Trajectory Prediction Learning Using Deep Generative Models

MSc. Thesis of Jing Li
Department of Electrical Engineering and Computer Science

Dec 19, 2023
Introduction
Trajectory/Mobility Data

Trajectory: A Sequence of (Spatiotemporal) Points

Vast Amounts of Trajectory/Mobility Data
Trajectory-related Problems

- trajectory similarity
- trajectory clustering
- trajectory imputation
- pedestrian crowd behavior

...
Problem of Interest: Trajectory Prediction

Predict future trajectory
Plethora of Applications

Ride-sharing services

Next POI recommendation

Autonomous vehicles

Traffic flow optimization
Problem Statement
Trajectory Prediction

Let

- an observation area $M$
- an observation period $[0, W]$
- a set of objects $N$ and their history trajectories $S'$

Input: Given

- a moving object $i$ in $N$
- a partial trajectory $T = \langle p_{i1}, p_{i2}, \ldots, p_i \rangle$
- a prediction horizon $k > 0$

Output: We want to

predict the next $k$ spatiotemporal points $\langle p_{i1}, p_{i2}, \ldots, p_{i+k} \rangle$ of the partial trajectory $T$
Overview

‣ Higher-order Mobility Flow Data
‣ (Revisit) Problem Statement
‣ Existing Works
‣ Methodology
‣ Evaluation
‣ Conclusions
Higher-order Mobility Flow Data
Challenges of Working with Trajectory Data

Data Sparsity
Limited data

Model Incompatibility
Not compatible with well-known machine learning models

Imbalanced Data
80% of the data is generated by 20% of the users

Low accuracy and completeness
Data Quality
Why hexagons?

- More **circular** that fully tessellates the space
- Same **distance** to all adjacent neighbours
Trajectories: Sequences of Hexagons

Trajectory: $h_1, h_2, h_3 \ldots h_{20}, h_{21}, h_{22}$
Treat *Trajectories* as Language *Statement*
Treat Trajectories as Language Statements

Hexagons represent ‘tokens’ & trajectories represent ‘sentences’

Trajectory: 

Sentence: I like to learn English

Advantages:

- Reduced data sparsity
- More compatible with well-known ML models (e.g., sequence models, LLMs)
Point2Hex: Overview of the Pipeline

GPS Traces or POI Check-Ins (input)

Linestring of Trajectories (Map-matching)

Map Tessellation with Trajectories (Hexagon-shaped cells)

Intersection of Linestrings and Polygons (Computational Geometry)

Higher-order Mobility Flow (Output)
Code of Point2Hex (Data Generator)

The code to generate your HO dataset from raw GPS points

Check it on GitHub
## Datasets: Higher-order Mobility Flow

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Trajectories</th>
<th>Time Period</th>
<th>Resolutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>HO-T-Drive</td>
<td>65,117</td>
<td>02/02/08 - 02/08/08</td>
<td>{6,…,10}</td>
</tr>
<tr>
<td>HO-Porto</td>
<td>1,668,859</td>
<td>07/01/13 - 06/30/14</td>
<td>{6,…,10}</td>
</tr>
<tr>
<td>HO-Rome</td>
<td>5,873</td>
<td>02/01/14 - 03/02/14</td>
<td>{6,…,10}</td>
</tr>
<tr>
<td>HO-GeoLife</td>
<td>2,100</td>
<td>04/01/07 - 10/31/11</td>
<td>{6,…,10}</td>
</tr>
<tr>
<td>HO-FourSquare-NYC</td>
<td>49,983</td>
<td>04/12/12 - 02/16/13</td>
<td>{6,…,10}</td>
</tr>
<tr>
<td>HO-FourSquare-TKY</td>
<td>117,593</td>
<td>04/12/12 - 02/16/13</td>
<td>{6,…,10}</td>
</tr>
<tr>
<td>HO-NYC-Taxi</td>
<td>2,062,554</td>
<td>01/01/16 - 06/30/16</td>
<td>{6,…,10}</td>
</tr>
</tbody>
</table>
(Revisited) Problem Statement
Trajectory Prediction (Revisited)

