A Quality Assuring Multi-armed Bandit Mechanism for Crowdsourcing

Shweta Jain
Joint Work With: Satyanath Bhat, Sujit Gujar, Onno Zoeter and Prof. Y. Narahari

Dept. of CSA, IISc

August 3, 2015

NMI Workshop on Static and Dynamic Mechanism Design

Mechanism Design (MD)

Given: A set of utility maximizing (strategic) agents with private information and a social choice function that captures desirable (social) goals.

MD provides a game theoretic setting to explore if the given social choice function can be implemented as an equilibrium outcome of an induced game.

Example: Vickrey Auction[1]

[1] Y.Narahari: Game Theory and Mechanism Design. IISc Press and WSPC,2014

Multi-Armed Bandit (MAB) Problems

Given: A set of arms (agents) with unknown reward distributions.

MAB solution provides allocation policies so as to learn these distributions efficiently through intelligent exploration.

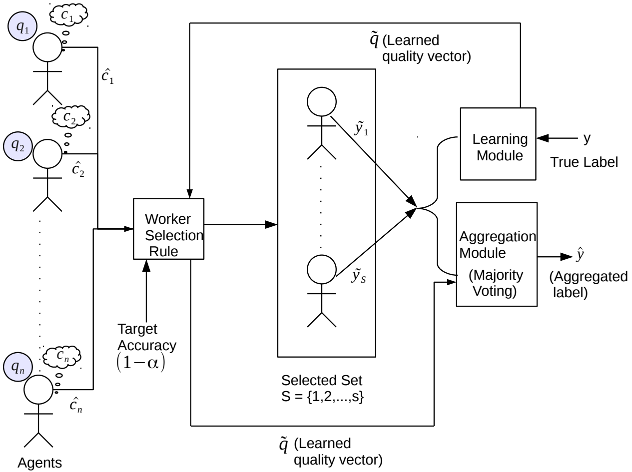
Example: UCB algorithm[2]

[2]Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. Machine Learning, 2002.

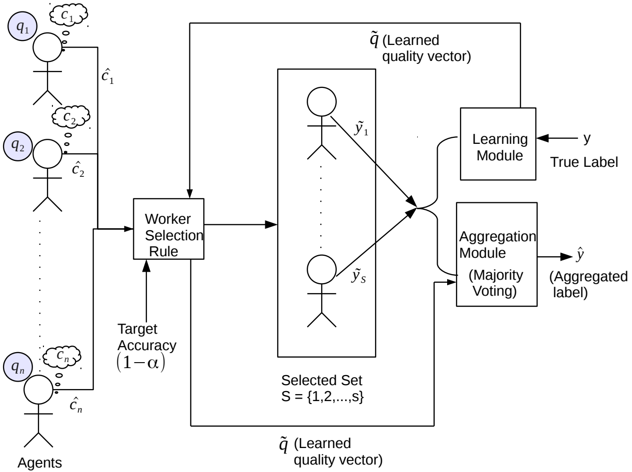
Multi-Armed Bandit Mechanisms

- Modern problems involve strategic agents with private information and unknown information.
- Examples:
  - Sponsored Search Auctions on the Web
  - Crowdsourcing
  - Online Auctions/Internet Markets
  - etc.
- In MAB mechanisms, we seek to learn certain parameters while eliciting private information truthfully.
- MAB and MD are extremely well investigated as individual problems. Interesting research questions arise when you try to meld them.

Our Setting (Assured Accuracy Bandit)



Our Setting (Assured Accuracy Bandit)



$P(\hat{y} \neq y) < \alpha$ (Accuracy Constraint)

The Optimization Problem (to be solved for each task)

- Suppose $q = (q_1, q_2, \ldots, q_n)$ is a vector of qualities of workers
- Let $f_S(q)$ represent some measure of error probability
- $1 - f_S(q)$ represents the accuracy with quality profile q
- For each task, $t = 1, 2, \ldots, T$, we wish to solve:

$$\min_{S^t \subseteq N} \sum_{i \in S^t} c_i \quad (1)$$

$$\text{s.t. } f_{S^t}(q) < \alpha \quad \text{(Accuracy Constraint)} \quad (2)$$

We assume function $f_S(q)$ satisfies Monotonicity and Bounded Smoothness properties.

