搜索算法实习生转正答辩

刘宗涛

2019.08

个人简介

- 2017-2020 浙江大学 计算机科学与技术专业硕士
- 2013-2017 浙江大学 计算机科学与技术专业学士
- 研究领域：数据挖掘
- 论文发表：AAAI, WWW
- 技能：C/C++, Python, SQL
- 实习内容：抖音点击标签修正，ctr+playtime多目标预估模型

学术研究

- 城市移民融入(AAAI'18)
- 早期移民流失(WWW'18)
- 社交感知的时序补全(WWW'19)
- 强化学习派工规划方法

实习内容

- 建立Doc信息收集流程
- 修正抖音点击标签
- 调研ctr + 播放时长多目标模型
- 制作离线训练数据并例行化

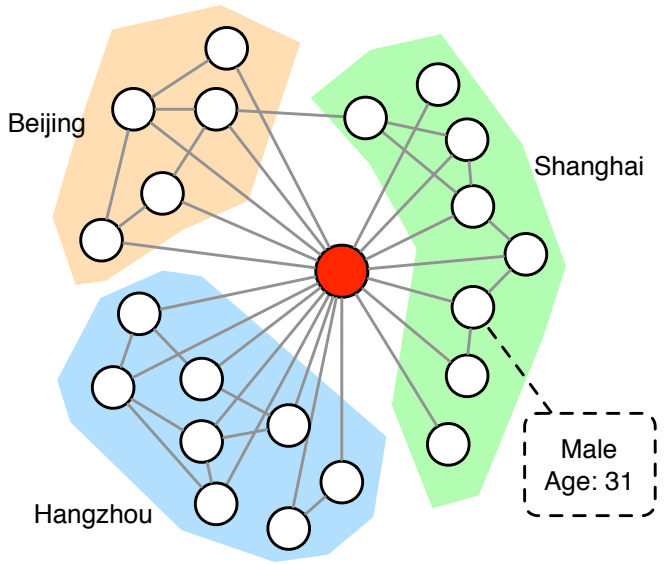
移居者的都市梦：城市移民群体行为研究

- 问题
  - 移民群体和本地人群在行为模式上存在怎样的差异？
  - 这些差异多大程度能帮助我们区分这两类群体？
  - 是否能衡量移民者的融入程度？

- Yang Yang, Chenhao Tan, Zongtao Liu, Fei Wu, and Yueting Zhuang. Urban Dreams of Migrant: A Case Study of Migrant Integration in Shanghai. **AAAI'18.**

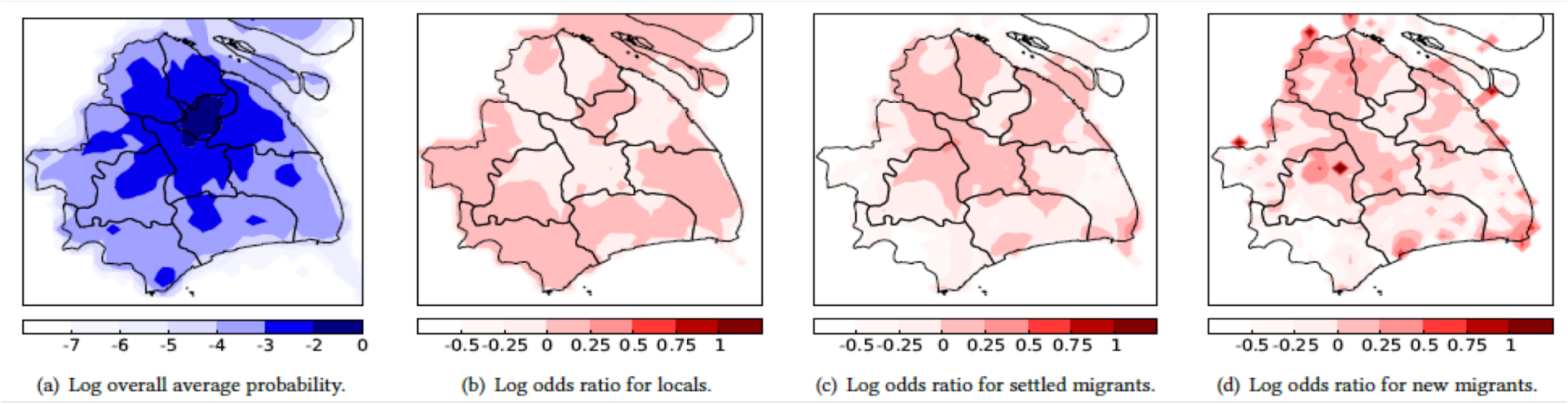
用户通话网络

- 数据来源：2016年9月上海电信全网通话元数据
  - 通话记录 + 基站GPS信息
  - 7亿条通话记录，5400万用户



不同群体的行为模式差异

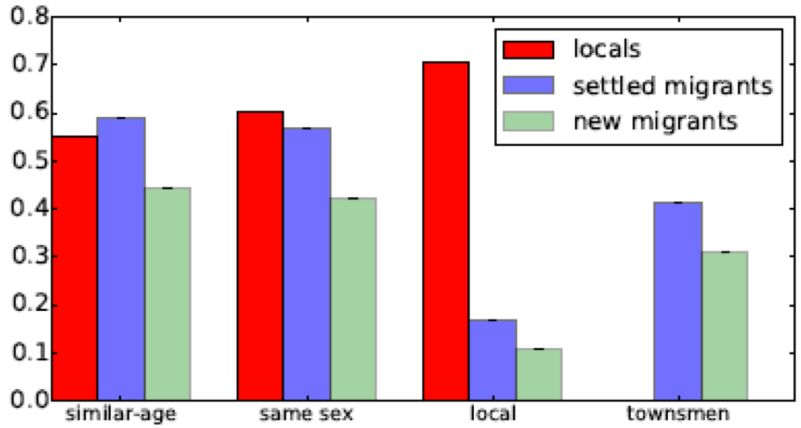
- **本地人**：出生在本地的上海人
- **老移民者**：在上海已经生活了一段时间、安顿下来了的移民者
- **新移民者**：刚来上海一周的移民者



