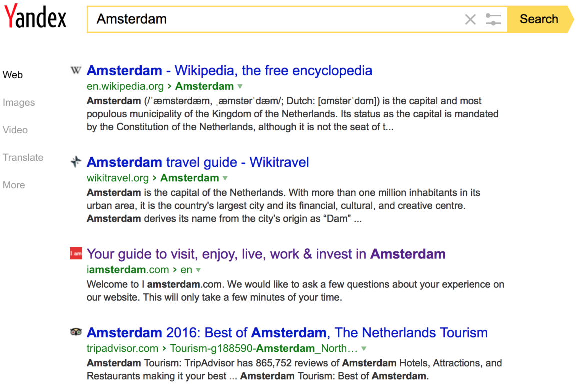
Outline

Understanding user behavior is the key

The ability to accurately predict the behavior of a particular user allows search engines to construct optimal result pages

User behavioral signals



Actions
(e.g., click, first/last click, long click, satisfied click, repeated click)

Times between actions
(e.g., time between clicks, time to first/last click)

Interpretation is difficult

Biases in user behavior — (statistically significant) differences between probability distributions of user behavioral signal observed in different contexts

Clicks are biased towards:

- higher ranked results (position bias)
- visually salient results (attention bias)
- previously unseen results (novelty bias)

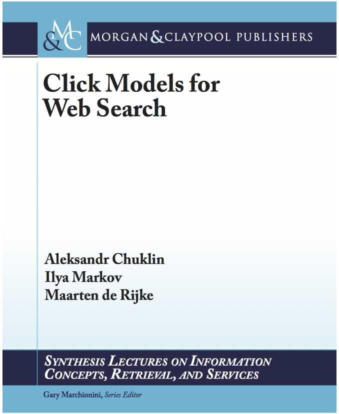
CLICK MODELS
account for biases in clicks

Click dwell times are biased

Times to first/last/satisfied clicks are biased

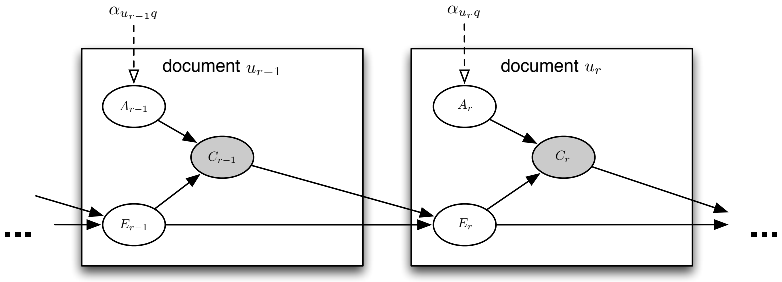
Applications of user behavior models

- Understand users
- Simulate users
- Features for ranking
- Evaluate search



Outline

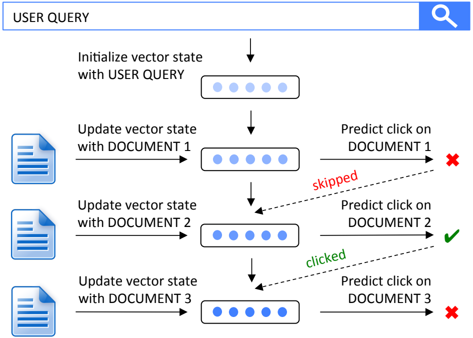
Traditional click models



Graphical representation of the cascade click model.

Pros: Based on the probabilistic graphical model (PGM) framework
Cons: Structure of the underlying PGM has to be set manually

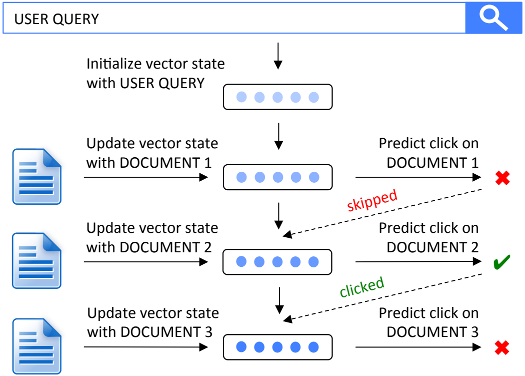
Neural click modeling framework



A neural click model for web search [Borisov et al., 2016].

