Context-aware Deep Model for Joint Mobility and Time Prediction

Rao Xuan

2021-9-29

Research direction

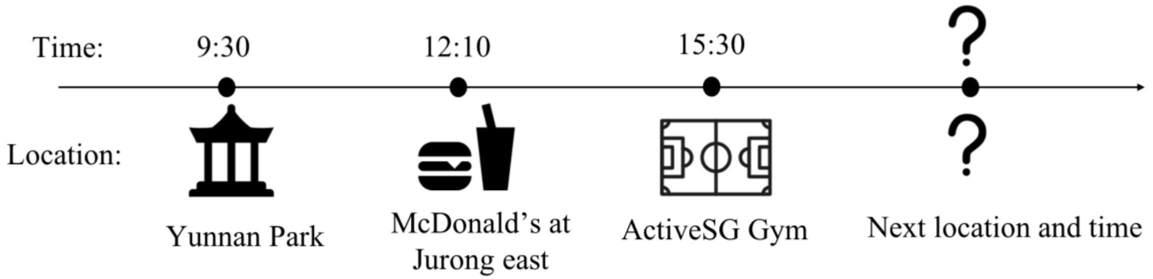

Figure 1: An example of joint mobility and time prediction

- propose a novel context-aware deep model called **DeepJMT** for jointly performing mobility prediction (to know where) and time prediction (to know when)

Framework

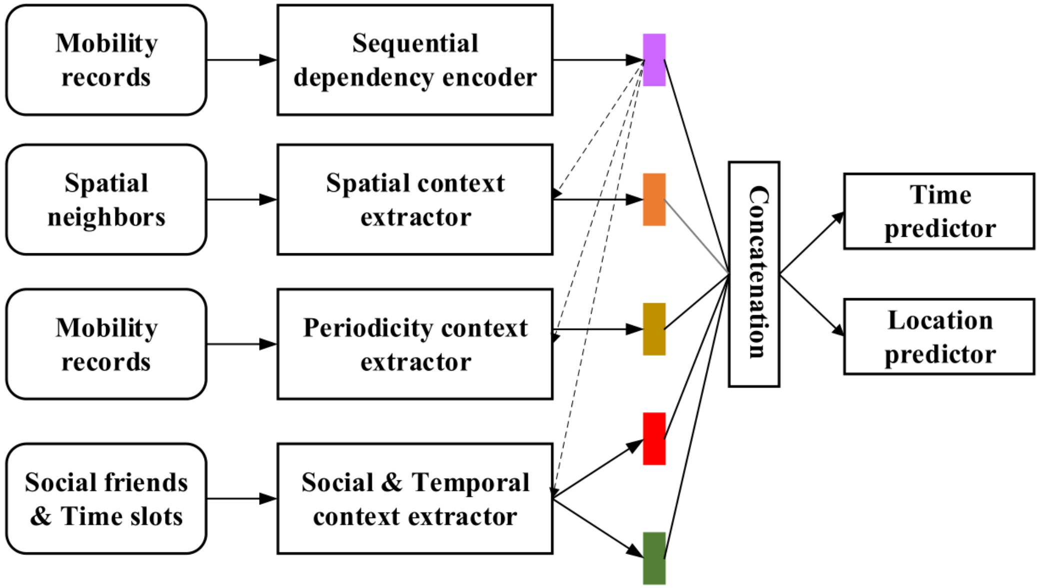

Figure 2: Overview structure of DeepJTM. DeepJMT is composed of a sequential dependency encoder, a spatial context extractor, a periodicity context extractor, a social & temporal context extractor and two predictors for mobility prediction and time prediction respectively.

- **Sequential dependency encoder**: capture a user's mobility regularities and temporal patterns.
- **Spatial context extractor**: extract user's location semantics.
- **Periodicity context extractor**: extract user's periodicity.
- **Social & temporal context extractor**: extract the mobility and temporal evidence from social relationships.

