Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction

引言

高效的交通系统需要精准的出行需求预测系统，使得交通系统能够提前进行资源分配，避免不必要的能源消耗。当前网约车服务的兴起，使得人们能够采集到大量交通数据，如何借助海量的数据预测出行需求进行在AI界引发了越来越多的关注。

本文研究如何利用历史数据对某一区域下一时刻的打车服务需求进行预测。前人在交通预测问题上通常采用以ARIMA及其变体为代表的传统时间序列预测方法，并基于时间序列考虑如何引入空间因素和各种外部因素，但仍然未能捕捉复杂的非线性时空相关性。深度学习也开始应用于交通预测问题中，但目前方法都没能同时捕捉时空关系（CNN捕捉空间关系，LSTM捕捉时间关系）。

文章提出将CNN与LSTM放在一个框架下以同时捕获两种关系，其关键思想在于：

- Local CNN的提出：作者认为在输入中包含弱相关区域会损害模型的性能，只考虑输入空间上邻近的区域（空间上邻近区域常常认为具有更强的需求模式相关性）
- 图与图嵌入方法的使用：Local CNN的使用会忽略一些空间距离远但需求模式相关度较高的区域，因此可以使用图来描述这种需求模式语义相关性（边权为需求模式相关度），并通过图嵌入方法将其作为环境特征输入到模型中。

文章贡献：

- 提出能够同时包含空间、时间以及语义关系的统一多视图模型
- 提出能够捕捉本地特征及与邻近区域关系的local CNN模型
- 根据需求模式的相似性建立了一个区域图，用于对具有相关性的远距离区域进行建模。这种隐含的语义关系通过图嵌入方式进行学习。
- 在滴滴的大规模数据集上进行了大量的测试，结果一致表明其优于其他前沿预测方法

相关文献工作

交通预测领域研究问题包含对交通量、出租车上下车次数、交通流和出租车需求预测等交通数据的预测。其公式化表示基本一致，目的都在于实现对某一位置在某一时间点交通数据的预测。

- 以ARIMA及其变体为代表的传统时间序列预测方法被广泛应用于交通预测。
- 近年来研究进一步探索了外部环境数据对交通预测的作用（场地类型、天气因素、节假日），各种不同的技巧也被用于描述空间关系（矩阵分解、规范化平滑）。

  > 以上研究均假设邻近区域需求模式一致，但基于传统方法因此无法刻画时空复杂非线性关系

- 深度学习兴起，开始被用于交通数据预测
  - CNN：
    - 将整个城市所有区域的交通数据作为输入

        - 只能刻画空间关系
    - LSTM：只能刻画时间关系

本文工作与前人最大区别：在深度学习模型下**同时考虑时空关系**

预备知识

基于2017 Zhang文章进行定义。

空间（locations，L）：将城市划分为不重叠的栅格区域：$L = l_1, l_2, \ldots, l_i, \ldots, l_N$。

时间（time intervals，T）：按时间顺序，以30min为区间划分时间片：$\mathcal{I} = I_0, I_1, \ldots, I_t, \ldots, I_T$。

出租车请求（taxi request，o）：$(o.t, o.l, o.u)$，三个分量分别代表时间戳、位置和用户身份编号

需求（demand，y）：$y_t^i = |\{o : o.t \in I_t \vee o.l \in I_i\}|$，$|\cdot|$代表集合的势，表示区域$i$在区间$t$的出租车总请求量

外部环境特征（context features）：所有外部环境特征（如时间特征、空间特征、气象因素）构成的外部环境特征向量$\mathrm{e}_t^i \in \mathbb{R}^r$，维度$r$为特征数量。

需求预测问题（demand prediction problem）定义如下：

$$y_{t+1}^i = \mathcal{F}(\mathcal{Y}_{t-h,\ldots,t}^L, \mathcal{E}_{t-h,\ldots,t}^L)$$

其中，$\mathcal{Y}_{t-h,\ldots,t}^L$代表历史需求（$t-h$到$t$时间段内），$\mathcal{E}_{t-h,\ldots,t}^L$代表所有区域的历史外部环境特征。

