Dynamic Structural Econometrics 2025
The University of Hong Kong

11 December, 2025

Learning-by-Doing: New Tasks Are Interesting, Familiar Tasks Are Easier–
Exploring the Dynamic Behavior of Workers on the Crowdsourcing Platform

Da Eun, Han

Outline

appendix

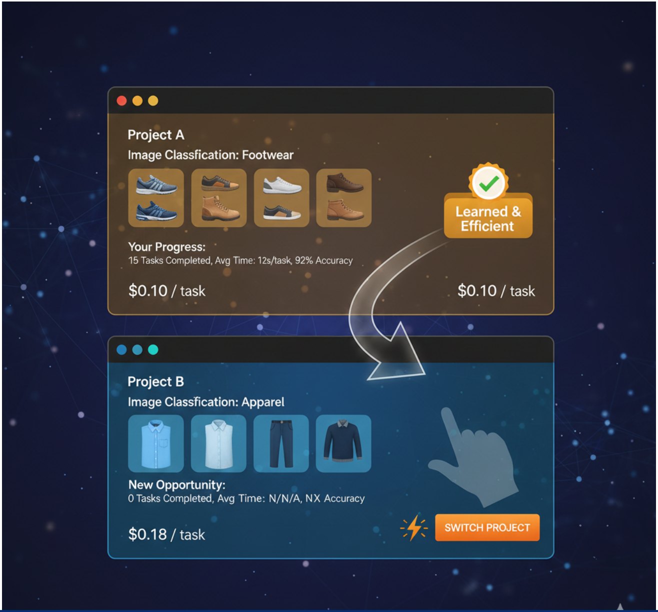
– Motivation –

Motivation

A Worker's Dynamic Choice on a Crowdsourcing Platform

- A worker completes several tasks on Project A and becomes faster over time.
- Suddenly, a new Project B appears with a higher price.
- Should she stay (benefiting from learning) or switch (chasing better pay)?

Motivation
A Worker's Dynamic Choice



Motivation

Economic Trade-off: Staying vs. Switching

- Staying
  - $+$ Learning-by-doing increases productivity
  - $+$ Sunk training cost already paid
  - \- Lower price / unpleasant tasks / boredom
- Switching
  - $+$ Higher prices or better project features
  - \- Loss of accumulated experience
  - \- New training $\rightarrow$ higher short-term costs
- Key implication: Retention is not trivial. It reflects forward-looking optimization.

Motivation

Why Retention & Why Dynamic?

- **Retention matters:**
  - Experienced workers produce faster output.
  - Requesters benefit from consistent labeled data.
  - Platforms avoid re-training new workers and maintain stable quality.
- **Retention is a dynamic choice:**
  - Learning-by-doing lowers future costs.
  - Switching resets experience $\rightarrow$ short-run costs.
  - Task attributes shape forward-looking incentives.
- **Implication:** A dynamic structural model is required to quantify learning, experience, and switching costs.

- Workers often open and consider multiple projects within a session.
- The median session includes 2 projects, but each project receives only 1 task.
- $\Rightarrow$ Workers sample projects shallowly and switch very early.
- This early-switching pattern motivates a dynamic structural model to quantify experience and switching costs.

– Overview –

Overview

Research question

- What factors influence platform workers' incentive to continue the same project on a crowdsourcing platform?
    - How does learning-by-doing affect labor supply costs?

Overview

Research question

- What factors influence platform workers' incentive to continue the same project on a crowdsourcing platform?
  - How does learning-by-doing affect labor supply costs?

Framework

- Reduced-form analysis with probit models to study short-run effects
- Dynamic structural model to study long-run effects and run counterfactuals

– Literature Review –

Literature Review

- Dynamic labor supply models: Acemoglu and Autor (2011), Stinebrickner (2001), Rust (1989), Buchholz, Shum, and Xu (2023), Myck and Reed (2006)
  - We extend dynamic structural labor supply to microtask platform workers.

Literature Review

- Dynamic labor supply models: Acemoglu and Autor (2011), Stinebrickner (2001), Rust (1989), Buchholz, Shum, and Xu (2023), Myck and Reed (2006)
  - We extend dynamic structural labor supply to microtask platform workers.
- Online Worker Motivation (Experimental models): Sigala (2015), Lausen et al. (2016), Horton and Chilton (2010)
  - Prior work focuses on short-run incentives; we study long-run (learning-by-doing) behavior.

Literature Review

- Dynamic labor supply models: Acemoglu and Autor (2011), Stinebrickner (2001), Rust (1989), Buchholz, Shum, and Xu (2023), Myck and Reed (2006)
  - We extend dynamic structural labor supply to microtask platform workers.
- Online Worker Motivation (Experimental models): Sigala (2015), Lausen et al. (2016), Horton and Chilton (2010)
  - Prior work focuses on short-run incentives; we study long-run (learning-by-doing) behavior.
- Methodology: Rust (1987), Abbring and Daljord (2020), Sargent and Ljungqvist (2000)

– Results Summary –

Results Summary

- Reduced-form findings
  1. Workers tend to repeat similar tasks.
  2. Task attributes and demographics significantly affect continuation.

Results Summary

- Reduced-form findings
  1. Workers tend to repeat similar tasks.
  2. Task attributes and demographics significantly affect continuation.
- Structural model findings
  1. Workers have non-refundable "experience" costs (project-specific training).
  2. Workers learn by doing
     - $\Downarrow$ short-term costs for repeating a task.
  3. Learning-by-doing works for complex tasks.

– Data –

Data

Source

- One East-European crowdsourcing platform big data (every second)

▸ What is crowdsourcing? ▸ Task example

Structure

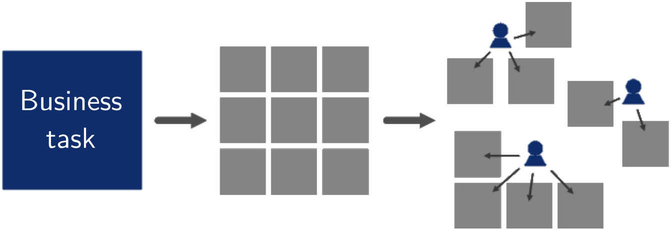
- Overall, 1,027 projects, 61,812 workers, 6,854,405 assignments
- Four data levels: worker, project, assignment, and session levels

Final sample

- Workers: no bots and occasional platform visitors
- Assignments: only browser-available and submitted assignments
- Time frame: One week, October 1 – October 7, 2021

Definition

Crowdsourcing is a term that combines "crowd" and "sourcing." "Crowd" means a large group of people. "Sourcing" comes from "outsourcing." Crowdsourcing differs from outsourcing because it often does not need professional knowledge or skills. Crowdsourcing involves breaking down a big project into small tasks (microtasks). Workers distributed over the Internet do these tasks.