Sequential dependency encoder

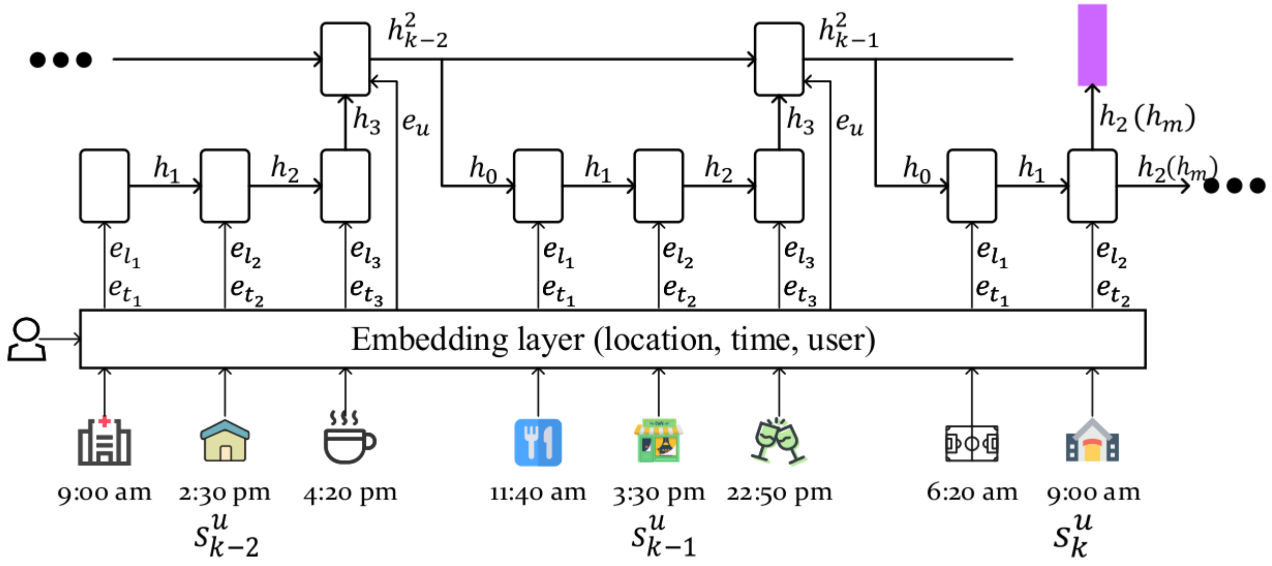

Figure 3: Structure of sequential dependency encoder. We show the latest three trajectories s^u_{k-2}, s^u_{k-1}, and s^u_k of user u till the current spatial-temporal point p^k_m.

- M = {p 1 ,p 2 ,..., } : Some user mobility records.
- Split the whole mobility records M into non-overlapping trajectories set S,where each element is a subsequence of M .
- The maximum time gap between two consecutive points at most 6 hours.

