 3 Nodes and edges represent graphs

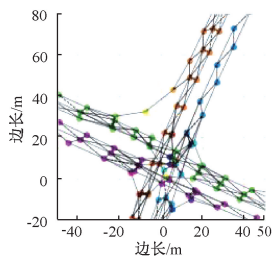
车道图结构如图 4 所示,图 4(a)展示了某一车辆驾驶场景的车道栅格表示,图 4(b)~4(f)按时间顺序展示了该车辆连续运动下的车道图结构。



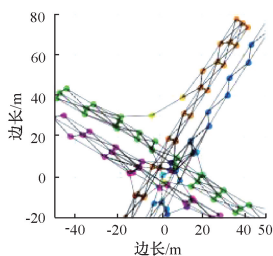
(a) 驾驶场景栅格表示



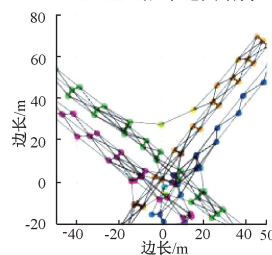
(b) 1.0 s后车道图结构



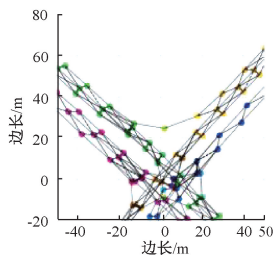
(c) 1.5 s后车道图结构



(d) 2.0 s后车道图结构



(e) 2.5 s后车道图结构



(f) 3.0 s后车道图结构

图 4 车道图结构

Figure 4 Lane diagram structure

为有效预测目标车辆轨迹,需要考虑目标车辆与周围交通参与者交互对自身的影响。其中影响较

大的一般为空间距离较近的交通参与者,特别是与自身运行路线有交互的交通参与者。因此使用缩放点积注意力,用附近的车辆轨迹编码来更新节点编码。这些用来更新节点编码的交通参与者设定在车道节点距离阈值内,这样既使得节点更新规则更加合理,又能减少轨迹生成器的工作。

首先通过线性投影周围车辆的编码 ϵ_{sur}^i 获得键 K 和值 Val ,线性投影节点编码 ϵ_{node}^v 获得查询 Q 。

$$Q = \text{linear}(\epsilon_{node}^v); \quad (4)$$

$$K, Val = \text{linear}(\epsilon_{sur}^i). \quad (5)$$

然后使用缩放点积注意力得到需要关注的交通参与者注意力值 Att 。

$$Att = \text{softmax}(QK^T) Val. \quad (6)$$

最后将注意层的输出与原始节点编码进行拼接,得到更新后的节点编码。

$$\epsilon_{node}^v = \text{concat}(\epsilon_{node}^v, Att). \quad (7)$$

此时节点特征包含车道信息和周围车辆信息,为了考虑更多的车辆和车道信息,使用图神经网络(GNN)从邻近节点来聚集局部上下文更新节点信息。

$$\epsilon_{node}^v = \text{GNN}(\epsilon_{node}^v, E_{suc}, E_c). \quad (8)$$

更新节点时,将后继边和近边视为双向的,这样便于沿着每个节点周围的所有方向聚合上下文。GNN层的输出是车道图编码器学习到的最终节点编码。使用GCN层时,采用文献[7]中的逐层传播规则,邻接矩阵包括后继边、近边和自循环,每个节点的输入输出维度为32。使用GAT层时,采用文献[8]中的逐层传播规则,选择一个注意力头,输入输出维度为32。

2.4.2 策略学习

使用GAIL来生成学习的策略,在GAIL中,生成器G代表策略网络,它会根据不同的状态采取不同的动作。而判别器D则要区分示例数据和G生成的数据,这可以转化成一个简单的二分类问题,即将当前的状态和动作作为输入,得到这个动作来自示例数据的概率。如果这个状态动作对来自示例样本,那么这个概率越接近于1越好;而如果这个状态动作对来自G生成的样本,那么这个概率越接近于0越好。对G来说,希望自己的策略越来越接近于示例策略,那么就可以使用D输出的概率作为奖励,来更新自身的策略,如果D给出的概率越高,说明在这一状态下采取的动作是一个较优的动作,就提高该动作出现的概率,反之则是一个较差的动作,降低其出现的概率。当D无法区分G生成的数据和真实数据时,则G已成功匹配真实数据分布。

