Trajectory-User Link with Attention Recurrent Networks

Tao Sun[1,2], Yongjun Xu[1], Fei Wang[1], Lin Wu[1], Tangwen Qiany[1,2], Zezhi Shao[1,2]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

[2] University of Chinese Academy of Sciences, Beijing, China

Email: {suntao, xyj, wangfei, wulin, qiantangwen, shaozezhi16}@ict.ac.cn

Outline

- Introduction of TUL
- Motivation
- Method Description
- Experiment & Conclusion

Introduction of TUL

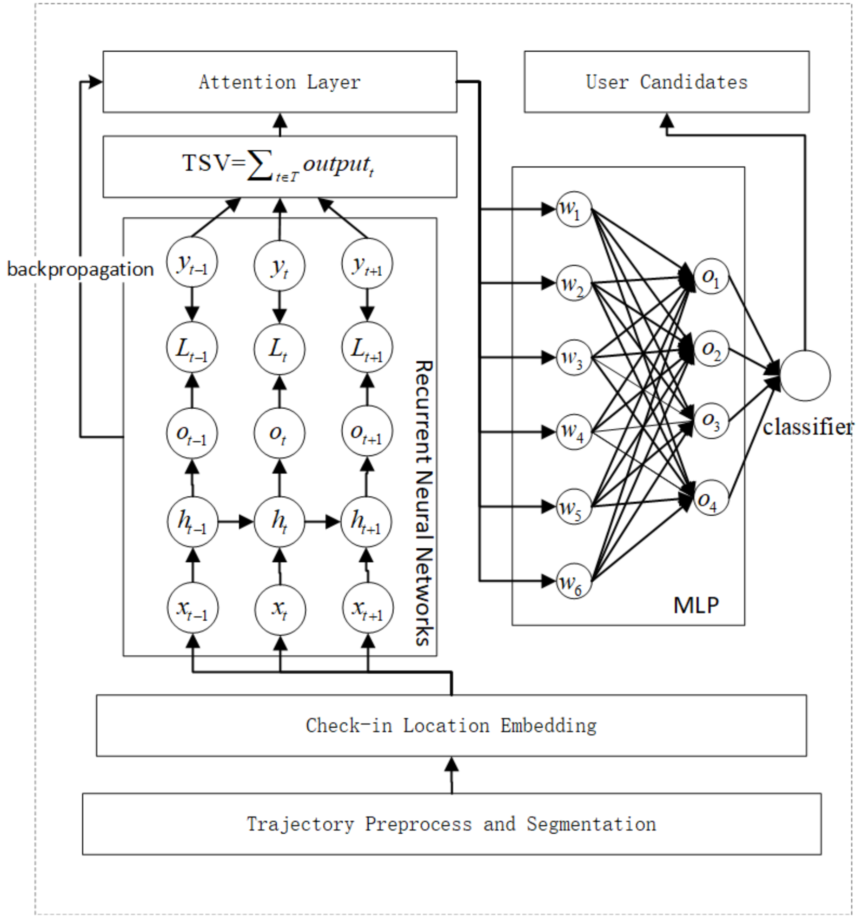
- **Trajectory User Link:** identify anonymous trajectories and link them to users who generate them.
  - Trajectory clustering based TUL
  - Recurrent neural networks based TUL

Motivation

- The check-in embedding process enhances the semantic information for locations meanwhile maps the different location in the similar vectors, which results in hardly distinguish the accompanied patterns.

- As the growing of trajectory length, it will be harder to model whole trajectory information for recurrent neural networks.