Sequential dependency encoder

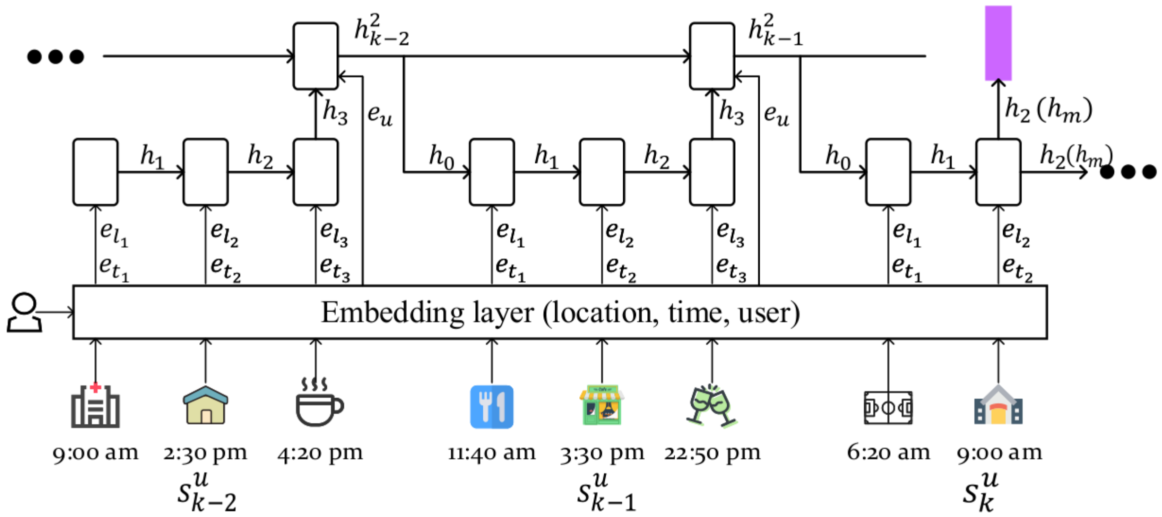

Figure 3: Structure of sequential dependency encoder. We show the latest three trajectories s_{k-2}^u, s_{k-1}^u, and s_k^u of user u till the current spatial-temporal point p_m^k.

- Sequential dependency encoder corresponds to a hierarchical recurrent neural network (RNN), where GRU units are used at both the low-level RNN and the high-level RNN.

$$h_i = \mathrm{GRU}_{low}(e_{l_i} \oplus e_{t_i}, h_{i-1}) \qquad (2)$$

- ◆ The low-level RNN is for modeling transitions within a trajectory.
- ◆ The last hidden state of the whole trajectory is sent to the high-level RNN.

$$h_j^2 = \mathrm{GRU}_{high}(e_u \oplus h_n, h_{j-1}^2) \qquad (3)$$

- ■ The high-level RNN is for modeling transitions between trajectories.
- ■ The hidden state is then fed to be the initial hidden state of the low-level RNN

Spatial Context Extractor

- Extract the semantics about a user's current location by leveraging its spatial neighbors

$$c_l = g_l\left(\sum_{l_i \in C(l)} \alpha_{l_i} e_{l_i}\right) \tag{4}$$

  - Use both the geographic distances and the dynamic influence for defining the weights.

$$D(l_x, l_y) = \exp\left(-\frac{dist(l_x, l_y)}{\beta}\right) \tag{5}$$

$$q^s_{l_i} = h_m^\top W_l e_{l_i} \tag{6}$$

$$\alpha_{l_i} = \frac{\exp\left(q^s_{l_i} \cdot D(l_i, l)\right)}{\sum_{l_\tau \in C(l)} \exp\left(q^s_{l_\tau} \cdot D(l_\tau, l)\right)} \tag{7}$$

Periodicity Context Extractor

- Periodicity phenomenon is reflected by not only the mobility that is close to the current step but also that of steps away.

    - Use an attention based GRU to extract periodical patterns from mobility records..

$$h'_i = \mathrm{GRU}_{period}\left(e_{l_i} \oplus e_{t_i} \oplus e_u, h'_{i-1}\right) \quad (8)$$

$$q^h_{h'_i} = h_m^\top W_g h'_i \quad (9)$$

$$\alpha_{h'_i} = \frac{\exp\left(q^h_{h'_i}\right)}{\sum_{\tau=1}^{k} \exp\left(q^h_{h'_\tau}\right)}, c_p = \sum_{i=1}^{k} \alpha_{h'_i} h'_i \quad (10)$$