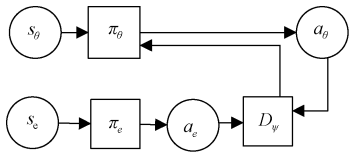


图5 GAIL流程图

Figure 5 GAIL Flow Chart

图5为GAIL流程图,用 π_θ 表示要学习的策略, π_e 表示专家策略,状态 s_θ 通过生成器生成动作 a_θ ,状态 s_e 通过专家策略生成动作 a_e , D_ψ 判别该动作是由生成器生成还是来自示例数据。生成器网络和专家策略网络通过concat操作融合车辆和节点特征,使用共享权值的全连接神经网络MLP输出采取动作的得分,这样可以减少训练参数。该MLP有2层隐藏层,每个隐藏层大小为32,激活函数为LeakyReLU。

$$a_\theta = \pi_\theta(a | s; \theta) = \text{MLP}(\text{concat}(\mathbf{\epsilon}_{\text{ego}}, \mathbf{\epsilon}_{\text{node}})) ; \quad (9)$$

$$a_e = \pi_e(a | s; e) = \text{MLP}(\text{concat}(\mathbf{\epsilon}_{\text{ego}}, \mathbf{\epsilon}_{\text{node}})) 。 \quad (10)$$

对于包含由 θ 参数化的策略 π_θ 和由 ψ 参数化的判别器 D_ψ 的GAIL,其损失函数可表示为

$$\max_{\psi} \min_{\theta} V(\theta, \psi) = \mathbb{E}_{(s,a) \sim X_e} [\ln D_\psi(s, a)] + \mathbb{E}_{(s,a) \sim X_\theta} [\ln(1 - D_\psi(s, a))] 。 \quad (11)$$

式中: $(s, a) \sim X_e$ 表示 (s, a) 采样自示例数据; $(s, a) \sim X_\theta$ 表示采样自生成器策略; $D_\psi(s, a)$ 表示判别器判别 (s, a) 来自示例数据的概率。通过反复执行梯度更新进行优化,使用Adam对参数 ψ 进行梯度上升,使用TRPO对生成器参数 θ 进行梯度下降,当从 π_θ 中提取的状态动作对 (s, a) 与从 π_e 中提取的状态动作对 (s, a) 无法区分时,判别器网络可以解释为向策略网络提供学习信号的奖励函数:

$$r(s, a; \psi) = -\ln(1 - D_\psi(s, a))。 \quad (12)$$

尽管它可能与专家优化的真实奖励函数有很大不同,但它可以用于将 π_θ 驱动到与 π_e 类似的状态动作空间区域,即使策略 π_θ 符合该数据集的车辆运动。

2.4.3 轨迹方案生成

将目标车辆的运动编码和节点编码输入策略网络,得到目标车辆在每个节点输出方向上的离散概率分布。假设在车辆轨迹中每个位姿都分配给图中最近的节点,这在车道交叉的路口容易出现姿态分配错误的情况。为了防止这种情况的发生,只考虑目标车辆运动方向偏航阈值内的车道节点。

$$P_{(u,v)} = \text{softmax}(\pi_\theta(a | s; \theta)), (u, v) \in \mathbf{E}。 \quad (13)$$

式中: $(u, v) \in \mathbf{E}$ 表示节点在偏航阈值内。

策略网络为目标车辆提供了合理的未来轨迹,通过循环输入当前状态 s ,可得到未来轨迹经过的

一个节点序列 $[v_1, v_2, \dots, v_M]$,使用多头注意力机制在节点序列上聚合地图和周围车辆信息,如图6所示。

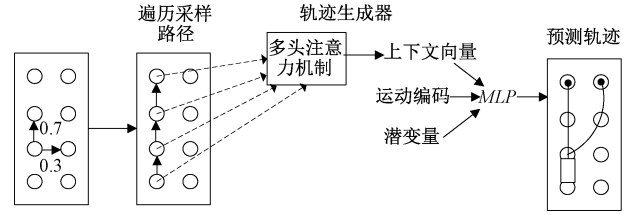


图6 轨迹生成器

Figure 6 Track generator

多头注意力可以从不同角度采样,获得更加丰富的特征信息,使用32个平行的注意力头部,线性投影目标车辆的运动编码以获得查询,同时线性投影节点特征 $[\mathbf{\epsilon}_{\text{node}}^1, \mathbf{\epsilon}_{\text{node}}^2, \dots, \mathbf{\epsilon}_{\text{node}}^M]$ 获取用于计算注意力的键和值,每个头 i 的计算方式:

$$Q_i = \text{linear}(\mathbf{\epsilon}_{\text{ego}}), i = 1, 2, \dots, 32; \quad (14)$$

$$K_i, Val_i = \text{linear}(\mathbf{\epsilon}_{\text{node}}^1, \mathbf{\epsilon}_{\text{node}}^2, \dots, \mathbf{\epsilon}_{\text{node}}^M)。 \quad (15)$$

每个头同时计算其注意力,最后将结果拼接起来,多头注意层输出一个上下文向量 \mathbf{C} 来编码路径。

$$Att_i = \text{softmax}(Q_i K_i^T) Val_i; \quad (16)$$

$$\mathbf{C} = \text{concat}(Att_1, Att_2, \dots, Att_{32})。 \quad (17)$$

每个不同的节点序列都会产生不同的上下文向量,能够沿着一组不完全相同的路线预测轨迹。

仅路线的多样性还不足以证明未来轨迹的多模态,目标车辆可以在预定路线上刹车、加速和转弯等。为了模拟车辆速度和加速度的连续变化,从一个连续分布中抽样潜在变量 z 来调整预测的轨迹,连续分布使用多元标准正态分布。

最后在轨迹生成器中采样得到轨迹 $\beta_{1:t_f}^k$ 。首先对 π_θ 抽样并得到 \mathbf{C}_k ,再从潜在分布中采样 z_k ,将它们与 $\mathbf{\epsilon}_{\text{ego}}$ 融合,并将它们通过MLP输出在 t_f 时间步上的未来位置。MLP隐藏层大小为128,输出大小为24,激活函数为LeakyReLU:

$$\beta_{1:t_f}^k = \text{MLP}(\text{concat}(\mathbf{\epsilon}_{\text{ego}}, \mathbf{C}_k, z_k))。 \quad (18)$$

采样过程会产生相似或重复的轨迹,本文的轻量级生成器允许同时采样大量轨迹,为了获得去冗余的最终轨迹分布 K 模式集,使用 K -means聚类算法,输出最终 K 预测集 $\beta_{1:t_f} = [\beta_{1:t_f}^1, \beta_{1:t_f}^2, \dots, \beta_{1:t_f}^K]$ 。使用与实际运动轨迹 α_{gt} 相关的平均位移误差来训练生成器,以避免惩罚模型输出的合理轨迹:

$$L = \min_K \frac{1}{t_f} \sum_{t=1}^{t_f} \|\alpha_t^k - \alpha_t^{\text{gt}}\|_2。 \quad (19)$$

3 实验仿真与结果分析

3.1 数据集分析

在数据集 nuScenes 上进行仿真,利用目标过去 2 s 的运动历史和地图来预测未来 6 s 的轨迹。这个自动驾驶数据集由 Motional 公司在波士顿和新加坡进行采集,这 2 个城市交通情况复杂,有助于增强数据的多样性,并且这 2 个城市位于 2 个大洲,交通规则、交通标志等多有不同,有助于验证方法的泛化性。用于采集的车辆装备了 1 个旋转雷达,5 个远程雷达传感器和 6 个相机。数据集包含 15 h 的驾驶时长,1 000 个场景,每个场景 20 s,带有地面实况注释和高清地图。车辆有手动标注的 3D 边框,以 2 Hz 的频率发布。

3.2 评价指标

在 Windows10 平台搭建 Python3.7 版本, Cuda10.2 版本,基于 pytorch1.7.1 的深度学习框架。计算机的配置为 Intel(R) Xeon(R) Silver 4110 CPU@ 2.10 GHz、32.0 GB RAM、NVIDIA Quadro p4000。采用了广泛使用的预测指标,例如, K 预测的最小位移误差 $MinADE_K$,即 K 个预测在所有预测范围内预测轨迹与真实轨迹的平均欧式距离,单位为 m;失误差率 $MissRate_K$,即距离地面实际位置 2 m 以上的预测在所有预测中所占比例;越野率 $offRoadRate$,衡量的是预测中离开车道的比例,因为在 nuScenes 中的所有示例没有偏离车道,这个数值理论上为零; K 预测的最小最终位移误差 $MinFDE_K$,即 K 个预测中实际最终位置和预测最终位置之间的最小欧氏距离,单位为 m。