Monotonicity and Bounded Smoothness

- *Monotonicity:* $f_S(q)$ is monotone if for all quality profiles q and q' such that $\forall i \in \mathcal{N},\ q'_i \leq q_i,$ we have,

$$f_S(q') < \alpha \implies f_S(q) < \alpha \ \ \forall S \subseteq \mathcal{N},\ \forall \alpha \in [0,1]$$

Monotonicity and Bounded Smoothness

- *Monotonicity:* $f_S(q)$ is monotone if for all quality profiles q and q' such that $\forall i \in \mathcal{N},\ q'_i \leq q_i,$ we have,

$$f_S(q') < \alpha \implies f_S(q) < \alpha \ \ \forall S \subseteq \mathcal{N},\ \forall \alpha \in [0,1]$$

- *Bounded smoothness:* $f_S(q)$ satisfies bounded smoothness property if there exists a monotone continuous function h such that if

$$\max_i |q_i - q'_i| \leq \delta \implies |f_S(q) - f_S(q')| \leq h(\delta) \ \ \forall S \subseteq \mathcal{N},$$

$$\forall q, q' \in [0.5, 1]$$

Regret Formulation

- The regret of an algorithm in MAB mechanisms is the difference between the cost the algorithm incurs and cost incurred by an optimal algorithm with known qualities.
- We assume there is a penalty L for not satisfying the constraint.
- Thus, the regret is given as:

$$\mathcal{R}(\mathcal{A}) = (1-\mu)(C(\mathcal{A}) - C(S^*)) + \mu L$$

- Penalty term L is typically high.

Lower Bounds on Regret in AAB Framework

Definition (Δ-Separated Property:)

q is Δ-*Separated* with respect to α if $\exists \Delta > 0$ such that, $\Delta = \inf_{S \subseteq \mathcal{N}} \; |f_S(q) - \alpha|$ i.e. $\forall S, \; f_S(q) \notin [\alpha - \Delta, \alpha + \Delta]$.

Theorem

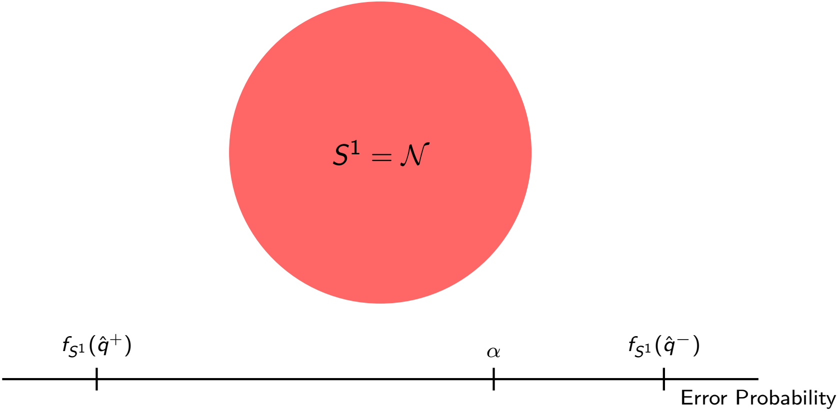
Any algorithm in AAB framework and satisfies $E[n_S(\mathcal{A})] = o(T^a)$ $\forall a > 0$ for any subset of worker S which is not optimal. Then, the following holds:

$$\liminf_{T \to \infty} \mathbb{E}[\mathcal{R}(\mathcal{A})] \geq \frac{\ln T}{(h^{-1}(\Delta))^2},$$

where, $\Delta = \inf_{S \subseteq \mathcal{N}} \; |f_S(q) - \alpha|$ and $h(.)$ is the bounded smooth function.

CCB-NS Algorithm

- Maintain upper confidence bound $\hat{q}^+$ and lower confidence bound $\hat{q}^-$ on qualities. Initialize: $\hat{q}_i^+ = 1$ and $\hat{q}_i^- = 0.5$, forall i.