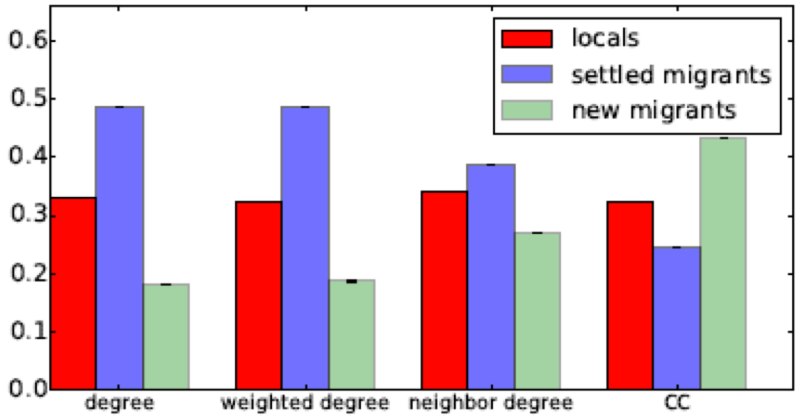
(a) Log overall average probability. (b) Log odds ratio for locals. (c) Log odds ratio for settled migrants. (d) Log odds ratio for new migrants.

本地人：1.7M，**老移民者**：1.0M，**新移民者**：34K

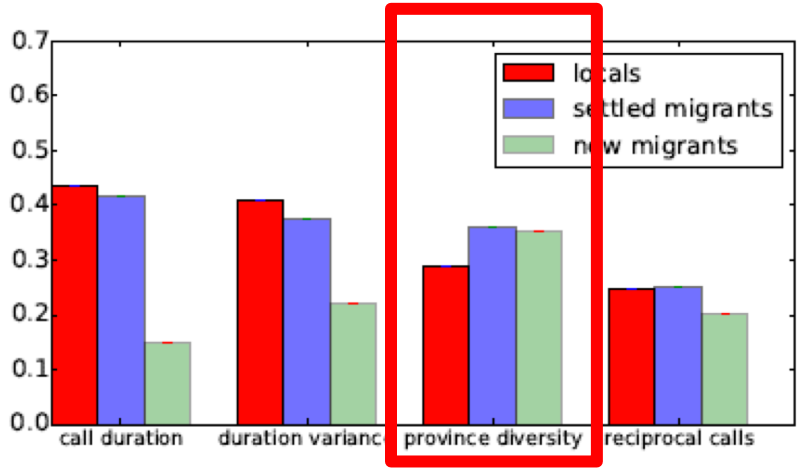
移民者有更多元的人际关系，更大的活动区域



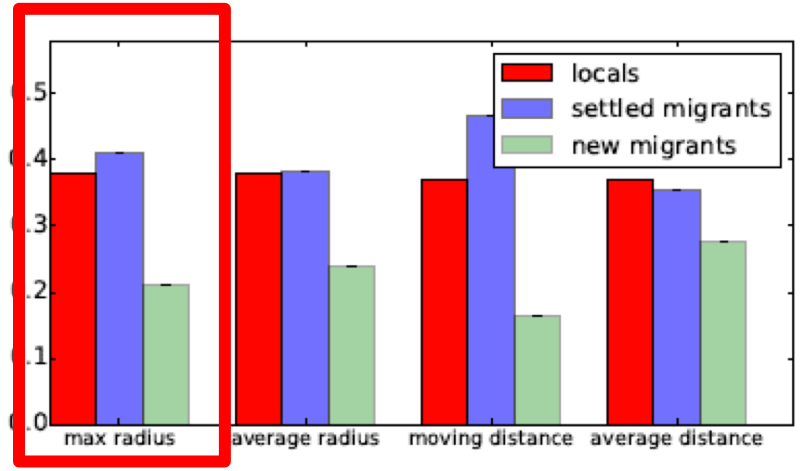
(a) Demographics of friends.



(b) Ego-network characteristics.