3.3 模型仿真比较与分析

首先评估了 $K = 5$ 与 $K = 10$ 时轨迹的最小平均位移误差、失误差率和越野率。表 1 展示了本文模型与 SG-Net^[10]、LaPred^[11]、Multipath^[12]、P2T^[13] 各指标的比较结果。由表 1 可看出,本文的模型明显优于其他模型。

其次评估了 $K = 1$ 、 $K = 5$ 与 $K = 10$ 时预测轨迹的最小最终位移误差 $MinFDE_K$,如表 2 所示。本文模型在 3 种情况下与 GOHOME^[14]、Multipath、MHA-JAM^[15]、LaPred 进行比较,在 $K = 5$ 与 $K = 10$ 时得到最好的结果,在 $K = 1$ 时排第 2 名。

对采用的方法进行比较。SG-Net 使用了编码器解码器架构,通过逐步估计目标位置,并将其作为下一步估计的状态对估计进行预测。LaPred 通过 1D-CNN 和 LSTM 提取车辆和车道特征,将这些特征与通过自监督学习的注意力权重融合在一起,使用

MTP 生成多个轨迹。Multipath 使用 ReaNet 先提取整个场景特征,然后为场景中的每个目标裁剪一个区域,以车辆为坐标中点为每个目标预测轨迹,通过模仿学习训练模型得到输出。P2T 使用栅格地图表示地图信息,通过全卷积网络输出目标和路径奖励,再使用逆强化学习得到路径和状态奖励,最后使用基于注意力的轨迹生成器生成轨迹。GOHOME 使用图结构表示高清地图和车辆轨迹,通过稀疏投影的热图输出表示交通场景中车辆的未来位置概率分布。MHA-JAM 使用 CNN 编码高清地图,LSTM 编码车辆轨迹,通过车辆和静态场景的联合表示生成注意头的键值对,令每个注意力头单独生成一个未来轨迹。

表 1 本文模型与其他模型的性能对比

Table 1 Comparison of performance between this model and other models					
模型	$MinADE_5/$ m	$MinADE_{10}/$ m	$MissRate_5$	$MissRate_{10}$	$OffroadRate$
SG-Net	1.86	1.40	0.67	0.52	0.05
Multipath	1.63	1.50	0.75	0.74	0.38
LaPred	1.47	1.12	0.53	0.46	0.09
P2T	1.45	1.16	0.64	0.46	0.03
本文模型	1.29	0.95	0.53	0.34	0.03

表 2 最小最终位移误差对比

Table 2 Minimum final displacement error comparison				m
模型	$MinFDE_1$	$MinFDE_5$	$MinFDE_{10}$	
GOHOME	6.99	2.87	1.94	
Multipath	9.34	3.62	2.93	
MHA-JAM	8.57	3.72	2.21	
LaPred	8.12	3.37	2.39	
本文模型	7.32	2.56	1.60	

3.4 消融实验

对 GNN 层和模仿学习方法进行消融研究,探索他们对模型的影响,如表 3 所示。首先,对 GNN 层进行消融研究,分别使用 GCN 和 GAT 聚合周围节点特征,生成最终节点表示。GAT 层相较于 GCN 层有所提升,这可能是因为注意力机制关注邻近节点,更有利于目标车辆采取接近真实的下一步动作。其次,进行模仿学习方法的研究,分别采取行为克隆和生成式对抗模仿学习。结果显示,使用生成式对抗模仿学习时,模型指标更优秀,这是因为行为克隆是一种监督学习,它不能预测专家数据中没有出现的情况,生成式对抗模仿学习则在专家数据中没有出现的场景中也有更好的性能。

对生成器的注意力机制和潜变量进行消融研究,探索他们对模型的影响,如表 4 所示。首先考虑

没有注意力机制和潜变量的模型预测结果,然后分别考虑只有注意力机制和只有潜变量时的模型预测结果。结果显示,当其他条件不变时,没有注意力机制和潜变量的模型预测结果更差。而只有注意力机制的模型比只有潜变量的模型表现更好,说明注意力机制较好地提取了轨迹节点中的重要信息。

