Bonus or Not?
Learn to Reward in Crowdsourcing

Ming Yin, Yiling Chen

Harvard University

IJCAI-15, Buenos Aires, Argentina, July 25-31, 2015

Monetary Rewards in Crowdsourcing

Monetary rewards are widely used in crowdsourcing.

(Performance-contingent) Monetary rewards can affect work quality! [Harris 2011; Yin *et al.* 2013; Ho *et al.* 2015]

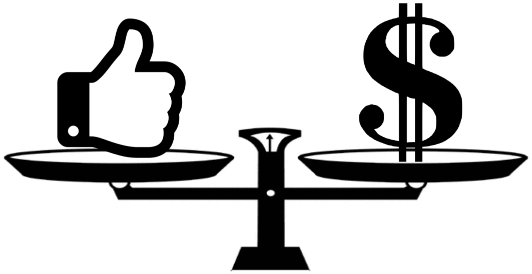
How to “Wisely” Use Bonus?

Whether and when to provide performance-contingent bonuses for a worker working on a sequence of tasks?

Common practice: Fixed or random policy

How do workers react to bonuses provided in selected tasks in a sequence?

What is the trade-off between improved quality and increased costs?



Our Approach

How do workers react to bonuses provided in selected tasks in a sequence?

What is the trade-off between improved quality and increased costs?

Input-output hidden Markov model

+

Requester utility function

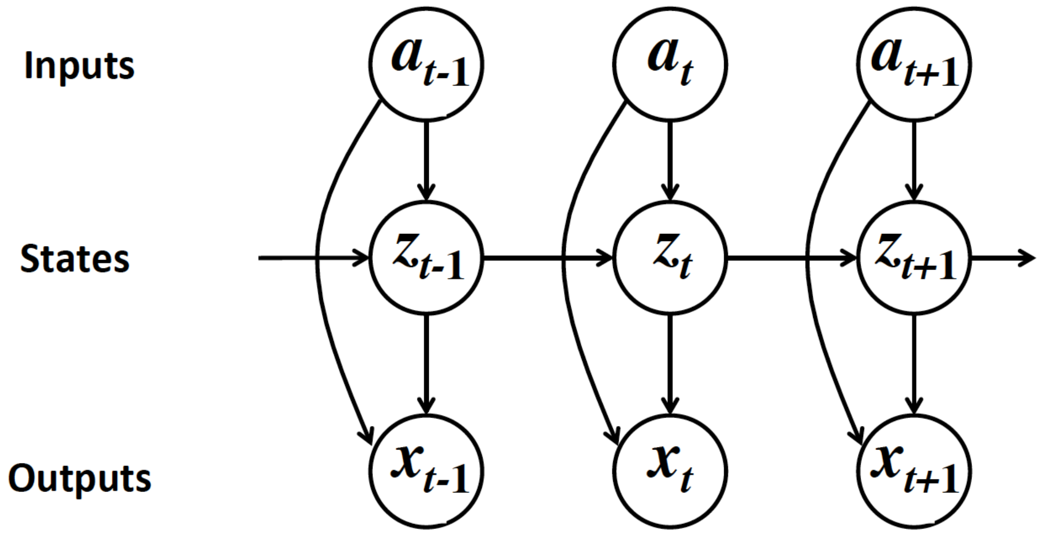
Whether and when to provide performance-contingent bonuses for a worker working on a sequence of tasks?

Online decision making

Requester's utility improves 27% compared to following the common practice policy!

