NEURAL NETWORKS AND CAUSAL RECOMMENDATION

Part III. Bandit Feedback and Likelihood Models for Recommendation

David Rohde, Olivier Jeunen, Flavian Vasile

Criteo Research

criteo

Collaborators:

Special thanks to:

- Dmytro Mykhaylov
- Martin Bompaire
- Stephen Bonner

criteo

Getting started with RecoGym and Google Colaboratory

- Open your favourite browser, and go to the RecoGym repository: `github.com/criteo-research/reco-gym`
- Go to the DS3 branch, and open the slides on your laptop (they can be found in the 'slides' folder)
- URLs linking to Google Colaboratory notebooks for the hands-on sessions:
  1. Exercise A
  2. Exercise B
  3. Exercise C
- Alternatively, clone the repository and run the notebooks locally.
- . . . you're all set!

criteo

criteo

AI Lab

http://cail.criteo.com

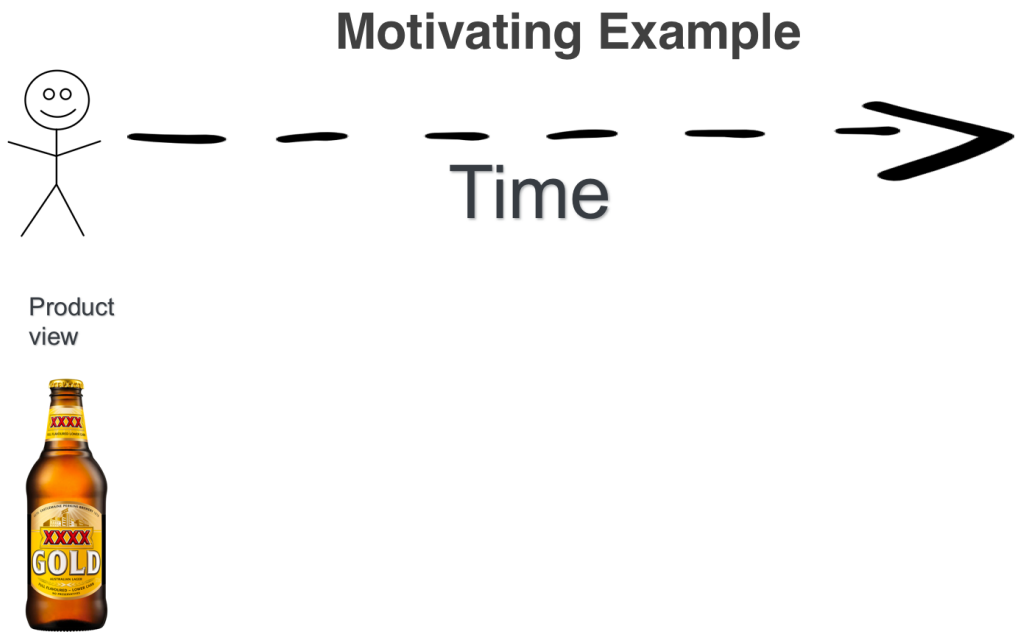
Notation

Symbol	Dimension	Description
u	Scalar	A given users id.
t	Scalar	sequential time.
P	Scalar	Total number of products.
v_t	Scalar	Product id viewed by user at time t.
r_t	Scalar	Product id recommended at time t.
c_t	Scalar	Product id recommended at time t.