模型框架

分为空间、时间、语义三个视图。整体架构如下：

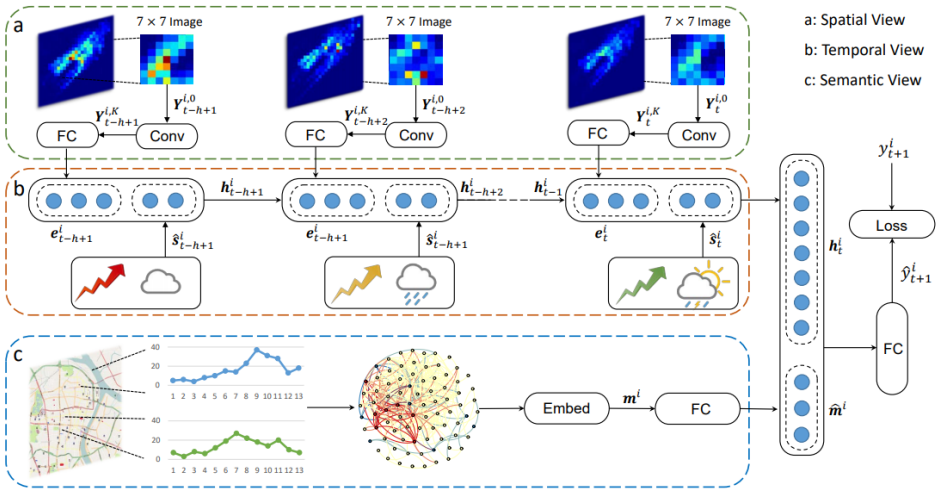

Figure 1: The Architecture of DMVST-Net. (a). The spatial component uses a local CNN to capture spatial dependency among nearby regions. The local CNN includes several convolutional layers. A fully connected layer is used at the end to get a low dimensional representation. (b). The temporal view employs a LSTM model, which takes the representations from the spatial view and concatenates them with context features at corresponding times. (c). The semantic view first constructs a weighted graph of regions (with weights representing functional similarity). Nodes are encoded into vectors. A fully connected layer is used at the end for jointly training. Finally, a fully connected neural network is used for prediction.

空间视图：Local CNN

对邻近区域的空间关系进行建模。

在每个时间段t，以区域i为中心，提取一个通道数为1的$S \times S$图像（值为对应需求数，S代表空间粒度，城市padding为0），可以得到对应的张量$\mathrm{Y}_t^i \in \mathbb{R}^{S \times S \times 1}$。将其作为输入$\mathrm{Y}_t^{i,0}$，通过$K$层卷积处理，输出$\mathrm{Y}_t^{i,K}$。

$$\mathrm{Y}_t^{i,k} = f(\mathrm{Y}_t^{i,k-1} * \mathrm{W}_t^k + \mathrm{b}_t^k)$$

激活函数为ReLU，$\mathrm{W}_t^k, \mathrm{b}_t^k$为所有区域所共享。输出$\mathrm{Y}_t^{i,k} \in \mathbb{R}^{S \times S \times \lambda}$再通过展开层（flatten layer）映射为特征向量$\mathrm{s}_t^i \in \mathbb{R}^{S^2 \lambda}$，该向量再经过一层全连接层来降维：

$$\hat{\mathrm{s}}_t^i = f(\mathrm{W}_t^{fc} \mathrm{s}_t^{i,k-1} + \mathrm{b}_t^{fc}), \text{其中} \ \hat{\mathrm{s}}_t^i \in \mathbb{R}^d$$

> 这里是等卷积吗？长宽输入输出一致

时间视图：LSTM

对需求时间序列的序列关系进行建模。

使用LSTM，其公式如下：

$$\begin{aligned}
\mathrm{i}_t^i &= \sigma(\mathrm{W}_i \mathrm{g}_t^i + \mathrm{U}_i \mathrm{h}_{t-1}^i + \mathrm{b}_i) \\
\mathrm{f}_t^i &= \sigma(\mathrm{W}_f \mathrm{g}_t^i + \mathrm{U}_f \mathrm{h}_{t-1}^i + \mathrm{b}_f) \\
\mathrm{o}_t^i &= \sigma(\mathrm{W}_o \mathrm{g}_t^i + \mathrm{U}_o \mathrm{h}_{t-1}^i + \mathrm{b}_o) \\
\theta_t^i &= \tanh(\mathrm{W}_g \mathrm{g}_t^i + \mathrm{U}_g \mathrm{h}_{t-1}^i + \mathrm{b}_g) \\
\mathrm{c}_t^i &= \mathrm{f}_t^i \circ \mathrm{c}_{t-1}^i + \mathrm{i}_t^i \circ \theta_t^i \\
\mathrm{h}_t^i &= \mathrm{o}_t^i \circ \tanh(\mathrm{c}_t^i)
\end{aligned}$$

