

# 搜索算法实习生转正答辩

刘宗涛

2019.08.28

# 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：社交网络数据挖掘
- 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

实验室工作

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)

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

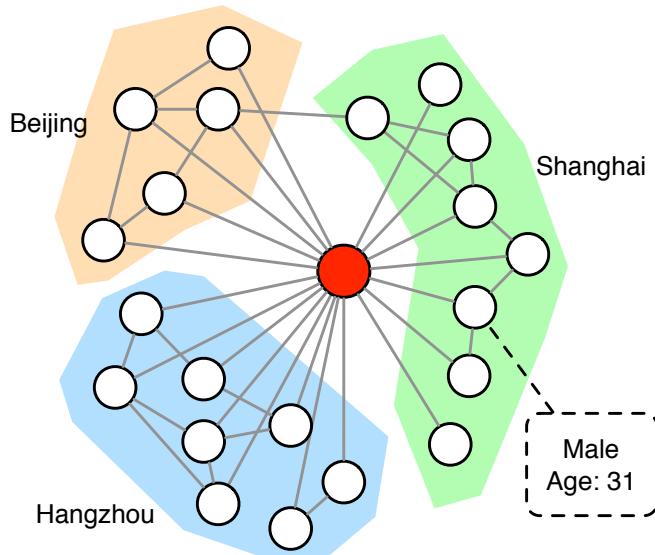
# 移居者的都市梦：城市移民群体行为研究

- 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？

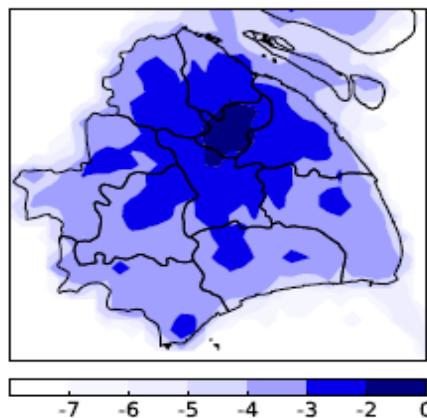
# 用户通话网络

- 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**

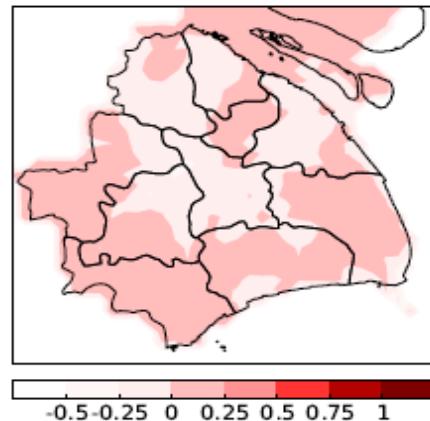


# 不同群体的行为模式差异

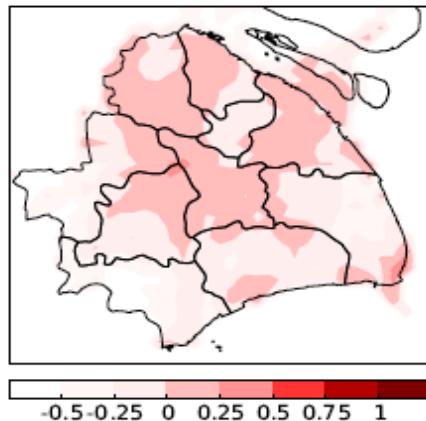
- **本地人**: 出生在本地的上海人
- **老移民者**: 在上海已经生活了一段时间、安顿下来了的移民者
- **新移民者**: 刚来上海一周的移民者



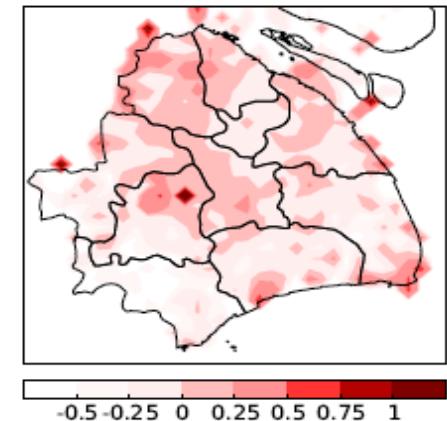
(a) Log overall average probability.



(b) Log odds ratio for locals.



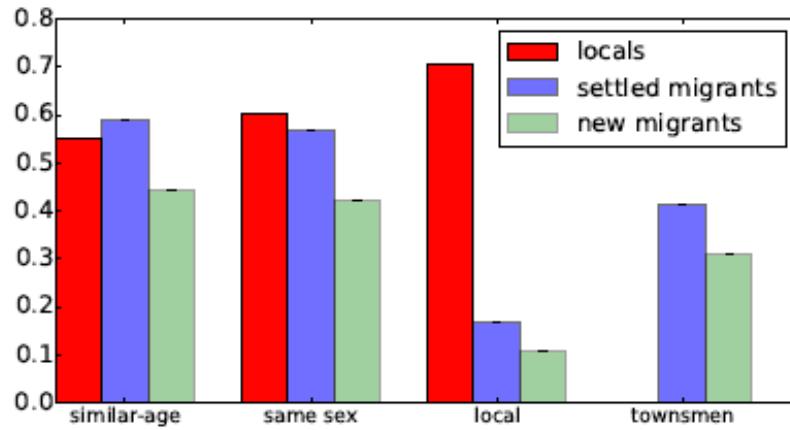
(c) Log odds ratio for settled migrants.



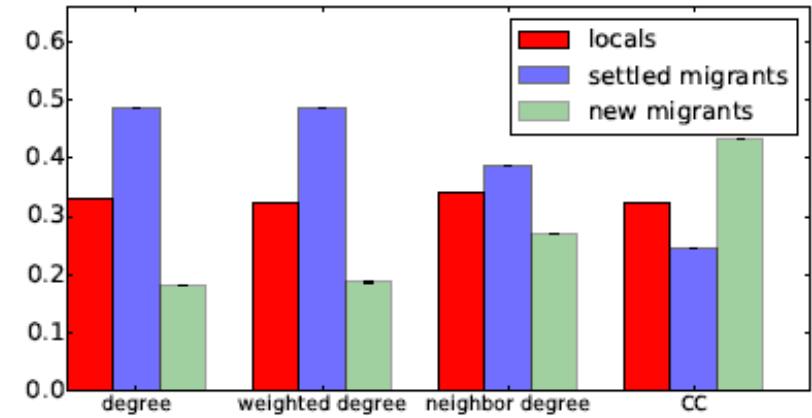
(d) Log odds ratio for new migrants.

