Incentives in Crowdsourcing: A Game-theoretic Approach

ARPITA GHOSH

Cornell University

NIPS 2013

Workshop on Crowdsourcing: Theory, Algorithms, and Applications

Contribution online from the crowds:

Contribution online from the crowds:

- Reviews (Amazon, Yelp), online Q&A sites (Y! Answers, Quora, StackOverflow), discussion forums
- Wikipedia
- Social media: Blogs, YouTube, ...

Contribution online from the crowds:

- Reviews (Amazon, Yelp), online Q&A sites (Y! Answers, Quora, StackOverflow), discussion forums
- Wikipedia
- Social media: Blogs, YouTube, ...
- Crowdsourcing: Paid and unpaid; microtasks and challenges
  - Amazon Mechanical Turk, Citizen Science (GalaxyZoo, FoldIt), Games with a Purpose, contests (Innocentive, Topcoder)

Contribution online from the crowds:

- Reviews (Amazon, Yelp), online Q&A sites (Y! Answers, Quora, StackOverflow), discussion forums
- Wikipedia
- Social media: Blogs, YouTube, ...
- Crowdsourcing: Paid and unpaid; microtasks and challenges
  - Amazon Mechanical Turk, Citizen Science (GalaxyZoo, FoldIt), Games with a Purpose, contests (Innocentive, Topcoder)
- Online education: Peer-learning, peer-grading

- Quality, participation varies widely across systems

- Quality, participation varies widely across systems
- How to *incentivize* high participation and effort?

- Quality, participation varies widely across systems
- How to *incentivize* high participation and effort?
- Two components to designing incentives:
  - Social psychology: What constitutes a reward?

Incentives and collective effort

- Quality, participation varies widely across systems
- How to *incentivize* high participation and effort?
- Two components to designing incentives:
  - Social psychology: What constitutes a reward?
  - Rewards are *limited*: How to *allocate* among self-interested users?
- A *game-theoretic* framework for incentive design

The game-theoretic approach to incentive design

- System design induces *rules* specifying allocation of rewards
- Self-interested users choose *actions* to maximize own payoff

The game-theoretic approach to incentive design

- System design induces *rules* specifying allocation of rewards
- Self-interested users choose *actions* to maximize own payoff
  - Participation ('Endogenous entry')

The game-theoretic approach to incentive design

- System design induces *rules* specifying allocation of rewards
- Self-interested users choose *actions* to maximize own payoff
  - Participation ('Endogenous entry')
  - Revealing information truthfully (ratings, opinions, ...)

The game-theoretic approach to incentive design

- System design induces *rules* specifying allocation of rewards
- Self-interested users choose *actions* to maximize own payoff
  - Participation ('Endogenous entry')
  - Revealing information truthfully (ratings, opinions, ...)
  - Effort:
    - Quality of content (UGC sites)
    - Output accuracy (crowdsourcing)
    - Quantity: Number of contributions, attemped tasks
    - Speed of response (Q&A forums), ...
- Incentive *design*: Allocate reward to align agent's incentives with system

Incentive design for crowdsourcing

Reward allocation problem varies across systems:

Reward allocation problem varies across systems: Why?

- Constraints, reward regimes, vary with nature of reward:
  - Monetary; social-psychological (attention, status, ...)
  - Attention rewards: Diverging [GM11, GH11]; subset constraints [GM12]
  - Money-like rewards: Bounded; sum constraints [GM12]

Incentive design for crowdsourcing

Reward allocation problem varies across systems: Why?

- Constraints, reward regimes, vary with nature of reward:
  - Monetary; social-psychological (attention, status, ...)
  - Attention rewards: Diverging [GM11, GH11]; subset constraints [GM12]
  - Money-like rewards: Bounded; sum constraints [GM12]
- Observability of (value of) agents' output
  - Can only reward what you can see

Reward allocation problem varies across systems: Why?

