



(12) 发明专利

(10) 授权公告号 CN 111613054 B

(45) 授权公告日 2021.07.06

(21) 申请号 202010377173.0

G06N 3/04 (2006.01)

(22) 申请日 2020.05.07

审查员 杨潘云

(65) 同一申请的已公布的文献号

申请公布号 CN 111613054 A

(43) 申请公布日 2020.09.01

(73) 专利权人 浙江大学

地址 310058 浙江省杭州市西湖区余杭塘路866号

(72) 发明人 赵春晖 崔紫强

(74) 专利代理机构 杭州求是专利事务所有限公司 33200

代理人 邱启旺

(51) Int. Cl.

G08G 1/01 (2006.01)

G06K 9/62 (2006.01)

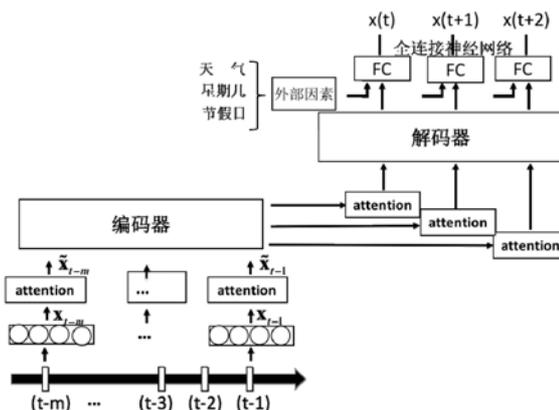
权利要求书3页 说明书8页 附图2页

(54) 发明名称

一种协同考虑时空相关性及贡献差异的多步交通速度预测方法

(57) 摘要

本发明公开了一种协同考虑时空相关性及贡献差异的多步交通速度预测方法,该方法使用了基于循环神经网络的编码-解码的网络架构来充分表达交通速度的时序特性。在编码部分,对相关路段速度值所构成的输入向量引入第一阶段的注意力机制,使其自适应学习不同相关路段在不同时刻的权重贡献大小;在解码部分引入第二阶段的注意力机制,来自适应学习不同历史时刻对当前预测时刻的权重贡献大小。同时,考虑到外部因素的影响,将解码器的输出与外部因素特征一同输入全连接神经网络得到最终输出。本方法可以更细粒度地、差异化地刻画交通数据的时空相关特性,且能进行多步交通速度预测,为交通速度预测方法的研究指明了新的方向。



1. 一种协同考虑时空相关性和贡献差异的多步交通速度预测方法,其特征在于,该方法包括以下步骤:

步骤1:建模数据预处理:对原始速度数据根据路段和时间段进行均值化处理,对天气、节假日外部因素数据进行类别分类;

步骤2:确定模型的输入变量和外部离散特征向量;

(2.1) 设c为待预测路段的编号,根据地理位置,按照与预测路段的交叉和临近的优先原则选取前k个路段,作为与路段c相关的路段,这些相关路段的速度将作为变量用于预测模型的建立;

(2.2) m为建模所需要的历史时间窗口的大小;设 t_p 为第一个预测时刻,则模型预测该时刻所需的输入包括该时刻之前的m个时刻,对于历史时刻i, $i = t_p - m, t_p - m + 1, \dots, t_p - 1$,其对应所需的速度输入向量记为 $\mathbf{x}_i = [\mathbf{x}_i^1, \mathbf{x}_i^2, \dots, \mathbf{x}_i^k]$,其中k为相关路段的个数;此外还要输入预测时刻所对应的外部离散特征向量 d_t ,包括天气和日期数据;

步骤3:模型的建立和训练;

(3.1) 确定相关路段的贡献程度,得到新的输入向量;

编码器网络采用LSTM作为基础单元,历史时间步长为m,与步骤2中的历史时间窗口一致;对于历史时刻i所对应的原始输入向量x,首先计算每个相关路段的得分,如下式所示:

$$e_i^j = \mathbf{v}_e^T \tanh(\mathbf{W}_e[\mathbf{h}_{i-1}; \mathbf{s}_{i-1}] + \mathbf{U}_e \mathbf{x}^j) \quad (1)$$

其中, $j = 1, 2, \dots, k$, k为相关路段数量, \mathbf{h}_{i-1} 和 \mathbf{s}_{i-1} 分别为Encoder中LSTM单元当前时刻的前一时刻的f维隐藏层状态和细胞状态, $\mathbf{x}^j = [X_{t_p-m}^j, X_{t_p-m+1}^j, \dots, X_{t_p-1}^j]^T$ 是所有输入时刻的第j个相关路段速度值所组成的向量, $\mathbf{v}_e \in \mathbb{R}^m$, $\mathbf{W}_e \in \mathbb{R}^{m \times 2f}$, $\mathbf{U}_e \in \mathbb{R}^{m \times m}$ 是网络需要学习的参数;

使用类SoftMax对得分进行归一化处理,使其转化成类概率分布的权重系数:

$$\alpha_i^j = \frac{\exp(e_i^j)}{\sum_{g=1}^k \exp(e_i^g)} \quad (2)$$

然后将对应路段的速度值与对应的权重相乘,得到该时刻新的输入向量:

$$\hat{\mathbf{x}}_i = [\alpha_i^1 \mathbf{x}_i^1, \alpha_i^2 \mathbf{x}_i^2, \dots, \alpha_i^k \mathbf{x}_i^k] \quad (3)$$

(3.2) 对编码器网络的LSTM单元进行计算和更新;

将新的输入向量输入到LSTM单元,对于历史时刻i的LSTM单元,其计算和更新方式如下,首先计算遗忘门 f_i 、输入门 a_i 、输出门 o_i :

$$\mathbf{f}_i = \sigma(\mathbf{W}_f[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i] + \mathbf{b}_f) \quad (4)$$

$$\mathbf{a}_i = \sigma(\mathbf{W}_a[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i] + \mathbf{b}_a) \quad (5)$$

$$\mathbf{o}_i = \sigma(\mathbf{W}_o[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i] + \mathbf{b}_o) \quad (6)$$