Characterize the Bonus Impact with IOHMM

Transition probability: $P_{tr}(z_t \mid z_{t-1}, a_t)$



Emission probability: $P_e(x_t \mid z_t, a_t)$

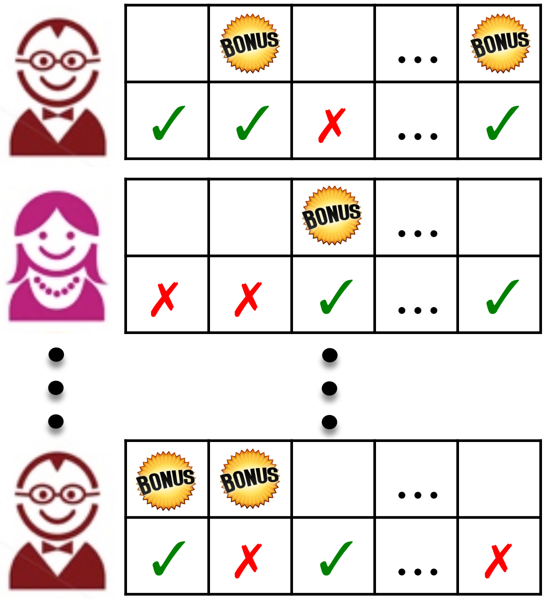
a_t : whether bonus is provided in task t

z_t : worker's hidden state in task t (out of K possibilities)

x_t : whether the answer in task t has high-quality

Learn the IOHMM

Training Dataset



Expectation-Maximization

Learned IOHMM

Transition Probability Matrices ($\boldsymbol{T}^a$)

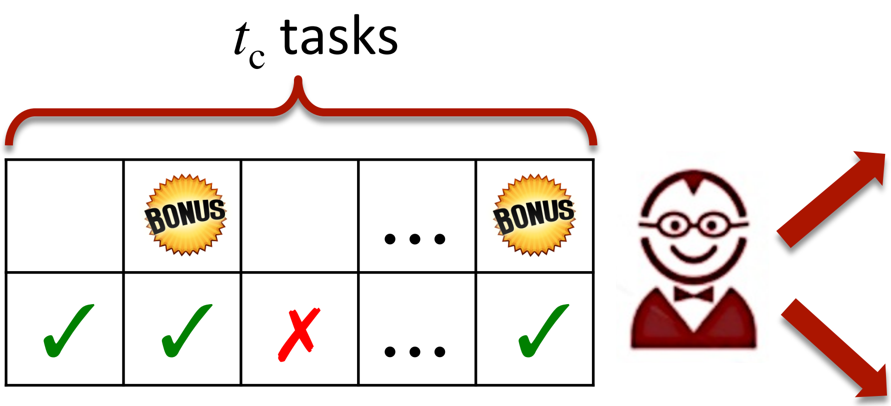
Emission Probability Matrices ($\boldsymbol{E}^a$)

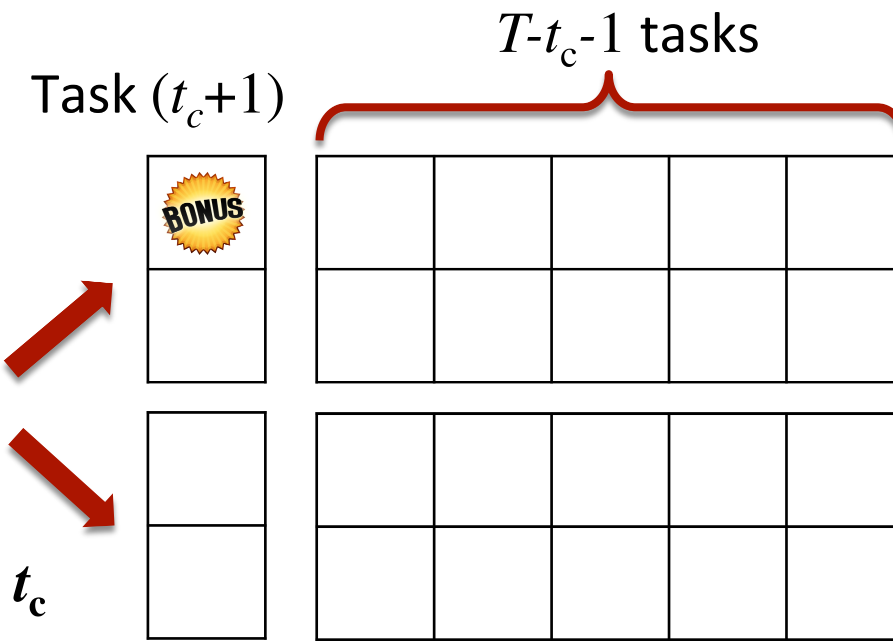
Initial State Belief ($\boldsymbol{b}_0$)

The Online Decision Making Problem

Requesters are assumed to have a quasi-linear utility function:

$$U = w_l N_{LQ} + w_h N_{HQ} - cN_{bonus}$$





For a worker who has completed t_c tasks in a sequence (hence $b(t_c)$)

Whether to place bonus on his next task to maximize requester's overall utility in the T-task session?