**本地人: 1.7M, 老移民者: 1.0M, 新移民者: 34K**

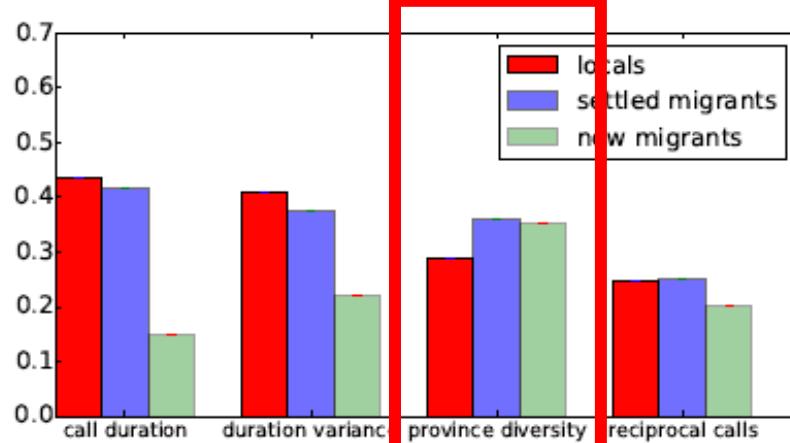
# 移民者有更多元的人际关系， 更大的活动区域



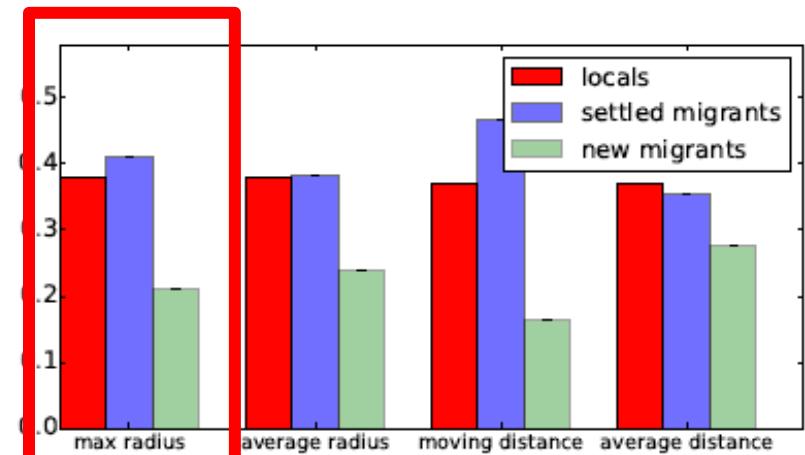
(a) Demographics of friends.



(b) Ego-network characteristics.



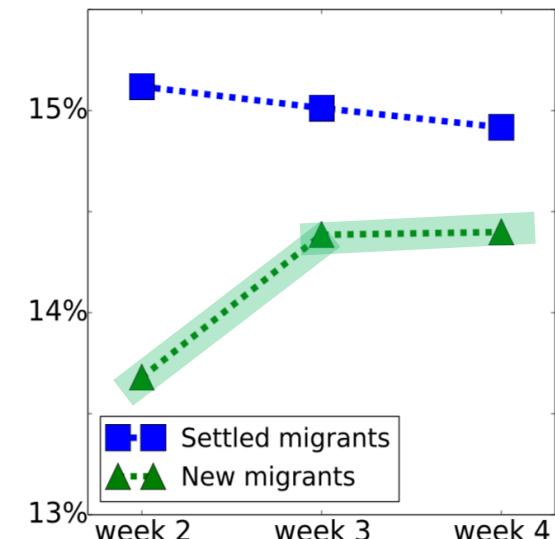
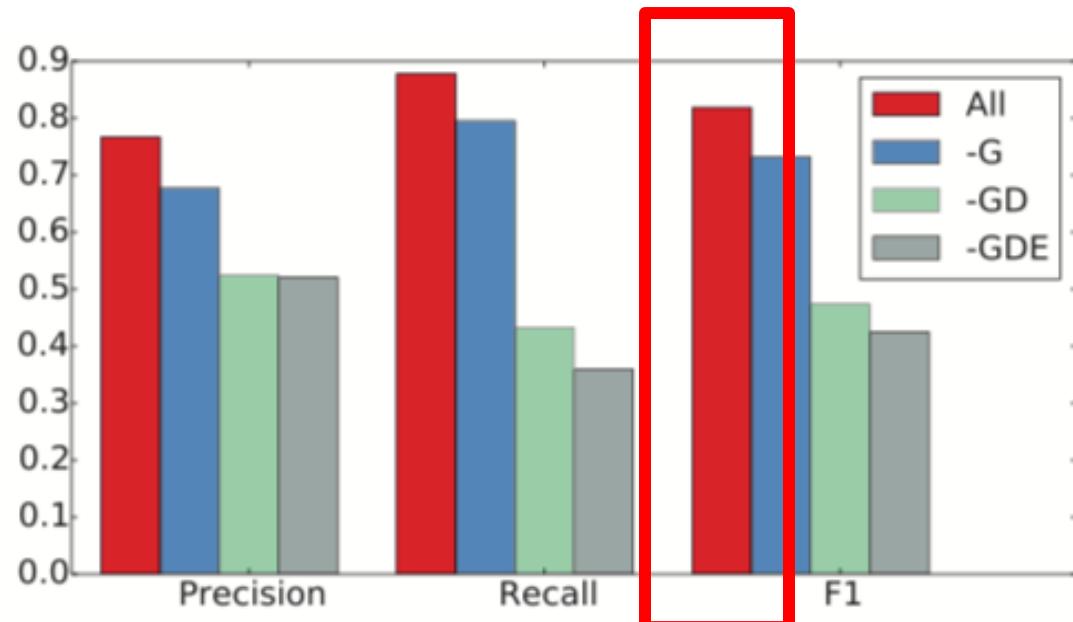
(c) Call behavior.



(d) Geographical features.

# 新移民逐渐向本地人趋近

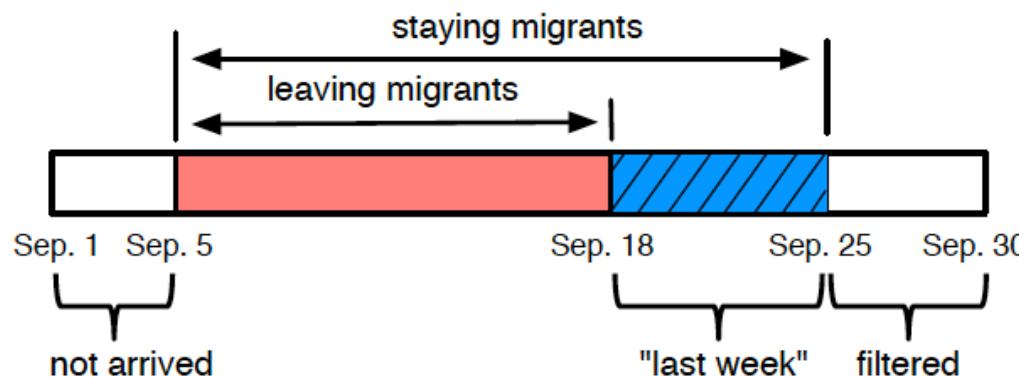
- 二分类：根据首周用户通话记录，判断其为本地人，老移民者



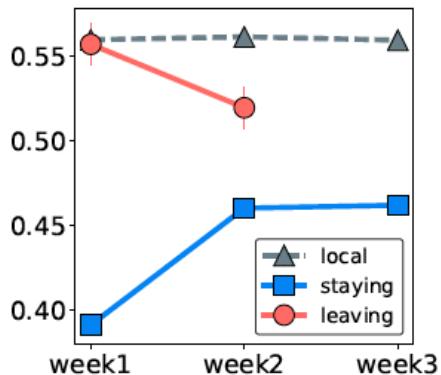
老移民者与新移民者中，被误判为本地人的比例

# 什么因素导致移民者离开都市？

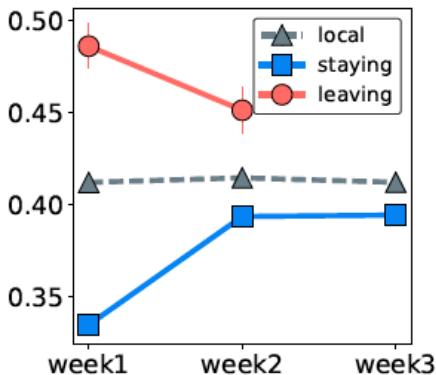
- 抵达都市后的**前两周**很关键！
- 进一步将新移民者划分为流失移民者(1.5K)和暂存移民者(34K)
- 2016年9月下旬，**4%**的新移民者离开了上海