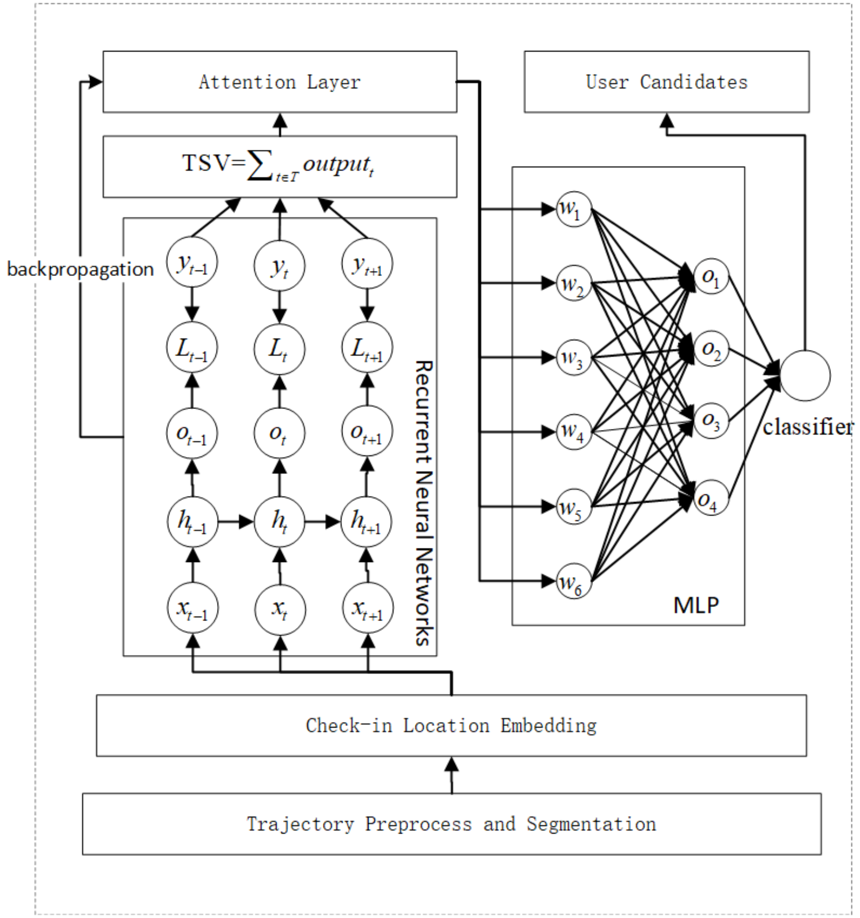
Method Description



Location Embedding:

$$
\begin{aligned}
p\left(v\left(p_i\right) | C(p_i, p)\right) &= \prod_{p' \in C(p_i, p)} p\left(v\left(p_i\right) | v\left(p'\right)\right) \\
&= \prod_{p' \in C(p_i, p)} \frac{\exp\left\{v\left(p_i\right) \cdot v\left(p'\right)\right\}}{\sum_{p'' \in C(p_i, p)} \exp\left\{v\left(p''\right) \cdot v\left(p'\right)\right\}}
\end{aligned}
$$

Method Description



Recurrent Neural Networks:

$$f_t = \sigma_g \left(W_f v\left(p_t\right) + U_f h_{t-1} + b_f \right)$$

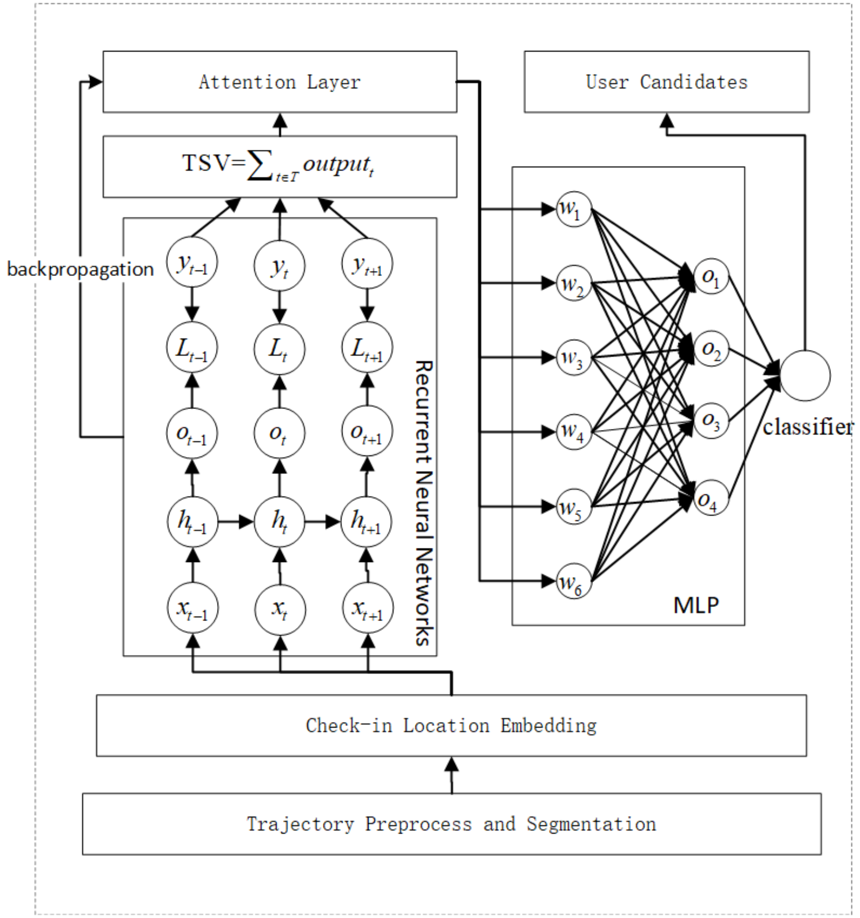
$$i_t = \sigma_g \left(W_i v\left(p_t\right) + U_i h_{t-1} + b_i \right)$$

