Microtask: multiple-choice question

- Example: relevance assessments for web search
  - Given a query and a list of categories; mark the one(s) that are appropriate.
  - Given two lists of search results for the same query, mark which one is better, or if they are the same.
- For simplicity, assume exactly one correct answer
Let’s use crowdsourcing!

- Crowdsourced “workers” vs. “experts”
  - workers are much cheaper but less reliable
  - assume they are biased towards the correct answer
- Ask workers one by one, then stop and output an answer
- Goal: correct answer with small error prob & total cost

Multiple pools of workers available, the platform can (adaptively) choose among them
Multiple “crowds”

- Multiple pools of workers (“crowds”)
  - different third-party providers; segmentation within the same provider
  - can differ in quality and per-worker costs; a crowd can be better at one microtask, worse at another

- Within same crowd $i$, workers are indistinguishable (random)

A worker from crowd $i$ gives “random” answer: i.i.d. sample from crowd-specific distribution, with (crowd-specific) bias towards the correct answer
The “Bandit Survey Problem” (n=2)

- Microtask: $n = 2$ answers: correct and incorrect
- crowds $i = 1..k$: (known) per-worker cost $c_i$, and latent bias $\epsilon_i$ towards the correct answer
- In each round $t$, algorithm picks a crowd $i = i_t$ (at cost $c_i$)
  - a “random worker” from this crowd gives an answer $x_t$

\[ x_t \text{ is correct independently with probability } \frac{1}{2} + \epsilon_i \]

- Eventually, algorithm stops and outputs one answer
  Performance measures: total cost and error rate.

Alternative interpretation: a single “crowd”, two ways of asking the same question (e.g., with or without a picture)
“Bandit surveys” vs. *multi-armed bandits*

- in each round, algorithm chooses from a fixed set of *arms* (crowds), gets partial feedback: *only* for the chosen arm
- ... but the feedback is very different
  - *bandits*: algorithm collects (a known amount of) reward, the goal is to maximize total reward
  - *bandit surveys*: algorithm collects info, the value of this info is not explicitly given, the goal is to guess the correct answer in small #rounds

A problem on “explore-exploit tradeoff”, but *not* a multi-armed bandit problem
Bandit survey algorithms

- **Bandit survey algorithm**: which crowd to choose
- **stopping rule**: when to stop, which answer to output

Caveat: there is no “best” stopping rule!

- How to compare two crowd selection algorithms?
  - idea: use them with the same stopping rule,
  - ideally, want the comparison to be consistent across all (reasonable) stopping rules
Our contributions

- Focus on selection between crowds
- Algorithms that compete with the *best fixed crowd*
  new algorithms, theory and experiments

*New twist*: with >2 options, a fixed *distribution over crowds* may outperform the “best fixed crowd”
- Our algorithm probably improves over “best fixed crowd”

*Final note*: Adaptive task assignment / pricing in crowdsourcing systems is a rich research direction