Addressing Delayed Feedback for Continuous Training with Neural Networks in CTR prediction

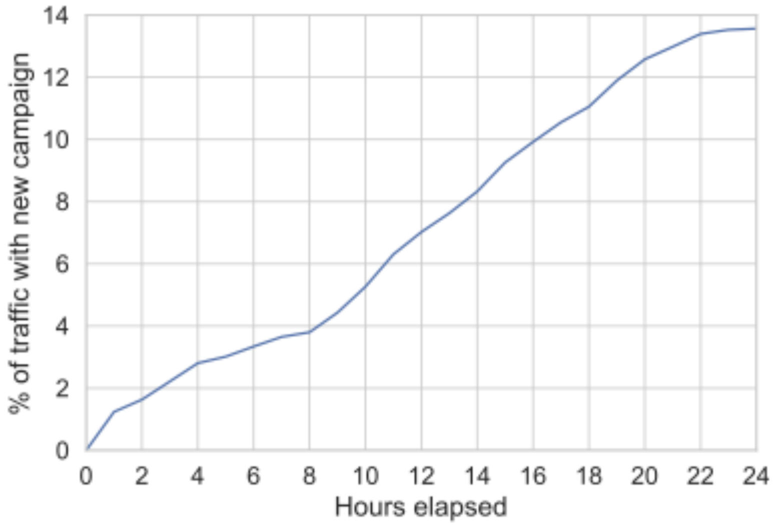
SI Ktena, A Tejani, L Theis, P Kumar Myana,
D Dilipkumar, F Huszár, S Yoo, W Shi

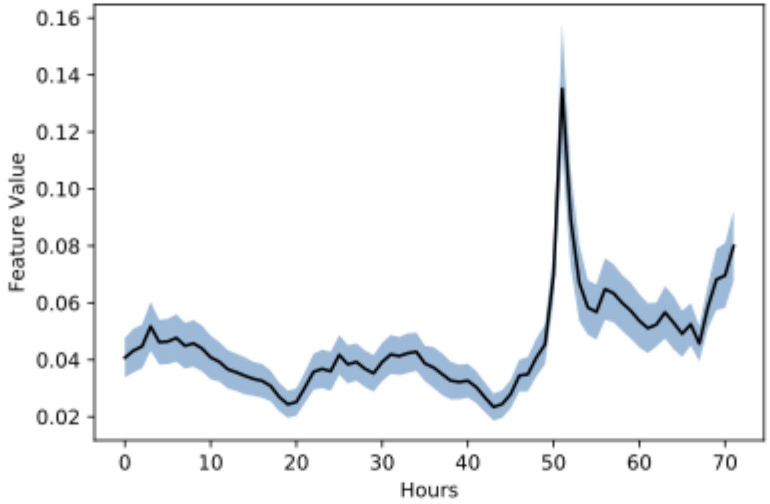
RecSys 2019

Why continuous training?

Why continuous training?

New campaign IDs + non-stationary features





Challenge: Delayed feedback

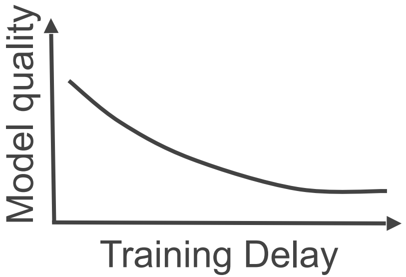
Fact:

Users may click ads after
1 second
1 minute
or 1 hour

Challenge: Delayed feedback

Why is it a challenge?

Should we wait? → Delays model training



Should we not wait? How do we decide the label?