$$o_t = \sigma_g \left(W_o v\left(p_t\right) + U_o h_{t-1} + b_o \right)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c \left(W_c v\left(p_t\right) + U_c h_{t-1} + b_c \right)$$

$$h_t = o_t \circ \sigma_h \left(c_t \right)$$

Method Description



Attention Layer:

$$TSV = \frac{1}{n}\sum_{t=1}^{n} output_t$$

$$score\left(h_t, \bar{h}_s\right) = {h_t}^T W \bar{h}_s$$

$$TSV_{attention} = \sum_{s} a_{ts}\bar{h}_s$$

$$\mathcal{L} = cross\_entropy\left(V_u, softmax\left(MLP\left(TSV\right)\right)\right)$$

Experiment

- Datasets:
  - Gowalla: 201 users and 19968 trajectories.
  - Brightkite: 92 users and 19904 trajectories.
- Baseline:
  - TULER: TULER uses the check-in embedding to enhance the check-in locations information and employs RNNs to model the sequence features. TULER employs LSTM, GRU and Bi-LSTM as RNN models which called: TULER-LSTM, TULER-GRU, TULER-LSTMS, TULER-GRU-S and Bi-TULER.
  - TULVAE: TULVAE learns the human movements in a neural generative architecture with stochastic latent variables than span hidden states in RNN. TULVAE includes HTULER-L, HTULER-G, HTULER-B and TULVAE. TULVAE was the state-of-the-art method for TUL problem.