Decision Making with the Learned IOHMM

$EU_{\max}(\boldsymbol{b}, a, l)$: The maximum expected utility a requester can obtain for the next l tasks, when the current state belief is $\boldsymbol{b}$, the input for the next task is a.

$$EU_{\max}(\boldsymbol{b}, a, l) = \begin{cases} l = 1: & R(\boldsymbol{b}, a) \\ l > 1: & R(\boldsymbol{b}, a) + \sum_{x \in \{0,1\}} \left(\sum_{i=1}^{K} b(i) \sum_{j=1}^{K} P_{tr}(j|i, a) P_e(x|j, a) \right) V(b'_{a,x}, l-1) \end{cases}$$

$$a_{t_c+1} = \underset{a \in \{0,1\}}{\operatorname{argmax}}\, EU_{max}(\boldsymbol{b}(t_c), a, T - t_c)$$

Heuristic Algorithms

This problem is equivalent to solve a finite-horizon POMDP, which is computationally hard.

- ***n*-step look-ahead**: Consider to maximize requester utility in (at most) the next n tasks.
- **MLS-MDP**: Estimate the most likely sequence (MLS) and follow the optimal MDP policy for the current state.
- **Q-MDP**: Calculate Q-function values of the MDP; choose the input level that maximize the expected Q-function values given the current belief.

MTurk Experiment Task: Word Puzzle

O	V	H	L	E	G	T	Y	J	E	Z	J
P	V	L	M	Q	A	K	L	M	L	P	Q
M	A	A	Y	W	M	T	A	K	U	G	M
P	G	E	Q	G	E	G	M	X	U	A	Q
Y	A	K	B	C	W	H	R	S	S	M	K
Q	M	K	G	A	M	E	A	N	K	E	I
A	E	G	D	E	M	K	L	L	M	J	Y
E	M	G	M	A	N	B	E	A	X	F	P
M	X	A	G	Z	K	I	G	G	A	M	E
W	G	D	B	D	S	Y	P	Z	A	E	X
Z	P	W	Y	W	J	E	H	T	M	M	M
R	K	I	K	M	I	O	V	N	L	V	E

O	V	H	L	**E**	**G**	T	Y	J	**E**	Z	J
P	V	L	**M**	Q	**A**	K	L	**M**	L	P	Q
M	A	**A**	Y	W	**M**	T	**A**	K	U	**G**	M
P	**G**	E	Q	G	**E**	**G**	M	X	U	**A**	Q
Y	**A**	K	B	C	W	H	R	S	S	**M**	K
Q	**M**	K	**G**	**A**	**M**	**E**	A	N	K	**E**	I
A	**E**	G	D	**E**	**M**	K	L	L	**M**	J	Y
E	M	G	**M**	**A**	N	B	E	**A**	X	F	P
M	X	**A**	**G**	Z	K	I	**G**	**G**	**A**	**M**	**E**
W	**G**	D	B	D	S	Y	P	Z	**A**	E	X
Z	P	W	Y	W	J	E	H	T	M	**M**	M
R	K	I	K	M	I	O	V	N	L	V	**E**

Find the appearance of the target word as many times as possible

Collect the Training Dataset

- $\mathbf{50}$ workers: each completes a HIT of $T = \mathbf{9}$ word puzzle tasks.

- $\mathbf{20\%}$ of the tasks are randomly selected as bonus tasks.

- A worker can earn a $\mathbf{5}$-cent bonus in a bonus task if she finds out more than $\mathbf{80\%}$ of all appearances of the target word (i.e. the answer is of high-quality).

A Peek into the Learned IOHMM

$K = 2$ hidden states in the Learned IOHMM.