其中, $[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i]$ 是前一个时刻的隐藏层状态 \mathbf{h}_{i-1} 与当前时刻的输入 $\hat{\mathbf{x}}_i$ 在向量维度上的拼接; σ 表示sigmoid函数, $\mathbf{W}_f, \mathbf{W}_a, \mathbf{W}_o, \mathbf{b}_f, \mathbf{b}_a, \mathbf{b}_o$ 都是网络要学习的参数;

然后计算更新当前的细胞状态：

$$\mathbf{s}_i = \mathbf{f}_i \odot \mathbf{s}_{i-1} + \mathbf{a}_i \odot \tanh(\mathbf{W}_s[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i] + \mathbf{b}_s) \quad (7)$$

其中, $\mathbf{W}_s, \mathbf{b}_s$ 是待学习的参数, 符号 \odot 表示对应元素相乘;

计算隐藏层状态如下:

$$\mathbf{h}_i = \mathbf{o}_i \odot \tanh(\mathbf{s}_i) \quad (8)$$

(3.3) 确定不同历史时刻的贡献程度, 得到新的输入向量;

解码器网络也是由LSTM单元构成; 对于当前的预测时刻 t ($t = t_p, t_p+1, t_p+2, t_p$ 为第一个预测时刻), 首先我们利用解码器中当前时刻的前一时刻的隐藏层状态和细胞状态以及编码器中的每个隐藏层状态, 生成不同历史时刻对当前预测时刻的贡献分数, 如下:

$$l_i^t = \mathbf{v}_d^T \tanh(\mathbf{W}_d[\mathbf{h}'_{t-1}; \mathbf{s}'_{t-1}] + \mathbf{U}_d \mathbf{h}_i), \quad i = t_p - m, t_p - m + 1, \dots, t_p - 1 \quad (9)$$

其中, \mathbf{h}'_{t-1} 和 \mathbf{s}'_{t-1} 分别是预测时刻的前一时刻的解码器LSTM单元的隐藏层状态和细胞状态, \mathbf{h}_i 是历史时刻 i 的编码器LSTM单元的隐藏层状态, $\mathbf{v}_d, \mathbf{W}_d, \mathbf{U}_d$ 是要学习的参数;

采用类SoftMax的方式对得分进行归一化, 得到历史时刻 i 隐藏层的权重系数:

$$\beta_i^t = \frac{\exp(l_i^t)}{\sum_{j=1}^m \exp(l_j^t)} \quad (10)$$

然后将权重系数与隐藏层状态对应相乘再求和, 计算出预测时刻 t 所对应的编码器中所有时刻的隐藏层状态的加权和 \mathbf{c}_t :

$$\mathbf{c}_t = \sum_{i=1}^m \beta_i^t \mathbf{h}_i \quad (11)$$

将 \mathbf{c}_t 与解码器中上一时刻的输出 \mathbf{y}_{t-1} 在维度上进行拼接, 作为解码器中当前时刻LSTM单元的输入向量:

$$\mathbf{x}_t = [\mathbf{c}_t; \mathbf{y}_{t-1}] \quad (12)$$

(3.4) 采用步骤(3.2)的方法对解码器网络的LSTM单元进行计算和更新;

(3.5) 构建考虑外部因素的全连接神经网络;

在预测 t 时刻的交通速度时, 我们将解码器的输出和由外部因素组成的离散特征拼接起来, 并将它们输入到一个全连接神经网络, 得到最终的输出 $\hat{\mathbf{y}}_t$,

$$\hat{\mathbf{y}}_t = \sigma(\mathbf{W}_n[\mathbf{y}_t; \mathbf{d}_t] + \mathbf{b}_n) \quad (13)$$

其中, \mathbf{y}_t 是 t 时刻解码器的输出值, \mathbf{d}_t 是 t 时刻的外部离散特征向量, 包括天气和日期, \mathbf{W}_n 和 \mathbf{b}_n 是待学习的参数;

(3.6) 模型的训练;

采用Adagrad梯度下降法对步骤(3.1)至(3.5)中的参数进行反向传播更新, 获得最终参数得到训练好的模型;

步骤4: 模型在线预测;

对测试样本, 首先按照步骤1到步骤2, 对测试样本的格式进行转化, 然后将测试样本直接输入到步骤3所训练好的模型中, 模型输出最终的预测结果, 完成预测。

2. 根据权利要求1所述的方法, 其特征在于, 所述步骤1具体为: 对于原始速度数据, 将

获取的车辆速度数据按照不同路段分类;对需要分析的每一个路段,以一小时为时间间隔,将每天划分成24个时间段;对于第 z 个时间间隔, $z=0,1,2,\dots,23$,用该时间间隔内该路段上速度的平均值来表示第 z 个时间间隔的速度,每一个时刻对应一个固定的速度值;则路段 u 在 p 天内获得的速度向量可以表示为 $v_u^T(1 \times 24p)$,其中, $u=1,2,\dots,L$,为全部路段的代号, L 为路段总数;

对于天气,按照降水量大小将其分类,分别用不同标签表示;将日期类型分为三类:工作日、周末、节假日,分别用不同标签表示;获取上述速度变量对应时刻的天气和日期类型,路段 u 在 p 天内的天气数据表示为 $w_u^T(1 \times 24p)$,日期类别数据表示为 $d_u^T(1 \times 24p)$ 。

一种协同考虑时空相关性及贡献差异的多步交通速度预测方法

技术领域

[0001] 本发明属于交通时序数据分析和预测研究领域,特别是涉及一种基于时空相关性及外部因素特征和具有双阶段注意力机制的序列网络的交通速度预测方法。

背景技术

[0002] 随着机动车数量的增长,交通拥堵问题日益严重,给人们的出行带来了诸多不便,也对环境造成了污染,甚至威胁到了人们的生命安全。交通拥堵带来的直接问题是出行时间增加。另一方面,机动车数量的增加所带来的交通拥堵使得行驶车辆启动、停止次数增加、路上等待时间增加,导致了汽车尾气排放量的增加,造成大气的污染,严重危害着人类健康。与此同时,交通拥堵也是交通事故频发的一大诱因。据统计,20世纪全世界因交通事故而死亡的人数达2585万。