Table 1: Notations and Definitions

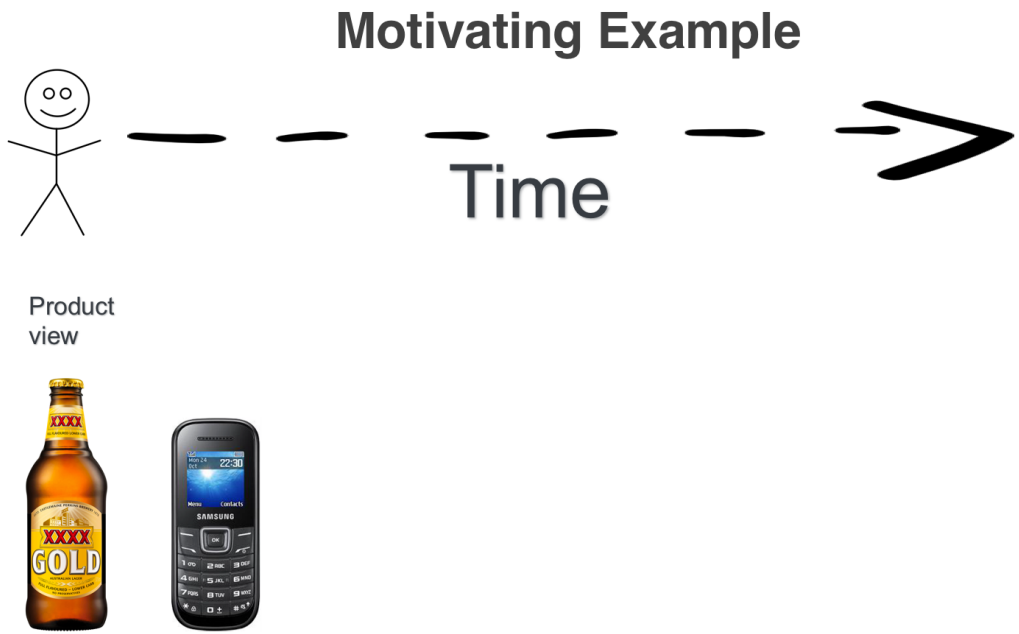
criteo

Classic Reco



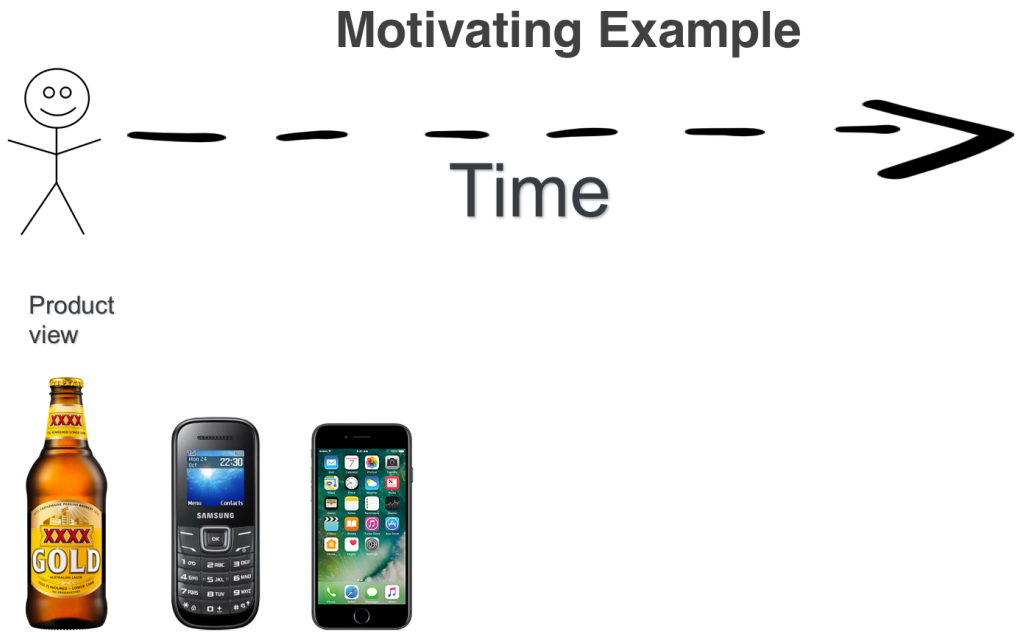
criteo

Classic Reco



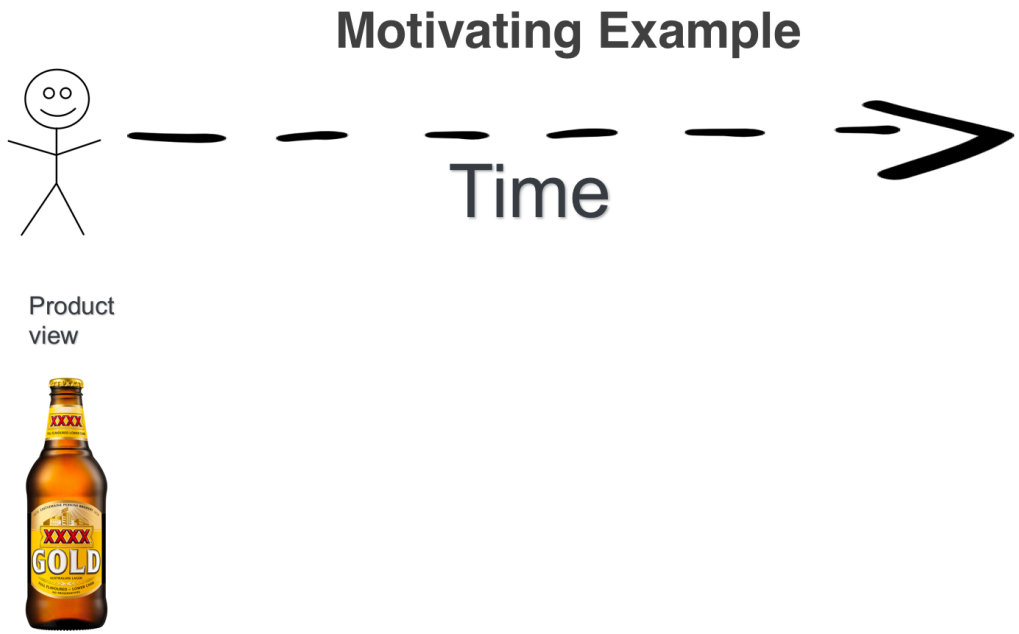
criteo

Classic Reco



criteo

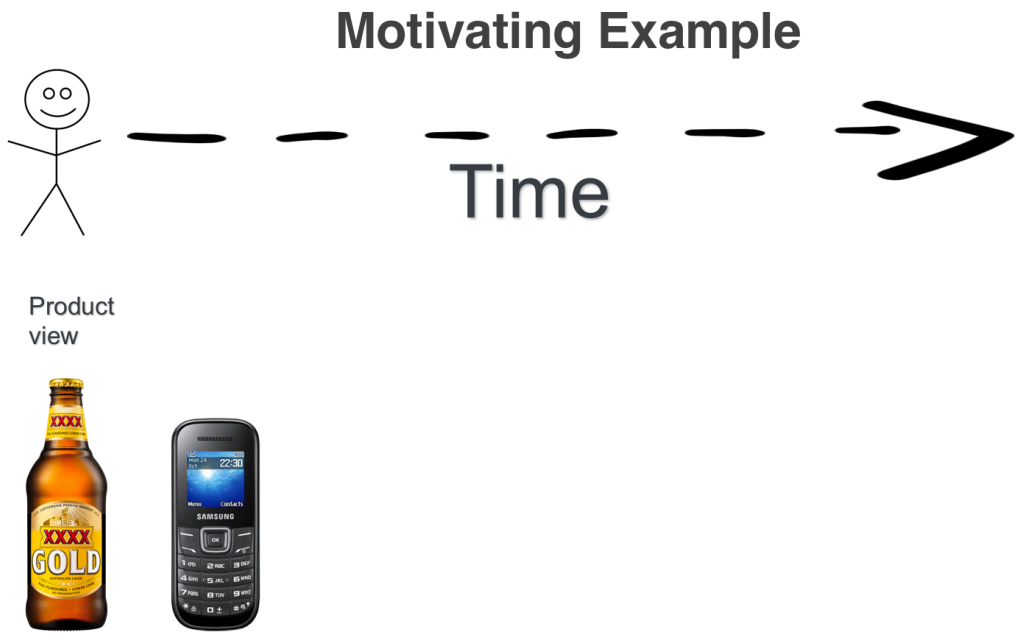
Classic Reco



$$P(V_1 = \text{beer})$$

criteo

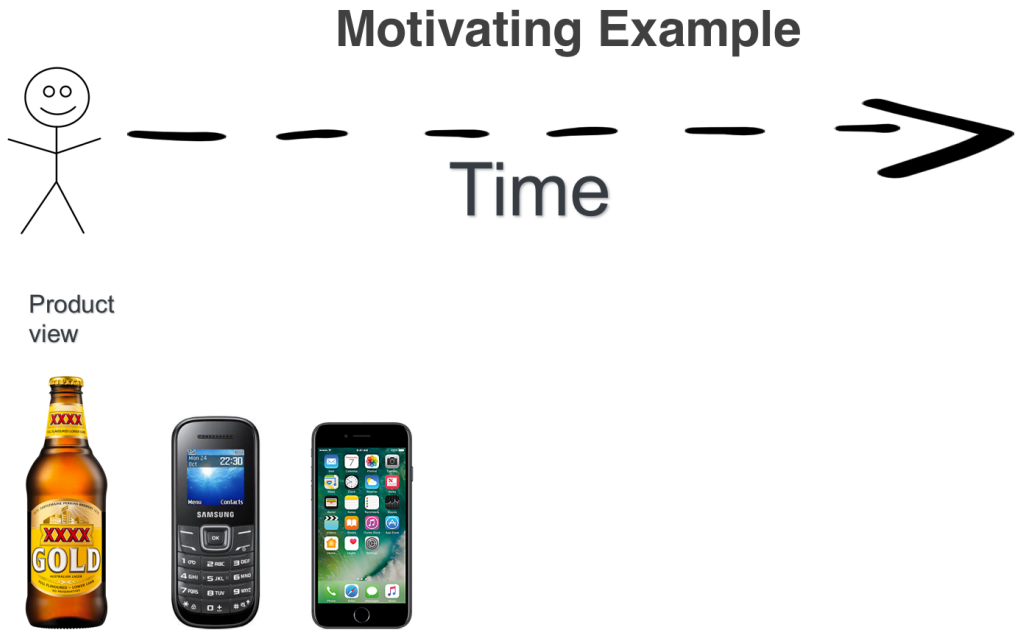
Classic Reco



$$P(V_2 = \text{phone A} | V_1 = \text{beer})$$

criteo

Classic Reco



$$P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A})$$

criteo

Next Item Prediction, hold out V_3

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

criteo

Next Item Prediction, hold out V_3

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened?

criteo

Next Item Prediction, hold out V_3

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened? $V_3 = \text{phone B}$

criteo

Next Item Prediction, hold out V_3

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened? $V_3 = \text{phone B}$

Our model assigned: $P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) = 0.24$

criteo

Next Item Prediction, hold out V_3

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened? $V_3 = \text{phone B}$

Our model assigned: $P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) = 0.24$

Of course better would be higher, but how good is this?

criteo

Next Item Prediction, hold out V_3: Log likelihood

What actually happened?

criteo

Next Item Prediction, hold out V_3: Log likelihood

What actually happened? $V_3 = \text{phone B}$

criteo

Next Item Prediction, hold out V_3: Log likelihood

What actually happened? $V_3 = \text{phone B}$

Our model assigned: $P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) = 0.24$

criteo

Next Item Prediction, hold out V_3: Log likelihood

What actually happened? $V_3 = \text{phone B}$

Our model assigned: $P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) = 0.24$

Log Likelihood:
$\log P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) = -1.427116$

criteo

Next Item Prediction, hold out V_3: Log likelihood

What actually happened? $V_3 = \text{phone B}$

Our model assigned: $P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) = 0.24$

Log Likelihood:
$\log P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) = -1.427116$

Rationale: you want to assign high values to things that happend.

criteo

Next Item Prediction, hold out V_3: HR@K Recall@K Precision@K - K=2

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened?

Next Item Prediction, hold out V_3: HR@K Recall@K Precision@K - K=2

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened? $V_3 = \text{phone B}$

criteo

Next Item Prediction, hold out V_3: HR@K Recall@K Precision@K - K=2

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened? $V_3 = \text{phone B}$

Our model assigned:
$P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) = 0.24$, which is 3rd place

How good is 3rd place?

criteo

Next Item Prediction, hold out V_3: HR@K Recall@K Precision@K - K=2

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened? $V_3 = \text{phone B}$

Our model assigned:
$P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) = 0.24$, which is 3rd place

How good is 3rd place? HR@K if K=2, our model assigned phone B the third highest spot so it just misses out.

criteo

Next Item Prediction, hold out V_3: HR@K Recall@K Precision@K - K=3

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened?

criteo

Next Item Prediction, hold out V_3: HR@K Recall@K Precision@K - K=3

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened? $V_3 = \text{phone B}$

criteo

Next Item Prediction, hold out V_3: HR@K Recall@K Precision@K - K=3

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened? $V_3 = \text{phone B}$

Our model assigned:
$P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) = 0.24$, which is 3rd place

How good is 3rd place?

criteo

Next Item Prediction, hold out V_3: HR@K Recall@K Precision@K - K=3

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened? $V_3 = \text{phone B}$

Our model assigned:
$P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) = 0.24$, which is 3rd place

How good is 3rd place? HR@K if K=3, our model assigned phone B the third highest spot so it is in.

criteo

Next Item Prediction, hold out V_3: HR@K Recall@K Precision@K - K=3

$$
\begin{aligned}
P(V_3 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.27 \\
P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.24 \\
P(V_3 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.10 \\
P(V_3 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.09 \\
P(V_3 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}) &= 0.30
\end{aligned}
$$

What actually happened? $V_3 = \text{phone B}$

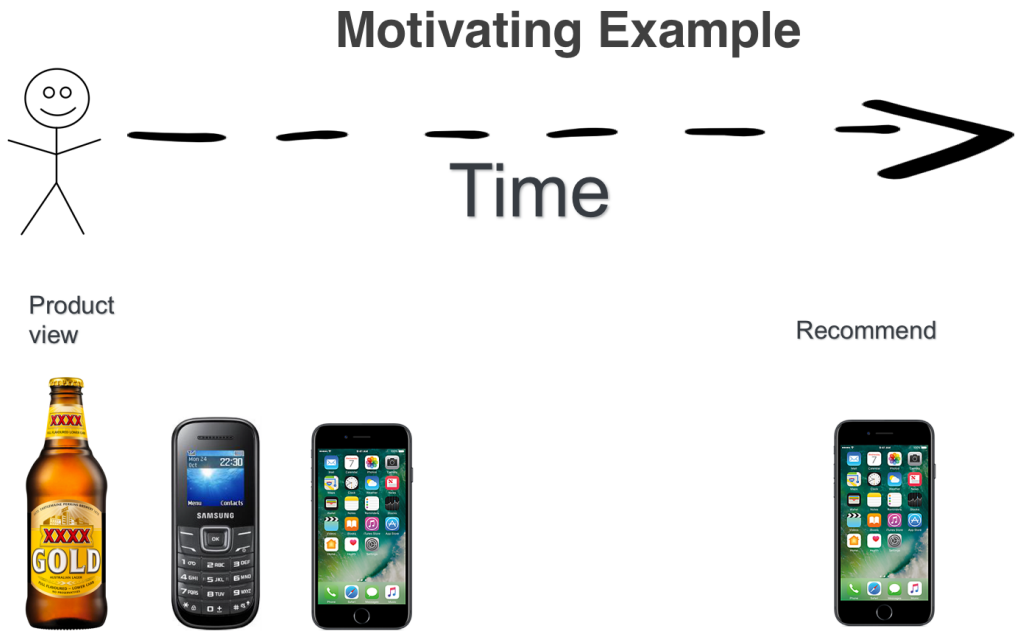
Our model assigned:
$P(V_3 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}) = 0.24$, which is 3rd place

How good is 3rd place? HR@K if K=3, our model assigned phone B the third highest spot so it is in.

Rationale: you want a ranking that assigns high positions to events that occur. The position matters more than the probability.