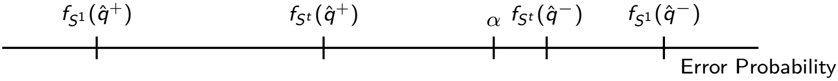


CCB-NS Algorithm

- For the task t, $\hat{q}_i^+ = \hat{q}_i + \sqrt{\frac{1}{2n_{i,t-1}} ln(\frac{2}{\mu})}$, $\hat{q}_i^- = \hat{q}_i - \sqrt{\frac{1}{2n_{i,t-1}} ln(\frac{2}{\mu})}$.

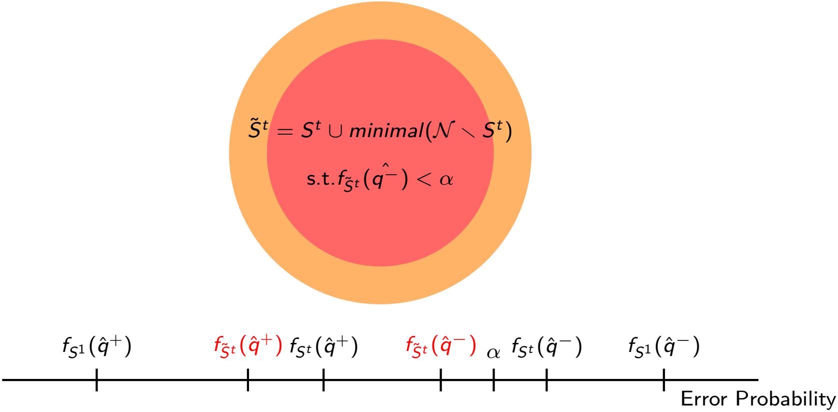
$$S^t = \arg\min_{S \subseteq \mathcal{N}} \sum_{i \in S} c_i$$

$$\text{s.t. } f_S(\hat{q}^+) < \alpha$$



CCB-NS Algorithm

- For the task t, $\hat{q}_i^+ = \hat{q}_i + \sqrt{\frac{1}{2n_{i,t-1}} ln(\frac{2}{\mu})}$, $\hat{q}_i^- = \hat{q}_i - \sqrt{\frac{1}{2n_{i,t-1}} ln(\frac{2}{\mu})}$.

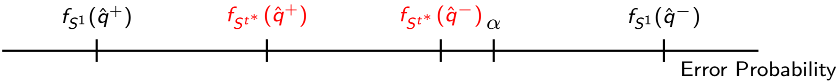


CCB-NS Algorithm

- If for task t^*:

$$S^{t^*} = \arg\min_{S \subseteq \mathcal{N}} \sum_{i \in S} c_i$$

$$\text{s.t. } f_S(\hat{q}^+) < \alpha$$



- Return S^{t^*} for all the remaining tasks.

Properties of CCB-NS

- CCB-NS is an adaptive exploration separated learning algorithm

Theorem

CCB-NS satisfies the accuracy constraint with probability at least $(1-\mu)$ at every round t

Lemma

Set S^{t^*} returned by the CCB-NS algorithm is an optimal set with probability at least $1-\mu$. That is, $C(S^{t^*}) = C(S^*)$ w.p. $(1-\mu)$

Can be proved using monotonicity properties of error probability function

Properties of CCB-NS

- Let $\Delta = \min_{S \subseteq \mathcal{N}} |f_S(.) - \alpha|$
  (minimum difference between error tolerance α and error probability value of any set)