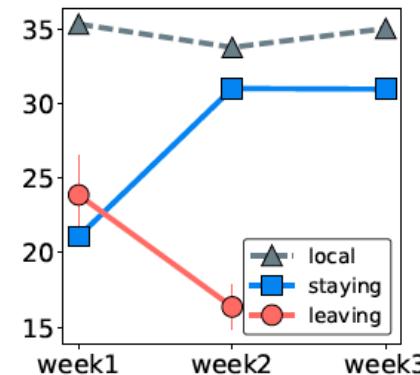
# 初步构建当地关系网络



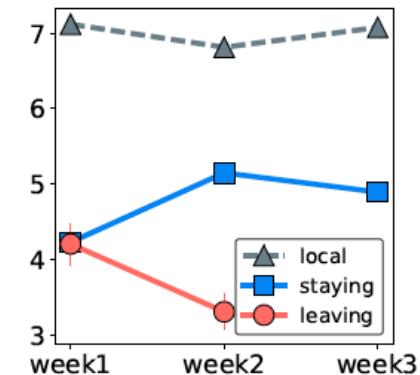
(a) Proportion of same sex contacts.



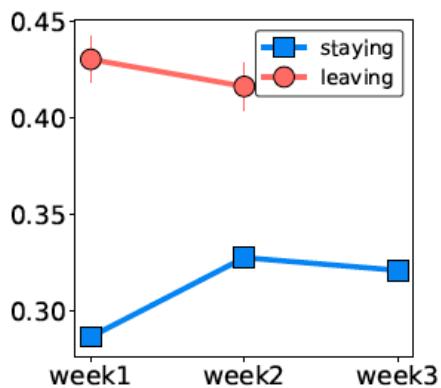
(b) Proportion of similarly aged contacts.



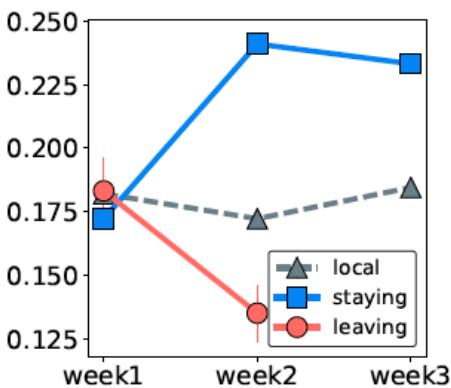
(c) Degree.



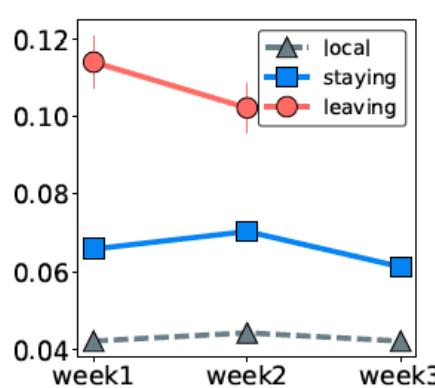
(d) Average degree of contacts.



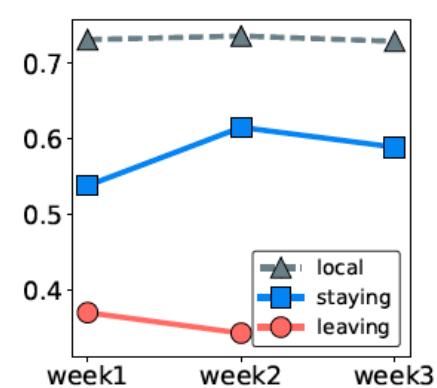
(e) Proportion of townspeople.



(f) Province diversity.



(g) Clustering coefficient.

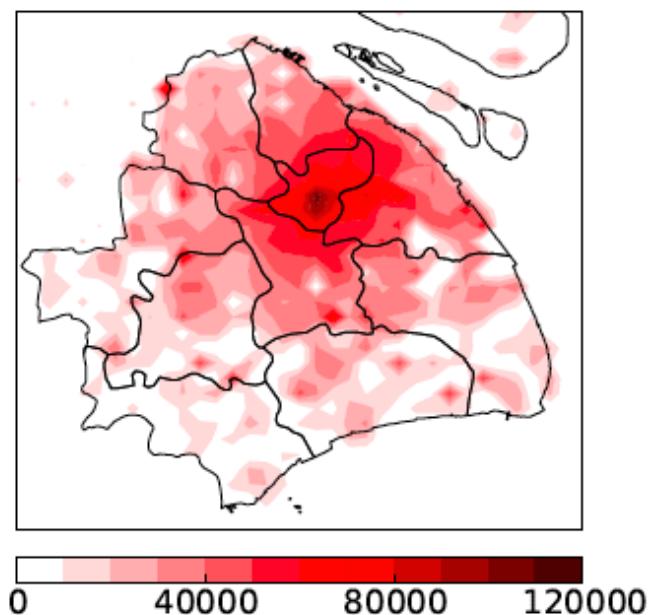


(h) Communication diversity.

