Toloka

Toloka

RecSys 2022 Tutorial

Improving Recommender Systems with Human-in-the-Loop

Dmitry Ustalov, Natalia Fedorova, Nikita Pavlichenko, Maxim Kunakov, Fedor Zhdanov

Toloka

Part V
From Human Labels to Ground Truth

Fedor Zhdanov,
Head of Machine Learning

Tutorial Schedule

Part I Intro: 10 min
Introduction

Part III: 20 min
Human-in-the-Loop Essentials

Part III: 20 min
Human-in-the-Loop Essentials

Part IV: 50 min
Hands-On Practice Session

Coffee Break :
10 min

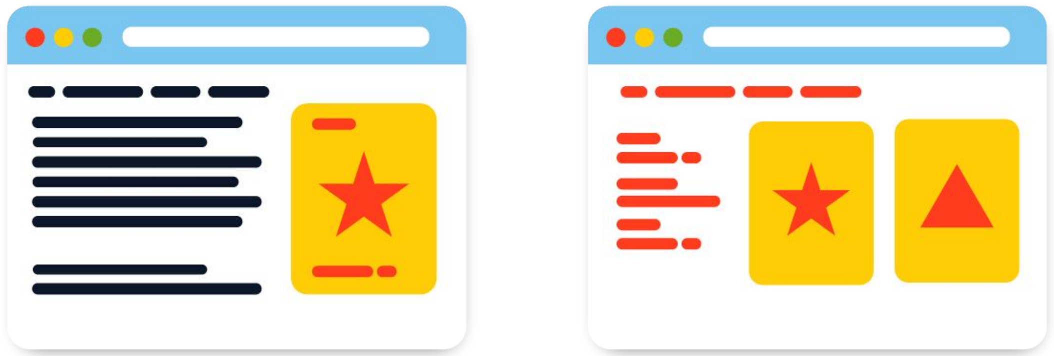
Part V: 30 min
From Human Labels to Ground Truth

