

基于贝叶斯深度图学习的城市出行时间分布估计

摘要

1. 引言

2. 初步工作

2.1 相关工作

2.2 行程时间分布估计

3. 方法论

3.1 概述

3.2 地图匹配与行驶时间建模

3.3 旅行时间分布估计器

3.4 对抗性训练模式

3.5 DeepTTDE训练与推理

4. 案例研究

4.1 数据和配置

4.2 行程时间分布估计

摘要

路段行程时间估计在智能交通运营管理中起着至关重要的作用。由于交通的不稳定带来的不确定性，旅行时间估计用概率分布比确定性模型更好地描述。现有的旅行时间分布估计方法大多基于预定义的概率分布。其他方法虽然放松了约束，但不能利用拓扑信息，而且数据效率低。本文提出了一种新的基于贝叶斯和几何深度学习的方法，用于基于车辆GPS轨迹估计城市交通网络中道路链路的行程时间分布。特别地，首先对历史轨迹或实时轨迹进行预处理，构建部分走时图，并将其输入到定制的贝叶斯图自动编码器中，重建多个完整的走时图。此外，我们还采用了一个辅助神经网络来促进所提方法的参数训练，遵循对抗性训练的原则。为了评价所提出的方法，我们在一系列综合案例研究中使用了一个真实世界的车辆轨迹数据集。实验结果表明，所提出的方法优于目前最优的基线，约10%的Kullback–Leibler散度减少。

1. 引言

智能交通系统(ITS)是智能城市的重要组成部分[1]。智能交通作为现代经济中的基础性和战略性产业,以缓解交通拥堵、减少化石燃料消耗、提高运输效率为目标,引起了业界和研究界的广泛关注[2]、[3]。在ITS中,行程时间估计在支持交通监控和车辆路径选择等高级ITS服务方面起着至关重要的作用[4]。更好的估计会改善这些服务的性能,进而导致交通用户的更好体验,包括但不限于车辆和通勤者[5]–[7]。

现有的工业解决方案和研究工作大多是基于历史数据准确地提供交通网络中道路链路的行程时间[8]。然而,准确的路段行程时间受到许多难以记录的因素的影响,如车辆状况和驾驶方式[9]。结果,这些不确定性可能阻碍这些去终止旅行时间估计器产生可靠的估计[4], [8]。针对这一问题,学术界对所谓的行程时间分布估计(TTDE)的研究日益关注,它假定路段行程时间是由交通的异质性和动态性驱动的时变随机变量[10]、[11]。Pioneer的TTDE工作通常假设旅行时间服从高斯或对数正态分布,并开发了经验结果来讨论它们各自的适用性[12], [13]。然而,这些参数模型虽然易于理论分析,但不能提供足够的表达来表示旅行时间动力学[9], [14]。这种关注导致了目前在没有高斯假设的情况下对TTDE的最先进的研究。

一般说来,在没有预先假设的情况下,有两种主流的旅行时间分布方法。最近, Prokhorchuk等人。文[4]提出用高斯copula图形lasso模型将旅行时间的非高斯特性转化为高斯特性。其主要原理是利用copula模型中的协方差矩阵来挖掘相邻路段之间行程时间的统计相关性。由于它的参数性质,所提出的方法可以利用低分辨率和稀疏的GPS轨迹估计时间分布,而不依赖于以前常见的假设,如高斯性和链路无关性。随后,段等人。更进一步,设计了一个基于核密度估计的TTDE模型[8]。采用这种非参数概率密度估计方法,可以很好地捕捉随道路条件变化的时变行程时间动力学。

然而,在TTDE问题上存在研究空白。现有的TTDE方法主要使用稀疏的车辆轨迹[4]或有限的交通检测器[8]来估计分布。一方面,这些方法可以应用到实际的交通网络中,在低数量和/或低质量的交通数据支持下,扩大了其适用性。另一方面,当处理大量的组织车辆痕迹时,要么必须采用数据下采样[4],要么可能会经历很高的计算负担[8],这是广泛预期的,也是当代众包技术非常合理的。因此,就产生了数据利用不足的问题。目前已有相关的路线行程时间预测方法,旨在提供城市区域内两个地点之间任意路线的时间估计。然而,这些方法都不能满足TTDE的要求,因为后者更好地用于实时生成最优路线,而前者更多地用于行程分析和离线路线推断[15]。研究差距倡导学术界和工业界为TTDE设计新的解决方案。

数据高效的TTDE方法的前景使深度学习成为一个引人注目的解决方案,它利用多层感知器从原始数据中挖掘潜在的特征[16]。为了弥补这一研究空白,本文基于深度学习技术的最新发展,提出了一种新的城市域TTDE方法。我们特别考虑了存在于原始的、历史的和实时的车辆GPS轨迹中的不确定性,并使用贝叶斯深度学习[17]的原理来从中学习。此外,我们采用图卷积概念[18]进行非欧几里得空间特征提取,它比基于标准张量的数据分析[7]更好地描述了交通网络。该模型是利用数据挖掘和机器学习研究中的深度神经网络构建块建立的,为TTDE提供了一种新的解决方案,并为未来TTDE模型的发展提供了见