Learns patterns of user behavior directly from click-through data

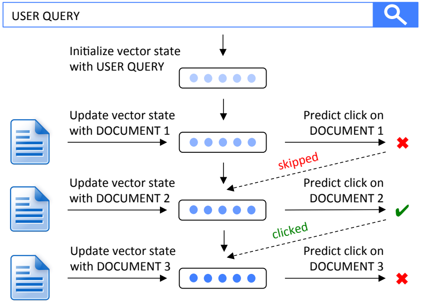
Distributed representations $(\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \ldots)$



We model user browsing behavior as a sequence of vector states $(\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \ldots)$ that describes the information consumed by the user as it evolves during a query session.

Mappings I, U and Function F

$$\mathbf{s}_0 = \mathcal{I}(q)$$
$$\mathbf{s}_{r+1} = \mathcal{U}(\mathbf{s}_r, i_r, d_{r+1})$$



$$P(C_{r+1} = 1 \mid q, i_1, \ldots, i_r, d_1, \ldots, d_{r+1}) = \mathcal{F}(\mathbf{s}_{r+1})$$

q — user query

d_r — document at rank r

i_r — user interaction with document at rank r

Neural click modeling framework $\rightarrow$ $\text{NCM}^{\{\text{RNN, LSTM}\}}_{\{\text{QD, QD+Q, QD+Q+D}\}}$

Representations of q, d_r and i_r

Use three sets: QD, QD+Q, QD+Q+D

Parameterization of $\mathcal{I}$, $\mathcal{U}$ and $\mathcal{F}$

$\mathcal{I}$ Feed-forward neural network
$\mathcal{U}$ Recurrent neural network (RNN, LSTM)
$\mathcal{F}$ Feed-forward neural network
(with one output unit and the sigmoid activation function)

Training

Stochastic gradient descent
(with AdaDelta update rules and gradient clipping)

Experimental setup

Dataset

Yandex Relevance Prediction dataset[1]
($146, 278, 823$ query sessions)

Tasks and evaluation metrics

Click prediction task (log-likelihood, perplexity)
Relevance prediction task (NDCG)

Baselines

Dynamic Bayesian network (DBN), Dependent click model (DCM)
Click chain model (CCM), User browsing model (UBM)

Results on click prediction task

Click model	Perplexity	Log-likelihood
DBN	1.3510	−0.2824
DCM	1.3627	−0.3613
CCM	1.3692	−0.3560
UBM	1.3431	−0.2646
$\text{NCM}^{\text{RNN}}_{\text{QD}}$	1.3379	−0.2564
$\text{NCM}^{\text{LSTM}}_{\text{QD}}$	1.3362	−0.2547
$\text{NCM}^{\text{LSTM}}_{\text{QD+Q}}$	1.3355	−0.2545
$\text{NCM}^{\text{LSTM}}_{\text{QD+Q+D}}$	1.3318	−0.2526

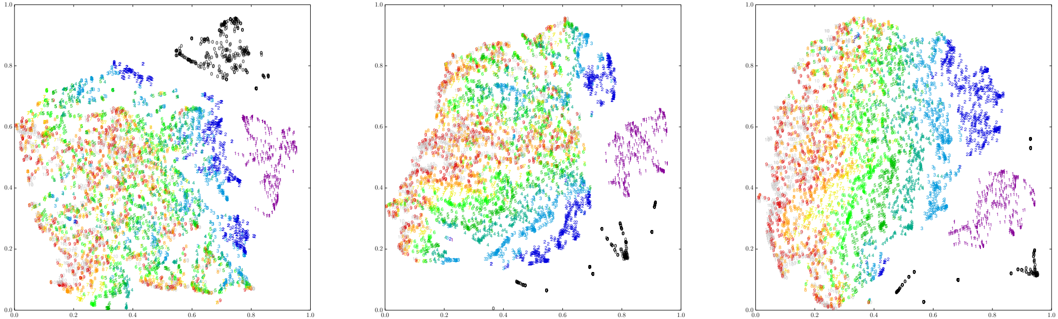
Differences between all pairs of click models are statistically significant $(p < 0.001)$