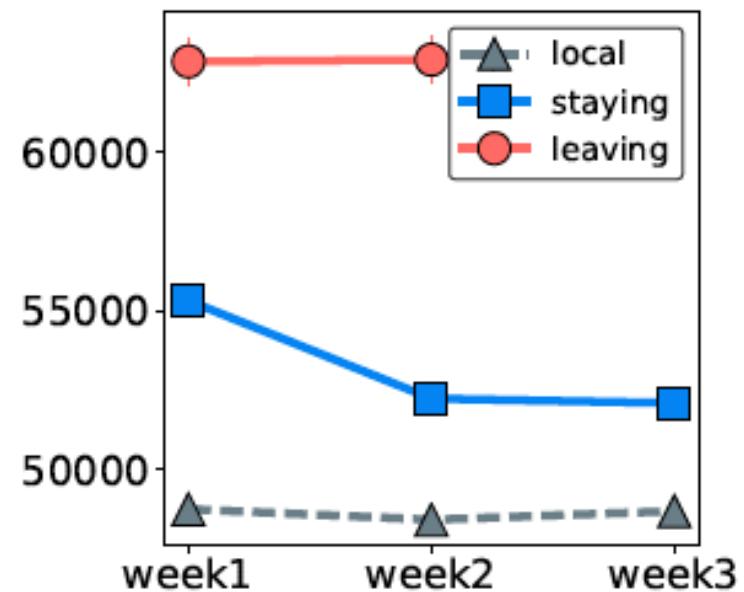
积极扩展人脉、发展**多样性**的关系与移民者能否留在都市的关联性很强

# 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



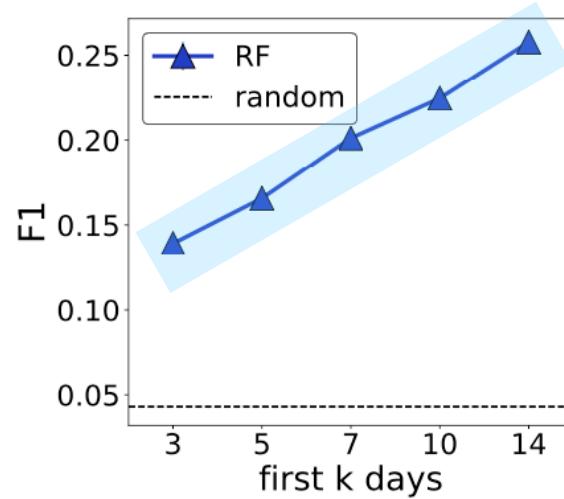
(a)上海市房价热点分布



(b)用户居住地的平均房价

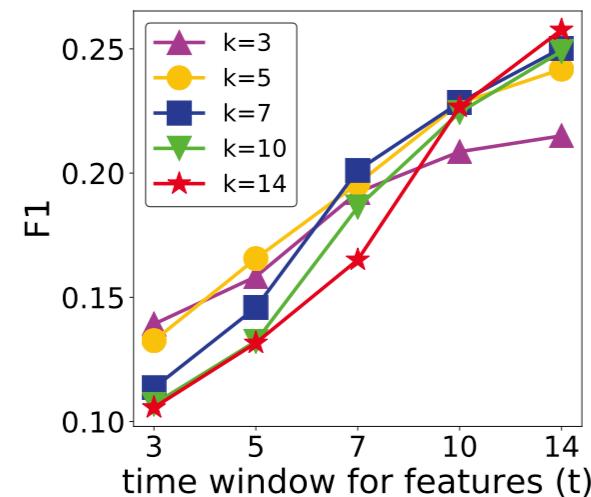
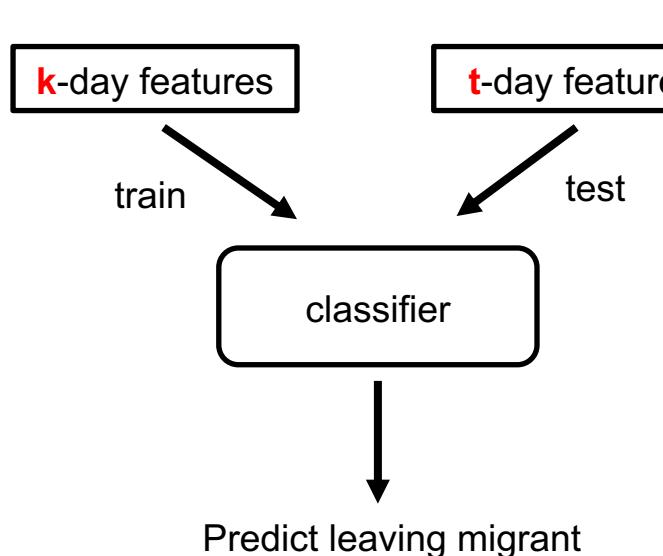
# 流失移民 vs. 留存移民

- 在移居早期识别移民的流失
  - 基于前k天数据提取特征，预测其两周后是否会离开上海
    - 如果可以，政府和公益团体也许能为这类移民提供帮助



# 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！

# 社交感知的时序补全算法

- 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

# 社交感知的时序补全算法

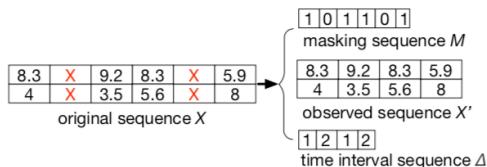
## Definition

social network:  $G = \langle V, E \rangle$

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

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

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:

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

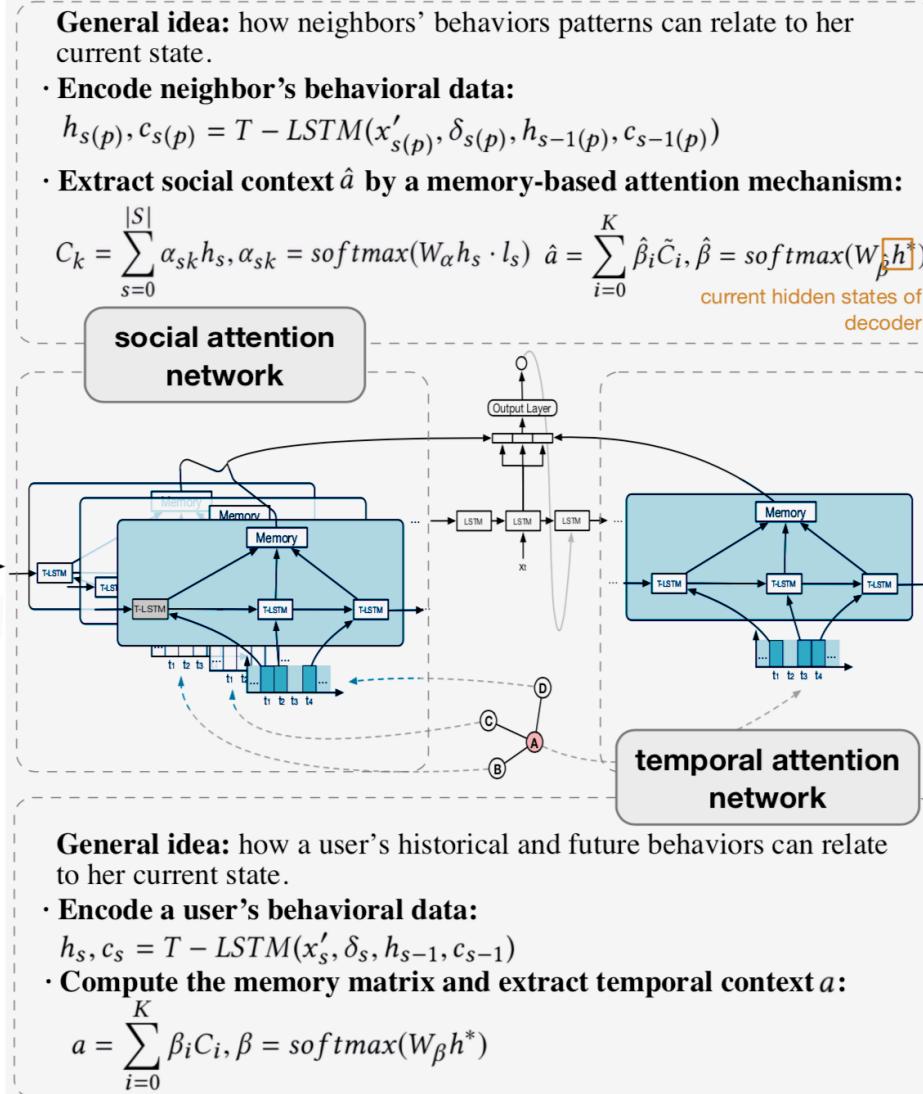
$$\hat{c}_t^s = c_{t-1}^s \cdot g(\delta) \quad \text{decaying function}$$

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

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

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

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



## 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  $p \sim \mathcal{U}(1)$  :

$$\text{if } p > \gamma \text{ then } x' = x_{t-1}^* \\ \text{else } x' = x_{t-1}^* \cdot (1 - m_{t-1}) + x_{t-1} \cdot m_{t-1}$$

predicted value

4. Compute loss and apply updates.

## Imputation:

for each input in decoding step :

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

# 实验结果

## 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	<u>1.6223</u>	<u>2.6731</u>
STI	<b>1.5837</b>	<b>2.6412</b>			

## 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
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	<b>1.4628</b>	2.2337	1.4985	2.3364	1.5463	<b>2.3432</b>	<b>1.5672</b>	2.4208	1.6161	2.4593
	STI	1.4667	<b>2.2172</b>	<b>1.4864</b>	<b>2.2574</b>	<b>1.5207</b>	2.3745	1.5696	<b>2.3924</b>	<b>1.6159</b>	<b>2.4505</b>
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	<b>2.0008</b>	<b>2.9426</b>	<b>2.0787</b>	<b>3.0858</b>	<b>2.1258</b>	<b>3.1306</b>	<b>2.2795</b>	<b>3.3187</b>	<b>2.4963</b>	<b>3.5972</b>

