Towards closed-loop crowdsourcing and human computation

JORDAN W. SUCHOW, VISHAL H. LALL, FREDERICK CALLAWAY, M PACER, & THOMAS L. GRIFFITHS

Department of Psychology, University of California, Berkeley

Berkeley
UNIVERSITY OF CALIFORNIA

Crowdsourcing and human computation integrate cognition into computationally mediated workflows. Here, we introduce Judicious, a suite of human-in-the-loop elementary random procedures that can be inserted into probabilistic programs to run and model closed-loop crowdsourcing and human-computation experiments. Each procedure abstracts the procurement of a human decision into a single blocking function call, thereby enabling flexible and idiomatic commingling of inference and behavioral data collection. We discuss the architecture of our Python-based implementation of Judicious and provide examples of tasks and algorithmic workflows.

ALGORITHMS

Image labeling

```
import judicious
judicious.seed(9145822646)
url = "http://placekitten.com/200/300"
label = judicious.label(url)
print(label)
```

Transmission chain

```
text = "One night two young men from Egulac..."

for _ in range(10):
    text = judicious.reproduce(text)
    print(text)
```

Bayesian bandits for A/B testing jokes

```
import thompson_sampling

joke1 = judicious.joke()
joke2 = judicious.joke()

for _ in range(10):
    joke = thompson_sampling.pull()
    is_funny= judicious.agreement(“Is funny”, joke) > 4
    thompson_sampling.reward(joke, is_funny)
```

Simultaneous perturbation stochastic approximation

```
import numpy as np
import judicious

z2 = np.zeros(shape=(1, 16)).astype(‘float32')

for j in range(20):

    noise = NOISE_LEVEL * np.random.normal(size=(N, 16))
    z2s = z2 + noise

    ranks = judicious.rank_the(
        category="Alan Turing",
        images=z2s,
    )
    w = [WEIGHTS[ranks.index(i)] for i in range(N)]
    z2 = z2 + (ALPHA**j) * np.dot(noise.T, w)/(N*NOISE_LEVEL)
```

TASKS

—Labeling an image
—Solving a reCAPTCHA
—Intertemporal choice problems
—Drawing an image based on a caption
—Playing any of 100s of Atari 2600 games
—Defining a word
—Playing a move of a chess game
—Rating agreement with a statement
— Judging the age of a person in a photograph
— Perceptually comparing two numerosities
— Copyediting text
— Identifying a letter presented in visual noise
— Answering a question about an iFramed site
—Ranking images along some property
— Redacting text for illicit material
— Comparing the similarity of two words to a target
—Reproducing text verbatim from memory
— Summarizing text
—Solving the trolley problem
— Telling a joke
— Building a tower using blocks

Integrating human-decision-based ERPs directly into probabilistic programs enables closed-loop modeling and running of crowdsourcing and human-computation experiments; this could be used, for example, for active experiment design.

Crowdsourcing and human computation integrate cognition into computationally mediated workflows. Here, we introduce Judicious, a suite of human-in-the-loop elementary random procedures that can be inserted into probabilistic programs to run and model closed-loop crowdsourcing and human-computation experiments. Each procedure abstracts the procurement of a human decision into a single blocking function call, thereby enabling flexible and idiomatic commingling of inference and behavioral data collection. We discuss the architecture of our Python-based implementation of Judicious and provide examples of tasks and algorithmic workflows.

ALGORITHMS

TASKS