Initial state belief: $\boldsymbol{b_0} = (0.67, 0.33)$

Emission probability matrices:

No Bonus $\quad \boldsymbol{E}^0 = \begin{array}{c} \\ S1 \\ S2 \end{array} \begin{array}{c} \begin{array}{cc} LQ & HQ \end{array} \\ \begin{pmatrix} 0.10 & 0.90 \\ 0.88 & 0.12 \end{pmatrix} \end{array}, \quad \boldsymbol{E}^1 = \begin{array}{c} \\ S1 \\ S2 \end{array} \begin{array}{c} \begin{array}{cc} LQ & HQ \end{array} \\ \begin{pmatrix} 0.13 & 0.87 \\ 0.61 & 0.39 \end{pmatrix} \end{array}$ Bonus

- State 1: “Diligent”
- State 2: “Lazy”, but can be improved with bonus

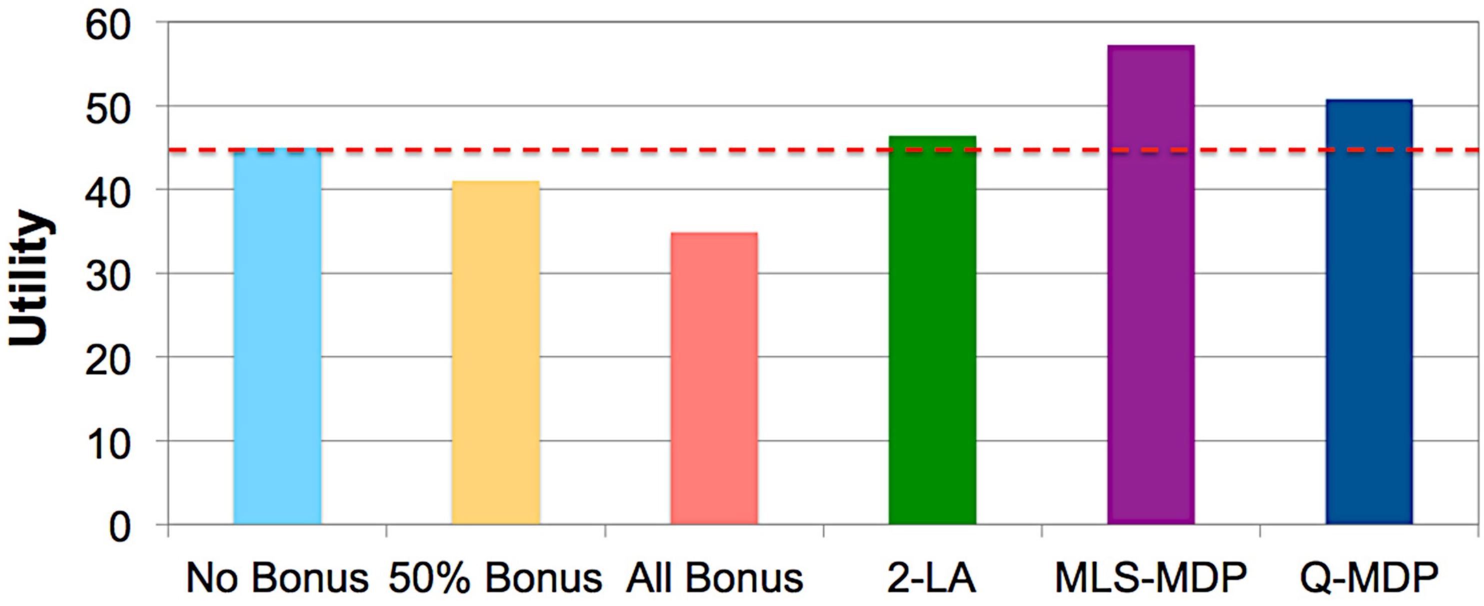
Transition probability matrices:

No Bonus $\quad \boldsymbol{T}^0 = \begin{array}{c} \\ S1 \\ S2 \end{array} \begin{array}{c} \begin{array}{cc} S1 & S2 \end{array} \\ \begin{pmatrix} 0.92 & 0.08 \\ 0 & 1 \end{pmatrix} \end{array}, \quad \boldsymbol{T}^1 = \begin{array}{c} \\ S1 \\ S2 \end{array} \begin{array}{c} \begin{array}{cc} S1 & S2 \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0.09 & 0.91 \end{pmatrix} \end{array}$ Bonus