$\mathrm{i}_t^i, \mathrm{f}_t^i, \mathrm{o}_t^i$分别为输入、遗忘、输出门，$\mathrm{g}_t^i \in \mathbb{R}^{r+d}$为空间视图输出与环境因素向量的拼接向量，最终输出$\mathrm{h}_t^i$：

$$\mathrm{g}_t^i = \hat{\mathrm{s}}_t^i \oplus \mathrm{e}_t^i$$

语义视图：结构化嵌入

具有相同功能属性的区域会有相近的需求模式，但它们在空间上可能相隔很远。因此，文章通过建立全连接图$G(V, E, D)$来描绘这种语义上的相关性。节点集合V即区域集合L，$E \in V \times V$，边权D为相似度ω。

$$w_{ij} = \exp(-\alpha \mathrm{DTW}(i, j))$$

α为距离衰减因子（$\alpha = 1$），$\mathrm{DTW}(i, j)$为两个区域需求模式间的DTW（动态时间规整）值（描述两个序列的相似程度指标，这里比较的序列为**平均周需求序列**）

使用图嵌入方法LINE将该图嵌入到低维向量空间得到m^i，再通过全连接层转化为$\hat{\mathrm{m}}^i$以实现整个网络的共同训练：

$$\hat{\mathrm{m}}^i = f(W_{fe}\mathrm{m}^i + b_{fe})$$

> 语义视图反映的是所有时间段的需求模式相关性，因此没有下标。

预测元件

将时间视图和语义视图输出拼接为q_t^i：

$$\mathrm{q}_t^i = \mathrm{h}_t^i \oplus \hat{\mathrm{m}}^i$$

整合通过全连接层输出$[0,1]$的最终值（输入已标准化）：

$$\hat{y}_{t+1}^i = \sigma(W_{ff}\mathrm{q}_t^i + b_{ff})$$

训练过程

考虑到MSE易受极端值支配，损失函数由MSE和MAE组成：

$$\mathcal{L}(\theta) = \sum_{i=1}^{N}((y_{t+1}^i - \hat{y}_{t+1}^i)^2 + \gamma(\frac{y_{t+1}^i - \hat{y}_{t+1}^i}{y_{t+1}^i})^2)$$

训练过程算法如下，采用Adam优化求解器，使用Tensorflow和Keras搭建网络架构：

Algorithm 1: Training Pipeline of DMVST-Net

Input: Historical observations: $\mathcal{Y}^L_{1,\dots,t}$; Context features: $\mathcal{E}^L_{t-h,\dots,t}$; Region structure graph $G = (V, E, D)$; Length of the time period h;
Output: Learned DMVST-Net model

Initialization;
for $\forall i \in L$ **do**
  Use LINE on G and get the embedding result $\mathbf{m}^i$;
  for $\forall t \in [h, T]$ **do**
    $\mathcal{S}_{spa} = [\mathbf{Y}^i_{t-h+1}, \mathbf{Y}^i_{t-h+2}, ..., \mathbf{Y}^i_t]$;
    $\mathcal{S}_{cox} = [\mathbf{e}^i_{t-h+1}, \mathbf{e}^i_{t-h+2}, ..., \mathbf{e}^i_t]$;
    Append $< \{\mathcal{S}_{spa}, \mathcal{S}_{cox}, \mathbf{m}^i\}, y^i_{t+1} >$ to Ω_{bt} ;
  end
end
Initialize all learnable parameters θ in DMVST-Net;
repeat
  Randomly select a batch of instance Ω_{bt} from Ω;
  Optimize θ by minimizing the loss function Eq. (9) with Ω_{bt}
until *stopping criteria is met*;