Social & Temporal Context Extractor

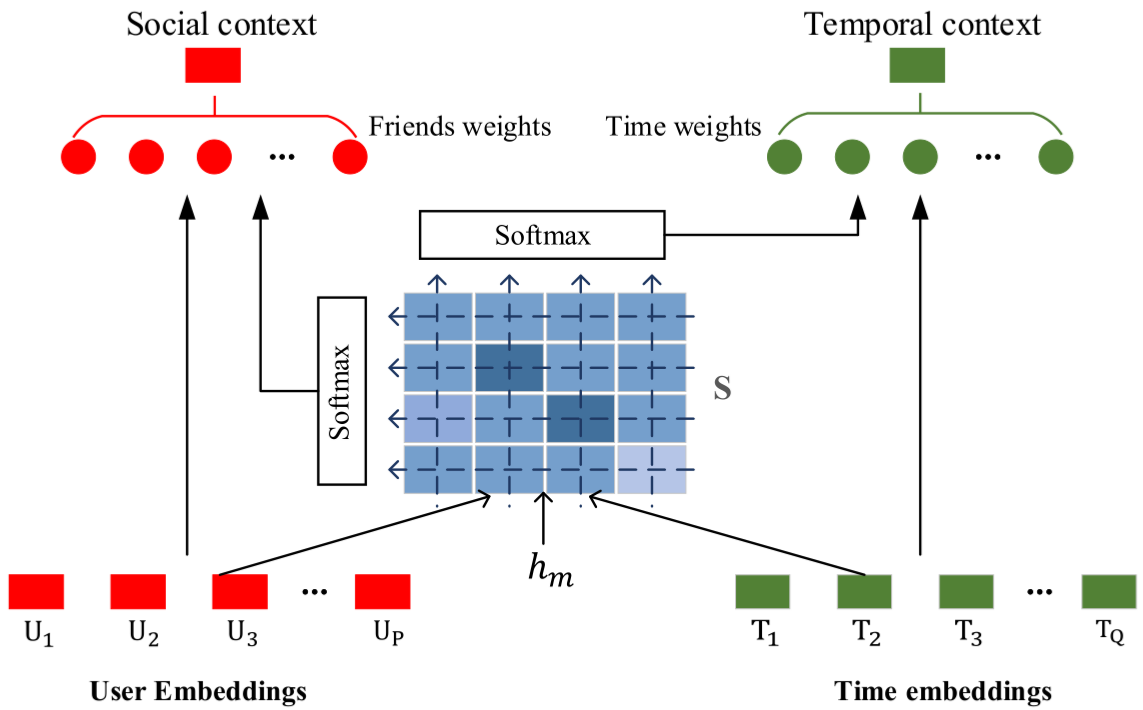

Figure 4: Structure of social & time context extractor

- Social context captures the aggregated influence on a user from his/her friends based on their similarity.
- Temporal context captures a user's preference on time slots based on his/her friends' preferences.
- Use use co-attention mechanism to jointly reason about the co-attention weights of different user-time pairs.

$$S_{ij} = [W_u h_m \oplus U_i]^\top W_s [W_t h_m \oplus T_j] \tag{11}$$

- ◆ A score which indicates the influence of user's i-th friend during time slot j considering user's current mobility status.

$$c_u = g_u(softmax(pool_{row}(S))^\top \cdot U) \tag{12}$$

$$c_t = g_t(softmax(pool_{col}(S))^\top \cdot T) \tag{13}$$

- ◆ Select the most representative time slot for each friend and the most representative friend for each time slot by max-pooling function.

Training and Inference

- Training

◆ Compute a probability distribution using a softmax function for mobility prediction.

$$P(l_{m+1} = l_i | \mathcal{H}_{t_m}) = \frac{\exp\left(W_i^\top \theta_m^l\right)}{\sum_{\tau=1}^{N} \exp\left(W_\tau^\top \theta_m^l\right)} \tag{14}$$

◆ Model the conditional intensity function using neural networks.

$$\lambda(t) = \exp\left(v^\top \cdot \theta_m^t + w \cdot \xi_m + b\right) \tag{15}$$

◆ Derive the corresponding conditional density function based on a conditional intensity function.

$$f(t) = \lambda(t) \exp\left(-\int_{t_m}^{t} \lambda(\tau) d\tau\right) \tag{16}$$

◆ Define the loss function as a combination of time prediction loss and mobility prediction loss.

$$L = -\sum_{s_t^u \in S_u} \sum_{m=1}^{|s_t^u|-1} \left(\log P\left(l_{m+1} | \mathcal{H}_{t_m}\right) + \log f\left(t_{m+1}\right)\right) \tag{18}$$

Training and Inference

- Inference

  - Sort and pick top- K locations with the highest probabilities as the predicted locations for mobility prediction.

  - For time prediction, the predicted time for the next location is calculated using the following integration

$$\hat{t}_{m+1} = \mathbb{E}[\hat{t}_{m+1} | \mathcal{H}_{t_m}] = \int_0^\infty t \cdot f(t) dt \qquad (19)$$

  - The integration does not have a closed-form solution.
    Use numerical methods to approximate the integration.