- No Bonus: a small chance to “slack off” from the diligent state
- Bonus: a small chance to “promote” to the diligent state

The Effectiveness of Dynamic Control

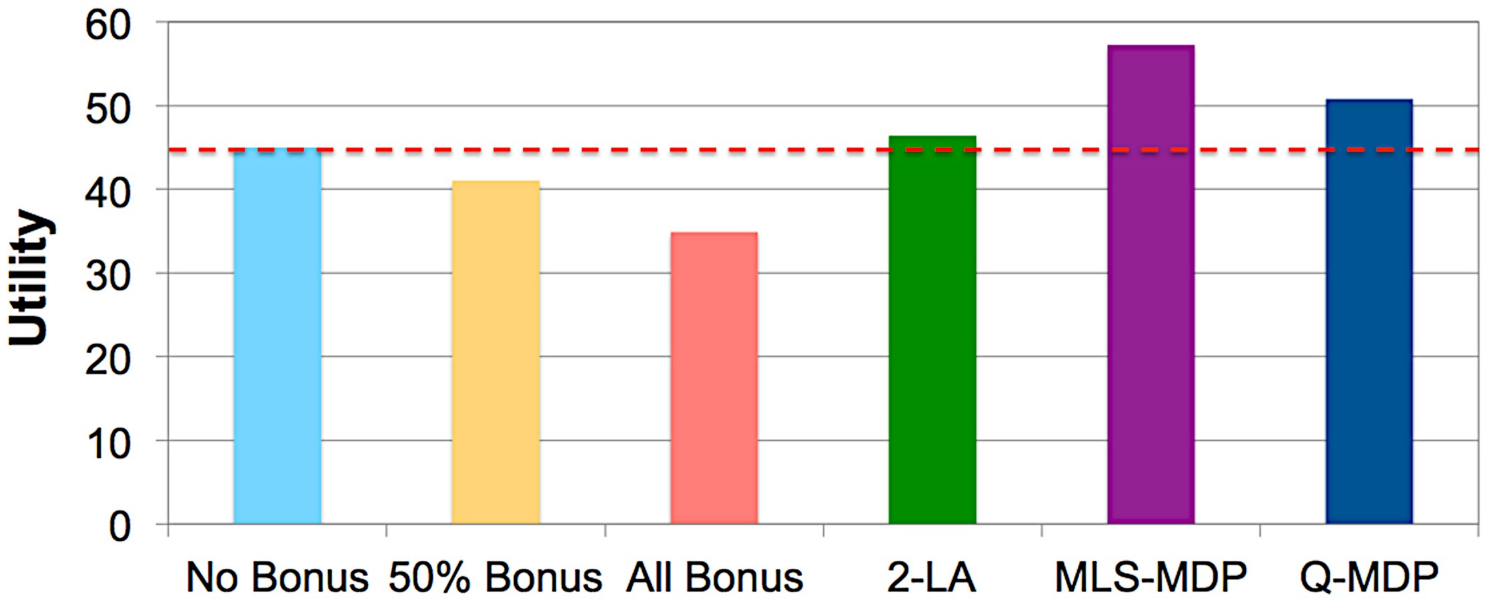
- **6** treatments, **50** random workers per treatment
- $w_l = 0, w_h = 0.15, c = 0.05$



An increase in requester utility beyond the best baseline as much as **27.22%**!