Part VI: 10 min
Conclusion

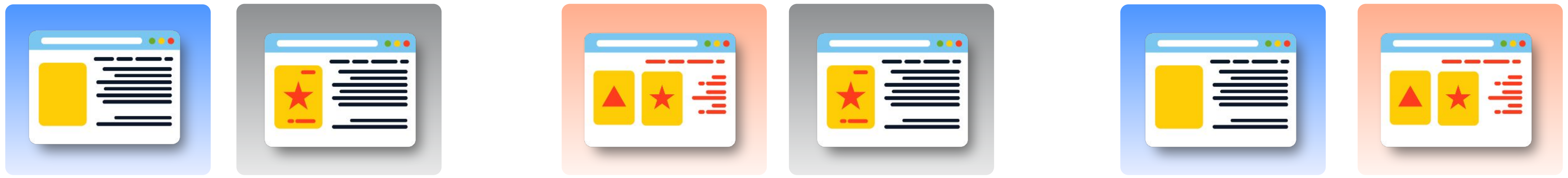
Answer Aggregation

Difference from Multiclassification

- The latent label assumption is not satisfied when comparing complex items



- Different tasks may contain common items



Formalization

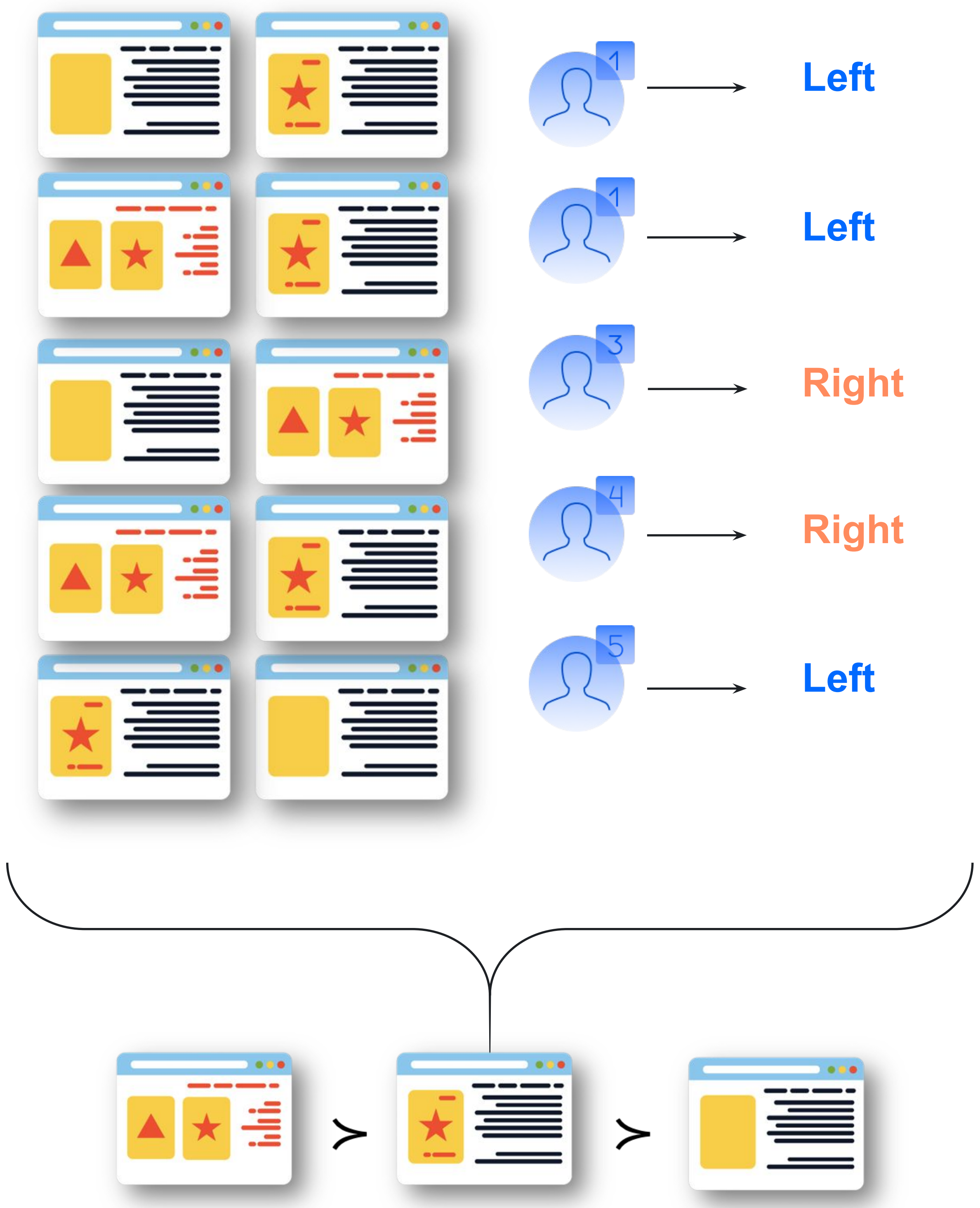
- Answers: Left or Right

- Items $d_j \in \{1, \ldots, N\}$

- Given pairwise comparisons for items in D:

$$P = \{(w_k, d_i, d_j): i \succ_k j\}$$

- Obtain **a ranking** π over items $D \rightarrow \{1, \ldots, N\}$ based on answers in P



Bradley and Terry Model (BT)

- Assume that each item $d_i \in D$ has a latent "quality" score $s_i \in \mathbb{R}$

- The probability that $d_i \in D$ will be preferred in a comparison over $d_j \in D$

$$\Pr(i \succ j) = f\left(s_i - s_j\right),$$

where $f(x) = {}^{1}/_{1+e^{-x}}$.

The model assumes that all performers are equally good and truthful!