解。这项工作的主要努力概述如下：

- 提出了一种基于深度学习技术的TTDE方法。采用贝叶斯和几何深度学习的设计原理来构造新的深度神经网络。
- 为了提高该方法的训练性能，我们采用了一种对抗性训练方法，并设计了该方法的训练方案和在线推理方案。
- 我们在一个具有十亿多个GPS记录的真实世界数据集上验证了所提方法的有效性。实证结果证明了该方法的重要性和敏感性。

据我们所知，这是利用深度学习估计道路路段行程时间分布的开创性工作之一。该方法不仅适用于TTDE，而且可以进一步适用于智能交通系统中的各种概率估计问题。

本文的其余部分组织如下。第2节简要概述了相关工作，并对TTDE问题进行了阐述。第三节详细介绍了基于深度学习的TTDE方法，并对其训练和推理进行了详细的介绍。第4节介绍了一个真实世界数据集的案例研究，以证明该方法的有效性。最后，在第五节对本文工作进行了总结。

2. 初步工作

2.1 相关工作

旅行时间估计是文献[4]中研究较多的一个研究课题，是时空数据挖掘[19]的一个应用。相关研究一般可分为两大类，即确定性行程时间估计，其目的是估计道路链路的平均行程时间，而TTDE则提供了行程时间的估计概率分布。随着big 2数据时代的出现，这两个问题现在都明显依赖于历史旅行时间数据的可用性[20]。提出了多种基于数据挖掘的方法来开发均值估计，包括但不限于线性回归[21]、支持向量回归[22]、人工神经网络[23]、隐马尔可夫模型[24]等。我们请读者参阅[4]、[7]以获得确定性旅行时间估计文献的详细概述。

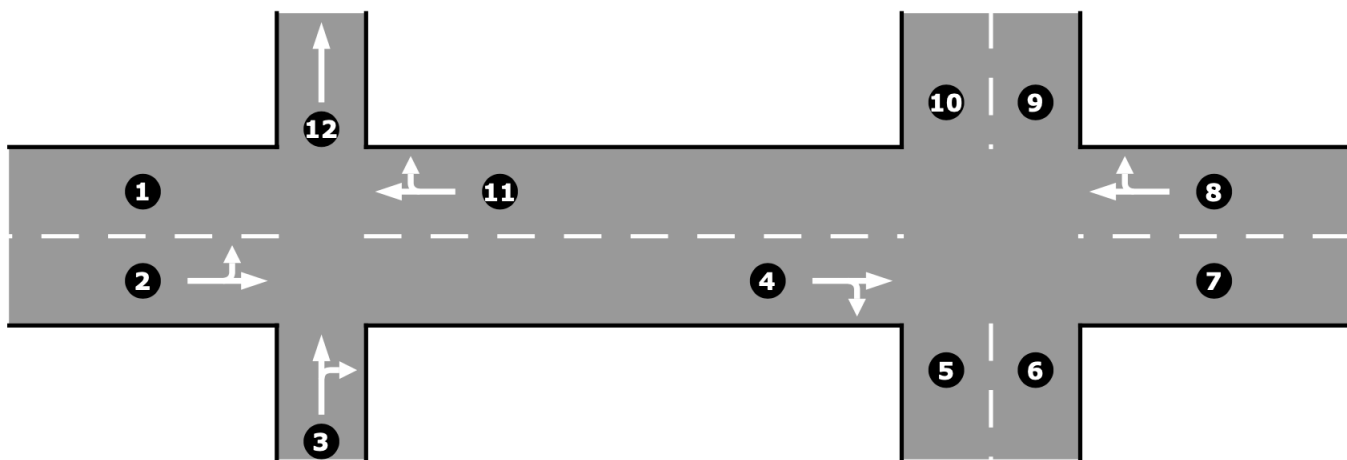
随着确定性估计问题的出现，TTDE在过去几年中受到了越来越多的关注[8]。在高斯或对数正常旅行时间的假设下进行了一系列研究。文献[12]和[13]分析了高斯分布和对数正态分布在交通流条件和空间相关性下的旅行时间建模中的适用性。参考文献[25]进一步研究了替代概率分布作为基本参数模型。文献[26]使用高斯混合模型推导了旅行时间。文献[4]，[27]，[28]分别将高斯copula与图形套索、贝叶斯推断和截面溢出分析相结合，用于TTDE。参考文献[29]采用移动平均方法来估计威慑旅行时间，并采用分布估计算法来提供先前估计的置信区间。另一方面，放宽高斯假设的尝试导致了基于广泛学习技术的TTDE方法。例如，文献[30]设计了一种基于马尔可夫链的启发式算法来计算主干道连接的行程时间分