The Effectiveness of Dynamic Control

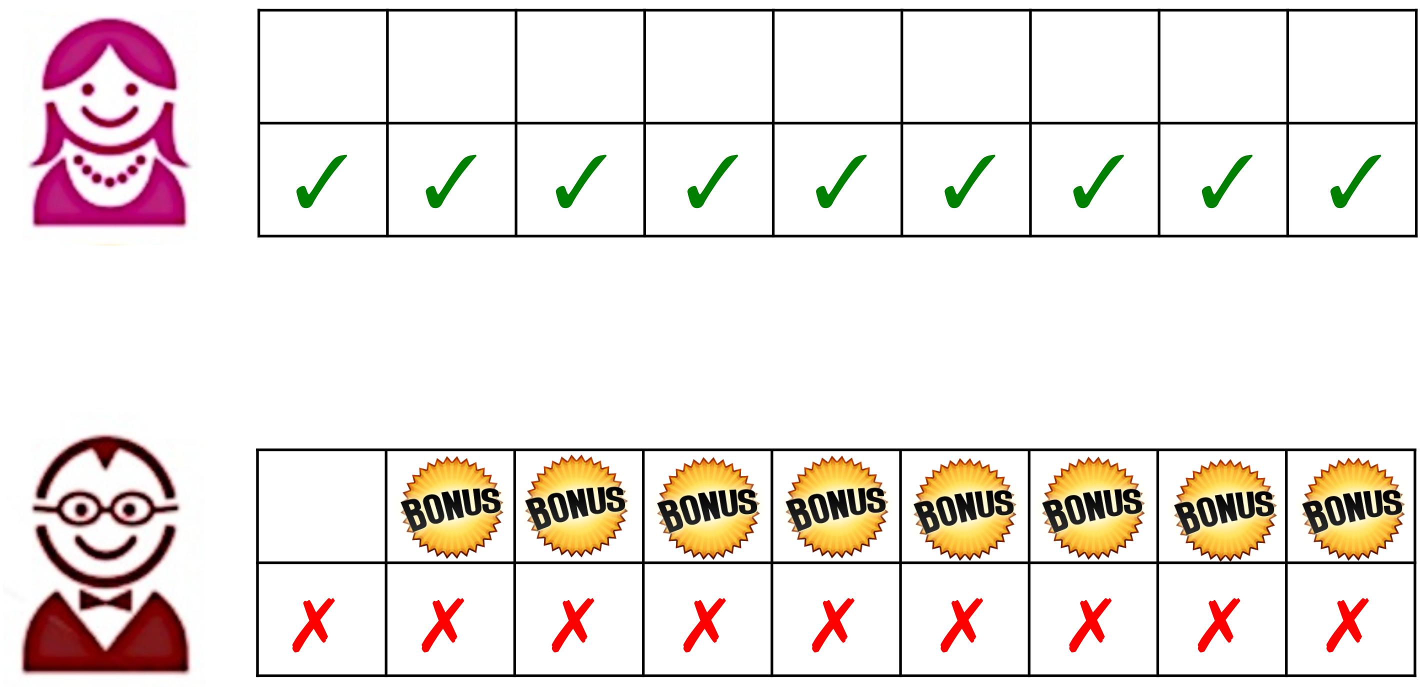
- **6** treatments, **50** random workers per treatment
- $w_l = 0, w_h = 0.15, c = 0.05$



# HQ answers	300	331	349	322	395	368
Cost ($)	0	8.60	17.45	1.90	2.00	4.40

More high-quality answers, **lower** costs!

Dynamic Control Example



Strategically focus on incentivizing "lazy" workers!

Dynamic Control Example

	BONUS	BONUS	BONUS	BONUS	BONUS			
✗	✗	✗	✓	✓	✓	✓	✓	✓

				BONUS	BONUS	BONUS		
✓	✓	✗	✗	✓	✓	✓	✓	✗

Provide bonus at the right timing!

How Robust is Our Approach?

- **Different worker behavior models**
  - Model 1: Worker's accuracy changes from acc^l to acc^h when bonus is provided
  - Model 2: Work quality is decided by comparing the actual payment with the reference of "appropriate payment level"
- **Different worker characteristics**
  - Skill level (α_i); Sensitivity to monetary rewards (β_i)
- **Different composition of worker population**
  - $p_{\alpha,} p_{\beta}$

Follow a dynamic control policy always match or improve requester utility compared to the best baseline!

Summary

- We propose to use IOHMM to learn the impact of bonuses on work quality and use this model to dynamically control the placement of bonus.
- Both MTurk experiment and simulation show that our approach can robustly lead to improved requester utility.
- **Shows the promise of algorithmically controlling incentives in crowdsourcing!**

Thank you!