Solution: accept “fake negative”

	Event	Label	Weight
Time	(user1, ad1, t1) imp		1
	(user2, ad1, t2) imp		1
	(user1, ad1, t3) click		1

Solution: accept “fake negative”

Time

Event	Label	Weight
(user1, ad1, t1) imp	−	1
(user2, ad1, t2) imp	−	1
(user1, ad1, t3) click		1

Solution: accept “fake negative”

Time	Event	Label	Weight
	(user1, ad1, t1) imp	−	1
	(user2, ad1, t2) imp	−	1
	(user1, ad1, t3) click	+	1

Solution: accept “fake negative”

Time	Event	Label	Weight
	(user1, ad1, t1) imp	−	1
	(user2, ad1, t2) imp	−	1
	(user1, ad1, t3) click	+	1

same

features

Solution: accept "fake negative"

Event	Label	Weight
(user1, ad1, t1) imp	−	1
(user2, ad1, t2) imp	−	1
(user1, ad1, t3) click	+	1

Time

same features

Solution: accept "fake negative"

Time

Event	Label	Weight
(user1, ad1, t1) imp	−	1
(user2, ad1, t2) imp	−	1
(user1, ad1, t3) click	+	1

same features

Assume **X** #Clicks out of **Y** #Impressions

Works well when CTR is low, **where X/Y ~= X/ (X+Y)**

Delayed feedback models

Delayed feedback models

Delayed feedback models

- The probability of click is not constant through time [*Chapelle 2014*]

Delayed feedback models

- The probability of click is not constant through time [*Chapelle 2014*]
- Second model similar to survival time analysis models captures the delay between impression and click

Delayed feedback models

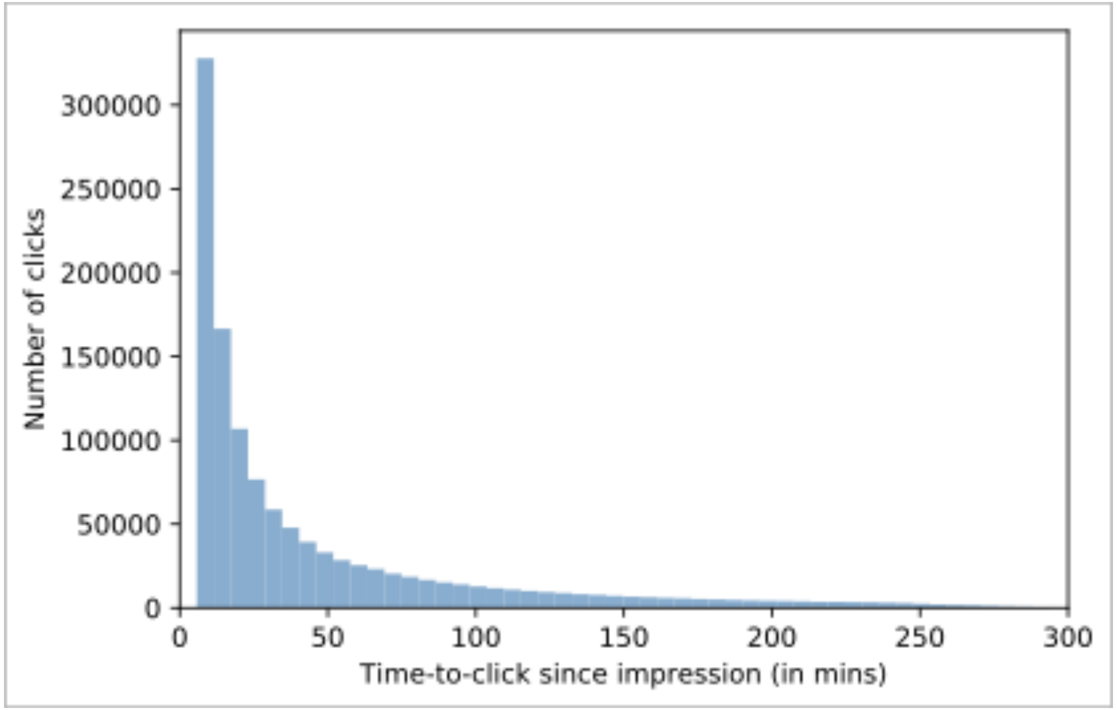
- The probability of click is not constant through time [*Chapelle 2014*]
- Second model similar to survival time analysis models captures the delay between impression and click
- Assume an exponential distribution or other non-parametric distribution