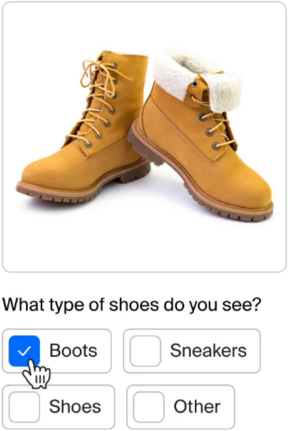


Back

Motivation

Crowdsourcing Platforms: Project Types

- Micro tasks
  - Filling out surveys
  - Classification
  - Web searching
  - Content moderation
  - Text or audio transcription

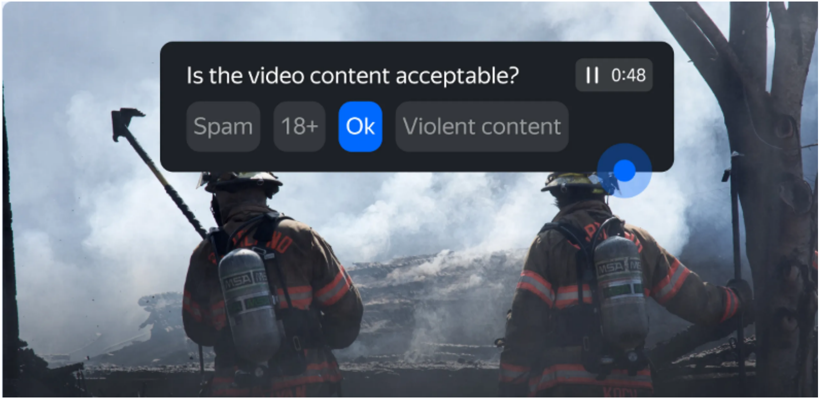


Back

Data

Crowdsourcing Platforms: Project Types

Content Moderation



Back

– Reduced Form Analysis –

Definition

A session is defined as a working day or shift for an online worker, with new sessions starting if more than one hour has passed since the last task performed

Reduced Form Analysis

Empirical Evidence: Project Continuation

Definition

A session is defined as a working day or shift for an online worker, with new sessions starting if more than one hour has passed since the last task performed

Evidence 1: Workers tend to do similar tasks

Table: Summary Statistics for Session Characteristics

Statistic	Mean	St. Dev.	Median
Session hour	0.854	1.366	0.367
Tasks completed	19.34	32.87	10
Earnings ($)	0.381	1.893	0.08
Hourly earnings ($/h)	0.626	3.52	0.207
Number of projects	2.391	2.164	2

Note: Overall, there are 358,564 sessions.

Data

Empirical Evidence: Demographic Characteristics

Evidence 2: Age, higher education, and English skills influence the likelihood of continuing the same project

Table: Probit model with Socio-Demographic Variables

	Probability of continuing the same project		
	(1)	(2)	(3)
Age	-0.0019*	-0.0049*	-0.0019*
	(0.0010)	(0.0025)	(0.0010)
Gender	-0.0162	-0.0063	-0.0162
	(0.0142)	(0.0065)	(0.0142)
University degree	0.0360***	0.1023***	0.0360***
	(0.0081)	(0.0130)	(0.0081)
Experience	-0.0002	-0.0030	-0.0002
	(0.0005)	(0.0019)	(0.0005)
Able to speak English	0.0957***	0.4210***	0.0957***
	(0.0227)	(0.0503)	(0.0227)
Observations	5,318,777	5,329,220	5,318,777
Project fixed effects	✓		✓
Project type fixed effects	✓	✓	

Reduced Form Analysis

Empirical Evidence: Learning-by-doing

Evidence 3: Assignment completion time decreases, i.e., productivity increases, as the number of tasks increases

▸ Number of tasks

Table: Assignment Completion Time by Number of Tasks

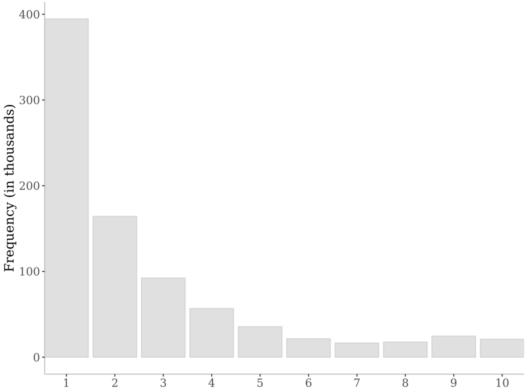
Number of tasks	Completion time (min)	Task price	Price per minute	Grade
1	3.231	0.023	0.017	0.774
2	2.201	0.018	0.016	0.744
3	1.946	0.019	0.018	0.725
4	1.883	0.019	0.018	0.705
5	1.888	0.019	0.019	0.755

Note: Overall, there are 1,639,926 assignments.

Data

Number of Tasks

Figure: Tasks Completed Within One Worker-Project-Session Distribution



Note: This graph shows how many tasks worker completes before switching to another project. We illustrate distribution for first 10 bins (tasks) that account for 90% of worker-project-sessions.