布。文献[31]提出了一种基于始发目的地对的Bayesian网络的TTDE算法，并利用期望最大化算法解决路径不确定性问题。参考文献[20]提出了一种基于深度学习的生成对抗网络，考虑出行信息最大化来估计短途和长途出行的分布。文献[32]提出了一个深度生成模型来估计包含多个道路连接的路线的行程时间分布，该模型比旅行前规划更适合于旅行后分析和路线恢复[15]。文献[33]采用了一种编码器-解码器深度神经网络，利用道路起伏、交通周期和路线轨迹的嵌入来估计行程时间。该模型的目标是提供确定性的旅行时间估计路线比道路连接，这仍然不是一个确切的匹配TTDE。

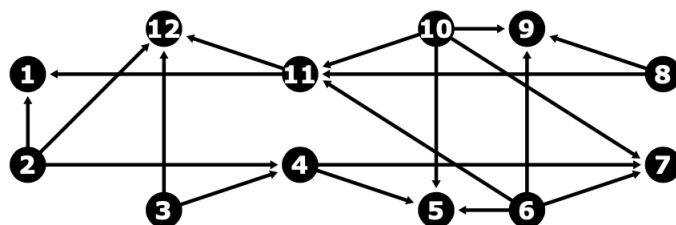
综上所述，人们提出了许多方法来正确估计旅行时间分布。然而，较少的研究集中于从数据不确定性中建模和学习，这是重要的，包括在此背景下的认知（系统）和自适应（measurement）的不确定性。此外，交通网络所调节的空间数据相关性没有被考虑在内，从而可能导致性能下降。为了解决这些问题，本文提出了一种新的基于深度学习的TTDE方法。

2.2 行程时间分布估计

利用稀疏实时GPS数据估计城市交通网络中道路链路的行程时间分布。将交通网络建模为一个有向图 $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$,其中 \mathcal{V} 是路段（路段）的集合， \mathcal{E} 表示路段在 \mathcal{V} 中的连通性，即：如果车辆可以从 $v_j \in \mathcal{V}$ 直接穿越到 $v_i \in \mathcal{V}$,则为 $(v_i, v_j) \in \mathcal{E}$ 。图1给出了这种图表示的一个说明性例子，它不同于通常的道路网络中交叉口由图节点表示的图表示，在利用链接相关性时更具有表现力[34]。TTDE的目标是在给定普遍存在和稀疏的实时GPS轨迹的情况下，获得所有链路 $v \in \mathcal{V}$ 在时隙 t 长度为 Δ 时的行程时间分布 $Pr_{v,t}(T)$ 。路径行程时间分布可以通过累加构成链路的行程时间分布得到[4]。



(a) A toy transportation network with 12 links.



(b) The corresponding directed graph.

Fig. 1. An example of the graph representation of transportation network.

TTDE的主要挑战是三重的，每一个都引起了数据挖掘领域的研究兴趣，但没有得到充分的集体解决，尤其是TTDE。首先，GPS轨迹中的认知和任意不确定性使得任何发现潜在旅程时间干扰的数据挖掘策略都具有挑战性。一般的数据驱动方法可能会受到干扰，取而代之的是学习噪声模式，从而产生较差的估计。开发一个量身定制的估计器来充分利用存在不确定性的大量历史数据是一项具有挑战性的任务。其次，连接链路的行程时间变化是高度相关的，因此估计者需要利用链路的行程时间如何依赖于其他[4], [7]。在交通网络中，这种相关性主要由邻接矩阵捕获。由于计算EF-FORT致力于探索结构或空间数据之间的关系，因此不使用这种拓扑信息可能会导致性能下降。第三，TTDE将旅行时间视为随机变量，并产生它们的估计，这与研究得很好的确定性旅行时间估计不同，后者发展了定点估计[8]。如何推断出准确可靠的概率分布仍然是数据挖掘和表示学习界的一个悬而未决的问题。事实上，没有合作应对这些挑战的解决办法，早先在其他问题上的努力也不能轻易地转移到TTDE上。考虑到这些挑战，我们利用多个构建块和深度学习的设计概念为TTDE提供了一个解决方案。

3. 方法论

为了克服以前的挑战，在本工作中，我们提出了一个新的解决方案，用于估计道路连接的全市旅行时间分布，名为DeepTTDE。该解决方案的灵感来自贝叶斯深度学习、图卷积和对抗生成模型的最新研究和进展。它试图将两者的优点结合起来，以应对TTDE以前的所有挑战。在本节中，我们首先概述了拟议的DeepTTDE，然后详细说明了其组成组件。最后，我们讨论了独特的DeepTTDE训练和推理方案。

3.1 概述

图2显示了拟议的DeepTTDE的总体框架。三个主要模块合作导出旅行时间分布，即地图匹配和旅行时间建模、旅行时间分布估计器和对抗训练模型。给定GPS轨迹的集合，我们首先将原始地理坐标转换为图 \mathcal{G} 上的路径，通过该路径计算每个轨迹的链路旅行时间是非常简单的。随后，利用轨迹链路旅行时间数据，通过独特的旅行时间建模算法构建多个旅行时间图。每个旅行时间图都是来自整个城市潜在旅行时间分布的单个样本，我们采用贝叶斯图自动编码器来模拟这些样本的分布。最后，引入了一个辅助对抗训练神经网络，以辅助前一个图自动编码器的参数训练，从而提高估计分布的准确性。