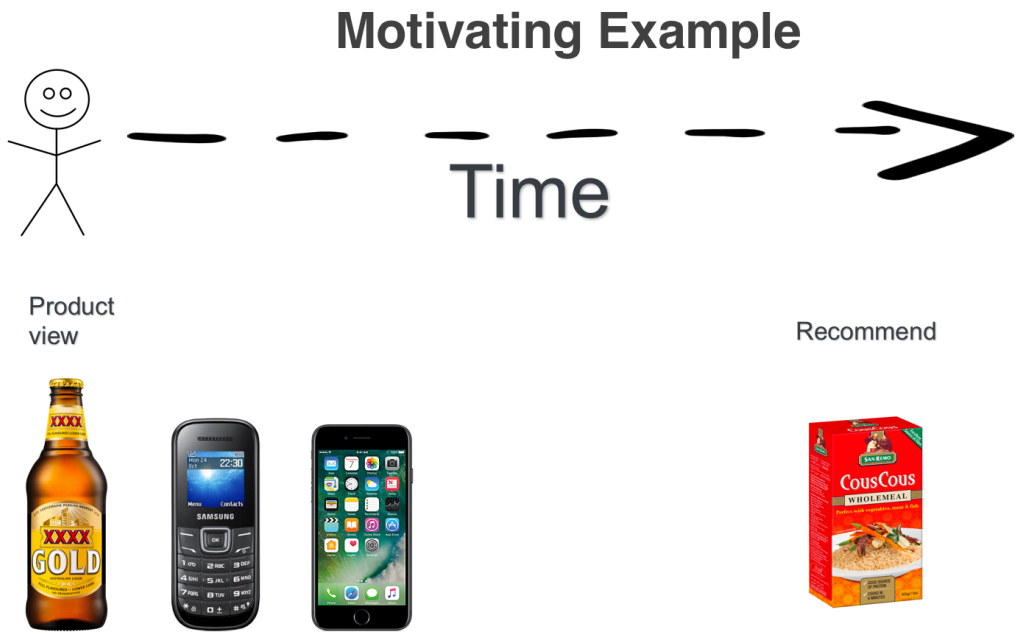
criteo

Real Reco

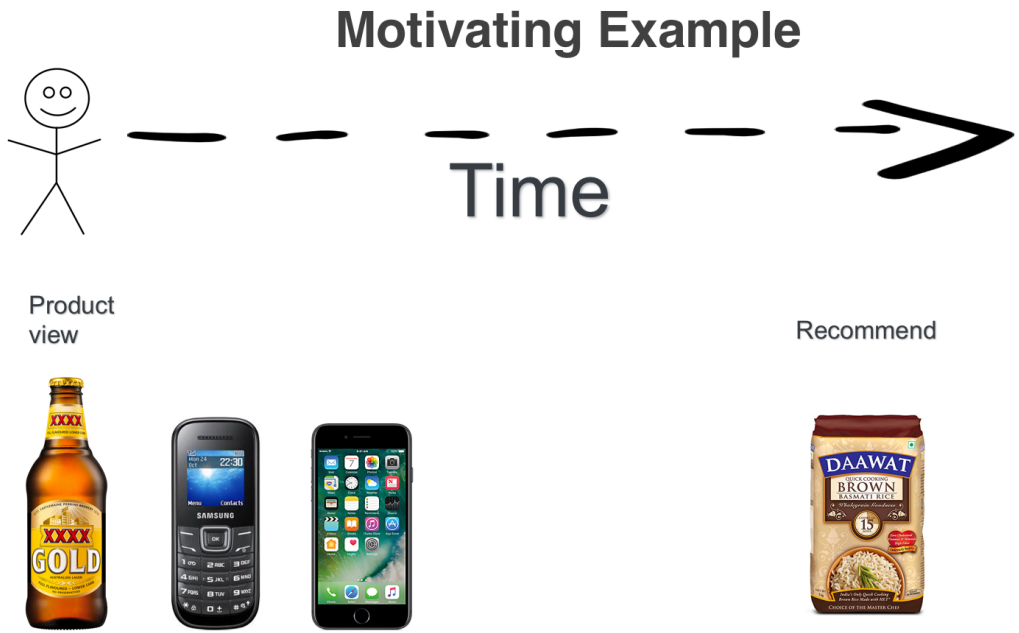


criteo

Real Reco

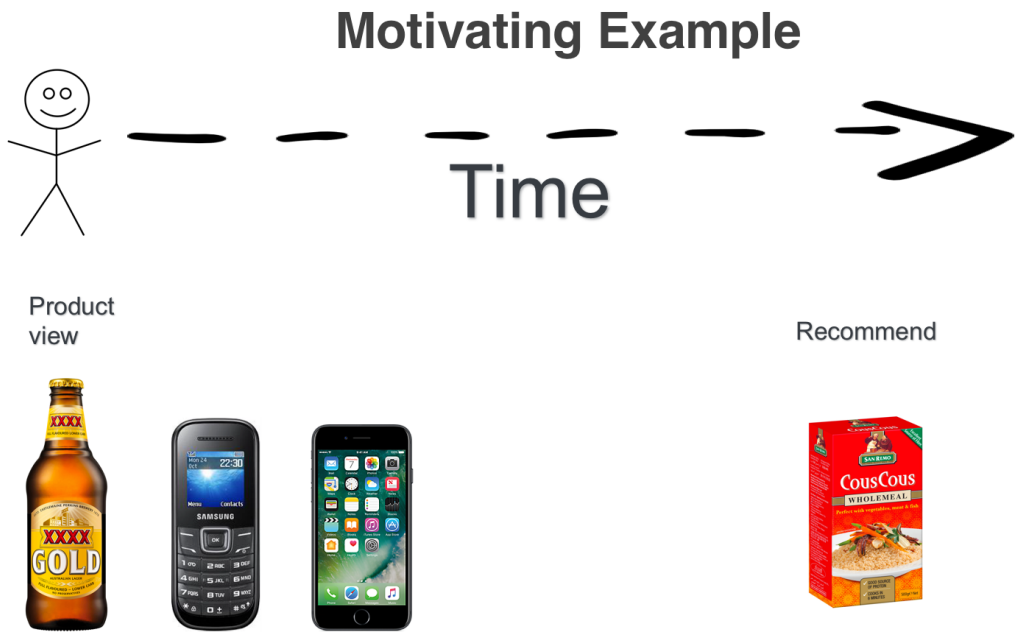

Motivating Example
Time
Product view
Recommend
XXXX
GOLD
SAMSUNG
CousCous
WHOLEMEAL

Real Reco



criteo

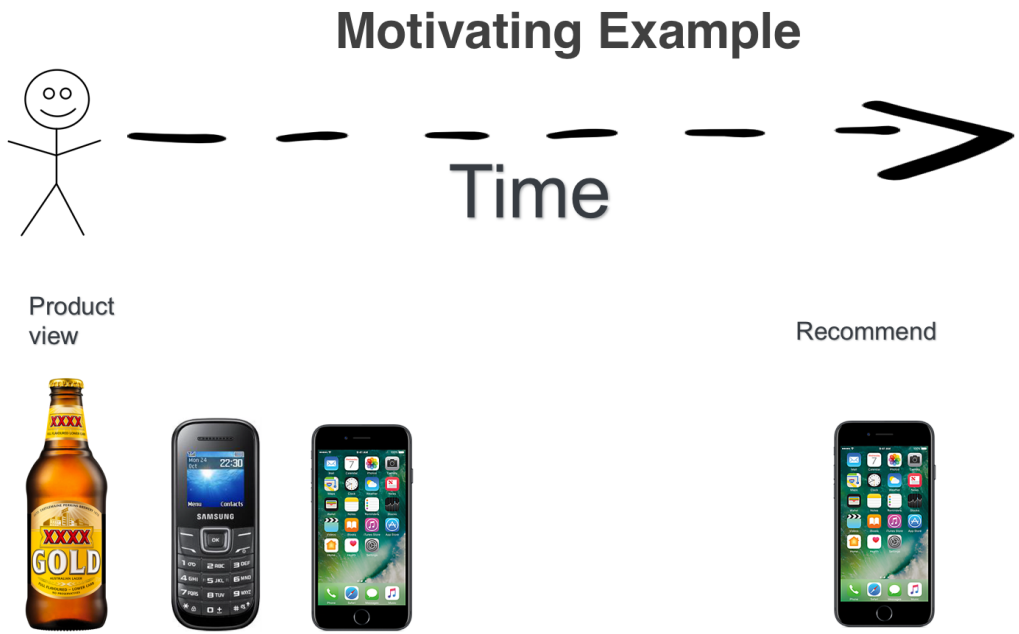
The Real Reco Problem



$$P(C_4 = 1 | A_4 = \text{couscous}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone B})$$

criteo

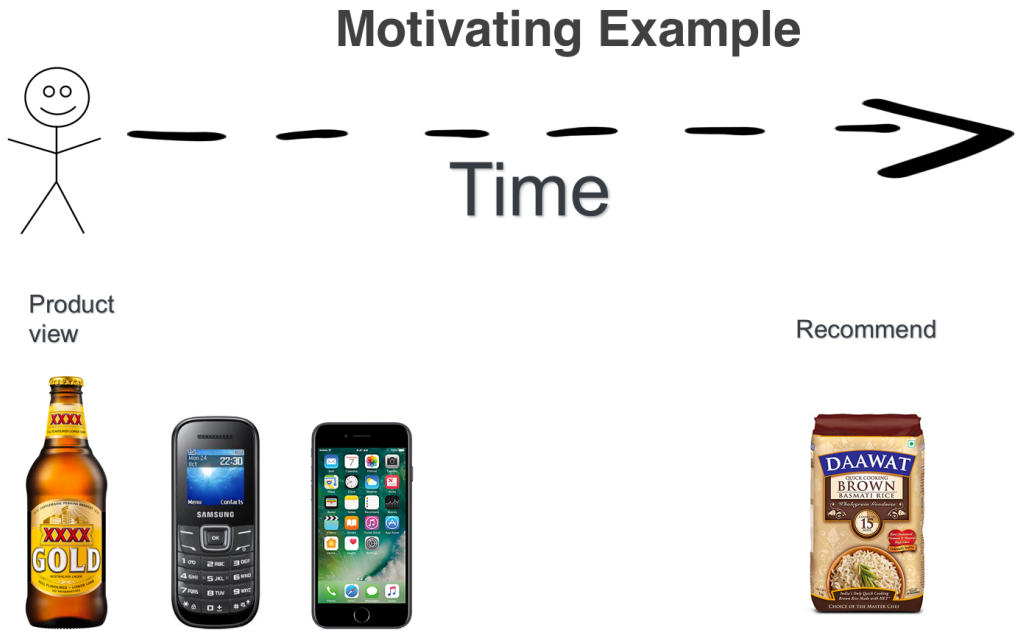
The Real Reco Problem



$$P(C_4 = 1 | A_4 = \text{phone B}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone B})$$

criteo

The Real Reco Problem



$$P(C_4 = 1 | A_4 = \text{rice}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone B})$$

criteo

What we have, and what we want:

Our model:

$$P(V_4 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.27$$
$$P(V_4 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.28$$
$$P(V_4 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = .12$$
$$P(V_4 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = .14$$
$$P(V_4 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.19$$

criteo

What we have, and what we want:

Our model:

$$
\begin{aligned}
P(V_4 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) &= 0.27 \\
P(V_4 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) &= 0.28 \\
P(V_4 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) &= .12 \\
P(V_4 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) &= .14 \\
P(V_4 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) &= 0.19
\end{aligned}
$$

Which is a vector of size P that sums to 1.

We want:

criteo

What we have, and what we want:

Our model:

$$P(V_4 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.27$$
$$P(V_4 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.28$$
$$P(V_4 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = .12$$
$$P(V_4 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = .14$$
$$P(V_4 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.19$$

Which is a vector of size P that sums to 1.

We want:

$$P(C_4 = 1 | A_4 = \text{phone A}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = ?$$
$$P(C_4 = 1 | A_4 = \text{phone B}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = ?$$
$$P(C_4 = 1 | A_4 = \text{rice}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = ?$$
$$P(C_4 = 1 | A_4 = \text{couscous}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = ?$$
$$P(C_4 = 1 | A_4 = \text{beer}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = ?$$

criteo

What we have, and what we want:

Our model:

$$P(V_4 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.27$$
$$P(V_4 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.28$$
$$P(V_4 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = .12$$
$$P(V_4 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = .14$$
$$P(V_4 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.19$$

Which is a vector of size P that sums to 1.

We want:

$$P(C_4 = 1 | A_4 = \text{phone A}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = ?$$
$$P(C_4 = 1 | A_4 = \text{phone B}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = ?$$
$$P(C_4 = 1 | A_4 = \text{rice}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = ?$$
$$P(C_4 = 1 | A_4 = \text{couscous}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = ?$$
$$P(C_4 = 1 | A_4 = \text{beer}, V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = ?$$

Which is a vector of size P where each entry is between 0 and 1.

criteo

An implicit assumption:

Our model predicts:

$$
\begin{aligned}
P(V_4 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) &= 0.27 \\
P(V_4 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) &= 0.28 \\
P(V_4 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) &= .12 \\
P(V_4 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) &= .14 \\
P(V_4 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) &= 0.19
\end{aligned}
$$

So let's recommend phone B, it has the highest next item probability.

criteo

An implicit assumption:

If:

criteo

An implicit assumption:

If:

$$P(V_n = a | V_1 = v_1 .. V_{n-1} = v_{n-1}) > P(V_n = b | V_1 = v_1 .. V_{n-1} = v_{n-1})$$

criteo

An implicit assumption:

If:

$$P(V_n = a | V_1 = v_1 .. V_{n-1} = v_{n-1}) > P(V_n = b | V_1 = v_1 .. V_{n-1} = v_{n-1})$$

Assume:

$$\begin{aligned} &P(C_n = 1 | A_n = a, V_1 = v_1 .. V_{n-1} = v_{n-1}) \\ &\qquad > P(C_n = 1 | A_n = b, V_1 = v_1 .. V_{n-1} = v_{n-1}) \end{aligned}$$

criteo

An implicit assumption:

If:

$$P(V_n = a | V_1 = v_1 .. V_{n-1} = v_{n-1}) > P(V_n = b | V_1 = v_1 .. V_{n-1} = v_{n-1})$$

Assume:

$$\begin{aligned} &P(C_n = 1 | A_n = a, V_1 = v_1 .. V_{n-1} = v_{n-1}) \\ &\qquad > P(C_n = 1 | A_n = b, V_1 = v_1 .. V_{n-1} = v_{n-1}) \end{aligned}$$

Vast amounts of academic Reco work assumes this implicitly!

criteo

The assumption is testable

Our model predicts:

$$P(V_4 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.27$$
$$P(V_4 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.28$$
$$P(V_4 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = .12$$
$$P(V_4 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = .14$$
$$P(V_4 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.19$$

So let's recommend phone B, it has the highest next item probability.

criteo

The assumption is testable

Our model predicts:

$$P(V_4 = \text{phone A} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.27$$
$$P(V_4 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.28$$
$$P(V_4 = \text{rice} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = .12$$
$$P(V_4 = \text{couscous} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = .14$$
$$P(V_4 = \text{beer} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone A}) = 0.19$$

So let's recommend phone B, it has the highest next item probability. - but we will also sometimes choose other actions with a small probability, so we can check if these items have lower click through rates.

criteo

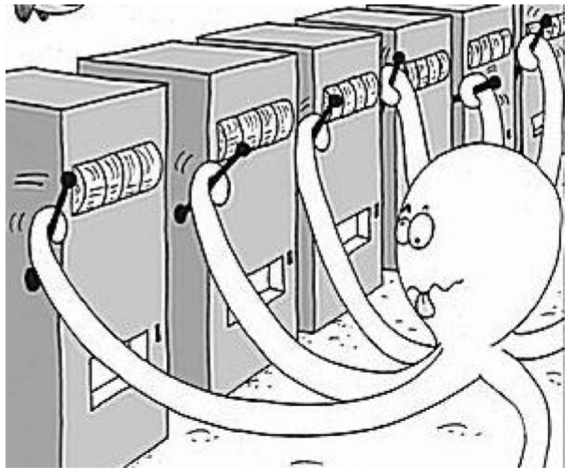
History: Split History of Recommender Systems

Next Item or Missing Link Prediction

e.g. Netflix Prize or Movie Lens

	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6	movie 7	movie 8	movie 9	movie 10
user 1			1		2					
user 2		2		3	3			4		
user 3							5	3		4
user 4	2				3			2		
user 5		4				5			3	
user 6			2							
user 7			2					4	2	3
user 8	3	4				4				
user 9									3	
user 10			1		2					

Computational Advertising, e.g. Bandits, Conterfactual Risk Minimisation, Contextual Bandits, Reinforcement Learning.



criteo

Classic RecSys Datasets: Are they logs of recommender systems?