Back

Reduced Form Analysis

Empirical Evidence: Assignment Characteristics

Evidence 4: Price, training, grade, complexity influence the likelihood of continuing the same project

Table: Probit model with Assignment Characteristics

	Probability of continuing the same project		
	(1)	(2)	(3)
Price $\geq$ \$ 0.06	0.4858***	0.4848***	0.5014***
	(0.0109)	(0.0109)	(0.0115)
\$ 0.02 $\leq$ Price $<$\$ 0.06	0.1117***	0.1118***	0.1471***
	(0.0051)	(0.0051)	(0.0052)
Training	0.2429***	0.2442***	0.2536***
	(0.0048)	(0.0048)	(0.0050)
Project grade	0.3741***	0.3762***	0.3839***
	(0.0058)	(0.0058)	(0.0067)
Task complexity	-0.0352***	-0.0337***	-0.0893***
	(0.0079)	(0.0079)	(0.0082)
Observations	1,259,860	1,259,860	1,259,860
Worker fixed effects	✓	✓	✓
Session fixed effects		✓	✓
Project type fixed effects			✓

Back to dynamic model

– Dynamic Structural Model –

Behavioral Model
Setting

- Crowdsourcing platform workers complete tasks of various kinds of projects

Behavioral Model
Setting

- Crowdsourcing platform workers complete tasks of various kinds of projects
- We model the project switching decisions

Behavioral Model
Setting

- Crowdsourcing platform workers complete tasks of various kinds of projects
- We model the project switching decisions
- *i* represents a worker, *t* indexes time: $t \in [1, \infty)$

Behavioral Model
Setting

- Crowdsourcing platform workers complete tasks of various kinds of projects
- We model the project switching decisions
- i represents a worker, t indexes time: $t \in [1, \infty)$
- At time t, worker i has state s_{it} and an action a_{it}

Behavioral Model
Setting

- Crowdsourcing platform workers complete tasks of various kinds of projects
- We model the project switching decisions
- i represents a worker, t indexes time: $t \in [1, \infty)$
- At time t, worker i has state s_{it} and an action a_{it}
- There are observable state variables x_{it} and unobservable state variable ϵ_{it} ($s_{it} = \{x_{it}, \epsilon_{it}\}$)

Behavioral Model
Setting

- Crowdsourcing platform workers complete tasks of various kinds of projects
- We model the project switching decisions
- i represents a worker, t indexes time: $t \in [1, \infty)$
- At time t, worker i has state s_{it} and an action a_{it}
- There are observable state variables x_{it} and unobservable state variable ϵ_{it} ($s_{it} = \{x_{it}, \epsilon_{it}\}$)
- Every period t the worker decides to switch projects ($a_{it} = switch$) or stay on the same one ($a_{it} = stay$)

Behavioral Model
Setting

- Crowdsourcing platform workers complete tasks of various kinds of projects
- We model the project switching decisions
- i represents a worker, t indexes time: $t \in [1, \infty)$
- At time t, worker i has state s_{it} and an action a_{it}
- There are observable state variables x_{it} and unobservable state variable ϵ_{it} ($s_{it} = \{x_{it}, \epsilon_{it}\}$)
- Every period t the worker decides to switch projects ($a_{it} = switch$) or stay on the same one ($a_{it} = stay$)

Behavioral Model

Current Utility Function: Overview

Utility Function

$$U^{stay} = \pi(g_{it}, p_{it}; \theta_\pi) - c(tr_{it}, h_{it}, k_{it}; \theta_c) + \epsilon_{it}^{stay}$$
$$U^{switch} = -f(e_{it}; \theta_f) + \epsilon_{it}^{switch}$$

where

- $\pi(g_{it}, p_{it}; \theta_\pi)$ – profits from staying on the same project
- $c(tr_{it}, h_{it}, k_{it}; \theta_c)$ – costs from staying on the same project

Utility Function

$$U^{stay} = \pi(g_{it}, p_{it}; \theta_\pi) - c(tr_{it}, h_{it}, k_{it}; \theta_c) + \epsilon_{it}^{stay}$$
$$U^{switch} = -f(e_{it}; \theta_f) + \epsilon_{it}^{switch}$$

where

- $\pi(g_{it}, p_{it}; \theta_\pi)$ – profits from staying on the same project
- $c(tr_{it}, h_{it}, k_{it}; \theta_c)$ – costs from staying on the same project
- $f(e_{it}; \theta_f)$ – project switching costs

Utility Function

$$U^{stay} = \pi(g_{it}, p_{it}; \theta_\pi) - c(tr_{it}, h_{it}, k_{it}; \theta_c) + \epsilon_{it}^{stay}$$
$$U^{switch} = -f(e_{it}; \theta_f) + \epsilon_{it}^{switch}$$

where

- $\pi(g_{it}, p_{it}; \theta_\pi)$ – profits from staying on the same project
- $c(tr_{it}, h_{it}, k_{it}; \theta_c)$ – costs from staying on the same project
- $f(e_{it}; \theta_f)$ – project switching costs
- ϵ_{it}^{stay} and ϵ_{it}^{switch} are unobservable state variables when agent continues the same project and switches to other, respectively
- in other words, contemporaneous idiosyncratic shocks unobservable for a researcher but observable for agents

Utility Function

$$U^{stay} = \pi(g_{it}, p_{it}; \theta_\pi) - c(tr_{it}, h_{it}, k_{it}; \theta_c) + \epsilon_{it}^{stay}$$

Profit function:

$$\pi(g_{it}, p_{it}; \theta_\pi) = \theta_g g_{it} + \theta_p p_{it}$$

where

- *Project grade* (g_{it}) shows the pleasantness of the project

Behavioral Model

Current Utility Function: Profit Function

Utility Function

$$U^{stay} = \pi(g_{it}, p_{it}; \theta_{\pi}) - c(tr_{it}, h_{it}, k_{it}; \theta_c) + \epsilon_{it}^{stay}$$

Profit function:

$$\pi(g_{it}, p_{it}; \theta_{\pi}) = \theta_g g_{it} + \theta_p p_{it}$$

where

- *Project grade* (g_{it}) shows the pleasantness of the project
- It is equal to one if project grade is higher than the median grade; zero otherwise

Behavioral Model

Current Utility Function: Profit Function

Utility Function

$$U^{stay} = \pi(g_{it}, p_{it}; \theta_\pi) - c(tr_{it}, h_{it}, k_{it}; \theta_c) + \epsilon_{it}^{stay}$$

Profit function:

$$\pi(g_{it}, p_{it}; \theta_\pi) = \theta_g g_{it} + \theta_p p_{it}$$

where

- *Project grade* (g_{it}) shows the pleasantness of the project
- It is equal to one if project grade is higher than the median grade; zero otherwise
- *Paid task price* (p_{it}) distinguishes three price categories: 1) $\$0.01$ USD, 2) $\$0.02 - \$0.05$ USD, 3) $\geq \$0.06$ USD