Results on relevance prediction task

	NDCG			
Click model	@1	@3	@5	@10
DBN	0.717	0.725	0.764	0.833
DCM	0.736	0.746	0.780	0.844
CCM	0.741	0.752	0.785	0.846
UBM	0.724	0.737	0.773	0.838
$\text{NCM}_{\text{QD}}^{\text{RNN}}$	0.762	0.759	0.791	0.851
$\text{NCM}_{\text{QD}}^{\text{LSTM}}$	0.756	0.759	0.789	0.850
$\text{NCM}_{\text{QD+Q}}^{\text{LSTM}}$	0.775	0.773	0.799	0.857
$\text{NCM}_{\text{QD+Q+D}}^{\text{LSTM}}$	0.755	0.755	0.787	0.847

Improvements of $\text{NCM}_{\text{QD}}^{\text{RNN}}$, $\text{NCM}_{\text{QD}}^{\text{LSTM}}$ and $\text{NCM}_{\text{QD+Q}}^{\text{LSTM}}$ over baseline click models are statistically significant ($p < 0.05$)

Analysis

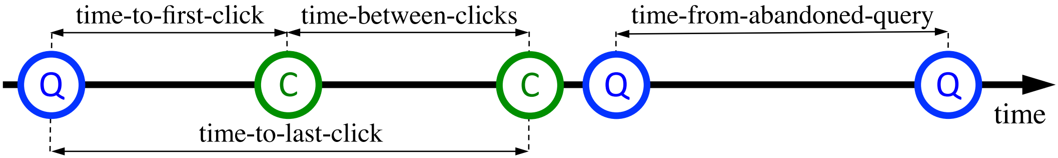


Learns regularities in user browsing behavior that

1. have been manually encoded in existing click models, such as **ranks** and **distances to previous clicks** (large clusters on t-SNE projections of vector states $\mathbf{s}_r$)
2. can not be manually encoded in traditional click models (small clusters on t-SNE projections of vector states $\mathbf{s}_r$)

Outline

Times between user actions



- **time-to-first-click** (reflects quality of result presentation)
- **time-between-clicks** (proxy for click dwell time)
- **time-to-last-click** (reflects quality of search engine results)
- **time-from-abandoned-query** (reflects quality of search engine results in query sessions with no clicks)

How to interpret times between user actions

Average

$$\widehat{t} = \frac{1}{N}\sum_{i=1}^{N}\tau_i$$

Uncertainty: $\frac{1}{2}(3 + 600)$ vs. $\frac{1}{7}(30 + 28 + 45 + 23 + 100 + 23 + 58)$

How to interpret times between user actions

Average

$$\widehat{t} = \frac{1}{N}\sum_{i=1}^{N} \tau_i$$

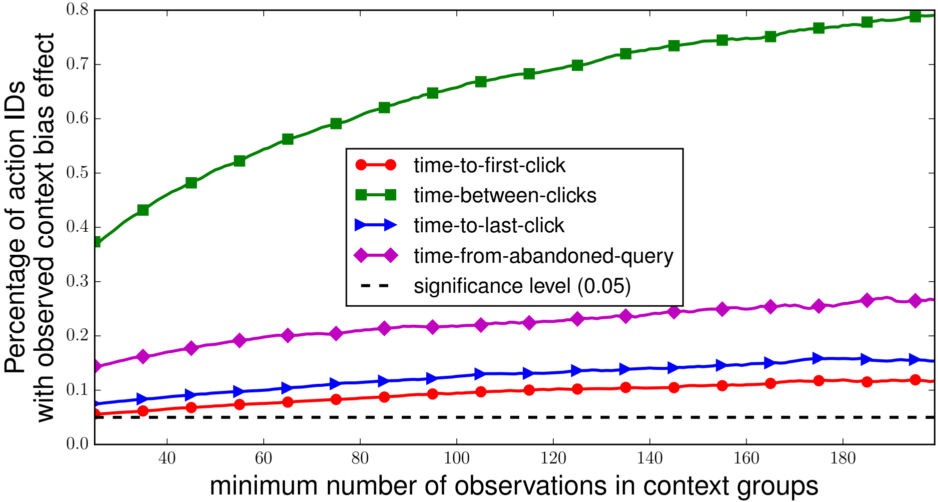
Uncertainty: $\frac{1}{2}(3+600)$ vs. $\frac{1}{7}(30+28+45+23+100+23+58)$