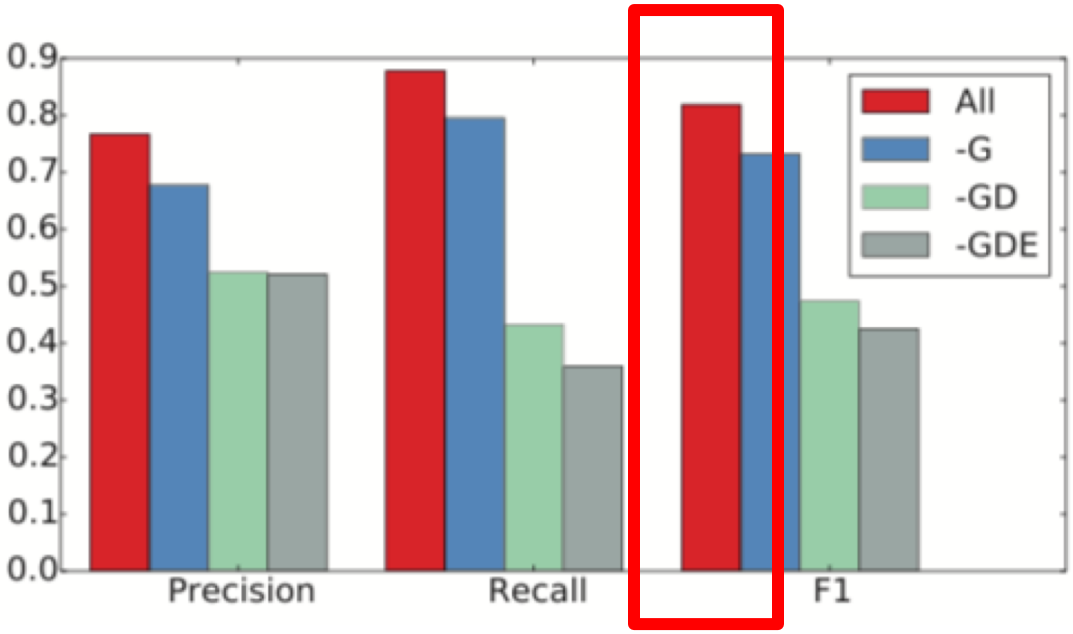
(c) Call behavior.

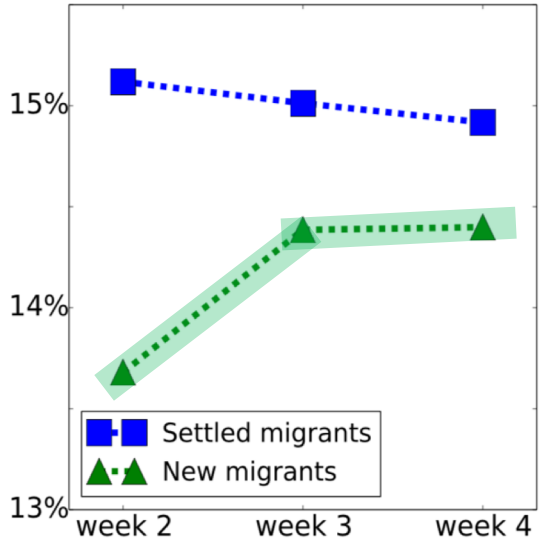


(d) Geographical features.

伴随居住时间的增加，

- 二分类：根据首周用户通话记录，判断其为本地人, 老移民者

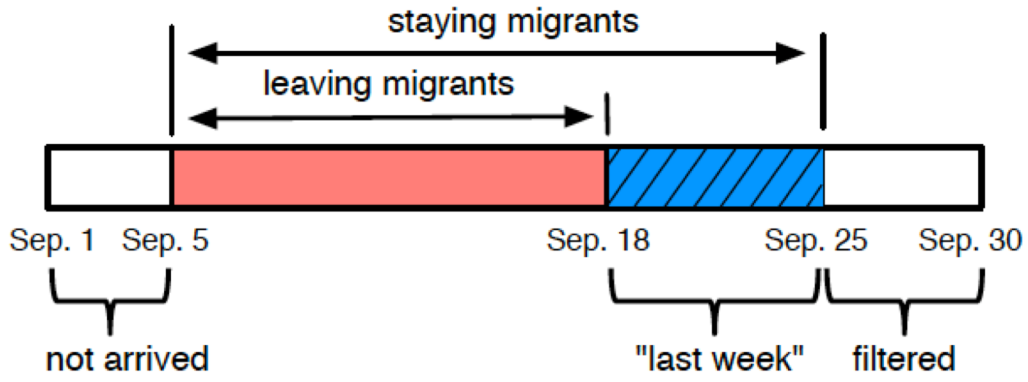




老移民者与新移民者中，被误判为本地人的比例

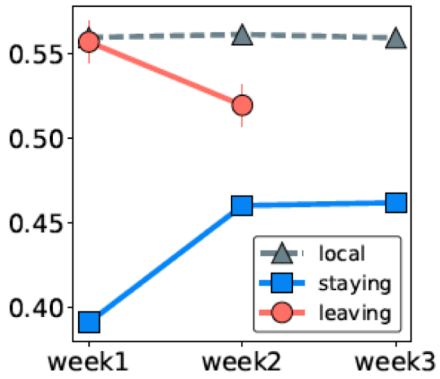
什么因素导致移民者离开都市？

- 抵达都市后的**前两周**很关键！
- 进一步将新移民者划分为离开都市的人和留在都市的人
- 2016年9月下旬，**4%**的新移民者离开了上海

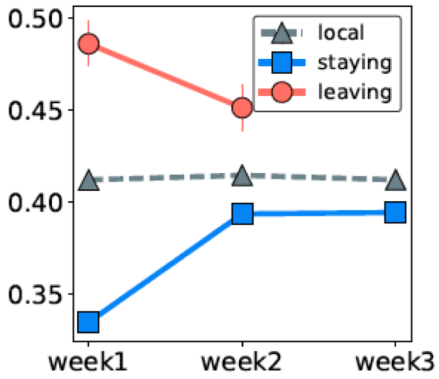


- Yang Yang, Zongtao Liu, Chenhao Tan, Fei Wu, and Yueting Zhuang. To Stay or to Leave: Exploring the Initial Period of Migrant Integration. **WWW'18.**