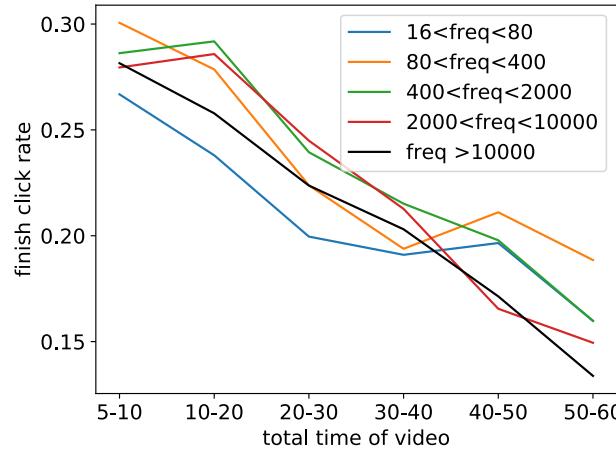
# 抖音点击标签修正

- ctr预估分值是多目标融合中重要的因子：
  - `final_score = f(相关性, 权威性, ctr预估分值)`
- 抖音现采用播放完成标签
  - 对长视频存在打压
  - 提供的正样本不足 (`duration > 0` 占比 0.14)
  - 数据收集：建立了 `doc_info` 数据收集的流程
    - `dm_search.aweme_search_docinfo_instance_daily`



# 抖音点击标签修正

- ctr预估分值是多目标融合中重要的因子：
  - $\text{final\_score} = f(\text{相关性}, \text{权威性}, \text{ctr预估分值})$
- 抖音现采用播放完成标签
  - 对长视频存在打压
  - 提供的正样本不足 ( $\text{duration} > 0$  占比 0.14)

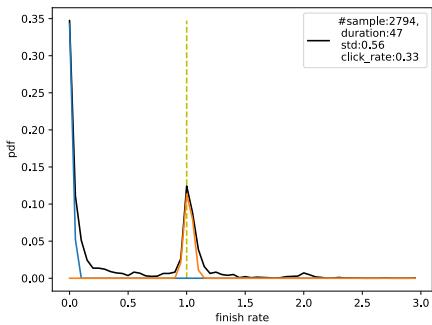


2019年7月15日，取不同频率query，前15个播放时长大于零的视频，播放完成率

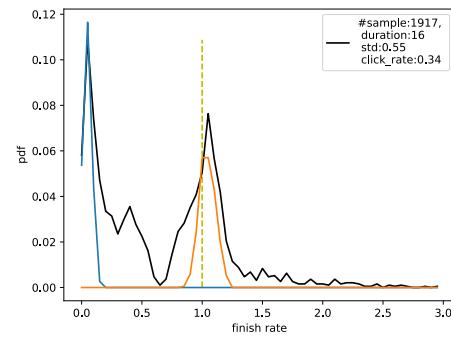
# 抖音点击标签修正

## • 数据观察

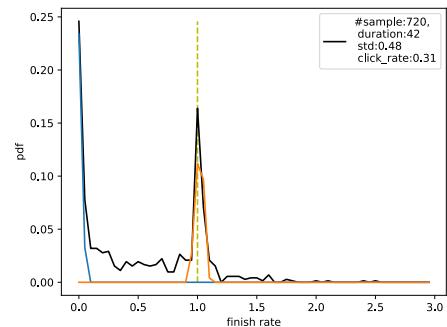
### 播放比率分布



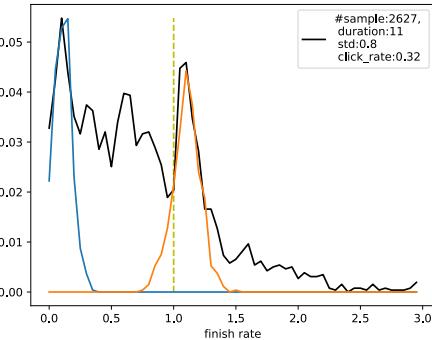
吃鸡



当男朋友爱上追偶像剧

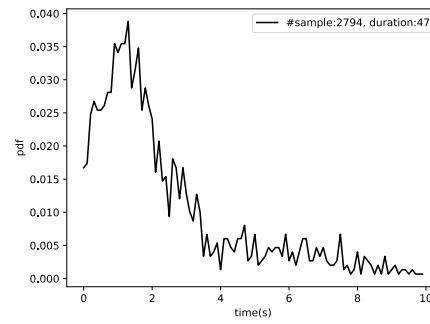


奥迪A7

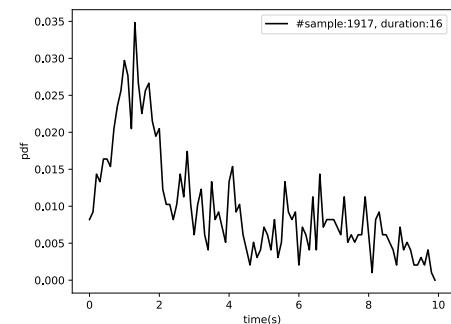


李现杨紫商业互吹

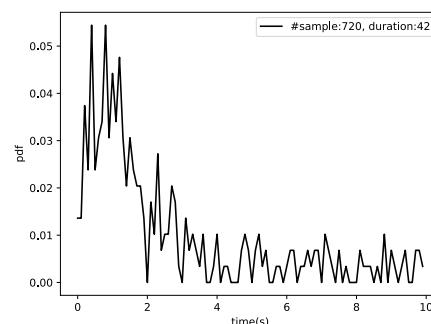
### 播放时长（截止10s）分布



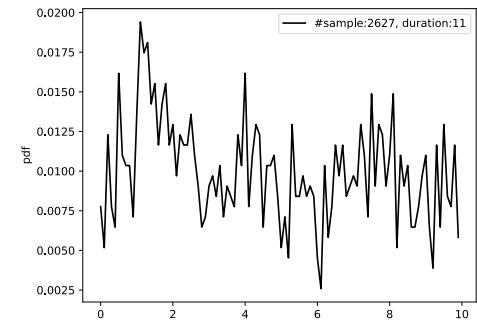
吃鸡



当男朋友爱上追偶像剧



奥迪A7

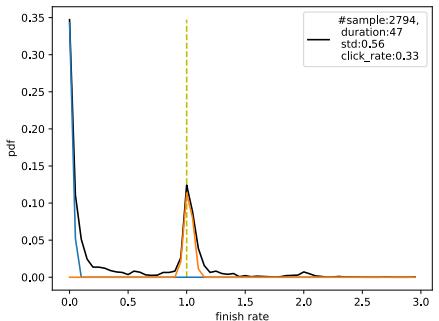


李现杨紫商业互吹

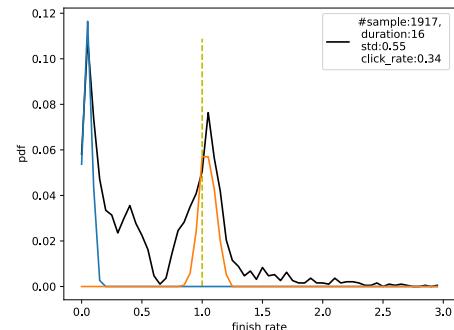
在接近播放完成时刻存在聚合性的流失分布  
用播放比率阈值来替代播放完成作为标签可  
以带来更多有效样本！

