Crowd-Learning: Improving the Quality of Crowdsourcing Using Sequential Learning

Mingyan Liu
(Joint work with Yang Liu)

Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI

March 2015

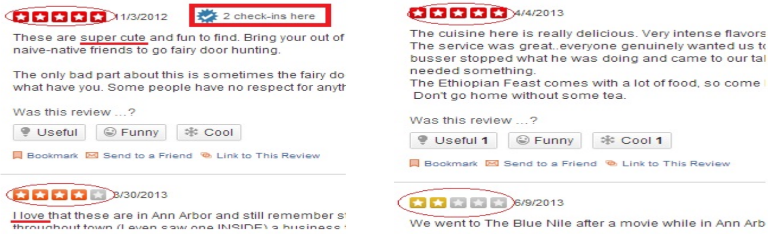
The power of crowdsourcing

Tapping into enormous resources in sensing and processing:

- Data collection: participatory sensing, user-generated map
- Data processing: image labeling, annotation
- Recommendation: rating of movies, news, restaurants, services
- Social studies: opinion survey, the science of opinion survey

Scenario I: recommender systems

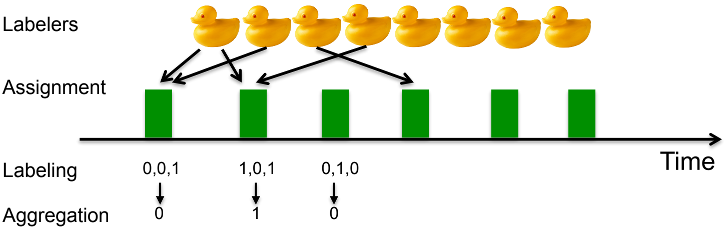
E.g., Yelp, movie reviews, news feed



- A user shares experience and opinion
- Measure of quality subjective: not all ratings should be valued equally

Scenario II: crowdsourcing markets

E.g., using AMTs



- Paid workers perform computational tasks.
- Measure of quality objective but hard to evaluate: competence, bias, irresponsible behavior, etc.

Our objective

To make the most effective use of the crowdsourcing system

- Cost in having large amount of data labeled is non-trivial
- There may also be time constraint

A sequential/online learning framework

- Over time learn which labelers are more competent, or whose reviews/opinion should be valued more.
- Closed-loop, causal.

Multiarmed bandit (MAB) problems

A sequential decision and learning framework:

- Objective: select the best of a set of choices ("arms")
- Principle: repeated sampling of different choices ("exploration"), while controlling how often each choice is used based on their empirical quality ("exploitation").
- Performance measure: "regret" – difference between an algorithm and a benchmark.

Challenge in crowdsourcing: ground truth

- True label of data remains unknown
- If view each labeler as a choice/arm: unknown quality of outcome ("reward").

Multiarmed bandit (MAB) problems

A sequential decision and learning framework:

- Objective: select the best of a set of choices ("arms")
- Principle: repeated sampling of different choices ("exploration"), while controlling how often each choice is used based on their empirical quality ("exploitation").
- Performance measure: "regret" – difference between an algorithm and a benchmark.

Challenge in crowdsourcing: ground truth

- True label of data remains unknown
- If view each labeler as a choice/arm: unknown quality of outcome ("reward").

Key ideas and features

Dealing with lack of ground truth:

- Recommender system: input from others is calibrated against one's own experience
- Labeler selection: mild assumption on the collective quality of the crowd; quality of an individual is estimated against the crowd.

Online and offline uses:

- Learning occurs as data/labeling tasks arrive.
- Can be equally used offline by processing data sequentially.

Performance measure

- Weak regret: comparing against optimal static selections.
- Will also compare with offline methods.

Outline of the talk

The recommender system problem

- Formulation and main results
- Experiments using MovieLens data

The labeler section problem

- Formulation and main results
- Experiments using a set of AMT data

Discussion and conclusion

Model

Users/Reviewers and options:

- M users/reviewers: $i, j \in \{1, 2, ..., M\}$.
- Each has access to N options $k, l \in \{1, 2, ..., N\}$.
- At each time step a user can choose up to K options: $a^i(t)$.