- Constraints, reward regimes, vary with nature of reward:
  - Monetary; social-psychological (attention, status, ...)
  - Attention rewards: Diverging [GM11, GH11]; subset constraints [GM12]
  - Money-like rewards: Bounded; sum constraints [GM12]
- Observability of (value of) agents' output
  - Can only reward what you can see
  - Perfect rank-ordering: Contests [...]
  - Imperfect: Noisy votes in UGC [EG13, GH13]
  - Unobservable: Judgement elicitation [DG13]

- The setting: User-generated content
  (Reviews, Q&A forums, comments, videos, articles, . . .)
- *Quality* of contributions varies widely:

- The setting: User-generated content
  (Reviews, Q&A forums, comments, videos, articles, . . .)
- *Quality* of contributions varies widely: Sites want to display best contributions

- The setting: User-generated content
  (Reviews, Q&A forums, comments, videos, articles, ...)
- *Quality* of contributions varies widely: Sites want to display best contributions
- But quality is not directly observable:

- The setting: User-generated content
  (Reviews, Q&A forums, comments, videos, articles, . . .)
- *Quality* of contributions varies widely: Sites want to display best contributions
- But quality is not directly observable: Infer quality from viewer votes

Learning & incentives in user-generated content

Joint work with Patrick Hummel, ITCS'13

- The setting: User-generated content
  (Reviews, Q&A forums, comments, videos, articles, ...)
- *Quality* of contributions varies widely: Sites want to display best contributions
- But quality is not directly observable: Infer quality from viewer votes
- How to display contributions to optimize overall viewer experience?

A multi-armed bandit problem

A multi-armed bandit problem

- Learning contribution qualities: Multi-armed bandit problem
  - Arms: Contributions
  - Success probability: Contribution's 'quality'

A multi-armed bandit problem

- Learning contribution qualities: Multi-armed bandit problem
  - Arms: Contributions
  - Success probability: Contribution's 'quality'
- Contributors: Agents with cost to quality, benefit from views

A multi-armed bandit problem

- Learning contribution qualities: Multi-armed bandit problem
  - Arms: Contributions
  - Success probability: Contribution's 'quality'
- Contributors: Agents with cost to quality, benefit from views
- *Arms are endogenous*!
  - Contributors *choose* whether to participate, content quality

A multi-armed bandit problem

- Learning contribution qualities: Multi-armed bandit problem
  - Arms: Contributions
  - Success probability: Contribution's 'quality'
- Contributors: Agents with cost to quality, benefit from views
- *Arms are endogenous*!
  - Contributors *choose* whether to participate, content quality
- What is a good learning algorithm in this setting?

Overview

- Strategic contributors: Decide participation, quality
- Viewers vote on displayed contributions
- Mechanism: Decides which contribution to display
- Metric: *Equilibrium* regret

Model: Content and feedback

- Contribution quality q: Probability of viewer upvote

Model: Content and feedback

- Contribution quality q: Probability of viewer upvote
- Stream of T viewers: Each viewer shown, votes on, one contribution

Model: Content and feedback

- Contribution quality q: Probability of viewer upvote
- Stream of T viewers: Each viewer shown, votes on, one contribution
- Viewers need not vote 'perfectly': $q \in [0, \gamma]$

Model: Content and feedback

- Contribution quality q: Probability of viewer upvote
- Stream of T viewers: Each viewer shown, votes on, one contribution
- Viewers need not vote 'perfectly': $q \in [0, \gamma]$
  - Mechanism should be robust to $\gamma < 1$

- $K = K(T)$ potential contributors

Contributors

- $K = K(T)$ potential contributors
- Contributors are *strategic* agents

- $K = K(T)$ potential contributors
- Contributors are *strategic* agents, choosing
  - Whether or not to participate:

Contributors

- $K = K(T)$ potential contributors
- Contributors are *strategic* agents, choosing
  - Whether or not to participate: Probability β_i

Contributors

- $K = K(T)$ potential contributors
- Contributors are *strategic* agents, choosing
  - Whether or not to participate: Probability β_i
  - Contribution quality: q_i

Contributors

- $K = K(T)$ potential contributors
- Contributors are *strategic* agents, choosing
  - Whether or not to participate: Probability β_i
  - Contribution quality: q_i
- Actual number of contributions (arms): $k(T) \leq K(T)$

Contributor utilities

- Cost: Quality q incurs cost $c(q)$
  - $c(q)$ increasing, continuously differentiable

Contributor utilities

- Cost: Quality q incurs cost $c(q)$
  - $c(q)$ increasing, continuously differentiable
- Benefit from *views*

Contributor utilities

- Cost: Quality q incurs cost $c(q)$
  - $c(q)$ increasing, continuously differentiable
- Benefit from *views*
  - Psychological ([Huberman et al 09, ...]) or monetary benefit

Contributor utilities

- Cost: Quality q incurs cost $c(q)$
  - $c(q)$ increasing, continuously differentiable
- Benefit from *views*
  - Psychological ([Huberman et al 09, ...]) or monetary benefit
  - n_i^t: Views allocated to i until period t
  - Total benefit to i: n_i^T