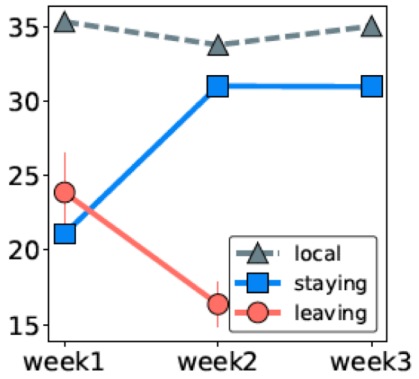
初步构建当地关系网络



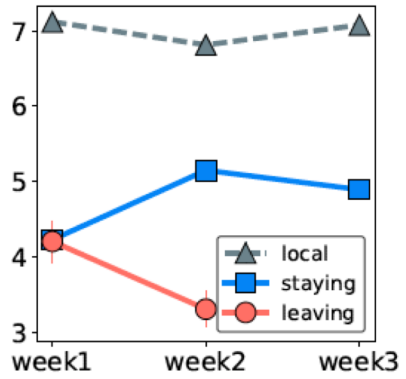
(a) Proportion of same sex contacts.



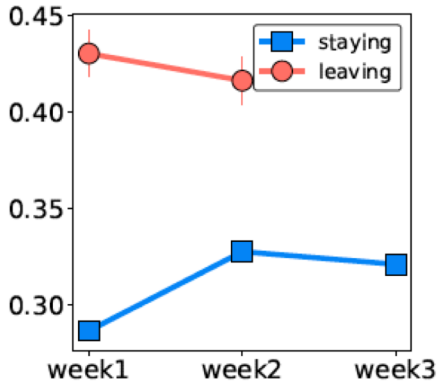
(b) Proportion of simiarly aged contacts.



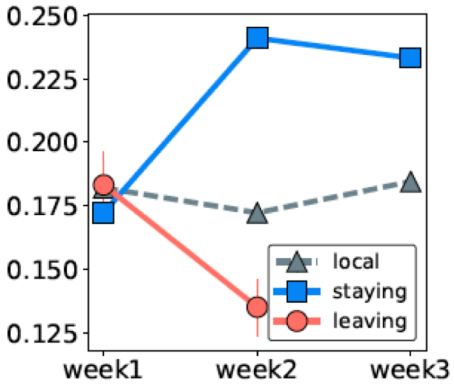
(c) Degree.



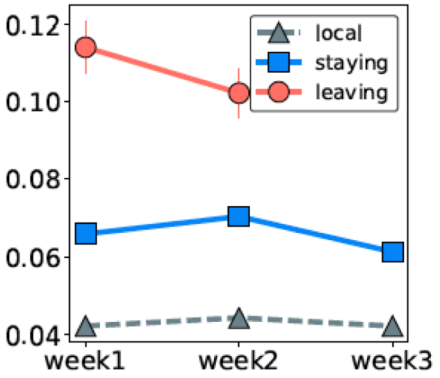
(d) Average degree of contacts.



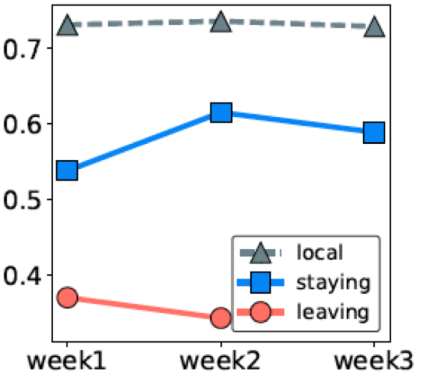
(e) Proportion of townspeople.



(f) Province diversity.



(g) Clustering coefficient.

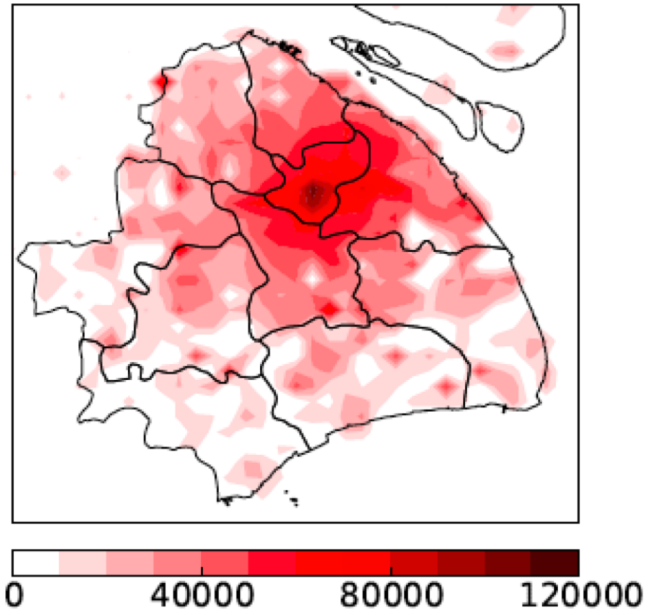


(h) Communication diversity.