[0003] 早期常用的方法以时间序列方法为主,时间序列模型通过对历史时间序列进行数学建模,然后将其应用于预测未来的时间序列,如ARIMA方法。近二十年来,很多研究将包括支持向量机等统计学习方法应用于时序问题建模。近几年,随着人工智能的发展,一些深度学习方法,如循环神经网络、卷积神经网络等以及宽度学习方法也被应用在交通预测中。传统的时间序列模型拟合能力弱,切缺乏非线性表达能力,难以模拟道路的真实状况,且没有考虑到空间相关性。基于宽度学习的方法具有建模速度快的优势,基于循环神经网络的方法和基于卷积神经网络的方法分别可以有效地表达交通速度的时间相关性和空间相关性。但这些方法都没有区分不同时刻对于预测时刻的贡献度以及不同相关路段对于预测路段的贡献度,且大多没有考虑天气、节假日等外部因素的影响。基于此,本发明所提出的方法,既考虑了相关路段的影响,又考虑了历史时刻的影响,且融入了天气和节假日等外部因素的影响。此外,本发明还通过注意力机制表达了不同相关路段和不同历史时刻的贡献差异性,体现了较以往方法更加细粒度的时空相关关系。到目前为止,尚未见到与本发明相关的研究报道。

发明内容

[0004] 本发明的目的在于针对现有交通速度预测方法的不足,提供一种考虑时空相关性和贡献差异的多步交通速度预测方法。

[0005] 本发明的目的是通过以下技术方案来实现的:一种考虑时空相关性和贡献差异的多步交通速度预测方法,该方法包括以下步骤:

[0006] 步骤1:建模数据预处理:对原始速度数据根据路段和时间段进行均值化处理,对天气、节假日等外部因素数据进行类别分类。

[0007] 步骤2:确定模型的输入变量和外部离散特征向量。

[0008] (2.1) 设 c 为待预测路段的编号,根据地理位置,按照与预测路段的交叉和临近的优先原则选取前 k 个路段,作为与路段 c 相关的路段,这些相关路段的速度将作为变量用于

预测模型的建立。

[0009] (2.2) m 为建模所需要的历史时间窗口的大小。设 t_p 为第一个预测时刻,则模型预测该时刻所需的输入包括该时刻之前的 m 个时刻,对于历史时刻 $i, i = t_p - m, t_p - m + 1, \dots, t_p - 1$,其对应所需的速度输入向量记为 $\mathbf{x}_i = [\mathbf{x}_i^1, \mathbf{x}_i^2, \dots, \mathbf{x}_i^k]$,其中 k 为相关路段的个数。此外还要输入预测时刻所对应的外部离散特征向量 \mathbf{d}_t ,包括天气和日期数据。

[0010] 步骤3:模型的建立和训练。

[0011] (3.1) 确定相关路段的贡献程度,得到新的输入向量。

[0012] 编码器网络采用LSTM作为基础单元,历史时间步长为 m ,与步骤2中的历史时间窗口一致。对于历史时刻 i 所对应的原始输入向量 \mathbf{x} ,首先计算每个相关路段的得分,如下式所示:

$$[0013] \quad e_i^j = \mathbf{v}_e^T \tanh(\mathbf{W}_e[\mathbf{h}_{i-1}; \mathbf{s}_{i-1}] + \mathbf{U}_e \mathbf{x}^j) \quad (1)$$

[0014] 其中, $j=1, 2, \dots, k$, k 为相关路段数量, \mathbf{h}_{i-1} 和 \mathbf{s}_{i-1} 分别为Encoder中LSTM单元当前时刻的前一时刻的 f 维隐藏层状态和细胞状态, $\mathbf{x}^j = [\mathbf{x}_{t_p-m}^j, \mathbf{x}_{t_p-m+1}^j, \dots, \mathbf{x}_{t_p-1}^j]^T$ 是所有输入时刻的第 j 个相关路段速度值所组成的向量, $\mathbf{v}_e \in \mathbb{R}^m, \mathbf{W}_e \in \mathbb{R}^{m \times 2f}, \mathbf{U}_e \in \mathbb{R}^{m \times m}$ 是网络需要学习的参数。

[0015] 使用类SoftMax对得分进行归一化处理,使其转化成类概率分布的权重系数:

$$[0016] \quad \alpha_i^j = \frac{\exp(e_i^j)}{\sum_{g=1}^k \exp(e_i^g)} \quad (2)$$

[0017] 然后将对应路段的速度值与对应的权重相乘,得到该时刻新的输入向量:

$$[0018] \quad \hat{\mathbf{x}}_i = [\alpha_i^1 \mathbf{x}_i^1, \alpha_i^2 \mathbf{x}_i^2, \dots, \alpha_i^k \mathbf{x}_i^k] \quad (3)$$

[0019] (3.2) 对编码器网络的LSTM单元进行计算和更新。

[0020] 将新的输入向量输入到LSTM单元,对于历史时刻 i 的LSTM单元,其计算和更新方式如下,首先计算遗忘门 \mathbf{f}_i 、输入门 \mathbf{a}_i 、输出门 \mathbf{o}_i :

$$[0021] \quad \mathbf{f}_i = \sigma(\mathbf{W}_f[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i] + \mathbf{b}_f) \quad (4)$$

$$[0022] \quad \mathbf{a}_i = \sigma(\mathbf{W}_a[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i] + \mathbf{b}_a) \quad (5)$$

$$[0023] \quad \mathbf{o}_i = \sigma(\mathbf{W}_o[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i] + \mathbf{b}_o) \quad (6)$$

[0024] 其中, $[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i]$ 是前一个时刻的隐藏层状态 \mathbf{h}_{i-1} 与当前时刻的输入 $\hat{\mathbf{x}}_i$ 在向量维度上的拼接。 σ 表示sigmoid函数, $\mathbf{W}_f, \mathbf{W}_a, \mathbf{W}_o, \mathbf{b}_f, \mathbf{b}_a, \mathbf{b}_o$ 都是网络要学习的参数。