Contributor utilities

- Cost: Quality q incurs cost $c(q)$
  - $c(q)$ increasing, continuously differentiable
- Benefit from *views*
  - Psychological ([Huberman et al 09, ...]) or monetary benefit
  - n_i^t: Views allocated to i until period t
  - Total benefit to i: n_i^T
- *Mechanism*: Decides which contribution to display at t

Contributor utilities

- Cost: Quality q incurs cost $c(q)$
  - $c(q)$ increasing, continuously differentiable
- Benefit from *views*
  - Psychological ([Huberman et al 09, ...]) or monetary benefit
  - n_i^t: Views allocated to i until period t
  - Total benefit to i: n_i^T
- *Mechanism*: Decides which contribution to display at t
- Utility: $u_i = E[n_i^T(q_i, q_{-i}, k(T))] - c(q_i)$

Mapping to MAB

- $K(T)$ potential contributors, or arms
- Viewer t: Pull of arm at time t
- T: Time horizon or total number of viewers
- Content quality q_i: Success probability of arm i

Mapping to MAB

- $K(T)$ potential contributors, or arms
- Viewer t: Pull of arm at time t
- T: Time horizon or total number of viewers
- Content quality q_i: Success probability of arm i
- Actual number of arms $k(T)$, qualities q_i, determined **endogenously** in response to learning algorithm

How good is a learning algorithm as a *mechanism*?

How good is a learning algorithm as a *mechanism*?

- Performance measure: *Equilibrium* regret

How good is a learning algorithm as a *mechanism*?

- Performance measure: *Equilibrium* regret
  - Recall contributors **choose** $q \in [0, \gamma]$

How good is a learning algorithm as a *mechanism*?

- Performance measure: *Equilibrium* regret
  - Recall contributors **choose** $q \in [0, \gamma]$
- *Strong regret* of mechanism $\mathcal{M}$:

How good is a learning algorithm as a *mechanism*?

- Performance measure: *Equilibrium* regret
  - Recall contributors **choose** $q \in [0, \gamma]$
- *Strong regret* of mechanism $\mathcal{M}$: Regret wrt γ,

How good is a learning algorithm as a *mechanism*?

- Performance measure: *Equilibrium* regret
  - Recall contributors **choose** $q \in [0, \gamma]$
- *Strong regret* of mechanism $\mathcal{M}$: Regret wrt γ, in symmetric mixed-strategy Bayes-Nash equilibrium $(\beta, F(q))$,

How good is a learning algorithm as a *mechanism*?

- Performance measure: *Equilibrium* regret
  - Recall contributors **choose** $q \in [0, \gamma]$
- *Strong regret* of mechanism $\mathcal{M}$: Regret wrt γ, in symmetric mixed-strategy Bayes-Nash equilibrium $(\beta, F(q))$,

$$R(T) = \gamma T - E[\sum_{t=1}^{T} q_t]$$

How good is a learning algorithm as a *mechanism*?

- Performance measure: *Equilibrium* regret
  - Recall contributors **choose** $q \in [0, \gamma]$
- *Strong regret* of mechanism $\mathcal{M}$: Regret wrt γ, in symmetric mixed-strategy Bayes-Nash equilibrium $(\beta, F(q))$,

$$R(T) = \gamma T - E[\sum_{t=1}^{T} q_t]$$

- Strong sublinear equilibrium regret:

How good is a learning algorithm as a *mechanism*?

- Performance measure: *Equilibrium* regret
  - Recall contributors **choose** $q \in [0, \gamma]$
- *Strong regret* of mechanism $\mathcal{M}$: Regret wrt γ, in symmetric mixed-strategy Bayes-Nash equilibrium $(\beta, F(q))$,

$$R(T) = \gamma T - E[\sum_{t=1}^{T} q_t]$$

- Strong sublinear equilibrium regret: $\lim_{T\to\infty} \frac{R(T)}{T} = 0$

How good is a learning algorithm as a *mechanism*?