Behavioral Model

Current Utility Function: Continuation Cost Function

Utility Function

$$U^{stay} = \pi(g_{it}, p_{it}; \theta_\pi) - c(tr_{it}, h_{it}, k_{it}; \theta_c) + \epsilon_{it}^{stay}$$

Continuation cost function:

$$c(tr_{it}, h_{it}, k_{it}; \theta_c) = \theta_{tr} tr_{it} + \theta_h h_{it} + \theta_k k_{it} + \theta_{hk} h_{it} \times k_{it}$$

where

- *Training* (tr_{it}) - the existence of training or examination tasks

Behavioral Model

Current Utility Function: Continuation Cost Function

Utility Function

$$U^{stay} = \pi(g_{it}, p_{it}; \theta_\pi) - c(tr_{it}, h_{it}, k_{it}; \theta_c) + \epsilon_{it}^{stay}$$

Continuation cost function:

$$c(tr_{it}, h_{it}, k_{it}; \theta_c) = \theta_{tr} tr_{it} + \theta_h h_{it} + \theta_k k_{it} + \theta_{hk} h_{it} \times k_{it}$$

where

- *Training* (tr_{it}) - the existence of training or examination tasks
- *Task complexity* (h_{it}) is one if the average task completion time is over five minutes; zero otherwise

Behavioral Model

Current Utility Function: Continuation Cost Function

Utility Function

$$U^{stay} = \pi(g_{it}, p_{it}; \theta_\pi) - c(tr_{it}, h_{it}, k_{it}; \theta_c) + \epsilon_{it}^{stay}$$

Continuation cost function:

$$c(tr_{it}, h_{it}, k_{it}; \theta_c) = \theta_{tr} tr_{it} + \theta_h h_{it} + \theta_k k_{it} + \theta_{hk} h_{it} \times k_{it}$$

where

- *Training* (tr_{it}) - the existence of training or examination tasks
- *Task complexity* (h_{it}) is one if the average task completion time is over five minutes; zero otherwise
- *Number of completed assignments*, k_{it}, shows a learning-by-doing effect

Behavioral Model
Current Utility Function: Continuation Cost Function

Utility Function

$$U^{stay} = \pi(g_{it}, p_{it}; \theta_\pi) - c(tr_{it}, h_{it}, k_{it}; \theta_c) + \epsilon_{it}^{stay}$$

Continuation cost function:

$$c(tr_{it}, h_{it}, k_{it}; \theta_c) = \theta_{tr} tr_{it} + \theta_h h_{it} + \theta_k k_{it} + \theta_{hk} h_{it} \times k_{it}$$

where

- *Training* (tr_{it}) - the existence of training or examination tasks
- *Task complexity* (h_{it}) is one if the average task completion time is over five minutes; zero otherwise
- *Number of completed assignments*, k_{it}, shows a learning-by-doing effect
- *Interaction of task complexity and number of completed assignments*, $(h \times k)_{it}$, shows how worker reaction to the task complexity changes with "experience"

Behavioral Model
Current Utility Function: Switching Cost Function

Utility Function

$$U^{switch} = -f(e_{it}; \theta_f) + \epsilon_{it}^{switch}$$

Switching cost function:

$$f(e_{it}; \theta_f) = \theta_{SW} + \theta_e e_{it}$$

where

- θ_{SW} - the fixed switching cost

Utility Function

$$U^{switch} = -f(e_{it}; \theta_f) + \epsilon_{it}^{switch}$$

Switching cost function:

$$f(e_{it}; \theta_f) = \theta_{SW} + \theta_e e_{it}$$

where

- θ_{SW} - the fixed switching cost
- e_{it} - the loss of past experience from switching to another project

Behavioral Model

Value Function

- Every period t, the worker i sees state values s_{it}

Behavioral Model
Value Function

- Every period t, the worker i sees state values s_{it}
- She then chooses action $a_{it} \in (stay; switch)$ to maximize her discounted expected utility stream, i.e., value function

Behavioral Model
Value Function

- Every period t, the worker i sees state values s_{it}
- She then chooses action $a_{it} \in (stay; switch)$ to maximize her discounted expected utility stream, i.e., value function
- Solution to the decision problem is a sequence of decision rules $\{a_{i,t+1}, a_{i,t+2}, ...\}$

Behavioral Model
Value Function

- Every period t, the worker i sees state values s_{it}
- She then chooses action $a_{it} \in (stay; switch)$ to maximize her discounted expected utility stream, i.e., value function
- Solution to the decision problem is a sequence of decision rules $\{a_{i,t+1}, a_{i,t+2}, ...\}$
- Bellman's principle of optimality lets us to break down the problem into a sequence of single decisions

Baseline Rust model

Behavioral Model
Value Function

Choice-specific value functions:[1]

$$V^{stay} = U^{stay} + \beta E_{x',\epsilon'|x,\epsilon,d=stay} V(x',\epsilon')$$
$$V^{switch} = U^{switch} + \beta E_{x',\epsilon'|x,\epsilon,d=switch} V(x',\epsilon')$$

$$V(x,\epsilon) = \max_{d \in (stay, switch)} \left[V^{stay}, V^{switch} \right]$$

[1] We remove the indexes i and t and denote the period $t+1$ as $'$ to save space

Behavioral Model
Value Function

Choice-specific value functions:[1]

$$V^{stay} = U^{stay} + \beta E_{x',\epsilon'|x,\epsilon,d=stay} V(x',\epsilon')$$
$$V^{switch} = U^{switch} + \beta E_{x',\epsilon'|x,\epsilon,d=switch} V(x',\epsilon')$$

$$V(x,\epsilon) = \max_{d\in(stay,switch)} \left[V^{stay}, V^{switch} \right]$$

- The optimal decision rule is the argument, d, of value function's maximum value:
$$\alpha(x,\epsilon) = \arg \max_{d\in(stay,switch)} \left[V^{stay}, V^{switch} \right]$$

[1] We remove the indexes i and t and denote the period $t+1$ as $'$ to save space