表 3 GNN 层和模仿学习方法消融研究对比

Table 3 Comparison of GNN layer and imitation learning methods ablation studies compared									
GNN 层	学习策略	$MinADE_5/m$	$MinADE_{10}/m$	$MissRate_5$	$MissRate_{10}$	$offRoadRate$	$MinFDE_1/m$	$MinFDE_5/m$	$MinFDE_{10}/m$
—	GAIL	1.48	1.12	0.66	0.53	0.04	9.04	2.94	1.85
GCN	BC	1.43	1.09	0.64	0.51	0.04	8.53	2.76	1.75
GCN	GAIL	1.38	1.02	0.61	0.43	0.04	7.92	2.66	1.68
GAT	BC	1.31	0.97	0.56	0.37	0.03	7.57	2.59	1.63
GAT	GAIL	1.29	0.95	0.53	0.34	0.03	7.32	2.56	1.60

表 4 生成器消融研究对比

Table 4 Comparison of Generator ablation studies compared									
模型		$MinADE_5/m$	$MinADE_{10}/m$	$MissRate_5$	$MissRate_{10}$	$offRoadRate$	$MinFDE_1/m$	$MinFDE_5/m$	$MinFDE_{10}/m$
—	—	1.59	1.12	0.57	0.48	0.07	9.12	2.87	1.82
Attention	—	1.38	1.07	0.63	0.43	0.05	7.93	2.65	1.64
—	潜变量	1.43	1.10	0.65	0.47	0.04	8.06	2.69	1.71
Attention	潜变量	1.29	0.95	0.53	0.34	0.03	7.32	2.56	1.60

3.5 结果评判

nuScenes 轨迹预测结果如图 7 所示。可以通过不同场景的预测结果来评判模型的效果,每个示例从左至右分别展示了地图场景、生成的轨迹和真实轨迹。图 7(a)为目标车辆沿着左转车道行驶,并且航向角已经向左侧偏出,预测出该车辆将左转行驶到正确道路上;图 7(b)为目标车辆在丁字路口的最右侧车道上,预测出该车辆更多可能向右转向行驶到右侧车道,也可能继续直行;图 7(c)为目标车辆在直行车道上,但航向角较大向左偏出,预测出它极大可能驶向左转车道,并有可能直线行驶;图 7(d)为目标车辆在低速经过十字路口时,预测出它的多种换道可能,这非常符合现实中的场景;图 7(e)为目标车辆在丁字路口,航向角向左偏出,预测它向左转向;图 7(f)为目标车辆在多出口直行车道上,预测它极大可能继续直行,并有多转向可能。本文的预测涵盖了车道变换的可能性,显示出良好的多模态性能,真实轨迹均可从预测轨迹中找出,并且准确度高。

道上,但航向角较大向左偏出,预测出它极大可能驶向左转车道,并有可能直线行驶;图 7(d)为目标车辆在低速经过十字路口时,预测出它的多种换道可能,这非常符合现实中的场景;图 7(e)为目标车辆在丁字路口,航向角向左偏出,预测它向左转向;图 7(f)为目标车辆在多出口直行车道上,预测它极大可能继续直行,并有多转向可能。本文的预测涵盖了车道变换的可能性,显示出良好的多模态性能,真实轨迹均可从预测轨迹中找出,并且准确度高。