# 标签修正

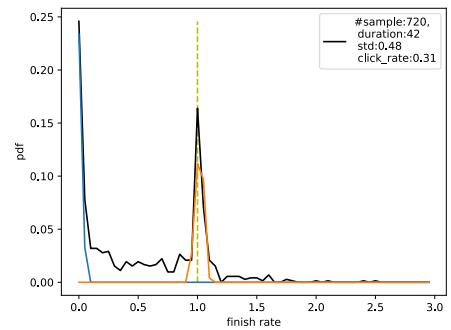
播放比率分布



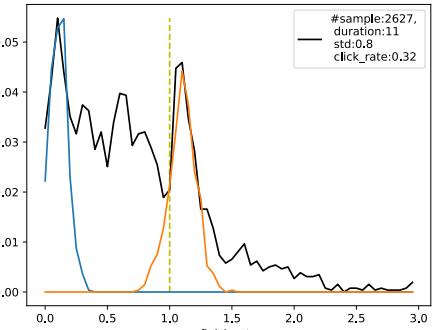
吃鸡



当男朋友爱上追偶像剧

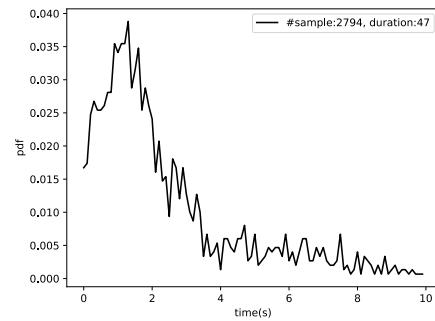


奥迪A7

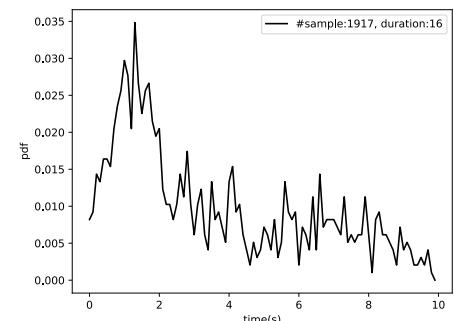


李现杨紫商业互吹

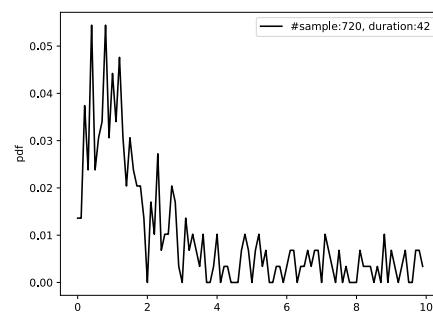
播放时长（截止 10s）分布



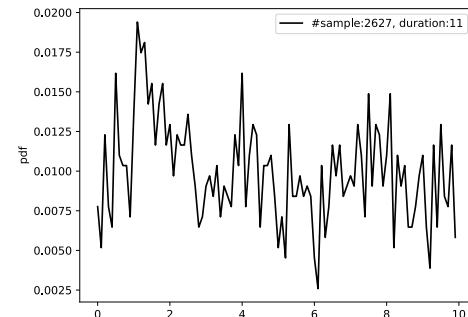
吃鸡



当男朋友爱上追偶像剧



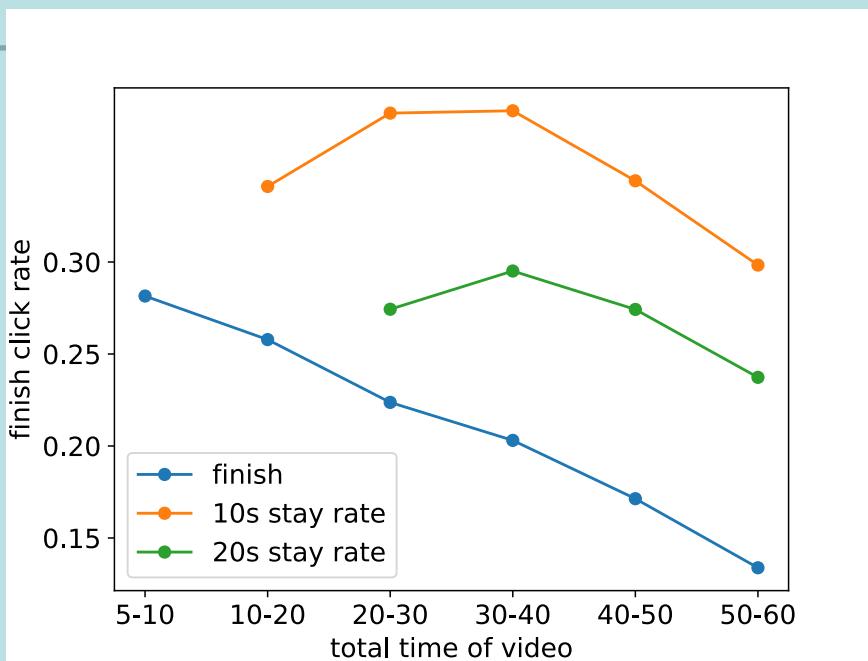
奥迪A7



李现杨紫商业互吹

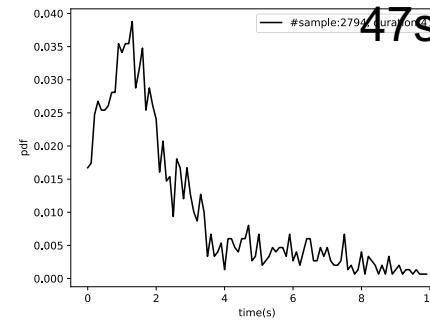
# 标签修正

不同视频在前10s有接近的分布状况  
是否前k s的分布都有类似的现象

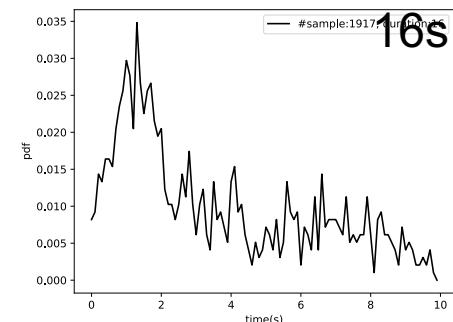


和播放时长相关的统计量和播放长度  
不敏感，播放时长可能是一个衡量用  
户满意的一个比较好的指标

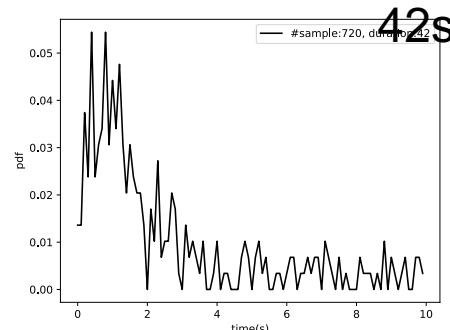
播放时长（截止10s）分布



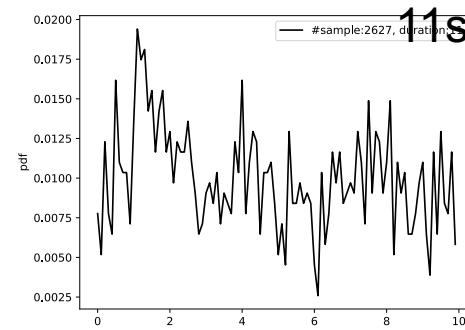
吃鸡



当男朋友爱上追偶像剧



奥迪A7



李现杨紫商业互吹

# 抖音点击标签修正

- 以不同的播放完成比例阈值取代播放完成标签
  - 0.8, 0.85, 0.9, 0.95

# 离线训练数据制作

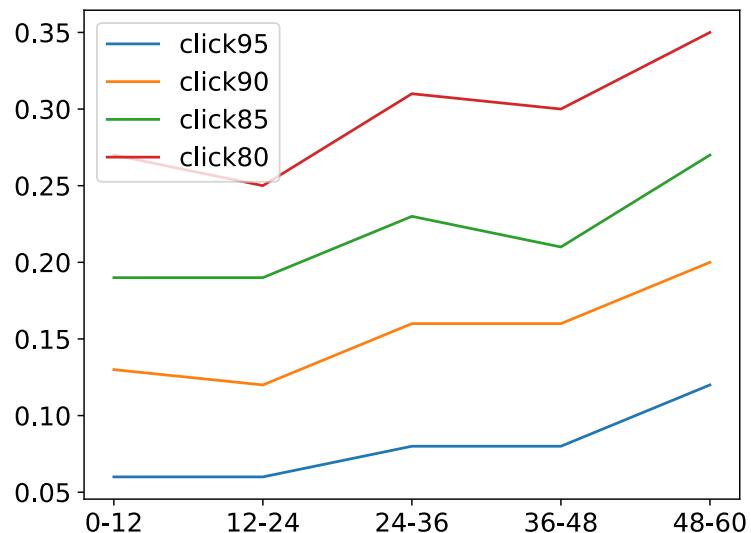
- 特征抽取的有关代码添加了新增标签
  - finish, finish80, finish85, finish90, finish95
- 采样规则调整
- 进行了四个月的离线训练数据制作，并例行化(impression limit + 特征抽取)