实验

数据集

数据集设置见下图，筛去了需求数<10的样本。

条目	信息
数据集	广州市滴滴出行数据（2017.2.1-2017.3.26）
区域划分	20×20，0.7km×0.7km
环境特征	时间特征（前4时间片平均）、空间特征（区域中心经纬度）、天气特征、活动特征（节假日）
训练集/测试集	前47天/后7天
区间长度	30min
输入时间序列	8×30min=4h

评价指标

MAPE与RMSE

比较基准

参与比较的模型如下，均采用同样代价函数。

- Historical average (HA)
- Autoregressive integrated moving average (ARIMA)
- Linear regression (LR)：包含最小二乘回归、岭回归、lasso回归
- Multiple layer perceptron (MLP)：(128,128,64,64)
- XGBoost
- ST-ResNet

同时进行了使用不同组件的效果对比：

- 时间视图
- 时间视图+语义视图
- 时间视图+空间视图（直接使用邻居节点需求）：为了与后者比较，体现出LCNN的优点
- 时间试图+空间视图（Local CNN，LCNN）
- DMVST-Net

预处理与参数设置

预处理

- 历史需求：进行Max-Min标准化
- 环境因素：对离散变量进行独热编码，连续变量Max-Min标准化

参数设置

参数	设置值
粒度S	9
卷积层数K	3
卷积核大小τ	3×3
卷积核数λ	64
输出维度d	64
序列长度h	8
图嵌入输出维数	32
语义层输出维数	6
批大小	64
early-stop round	10
max epoch	100

训练集中，前90%用于训练，10%用于验证，并使用了early-stop。

测试结果

SMVST-Net均显著优于其他方法。

与其他方法对比

Table 1: Comparison with Different Baselines

Method	MAPE	RMSE
Historical average	0.2513	12.167
ARIMA	0.2215	11.932
Ordinary least square regression	0.2063	10.234
Ridge regression	0.2061	10.224
Lasso	0.2091	10.327
Multiple layer perceptron	0.1840	10.609
XGBoost	0.1953	10.012
ST-ResNet	0.1971	10.298
DMVST-Net	**0.1616**	**9.642**

不同组件效果对比

Table 2: Comparison with Variants of DMVST-Net

Method	MAPE	RMSE
Temporal view	0.1721	9.812
Temporal + Semantic view	0.1708	9.789
Temporal + Spatial (Neighbor) view	0.1710	9.796
Temporal + Spatial (LCNN) view	0.1640	9.695
DMVST-Net	**0.1616**	**9.642**

一周不同时间不同方法效果对比

周末预测效果差于工作日，即周末更难进行预测。（工作日的白天都需要通勤，形成相似需求模式）

因此，文章使用工作日相比周末预测误差的相对增加作为指标，测试了模型的健壮性，结果证明SMVST-Net的健壮性要普遍优于其他方法。

$$\frac{\bar{wk} - \bar{wd}}{\bar{wd}}$$

$\bar{wk}, \bar{wd}$分别代表工作日和周末的平均预测误差。

LSTM输入序列长度和LCNN输入粒度对结果的影响

- 输入序列长度：随着输入序列长度增加，误差下降，但超过4h后，参数不断增多，使得训练变得困难，误差不再继续下降。
- 输入粒度大小：当粒度超过9后，随着选择邻近范围大小的增加，本地的显著关联会逐渐被平均掉，证明了LCNN的有效性。

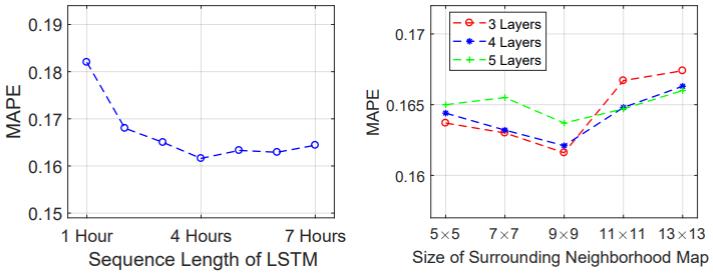

Figure 3: (a) MAPE with respect to sequence length for LSTM. (b) MAPE with respect to the input size for local CNN.

结论与讨论

测试结果证明SMVST-Net在时空关系和语义关系的捕捉上取得了较好的效果，优于其他模型。此外，文章将在后续继续探究如何进一步改进性能以获得更好的解释性，以及将更多的隐藏信息（如POI）纳入模型考虑范围内。