- MovieLens:

Classic RecSys Datasets: Are they logs of recommender systems?

- MovieLens: no, explicit feedback of movie ratings

criteo

Classic RecSys Datasets: Are they logs of recommender systems?

- MovieLens: no, explicit feedback of movie ratings
- Netflix Prize:

Classic RecSys Datasets: Are they logs of recommender systems?

- MovieLens: no, explicit feedback of movie ratings
- Netflix Prize: no, explicit feedback of movie ratings

criteo

Classic RecSys Datasets: Are they logs of recommender systems?

- MovieLens: no, explicit feedback of movie ratings
- Netflix Prize: no, explicit feedback of movie ratings
- Yoochoose (RecSys 15):

Classic RecSys Datasets: Are they logs of recommender systems?

- MovieLens: no, explicit feedback of movie ratings
- Netflix Prize: no, explicit feedback of movie ratings
- Yoochoose (RecSys 15): no, implicit session based behavior

Classic RecSys Datasets: Are they logs of recommender systems?

- MovieLens: no, explicit feedback of movie ratings
- Netflix Prize: no, explicit feedback of movie ratings
- Yoochoose (RecSys 15): no, implicit session based behavior
- 30 Music:

Classic RecSys Datasets: Are they logs of recommender systems?

- MovieLens: no, explicit feedback of movie ratings
- Netflix Prize: no, explicit feedback of movie ratings
- Yoochoose (RecSys 15): no, implicit session based behavior
- 30 Music: no, implicit session based behavior

criteo

Classic RecSys Datasets: Are they logs of recommender systems?

- MovieLens: no, explicit feedback of movie ratings
- Netflix Prize: no, explicit feedback of movie ratings
- Yoochoose (RecSys 15): no, implicit session based behavior
- 30 Music: no, implicit session based behavior
- Yahoo News Feed Dataset

Classic RecSys Datasets: Are they logs of recommender systems?

- MovieLens: no, explicit feedback of movie ratings
- Netflix Prize: no, explicit feedback of movie ratings
- Yoochoose (RecSys 15): no, implicit session based behavior
- 30 Music: no, implicit session based behavior
- Yahoo News Feed Dataset yes!
- Criteo Dataset for counterfactual evaluation of RecSys algorithms:

Classic RecSys Datasets: Are they logs of recommender systems?

- MovieLens: no, explicit feedback of movie ratings
- Netflix Prize: no, explicit feedback of movie ratings
- Yoochoose (RecSys 15): no, implicit session based behavior
- 30 Music: no, implicit session based behavior
- Yahoo News Feed Dataset yes!
- Criteo Dataset for counterfactual evaluation of RecSys algorithms: yes!

criteo

Classic RecSys Datasets: Are they logs of recommender systems?

- MovieLens: no, explicit feedback of movie ratings
- Netflix Prize: no, explicit feedback of movie ratings
- Yoochoose (RecSys 15): no, implicit session based behavior
- 30 Music: no, implicit session based behavior
- Yahoo News Feed Dataset yes!
- Criteo Dataset for counterfactual evaluation of RecSys algorithms: yes!

The Criteo Dataset shows a log of recommendations and if they were successful in getting users to click.

criteo

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K:

criteo

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k

criteo

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction

criteo

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction
- DCG:

criteo

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction
- DCG: Are we assigning high score to an item

criteo

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction
- DCG: Are we assigning high score to an item no, evaluates next item prediction

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction
- DCG: Are we assigning high score to an item no, evaluates next item prediction
- Log likelihood:

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction
- DCG: Are we assigning high score to an item no, evaluates next item prediction
- Log likelihood: unclear, but ususally next item prediction

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction
- DCG: Are we assigning high score to an item no, evaluates next item prediction
- Log likelihood: unclear, but ususally next item prediction
- Inverse Propensity Score estimate of click through rate:

criteo

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction
- DCG: Are we assigning high score to an item no, evaluates next item prediction
- Log likelihood: unclear, but ususally next item prediction
- Inverse Propensity Score estimate of click through rate: to be explained later.

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction
- DCG: Are we assigning high score to an item no, evaluates next item prediction
- Log likelihood: unclear, but ususally next item prediction
- Inverse Propensity Score estimate of click through rate: to be explained later. yes! (although it is often noisy)

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction
- DCG: Are we assigning high score to an item no, evaluates next item prediction
- Log likelihood: unclear, but ususally next item prediction
- Inverse Propensity Score estimate of click through rate: to be explained later. yes! (although it is often noisy)
- AB Test:

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction
- DCG: Are we assigning high score to an item no, evaluates next item prediction
- Log likelihood: unclear, but ususally next item prediction
- Inverse Propensity Score estimate of click through rate: to be explained later. yes! (although it is often noisy)
- AB Test: i.e. run a randomized control trial live

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction
- DCG: Are we assigning high score to an item no, evaluates next item prediction
- Log likelihood: unclear, but ususally next item prediction
- Inverse Propensity Score estimate of click through rate: to be explained later. yes! (although it is often noisy)
- AB Test: i.e. run a randomized control trial liveyes, but the academic literature has no access to this

criteo

Classic RecSys Evaluation Metrics: Do they evaluate the quality of recommendation?

- Recall@K, Precision@K, HR@K: How often is an item in the top k no, evaluates next item prediction
- DCG: Are we assigning high score to an item no, evaluates next item prediction
- Log likelihood: unclear, but ususally next item prediction
- Inverse Propensity Score estimate of click through rate: to be explained later. yes! (although it is often noisy)
- AB Test: i.e. run a randomized control trial liveyes, but the academic literature has no access to this

If the dataset does not contain a log of recommendations and if they were successful, you cannot compute metrics of the recommendation quality.

criteo

Offline evaluation that predicts an AB test result

Imagine we have a new recommendation stratergy $\pi_t(A_n = a_n | V_1 = v_1, ..., V_n = v_{n-1})$, can we predict how well it will perform if we deploy it? We collect logs from a different policy: $\pi_0(A_n = a_n | V_1 = v_1, ..., V_n = v_{n-1})$

criteo

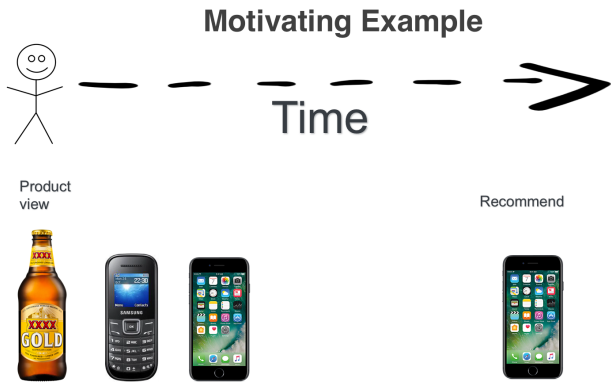
Offline evaluation that predicts an AB test result

Imagine we have a new recommendation stratergy $\pi_t(A_n = a_n | V_1 = v_1, ..., V_n = v_{n-1})$, can we predict how well it will perform if we deploy it? We collect logs from a different policy: $\pi_0(A_n = a_n | V_1 = v_1, ..., V_n = v_{n-1})$

Let's examine some hypothetical logs...

criteo

Offline evaluation IPS



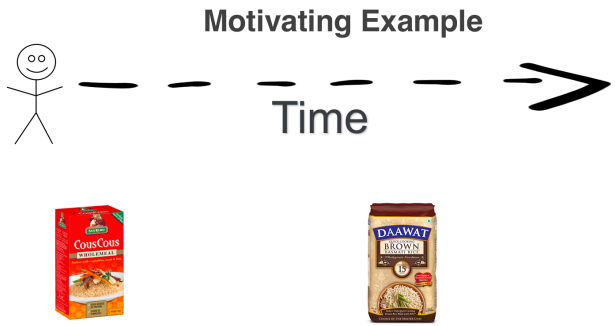
Imagine the user clicks

$$\pi_t(A_4 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone B}) = 1$$
$$\pi_0(A_4 = \text{phone B} | V_1 = \text{beer}, V_2 = \text{phone A}, V_3 = \text{phone B}) = 0.8$$

The new policy will recommend "phone B" the $\frac{1}{0.8} = 1.25\times$ as often as the old policy.

criteo

Offline evaluation IPS



criteo

Imagine the user clicks

$$\pi_t(A_2 = \text{rice} | V_1 = \text{couscous}) = 1$$
$$\pi_0(A_2 = \text{rice} | V_1 = \text{couscous}) = 0.01$$

The new policy will recommend "rice" the $\frac{1}{0.01} = 100\times$ as often as the old policy.

criteo

Offline evaluation IPS

$$\text{CTR estimate} = \frac{1}{N}\sum_{n}^{N}\frac{c_n \pi_t(A_n = a_n | V_1 = v_1 .. V_n = v_n)}{\pi_0(A_n = a_n | V_1 = v_1 .. V_n = v_n)}$$

criteo

Offline evaluation IPS

$$\text{CTR estimate} = \frac{1}{N}\sum_{n}^{N}\frac{c_n \pi_t(A_n = a_n | V_1 = v_1 .. V_n = v_n)}{\pi_0(A_n = a_n | V_1 = v_1 .. V_n = v_n)}$$

We only look at the clicks i.e. $c_n = 1$ (otherwise the contribution is 0).

criteo

Offline evaluation IPS

$$\text{CTR estimate} = \frac{1}{N}\sum_{n}^{N}\frac{c_n \pi_t(A_n = a_n | V_1 = v_1 .. V_n = v_n)}{\pi_0(A_n = a_n | V_1 = v_1 .. V_n = v_n)}$$

We only look at the clicks i.e. $c_n = 1$ (otherwise the contribution is 0).

When the new policy differs markedly from the old, the weights become very high (to compensate for the fact that these are rare examples in your sample). These rare high values contribute a lot to the variance of the estimator.

Offline evaluation IPS

$$\text{CTR estimate} = \frac{1}{N}\sum_{n}^{N}\frac{c_n \pi_t(A_n = a_n | V_1 = v_1 .. V_n = v_n)}{\pi_0(A_n = a_n | V_1 = v_1 .. V_n = v_n)}$$

We only look at the clicks i.e. $c_n = 1$ (otherwise the contribution is 0).

When the new policy differs markedly from the old, the weights become very high (to compensate for the fact that these are rare examples in your sample). These rare high values contribute a lot to the variance of the estimator.

IPS actually atempts to predict the AB test, but it often has less than spectacular results...

Real world Reco

Real world reco is an anonymous hybrid of different methodologies.

Often poor agreement between online and offline metrics are reported..

criteo

Real world Reco

Real world reco is an anonymous hybrid of different methodologies.

Often poor agreement between online and offline metrics are reported..

why?