Fit distribution (e.g., exponential, gamma, Weibull)

$$\widehat{\theta} = \arg\max_{\theta} \prod_{i=1}^{N} f(\tau_i \mid \theta)$$

Context bias: $f_{\text{high expectation}}(\tau = 15 \mid \theta_1)$ vs. $f_{\text{low expectation}}(\tau = 10 \mid \theta_2)$

Detected context bias effect



Context-aware time modeling (naive)

$$Time(action, context) \sim Gamma(\boldsymbol{k}(act, ctx), \boldsymbol{\theta}(act, ctx))$$

Context-aware time modeling

$$
\begin{aligned}
Time(action, context) \sim Gamma(&\\
&\boldsymbol{a}_k(ctx) \cdot \boldsymbol{k}(act) + \boldsymbol{b}_k(ctx),\\
&\boldsymbol{a}_\theta(ctx) \cdot \boldsymbol{\theta}(act) + \boldsymbol{b}_\theta(ctx))
\end{aligned}
$$

Parameter estimation

$$
\begin{aligned}
Time(action, context) \sim Gamma(&\\
&\boldsymbol{a}_k(ctx) \cdot \boldsymbol{k}(act) + \boldsymbol{b}_k(ctx),\\
&\boldsymbol{a}_\theta(ctx) \cdot \boldsymbol{\theta}(act) + \boldsymbol{b}_\theta(ctx))
\end{aligned}
$$

1. Fix context-independent parameters
2. Optimize context-dependent parameters using *neural networks*
3. Fix context-dependent parameters
4. Optimize context-independent using *gradient descent*
5. Repeat until convergence

Parameter estimation

- We do not know the form of context-dependent parameters
  $\Longrightarrow$ neural networks

- We know the form of context-independent parameters (Gamma distribution)
  $\Longrightarrow$ direct optimization

Dataset

3 months of log data from Yandex search engine

Time between actions	Max time	# Observations
Time-to-first-click	1 min	$30,747,733$
Time-between-clicks	5 min	$6,317,834$
Time-to-last-click	5 min	$30,446,973$
Time-from-abandoned-query	1 min	$11,523,351$

Evaluation tasks

Task1. Predict time between clicks

- Log-likelihood
- Root mean squared error (MSE)

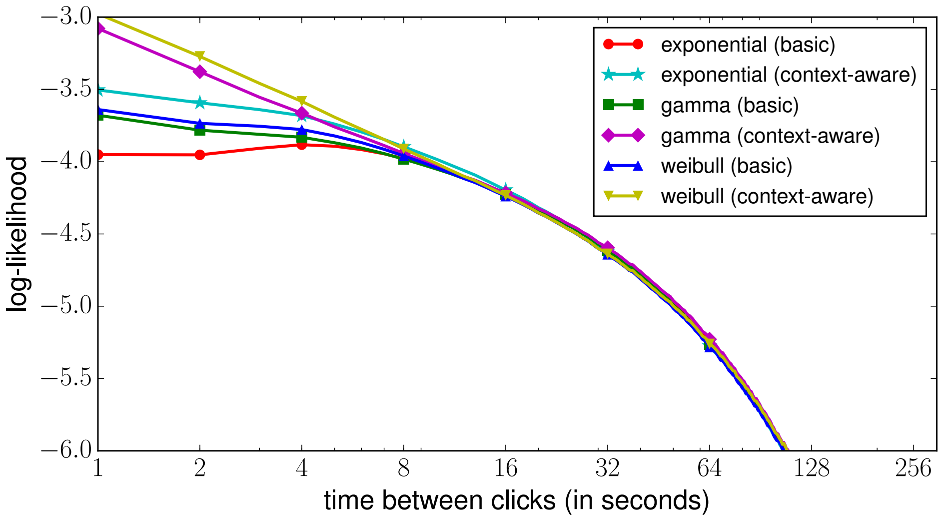
Task2. Rank results based on time between clicks

