How Do Your Neighbors Disclose Your Information: Social-Aware Time Series Imputation

Zongtao Liu†, Yang Yang†, Wei Huang†, Zhongyi Tang‡, Ning Li‡, Fei Wu†

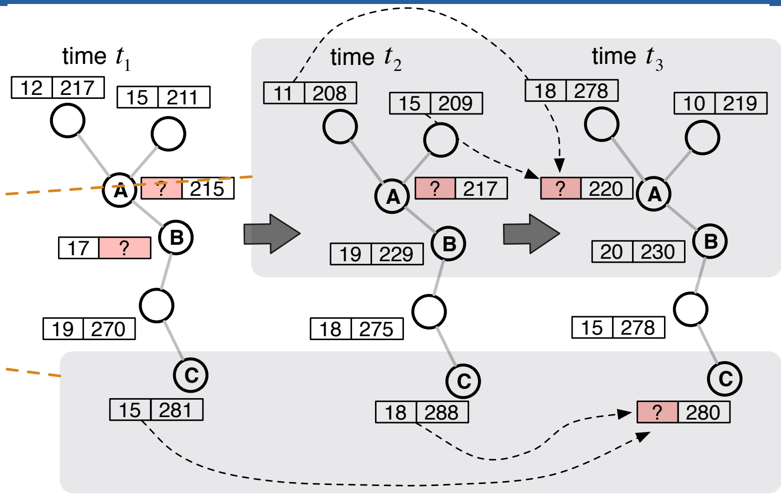
†Zhejiang University , ‡State Grid Zhejiang Jinyun Power Supply Co. Ltd

{tomstream, yangya}@zju.edu.cn

Social-Aware Time Series Imputation Problem

In a social network, how can we infer missing records of users?

1. **Surrounding influence:** how to model the connection between the missing observations and social context.
2. **Temporal influence:** how to model the connection between the missing observations and temporal context.
3. How to handle **irregular time intervals**.



Our Approach

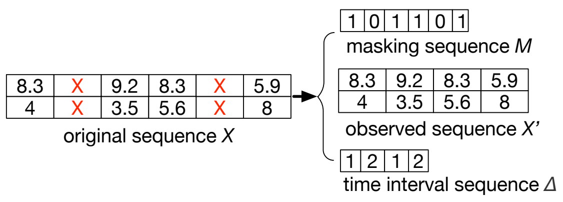
Definition

social network: $G = < V, E >$

behavior data : $X = \{x_1, x_2, ..., x_T\}$

observed data : $X' = \{x_{s_1}, x_{s_2}, ..., x_{s_L}\}$

time intervals : $\delta_l = \begin{cases} 1, & l = 1 \\ s_l - s_{l-1}, & l \neq 1 \end{cases}$



Time gap-aware LSTM (T-LSTM)

In encoding step, we use a variant LSTM to handle irregular time gaps.

The original memory cell is replaced by:

$$c_t^s = tanh(W_d c_{t-1} + b_d)$$

$$\hat{c}_t^s = c_{t-1}^s \cdot \boxed{g(\delta)}$$ decaying function

$$c_{t-1}^l = c_{t-1} - c_{t-1}^s$$

$$c_{t-1}^* = c_{t-1}^l + \hat{c}_t^s$$

$$\tilde{c} = tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$c_t = f_t \cdot c_{t-1}^* + i_t \cdot \tilde{c}$$

General idea: how neighbors' behaviorial patterns can relate to her current state.