Theorem

The number of exploration rounds by the CCB-NS algorithm is bounded by $\frac{2n}{(h^{-1}(\Delta))^2} ln(\frac{2n}{\mu})$ *with probability* $(1 - \mu)$

where h is the bounded smoothness function

Strategic Version

- Costs c_i are private information of workers
- Valuations: $v_i = -c_i, \forall i \in \mathcal{N}$
- Utilities (Quasilinear):
$$\sum_{t=1}^{T} u_i(\tilde{q}(t), \hat{c}, c_i) = -c_i \sum_{t=1}^{T} \mathcal{A}_i^t(\tilde{q}(t), \hat{c}) + \mathcal{P}_i^t(\tilde{q}(t), \hat{c})$$
- $\mathcal{A}_i^t$ represents whether the t^{th} task is allocated to worker i
- $\mathcal{P}_i^t$ is the monetary transfer to worker i for task t

Some Definitions

- Success Realization: A success realization is a matrix ρ s.t.,

$$\rho_{it} = \begin{cases} 1 \text{ If } \tilde{y}_i^t = y_t \\ 0 \text{ if } \tilde{y}_i^t \neq y_t \\ -1 \text{ if worker } i \text{ is not selected for task } t \end{cases}$$

- Ex-post Monotone Allocation: An allocation rule $\mathcal{A}$ is ex-post monotone if $\forall \rho \in \{0, 1, -1\}^{n \times T}, \ \ \forall i \in \mathcal{N}, \ \ \forall \hat{c}_{-i} \in [0, 1]^{n-1}$,

$$\hat{c}_i \leq \hat{c}_i' \Rightarrow \mathcal{A}_i(\hat{c}_i, \hat{c}_{-i}; \rho) \geq \mathcal{A}_i(\hat{c}_i', \hat{c}_{-i}; \rho)$$

  $\mathcal{A}_i(\hat{c}_i, \hat{c}_{-i})$ is the number of tasks assigned to worker i with bids $\hat{c}_i$ and $\hat{c}_{-i}$

- Ex-post Truthful Mechanism: A mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{P})$ is ex-post truthful if
  $\forall \rho \in \{0, 1, -1\}^{n \times T}, \ \ \forall i \in \mathcal{N}, \ \ \forall \hat{c}_{-i} \in [0, 1]^{n-1}$,

$$-c_i \mathcal{A}_i(c_i, \hat{c}_{-i}; \rho) + \mathcal{P}_i(c_i, \hat{c}_{-i}; \rho) \geq -c_i \mathcal{A}_i(\hat{c}_i, \hat{c}_{-i}; \rho) + \mathcal{P}_i(\hat{c}_i, \hat{c}_{-i}; \rho) \ \forall \hat{c}_i \in [0$$

An Ex-post Monotone Allocation Algorithm, CCB-S

Input: Task error tolerance α, confidence level μ, tasks $\{1, 2, \ldots, T\}$, workers $\mathcal{N}$, costs c
Output: Worker selection set S^t, Label $\hat{y}_t$ for task t
Initialization: $\forall i$, $\hat{q}_i^+ = 1$, $\hat{q}_i^- = 0.5$, $k_{i,1} = 0$, $S^1 = \mathcal{N}$, and $\hat{y}_1 = \text{AGGREGATE}(\tilde{y}(S^1))$
Observe true label y_1
$\forall i \in \mathcal{N}$, $n_{i,1} = 1$, $k_{i,1} = 1$ if $\tilde{y}_i = y_1$ and $\hat{q}_i = k_{i,1}/n_{i,1}$
for $t = 2$ to T

Let $S^t = \underset{S \subseteq \mathcal{N}}{\arg\min} \sum_{i \in S} c_i$ s.t. $f_S(\hat{q}^+) < \alpha$

% Explore

if $f_{S^t}(\hat{q}^-) > \alpha$ **then**

$S^t = \mathcal{N}$

$\hat{y}_t = \text{AGGREGATE}(\tilde{y}(S^t))$

Observe true label y_t; $\forall i \in S^t$: $n_{i,t} = n_{i,t} + 1$, $k_{i,t} = k_{i,t} + 1$ if $\tilde{y}_i = y_t$, $\hat{q}_i = k_{i,t}/n_{i,t}$,