# 抖音点击标签修正

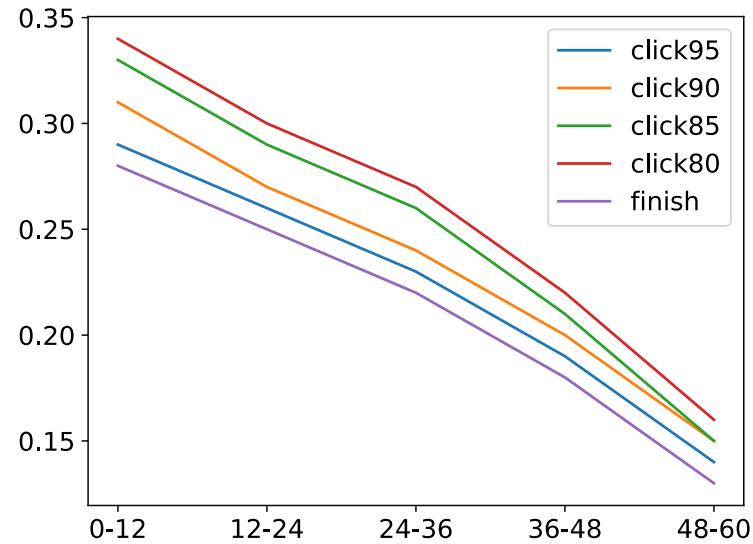
- 以不同的播放完成比例阈值取代播放完成标签
  - 0.8, 0.85, 0.9, 0.95
- 已开AB两天
  - 播放完成比-8.6%，播放时长+0.7%，有点比-0.4%
  - 数据中视频的正例比
    - 原：0.0324，现：0.0360，新的正例中未finish  
 $\frac{0.0360 - 0.0324}{0.0360} = 10\%$ ，如果模型其他方面效果持平，播放完成数的降幅应该在10%左右

# 抖音点击标签修正

- 在引入了更多长视频正例的同时，也引入了一些带有噪声的短视频正例



使用播放完成比阈值取代播放完成标签  
不同时长视频增加的平均正样本比例



不同时长各类标签的平均正例比

# 抖音点击标签修正

- 在引入了更多长视频正例的同时，也引入了一些带有噪声的短视频正例
  - 改进：分段函数
    - $thres = sign(duration - 10) * 0.9$
    - $thres = 1 - \frac{max(duration - 10, 0)}{60 - 10} * 0.1$

# 抖音点击标签修正

- 数据中视频的正例比
- 正例比：
  - flow: 0.24
  - batch: 0.03
- 没有纠偏，导致ctr降低
  - sample rate
  - 多目标融合

$$p = \frac{1}{1 + e^{-(w^T x + b + \log(n))}}$$

# ctr + 播放时长多目标模型

- 播放时长模型
  - ctr + 播放时长多目标模型
  - 为什么使用多目标?
    - 减少训练开销
    - 学习共享隐层特征，减少过拟合

# ctr + 播放时长多目标模型

- 如何设计多目标任务模型

- Shared-Bottom
- MMoE (KDD' 18)

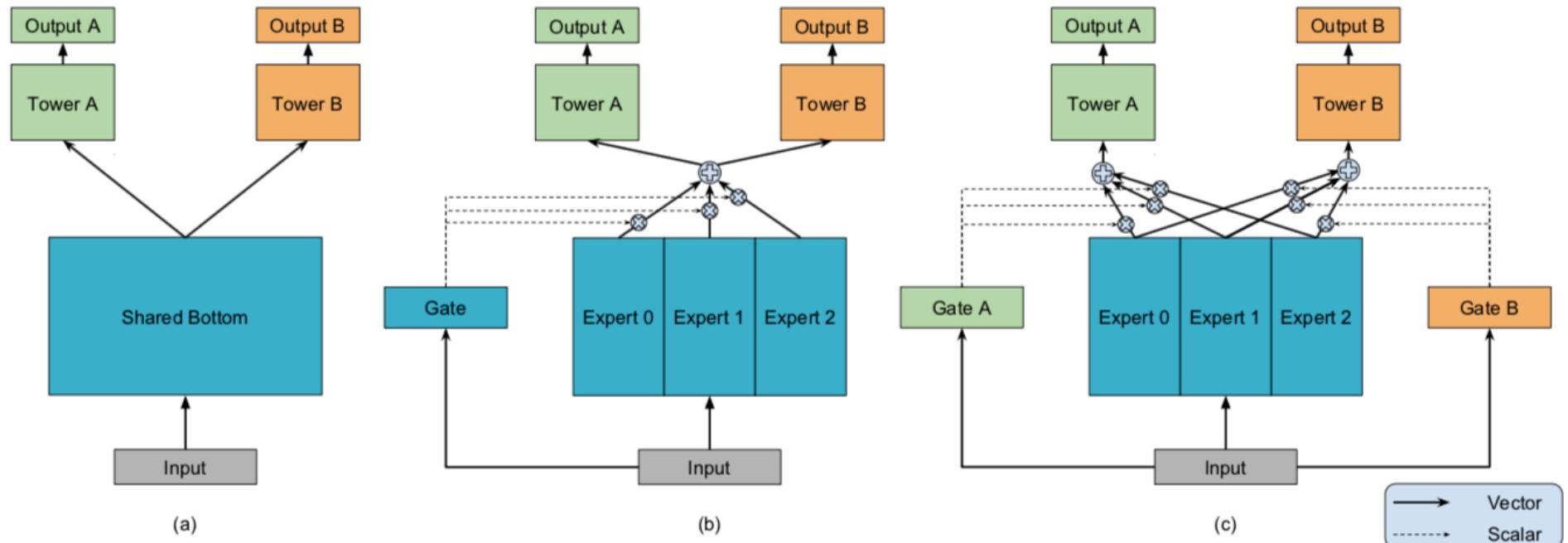
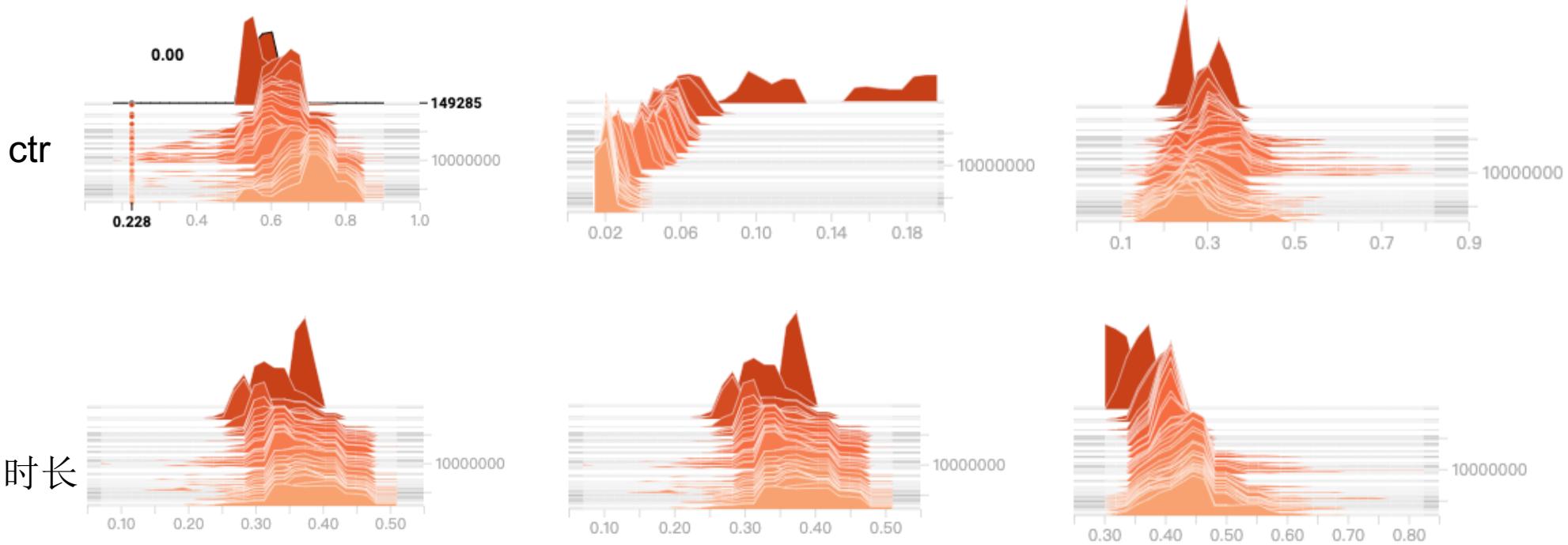