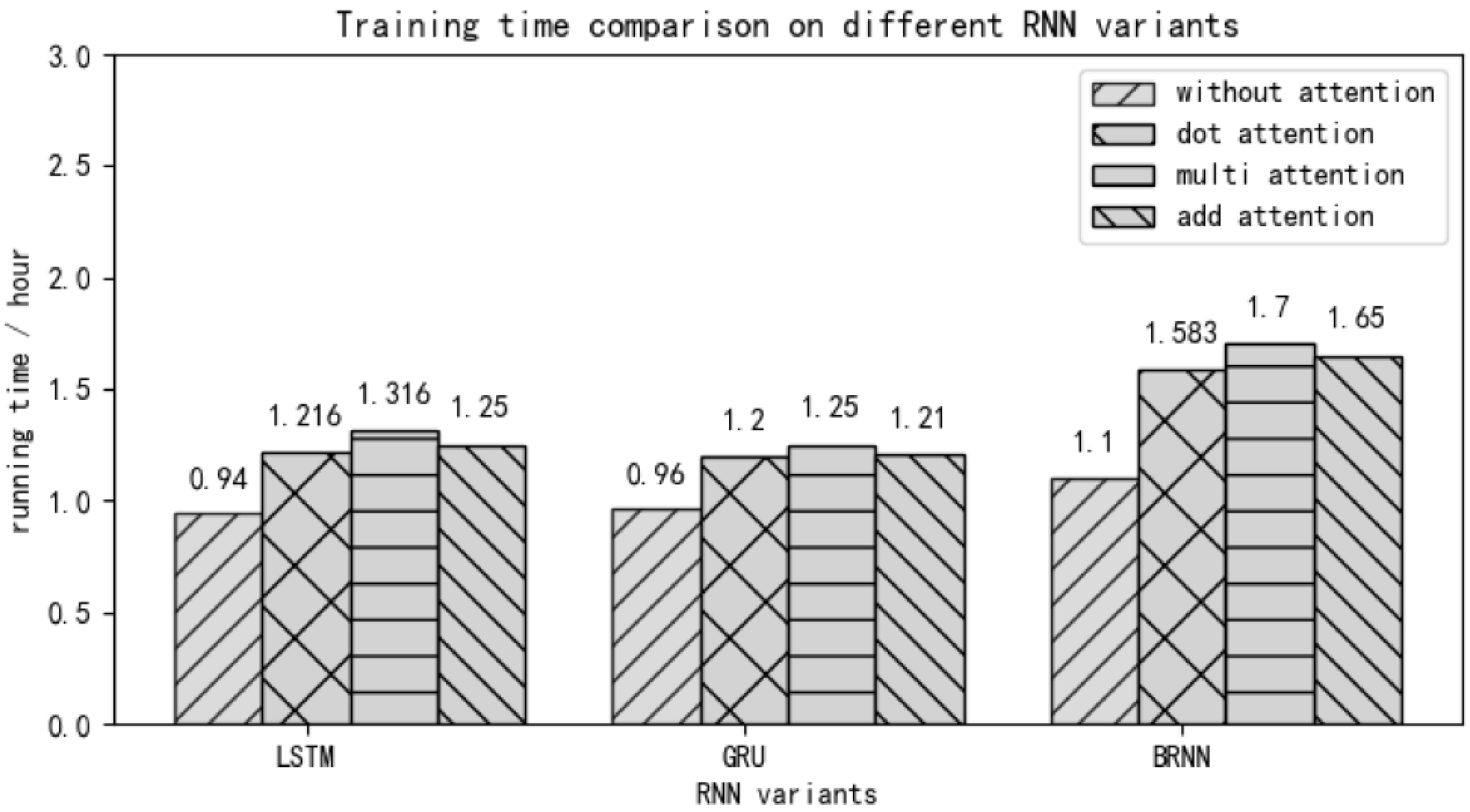
Experiment

Method \ Metric	Acc@1	Acc@5	macro-P	macro-R	macro-F1	Acc@1	Acc@5	macro-P	macro-R	macro-F1
	$\lvert U\rvert = 112$					$\lvert U\rvert = 201$				
TULER-LSTM	41.79	57.89	33.61	31.33	32.43	41.24	56.88	31.70	28.60	30.07
TULER-GRU	42.61	57.95	35.22	32.69	33.91	40.85	57.31	29.52	27.80	28.64
TULER-LSTM-S	42.11	58.01	33.49	31.97	32.71	41.22	57.70	29.34	28.68	29.01
TULER-GRU-S	41.35	58.45	32.51	31.79	32.15	41.07	57.49	29.08	27.17	28.09
Bi-TULER	42.67	59.54	37.55	33.04	32.15	41.95	57.58	32.15	31.66	31.90
HTULER-L	43.89	60.90	35.95	34.32	35.12	43.40	60.25	34.43	33.63	34.02
HTULER-G	43.33	60.74	37.71	34.47	36.01	42.88	59.41	32.72	32.54	32.63
HTULER-B	44.21	62.28	36.48	33.51	34.93	44.50	60.93	34.89	34.46	34.67
TULVAE	44.35	64.46	40.28	32.89	36.21	45.40	62.39	36.13	34.71	35.41
TULAR-LSTM-M	48.74	70.02	45.45	40.73	42.95	47.47	67.61	41.20	38.69	39.91
TULAR-LSTM-A	48.42	70.02	45.42	40.06	42.57	47.79	67.70	40.26	39.22	39.73
TULAR-LSTM-D	48.00	68.65	47.17	41.38	44.08	47.26	67.08	38.65	38.72	38.69
TULAR-GRU-M	46.22	66.87	43.45	38.13	40.61	45.79	64.35	36.89	36.98	36.93
TULAR-GRU-A	46.22	69.28	40.58	40.16	40.37	46.26	66.29	36.86	37.96	37.40
TULAR-GRU-D	47.48	67.92	40.90	38.46	39.64	45.11	64.30	37.36	37.35	37.36
TULAR-BRNN-M	48.63	69.49	46.18	40.72	43.28	48.00	67.29	40.22	39.78	40.00
TULAR-BRNN-A	48.21	70.85	43.85	41.31	42.41	**48.84**	**66.82**	**40.38**	**39.64**	**40.01**
TULAR-BRNN-D	**49.05**	**70.12**	**43.26**	**40.91**	**42.05**	48.21	67.40	41.51	40.24	40.87