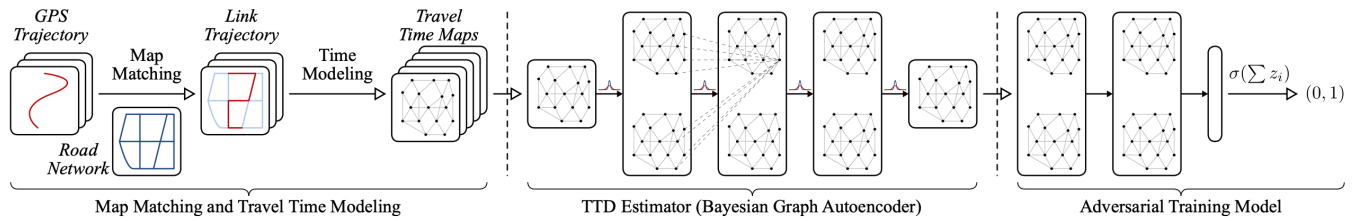


Fig. 2. Overview of DeepTTDE.

在深入探讨组成DeepTTDE的每个模块的细节之前，我们首先解释所使用技术背后的原理，特别是贝叶斯图自动编码器结构。所有模块都是围绕旅行时间分布估计器的要求设计的。虽然基于深度学习的技术在解决大量问题方面表现出了最先进的性能，但它们通常受到一系列限制，包括在不确定性学习方面的薄弱、非欧几里德表示、并且很容易被对抗样本[17], [35]破坏。人们可能会发现，第2.2节中引入的TTDE的主要挑战与一般深度学习技术的上述限制明显重叠。

DeepTTDE通过采用贝叶斯深度学习、几何深度学习 and 对抗性机器学习的设计原则来解决挑战和局限性。贝叶斯深度学习能够解释认识不确定性和自由不确定性，这两种不确定性在交通数据中都很普遍[17], [36]。特别是，该模型在每个神经网络参数上放置先验分布，然后通过贝叶斯推理近似后验分布。因此，模型不确定性通过参数分布的形状捕获，输入数据中的统计不确定性反映在后验中。其次，几何深度学习使DeepTTDE能够在计算过程中有效地表示图结构，减轻了试图从未明确显示道路连通性的交通数据中提取拓扑信息的深度学习模型所带来的巨大计算负担[35]。第三，贝叶斯模型本质上是概

率性的，它使一般深度学习模型能够表示不确定性和推断分布，这完美地满足了TTDE关于估计概率分布的要求[36]。利用条件概率作为神经网络参数，贝叶斯模型通过采样这些参数来估计分布，而不是添加输出噪声或设置多个输入场景。

3.2 地图匹配与行驶时间建模

在DeepTTDE中，地图匹配和旅行时间建模的主要目标是为后续旅行时间分布估计器提取潜在信息提供足够的行驶时间地图。顾名思义，它有两个主要步骤。地图匹配的目的是将GPS轨迹投影到道路网上。由于这个问题已经被广泛研究[37]–[39]，我们采用了一种**基于交互式投票的算法实现【37】**来在DeepTTDE中执行地图匹配。这也是相关研究中广泛采用的方式，一些例子见[4],[7]。

在开发地图匹配路径后，行驶时间建模尝试计算路径中每个链路的行驶时间。这可以通过假设车辆在连续轨迹点之间遵循恒定速度【4】来实现。具体来说，对于具有 k 个GPS位置的任意轨迹，让第一个轨迹点的映射位置为 p_1 ，最后一个轨迹点的映射位置为 p_k 。如果连续位置 p_i 和 p_{i+1} 都在同一链路上，则它们之间的采样时间差，用 $\Delta_{i \rightarrow i+1}$ 表示，将累积到相应链路的旅行时间。否则，“从”链接的行程时间增加 $\frac{d_{i \rightarrow i+1}}{d_{i \rightarrow i+1} + d_{i+1 \rightarrow i}} \Delta_{i \rightarrow i+1}$ ，剩余时间增加到“到”链接的行程时间，其中 $d_{i \rightarrow i+1}$ 和 $d_{i+1 \rightarrow i}$ 是从 p_i 和 p_{i+1} 到两个链接的交叉点的距离。

因此，可以基于计算的链路旅行时间构建多个旅行时间图。对于每个旅行时间图，我们首先根据图1创建一个有向运输图，其中每个节点嵌入了一个道路连接。每个节点 v_i 被初始化为一个元组 $X_i = (t_i, c_i, f_i^1, f_i^2, \dots)$ ，其中 t_i 是 v_i 的瞬时旅行时间， c_i 是嵌入1的采样时间，和 f_i^j 是相应道路链路的静态特征，例如限速、车道计数、链路长度等。然后，我们随机选择长度为15分钟的时间间隔，并聚合该周期内的所有轨迹。轨迹被排列并迭代地选择以包括在旅行时间地图中：如果其构成的链接都没有被其他链接覆盖，则包括一个轨迹。如果包括轨迹，则其所有构成链路的 t_i 和 c_i 值将用先前计算的相应旅行时间和采样时间更新。此过程重复，直到覆盖所有链接的预定义部分（案例研究中70%，以避免过度计算负担，值取自【4】），或检查所有轨迹。生成的旅行时间图是地面真相全市旅行时间分布的真实样本。值得注意的是，地图匹配和旅行时间建模预处理是基于轨迹的交通大数据分析的典型。我们打开了探索传统方法的替代方案的可能性，作为未来可能的研究方向。