$\hat{q}_i^+ = \hat{q}_i + \sqrt{\frac{1}{2n_{i,t}} ln(\frac{2}{\mu})}$, $\hat{q}_i^- = \hat{q}_i - \sqrt{\frac{1}{2n_{i,t}} ln(\frac{2}{\mu})}$

else

$t^* = t$, $\hat{y}_t = \text{AGGREGATE}(\tilde{y}(S^t))$

Break

%Exploit

for $t = t^* + 1$ to T

$S^t = S^{t^*}$, $\hat{y}_t = \text{AGGREGATE}(\tilde{y}(S^t))$

Properties of CCB-S

Theorem

Number of exploration rounds by the CCB-S algorithm is bounded by $\frac{2}{(h^{-1}(\Delta))^2} ln(\frac{2n}{\mu})$ *with probability* $(1-\mu)$

Theorem

Allocation rule given by the CCB-S algorithm ($\mathcal{A}^{CCB-S}$) *is ex-post monotone and thus produces an ex-post incentive compatible and ex-post individual rational mechanism*

Properties of CCB-S

Theorem

Number of exploration rounds by the CCB-S algorithm is bounded by $\frac{2}{(h^{-1}(\Delta))^2} ln(\frac{2n}{\mu})$ *with probability* $(1-\mu)$

Theorem

Allocation rule given by the CCB-S algorithm ($\mathcal{A}^{CCB-S}$) *is ex-post monotone and thus produces an ex-post incentive compatible and ex-post individual rational mechanism*

Note: CCB-S algorithm only provides an allocation rule. For the payment rule we use transformation given by Babaioff et. al. as a black box.

Properties of CCB-S (continued)

Theorem

Allocation rule given by the CCB-S algorithm $\mathcal{A}^{CCB-S}$ is ex-post monotone.

Proof:

We need to prove:

$$\mathcal{A}_i^t(\hat{c}_i, c_{-i}; \rho) \leq \mathcal{A}_i^t(c_i, c_{-i}; \rho)$$
$$\forall \rho \forall i \in \mathcal{N}, \ \forall t \in \{1, 2, \ldots, T\}, \ \forall \hat{c}_i \geq c_i$$

For notation convenience, assume ρ is fixed and denote $\mathcal{A}_i^t(\hat{c}_i, c_{-i}; \rho)$ as $\mathcal{A}_i^t(\hat{c}_i, c_{-i})$

Proof Continued

- Prove by induction:

Proof Continued

- Prove by induction:
- $\mathcal{A}_j^1(\hat{c}_i, c_{-i}) = \mathcal{A}_j^1(c_i, c_{-i}) = 1 \ \forall j$ (since task 1 is given to all workers)

Proof Continued

- Prove by induction:
- $\mathcal{A}_j^1(\hat{c}_i, c_{-i}) = \mathcal{A}_j^1(c_i, c_{-i}) = 1 \; \forall j$ (since task 1 is given to all workers)
- Let t be the largest time step such that, $\forall j$, $\mathcal{A}_j^{t-1}(\hat{c}_i, c_{-i}) = \mathcal{A}_j^{t-1}(c_i, c_{-i}) = t - 1$ (Exploration round with $\hat{c}_i$ and c_i)

Proof Continued

- Prove by induction:
- $\mathcal{A}_j^1(\hat{c}_i, c_{-i}) = \mathcal{A}_j^1(c_i, c_{-i}) = 1 \ \forall j$ (since task 1 is given to all workers)
- Let t be the largest time step such that, $\forall j$, $\mathcal{A}_j^{t-1}(\hat{c}_i, c_{-i}) = \mathcal{A}_j^{t-1}(c_i, c_{-i}) = t - 1$ (Exploration round with $\hat{c}_i$ and c_i)
- And $\exists i$ such that,

$$\mathcal{A}_i^t(\hat{c}_i, c_{-i}) \neq \mathcal{A}_i^t(c_i, c_{-i})$$

Proof Continued

- Prove by induction:
- $\mathcal{A}_j^1(\hat{c}_i, c_{-i}) = \mathcal{A}_j^1(c_i, c_{-i}) = 1 \ \forall j$ (since task 1 is given to all workers)
- Let t be the largest time step such that, $\forall j$, $\mathcal{A}_j^{t-1}(\hat{c}_i, c_{-i}) = \mathcal{A}_j^{t-1}(c_i, c_{-i}) = t - 1$ (Exploration round with $\hat{c}_i$ and c_i)
- And $\exists i$ such that,

$$\mathcal{A}_i^t(\hat{c}_i, c_{-i}) \neq \mathcal{A}_i^t(c_i, c_{-i})$$

- Since the costs and quality estimates are the same for all the workers till tasks t, this can happen only when in one case worker i is selected, while in the other case worker i is not selected