- nDCG@$\{1, 3, 5, 10\}$

Task 1. Predicting time

Time model	Distribution	Log-likelihood	RMSE
Basic	exponential	-4.9219	60.73
	gamma	-4.9105	60.76
	Weibull	-4.9077	60.76
Context-aware	exponential	-4.8787	58.93
	gamma	-4.8556	58.98
	Weibull	-4.8504	58.94

Results on time prediction task (time-between-clicks)



Task 2. Ranking results

		NDCG			
Time model	Distribution	@1	@3	@5	@10
Average	—	0.651	0.693	0.728	0.812
Context-aware	exponential	0.668	0.710	0.743	0.820
	gamma	0.675	0.715	0.748	0.822
	Weibull	0.671	0.709	0.745	0.821

Summary

- Remove context bias from time between actions
- Predict user search interactions better (**Task 1**)
- Use the context-independent component for better document ranking (**Task 2**)

Outline

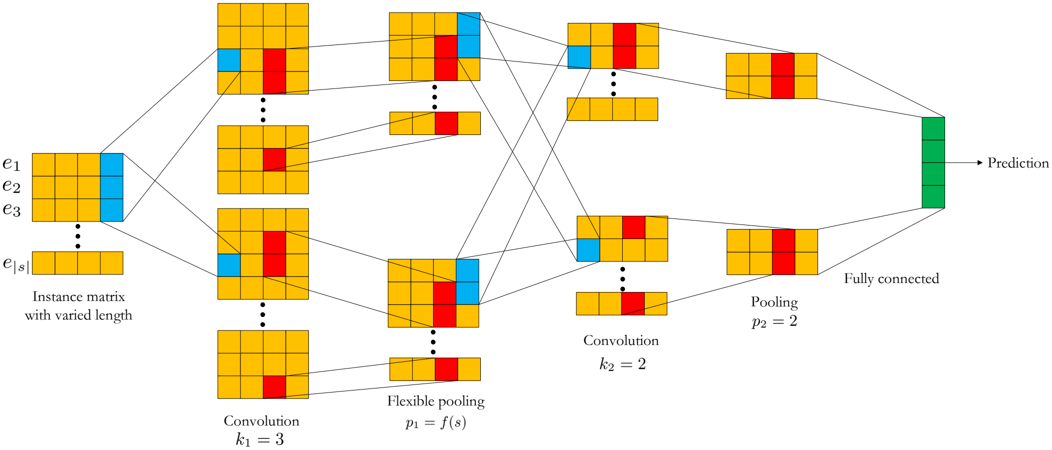
Web search vs. sponsored search

In web search we work (mostly) with query sessions and search sessions
In sponsored search we need to consider longer user histories

Recurrent Neural Networks (RNNs) can be used not only to account for biases, but also to infer user interests and behavioral patterns from very long sequences of user actions

Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks [Zhang et al., 2014]

Web search vs. sponsored search



A Convolutional Click Prediction Model [Liu et al., 2015]

Outline

Future work

Neural Networks — an alternative to probabilistic graphical models (PGMs) that allows to learn patterns of user behavior directly from the data

Understanding and modelling user behavior with PGMs — is a mature field
We expect many ideas to be transfered from PGM to neural framework

By 2027 (50th anniversary of SIGIR), user behavior models

- will learn patterns of user behavior directly from the data
- will take very long user history into account
- will extract signals from images, videos, user voice and background sounds
- will improve our understanding of humankind