- Performance measure: *Equilibrium* regret
  - Recall contributors **choose** $q \in [0, \gamma]$
- *Strong regret* of mechanism $\mathcal{M}$: Regret wrt γ, in symmetric mixed-strategy Bayes-Nash equilibrium $(\beta, F(q))$,

$$R(T) = \gamma T - E[\sum_{t=1}^{T} q_t]$$

- Strong sublinear equilibrium regret: $\lim_{T\to\infty} \frac{R(T)}{T} = 0$ in **every** symmetric equilibrium of $\mathcal{M}$

The UCB algorithm, as a mechanism

- q_i^t: Estimated quality of i at time t
- UCB algorithm $\mathcal{M}_{\mathrm{UCB}}$:
  - Display all arms once, then
  - Display $i = \arg\max \quad q_i^t + \sqrt{\frac{2 \ln T}{n_i^t}}$

The UCB algorithm, as a mechanism

- q_i^t: Estimated quality of i at time t
- UCB algorithm $\mathcal{M}_{\mathrm{UCB}}$:
  - Display all arms once, then
  - Display $i = \arg\max \quad q_i^t + \sqrt{\frac{2 \ln T}{n_i^t}}$
- Theorem: Mechanism $\mathcal{M}_{\mathrm{UCB}}$ always has a symmetric mixed-strategy equilibrium $(\beta, F(q))$

UCB as a mechanism: The good news

UCB as a mechanism: The good news

Theorem

If $K(T)$ is such that $\lim_{T\to\infty} \frac{T}{K(T)\ln T} = \infty$*:*

UCB as a mechanism: The good news

Theorem

If $K(T)$ is such that $\lim_{T \to \infty} \frac{T}{K(T) \ln T} = \infty$:

- $\beta = 1$ *in any equilibria of* $\mathcal{M}_{\mathrm{UCB}}$ *for sufficiently large* T

UCB as a mechanism: The good news

Theorem

If $K(T)$ is such that $\lim_{T\to\infty} \frac{T}{K(T)\ln T} = \infty$:

- $\beta = 1$ *in any equilibria of* $\mathcal{M}_{\mathrm{UCB}}$ *for sufficiently large* T
- *For any fixed* $q^* < \gamma$*, the probability of choosing quality* $q \leq q^*$ *in any equilibrium goes to* 0 *as* $T \to \infty$.

UCB as a mechanism: The good news

Theorem

If $K(T)$ is such that $\lim_{T\to\infty} \frac{T}{K(T)\ln T} = \infty$:

- *$\beta = 1$ in any equilibria of $\mathcal{M}_{\mathrm{UCB}}$ for sufficiently large T*
- *For any fixed $q^* < \gamma$, the probability of choosing quality $q \leq q^*$ in any equilibrium goes to 0 as $T \to \infty$.*
- *$\mathcal{M}_{\mathrm{UCB}}$ achieves strong sublinear equilibrium regret.*

UCB as a mechanism: The bad news

Theorem

Suppose $\lim_{T\to\infty} \frac{T}{K(T)} = r < \infty$.

UCB as a mechanism: The bad news

Theorem

Suppose $\lim_{T\to\infty} \frac{T}{K(T)} = r < \infty$. *Then for sufficiently large* T, *any equilibrium has the property that no agent chooses quality greater than* $q = c_T^{-1}(1 + c(0))$.

Improving equilibrium regret: A modified UCB mechanism

- $\mathcal{M}_{\mathrm{UCB-MOD}}$: Run UCB on random subset of $\min\{\lfloor\sqrt{T}\rfloor, k(T)\}$ arms
  - Exploring random subset: $\mathcal{M}_{\mathrm{1-FAIL}}$ [Berry et al'97]
  - $\mathcal{M}_{\mathrm{1-FAIL}}$ achieves strong sublinear regret as an algorithm for large $K(T)$

Improving equilibrium regret: A modified UCB mechanism

- $\mathcal{M}_{\mathrm{UCB-MOD}}$: Run UCB on random subset of $\min\{\lfloor\sqrt{T}\rfloor, k(T)\}$ arms
  - Exploring random subset: $\mathcal{M}_{\mathrm{1-FAIL}}$ [Berry et al'97]
  - $\mathcal{M}_{\mathrm{1-FAIL}}$ achieves strong sublinear regret as an algorithm for large $K(T)$, but *not* as a mechanism

- $\mathcal{M}_{\mathrm{UCB-MOD}}$: Run UCB on random subset of $\min\{\lfloor\sqrt{T}\rfloor, k(T)\}$ arms
  - Exploring random subset: $\mathcal{M}_{\mathrm{1-FAIL}}$ [Berry et al'97]
  - $\mathcal{M}_{\mathrm{1-FAIL}}$ achieves strong sublinear regret as an algorithm for large $K(T)$, but *not* as a mechanism

Theorem

$\mathcal{M}_{\mathrm{UCB-MOD}}$ *achieves strong sublinear equilibrium regret for all* $\gamma \leq 1$ *and cost functions* c*, for all* $K(T) \leq T$.