3.3 旅行时间分布估计器

对于从原始GPS轨迹构建地面真实旅行时间样本的上一个模块，后续旅行时间分布估计器的目的是学习相对于时间的旅行时间分布动态。在DeepTTDE中，由于第3.1节中介绍的优点，提出了一个贝叶斯图自

动编码器来实现这一目标。为了描述所提出的深度学习模型的体系结构，我们首先简要介绍了它对非欧几里德数据处理的基本操作，即图卷积。

图卷积是图卷积网络【18】中最广泛认识的一种数值运算，它聚合了图中相邻节点的节点信息。当用于神经网络时，它的目的是从原始数据中提取局部图形特征。该操作继承了处理欧几里德空间数据的规范卷积神经网络（CNN）的卷积滤波器概念。这样的过滤器通过定义参数一致接受字段，对源数据执行邻域混合，从而导致本地信息共享[16]。虽然CNN中的接受字段通常是矩形的，但图卷积使用图连接性作为邻域混合的过滤器，以应对输入图数据的非欧几里德结构。给定图的每个节点上具有自环的增强邻接矩阵，用 \dot{A} 表示，图卷积采用特征 x 来计算输出 y ，如下【18】：

$$y = \sigma(\dot{D}^{-\frac{1}{2}} \dot{A} \dot{D}^{-\frac{1}{2}} x w)$$

其中， $\dot{D} = \text{diag}(\dot{A}1)$ 是对角度矩阵， w 是要通过反向传播学习的该图卷积的权重参数，函数 σ 是在本工作中一个非线性函数— $\text{ReLU}(x) = x^+ = \max(0, x)$ 。

(1)可以解释为对局部化频谱滤波器网络的一阶近似，其本身是对使用图傅里叶变换根据卷积定理在频域中执行卷积的频谱网络的局部近似[40],[41]。这将图卷积与欧几里德空间卷积联系起来，后者的优点也适用于前者。

请注意，（1）定义的图卷积是基于图拉普拉斯是对称的前提，这意味着一个无向图。这个概念与图1中交通网络是有向的说明相矛盾。解决这个问题的典型方法是将有向图放松为无向图，从而构造对称拉普拉斯矩阵。考虑到运输网络被建模为图G，我们在导出 $\dot{A} = A + I$ 时，使用边缘权重矩阵的对称矩阵作为邻接 $A = \{a_{ij}\}$ ，即，如果 $(v_i, v_j) \in \varepsilon$ 或 $(v_j, v_i) \in \varepsilon$ ，则 $a_{ij} = 1$ 。

在运输网络背景下，这一修改意味着，虽然车辆只能在 $(v_i, v_j) \in \varepsilon$ 的情况下从链路 v_i 行驶到 v_j ，但 v_i 的旅行时间分布也与 v_j 的旅行时间分布相关。总体而言，这种图卷积运算反映了交通网络中交通量的物理关系。在实践中，相邻链路的旅行时间显示出强烈的相关性。例如，如果链路拥塞，其合并流量c也可能阻塞。此特征可以由（1）中的邻接链路捕获，以便神经网络知道一个节点（道路链路）的交通状态可以传播到其相邻节点。

通过上述图卷积，人们已经可以构建一个深度神经网络来捕获链路间的旅行时间相关性并执行旅行时间估计。然而，一个重大的挑战仍然存在：无法制定不确定性估计。给定第3.2节中构建的确定性旅行时间图，（1）只能计算确定性输出。这是贝叶斯深度学习解决的。从概率论的角度来看，具有一层或多层图卷积的神经网络是概率模型 $Pr(y|x, \omega)$ 。其中 ω 是所有权重参数的集合，包括（1）中的 w 和其他神经网络权重。在没有贝叶斯推理的情况下，固定 x 值 ω 的训练遵循最大似然估计，训练集 $\mathcal{D} = \{y^{(i)}|x^{(i)}\}_i$ ，即， $\omega^* = \arg \max_{\omega} Pr(\mathcal{D}; \omega)$ 。如果采用正则化项来避免过拟合，则最优参数遵循最大后验概率【42】：

$$\begin{aligned}\omega^* &= \arg \max_{\omega} \log Pr(\mathcal{D}|\omega) + \log Pr(\omega) \\ &= \arg \max_{\omega} \log Pr(\mathcal{D}|\omega)\end{aligned}$$

如果我们考虑参数实际遵循嵌入在训练集中的后验分布，概率模型可以利用数据不确定性，并使用贝叶斯推理估计分布[17], [43], [44]。如果我们考虑参数实际遵循嵌入在训练集中的后向分布，概率模型可以利用数据不确定性，并使用贝叶斯推理估计分布[17], [43], [44]。