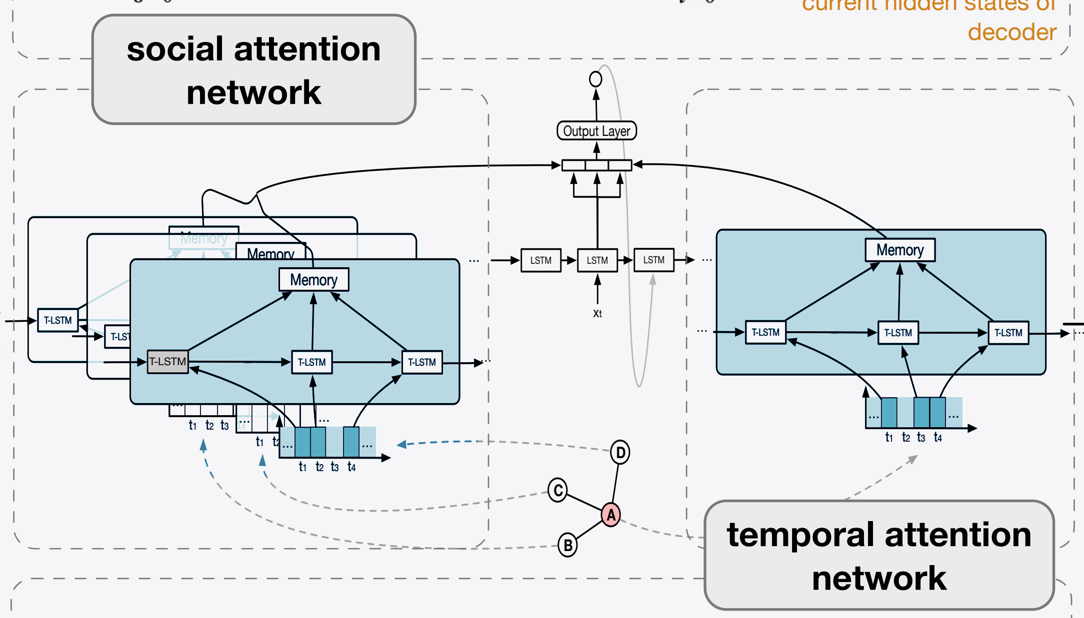
· **Encode neighbor's behavioral data:**

$$h_{s(p)}, c_{s(p)} = T - LSTM(x'_{s(p)}, \delta_{s(p)}, h_{s-1(p)}, c_{s-1(p)})$$

· **Extract social context $\hat{a}$ by a memory-based attention mechanism:**

$$C_k = \sum_{s=0}^{|S|} \alpha_{sk} h_s, \alpha_{sk} = softmax(W_\alpha h_s \cdot l_s) \quad \hat{a} = \sum_{i=0}^{K} \hat{\beta}_i \tilde{C}_i, \hat{\beta} = softmax(W_{\hat{\beta}} \boxed{h^*})$$

current hidden states of decoder



General idea: how a user's historical and future behaviors can relate to her current state.

· **Encode a user's behavioral data:**

$$h_s, c_s = T - LSTM(x'_s, \delta_s, h_{s-1}, c_{s-1})$$

· **Compute the memory matrix and extract temporal context a:**

$$a = \sum_{i=0}^{K} \beta_i C_i, \beta = softmax(W_\beta h^*)$$

Learning and Imputation

Predict the targets:

$$x_t^* = \phi(h_t^*, \hat{a}_t, a_t)$$

Loss function:

$$\mathcal{L}(X^N, X^{*N}) = \sum_{n=1}^{N} [\sum_{t=1}^{T} \sum_{d=1}^{D} m_t^{(n)} \times (x_t^{d(n)} - x_t^{*d(n)})^2)]$$

Training:

1. Draw a mini-batch of sequences and their neighbors' data;
2. Compute social context and temporal context;
3. For each input in decoding step, sample $p \sim \mathcal{U}(1)$:

if $p > \gamma$ **then** $x' = x_{t-1}^*$

else $x' = \boxed{x_{t-1}^*} \cdot (1 - m_{t-1}) + x_{t-1} \cdot m_{t-1}$

predicted value

4.Compute loss and apply updates.

Imputation:

for each input in decoding step :

$$x' = x_{t-1}^* \cdot (1 - m_{t-1}) + x_{t-1} \cdot m_{t-1}$$

Experimental Results

Scenario

Given a user v, her neighbors are people whose living places are close to v.

Datasets

Electrical Consumption (EC) : Time series of daily electrical usage recorded by 80,000 watt-hour meters. Each series has 90 timestamps.

Real-Time Voltage (RV) : Electricity load series, each of which describes voltage values in three phases. Each series has 32 timestamps.

Tasks

Randomly Missing: Elements are randomly dropped with a missing rate.

Simulated Missing: An element is dropped if there exists a missing elements after 90 days (only on EC dataset).