Empirical Strategy: Estimation of Parameters

Nested Fixed Point Algorithm

- The nested fixed point algorithm (NFXP) is a gradient iterative search method NFXP in detail
- It calculates the maximum likelihood estimator of structural parameters

Empirical Strategy: Estimation of Parameters
Nested Fixed Point Algorithm

- The nested fixed point algorithm (NFXP) is a gradient iterative search method

NFXP in detail

- It calculates the maximum likelihood estimator of structural parameters
  - Outer algorithm looks for different parameter values $\hat{\theta}_u$

Empirical Strategy: Estimation of Parameters

Nested Fixed Point Algorithm

- The nested fixed point algorithm (NFXP) is a gradient iterative search method

▸ NFXP in detail

- It calculates the maximum likelihood estimator of structural parameters
  - Outer algorithm looks for different parameter values $\hat{\theta}_u$
  - Inner algorithm solves the dynamic programming problem for each trial $\hat{\theta}_u$
  - It computes the value function and then log-likelihood function

– Results –

Results

Baseline Results (1)

▸ Comparison with probit

Utility Function

$$U^{stay} = \pi(g_{it}, p_{it}; \theta_\pi) - c(tr_{it}, h_{it}, k_{it}; \theta_c) + \epsilon_{it}(0)$$

Policy implications: Raising the monetary reward and creating a pleasant and high-quality project environment increases workers' staying profit

Table: Rust Model Estimates – Baseline Results

Functions	Variables	Estimates	(std)
Profit function, π	Price $\geq$ \$ 0.06	0.089***	(0.014)
	\$ 0.02 $\leq$ Price $<$ \$ 0.06	0.104***	(0.006)
	Grade $\geq$ median	0.021***	(0.004)
Staying cost function, c	Training	-0.268***	(0.005)
	Number of tasks	-0.038***	(0.001)
	Complexity	0.068***	(0.008)
	Number of tasks $\times$ Complexity	-0.048***	(0.003)
Switching cost function, f	Fixed cost	-0.110***	(0.004)
	Loss of past experience	0.118***	(0.009)

Note: Overall, there are 1,639,926 assignments.

Results

Baseline Results (2)

Utility Function

$$U^{stay} = \pi(g_{it}, p_{it}; \theta_\pi) - c(tr_{it}, h_{it}, k_{it}; \theta_c) + \epsilon_{it}(0)$$

Policy implications: Adding training and assessment tasks to projects, including more tasks in projects, dividing a complex task into simpler tasks will decrease workers' staying costs

Table: Rust Model Estimates – Baseline Results

Functions	Variables	Estimates	(std)
Profit function, π	Price $\geq$ \$ 0.06	0.089***	(0.014)
	\$ 0.02 $\leq$ Price $<$\$ 0.06	0.104***	(0.006)
	Grade $\geq$ median	0.021***	(0.004)
Staying cost function, c	Training	-0.268***	(0.005)
	Number of tasks	-0.038***	(0.001)
	Complexity	0.068***	(0.008)
	Number of tasks $\times$ Complexity	-0.048***	(0.003)
Switching cost function, f	Fixed cost	-0.110***	(0.004)
	Loss of past experience	0.118***	(0.009)

Note: Overall, there are 1,639,926 assignments.

Utility Function

$$U^{switch} = -f(e_{it}; \theta_f) + \epsilon_{it}(1)$$

Policy implications: Switching costs are low but the loss of past experience (training on past projects) increases them

Table: Rust Model Estimates – Baseline Results

Functions	Variables	Estimates	(std)
Profit function, π	Price $\geq$ \$ 0.06	0.089***	(0.014)
	\$ 0.02 $\leq$ Price <\$ 0.06	0.104***	(0.006)
	Grade $\geq$ median	0.021***	(0.004)
Staying cost function, c	Training	-0.268***	(0.005)
	Number of tasks	-0.038***	(0.001)
	Complexity	0.068***	(0.008)
	Number of tasks $\times$ Complexity	-0.048***	(0.003)
Switching cost function, f	Fixed cost	-0.110***	(0.004)
	Loss of past experience	0.118***	(0.009)

Note: Overall, there are 1,639,926 assignments.

High education group

– Conclusion –

Conclusion

This paper

- Sheds a light on the crowdsourcing platform workers' dynamic choice in undertaking tasks
  - Studies the determinants of the labor supply decision focusing on learning costs
  - Gives policy recommendations regarding price, training, and task complexity

Conclusion

This paper

- Sheds a light on the crowdsourcing platform workers' dynamic choice in undertaking tasks
  - Studies the determinants of the labor supply decision focusing on learning costs
  - Gives policy recommendations regarding price, training, and task complexity

Further research directions

- Demographics and counterfactuals Counterfactuals results
- Adding other decisions (leave the platform, unsubmit, etc.)
- Extending the period

Results

High Education

Table: Rust Model Estimates – High Education

Functions	Variables	Baseline	High Education
Profit function, π	Price $\geq$ \$ 0.06	0.089***	0.121***
		(0.014)	(0.018)
	\$ 0.02 $\leq$ Price $<$ \$ 0.06	0.104***	0.12***
		(0.006)	(0.008)
	Grade $\geq$ median	0.021***	0.008
		(0.004)	(0.006)
Continuation cost function, c	Training	-0.268***	-0.334***
		(0.005)	(0.007)
	Number of tasks	-0.038***	-0.04***
		(0.001)	(0.001)
	Complexity	0.068***	0.019
		(0.008)	(0.012)
	Number of tasks $\times$ Complexity	-0.048***	-0.024***
		(0.003)	(0.004)
Switching cost function, f	Fixed cost	-0.110***	-0.076***
		(0.004)	(0.006)
	Loss of past experience	0.118***	0.099***
		(0.009)	(0.014)

Note: Overall, there are 1,639,926 assignments.

Back

Results
High Education: Interpretation

- Schooling has a positive impact on productivity and learning efficiency (Same signs though the different size of effects)
- High education group reacts more positively to high prices, learning-by-doing and training
- Effects for grade are smaller
- The effects of the loss of past experience and task complexity are also smaller

Back

Results
Counterfactuals

Table: Counterfactuals

	Long-run switching probability
Baseline	0.222
No past experience effect	0.245
No learning effect	0.409
No past experience and learning effect	0.424

Note: We run simulations with 10,000 assignments.