Delayed feedback models

- The probability of click is not constant through time [*Chapelle 2014*]
- Second model similar to survival time analysis models captures the delay between impression and click
- Assume an exponential distribution or other non-parametric distribution

$$P(d|\mathbf{x}, \mathbf{c} = 1) = \lambda(\mathbf{x}) \exp(-\lambda(\mathbf{x})d)$$

Delayed feedback models



Our approach

Importance sampling

- p is the actual data distribution
- b is the biased data distribution

$$\mathcal{L}(\theta) = -\mathbb{E}_p[\log f_\theta(\mathbf{y}|\mathbf{x})] = -\sum_{\mathbf{x},\mathbf{y}} p(\mathbf{x},\mathbf{y}) \log f_\theta(\mathbf{y}|\mathbf{x})$$

$$\mathbb{E}_p[\log f_\theta(\mathbf{y}|\mathbf{x})] = \mathbb{E}_b\Big[\frac{p(\mathbf{x},\mathbf{y})}{b(\mathbf{x},\mathbf{y})} \log f_\theta(\mathbf{y}|\mathbf{x})\Big]$$

Importance weights $w(\mathbf{x},\mathbf{y}) = \dfrac{p(\mathbf{x},\mathbf{y})}{b(\mathbf{x},\mathbf{y})}$

- Continuous training scheme -> potentially wait infinite time for positive engagement

- Two models
  - Logistic regression
  - Wide-and-deep model

- Four loss functions
  - Delayed feedback loss [*Chapelle, 2014*]
  - Positive-unlabeled loss [*du Plessis et al., 2015*]
  - Fake negative weighted
  - Fake negative calibration

- Continuous training scheme -> potentially wait infinite time for positive engagement

- Two models
  - Logistic regression
  - Wide-and-deep model

- Four loss functions
  - Delayed feedback loss [*Chapelle, 2014*]
  - Positive-unlabeled loss [*du Plessis et al., 2015*]
  - Fake negative weighted
  - Fake negative calibration

both rely on importance sampling

Delayed feedback loss

Assume exponential distribution for time delay

$$\lambda(\mathbf{x}) = \exp(\mathbf{w}_d \cdot \mathbf{x})$$

$$arg \min_{\theta, \mathbf{w}_d} \mathcal{L}_{DF}(\theta, \mathbf{w}_d) + \alpha(\|\theta\|_2^2 + \|\mathbf{w}_d\|_2^2)$$

Delayed feedback loss

Assume exponential distribution for time delay

$$\lambda(\mathbf{x}) = \exp(\mathbf{w}_d \cdot \mathbf{x})$$

$$arg \min_{\theta, \mathbf{w}_d} \mathcal{L}_{DF}(\theta, \mathbf{w}_d) + \alpha(\|\theta\|_2^2 + \|\mathbf{w}_d\|_2^2)$$

$$\mathcal{L}_{DF}(\theta, \mathbf{w}_d) = -\sum_{\mathbf{x}, y=1} \log f_\theta(\mathbf{x}) + \log \lambda(\mathbf{x}) - \lambda(\mathbf{x})d - \sum_{\mathbf{x}, y=0} \log[1 - f_\theta(\mathbf{x}) + f_\theta(\mathbf{x}) \exp(-\lambda(\mathbf{x})e)]$$

Fake negative weighted & calibration

$$\mathcal{L}_{IS}(\theta) = -\sum_{\mathbf{x},\mathbf{y}} b(\mathbf{y}=1|\mathbf{x})(1+p(\mathbf{y}=1|\mathbf{x}))\log f_\theta(\mathbf{x})+$$
$$b(\mathbf{y}=0|\mathbf{x})p(\mathbf{y}=0|\mathbf{x})(1+p(\mathbf{y}=1|\mathbf{x}))\log f_\theta(\mathbf{y}=0|\mathbf{x})$$