根据这一原则，原始图卷积可以相应地修改，用 $y = f_{\omega}(x)$ 表示，以使用零均值高斯作为参数空间 $Pr(\omega)$ 上的先验分布来应对旅行时间不确定性[45], [46]。请注意，这并不对旅行时间分布施加高斯假设，因此克服了现有研究的局限性。根据贝叶斯定理，后验分布可以通过：

$$Pr(\omega|\mathcal{D}) = \frac{Pr(\mathcal{D}|\omega)Pr(\omega)}{Pr(\mathcal{D})} = \frac{Pr(\omega) \prod_i l(y^{(i)}|f_{\omega}(x^{(i)}))}{\int Pr(\omega) \prod_i l(y^{(i)}|f_{\omega}(x^{(i)}))d\omega}$$

其中 $l(\cdot)$ 是可能性。尽管如此，边际概率 $Pr(D)$ 无法通过分析估计。为了克服这个问题，我们使用变分推理来通过由 ϕ [47]参数化的变分分布 $q_{\phi}(\omega)$ 来近似后验。通过最小化Kullback–Leibler (KL)相对于 ϕ 的发散，可以找到最接近 $Pr(\omega|\mathcal{D})$ 的最佳变分布，如下所示：

$$KL(q_{\phi}(\omega)||Pr(\omega|\mathcal{D})) = \int q_{\phi}(\omega) \log \frac{q_{\phi}(\omega)}{Pr(\omega|\mathcal{D})} d\omega$$

使用(3), (4)可重写为：

$$\begin{aligned}KL(q_{\phi}(\omega)||Pr(\omega|\mathcal{D})) &= \int q_{\phi}(\omega) \log \frac{q_{\phi}(\omega)}{Pr(\omega|\mathcal{D})} d\omega \\ &\quad - \sum_i \int q_{\phi}(\omega) \log l(y^{(i)}|f_{\omega}(x^{(i)})) d\omega \\ &\quad + \log \int Pr(\omega) \prod_i l(y^{(i)}|f_{\omega}(x^{(i)})) d\omega\end{aligned}$$

为了正确地最小化这种KL发散，通常使用数据子采样【48】和重新参数化[47][46]。其关键思想是减少涉及D的计算中的样本总数，以减轻计算负担。此外，神经网络参数通过确定性可微变换 $\varphi = \mu + \sigma\varepsilon$ 重新参数化，其中 $\varphi \sim \mathcal{N}(\mu, \sigma\sigma^T)$ 和 $\varepsilon \sim \mathcal{N}(0, I)$ [47]。

随后，（5）可以通过随机优化器优化。得到的最优 μ^* 和 σ^* 将最优值反映到原始 $KL(q_\varphi(\omega)||Pr(\omega|\mathcal{D}))$ ，因此 $q_\varphi^*(\omega)$ 是 $Pr(\omega|\mathcal{D})$ 。估计的分布可以近似为：

$$Pr(y|x, \mathcal{D}) = \int Pr(y|x, \omega) q_{\phi^*}(\omega) d\omega = q_{\phi^*}(y|x)$$

我们请读者参考[47],[48]了解相关的理论分析。

总之，贝叶斯推理用参数化的高斯先验 $\mathcal{N}(\mu, \sigma\sigma^T)$ 取代每个神经网络参数（例如，w in (1)）。在通过KL发散（5）训练新的参数 μ 和 σ 后，我们可以采样K ϵ 值，从先验中绘制固定值参数。由于每个参数样本都可以生成一个确定性推断，K个推断自然形成了后验分布--在我们的上下文中的旅行时间分布。

在贝叶斯图卷积的帮助下，我们构造了一个用于旅行时间分布估计的贝叶斯图自动编码器。图2示出了所提出的自动编码器的分层体系结构，该体系结构以一个旅行时间图作为输入x并开发K个输出y_k，每个输出y_k代表一个可能的完整的城市范围旅行时间图。

在此过程开始时，原始节点特征 $\langle t_i, c_i, f_i^1, f_i^2, \dots \rangle$ 用z-score归一化标准化，缺失的旅行时间条目用零赋值。生成的零填充图特征矩阵输入到自动编码器中，自动编码器采用由(1)和（6）共同定义的三个连续贝叶斯图卷积，用于特征和不确定性学习。在每个卷积中，节点特征的数量分别扩展/压缩到128、256和128。最后一个卷积操作附加了一个Tanh激活的贝叶斯图卷积，输出特征的数量为1，以便重建旅行时间图，并生成了缺失值。

3.4 对抗性训练模式

在前面的小节中，我们提出了一个贝叶斯图自动编码器作为旅行时间分布估计器。估计器接受预匹配和旅行时间建模模块生成的旅行时间地图，从原始数据中学习，估计全市范围内的旅行时间分布。然而，对自动编码器的训练结果表明，其神经网络参数的优化是不合格的。