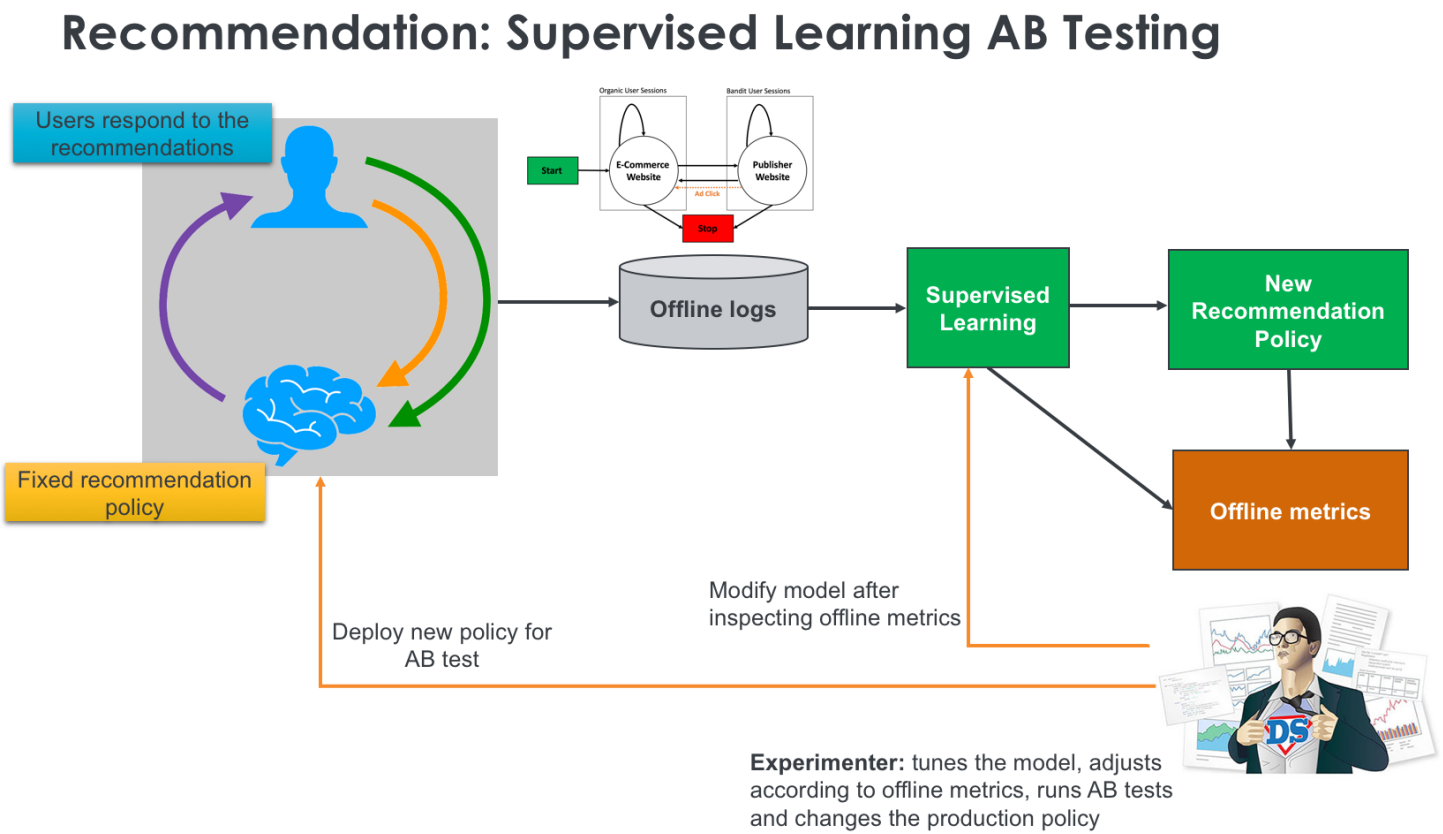
criteo

Real world Reco

Real world reco is an anonymous hybrid of different methodologies.
Often poor agreement between online and offline metrics are reported..
why? What does real world reco look like?

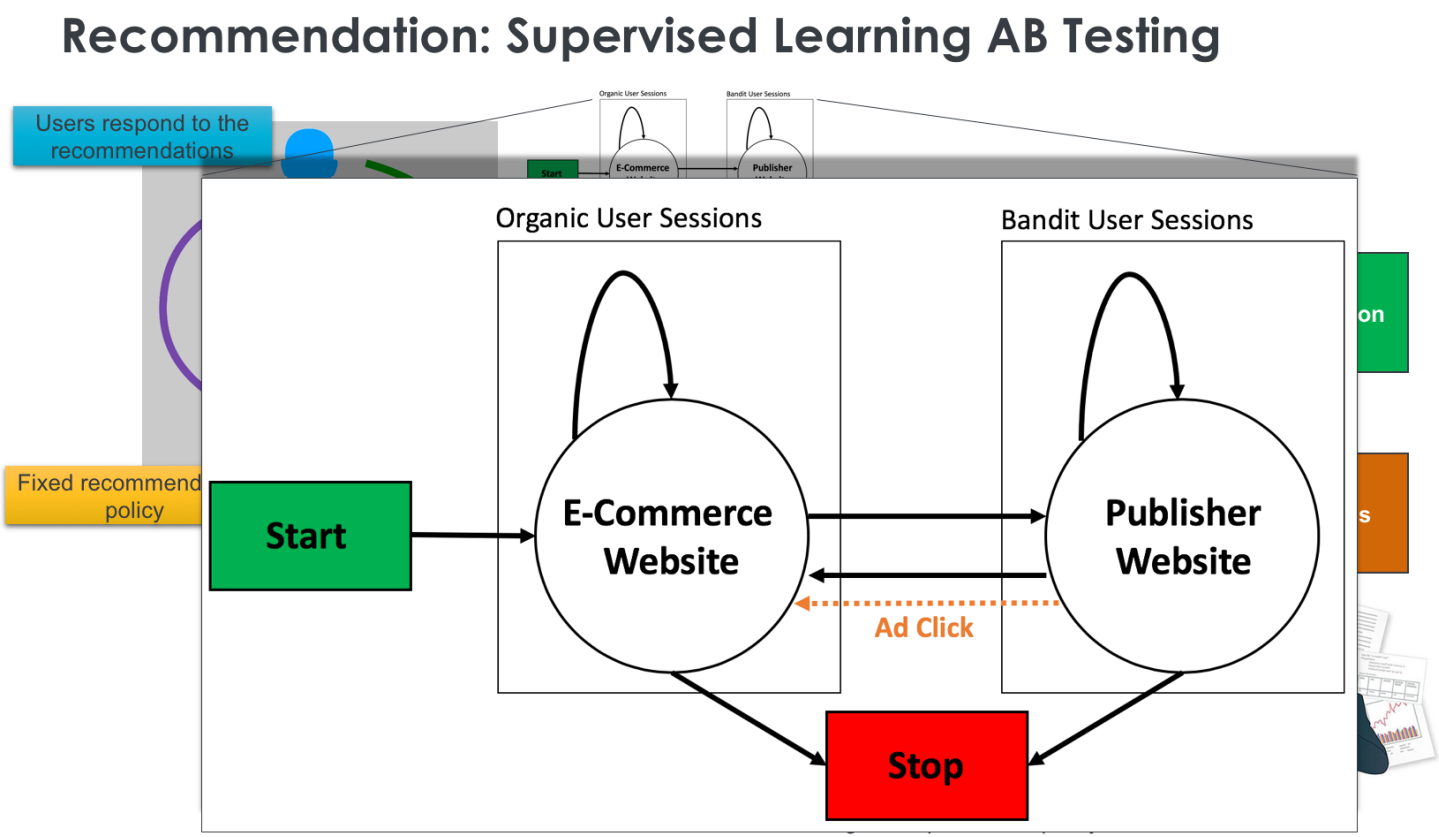
criteo

Recommendation as Supervised Learning and AB Testing



criteo

Recommendation as Supervised Learning and AB Testing



Let's take a step back...

How are we improving large-scale Recommender Systems in the Real World

- Learn a supervised model from past user activity
- Evaluate offline and decide whether to A/B test
- A/B test
- If positive and scalable, roll-out
- If not, try to understand what happened and try to create a better model of the world using the same past data
- Repeat

criteo

Supervised learning with the wrong objective

We are operating under the assumption that the best recommendation policy is in some sense the **optimal auto-complete of natural user behavior**

criteo

Supervised learning with the wrong objective

From the point of view of business metrics, **learning to autocomplete behavior is a great initial recommendation policy**, especially when no prior user feedback is available.

criteo

Supervised learning with the wrong objective

From the point of view of business metrics, **learning to autocomplete behavior is a great initial recommendation policy**, especially when no prior user feedback is available.

However, after a first golden age, where all offline improvements turn into positive A/B tests, the *naive recommendation* optimization objective and the business objective will start to diverge.

Aligning the Recommendation objective with the business

- Of course, we could start incorporating user feedback that we collected while running the initial recommendation policies
- We should be able to continue bringing improvements using feedback that is now aligned with our business metrics (ad CTR, post click sales, dwell time, number of videos watched, ...)

Modern Recommendation Systems Research

Q: How does the literature on Large Scale Recommendation Systems look right now?

A: Most of the latest publications are talking about ways of using Deep Learning for Recommendation:

- **Matrix Factorization extensions:** Word2Vec, Deep and Wide, Neural MF, Node2Vec
- **Content-based recommendation:** CNNs for Image, Text, Sounds to compute item similarities
- **Next event prediction / user activity modeling:** RNNs, TCNs

Modern Recommendation Systems Research

We see great improvements in offline metrics!

- **Regression/Classification metrics:** MSE, AUC, NLL
- **Ranking metrics:** MPR, Precision@k, NDCG

criteo

What is wrong with this picture?

In the same time, as practitioners, we see difficulties in improving the Real World Recommendation models!

Offline - online metrics alignment for recommendation is a recognized problem:

- **Previous work:** Missing Not At Random (MNAR) framework for Matrix factorization, Bandits Literature
- **New avenues:** *REVEAL: Offline evaluation for recommender systems Workshop at RecSys 2019* (this September in Copenhagen) `https://recsys.acm.org/recsys19/reveal/`

criteo

The relationship between Reinforcement Learning and Recommendation

- We are doing Reinforcement Learning by hand!

criteo

The relationship between Reinforcement Learning and Recommendation

- We are doing Reinforcement Learning by hand!
- Furthermore, we are trying to do RL using the Supervised Learning Framework

criteo

The relationship between Reinforcement Learning and Recommendation

- We are doing Reinforcement Learning by hand!
- Furthermore, we are trying to do RL using the Supervised Learning Framework
- Standard test data sets do not let us explore this aspect of recommender systems

criteo

The relationship between Reinforcement Learning and Recommendation

- We are doing Reinforcement Learning by hand!
- Furthermore, we are trying to do RL using the Supervised Learning Framework
- Standard test data sets do not let us explore this aspect of recommender systems

.. but how do you evaluate a reinforcement learning algorithm?

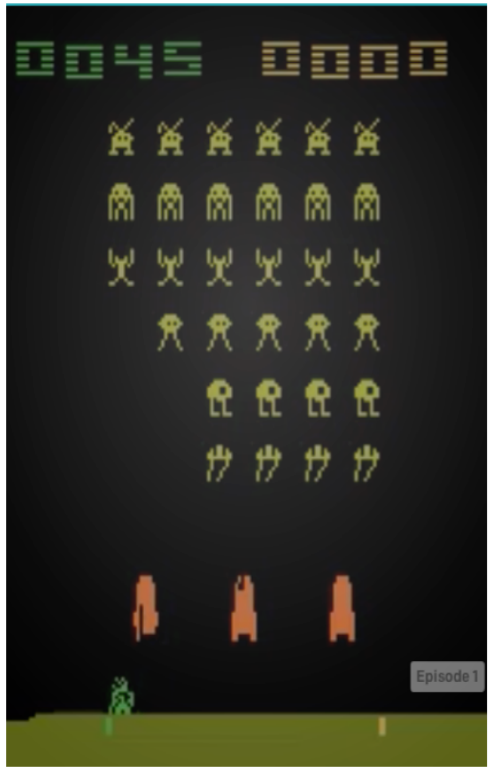
criteo

Open AI Gym

Problem: The reinforcement learning community lacked a common set of tools to allow comparison of reinforcement learning algorithms.

