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

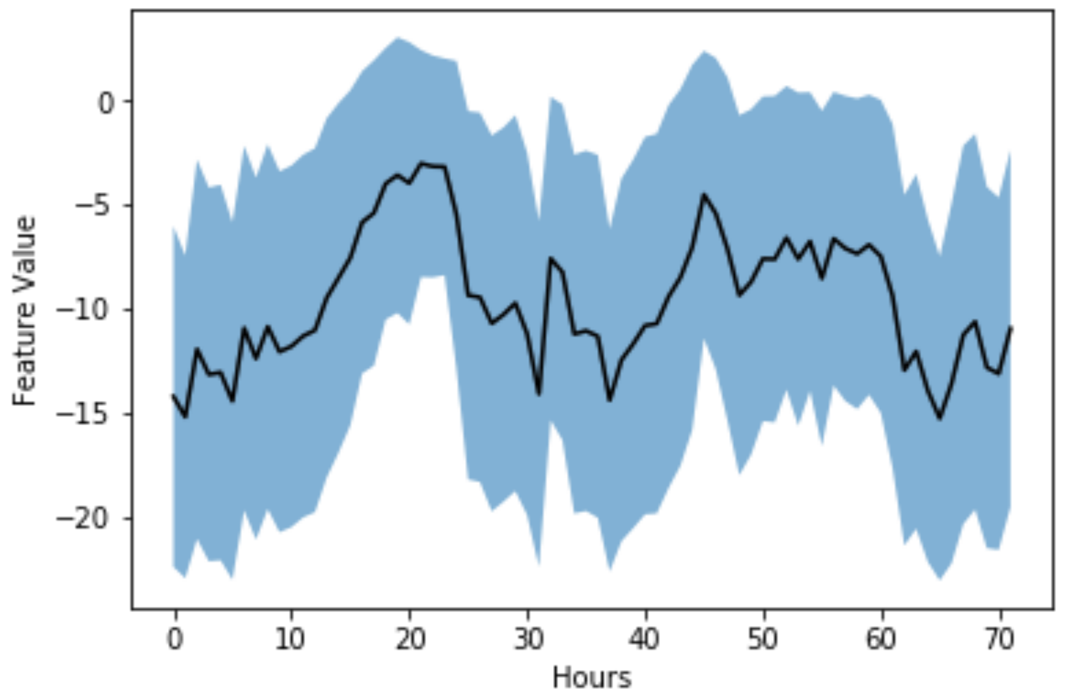
Sofia Ira Ktena*, Alykhan Tejani, Lucas Theis, Pranay Kumar Myana, Deepak Dilipkumar, Ferenc Huszár, Steven Yoo, Wenzhe Shi

*siraktena@twitter.com
{first name initial}{surname}@twitter.com

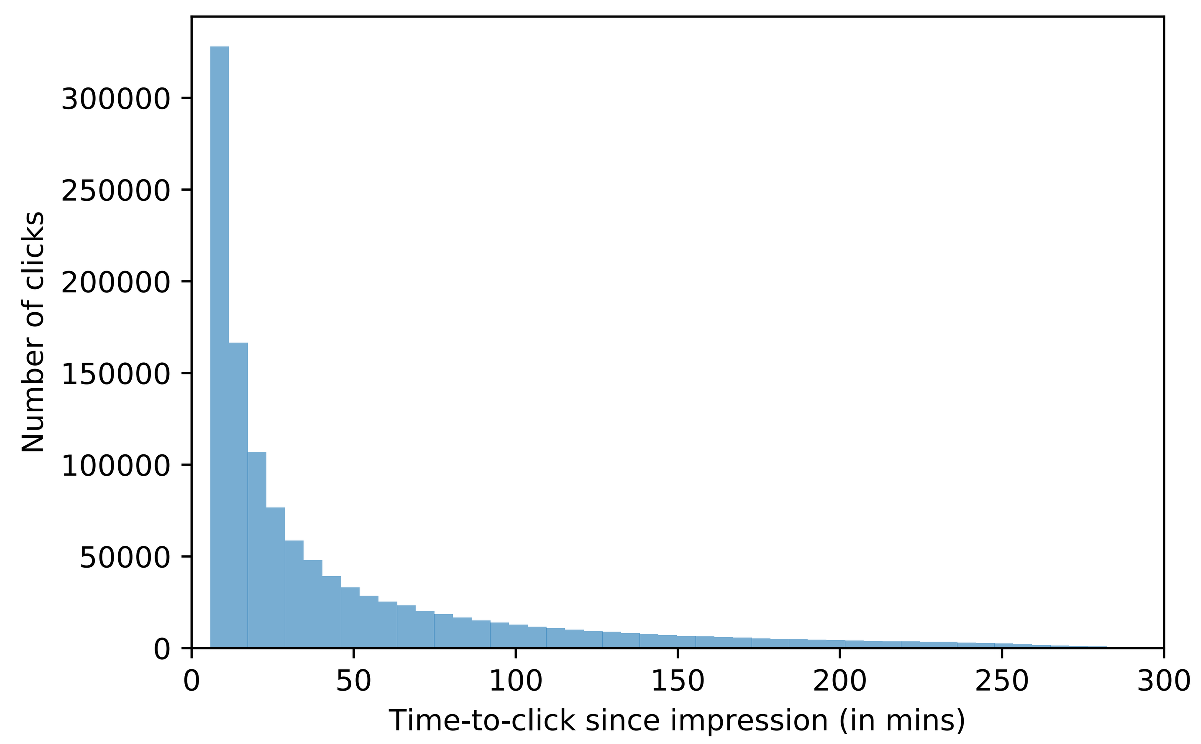
Twitter

## #TheProblem

**Non-stationarity**: Feature distributions and click through rate (CTR) exhibit large shifts over time due to seasonality and other factors