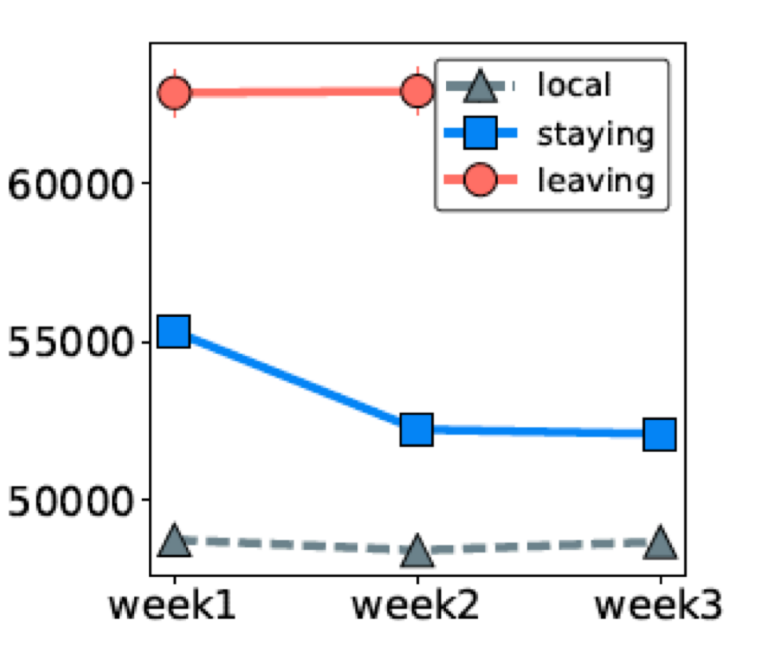
积极扩展人脉、发展**多样性**的关系与移民者能否留在都市的关联性很强

找到合适的居住地

- 根据用户的GPS数据，挖掘其居住地，结合上海市房地产数据，分析用户居住地的房价



(a)上海市房价热点分布



(b)用户居住地的平均房价

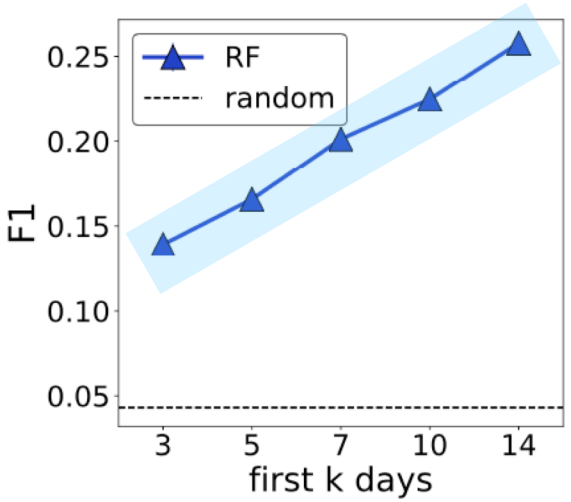
流失移民 vs. 留存移民

- 根据新移民者过去两周的通话数据，预测其两周后是否会离开上海

Feature sets	Precision	Recall	F1
all features	0.1597	0.6659	0.2576
ego network properties	0.1347	0.6580	0.2234
housing price information	0.1067	0.5978	0.1809
call behavior	0.0984	0.5853	0.1683
geographical information	0.0863	0.5691	0.1498

流失移民 vs. 留存移民

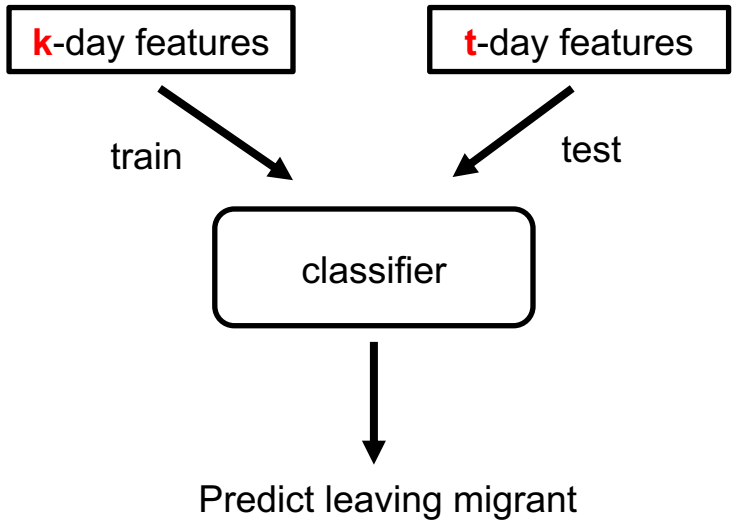
- 在移居早期识别移民的流失
  - 是否能在小于两周的时间识别流失的移民?
    - 如果可以, 政府和公益团体也许能为这类移民提供帮助
  - 基于前k天数据提取特征进行训练和预测：

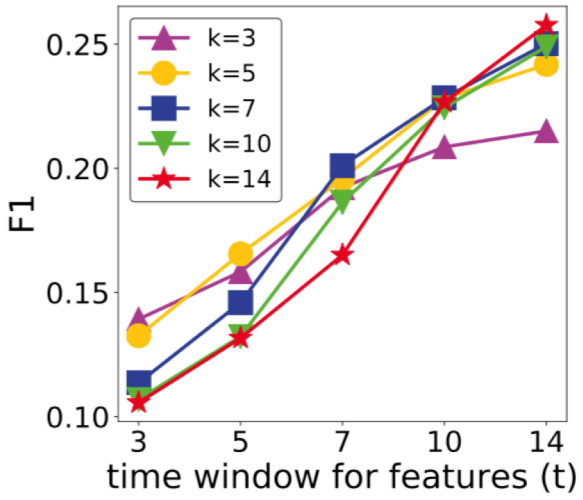


(c) F1.