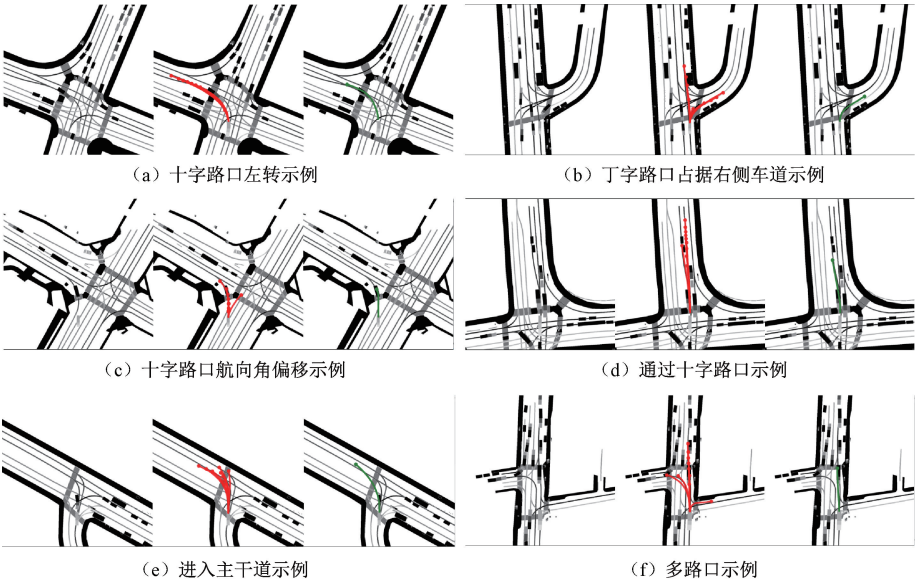


图 7 轨迹预测结果示例图

Figure 7 Example plot of trajectory prediction results

4 结论

本文使用有向图构建高清驾驶场景地图,使用

GAIL 方法拟合数据集状态动作对分布,通过采样遍历生成自动驾驶预测轨迹。基于有向图构建高清地图能够更好地提取地图拓扑结构,采用 GAIL 的方

法能够解决行为克隆级联错误的问题,得到更好的结果。量化结果相比于其他方法, $K=5$ 时,最小最终位移误差 $MinFDE_5$ 提高了 10.8%; $K=10$ 时最小最终位移误差 $MinFDE_{10}$ 提高了 17.53%,最小平均位移误差 $MinADE_{10}$ 提高了 9.52%,失误差率 $MissRate_{10}$ 减少了 28.26%。结果显示本文的方法能够得到符合真实轨迹的长距离预测,相比于其他方法,在 nuScenes 数据集上具有更好的性能。

参考文献:

- [1] 中华人民共和国国家发展和改革委员会. 智能汽车创新发展策略[EB/OL]. (2020-02-24)[2022-12-24]. https://www.ndrc.gov.cn/xxgk/zcfb/tz/202002/t20200224_1235917.html.
National Development and Reform Commission. Innovative development strategy for intelligent vehicles[EB/OL]. (2020-02-24)[2022-12-24] https://www.ndrc.gov.cn/xxgk/zcfb/tz/202002/t20200224_1235917.html.
- [2] BANSAL M, KRIZHEVSKY A, OGALE A. Chauffeur-Net: learning to drive by imitating the best and synthesizing the worst[EB/OL]. (2018-12-07)[2022-12-24]. <https://arxiv.org/abs/1812.03079>.
- [3] 王丙琛, 司怀伟, 谭国真. 基于深度强化学习的自动驾驶车控制算法研究[J]. 郑州大学学报(工学版), 2020, 41(4): 41-45, 80.
WANG B C, SI H W, TAN G Z. Research on autopilot control algorithm based on deep reinforcement learning[J]. Journal of Zhengzhou University (Engineering Science), 2020, 41(4): 41-45, 80.
- [4] HO J, ERMON S. Generative adversarial imitation learning[C]//Proceedings of the 30th International Conference on Neural Information Processing Systems. New York: ACM, 2016: 4572-4580.
- [5] CAESAR H, BANKITI V, LANG A H, et al. nuScenes: a multimodal dataset for autonomous driving[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 11618-11628.
- [6] LIANG M, YANG B, HU R, et al. Learning lane graph representations for motion forecasting[C]//Computer Vision-ECCV 2020: 16th European Conference. New York: ACM, 2020: 541-556.
- [7] KIPF T N, WELING M. Semi-supervised classification with graph convolutional networks[EB/OL]. (2016-09-09)[2022-12-24]. <https://arxiv.org/abs/1609.02907>.
- [8] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[EB/OL]. (2017-10-30)[2022-12-24]. <https://arxiv.org/abs/1710.10903>.
- [9] 张三川, 马啸. 基于轨迹加权预测的主动避撞安全距离模型及算法[J]. 郑州大学学报(工学版), 2022, 43(3): 104-110.
ZHANG S C, MA X. A safe distance model and algorithm for active collision avoidance based on weighted prediction of trajectory[J]. Journal of Zhengzhou University(Engineering Science), 2022, 43(3): 104-110.
- [10] WANG C H, WANG Y C, XU M Z, et al. Stepwise goal-driven networks for trajectory prediction[C]//IEEE Robotics and Automation Letters. Piscataway: IEEE, 2022: 2716-2723.
- [11] KIM B, PARK S H, LEE S, et al. LaPred: lane-aware prediction of multi-modal future trajectories of dynamic agents[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2021: 14631-14640.
- [12] CHAI Y N, SAPP B, BANSAL M, et al. MultiPath: multiple probabilistic anchor trajectory hypotheses for behavior prediction[EB/OL]. (2019-10-12)[2022-12-24]. <https://arxiv.org/abs/1910.05449>.
- [13] DEO N, TRIVEDI M M. Trajectory forecasts in unknown environments conditioned on grid-based plans[EB/OL]. (2021-04-29)[2022-12-24]. <https://arxiv.org/abs/2001.00735>.
- [14] GILLES T, SABATINI S, TSISHKOU D, et al. GO-HOME: graph-oriented heatmap output for future motion estimation[C]//2022 International Conference on Robotics and Automation (ICRA). New York: ACM, 2022: 9107-9114.
- [15] MESSAOUD K, DEO N, TRIVEDI M M, et al. Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation[C]//2021 IEEE Intelligent Vehicles Symposium (IV). New York: ACM, 2021: 165-170.