为了加速模型训练过程，在DeepTTDE中，我们使用了一个辅助对抗训练神经网络（也称为鉴别器[49]），在KL发散训练目标中添加了一个时间图真实性项。对抗训练的动机类似于将鉴别器纳入生成对抗网络。虽然现有的贝叶斯图自动编码器能够预测行程时间分布，但它可能需要更大的模型容量来根据潜在的交通领域信息进行估计和正则化输出，以帮助模型的性能。相反，我们使用对抗性训练神经网络来确定生成的旅行时间分布是否与从地面真实情况中提取的交通动力学一致。第4.4节将介绍的案例研究也从经验上揭示了其有效性。

特别地，建立了具有三层图卷积的鉴别器神经网络，每层分别具有64、16和1个输出特征。神经网络接受由前面的贝叶斯图自动编码器导出的任何重构旅行时间图 y_k ，并输出特征图 z_k 。最后，用求和方法对特征进行聚合，并用sigmoid函数将聚合的特征映射到 (0,1)。该输出反映鉴别器是否认为相应的 y_k 是“真实的”。

该鉴别器的目标是帮助估计器生成与真实对应物不可区分的旅行时间图，从而通过将这种额外的真实性信息添加到适应度目标中来加速估计器的训练。估计器和判别器形成一个两人极小极大对策，目标如下：

$$\min_{\phi} \max_{\Phi} V = \mathbb{E}_{Pr(x)} [\log(1 - h_{\Phi}(f_{\omega}(x)))] \\ + \mathbb{E}_{Pr(x)} [\log h_{\Phi}(x)] + KL(q_{\phi}(\omega) || Pr(\omega) | \mathcal{D})$$

其中 $h_{\Phi}(z)$ 描述由 Φ 参数化的鉴别器。在训练开始时，随机初始化的 Φ 使鉴别器很容易拒绝生成的旅行时映射 $f_{\omega}(x)$ ，从而得到较大的 $\mathcal{E}_{Pr(x)} [\log(1 - h_{\Phi}(f_{\omega}(x)))]$ 值。

然后，训练过程试图调整 Φ 以生成逼真的旅行时间图，从而折衷鉴别器并与真实的旅行时间图相似。同时，还不断优化 Φ ，以提高判别器识别生成地图的能力。重复这个过程，直到估计器参数足够好，可以通过学习原始数据中的潜在旅行时间分布来绕过鉴别器。有兴趣的读者可以转到[49]–[51]来了解对抗训练机制的更详细的分析。

3.5 DeepTTDE训练与推理

在前面的小节中，我们介绍了DeepTTDE中的三个MA-JOR模块。在采用该模型进行在线行程时间分布推断之前，首先需要调整神经网络参数值，以充分捕捉潜在的原始数据特征。虽然贝叶斯图自动编码器可以用任意随机梯度下降变量训练 (1) 和(5)，但输入旅行时图如何作为神经网络的输入和目标输出仍不清楚。因此，我们提出了一个为DeepTTDE量身定制的训练方案，以考虑其旅行时间建模和贝叶斯深度学习的多样本性质。

在给定原始GPS轨迹集合的情况下，地图匹配和旅行时建模模块能够构建多个旅行时地图，每个旅行时地图都包含任意15min时间间隔的确定性不完全全市旅行时数据（第3.2节）。对于由道路链路数据 $x_j^{(i)}$ 成的任意旅行时间图 $x^{(i)}$ ，我们分别使用集合 $\mathcal{V}^{(i)+}$ 和 $\mathcal{V}^{(i)-}$ 来聚合旅行时间图中被任何原始轨迹覆盖和未被任何原始轨迹覆盖的链路。我们通过手动从 $\mathcal{V}^{(i)+}$ 中删除一些可用的链接数据来增强映射，并构造多个新的训练映射。算法1给出了用于增强过程的伪码。

特别地， M 个随机子集 $\mathcal{V}^{(i)*,m} \notin \mathcal{V}^{(i)+}$ 是独立生成的。然后我们复制 $x^{(i)}$ 来构造 M 个相同的 $x^{(i),m}$ 映射。在每个 $x^{(i),m}$ 中， $\mathcal{V}^{(i)*,m}$ 中链接的可用特征被移除。因此，在 N 个原始地图的基础上，开发了 $N \times M$ 的扩展行程时间地图。后续行程时间分布估计器的目标是基于部分 $x^{(i),m}$ 恢复 $x^{(i)}$ 。将两个地图聚合形成训练数据集 \mathcal{X} 。

对贝叶斯图自动编码器和对抗性训练模型中的神经网络参数进行迭代调整。在每次迭代中，自动编码器取一个 $x^{(i),m}$ 并推断 K 个不同的 $\dot{x}^{(i),m}$ 来估计 $x^{(i)}$ ，用 $\dot{x}_k^{(i),m}$ 表示。考虑到(7)，AUTOEN-K7 编码器的目标函数设计如下：

$$\min_{\phi} \mathbb{P}r(x) [\log(1 - h_{\Phi}(\dot{x}_k^{(i),m}))] + KL(q_{\phi}(\omega) || Pr(\omega | \mathcal{D}))$$