流失移民 vs. 留存移民

- 探究预测效果提升的原因
  - 解耦可能导致效果提升的两个因素：模型和特征





(d) Disentangling performance improvement.

使用5天数据，分类器就能达到用14天数据的预测效果！

基于社交感知的时序补全

- 时序数据补全
  - 时序数据缺失会影响基于这类数据的分析和建模，研究合适的补全方法十分有必要
  - 时序补全的方法
    - 插值，平滑
    - 深度模型：GRU-D，LSTM-impute
- 社交网络中的时间序列
  - 在社交网络分析中，时序数据也起着重要作用
  - 基于同质性(Homophily)现象，联系人的行为模式可以帮助他的数据缺失
  - 目前缺乏结合社交上下文和深度模型来进行时间序列补全的工作

- **Zongtao Liu**, Yang Yang, Wei Huang, Zhongyi Tang, Ning Li and Fei Wu. How Do Your Neighbors Disclose Your Information: Social-Aware Time Series Imputation. **WWW'19.**

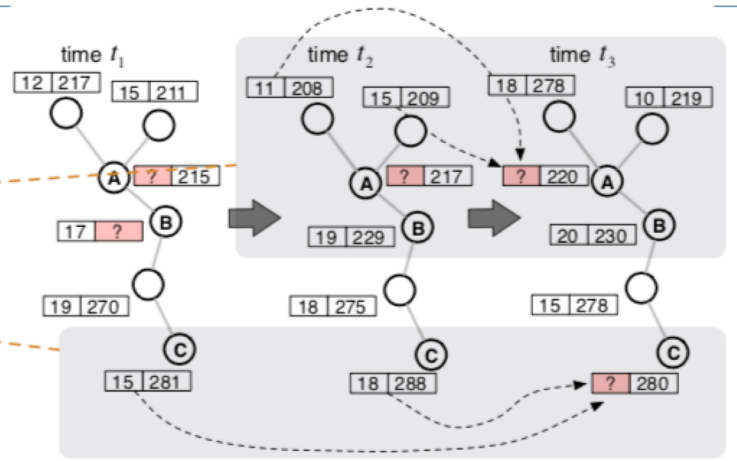
Introduction

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**.

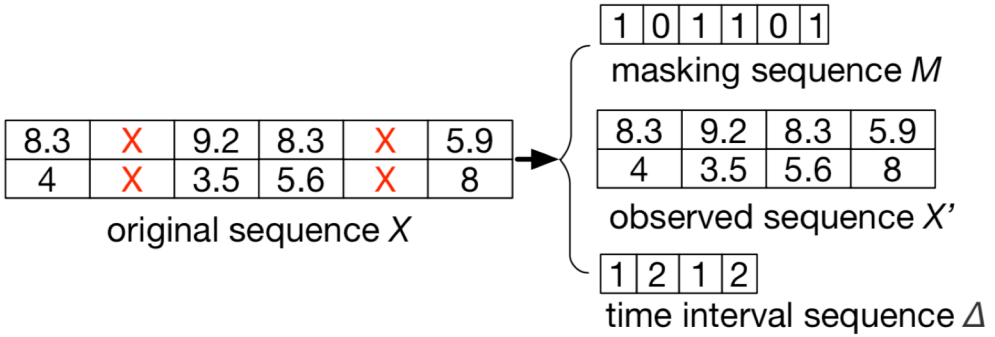


social network: $G =< V, E >$

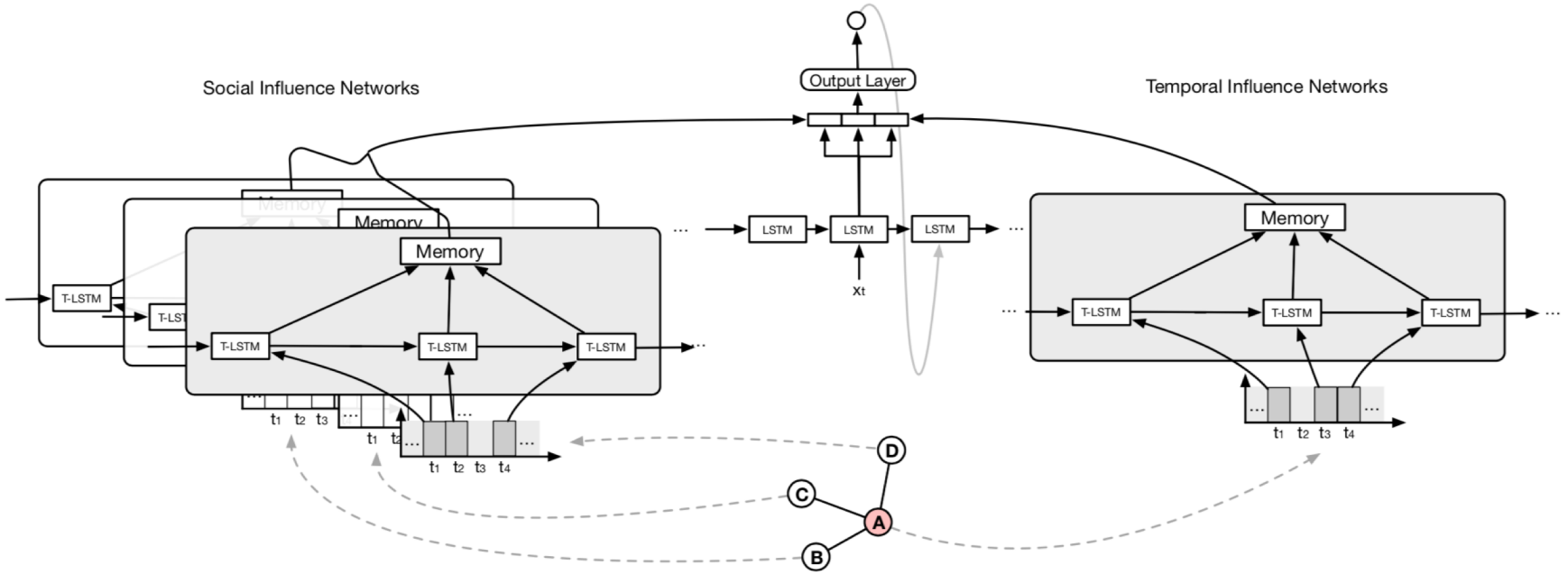
behavior data : $X = \{x_1, x_2, ..., x_T\}$