[0025] 然后计算更新当前的细胞状态:

$$[0026] \quad \mathbf{s}_i = \mathbf{f}_i \odot \mathbf{s}_{i-1} + \mathbf{a}_i \odot \tanh(\mathbf{W}_s[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i] + \mathbf{b}_s) \quad (7)$$

[0027] 其中, $\mathbf{W}_s, \mathbf{b}_s$ 是待学习的参数,符号 \odot 表示对应元素相乘。

[0028] 计算隐藏层状态如下:

$$[0029] \quad \mathbf{h}_i = \mathbf{o}_i \odot \tanh(\mathbf{s}_i) \quad (8)$$

[0030] (3.3) 确定不同历史时刻的贡献程度,得到新的输入向量。

[0031] 解码器网络也是由LSTM单元构成。对于当前的预测时刻 t ($t=t_p, t_p+1, t_p+2, \dots, t_p$ 为第一个预测时刻), 首先我们利用解码器中当前时刻的前一时刻的隐藏层状态和细胞状态以及编码器中的每个隐藏层状态, 生成不同历史时刻对当前预测时刻的贡献分数, 如下:

$$[0032] \quad l_i^t = \mathbf{v}_d^T \tanh(\mathbf{W}_d[\mathbf{h}_{t-1}^i; \mathbf{s}_{t-1}^i] + \mathbf{U}_d \mathbf{h}_i), \quad i = t_p - m, t_p - m + 1, \dots, t_p - 1 \quad (9)$$

[0033] 其中, \mathbf{h}_{t-1}^i 和 \mathbf{s}_{t-1}^i 分别是预测时刻的前一时刻的解码器LSTM单元的隐藏层状态和细胞状态, \mathbf{h}_i 是历史时刻 i 的编码器LSTM单元的隐藏层状态, $\mathbf{v}_d, \mathbf{W}_d, \mathbf{U}_d$ 是要学习的参数。

[0034] 采用类SoftMax的方式对得分进行归一化, 得到历史时刻 i 隐藏层的权重系数:

$$[0035] \quad \beta_i^t = \frac{\exp(l_i^t)}{\sum_{j=1}^m \exp(l_j^t)} \quad (10)$$

[0036] 然后将权重系数与隐藏层状态对应相乘再求和, 计算出预测时刻 t 所对应的编码器中所有时刻的隐藏层状态的加权和 \mathbf{c}_t :

$$[0037] \quad \mathbf{c}_t = \sum_{i=1}^m \beta_i^t \mathbf{h}_i \quad (11)$$

[0038] 将 \mathbf{c}_t 与解码器中上一时刻的输出 \mathbf{y}_{t-1} 在维度上进行拼接, 作为解码器中当前时刻LSTM单元的输入向量:

$$[0039] \quad \mathbf{x}_t = [\mathbf{c}_t; \mathbf{y}_{t-1}] \quad (12)$$

[0040] (3.4) 采用步骤(3.2)的方法对解码器网络的LSTM单元进行计算和更新。

[0041] (3.5) 构建考虑外部因素的全连接神经网络。

[0042] 在预测 t 时刻的交通速度时, 我们将解码器的输出和由外部因素组成的离散特征拼接起来, 并将它们输入到一个全连接神经网络, 得到最终的输出 $\hat{\mathbf{y}}_t$,

$$[0043] \quad \hat{\mathbf{y}}_t = \sigma(\mathbf{W}_n[\mathbf{y}_t; \mathbf{d}_t] + \mathbf{b}_n) \quad (13)$$

[0044] 其中, \mathbf{y}_t 是 t 时刻解码器的输出值, \mathbf{d}_t 是 t 时刻的外部离散特征向量, 包括天气和日期, \mathbf{W}_n 和 \mathbf{b}_n 是待学习的参数。

[0045] (3.6) 模型的训练。

[0046] 采用Adagrad梯度下降法对步骤(3.1)至(3.5)中的参数进行反向传播更新, 获得最终参数得到训练好的模型。

[0047] 步骤4: 模型在线预测。

[0048] 对测试样本, 首先按照步骤1到步骤2, 对测试样本的格式进行转化, 然后将测试样本直接输入到步骤3所训练好的模型中, 模型输出最终的预测结果, 完成预测。

[0049] 进一步地, 所述步骤1具体为: 对于原始速度数据, 将获取的车辆速度数据按照不同路段分类。对需要分析的每一个路段, 以一小时为时间间隔, 将每天划分成24个时间段。对于第 z 个时间间隔, $z=0, 1, 2, \dots, 23$, 用该时间间隔内该路段上速度的平均值来表示第 z 个时间间隔的速度, 每一个时刻对应一个固定的速度值。则路段 u 在 p 天内获得的速度向量可以表示为 $\mathbf{v}_u^T (1 \times 24p)$, 其中, $u=1, 2, \dots, L$, 为全部路段的代号, L 为路段总数。

[0050] 对于天气, 按照降水量大小将其分类, 分别用不同标签表示; 将日期类型分为三类: 工作日、周末、节假日, 分别用不同标签表示; 获取上述速度变量对应时刻的天气和日期类型, 路段 u 在 p 天内的天气数据表示为 $\mathbf{w}_u^T (1 \times 24p)$, 日期类别数据表示为 $\mathbf{d}_u^T (1 \times 24p)$ 。

[0051] 与现有技术相比,本发明的有益效果是:本方法从实际交通数据的特点出发,不仅考虑了交通速度的时间相关特性和空间相关特性,而且考虑了天气、节假日等外部因素的影响。时间特性上,我们利用LSTM来充分学习交通速度的时序关联性,还考虑了不同历史时刻对不同预测时刻的贡献差异性,并通过注意力机制来学习这种差异;空间特性上,我们在对目标路段进行预测时,充分考虑了相关路段对其的影响,考虑到不同相关路段对预测路段的影响不同,使用注意力机制来学习这种贡献差异。本方法充分考虑了不同历史时刻的贡献差异和不同相关路段的贡献差异。相比以往的方法,本方法是一种对时空特性刻画更加细粒度、更加全面的端到端方法,为交通速度预测提供了一种有效易行的方法,为实际道路状况预测的研究指明了新的方向。