Experiment

Method \ Metric	Acc@1	Acc@5	macro-P	macro-R	macro-F1	Acc@1	Acc@5	macro-P	macro-R	macro-F1
	$\lvert U\rvert = 34$					$\lvert U\rvert = 92$				
TULER-LSTM	48.26	67.39	49.90	47.20	48.51	43.01	59.84	38.45	35.81	37.08
TULER-GRU	47.84	67.42	48.88	46.87	47.85	44.03	61.36	38.86	36.47	37.62
TULER-LSTM-S	47.88	67.38	48.81	47.03	47.62	44.23	61.00	38.02	36.33	37.16
TULER-GRU-S	48.08	68.23	48.87	46.74	47.78	43.93	61.85	37.93	36.01	36.94
Bi-TULER	48.13	68.17	49.15	47.06	48.08	43.54	60.68	38.20	36.47	37.31
HTULER-L	49.44	71.13	51.51	47.31	49.32	45.26	63.55	41.61	38.13	39.79
HTULER-G	49.12	70.81	51.85	46.88	49.24	44.50	63.17	41.10	37.51	39.22
HTULER-B	49.78	70.69	52.45	47.98	48.90	45.30	63.93	41.82	39.32	38.60
TULVAE	49.82	71.71	51.26	46.43	48.72	45.98	64.84	43.15	39.65	41.32
TULAR-LSTM-M	52.86	74.11	51.19	49.40	50.28	**58.45**	**76.58**	**56.56**	**52.99**	**54.72**
TULAR-LSTM-A	**53.85**	**75.69**	**50.84**	**48.32**	**49.55**	56.32	75.41	56.09	51.06	53.46
TULAR-LSTM-D	53.16	75.59	48.69	48.62	48.66	58.04	75.11	56.33	52.75	54.48
TULAR-GRU-M	52.17	73.22	47.17	46.60	46.88	55.66	74.20	56.21	51.11	53.54
TULAR-GRU-A	52.56	72.92	52.97	49.08	50.95	55.61	74.35	54.17	50.72	52.39
TULAR-GRU-D	53.06	72.82	49.34	47.44	48.37	56.42	73.69	57.20	52.92	54.98
TULAR-BRNN-M	53.35	75.00	50.06	49.02	49.54	57.38	75.46	58.66	51.66	54.94
TULAR-BRNN-A	53.75	76.67	50.28	47.62	48.91	57.08	75.21	56.63	51.15	53.75
TULAR-BRNN-D	53.65	73.61	49.23	48.63	48.93	57.84	76.02	58.55	53.32	55.81

Experiment



Training time comparison on different RNN variants
3.0
2.5
2.0
1.5
1.0
0.5
0.0
running time / hour
without attention
dot attention
multi attention
add attention
0.94
1.216
1.316
1.25
0.96
1.2
1.25
1.21
1.1
1.583
1.7
1.65
LSTM
GRU
BRNN
RNN variants

Conclusion

- A novel Trajectory-User Link method, TULAR, is proposed, which improves both accuracy and efficiency for TUL problem. TULAR is an end-to-end trajectory identifying neural network framework, with superb scalability.
- TULAR introduces TSV, a representation learning method, mapping the variable-length trajectories to fixedlength vectors in feature space. Trajectories of the same user are likely more similar to each other in feature space, which is untenable in primitive geospatial space.
- Three different trajectory embedding methods and three different attention score measures are used in this paper. A lots of experiments are conducted to demonstrate our improvements, using four real world datasets and compared with several state-of-art methods.

Thanks for Listening!