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

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



Our Approach



Time-Gap Aware LSTM

- 运用T-LSTM对时间间隔不等的时间序列进行建模

$$
\begin{aligned}
g_t &= tanh(W_g x_t + U_g h_{t-1} + b_g) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_f) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot g_t \\
h_t &= o_t \cdot tanh(c_t)
\end{aligned}
$$

$$\Longrightarrow$$

$$
\begin{aligned}
c_t^s &= tanh(W_d c_{t-1} + b_d) \\
\hat{c}_t^s &= c_{t-1}^s \cdot g(\delta) \\
c_{t-1}^l &= c_{t-1} - c_{t-1}^S \\
c_{t-1}^* &= c_{t-1}^l + \hat{c}_t^s \\
\tilde{c} &= tanh(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \cdot c_{t-1}^* + i_t \cdot \tilde{c}
\end{aligned}
$$

Social Attention Network

- For each user v, we have a group of sequences $\{\hat{X}_{(1)}, \hat{X}_{(2)}, \ldots, \hat{X}_{(P)}\}$ from her neighbors $\{\hat{v}_{(1)}, \hat{v}_{(2)}, \ldots, \hat{v}_{(P)}\}$
- We extract their corresponding observed sequences $\{\hat{X'}_{(1)}, \hat{X'}_{(2)}, \ldots, \hat{X'}_{(P)}\}$ and interval sequences $\{\hat{\Delta}_{(1)}, \hat{\Delta}_{(2)}, \ldots, \hat{\Delta}_{(P)}\}$
- For each neighbor, we extract her hidden states $\{\hat{\mathrm{h}}_{1(p)}, \hat{\mathrm{h}}_{2(p)}, \ldots, \hat{\mathrm{h}}_{s(p)}\}$ through a T-LSTM network

- Compute fix-sized memory matrix

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

$$C_k = \sum_{s=0}^{|S|} \alpha_{sk} h_s$$

$$\alpha_{sk} = softmax(W_\alpha h_s \cdot l_s)$$

- Concatenate all neighbors memory matrices: $\tilde{C} = \{\hat{C}_{(1)}, \hat{C}_{(2)}, \ldots, \hat{C}_{(P)}\}$

Our Approach

- Social Attention Network
  - Generate social context vector $\hat{a}$:

$$h_t^*, c_t^* = LSTM(x_{t-1}^*, h_{t-1}^*, c_{t-1}^*)$$

$$\hat{a} = \sum_{i=0}^{K} \hat{\beta}_i \tilde{C}_i$$

$$\hat{\beta} = softmax(W_{\hat{\beta}} h^*)$$

Temporal Attention Network

- Temporal Attention Network
  - For each user v, we have her data X.
  - We extract their corresponding observed sequences X' and interval sequence Δ. For each neighbor, we extract her hidden states $\{h_1, h_2, \ldots, h_s\}$ through a T-LSTM network $h_s, c_s = T - LSTM(x'_s, \delta_s, h_{s-1}, c_{s-1})$
  - Generate temporal r a $a = \sum_{i=0}^{K} \beta_i C_i$ $\beta = softmax(W_\beta h^*)$
- Full connection $x_t^* = \phi(h_t^*, \hat{a}_t, a_t)$

Our Approach

- Loss function:

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

Algorithm 1 training procedure

while not converged **do**
  draw a mini-batch of sequences $\mathbf{X}^{(n)}$ and their corresponding social context sequence sets $\hat{\mathbf{X}}^{(n)}$
  // forward pass to encoder network
  compute social context vector $\hat{\mathbf{a}}^{(n)}$ and temporal context vector $\mathbf{a}^{(n)}$
  //we omit the symbol of batch size (n) in the following statements
  for t in $1, 2, ..., T$ **do**
    sample $p \sim \mathcal{U}(1)$
    if $p > \gamma$ **then**
      $h_t^*, c_t^* = LSTM(x_{t-1}^*, h_{t-1}^*, c_{t-1}^*)$
    else
      $h_t^*, c_t^* = LSTM(x_{t-1}^* \cdot (1 - m_{t-1}) + x_{t-1} \cdot m_{t-1}, h_{t-1}^*, c_{t-1}^*)$
    end if
    $x_t^* = \phi(h_t^*, \hat{a}_t, a_t)$
  end for
  compute the loss function $\mathcal{L}$
  // backward pass
  compute gradients and apply updates
end while