附图说明

[0052] 图1是本发明方法架构图。

[0053] 图2是使用本发明在不同预测步下的10个路段上的预测结果RMSE的均值与其他三种方法的对比图。

[0054] 图3是使用本发明在不同预测步下的10个路段上的预测结果的RMSE的标准差与其他三种方法的对比图。

具体实施方式

[0055] 下面结合附图和具体实例,对本发明作进一步详细说明。

[0056] 本实例所采用数据为杭州市8000辆出租车2013年10月1日~2014年1月31日的为期四个月,共计123天的GPS信号数据,GPS数据中包括速度值和采样时间以及位置信息。

[0057] 步骤1:建模数据预处理。

[0058] 对原始速度数据进行均值化处理,对天气、节假日等外部因素类别数据进行预处理。

[0059] 对于原始速度数据,将获取的车辆速度数据按照不同路段分类。对需要分析的每一个路段,以一小时为时间间隔,将每天划分成24个时间段。对于第 z 个时间间隔, $z=0,1,2,\dots,23$,用该时间间隔内该路段上速度的平均值来表示第 z 个时间间隔的速度,每一个时刻对应一个固定的速度值。则路段 u 在 p 天内获得的速度向量可以表示为 $v_u^T(1 \times 24p)$,其中, $u=1,2,\dots,L$,为全部路段的代号, L 为路段总数。本实例中,我们对杭州市出租车2013年10月1日~2014年1月31日的为期四个月,共计123天的GPS信号进行了预处理,并从中随机选出了246个路段的数据进行接下来的分析, $p=123,L=246$ 。

[0060] 对于天气,按照降水量大小将其分类,分别用不同标签表示;将日期类型分为三类:工作日、周末、节假日,分别用不同标签表示;获取上述速度变量对应时刻的天气和日期类型,路段 u 在 p 天内的天气数据表示为 $w_u^T(1 \times 24p)$,日期类别数据表示为 $d_u^T(1 \times 24p)$ 。本实例中,对于天气,按照降水量大小将其分为三类,第一类是干燥无降水的天气,比如晴天、多云等;第二类是中度湿润,比如小雨、中雨、小雪、大雾等;第三类是高度湿润,比如大雨、大雪等,分别用数值1、2、3表示。对于节假日的影响,将日期类型分为三类:工作日、周末、节假日,分别用数值1、2、3来表示。本实例共获取为期四个月123天每天24小时的天气,以及123天每天的节假日类型,天气和节假日类型数据均对应于速度向量。如表1所示为某一路

段预处理后的部分数据的格式示例。

[0061] 表1

路段代号	当前速度	时间	天气	节假日
186	35	2013/10/1 9:00	1	3
186	32.1429	2013/10/1 10:00	1	3
186	32	2013/10/1 11:00	1	3
186	33.3571	2013/10/1 12:00	1	3
186	34	2013/10/1 13:00	1	3
186	30.4	2013/10/1 14:00	1	3

[0063] 步骤2:确定模型的输入变量和外部离散特征向量。

[0064] (2.1) 设c为待预测路段编号,根据地理位置,按照与预测路段的交叉和临近的优先原则选取前k个路段,作为与路段c相关的路段,这些相关路段的速度将作为变量用于预测模型的建立,本实例中k的值取4,选取了10个预测路段。

[0065] (2.2) 设m为历史时间窗口的大小, t_p 为第一个预测时刻,则模型预测该时刻所需的输入包括该时刻之前的m个时刻,对于历史时刻i ($i=t_p-m, t_p-m+1, \dots, t_p-1$),其对应所需的速度输入向量记为 $\mathbf{x}_i = [\mathbf{x}_i^1, \mathbf{x}_i^2, \dots, \mathbf{x}_i^k]$,其中k为相关路段的个数。此外还要输入预测时刻所对应的外部离散特征向量 \mathbf{d}_t ,包括天气 w_{tp} 、节假日 d_{tp} 在内的外部因素。本实例中m值为24。

[0066] 步骤3:模型的建立和训练。

[0067] (3.1) 确定相关路段的贡献程度,得到新的输入向量。

[0068] 编码器网络采用LSTM作为基础单元,历史时间步长为m,与步骤2中的历史时间窗口一致。对于历史时刻i所对应的原始输入向量x,首先计算每个相关路段的得分,如下式所示:

$$[0069] \quad e_i^j = \mathbf{v}_e^T \tanh(\mathbf{W}_e[\mathbf{h}_{i-1}; \mathbf{s}_{i-1}] + \mathbf{U}_e \mathbf{x}^j) \quad (1)$$

[0070] 其中, $j=1, 2, \dots, k$, \mathbf{h}_{i-1} 和 \mathbf{s}_{i-1} 分别为Encoder中的LSTM单元的前一时刻的f维隐藏层状态和细胞状态, $\mathbf{X}^j = [\mathbf{X}_{t_p-m}^j, \mathbf{X}_{t_p-m+1}^j, \dots, \mathbf{X}_{t_p-1}^j]^T$ 是所有输入时刻的第j个相关路段速度值所组成的向量, $\mathbf{v}_e \in \mathbb{R}^m$, $\mathbf{W}_e \in \mathbb{R}^{m \times 2f}$, $\mathbf{U}_e \in \mathbb{R}^{m \times m}$ 是网络需要学习的参数。本实例中f取值为96。

[0071] 使用类SoftMax对得分进行归一化处理,使其转化成类概率分布的权重系数:

$$[0072] \quad \alpha_i^j = \frac{\exp(e_i^j)}{\sum_{g=1}^k \exp(e_i^g)} \quad (2)$$

[0073] 然后将对应路段的速度值与对应的权重相乘,得到该时刻新的输入向量:

$$[0074] \quad \hat{\mathbf{x}}_i = [\alpha_i^1 \mathbf{x}_i^1, \alpha_i^2 \mathbf{x}_i^2, \dots, \alpha_i^k \mathbf{x}_i^k] \quad (3)$$