Reinforcement Learning Autonomous Driving Trajectory
Prediction Based on Directed Graph

CUI Jianming¹, LIN Fanrong¹, ZHANG Di¹, ZHANG Luning¹, LIU Ming²

(1. School of Information Engineering, Chang'an University, Xi'an 710018, China; 2. National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China)

Abstract: As an important part of autonomous driving, trajectory prediction aimed to forecast the vehicle's driving path, so that the vehicle could make path planning according to the driving estimation, so as to make safe and accurate decisions. Firstly, in order to improve the accuracy of vehicle trajectory prediction, the directed graph method was used to construct a high-definition driving scene map, and the directed graph method vectorized the map information to effectively extract the map topology. Secondly, GAIL was used to learn the driving strategy of the dataset through the confrontation game between the generator and the discriminator, so as to adopt the corresponding driving behavior according to the current state. Finally, the multimodal prediction trajectory scheme was obtained by sampling traversal. Simulation was carried out on the nuScenes motion prediction dataset. The quantitative results showed that compared with other methods, when $K=5$, the minimum final displacement error $MinFDE_5$ was increased by 10.8%; when $K=10$, the minimum final displacement error $MinFDE_{10}$ increased by 17.53%, the minimum average displacement error $MinADE_{10}$ increased by 9.52%, and the error rate $MissRate_{10}$ decreased by 28.26%. The evaluation showed that the generated trajectories were multimodal, could conform to the basic structure of the scene, with improved accuracy.

Keywords: autonomous driving; trajectory prediction; directed graph; reinforcement learning; GAIL; attention mechanism; multimodal prediction

(上接第 45 页)

learning strategy was employed to allocate the remaining points, these unprocessed points were used as noise points, and they were classified into the class of its nearest neighbor. The improved density peak clustering algorithm was used to analyze the bounding box of the controlled knife, optimize the size of the priori box, and improve the matching degree between the priori box and the size of target object, so as to solve the problem that the clustering effect of the K -means clustering algorithm in the YOLOv5 model such as its unstability and the low convergence rate of large-scale data. Moreover, The C3 module in the Backbone network was combined with the CBAM attention mechanism, named CBAMC3 module, which could improve the model's ability to extract target features, solve the problem that the YOLOv5 algorithm was not effective for small target detection, and improve the model accuracy. The experimental results showed that the values of P , R , $mAP@0.5$, $mAP@0.5:0.95$ of the improved model YOLOv5-Plus on the customed data set were 98.14%, 95.80%, 97.56%, and 76.68%, respectively, which is 1.64%, 1.59%, 1.51%, and 3.26% higher than that of YOLOv5 before improvement, and also verified that the proposed model could effectively improve the detection performance of controlled knife in public areas.

Keywords: controlled knife detection; public safety; target detection; clustering algorithm; attention mechanism