Proof Continued

- Prove by induction:
- $\mathcal{A}_j^1(\hat{c}_i, c_{-i}) = \mathcal{A}_j^1(c_i, c_{-i}) = 1 \ \forall j$ (since task 1 is given to all workers)
- Let t be the largest time step such that, $\forall j$, $\mathcal{A}_j^{t-1}(\hat{c}_i, c_{-i}) = \mathcal{A}_j^{t-1}(c_i, c_{-i}) = t - 1$ (Exploration round with $\hat{c}_i$ and c_i)
- And $\exists i$ such that,

$$\mathcal{A}_i^t(\hat{c}_i, c_{-i}) \neq \mathcal{A}_i^t(c_i, c_{-i})$$

- Since the costs and quality estimates are the same for all the workers till tasks t, this can happen only when in one case worker i is selected, while in the other case worker i is not selected
- Let the two sets selected with c_i and $\hat{c}_i$ be $S(c_i)$ and $S(\hat{c}_i)$ respectively

Proof Continued

- Since the optimization problem involves cost minimization and quality updates are the same, we have,

$$\mathcal{A}_i^t(\hat{c}_i, c_{-i}) = t - 1 \text{ which implies } i \notin S(\hat{c}_i)$$
$$\mathcal{A}_i^t(c_i, c_{-i}) = t \text{ which implies } i \in S(c_i)$$

Proof Continued

- Since the optimization problem involves cost minimization and quality updates are the same, we have,

$$\mathcal{A}_i^t(\hat{c}_i, c_{-i}) = t - 1 \text{ which implies } i \notin S(\hat{c}_i)$$
$$\mathcal{A}_i^t(c_i, c_{-i}) = t \text{ which implies } i \in S(c_i)$$

- Since $i \notin S(\hat{c}_i)$, selected set $S(\hat{c}_i)$ satisfies the lower confidence bound too (exploitation round with bid $\hat{c}_i$)

Proof Continued

- Since the optimization problem involves cost minimization and quality updates are the same, we have,

$$\mathcal{A}_i^t(\hat{c}_i, c_{-i}) = t - 1 \text{ which implies } i \notin S(\hat{c}_i)$$
$$\mathcal{A}_i^t(c_i, c_{-i}) = t \text{ which implies } i \in S(c_i)$$

- Since $i \notin S(\hat{c}_i)$, selected set $S(\hat{c}_i)$ satisfies the lower confidence bound too (exploitation round with bid $\hat{c}_i$)
- Thus for the rest of the tasks, only $S(\hat{c}_i)$ is selected and thus we have, $\mathcal{A}_i^t(\hat{c}_i, c_{-i}) \leq \mathcal{A}_i^t(c_i, c_{-i})$

Minimum Knapsack Problem

- One example of optimization problem in assured accuracy framework:

$$\min_{S \in \mathcal{N}} C(S)$$
$$\text{s.t.} \sum_{i \in S} (2q_i - 1) \geq 6 \ln\left(\frac{1}{\alpha}\right)$$

- This can be formulated as minimum knapsack problem
- There exists a greedy algorithm to solve the above problem in polynomial time
- We have also extended the greedy algorithm to the monotone learning algorithm when the qualities are not known and costs are strategic
- The algorithm runs in polynomial time and does not select all the workers in the exploration phase

Directions for Future Work

- Non exploration-separated algorithms satisfying desirable mechanism properties with lower regret for the general optimization problem
- Extension to more general task settings
- Working with soft constraint formulation

Thank You