Why UCB works.

Extensions of result

- $\mathcal{M}_{\mathrm{UCB-MOD}}$ retains strong sublinear equilibrium regret if:
  - Each viewer is shown multiple contributions
  - Explore $\min\{G(T), k(T)\}$ arms: $G(T) \to \infty$, $G(T) = o(\frac{T}{\ln T})$
  - Heterogenous types: Cost functions $c_\tau(q)$
  - $q \in [\delta, \gamma]$, $\delta > 0$

Open questions

- Multi-armed bandits with *endogenous arms*: Strong sublinear equilibrium regret achievable with modified-UCB mechanism
- Many unanswered questions: Models, mechanisms
  - Probabilistic feedback
  - Sequential contributions
  - Quality-participation tradeoffs with $G(T)$

Open questions

- Multi-armed bandits with *endogenous arms*: Strong sublinear equilibrium regret achievable with modified-UCB mechanism
- Many unanswered questions: Models, mechanisms
  - Probabilistic feedback
  - Sequential contributions
  - Quality-participation tradeoffs with $G(T)$
  - What learning algorithms make good mechanisms when arms are endogenous?

- Crowdsourced *evaluation*: Replace expert by aggregated evaluation from 'crowd'
    - Image classification & labeling; content rating; abuse detection; MOOCs peer grading, ...
- How to *aggregate* evaluations from crowd?
    - Workers have different proficiencies;

- Crowdsourced *evaluation*: Replace expert by aggregated evaluation from 'crowd'
  - Image classification & labeling; content rating; abuse detection; MOOCs peer grading, ...
- How to *aggregate* evaluations from crowd?
  - Workers have different proficiencies; possibly unknown to system:

Incentives in crowdsourcing: Unobservable output

- Crowdsourced *evaluation*: Replace expert by aggregated evaluation from 'crowd'
  - Image classification & labeling; content rating; abuse detection; MOOCs peer grading, . . .
- How to *aggregate* evaluations from crowd?
  - Workers have different proficiencies; possibly unknown to system: Learn, weight to maximize accuracy

- Crowdsourced *evaluation*: Replace expert by aggregated evaluation from ‘crowd’
  - Image classification & labeling; content rating; abuse detection; MOOCs peer grading, . . .
- How to *aggregate* evaluations from crowd?
  - Workers have different proficiencies; possibly unknown to system: Learn, weight to maximize accuracy
- Input to aggregation problem comes from self-interested agents
- How to **incentivize** good evaluations from crowd?

- Incentivizing accurate evaluations, truthful reporting:
  - (i) Unobservable ground truth (ii) Effort-dependent accuracy
    (Information elicitation with *endogenous* proficiency)
  - Direct monitoring infeasible: Reward ‘agreement’

- Incentivizing accurate evaluations, truthful reporting:
  - (i) Unobservable ground truth (ii) Effort-dependent accuracy
    (Information elicitation with *endogenous* proficiency)
  - Direct monitoring infeasible: Reward ‘agreement’
  - Problem: Undesirable low-effort/second-guessing equilibria
    (e.g. always say ‘H’)

Incentives in crowdsourced evaluation

- Incentivizing accurate evaluations, truthful reporting:
  - (i) Unobservable ground truth (ii) Effort-dependent accuracy (Information elicitation with *endogenous* proficiency)
  - Direct monitoring infeasible: Reward 'agreement'
  - Problem: Undesirable low-effort/second-guessing equilibria (e.g. always say 'H')

- Mechanism [Dasgupta-Ghosh, WWW'13]: Maximum effort-truthful reporting is **highest-payoff** equilibrium!
  - (Assuming no task-specific collusions)
  - Use multiple tasks and ratings: Reward for agreement,

Incentives in crowdsourced evaluation

- Incentivizing accurate evaluations, truthful reporting:
  - (i) Unobservable ground truth (ii) Effort-dependent accuracy (Information elicitation with *endogenous* proficiency)
  - Direct monitoring infeasible: Reward 'agreement'
  - Problem: Undesirable low-effort/second-guessing equilibria (e.g. always say 'H')
- Mechanism [Dasgupta-Ghosh, WWW'13]: Maximum effort-truthful reporting is **highest-payoff** equilibrium!
  - (Assuming no task-specific collusions)
  - Use multiple tasks and ratings: Reward for agreement, but also identify and penalize *blind agreement*