NoisyBT

w_k  $\longleftrightarrow$ "reliability" γ_k and "bias" q_k

The likelihood of $i \succ_k j$ is

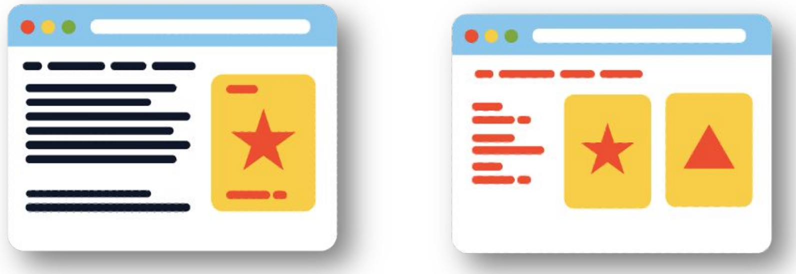
$$\Pr(i \succ_k j) = \underbrace{f(\gamma_k) f\left(s_i - s_j\right)}_{\text{Truthful answer}} + \underbrace{\left(1 - f(\gamma_k)\right) f\left((-1)^{(1 - \mathbb{I}(d_i \text{ was left}))} q_k\right)}_{\text{Random answer}},$$

where $\mathbb{I}(d_i \text{ was left})$ is the indicator for the order of d_i and d_j

d_i d_j

$\mathbb{I}(d_i \text{ was left}) = 1$

d_j d_i



$\mathbb{I}(d_i \text{ was left}) = 0$

NoisyBT: Example

Performer	Task	Left	Right
w_1	t_1	a	b
w_1	t_2	b	c
w_1	t_3	c	a
w_2	t_1	a	b
w_2	t_2	b	c
w_2	t_3	c	a

Item	Score
a	1.000
b	0.547
c	0.000

Performer	Bias	Skill
w_1	0.633	0.656
w_2	1.000	0.000

The model can be estimated using gradient descent.

Crowd-Kit

- **Crowd-Kit** is a general-purpose Python library for answer aggregation in crowdsourcing

- **Algorithms:** Majority Vote, Dawid-Skene, Bradley-Terry, NoisyBT, RASA, HRRASA, etc.

- **PyPI:** https://pypi.org/projects/crowd-kit/

NoisyBT with Crowd-Kit: Input

```python
import pandas as pd  # pip install pandas

from crowdkit.aggregation import NoisyBradleyTerry  # pip install -U crowd-kit

# In this example we will use the annotation results in the Toloka TSV format,
# but Crowd-Kit is platform-agnostic and it can handle any other format

df = pd.read_csv('annotation.tsv', sep='\t', dtype=str)

# Filter golden tasks and comparisons where both items were irrelevant

df = df[df['GOLDEN:result'].isna()]

df = df[df['OUTPUT:result' != '404']]
```

NoisyBT with Crowd-Kit: Processing

```
df['OUTPUT:result'] = df.apply(

   lambda row: row['INPUT:recommendation_a'] if row['OUTPUT:result'] = 'recommendation_a' else
row['INPUT:recommendation_b'], axis=1)

)

# We need the output column values to be the IDs of items that won a comparison

df = df.rename(columns={'INPUT:recommendation_a': 'left', 'INPUT:recommendation_a': 'right',
'OUTPUT:result': 'label', 'ASSIGNMENT:worker_id': 'worker'})

# We need left, right, label, and worker columns to run aggregations from Crowd-Kit
```

NoisyBT with Crowd-Kit: Aggregation

```
rankings = {}

for product, product_annotation in df.groupby('INPUT:initial_product'):

   rankings[product] = NoisyBradleyTerry(100).fit_predict(product_annotation).sort_values(

      ascending=False)

# For each product, we run a separate aggregation
```

Join our Slack:
recsys_2022

Fedor Zhdanov

Head of Machine Learning at Toloka

fedorzh@toloka.ai

https://toloka.ai/events/recsys-2022/

https://bit.ly/3eYIX2P