Let
- an observation area $M$
- an observation period $[0, W]$
- a set of objects $N$ and their history trajectories $S$

Input: Given
- a moving object $i$ in $N$
- a partial trajectory $T = <b_{i1}, b_{i2}, \ldots, b_{il}>$
- a prediction horizon $k > 0$

Output: We want to predict the next $k$ blocks $<b_{il+1}, b_{il+2}, \ldots, b_{il+k}>$ of the partial trajectory $T$
Contributions

- **Point2Hex**: GPS trajectories to HO mobility flow data
- Propose to leverage deep generative models for trajectory prediction
- Propose a transformer-based framework **TrajLearn**
- **TrajLearn** *outperforms* the state-of-the-art baselines
- Make the **source code** publicly available to facilitate the reproducibility
Literature Overview

- Computer Vision
- Trajectory Analysis
  - Macroscopic Analysis
  - Microscopic Analysis
    - Statistical Methods
    - Deep Learning Methods
General Related Work

Computer Vision Domain

- Predict future path or movement of objects in a scene (a small scale) over time

Out of the scope: Rely on camera-generated video frames

Macroscopic Trajectory Analysis

- Focus on high-level (city-level or region-level) mobility predictions (instead of individual level)

Different focus: crowd flow prediction [Lin et al. AAAI’19], taxi demand prediction [Yao et al. AAAI’18]
Matrix Factorization
- Decompose matrix into matrices that representing object preferences and location attributes
  
  **Examples:** Fused MF [Cheng et al. AAAI’12], GeoMF [Lian et al. SIGKDD’14], Rank-geofm [Li et al. SIGIR’15]

Markov Chain
- Model the sequence of visits as a chain of states, governed by transition probabilities
  
  **Examples:** HMM [Mathew et al. UbiComp’12], FPMC-LR [Cheng et al. IJCAI’13], Semantics-aware HMM [Shi et al. TKDE’19]

Limitations
- Limited scalability
- Often rely on assumptions about the data distribution
- Feature engineering is required
Deep Learning Methods - 1/2

RNN/LSTM/GRU

- Use recurrent neural networks to process sequential data

Examples: ST-RNN [Liu et al. AAAI'16], HST-LSTM [Kong et al. IJCAI'18], DeepTrip [Zhang et al. IEEE trans Intell Transp Syst'23]

Attention Mechanism

- Allow models to focus on different parts of the input sequence when producing the output

Examples: DeepMove [Feng et al. WWW’18], GeoSAN [Lian et al. KDD’20], STAN [Luo et al. WWW’21]

Limitations

- Mostly designed for the POI prediction
- Data sparsity and imprecision
Deep Learning Methods - 2/2

Specialized Works

● **DeepUrbanMomentum** [Jiang et al. AAAI’18]
  ○ **Limitations**: Need other information

● **Continuous Trajectory Prediction** [Sadri et al. IMWUT’18]
  ○ **Limitations**: Heavily rely on a single historical record of an individual

● **From movement purpose to mobility prediction** [Amichi et al. SIGSPATIAL’21]
  ○ **Limitations**: Need to add movement semantic to trajectories
Methodology
Architecture Overview

Beam Search

Output: $N$ sequences of hexagons (length $k$) with a prob.

<table>
<thead>
<tr>
<th>Prob.</th>
<th>$S_1$ 0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$S_2$ 0.1</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>$S_N$ 0.005</td>
</tr>
</tbody>
</table>

Decoder block #L
Decoder block #L-1
Decoder block #1
Feed Forward Neural Nets
Layer Norm
Masked Self-Attention
Layer Norm
Input of Transformer

The input to the transformer

\[ h_0 = B W_e + W_p \]

Where

: Higher-order mobility flow
: Block embedding matrix
: Position embedding matrix
Hidden State Computation

The hidden state of each transformer layer

\[ h'_j = h_{j-1} + \text{Self} - \text{Attention}(\text{LayerNorm}(h_{j-1})) \]
\[ h_j = h'_j + \text{FeedForward}(\text{LayerNorm}(h'_j)) \]