Moving beyond single tasks: Incentivizing overall contribution

- Problems so far: Incentives for *single* contribution/task

Moving beyond single tasks: Incentivizing overall contribution

- Problems so far: Incentives for *single* contribution/task
- Rewarding contributors for *overall* identity:
  - Badges, leaderboards, reputations, . . .
  - Virtual rewards for cumulative contribution
- Gamification rewards valued by agents; contribution to earn reward is costly
- Badges induce *mechanisms*!
  - *Design* affects participation, effort from users

Moving beyond single tasks: Incentivizing overall contribution

- Problems so far: Incentives for *single* contribution/task
- Rewarding contributors for *overall* identity:
  - Badges, leaderboards, reputations, ...
  - Virtual rewards for cumulative contribution
- Gamification rewards valued by agents; contribution to earn reward is costly
- Badges induce *mechanisms*!
  - *Design* affects participation, effort from users

- Different badge designs online:
  - Absolute 'milestone' badges (StackOverflow, Foursquare), versus competitive 'top-contributor' badges (Y!Answers, Tripadvisor)
  - Information about badge winners (StackOverflow vs Y! Answers)

Badges and incentive design

- Different badge designs online:
  - Absolute 'milestone' badges (StackOverflow, Foursquare), versus competitive 'top-contributor' badges (Y!Answers, Tripadvisor)
  - Information about badge winners (StackOverflow vs Y! Answers)

- What *incentives* do different badge designs create for participation and effort?
  - Game-theoretic analysis of badge design (Easley & Ghosh, ACM EC'13)
  - *'Absolute'* or *'competitive'* badges?
  - 'Competitive' badges: Fixed *number* or *fraction* of participants?
  - Visibility of information: Transparent or not?

Results

Wrapping up

- Incentive design for crowdsourcing
  - Data, inputs to algorithms come from self-interested agents
  - Mechanisms rather than algorithms: A game-theoretic framework

- Incentive design for crowdsourcing
  - Data, inputs to algorithms come from self-interested agents
  - Mechanisms rather than algorithms: A game-theoretic framework
- Lots more to understand!
  - Sequential decision making
  - Eliciting effort with strategic contributors and raters

Wrapping up

- Incentive design for crowdsourcing
  - Data, inputs to algorithms come from self-interested agents
  - Mechanisms rather than algorithms: A game-theoretic framework
- Lots more to understand!
  - Sequential decision making
  - Eliciting effort with strategic contributors and raters
  - Overall contributor rewards; *sustaining* contribution
  - Learning and incentives: Designing reputations

Wrapping up

- Incentive design for crowdsourcing
  - Data, inputs to algorithms come from self-interested agents
  - Mechanisms rather than algorithms: A game-theoretic framework

- Lots more to understand!
  - Sequential decision making
  - Eliciting effort with strategic contributors and raters
  - Overall contributor rewards; *sustaining* contribution
  - Learning and incentives: Designing reputations
  - Experimental and empirical: What do agents value, and how?

Why $\mathcal{M}_{\mathrm{UCB-MOD}}$ works

Lemma

Any arm with quality $q_i \leq q_{max}(T) - \delta$ receives $\Theta(\ln T)$ attention in expectation for all $\delta > 0$

- $q_{max}(T)$: *Highest-quality explored contribution*

- A purely algorithmic statement; proof by contradiction

Theorem

For any fixed $q^* < \gamma$, the probability that there is some agent explored by $\mathcal{M}_{\mathrm{UCB-MOD}}$ who chooses quality $q > q^*$ goes to 1 as $T \to \infty$ in every equilibrium of $\mathcal{M}_{\mathrm{UCB-MOD}}$.

- Proof by contradiction: Demonstrate profitable deviation (Involves strategic reasoning, not purely algorithmic)

Back to UCB

Badges and incentive design: An economic framework

(Easley & Ghosh, ACM EC'13)

- Design recommendations from analysis:
  - Competitive badges: Reward fixed *number*, not fraction of competitors
  - Absolute versus competitive badges 'equivalent' if population parameters known
  - With uncertainty, or unknown parameters, competitive badges more 'robust'
  - Sharing information about other users' performance: Depends on convexity of value as function of winners

Conclusion