Experiment

- Dataset Description:
  - The two datasets in our experiment are from the State Grid.
  - Several factors about electricity data collection
    - Electricity data is measured by watt-hour meters.
    - Each meter periodically transmits different types of data to its corresponding data collector.
    - Usually, a collector receives data from several meters that are geographically close.
    - Hence, we can consider that the meters connecting to the same collector record the electricity usage of people or families with social relationships

Experiment

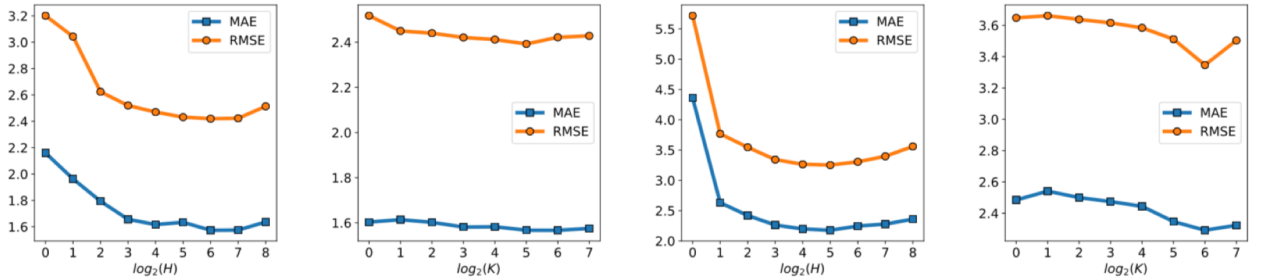
- Electrical Consumption (EC) :
  - This dataset is provided by the State Grid. In total, it includes the daily electrical consumption recorded by 80,000 watt-hour meters. This data spans from from January 1st 2018 to March 31st 2018. The length of each series is 90.
- Real-Time Voltage (RV) :
  - This dataset, provided by the State Grid, consists of around 20,000 electricity load series, each of which describes voltage values in three phases, recorded every 45 minutes in

Performance Comparison with Randomly Missing Elements

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	**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	**2.0008**	**2.9426**	**2.0787**	**3.0858**	**2.1258**	**3.1306**	**2.2795**	**3.3187**	**2.4963**	**3.5972**

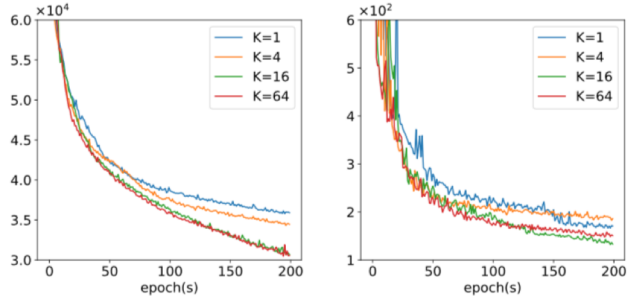
Table 1: Performance of different models in imputation using the EC and RV datasets.The best results are in boldface, and the second-best results are underlined.

Parameter Analysis



(a) Varing cell size H on EC dataset. (b) Varing memory size K on EC dataset. (c) Varing cell size H on RV dataset. (d) Varing memory size K on RV dataset.

Figure 5: Model parameter analysis. (a), (b) testing on EC dataset and (c), (d) testing on RV dataset. The Missing ratio is set as 0.5. (a), (c) presents the sensitivity of cell size H and (b), (d) shows the sensitivity of memory size K.



(a) Training curve on the EC dataset. (b) Training curve on the RV dataset.

Figure 6: Comparison concerning varing memory size K for the training loss curve on EC and RV datasets. It shows that a larger K tends to converge faster.

Comparison of Performance using Simulated Real-World Missing Data

- In the previous section, we experiment with the assumption that elements are randomly missing
- However, in the real-world, completely random missing is rare in time series .

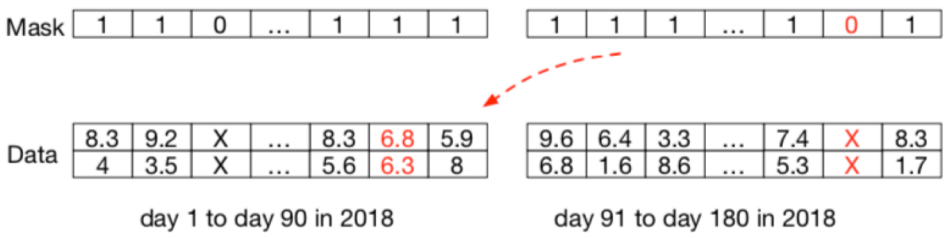


Figure 8: An illustration of ground truth generation on the EC dataset to simulate real-world missing data.

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 - social	<u>1.6223</u>	<u>2.6731</u>
STI	**1.5837**	**2.6412**			

Table 3: Performance on the EC dataset with simulated missing data.The best results are in boldface, and the second-best results are underlined.