其中，在计算KL散度时， $\dot{x}^{(i),m}$ 与 $y^{(i)}$ 相同，即 (5)。随后， $x^{(i)}$ 与 $\dot{x}_k^{(i),m}$ 都被输入到鉴别器中，以求出同样从 (7) 导出的下列目标函数：

$$\min_{\Phi} \mathbb{E}_{Pr(x)} [\log(1 - h_{\Phi}(\dot{x}_k^{(i),m}))] + \mathbb{E}_{Pr(x)} [\log h_{\Phi}(x^{(i)})]$$

对于每32次迭代，自动编码器和鉴别器的计算损失值被用来使用Adam优化器[52]调整所有神经网络参数。

通过训练好的DeepTTDE参数，我们可以使用贝叶斯图自动编码器来推断实时的全市出行时间分布。在使用第3.2节中介绍的可用信息构建部分地图后，DeepTTDE使用自动编码器生成 P 个完整的行程时间地图以进行预测。请注意，此超参数与之前的 K 不同，后者仅在训练期间用作网络参数样本数。对于运输网络中的每个道路链路，我们认为在所有 P 图中的每个确定性行程时间推断都是来自旅行时间分布的随机样本，它将近似于由数据给出的地面真实性。在评估推断分布时，我们使用两个离散分布

$$Q = \sum_r Q(r) \text{ 和 } R = \sum_r R(r) :$$

$$KL(Q || R) = \sum_r Q(r) \log \frac{Q(r)}{R(r)},$$

$$Hellonger(Q, R) = \frac{1}{\sqrt{2}} (\sum_r (Q(r)^{\frac{1}{2}} - R(r)^{\frac{1}{2}})^2)^{\frac{1}{2}}$$

其中 r 是直方图箱的索引。根据文[4]中的分析，在计算这两个公式时，我们将直方图箱的默认数目设为11。

4. 案例研究

在本节中，我们对一个真实世界的数据集进行了一系列全面的案例研究，以评估DeepTTDE的性能。具体来说，我们首先简要介绍了所使用的数据集和仿真配置。然后，对估计的旅行时间分布的精确性进行了比较研究，并考察了GPS轨迹稀疏性的影响。随后进行了烧蚀试验，验证了DeepTTDE组成模块的必要性。最后，通过一个超参数测试说明了超参数选择对模型性能的影响。

4.1 数据和配置

本文采用滴滴出行提供的KDD CUP2020数据集2进行深入的案例研究。特别是，该数据集包括2016年11月1日至2016年11月30日期间中国成都拼车车辆的GPS轨迹。它提供了大约1097亿个GPS记录，每隔2秒到4秒的采样间隔，从穿过城市的车辆中采样。从OpenStreetMap中获取交通网络拓扑，并按照3.2节的介绍进行地图匹配。

为了建立地面真实交通速度分布，我们首先将整个数据集拆分为2880个时间段，每个时间段包含一天中15分钟的所有车辆轨迹。如果一条轨迹跨越多个时间段，则该轨迹包含在所有时间段中。随后，我们采用了第3.2节中介绍的标准行程时间建模技术，以确定每个路段的每条轨道行程时间。由于多条轨迹可能在任意时间段内通过同一条道路连接，因此所有相应的行程时间值都将被聚合，并共同作为基本事实。在此期间内未被任何轨迹覆盖的特定链接被视为 $\mathcal{V}^{(i)-}$ 的一个元素，不参与绩效评估。我们将提出的DeepTTDE的TTDE结果与这些地面真实交通速度分布进行了比较。

当采用拟议的TTDE方法时，实时车辆轨迹可能只提供一小部分道路的瞬时行驶时间。为了模拟这种真实情况，我们使用第3.2节中提出的方法，为每个时间段构建 $N=75$ 个基本行程时间图，绘制总共216000个时间图。对于每个基本行程时间图，根据第3.5节，我们通过保留 \mathcal{V} 中保留 $\beta\%$ 链路的行程时间数据，随机构建 $M = 5$ 个图。因此，有1 080 000旅行时间地图可用于TTDE培训和评估。对于交叉验证，我们随机选择3600个时间段（540000个样本）进行培训，1800个时间段（270000个样本）进行提前停止验证，其余1800个时间段（270000个样本）进行测试。在每个神经网络训练阶段后对验证样本进行评估，如果该验证集的损失在过去三个阶段没有减少，则整个训练终止。在评估估计行程分布的质量时，计算 $\mathcal{V}^{(i)+}$ 中所有道路连接的KL散度和Hellinger距离（下表中简称Hlg），并根据地面真实值进行平均。除非另有说明，否则 $\beta = 30\%$ ， $K = 32$ ， $P = 512$ 。所有模拟都在配备两个Intel Xeon E5 CPU和128GB RAM的计算服务器上执行。nVidia RTX 2080Ti GPU用于神经网络计算加速。

4.2 行程时间分布估计

我们首先研究了估计行程时间分布与地面真实值之间的偏差，这是评估TTDE方法的最重要指标之一。具体而言，以下方法作为本实证研究的基线：