Open AI Gym: 2016 (Brockman et al.): Software standard (Python api) that allows comparison of the performance of reinforcement learning algorithms.

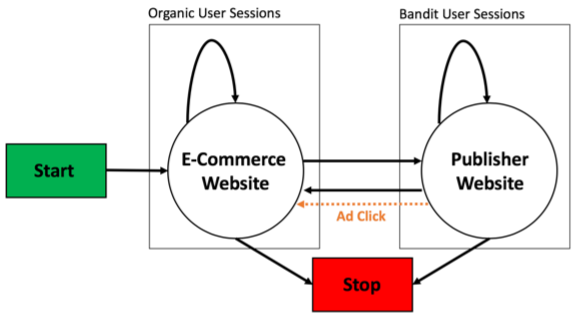
Core idea: environments (the problem) and agents (the RL algorithm) interact.



criteo

Introducing RecoGym

RecoGym: Simulates User Behavior (both Organic and Bandit)



- Allows online evaluation of new recommendation policies in a simulated environment
- Holistic view of user (organic and bandit) provides a framework for categorizing recommendation algorithms
- Key feature: adjustable parameter that changes the effectiveness of “Pure Organic” or next item prediction algorithms.

Algorithm 1: A simple Simulator

Input : $S \in \mathcal{R}^{3\times3}$ transition matrix between organic and bandit, $\Gamma \in \mathcal{R}^{P\times K}$ organic embeddings, μ_Γ organic popularity $\boldsymbol{\beta} \in \mathcal{R}^{P\times K}$ bandit embeddings, μ_β non-personalised ctr contribution $f(\cdot)$ montonic increasing function accounting for ad fatigue $\boldsymbol{m}$, U number of users, P number of products.

Output: Sequence of organic and bandit events

for $u \in 1..U$ **do**
  $t \leftarrow 0$
  $z_{u,0} \leftarrow$ organic
  $r_{u,0} \leftarrow$ undef
  $c_{u,0} \leftarrow$ undef
  $\omega_{u,0} \sim \mathcal{N}(0_{K\times1}, I_K)$
  $v_{u,0} \sim$ Categorical(softmax($\Gamma\omega_{u,0}$))
  while $z_{u,t} \neq$ *stop* **do**
    $t \leftarrow t+1$
    $\omega_{u,t} \sim \mathcal{N}(\omega_{u,t-1}, \sigma_\omega^2 I_K)$
    if $c_{u,t-1} \neq 1$ **then**
      $z_{u,t} \sim$ Categorical($S_{z_{u,t-1},\text{organic}}, S_{z_{u,t-1},\text{bandit}}, S_{z_{u,t-1},\text{stop}}$)
    else
      $z_{u,t} =$ organic
    end
    if $z_{u,t} =$ *organic* **then**
      $v_{u,t} \sim$ Categorical(softmax($\Gamma\omega_{u,t} + \mu_\Gamma$))
      $r_{u,t} \leftarrow$ undef
      $c_{u,t} \leftarrow$ undef
    end
    if $z_{u,t} =$ *bandit* **then**
      $r_{u,t}$ is geneated from the policy
      $c_{u,t} \sim$ Bernoulli($[f(\boldsymbol{\beta}\omega_{u,t} + \mu_\beta)]_{r_{u,t}}$)
    end
  end
end

criteo

f is an increasing function

$$f : \mathbb{R} \to (0, 1)$$

$$v_0 \sim \text{categorical}\left(\text{softmax}\left(\begin{pmatrix}\Gamma_{1,1} & \cdot & \Gamma_{1,K}\\ \cdot & \cdot & \cdot\\ \cdot & \cdot & \cdot\\ \Gamma_{P,1} & \cdot & \Gamma_{P,K}\end{pmatrix}\begin{pmatrix}\omega_{0,1}\\ \cdot\\ \omega_{0,K}\end{pmatrix} + \begin{pmatrix}\mu_1\\ \cdot\\ \cdot\\ \mu_P\end{pmatrix}\right)\right)$$

$$v_1 \sim \text{categorical}\left(\text{softmax}\left(\begin{pmatrix}\Gamma_{1,1} & \cdot & \Gamma_{1,K}\\ \cdot & \cdot & \cdot\\ \cdot & \cdot & \cdot\\ \Gamma_{P,1} & \cdot & \Gamma_{P,K}\end{pmatrix}\begin{pmatrix}\omega_{1,1}\\ \cdot\\ \omega_{1,K}\end{pmatrix} + \begin{pmatrix}\mu_1\\ \cdot\\ \cdot\\ \mu_P\end{pmatrix}\right)\right)$$

$$v_2 \sim \text{categorical}\left(\text{softmax}\left(\begin{pmatrix}\Gamma_{1,1} & \cdot & \Gamma_{1,K}\\ \cdot & \cdot & \cdot\\ \cdot & \cdot & \cdot\\ \Gamma_{P,1} & \cdot & \Gamma_{P,K}\end{pmatrix}\begin{pmatrix}\omega_{2,1}\\ \cdot\\ \omega_{2,K}\end{pmatrix} + \begin{pmatrix}\mu_1\\ \cdot\\ \cdot\\ \mu_P\end{pmatrix}\right)\right)$$

$$c_3|r_3 \sim \text{Bernoulli}\left(\left[f\left(\begin{pmatrix}\beta_{1,1} & \cdot & \beta_{1,K}\\ \cdot & \cdot & \cdot\\ \cdot & \cdot & \cdot\\ \beta_{P,1} & \cdot & \beta_{P,K}\end{pmatrix}\begin{pmatrix}\omega_{3,1}\\ \cdot\\ \omega_{3,K}\end{pmatrix} + \begin{pmatrix}\mu_1^*\\ \cdot\\ \cdot\\ \mu_P^*\end{pmatrix}\right)\right]_{r_3}\right)$$

$$c_4|r_4 \sim \text{Bernoulli}\left(\left[f\left(\begin{pmatrix}\beta_{1,1} & \cdot & \beta_{1,K}\\ \cdot & \cdot & \cdot\\ \cdot & \cdot & \cdot\\ \beta_{P,1} & \cdot & \beta_{P,K}\end{pmatrix}\begin{pmatrix}\omega_{4,1}\\ \cdot\\ \omega_{4,K}\end{pmatrix} + \begin{pmatrix}\mu_1^*\\ \cdot\\ \cdot\\ \mu_P^*\end{pmatrix}\right)\right]_{r_4}\right)$$

$$v_5 \sim \text{categorical}\left(\text{softmax}\left(\begin{pmatrix}\Gamma_{1,1} & \cdot & \Gamma_{1,K}\\ \cdot & \cdot & \cdot\\ \cdot & \cdot & \cdot\\ \Gamma_{P,1} & \cdot & \Gamma_{P,K}\end{pmatrix}\begin{pmatrix}\omega_{5,1}\\ \cdot\\ \omega_{5,K}\end{pmatrix} + \begin{pmatrix}\mu_1\\ \cdot\\ \cdot\\ \mu_P\end{pmatrix}\right)\right)$$

time

criteo

The RecoGym Session

In [1]:
```
import gym, reco_gym

# env_0_args is a dictionary of default parameters (i.e. number of products)
from reco_gym import env_1_args, Configuration

# You can overwrite environment arguments here:
env_1_args['random_seed'] = 41

# Initialize the gym for the first time by calling .make() and .init_gym()
env = gym.make('reco-gym-v1')
env.init_gym(env_1_args)


env.reset() # we call request to move to the first user
```

In [2]:
```
observation, reward, done, info = env.step(None)
# We specify None because we have need to learn about the user before we can act
```

In [3]:
```
observation.current_sessions
```

Out[3]:
```
[{'t': 0, 'u': 0, 'z': 'pageview', 'v': 4},
 {'t': 1, 'u': 0, 'z': 'pageview', 'v': 4},
 {'t': 2, 'u': 0, 'z': 'pageview', 'v': 4},
 {'t': 3, 'u': 0, 'z': 'pageview', 'v': 0}]
```

In [4]:
```
# We see the user is interested in product id 4 (they viewed it 3 times) and product id 0 (they viewed it once)
```

criteo

The RecoGym Session

```
In [5]: observation, reward, done, info = env.step(0)

In [6]: reward # the user does not click on our recommendation of product 0
Out[6]: 0

In [7]: observation.current_sessions # the user does not view any more products, so we learn nothing more about them
Out[7]: []
```

criteo

The RecoGym Session

```
In [8]: observation, reward, done, info = env.step(0) # we recommend product 0 again

In [9]: reward # they do not click again, ctr are usually low - this is not surprising
Out[9]: 0

In [10]: observation.current_sessions # again no additional product views the bandit session continues
Out[10]: []
```

criteo

The RecoGym Session

```
In [11]: observation, reward, done, info = env.step(4) # we now recommend product 4

In [12]: reward # they clicked!  We must have done something right on that last recommendation
Out[12]: 1

In [13]: observation.current_sessions # the user moved to the retailer website and viewed the product
Out[13]: [{'t': 7, 'u': 0, 'z': 'pageview', 'v': 4}]
```

criteo

Unlike RL we continue to use logs

Let's start with a random logging policy (a crazy thing to do, but a useful theoretical concept).

```
In [17]: env.generate_logs(100)
```

Out[17]:

	a	c	ps	ps-a	t	u	v	z
0	NaN	NaN	NaN	None	0	0	2.0	organic
1	NaN	NaN	NaN	None	1	0	4.0	organic
2	NaN	NaN	NaN	None	2	0	4.0	organic
3	NaN	NaN	NaN	None	3	0	4.0	organic
4	NaN	NaN	NaN	None	4	0	2.0	organic
5	NaN	NaN	NaN	None	5	0	4.0	organic
6	NaN	NaN	NaN	None	6	0	5.0	organic
7	NaN	NaN	NaN	None	7	0	4.0	organic
8	NaN	NaN	NaN	None	8	0	4.0	organic
9	NaN	NaN	NaN	None	9	0	4.0	organic
10	3.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	10	0	NaN	bandit
11	1.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	11	0	NaN	bandit
12	2.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	12	0	NaN	bandit
13	1.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	13	0	NaN	bandit
14	8.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	14	0	NaN	bandit
15	2.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	15	0	NaN	bandit
16	5.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	16	0	NaN	bandit
17	1.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	17	0	NaN	bandit
18	7.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	18	0	NaN	bandit
19	7.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	19	0	NaN	bandit
20	5.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	20	0	NaN	bandit

criteo

Organic Best Of vs Bandit Best Of

criteo

Exercise - Getting Started with RecoGym

Go to the notebook "Module III A.ipynb" Click Here

- Examine the logs, to start with we will look at non-personalized behavior only.
- Plot a histogram of organic product popularity. What is the most popular product?
- Plot the (non-personalized) click through rate of each product (with an error analysis). What is the most clicked-on product?
- Plot popularity against click through rate. How good a proxy is popularity for click through rate?
- Simulate an AB test comparing a simple recommender system (agent) that always recommends the highest ctr product with a recommender system that always recommends the organically most popular product.

criteo

Harder questions

Reflect on how the logging policy would affect these results.

criteo

Harder questions

Reflect on how the logging policy would affect these results.

What is personalization? What impact does personalization bring?

criteo

Harder questions

Reflect on how the logging policy would affect these results.

What is personalization? What impact does personalization bring?