Don’t apply any weights on the training samples, only calibrate the output of the network using the following formulation

$$p(\mathbf{y}=1|\mathbf{x}) = \frac{b(\mathbf{y}=1|\mathbf{x})}{1-b(\mathbf{y}=1|\mathbf{x})}$$

Fake negative weighted & calibration

$$\mathcal{L}_{IS}(\theta) = -\sum_{\mathbf{x},\mathbf{y}} b(\mathbf{y}=1|\mathbf{x})(1+p(\mathbf{y}=1|\mathbf{x}))\log f_\theta(\mathbf{x}) +$$
$$b(\mathbf{y}=0|\mathbf{x})p(\mathbf{y}=0|\mathbf{x})(1+p(\mathbf{y}=1|\mathbf{x}))\log f_\theta(\mathbf{y}=0|\mathbf{x})$$

Don’t apply any weights on the training samples, only calibrate the output of the network using the following formulation

$$p(\mathbf{y}=1|\mathbf{x}) = \frac{b(\mathbf{y}=1|\mathbf{x})}{1-b(\mathbf{y}=1|\mathbf{x})}$$

Experiments

Criteo data

- Small dataset & public
- Training - 15.5M / Testing: 3.5M examples

Logistic regression - Criteo data			
Loss function	Loss	RCE	PR-AUC
Log loss	0.3963	17.26	**0.5081**
Delayed feedback loss	0.3970	**17.32**	0.5080
PU loss	0.4065	15.10	0.5048
FN weighted	0.4008	16.30	0.5037
FN calibration	**0.3961**	17.29	0.4983

RCE: normalised version of cross-entropy (higher values are better)

Criteo data

- Small dataset & public
- Training - 15.5M / Testing: 3.5M examples

Logistic regression - Criteo data			
Loss function	Loss	RCE	PR-AUC
Log loss	0.3963	17.26	**0.5081**
Delayed feedback loss	0.3970	**17.32**	0.5080
PU loss	0.4065	15.10	0.5048
FN weighted	0.4008	16.30	0.5037
FN calibration	**0.3961**	17.29	0.4983

RCE: normalised version of cross-entropy (higher values are better)

Twitter data

- Large & proprietary due to user information
- Training: 668M ads w. FN / Testing: 7M ads

Wide & deep - Twitter data			
Loss function	Loss	RCE	PR-AUC
Log loss	0.5953	7.81	0.5872
Delayed feedback loss	0.5781	12.11	0.5781
PU loss	0.5567	13.57	**0.5927**
FN weighted	0.5568	13.54	0.5925
FN calibration	**0.5566**	**13.58**	0.5923

RCE: normalised version of cross-entropy (higher values are better)

Twitter data

- Large & proprietary due to user information
- Training: 668M ads w. FN / Testing: 7M ads

Wide & deep - Twitter data			
Loss function	Loss	RCE	PR-AUC
Log loss	0.5953	7.81	0.5872
Delayed feedback loss	0.5781	12.11	0.5781
PU loss	0.5567	13.57	**0.5927**
FN weighted	0.5568	13.54	0.5925
FN calibration	**0.5566**	**13.58**	0.5923

RCE: normalised version of cross-entropy (higher values are better)

Online (A/B test)

Wide & deep - Online experiment			
Loss function	Pooled RCE	RPMq	Monetized CTR
Log loss	7.68	100.00	100.00
PU loss	*12.27*	*137.00*	*118.59*
FN weighted	**13.39**	**155.10**	123.01
FN calibration	13.37	154.37	**123.19**

Pooled RCE: RCE on combined traffic generated by models

RPMq: Revenue per thousand requests

Conclusions

- Solve problem of delayed feedback in continuous training by relying on importance weights
- **FN weighted** and **FN calibration** proposed and applied for the first time
- Offline evaluation on large proprietary dataset and online A/B test

Future directions

- Address catastrophic forgetting and overfitting
- Exploration / exploitation strategies

Questions?

https://careers.twitter.com

@s0f1ra