- The chance of switching in the long run goes up as the past experience effect fades – if workers lack training in past projects, they would switch more
- Learning effect – training plus learning by doing
- Same results as the past experience effect but much larger in size

Back

Thank you for your attention!

– Appendix –

Referee comments

1. Education: Education signals a general skill level. Why more skilled workers may be more inclined to continue with the same project? (instead, a greater variety of tasks may seem easy for them – hence richer options to switch to).

A — Learning-by-doing effects level out general skills effects. As a worker completes more tasks on the same project, she becomes better.

Referee comments

2. Number of tasks: A higher number of tasks from the same project may either simplify further worker's experience with the project or render her bored: it seems natural that this hypothesis is checked by introducing a number of assignments performed along with its square at the probit regression stage.

A — If the number of tasks is one, it means the project has switched. The number of tasks variable will solely explain the switching decision.

3. Project type: Why is it the decision to switch to another project, rather than to another project category?

A — Skills are mainly applicable within a project.

Referee comments

4. Income reference point: In the literature review section, the relevance of an income reference point is discussed. Author could have tested, whether workers are guided by such a reference and whether it influences switching decisions. One of the ways to do that is to adjust the profit part of the utility function, accounting for the distance between a worker's daily earnings and its certain reference level (e.g., median within project category) at the brink of her switching decision.

A — A novel framework is necessary for the income reference point study. Income reference isn't about switching decisions but quitting decisions. We have to see when sessions change. We have to use different variables then. Prices $\Rightarrow$ cumulative earnings. The number of completed tasks within a project-session $\Rightarrow$ within a session.

Appendix

outline

Labor and learning

- Labor is a service that is delivered over time
- It is accompanied by investments in human capital (e.g., learning a job-specific skill)
- Workers' inertia – persistence in labor supply choices
    - Learning-by-doing effect
        - over time, workers improve skills and performance by doing
    - Training or certification costs are sunk costs of job switching
        - some job-specific skills may not be applicable outside
    - Fixed cost of job switching
        - e.g., updating a resume, networking events, job search platforms, interview attire, and possibly relocation expenses
- Firms benefit from no switching
    - secure returns to investments in on-the-job training, specific skills

Motivation
Online Labor Market

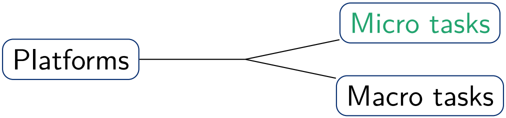
Online labor market:

- The online labor market has grown with the Internet
  - Globally, 163 million freelancers work online (Kässi, Lehdonvirta, and Stephany, 2021)
  - Number of online platforms is over 777 in 2020 (Rani et al., 2021)
- Online labor market has a significant policy impact as it:
  - brings new earning chances in underemployed areas
  - lowers job barriers for underrepresented groups (OECD, 2018)

Motivation
Platforms

"[Online labor] Markets have emerged not "in the wild," but within the context of highly structured platforms created by for-profit intermediaries" (Horton, 2010)

- There are two types of online labor platforms
- Platforms with macrotasks require professional knowledge, e.g., language teaching, graphic designing, IT programming, R&D
- We deal with microtask platforms, i.e., "crowdsourcing" platforms



Motivation

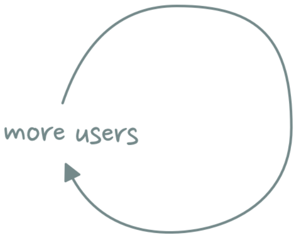
Crowdsourcing Platforms: Examples

A central aspect of platform economics – network effects

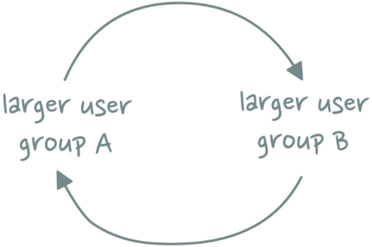
Network effect

Value of existing participants increases with the entry of new participants

Platform network effect



Cross-group network effect



Back

Motivation
Requesters

Requesters' benefits from workers' inertia:

1. Labeled data for AI: original purpose of creating crowdsourcing platforms was to help machine learning engineers create a large amount of data to train the machine (Bussler, 2021)
2. Research data collection: a cheap tool for academic researchers to collect data
   - helpful for social scientists (Brandt, 2022)
   - US student samples $\Rightarrow$ heterogeneous people from different countries and races via Internet access

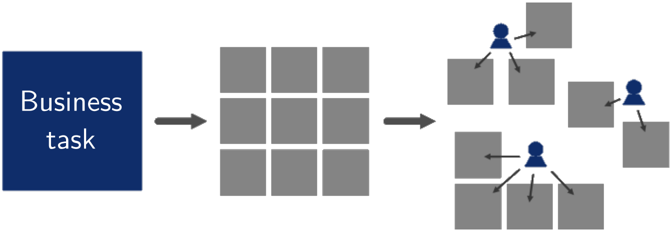
Back

Motivation

Crowdsourcing Platforms

Definition

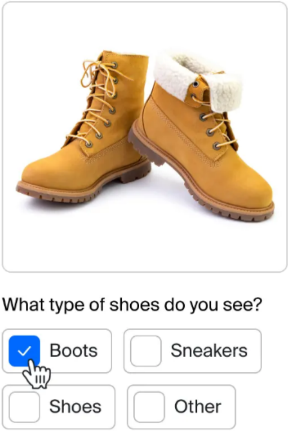
Crowdsourcing is a term that combines "crowd" and "sourcing." "Crowd" means a large group of people. "Sourcing" comes from "outsourcing." Crowdsourcing differs from outsourcing because it often does not need professional knowledge or skills. Crowdsourcing involves breaking down a big project into small tasks (microtasks). Workers distributed over the Internet do these tasks.