Where does traditional academic recommender systems research fit into this picture?

criteo

Evaluate an organic agent using the Bandit Signal

criteo

The cold start scenario

- You need to build a recommender system for a client, they have an existing system with existing organic traffic.

criteo

The cold start scenario

- You need to build a recommender system for a client, they have an existing system with existing organic traffic. This means you can build standard next item predictions to build models of user behavior and understand how user's preferences cluster together.

criteo

The cold start scenario

- You need to build a recommender system for a client, they have an existing system with existing organic traffic. This means you can build standard next item predictions to build models of user behavior and understand how user's preferences cluster together.
- They are currently running a recommender system with a random policy.

criteo

The cold start scenario

- You need to build a recommender system for a client, they have an existing system with existing organic traffic. This means you can build standard next item predictions to build models of user behavior and understand how user's preferences cluster together.
- They are currently running a recommender system with a random policy.
- Your task is to build a next item prediction model as a proxy for a recommendation algorithm. Then produce offline metrics to convince the engineering team to run an AB test.

The cold start scenario

- You need to build a recommender system for a client, they have an existing system with existing organic traffic. This means you can build standard next item predictions to build models of user behavior and understand how user's preferences cluster together.
- They are currently running a recommender system with a random policy.
- Your task is to build a next item prediction model as a proxy for a recommendation algorithm. Then produce offline metrics to convince the engineering team to run an AB test.
- Finally you evaluate your performance against production.

The cold start scenario

- You need to build a recommender system for a client, they have an existing system with existing organic traffic. This means you can build standard next item predictions to build models of user behavior and understand how user's preferences cluster together.
- They are currently running a recommender system with a random policy.
- Your task is to build a next item prediction model as a proxy for a recommendation algorithm. Then produce offline metrics to convince the engineering team to run an AB test.
- Finally you evaluate your performance against production.

Please look at notebook "Module III B.ipynb" Click here

criteo

Likelihood Based Agents

criteo

Likelihood Based Agent

$$c_n \sim \text{Bernoulli}\left(\sigma\left(\Phi([\boldsymbol{X}_n\ \boldsymbol{a}_n])^T\beta\right)\right)$$

where

- β are the parameters;
- $\sigma(\cdot)$ is the logistic sigmoid;
- $\Phi(\cdot)$ is a function that maps $\boldsymbol{X}, \boldsymbol{a}$ to a higher dimensional space and includes some interaction terms between $\boldsymbol{X}_n$ and $\boldsymbol{a}_n$.

Likelihood Based Agent

$$c_n \sim \text{Bernoulli}\left(\sigma\left(\Phi([\boldsymbol{X}_n\ \boldsymbol{a}_n])^T \beta\right)\right)$$

where

- β are the parameters;
- $\sigma(\cdot)$ is the logistic sigmoid;
- $\Phi(\cdot)$ is a function that maps $\boldsymbol{X}, \boldsymbol{a}$ to a higher dimensional space and includes some interaction terms between $\boldsymbol{X}_n$ and $\boldsymbol{a}_n$. *Why?*

criteo

Likelihood Based Agent

Or using $\Phi([\boldsymbol{X}_n \ \boldsymbol{a}_n]) = \boldsymbol{X}_n \otimes \boldsymbol{a}_n$:

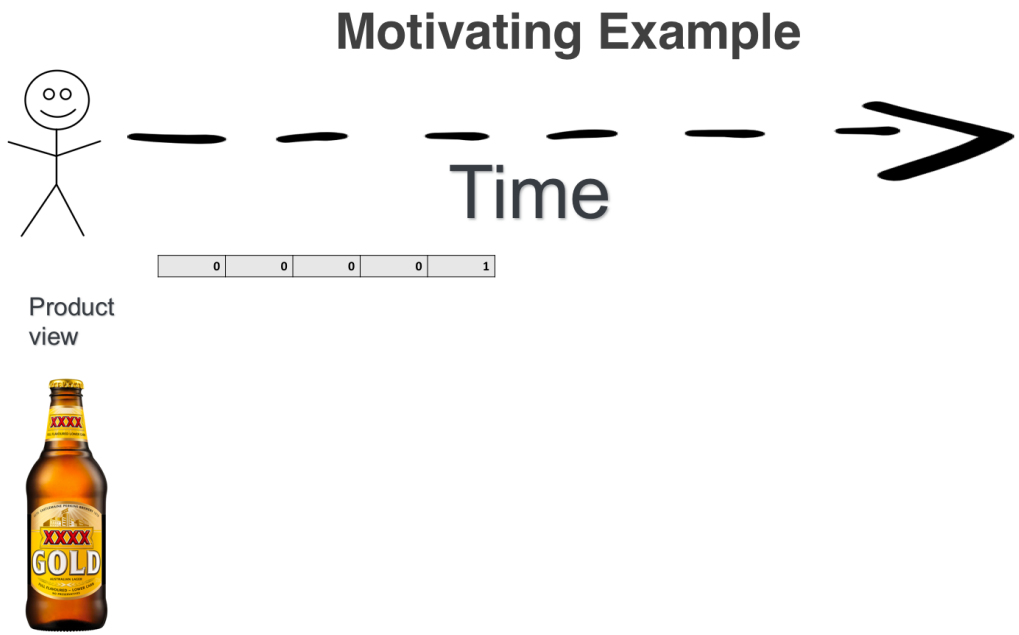
$$\hat{\beta}_{\text{lh}} = \text{argmax}_{\beta} \sum_n c_n \log \sigma \left((\boldsymbol{X}_n \otimes \boldsymbol{a}_n)^T \beta \right) + (1 - c_n) \log \left(1 - \sigma \left((\boldsymbol{X}_n \otimes \boldsymbol{a}_n)^T \beta \right) \right)$$

criteo

Feature Engineering to get the context vector $\boldsymbol{X}$

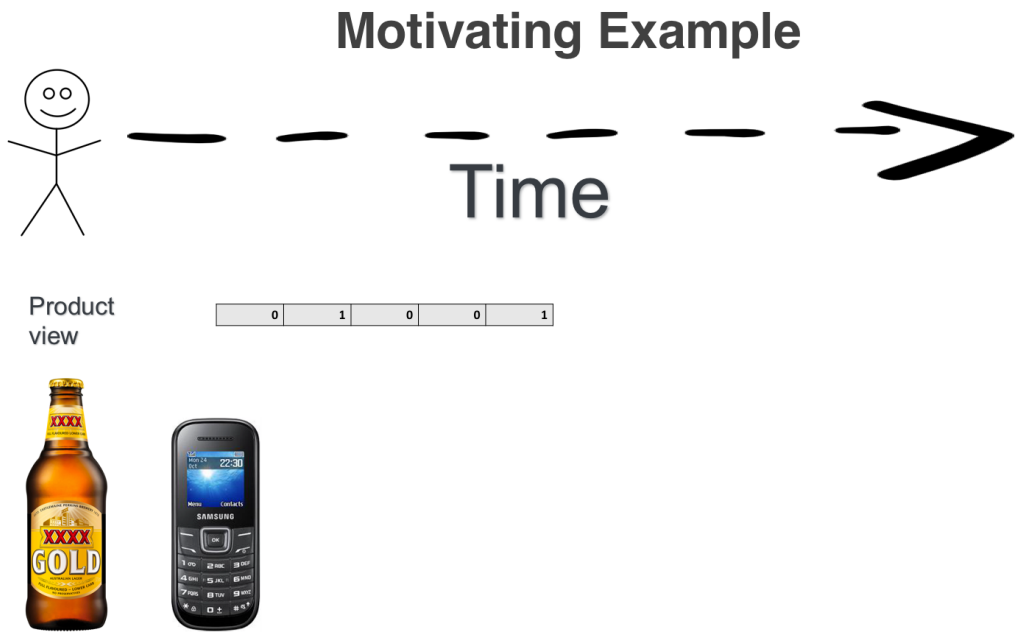
The above model requires us to specify the context $\boldsymbol{X}$, how can we do this?