Figure 1: (a) Shared-Bottom model. (b) One-gate MoE model. (c) Multi-gate MoE model.

# ctr + 播放时长多目标模型

- 如何设计多目标任务模型

- Shared-Bottom
- MMoE (KDD' 18)



# ctr + 播放时长多目标模型

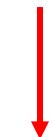
- 如何设计多目标任务模型

- Shared-Bottom
- MMoE (KDD' 18)
- GradNorm (ICLR' 18)

$$L(t) = \sum w_i(t) L_i(t)$$
$$G_W^{(i)}(t) \mapsto \bar{G}_W(t) \times [r_i(t)]^\alpha$$

$\uparrow$                      $\uparrow$                      $\uparrow$

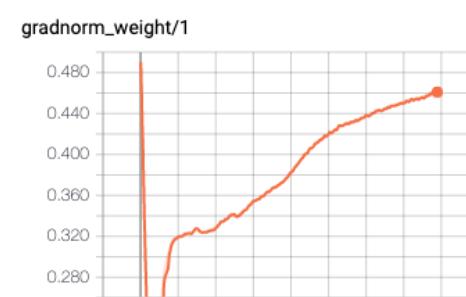
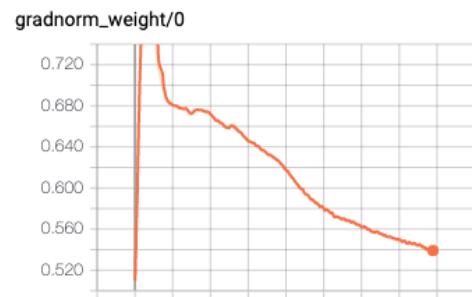
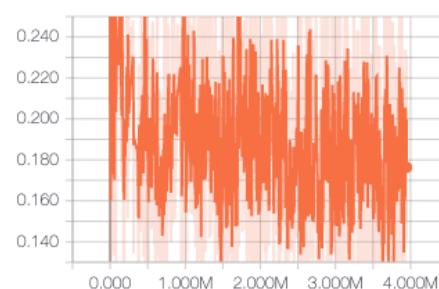
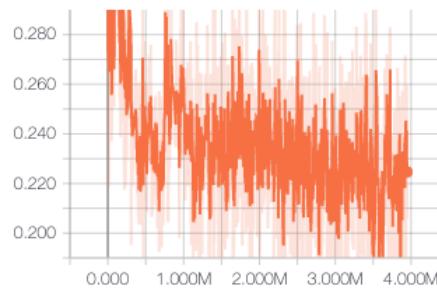
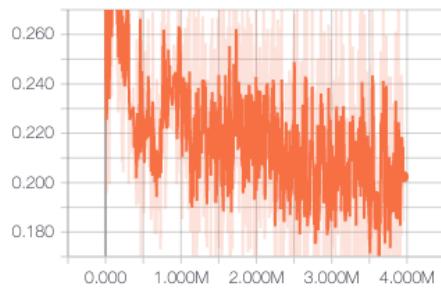
|| $\nabla_W w_i(t) L_i(t)$ ||<sub>2</sub>                    各loss平均梯度                    relative  
单个loss对共享层参数的梯度                    各loss平均梯度                    inverse training rate

$$L_{\text{grad}} = \sum_i |G_W^{(i)}(t) - \bar{G}_W(t) \times [r_i(t)]^\alpha|_1$$


# ctr + 播放时长多目标模型

- 如何设计多目标任务模型

- Shared-Bottom
- MMoE (KDD' 18)
- GradNorm (ICLR' 18)



# 问题与规划

- 有新想法如何比较方便验证
  - 如何比较方便的更新标签
    - 现状：更新标签需要重新制作（综搜一天的数据：3h-6h）
    - 方法1：分别制作特征和标签，然后聚合
    - 方法2：在模型中实时计算标签
      - Lagrange和Sail支持标签的实时计算
      - 统一在标签项存入用来计算label的基础信息
        - » `finish, play_time, total_time, is_video`

```
click, duration, span, is_video = parser.get_label(['click', 'duration', 'span',  
'is_video'])  
label = click * (1 - is_video) + tf.sign(duration>span*0.9) * is_video  
label = tf.identity(label, name="label")
```

# 问题与规划

- 有新想法如何比较方便验证
  - 如何比较方便的更新标签
    - 现状：更新标签需要重新制作（综搜一天的数据：3h–6h）
    - 方法1：分别制作特征和标签，然后聚合
    - 方法2：在模型中实时计算标签
  - 离线训练切在线训练，标签分布不一致如何调整
    - 离线标签对齐在线标签的正负样本比
  - 更好的离线自动化测试流程
    - 方便调整参数

# 问题与规划

- 多目标任务能走多远?
  - ctr + 播放时长 + (视频点击行为) + ?
    - ?: k秒留存率, 用户类型
  - 如何处理部分目标带来的牵扯效应 (1+1<2)
    - 合适的正则, 软约束
  - 迁移学习

# 收获和反思

- 跨组沟通的能力
  - 借鉴经验，请教问题
- 提前规划和复盘
  - 如何在基础设施不够完善/资源不足的情况下合理规划
  - 实现基线方法，查看大致效果，然后进行快速迭代
- 技术收获
  - 海量数据场景的机器学习
  - MapReduce
  - 开发能力

# 收获和反思

- 跨组沟通的能力
  - 借鉴经验，请教问题
- 提前规划和复盘
  - 如何在基础设施不够完善/资源不足的情况下合理规划
  - 实现基线方法，查看大致效果，然后进行快速迭代
- 技术收获
  - 海量数据场景的机器学习
  - MapReduce
  - 开发能力

Thanks  
Q&A