Motivation

Crowdsourcing Platforms: Project Types

- Micro tasks
  - Filling out surveys
  - Classification
  - Web searching
  - Content moderation
  - Text or audio transcription

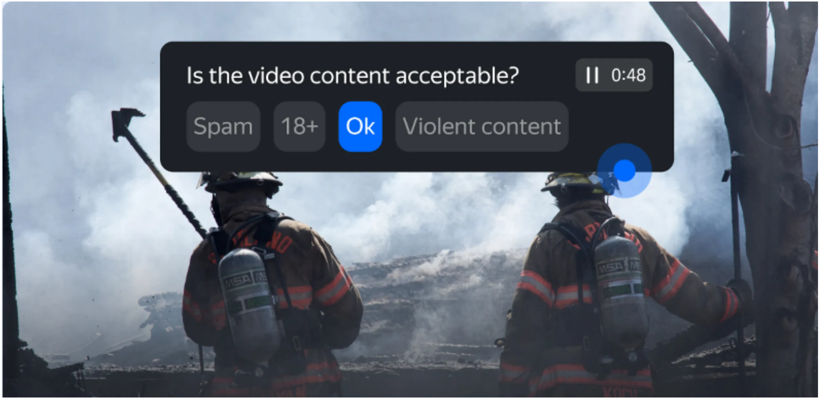


Back

Motivation

Crowdsourcing Platforms: Project Types

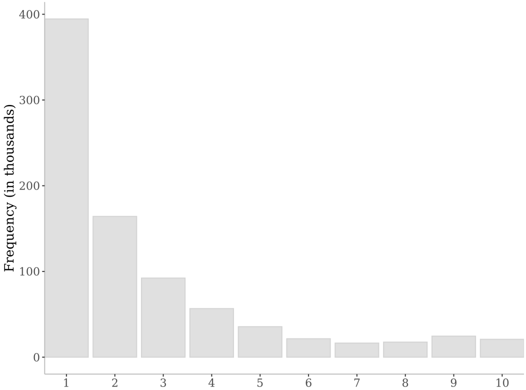
Content Moderation



Back

Data
Number of Tasks

Figure: Tasks Completed Within One Worker-Project-Session Distribution



Note: This graph shows how many tasks worker completes before switching to another project. We illustrate distribution for first 10 bins (tasks) that account for 90% of worker-project-sessions.

Back

Rust's Single Agent Model

Transition

- The decision at period t influences future state variables
- We assume that future state variables change under agent's expectations for future states, i.e., the transition probabilities
- We assume that the transition probability function, $p(x', \epsilon' | a, x, \epsilon)$, follows a Markov process
- Conditional independence assumption on the Markovian transition probabilities:

$$p(x', \epsilon' | a, x, \epsilon) = p(\epsilon' | a, x', x, \epsilon) \cdot p(x' | a, x, \epsilon) = p(\epsilon' | x') \cdot p(x' | a, x).$$

1. First step is to factor the joint density into a conditional term and a marginal
2. Second step is simplification
3. ϵ's cumulative distribution function is $G_\epsilon(\epsilon')$. Thus, $p(\epsilon' | x') = G_\epsilon(\epsilon')$ for all x'

Behavioral Model

Transition

In our case, we use the log-likelihood criterion to estimate θ. Likelihood function for a single agent looks like:

$$L(x_1, ..., x_T; a_1, ..., a_T | x_0, a_0; \theta) = \prod_{t=1}^{T} p(a_t, x_t | x_0, a_0, ..., x_{t-1}, a_{t-1}; \theta)$$

$$= \prod_{t=1}^{T} p(a_t, x_t | x_{t-1}, a_{t-1}; \theta) = \prod_{t=1}^{T} p(a_t | x_t; \theta) \times p(x_t | x_{t-1}, a_{t-1}; \theta_x)$$

The second step implies that x_t and i_t evolve as first-order Markov process. It relies on the conditional serial independence of ϵ. Last step breaks the joint probability into a conditional times a marginal. θ_x denotes the parameters of state evolution.

$$\log L = \sum_{t=1}^{T} \log p(a' | x'; \theta) + \sum_{t=1}^{T} \log p(x' | x, a; \theta_x)$$

Back

- We estimate the parameters of state evolution θ_x
- All transition probabilities for variables, except the k, are set from the data (Aguirregabiria and Mira, 2010)
- The number of tasks completed within one project, k, has the following deterministic transition probabilities:

$$p(k'|a=0,k) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, p(k'|a=1,k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Behavioral Model

Conditional Choice Probability

- Now we want to estimate the parameters of utility function θ_u
- $p(a'|x';\theta)$ is the Conditional Choice Probability (CCP). We obtain it by integrating the optimal decision rule over the unobservable state variables
- Let's define $\hat{V}(a, x_{it}) \equiv \tilde{V}(a, x_{it}) - \epsilon_{it}(a)$.
- Then conditional choice probability, $Prob(a = 0|x, \theta)$, is equal to:

$$
\begin{aligned}
Prob(a = 0|x, \theta) &= Prob(\tilde{V}(0, x_{it}) > \tilde{V}(1, x_{it})) \\
&= Prob(\hat{V}(0, x_{it}) + \epsilon_{it}(0) > \hat{V}(1, x_{it}) + \epsilon_{it}(1)) \\
&= Prob(\epsilon_{it}(0) - \epsilon_{it}(1) > \hat{V}(1, x_{it}) - \hat{V}(0, x_{it})) = \\
&\frac{\exp(\hat{V}(0, x_{it}))}{\exp(\hat{V}(0, x_{it})) + \exp(\hat{V}(1, x_{it}))}
\end{aligned}
$$

Behavioral Model

Conditional Choice Probability

We define $\hat{U}(a) \equiv U(a) - \epsilon(a)$. Then CCP takes form:

$$Prob(a|x,\theta) = \frac{\exp(\hat{V}(a, x_{it}))}{\sum_{a \in A} \exp(\hat{V}(a, x_{it}))} =$$

$$\frac{\exp(\hat{U}(a, x_{it}; \theta_u) + \beta E_{x', \epsilon' | a, x, \epsilon} V(x', \epsilon'))}{\sum_{a \in A} \exp(\hat{U}(a, x_{it}; \theta_u) + \beta E_{x', \epsilon' | a, x, \epsilon} V(x', \epsilon'))}$$