Feature Engineering



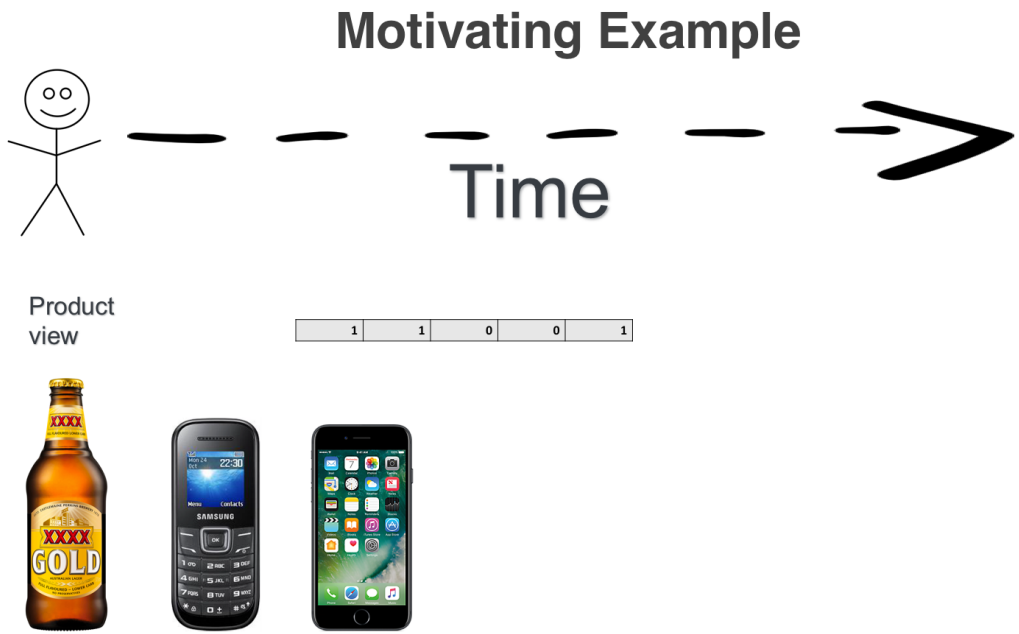
criteo

Feature Engineering

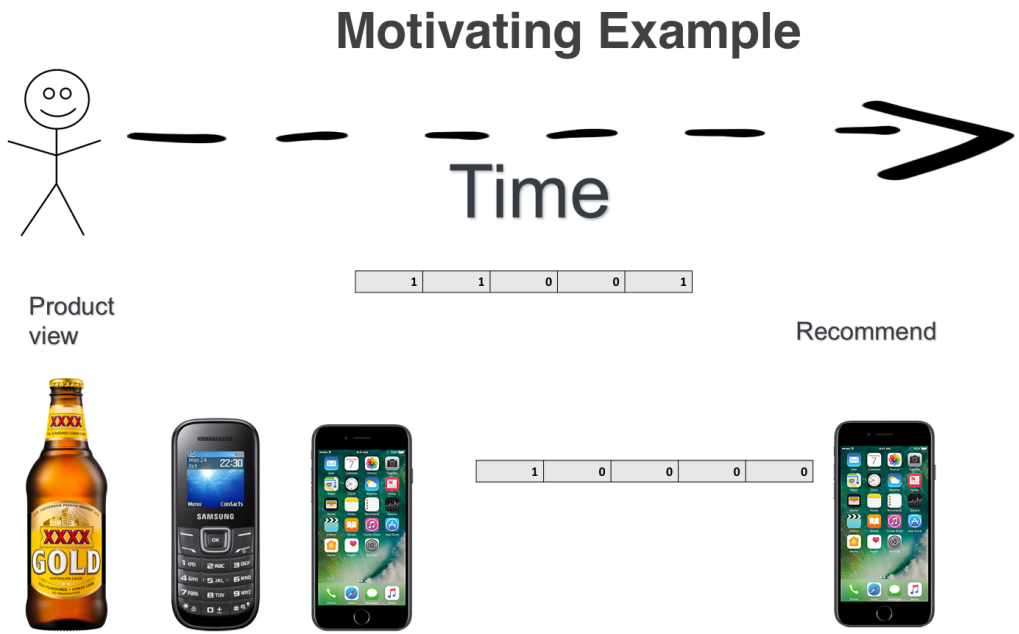


criteo

Feature Engineering



Feature Engineering



criteo

Feature Engineering

	a	c	ps	ps-a	t	u	v	z
0	NaN	NaN	NaN	None	0	0	0.0	organic
1	3.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	1	0	NaN	bandit
2	4.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	2	0	NaN	bandit
3	5.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	3	0	NaN	bandit
4	NaN	NaN	NaN	None	0	1	1.0	organic
5	2.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	1	1	NaN	bandit
6	8.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	2	1	NaN	bandit
7	4.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	3	1	NaN	bandit
8	NaN	NaN	NaN	None	4	1	6.0	organic
9	NaN	NaN	NaN	None	5	1	6.0	organic
10	NaN	NaN	NaN	None	6	1	4.0	organic
11	NaN	NaN	NaN	None	7	1	6.0	organic
12	NaN	NaN	NaN	None	8	1	2.0	organic
13	NaN	NaN	NaN	None	9	1	6.0	organic
14	NaN	NaN	NaN	None	10	1	2.0	organic
15	NaN	NaN	NaN	None	11	1	1.0	organic
16	NaN	NaN	NaN	None	12	1	6.0	organic
17	NaN	NaN	NaN	None	13	1	6.0	organic
18	NaN	NaN	NaN	None	14	1	4.0	organic
19	NaN	NaN	NaN	None	15	1	1.0	organic
20	NaN	NaN	NaN	None	16	1	1.0	organic
21	NaN	NaN	NaN	None	17	1	1.0	organic
22	NaN	NaN	NaN	None	18	1	6.0	organic
23	NaN	NaN	NaN	None	19	1	6.0	organic
24	NaN	NaN	NaN	None	20	1	1.0	organic
25	NaN	NaN	NaN	None	21	1	6.0	organic
26	3.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	22	1	NaN	bandit

```
history[0:8]
```

```
[array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([0., 1., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([0., 1., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([0., 1., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([0., 6., 2., 0., 2., 0., 9., 0., 0., 0.]),
 array([0., 6., 2., 0., 2., 0., 9., 0., 0., 0.])]
```

```
actions[0:8]
```

```
[array([0, 0, 0, 1, 0, 0, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 0, 0, 1, 0, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 0, 0, 0, 1, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 1, 0, 0, 0, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 0, 0, 0, 0, 0, 0, 1, 0], dtype=int8),
 array([0, 0, 0, 0, 1, 0, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 0, 1, 0, 0, 0, 0, 0, 0], dtype=int8),
 array([1, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int8)]
```

criteo

Feature Engineering

	a	c	ps	ps-a	t	u	v	z
0	NaN	NaN	NaN	None	0	0	0.0	organic
1	3.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	1	0	NaN	bandit
2	4.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	2	0	NaN	bandit
3	5.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	3	0	NaN	bandit
4	NaN	NaN	NaN	None	0	1	1.0	organic
5	2.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	1	1	NaN	bandit
6	8.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	2	1	NaN	bandit
7	4.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	3	1	NaN	bandit
8	NaN	NaN	NaN	None	4	1	6.0	organic
9	NaN	NaN	NaN	None	5	1	6.0	organic
10	NaN	NaN	NaN	None	6	1	4.0	organic
11	NaN	NaN	NaN	None	7	1	6.0	organic
12	NaN	NaN	NaN	None	8	1	2.0	organic
13	NaN	NaN	NaN	None	9	1	6.0	organic
14	NaN	NaN	NaN	None	10	1	2.0	organic
15	NaN	NaN	NaN	None	11	1	1.0	organic
16	NaN	NaN	NaN	None	12	1	6.0	organic
17	NaN	NaN	NaN	None	13	1	6.0	organic
18	NaN	NaN	NaN	None	14	1	4.0	organic
19	NaN	NaN	NaN	None	15	1	1.0	organic
20	NaN	NaN	NaN	None	16	1	1.0	organic
21	NaN	NaN	NaN	None	17	1	1.0	organic
22	NaN	NaN	NaN	None	18	1	6.0	organic
23	NaN	NaN	NaN	None	19	1	6.0	organic
24	NaN	NaN	NaN	None	20	1	1.0	organic
25	NaN	NaN	NaN	None	21	1	6.0	organic
26	3.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	22	1	NaN	bandit

```
history[0:8]
```

```
[array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([0., 1., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([0., 1., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([0., 1., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([0., 6., 2., 0., 2., 0., 9., 0., 0., 0.]),
 array([0., 6., 2., 0., 2., 0., 9., 0., 0., 0.])]
```

```
actions[0:8]
```

```
[array([0, 0, 0, 1, 0, 0, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 0, 0, 1, 0, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 0, 0, 0, 1, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 1, 0, 0, 0, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 0, 0, 0, 0, 0, 0, 1, 0], dtype=int8),
 array([0, 0, 0, 0, 1, 0, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 0, 1, 0, 0, 0, 0, 0, 0], dtype=int8),
 array([1, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int8)]
```

How would you interact the features?

criteo

Feature Engineering

	a	c	ps	ps-a	t	u	v	z
0	NaN	NaN	NaN	None	0	0	0.0	organic
1	3.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	1	0	NaN	bandit
2	4.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	2	0	NaN	bandit
3	5.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	3	0	NaN	bandit
4	NaN	NaN	NaN	None	0	1	1.0	organic
5	2.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	1	1	NaN	bandit
6	8.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	2	1	NaN	bandit
7	4.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	3	1	NaN	bandit
8	NaN	NaN	NaN	None	4	1	6.0	organic
9	NaN	NaN	NaN	None	5	1	6.0	organic
10	NaN	NaN	NaN	None	6	1	4.0	organic
11	NaN	NaN	NaN	None	7	1	6.0	organic
12	NaN	NaN	NaN	None	8	1	2.0	organic
13	NaN	NaN	NaN	None	9	1	6.0	organic
14	NaN	NaN	NaN	None	10	1	2.0	organic
15	NaN	NaN	NaN	None	11	1	1.0	organic
16	NaN	NaN	NaN	None	12	1	6.0	organic
17	NaN	NaN	NaN	None	13	1	6.0	organic
18	NaN	NaN	NaN	None	14	1	4.0	organic
19	NaN	NaN	NaN	None	15	1	1.0	organic
20	NaN	NaN	NaN	None	16	1	1.0	organic
21	NaN	NaN	NaN	None	17	1	1.0	organic
22	NaN	NaN	NaN	None	18	1	6.0	organic
23	NaN	NaN	NaN	None	19	1	6.0	organic
24	NaN	NaN	NaN	None	20	1	1.0	organic
25	NaN	NaN	NaN	None	21	1	6.0	organic
26	3.0	0.0	0.1	[0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, ...	22	1	NaN	bandit

```
history[0:8]
```

```
[array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([0., 1., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([0., 1., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([0., 1., 0., 0., 0., 0., 0., 0., 0., 0.]),
 array([0., 6., 2., 0., 2., 0., 9., 0., 0., 0.]),
 array([0., 6., 2., 0., 2., 0., 9., 0., 0., 0.])]
```

```
actions[0:8]
```

```
[array([0, 0, 0, 1, 0, 0, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 0, 0, 1, 0, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 0, 0, 0, 1, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 1, 0, 0, 0, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 0, 0, 0, 0, 0, 0, 1, 0], dtype=int8),
 array([0, 0, 0, 0, 1, 0, 0, 0, 0, 0], dtype=int8),
 array([0, 0, 0, 1, 0, 0, 0, 0, 0, 0], dtype=int8),
 array([1, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int8)]
```

How would you interact the features?

Is this a good feature engineering scheme? Can you improve it?

criteo

Pure Bandit Feedback

- Build a model using feature engineering using logistic regression (or a deep model)

criteo

Pure Bandit Feedback

- Build a model using feature engineering using logistic regression (or a deep model)
- Finally you evaluate your performance against production.

criteo

Pure Bandit Feedback

- Build a model using feature engineering using logistic regression (or a deep model)
- Finally you evaluate your performance against production.

Please look at notebook “Module III C.ipynb” Click here

criteo

Pure Bandit Feedback

- Build a model using feature engineering using logistic regression (or a deep model)
- Finally you evaluate your performance against production.

Please look at notebook “Module III C.ipynb” Click here

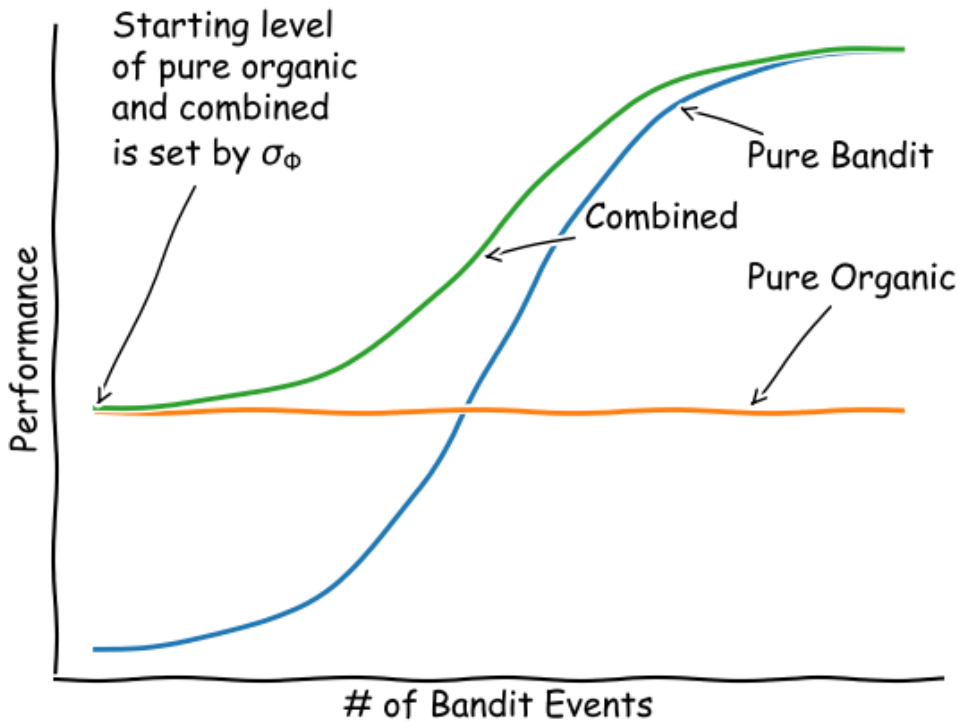
What is lacking in this approach?

criteo

Combining Organic Signal with Bandit Signal

criteo

Pure Organic vs Pure Bandit



criteo

Can we get the best of both worlds?

A simple algorithm:

Can we get the best of both worlds?

A simple algorithm:

Use one of the methods from the first part of the course on the organic data to prouduce product embeddings. Classic methods are SVD or word2vec.

Can we get the best of both worlds?

A simple algorithm:

Use one of the methods from the first part of the course on the organic data to prouduce product embeddings. Classic methods are SVD or word2vec.

A user's history is then a list of embeddings rather than non-descript identifiers.

Can we get the best of both worlds?

A simple algorithm:

Use one of the methods from the first part of the course on the organic data to prouduce product embeddings. Classic methods are SVD or word2vec.

A user's history is then a list of embeddings rather than non-descript identifiers.

The action space is also in the same embedding space.

Can we get the best of both worlds?

A simple algorithm:

Use one of the methods from the first part of the course on the organic data to prouduce product embeddings. Classic methods are SVD or word2vec.

A user's history is then a list of embeddings rather than non-descript identifiers.

The action space is also in the same embedding space.

Exercise: implement this approach, use module III B and C as a reference.

Can we get the best of both worlds?

A simple algorithm:

Use one of the methods from the first part of the course on the organic data to prouduce product embeddings. Classic methods are SVD or word2vec.

A user's history is then a list of embeddings rather than non-descript identifiers.

The action space is also in the same embedding space.

Exercise: implement this approach, use module III B and C as a reference.

What advantages or challenges do you have in this approach?

Thank You!

criteo