[0075] (3.2) 对编码器网络的LSTM单元进行计算和更新。

[0076] 将新的输入向量输入到LSTM单元,对于历史时刻i的LSTM单元,其计算和更新方式如下,首先计算遗忘门 f_i 、输入门 a_i 、输出门 o_i :

$$[0077] \quad \mathbf{f}_i = \sigma(\mathbf{W}_f[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i] + \mathbf{b}_f) \quad (4)$$

$$[0078] \quad \mathbf{a}_i = \sigma(\mathbf{W}_a[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i] + \mathbf{b}_a) \quad (5)$$

$$[0079] \quad \mathbf{o}_i = \sigma(\mathbf{W}_o[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i] + \mathbf{b}_o) \quad (6)$$

[0080] 其中, $[\mathbf{h}_{i-1}; \mathbf{x}_i]$ 是前一个时刻的隐藏层状态 \mathbf{h}_{i-1} 与当前时刻的输入 \mathbf{x}_i 在向量维度上的拼接。 σ 表示sigmoid函数, $\mathbf{W}_f, \mathbf{W}_a, \mathbf{W}_o, \mathbf{b}_f, \mathbf{b}_a, \mathbf{b}_o$ 都是网络要学习的参数。

[0081] 然后计算更新当前的细胞状态:

$$[0082] \quad \mathbf{s}_i = \mathbf{f}_i \odot \mathbf{s}_{i-1} + \mathbf{a}_i \odot \tanh(\mathbf{W}_s[\mathbf{h}_{i-1}; \hat{\mathbf{x}}_i] + \mathbf{b}_s) \quad (7)$$

[0083] 其中, $\mathbf{W}_s, \mathbf{b}_s$ 是待学习的参数, 符号 \odot 表示对应元素相乘。

[0084] 然后计算隐藏层状态如下:

$$[0085] \quad \mathbf{h}_i = \mathbf{o}_i \odot \tanh(\mathbf{s}_i) \quad (8)$$

[0086] (3.3) 确定不同历史时刻的贡献程度, 得到新的输入向量。

[0087] 解码器网络也是由LSTM单元构成。对于当前的预测时刻 t ($t = t_p, t_p+1, t_p+2, t_p$ 为第一个预测时刻), 首先我们利用解码器中当前时刻的前一时刻的隐藏层状态和细胞状态以及编码器中的每个隐藏层状态, 生成不同历史时刻对当前预测时刻的贡献分数, 如下:

$$[0088] \quad l_i^t = \mathbf{v}_d^T \tanh(\mathbf{W}_d[\mathbf{h}'_{t-1}; \mathbf{s}'_{t-1}] + \mathbf{U}_d \mathbf{h}_i), \quad i = t_p - m, t_p - m + 1, \dots, t_p - 1 \quad (9)$$

[0089] 其中, \mathbf{h}'_{t-1} 和 \mathbf{s}'_{t-1} 分别是预测时刻的前一时刻的解码器LSTM单元的隐藏层状态和细胞状态, \mathbf{h}_i 是历史时刻 i 的编码器LSTM单元的隐藏层状态, 也是编码器的第 i 个LSTM单元的隐藏层状态, m 是编码器部分中LSTM单元的数量, $\mathbf{v}_d, \mathbf{W}_d, \mathbf{U}_d$ 是要学习的参数。

[0090] 采用类SoftMax的方式对得分进行归一化, 得到第 i 个历史时刻隐藏层的权重系数:

$$[0091] \quad \beta_i^t = \frac{\exp(l_i^t)}{\sum_{j=1}^m \exp(l_j^t)} \quad (10)$$

[0092] 然后将权重系数与隐藏层状态对应相乘再求和, 计算出预测时刻 t 所对应的编码器中所有时刻的隐藏层状态的加权和 \mathbf{c}_t :

$$[0093] \quad \mathbf{c}_t = \sum_{i=1}^m \beta_i^t \mathbf{h}_i \quad (11)$$

[0094] 将 \mathbf{c}_t 与解码器中上一时刻的输出 \mathbf{y}_{t-1} 在维度上进行拼接, 作为解码器中当前时刻LSTM单元的输入向量:

$$[0095] \quad \mathbf{x}_t = [\mathbf{c}_t; \mathbf{y}_{t-1}] \quad (12)$$

[0096] 本实例中预测步数为3, 即解码器网络的LSTM单元个数为3。

[0097] (3.4) 对解码器网络的LSTM单元进行计算和更新, 步骤同(3.2)。

[0098] 即, 将新的输入向量输入到解码器网络的LSTM单元, 对于预测时刻 t 的LSTM单元, 其计算和更新方式如下, 首先计算遗忘门 f'_t 、输入门 a'_t 、输出门 o'_t :

$$[0099] \quad f'_t = \sigma(\mathbf{W}'_f[\mathbf{h}'_{t-1}; \mathbf{x}_t] + \mathbf{b}'_f)$$

$$[0100] \quad a'_t = \sigma(\mathbf{W}'_a[\mathbf{h}'_{t-1}; \mathbf{x}_t] + \mathbf{b}'_a)$$

$$[0101] \quad o'_t = \sigma(\mathbf{W}'_o[\mathbf{h}'_{t-1}; \mathbf{x}_t] + \mathbf{b}'_o)$$

[0102] 其中, $[h'_{t-1}; x_t]$ 是前一个时刻的隐藏层状态 h'_{t-1} 与当前时刻的输入 x_t 在向量维度上的拼接。 σ 表示sigmoid函数, $W'_f, W'_a, W'_o, b'_f, b'_a, b'_o$ 都是网络要学习的参数。

[0103] 然后计算更新当前的细胞状态:

$$[0104] \quad s'_t = f'_t \odot s'_{t-1} + a'_t \odot \tanh(W'_s[h'_{t-1}; x_t] + b'_s)$$