Behavioral Model
Discount Factor

- Agents use β to discount their future value
- In many applications, one does not estimate the parameter β
- This is because it is poorly identified (Abbring and Daljord, 2020)
- We assume β is 0.95 as it is the usual value for single agent models (Sargent and Ljungqvist, 2000)

Nested Fixed Point Algorithm

Algorithm:

1. Impose arbitrary values on a vector of structural parameters $\hat{\theta}_u^0$
2. Value function iteration: Start with $\hat{\theta}_u^0$
   - In the inner algorithm, we obtain the vector $\bar{V}(\hat{\theta}_u^0)$ by iterating in the Bellman equation
   - We start from guess $\bar{V}_0$
   - We iterate the Bellman equation $\bar{V}^{h+1}(x,\epsilon) = \log(\sum_{a\in A} \hat{U}(a',x';\hat{\theta}_u^0) + \beta E_{x',\epsilon'|a,x,\epsilon}\bar{V}^h(x',\epsilon'))$ until it converges
   - We stop when $\sup_{a,x}|\bar{V}^{h+1}(a,x) - \bar{V}^h(a,x)| < 0.000001$
   - h denotes the epoch of the value function iteration

Nested Fixed Point Algorithm

Algorithm:

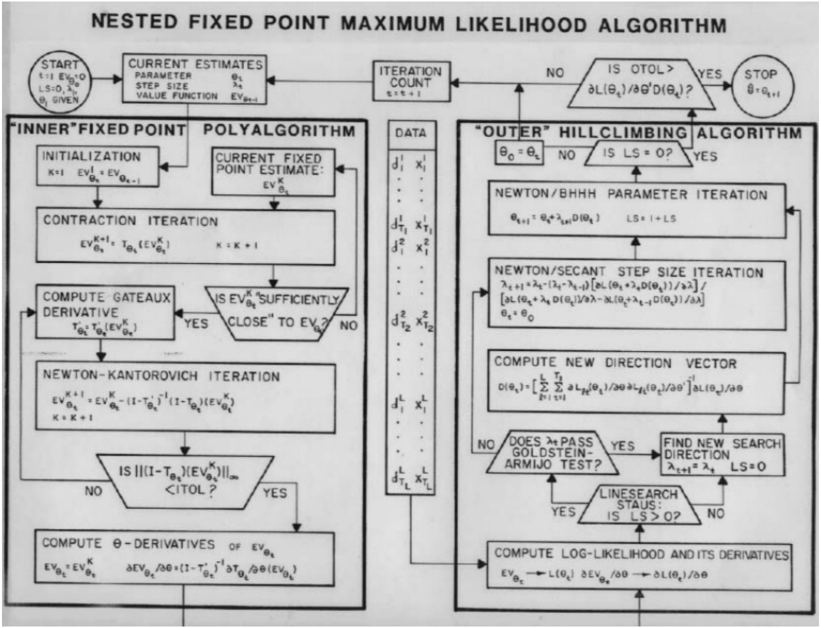
3. Given $\hat{\theta}_0$ and $\bar{V}(\hat{\theta}_u^0)$, we construct the choice probabilities $P(a|x, \hat{\theta}_u^0)$, the gradient $\frac{\partial L(\hat{\theta}_u^0)}{\partial \theta}$, and the matrix $\frac{\partial \bar{V}(\hat{\theta}_u^0)}{\partial \theta_u'}$
   In the outer algorithm, gradient $\frac{\partial L(\hat{\theta}_u^0)}{\partial \theta_u}$ is used to make a new iteration to obtain $\hat{\theta}_u^1$
   We proceed in this way until the distance between $\hat{\theta}_u^{m+1}$ and $\hat{\theta}_u^m$ or the difference in the likelihoods is smaller than a pre-specified convergence constant
   Outer algorithm is defined in a following way:

$$\hat{\theta}_u^{m+1} = \hat{\theta}_u^m + \left(\sum_{i=1}^{N} \frac{\partial L_i(\hat{\theta}_u^m)}{\partial \theta_u} \frac{\partial L_i(\hat{\theta}_u^m)}{\partial \theta_u'} \right)^{-1} \left(\sum_{i=1}^{N} \frac{\partial L_i(\hat{\theta}_u^m)}{\partial \theta_u} \right),$$

where $L_i(\theta_u)$ is the log-likelihood function for individual i

Back

Nested Fixed Point Algorithm



- Worker benefits from the higher price and pleasantness of the project
- Learning-by-doing and training have a positive impact on continuation
- Workers do not benefit from complexity itself but with interaction with number of tasks (i.e., with learning) workers get lower costs

Results
High Education

Table: Rust Model Estimates – High Education

Functions	Variables	Baseline	High Education
Profit function, π	Price $\geq$ \$ 0.06	0.089*** (0.014)	0.121*** (0.018)
	\$ 0.02 $\leq$ Price $<$ \$ 0.06	0.104*** (0.006)	0.12*** (0.008)
	Grade $\geq$ median	0.021*** (0.004)	0.008 (0.006)
Continuation cost function, c	Training	-0.268*** (0.005)	-0.334*** (0.007)
	Number of tasks	-0.038*** (0.001)	-0.04*** (0.001)
	Complexity	0.068*** (0.008)	0.019 (0.012)
	Number of tasks $\times$ Complexity	-0.048*** (0.003)	-0.024*** (0.004)
Switching cost function, f	Fixed cost	-0.110*** (0.004)	-0.076*** (0.006)
	Loss of past experience	0.118*** (0.009)	0.099*** (0.014)

Note: Overall, there are 1,639,926 assignments.

Back

Results

High Education: Interpretation

- Schooling has a positive impact on productivity and learning efficiency (Same signs though the different size of effects)
- High education group reacts more positively to high prices, learning-by-doing and training
- Effects for grade are smaller
- The effects of the loss of past experience and task complexity are also smaller

Back

Results
Counterfactuals

Table: Counterfactuals

	Long-run switching probability
Baseline	0.222
No past experience effect	0.245
No learning effect	0.409
No past experience and learning effect	0.424

Note: We run simulations with 10,000 assignments.

- The chance of switching in the long run goes up as the past experience effect fades – if workers lack training in past projects, they would switch more
- Learning effect – training plus learning by doing
- Same results as the past experience effect but much larger in size

Back