Where

- \text{LayerNorm}() : Layer normalization
- \text{Self-Attention}() : Masked multi-head self-attention operation
- \text{FeedForward}() : Position-wise feed-forward network
Activation Function

Gaussian Error Linear Unit (GELU)

\[ GELU(x) = x \cdot P(X \leq x) \]

Where and implemented as

\[ 0.5x \left( 1 + \tanh \left( \sqrt{\frac{2}{\pi}} (x + 0.044715x^3) \right) \right) \]
Next Block/Hexagon Prediction

Based on the probabilities of all possible next blocks

\[ P(b_{l+1} | B) = \text{softmax}(\text{FeedForward}(\text{LayerNorm}(h_L))) \]
### Model Training

**<EOT> in Trajectories**

- **Temporal cutoff**: time threshold

  Gap in GPS data beyond this threshold indicates the end of the trajectory

- **Spatial cutoff**: distance threshold

  Distance between consecutive GPS points is greater than this threshold
Model Training

Teaching Forcing

Without Teacher Forcing

With Teacher Forcing

Ground Truth
Beam Search with Constraints

A heuristic search algorithm that explores the most promising trajectory paths

- Initialization
- Beam expansion
- Beam pruning
- Termination
Beam Expansion

The probability at each step is updated based on their cumulative probabilities

\[ P(b_{i_1} \ldots b_{i_n}) = P(b_{i_1} \ldots b_{i_{n-1}}) \times P(b_{i_n} | b_{i_1} \ldots b_{i_{n-1}}) \]
Evaluation
Experimental Scenarios

RQ 1) Accuracy
- What is the accuracy performance of our method against baselines?

RQ 2) Sensitivity Analysis
- How does the performance vary with different input trajectory lengths and prediction lengths?

RQ 3) Map Resolution Analysis
- How does the performance vary with different tessellation levels?

RQ 4) Ablation Study
- How does beam search with the constraints impact the performance?
Datasets

Timely ordered trajectory data set is split into:

70% Training, 10 % Validation, 20% Testing

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Objects</th>
<th>Trajectories</th>
<th>Time Period</th>
<th>Resolutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>HO-Rome</td>
<td>315</td>
<td>5,873</td>
<td>02/01/14 - 03/02/14</td>
<td>{7, 8, 9}</td>
</tr>
<tr>
<td>HO-Porto</td>
<td>442</td>
<td>1,668,859</td>
<td>07/01/13 - 06/30/14</td>
<td>{7, 8, 9}</td>
</tr>
<tr>
<td>HO-GeoLife</td>
<td>57</td>
<td>2,100</td>
<td>04/01/07 - 10/31/11</td>
<td>{7, 8, 9}</td>
</tr>
</tbody>
</table>
Experimental Setup

Computational Environment
- NVIDIA RTX A6000 graphics card and 320GB of memory
- Implementation: Python 3, PyTorch 1.13

Map Tessellation and Resolutions
- H3 geo-indexing system

Deep Generative Model
- Based on the GPT-2 LLM architecture

Training Parameters
- AdamW optimizer with learning rate = $5 \times 10^{-3}$
- Batch size = 64
- Dropout rate = 0.1
Baselines

Statistical Methods
● MC

Deep Learning Methods
● LSTM
● GRU
● LSTM-ATTN
● DeepMove

Our Method
● TrajLearn
Metrics

Accuracy@N

- Measure the proportion of true samples included in the predictions

\[
\text{Accuracy@N} = \frac{|\{s \mid s \in P, true(s) \in Top_n(s)\}|}{|P|}
\]

BLEU Score

- Measure how many n-grams of the predicted sequence match with the n-grams in the actual sequence

\[
BLEU = BP \cdot \exp \left( \sum_{n=1}^{N} w_n \log p_n \right) \quad BP = \begin{cases} 
1 & \text{if } c > r \\
e^{(1-r/c)} & \text{if } c \leq r
\end{cases}
\]
### RQ 1: Model Accuracy Performance