[0105] 其中, W'_s, b'_s 是待学习的参数, 符号 \odot 表示对应元素相乘。

[0106] 计算隐藏层状态如下:

$$[0107] \quad h'_t = o'_t \odot \tanh(s'_t)$$

[0108] (3.5) 构建考虑外部因素的全连接神经网络。

[0109] 在预测 t 时刻的交通速度时, 我们将解码器的输出和由外部因素组成的离散特征拼接起来, 并将它们输入到一个全连接神经网络, 得到最终的输出 \hat{y}_t ,

$$[0110] \quad \hat{y}_t = \sigma(W_n[y_t; \mathbf{d}_t] + b_n) \quad (13)$$

[0111] 其中, y_t 是 t 时刻解码器的输出值, \mathbf{d}_t 是 t 时刻的外部离散特征向量, 包括天气和日期, W_n 和 b_n 是待学习的参数。本实例中全连接神经网络的隐藏层数为1, 隐藏层的神经元个数设为10。

[0112] (3.6) 模型的训练。

[0113] 采用Adagrad梯度下降法对步骤(3.1)至(3.5)中的参数进行反向传播更新, 得到模型的最终参数。本实例10个路段分别有2800条样本记录, 针对不同路段, 我们采用前2400个样本作为训练集, 后400个样本作为测试集, Epoch设置为2。

[0114] 步骤4: 模型在线预测。

[0115] 对不同路段的400个测试样本, 首先按照步骤1到步骤2, 对测试样本的格式进行转化, 然后将测试样本直接输入到步骤3所建立好的模型中, 模型输出最终的预测结果, 完成预测。

[0116] 计算本方法和其他方法在10个随机选择的路段上的测试集的预测结果与真实值的均方根误差RMSE, 如表2所示。其中, 其他方法包括基于时空宽度学习网络ST-BLN、深度神经网络DNN和支持向量回归机SVR的交通速度预测方法, 各方法的输入一致。可以看出, 本方法在每个预测步骤的每个路段上均优于其他三种方法。表明本方法对预测精度有较为明显的提高。此外, 为观察该方法的稳定性, 我们针对每个预测时间步分别计算了四种方法在10个路段的标准差, 结果表明所提方法是所有四种方法中最稳定的。为了对结果进行更加直观的展示, 我们将10个路段上的四种方法在不同路段预测的RMSE的平均值和标准偏差绘制成曲线图, 分别如图2和图3所示。该图可以进一步验证结果的有效性, 也表明了随着预测步长的增加, 本方法的准确性下降的速度小于其他方法。

[0117] 表2

[0118]

路段号	本方法			ST-BLN		
	60 (min)	120 (min)	180 (min)	60 (min)	120 (min)	180 (min)
1	2.754	3.154	3.715	2.773	3.451	4.126
2	2.613	3.122	3.601	2.649	3.272	4.028
3	1.729	2.244	2.803	1.753	2.503	3.219
4	2.816	3.412	4.017	2.842	3.615	4.315
5	2.350	2.965	3.456	2.357	3.102	3.902
6	2.083	2.602	3.204	2.109	2.794	3.621
7	2.362	2.919	3.412	2.378	3.067	3.812
8	1.815	2.475	2.982	1.902	2.591	3.205
9	2.798	3.125	3.573	2.903	3.262	3.904
10	1.812	2.361	2.808	1.825	2.552	3.225
Mean	2.313	2.838	3.357	2.349	3.021	3.736
Std	0.431	0.392	0.402	0.435	0.393	0.402
路段号	DNN			SVR		
	60 (min)	120 (min)	180 (min)	60 (min)	120 (min)	180 (min)
1	3.042	3.478	4.253	3.014	3.501	4.261
2	2.837	3.302	4.029	3.191	3.256	4.004
3	1.763	2.597	3.322	1.774	2.518	3.327
4	2.888	3.665	4.396	2.893	3.671	4.385
5	2.352	3.213	3.917	2.427	3.217	3.900
6	2.251	2.681	3.713	2.532	2.803	3.681

[0119]

7	2.520	3.124	3.808	2.490	3.121	3.826
8	2.001	2.672	3.212	2.107	2.659	3.204
9	2.924	3.311	3.952	2.910	3.278	3.868
10	1.887	2.578	3.238	1.854	2.564	3.215
Mean	2.447	3.062	3.784	2.519	3.059	3.767
Std	0.467	0.399	0.414	0.491	0.401	0.413