**Delayed feedback:** Fresh data may be not have complete information in a ***continuous training*** setup; engagements may occur with a delay of 1 hour or more after the ad is displayed

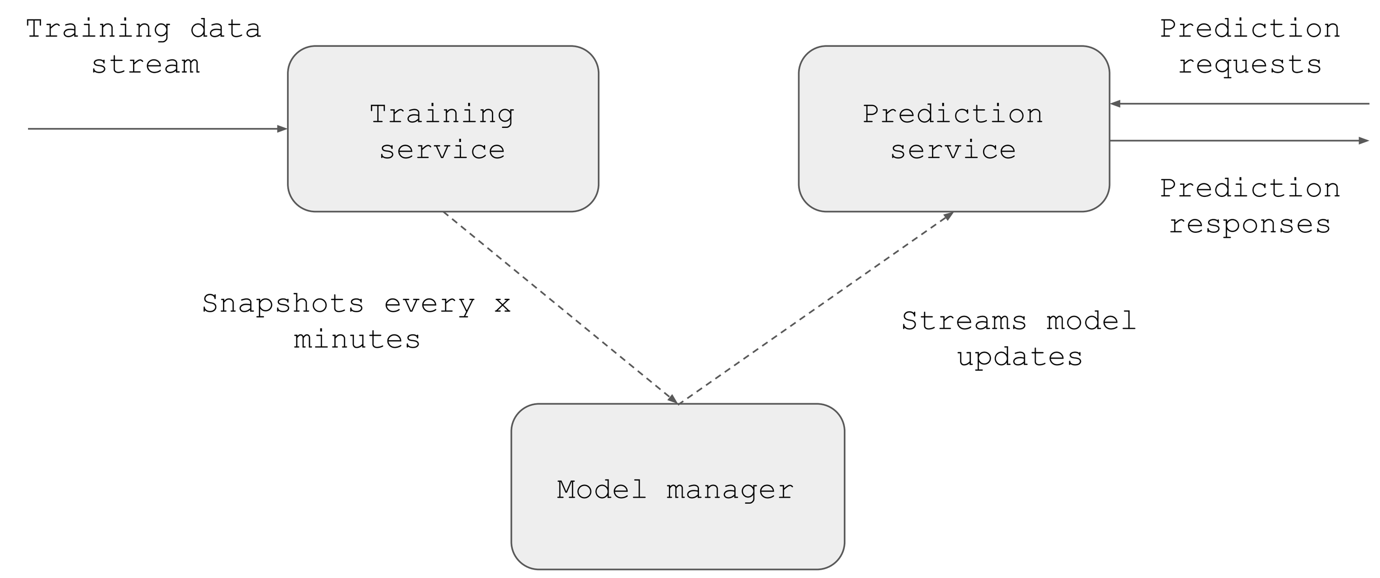



## #KeyIdeas

- Use a continuously trained model to estimate probability of click (pCTR) in display advertising, while addressing the issue of delayed feedback and **fake negatives**, without additional engineering cost
- **Importance sampling** allows to obtain an unbiased estimate of the expectation from the biased distribution ***b*** that includes fake negatives; ***p*** is the unobserved unbiased distribution

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

### Continuous training framework




## #LossFunctions

### Delayed feedback loss [1]

Assumes a separate model for the feedback delay ***d*** following an exponential distribution (no fake negatives):

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

This is trained jointly with the pCTR model and is discarded at inference time. Numerically stable formulation:

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

### Positive unlabelled loss [2]

Treat all negative examples in biased training data as unlabelled and leverage the fact that

$$p(\mathbf{y}=0)\mathbb{E}_{p(\mathbf{x}|\mathbf{y}=0)}\big[l(1-f_\theta(\mathbf{x}))\big] =$$
$$\mathbb{E}_{p(\mathbf{x})}\big[l(1-f_\theta(\mathbf{x}))\big] - p(\mathbf{y}=1)\mathbb{E}_{p(\mathbf{x}|\mathbf{y}=1)}\big[l(1-f_\theta(\mathbf{x}))\big]$$

To obtain:

$$\mathcal{L}_{PU}(\theta) = -\sum_{\mathbf{x},\mathbf{y}=1}[\log f_\theta(\mathbf{x}) - \log(1-f_\theta(\mathbf{x}))] - \sum_{\mathbf{x},\mathbf{y}=0}\log(1-f_\theta(\mathbf{x}))$$

### Fake negative weighted loss

Based on importance sampling

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

For **fake negative calibration** use transformation to model output

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

## #ExperimentsAndResults

### Offline experiments - Public data

- Train on 15.5 million examples, evaluate on 3.5 million examples

| Logistic regression - Criteo data [1] | | | |
|---|---|---|---|
| **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: improvement of a prediction relative to the straw man

### Offline experiments - Twitter data

- Train on 668 million video ads, evaluate on 7 million 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 |

### Online experiments

| Wide & deep | | | |
|---|---|---|---|
| **Loss function** | **Pooled RCE** | **RPMq** | **Monerized 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** |

- PU loss diverged online after 2 days
- RPMq [3]: revenue made per 1000 ad requests

## #References

[1] O. Chapelle, "Modeling delayed feedback in display advertising", KDD, 2014.
[2] M. Du Plessis, G. Niu, M. Sugiyama, "Convex formulation for learning from positive and unlabeled data", ICML, 2015.
[3] C. Li, Y. Lu, Q. Mei, D. Wang, S. Pandey, "Clickthrough prediction for advertising in twitter timeline", KDD, 2015.