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Resolution 7</th>
<th>Resolution 8</th>
<th>Resolution 9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Acc@1</td>
<td>Acc@3</td>
<td>Acc@5</td>
</tr>
<tr>
<td>Ho-Porto</td>
<td>LSTM</td>
<td>0.3284</td>
<td>0.5866</td>
<td>0.4908</td>
</tr>
<tr>
<td></td>
<td>LSTM-ATTN</td>
<td>0.5970</td>
<td>0.6318</td>
<td>0.6400</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>0.1113</td>
<td>0.1923</td>
<td>0.2065</td>
</tr>
<tr>
<td></td>
<td>DeepMove</td>
<td>0.5532</td>
<td>0.5877</td>
<td>0.5957</td>
</tr>
<tr>
<td></td>
<td>TRAJLEARN (ours)</td>
<td>0.6917</td>
<td>0.8066</td>
<td>0.8490</td>
</tr>
<tr>
<td></td>
<td>Improvement (%)</td>
<td>15.86 %</td>
<td>27.65 %</td>
<td>32.64 %</td>
</tr>
<tr>
<td>Ho-Rome</td>
<td>LSTM</td>
<td>0.2088</td>
<td>0.3982</td>
<td>0.4690</td>
</tr>
<tr>
<td></td>
<td>LSTM-ATTN</td>
<td>0.2820</td>
<td>0.3138</td>
<td>0.3227</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>0.1079</td>
<td>0.1522</td>
<td>0.1850</td>
</tr>
<tr>
<td></td>
<td>DeepMove</td>
<td>0.2966</td>
<td>0.3298</td>
<td>0.3385</td>
</tr>
<tr>
<td></td>
<td>TRAJLEARN (ours)</td>
<td>0.3406</td>
<td>0.4969</td>
<td>0.5793</td>
</tr>
<tr>
<td></td>
<td>Improvement (%)</td>
<td>9.98 %</td>
<td>-4.61 %</td>
<td>-10.81 %</td>
</tr>
<tr>
<td>Ho-GeoLife</td>
<td>LSTM</td>
<td>0.2153</td>
<td>0.4917</td>
<td>0.6050</td>
</tr>
<tr>
<td></td>
<td>LSTM-ATTN</td>
<td>0.5900</td>
<td>0.6086</td>
<td>0.6114</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>0.4944</td>
<td>0.5559</td>
<td>0.5621</td>
</tr>
<tr>
<td></td>
<td>DeepMove</td>
<td>0.6229</td>
<td>0.6435</td>
<td>0.6465</td>
</tr>
<tr>
<td></td>
<td>TRAJLEARN (ours)</td>
<td>0.5295</td>
<td>0.6742</td>
<td>0.7370</td>
</tr>
<tr>
<td></td>
<td>Improvement (%)</td>
<td>70.10 %</td>
<td>22.32 %</td>
<td>17.16 %</td>
</tr>
</tbody>
</table>
RQ 2) Parameter Sensitivity Analysis

(a) ACCURACY@1

(b) ACCURACY@3

(c) ACCURACY@5
RQ 3) Map Resolution Analysis

7: star
8: circle
9: triangle
RQ 4) Ablation Study

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy@1</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ho-Porto@7</td>
<td>0.6844</td>
<td>-1.07%</td>
</tr>
<tr>
<td>Ho-Porto@8</td>
<td>0.4992</td>
<td>-2.86%</td>
</tr>
<tr>
<td>Ho-Porto@9</td>
<td>0.5672</td>
<td>-1.76%</td>
</tr>
</tbody>
</table>

**Table:**
- **Embedding vector size**
- **# Attention heads**
- **# Transformer layers**
Conclusions
Summary

point2hex: software and datasets

GenAI for trajectory prediction

TrajLearn

Beam search
Limitations

- Discretization and precision:
  - Too coarse → miss important details
  - Too fine → increased computational complexity

- Data volume
  - May end up with a large amount of data → strain computational resources and require efficient data storage
Future Work - Interaction Prediction

Can we use trajectory prediction models for predicting mobility network interactions?
Future Work - Trajectory Foundation Model

Can we develop trajectory foundation models for addressing many trajectory-related tasks?

Trajectory similarity

Trajectory imputation
Papers Published/Submitted


Thank you!

Questions?