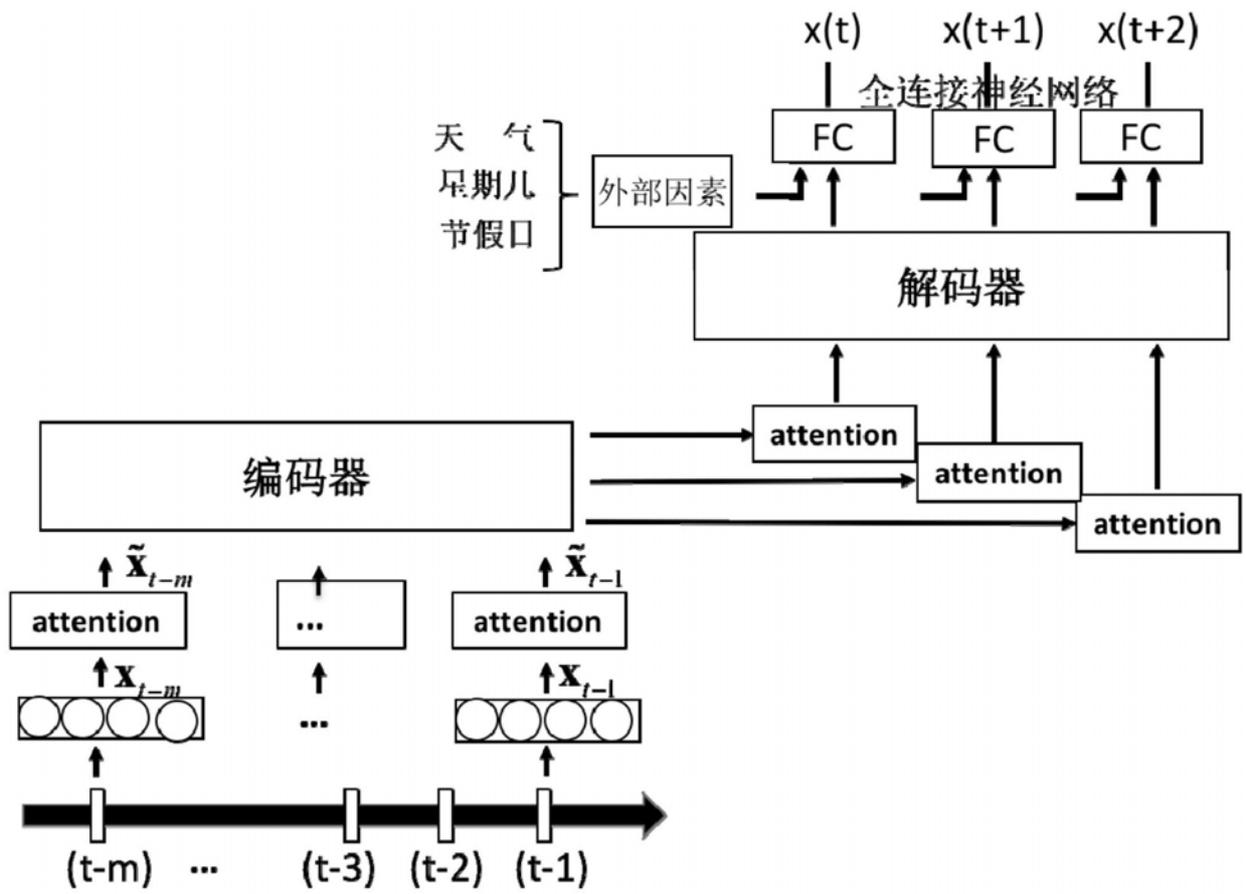


图1

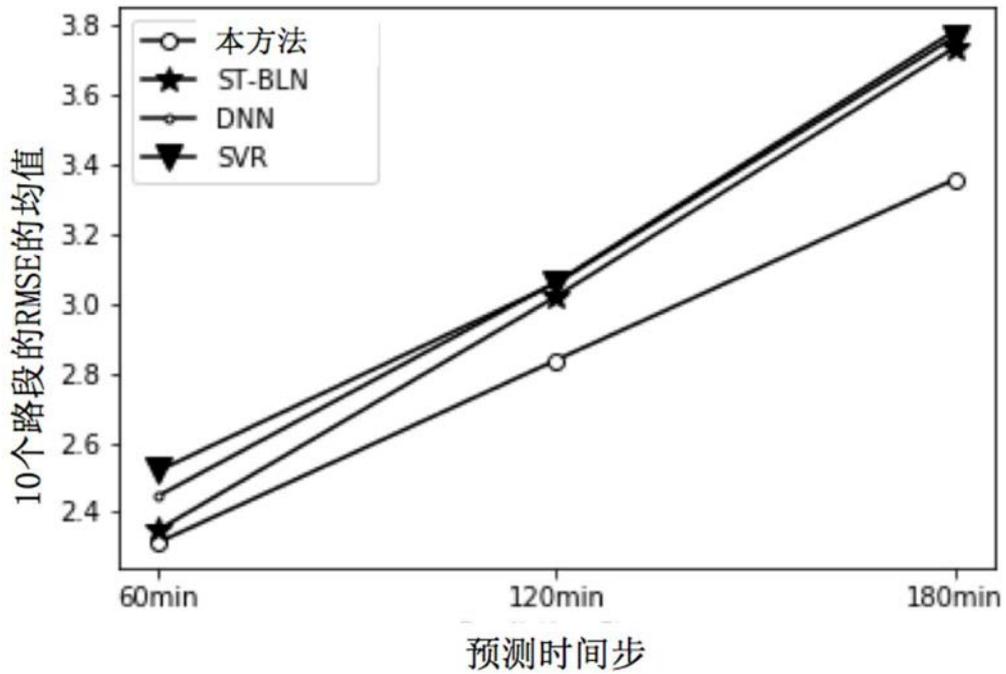


图2

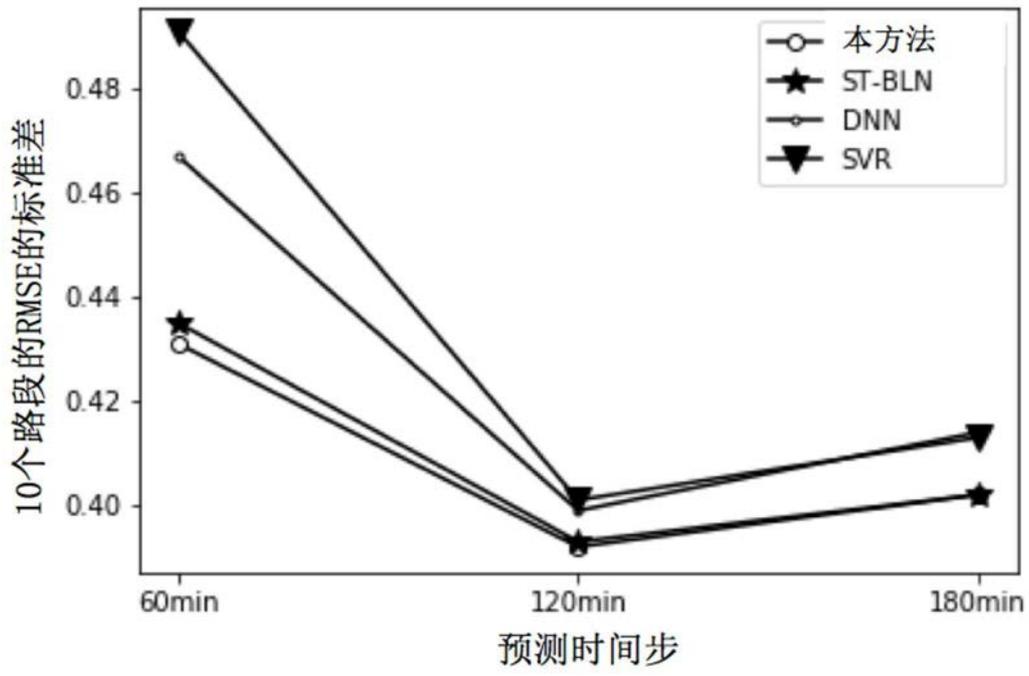


图3