Results with Simulated Missing (EC):

Method	MAE	RMSE	Method	MAE	RMSE
Mean	2.7626	4.1134	Median	2.8156	4.4493
Linear	1.7112	2.9973	Cubic	9.2609	67.5511
KNN	2.5144	3.9050	SoftImpute	2.5384	3.9342
MICE	2.8304	4.3208	MissForest	3.2628	4.9611
VAE	1.7067	3.0243	LSTM-Impute	2.4445	3.8235
GRU-D	1.9298	3.3543	STI - s	1.6223	2.6731
STI	**1.5837**	**2.6412**			

Results with Randomly Missing

Dataset	Missing Rate	0.2		0.3		0.4		0.5		0.6	
	Method	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
EC	Mean	3.3787	4.3235	3.3794	4.3263	3.3810	4.3295	3.3850	4.3375	3.3913	4.3498
	Median	3.2818	4.5337	3.2850	4.5394	3.2905	4.5478	3.3015	4.5654	3.3151	4.5838
	Linear	1.5783	2.5173	1.6246	2.5835	1.6674	2.6431	1.7249	2.7246	1.7972	2.8248
	Cubic	2.0246	3.1914	2.1461	3.4118	2.2667	3.6288	2.4358	4.0081	2.6691	4.7918
	KNN	2.2455	3.3251	2.4224	3.5077	2.5762	3.6617	2.7576	3.8407	2.9672	4.0431
	SoftImpute	2.4018	3.5193	2.6459	3.7814	2.8377	3.9767	2.9746	4.1007	3.0319	4.1303
	MissForest	4.0659	5.3842	4.0528	5.3695	4.0474	5.3664	4.0294	5.3412	4.0068	5.3174
	MICE	3.4634	4.5654	3.4590	4.5777	3.4578	4.5919	3.4538	4.6152	3.4550	4.6591
	VAE	1.5375	2.3085	1.5883	2.4382	1.6504	2.4979	1.6882	2.6148	1.7374	2.6515
	LSTM-Impute	3.0315	4.2238	3.1687	4.3324	3.2529	4.3206	3.4526	4.5627	3.7708	4.7990
	GRU-D	1.7024	2.5568	1.9385	2.7868	2.0511	2.9136	2.0780	2.9304	1.9568	2.8918
	STI - t - s	1.5066	2.3134	1.5384	2.4002	1.5822	2.4175	1.5903	2.4510	1.6851	2.5350
	STI - s	**1.4628**	2.2337	1.4985	2.3364	1.5463	**2.3432**	**1.5672**	2.4208	1.6161	2.4593
	STI	1.4667	**2.2172**	**1.4864**	**2.2574**	**1.5207**	2.3745	1.5696	**2.3924**	**1.6159**	**2.4505**
RV	Mean	4.0893	5.0340	4.0957	5.0435	4.1076	5.0581	4.1184	5.0835	4.1547	5.1397
	Median	4.0250	5.2811	4.0465	5.2929	4.0701	5.3301	4.0975	5.3541	4.1594	5.4246
	Linear	2.0697	3.4058	2.1316	3.4778	2.2179	3.5714	2.3255	3.7051	2.5487	3.9549
	Cubic	2.7329	4.4551	2.8801	4.7857	3.0976	5.3014	3.3495	5.8316	3.9971	7.7123
	KNN	3.1175	4.3509	3.3162	4.5230	3.5550	4.7334	3.8224	4.9665	4.1645	5.2793
	SoftImpute	4.0263	5.1599	5.4152	6.9389	6.4592	8.4186	6.4171	8.4777	5.3860	7.0291
	MissForest	4.1727	5.3729	4.1825	5.3942	4.2012	5.4243	4.2203	5.4701	4.2952	5.5940
	MICE	4.3518	5.7909	4.3806	5.8305	4.4099	5.8764	4.4302	5.9083	4.4641	5.9477
	VAE	2.3001	3.2631	2.7272	4.5136	3.3440	6.4581	3.6293	6.7901	4.4053	8.8703
	LSTM-Impute	3.0315	4.2238	3.1687	4.3324	3.2529	4.3206	3.4526	4.5627	3.7708	4.7991
	GRU-D	2.8582	4.1190	3.0640	4.3150	3.1822	4.3652	3.1583	4.4811	3.5772	4.7590
	STI - t - s	2.0641	3.0035	2.1920	3.1756	2.2661	3.2115	2.3573	3.3520	2.6095	3.7819
	STI - s	2.0487	3.0071	2.1167	3.1362	2.1383	3.1392	2.2912	3.3465	2.5390	3.6977
	STI	**2.0008**	**2.9426**	**2.0787**	**3.0858**	**2.1258**	**3.1306**	**2.2795**	**3.3187**	**2.4963**	**3.5972**

Code: https://github.com/tomstream/STI