Rewards:

- An IID random reward $r^i_l(t)$, both user and option dependent.
- Mean reward (unknown to the user): $\mu^i_l \neq \mu^i_k, l \neq k, \forall i$, i.e., different options present distinct values to a user.

Performance measure

Weak regret:

$$R^{i,a}(T) = T \cdot \sum_{k \in N_K^i} \mu_k^i - \mathbb{E}[\sum_{t=1}^{T} \sum_{k \in a^i(t)} r_k^i(t)]$$

A user's optimal selection (reward-maximization): top-K set N_K^i.

- General goal is to achieve $R^{i,a}(T) = o(T)$.
- Existing approach can achieve log regret uniform in time.

Example: UCB1 [Auer et all 2002]

Single-play version; extendable to multiple-play

Initialization: for $t \leq N$, play arm/choice t, $t = t + 1$

While $t > N$

- for each choice k, calculate its sample mean:

$$\bar{r}_k^i(t) = \frac{r_k^i(1) + r_k^i(2) + ... + r_k^i(n_k^i(t))}{n_k^i(t)}$$

- its index:

$$g^i_{k,t,n_k^i(t)} = \bar{r}_k^i(t) + \sqrt{\frac{L \log t}{n_k^i(t)}}, \quad \forall k$$

- play the arm with the highest index; $t = t + 1$

Key observation

A user sees and utilizes its own samples in learning.

- Can we improve this by leveraging other users' experience?
- Second-hand learning in addition to first-hand learning.

Basic idea:

- Estimate the difference between two users.
- Use this to calibrate others' observations or decisions so that they could be used as one's own.

How to model information exchange

Full information exchange:

- Users share their decisions and subsequent rewards $(k, r_k^i(t))$.

Partial information exchange:

- Only share decisions on which options were used without revealing evaluation (k).

Full information exchange

how to measure pairwise difference

Estimated distortion:

$$\tilde{\delta}_k^{i,j}(t) = \frac{\sum_{s\leq t} \log r_k^i(s)/n_k^i(t)}{\sum_{s\leq t} \log r_k^j(s)/n_k^j(t)}.$$

Converted average reward from j:

$$\pi^{i,j}(\bar{r}_k^j(t)) = \sum_{s\leq t} (r_k^j(s))^{\tilde{\delta}_k^{i,j}(t)}/n_k^j(t)$$

An index algorithm

Original UCB1 index: $\bar{r}_k^i(t) + \sqrt{\frac{2\log t}{n_k^i(t)}}$

Modified index: choose K highest in this value

$$(\text{U\_full}): \quad \frac{\bar{r}_k^i(t) \cdot n_k^i(t) + \overbrace{\sum_{j \neq i} \pi^{i,j}(\bar{r}_k^j(t)) \cdot n_k^j(t)}^{\text{converted rewards}}}{\sum_j n_k^j(t)} + \sqrt{\frac{2\log t}{\sum_j n_k^j(t)}},$$

- Converted average reward from j:

$$\pi^{i,j}(\bar{r}_k^j(t)) = \sum_{s \leq t} (r_k^j(s))^{\tilde{\delta}_k^{i,j}(t)} / n_k^j(t)$$

Regret bound

Theorem

The weak regret of user i under U_full is upper bounded by

$$R^i_{U\_full}(T) \leq \sum_{k \in \overline{N}^i_K} \left\lceil \frac{4(\sqrt{2}+\kappa M^{\gamma})^2 \log T}{M \cdot \Delta^i_k} \right\rceil + const.$$

where $\Delta^i_k = \mu^i_K - \mu^i_k$, *assuming* $\min_j\{E[\log r^j_k] - \delta^{i,j}_k\} > 0$, *and* $r^i_k = (r^j_k)^{\delta^{i,j}_k}$.

Compared with UCB1: $R_{\text{ucb1}}(T) \leq \sum_{k \in \overline{N}_K} \lceil \frac{8 \log T}{\Delta_k} \rceil + \text{const.}$

- When M large roughly $\sqrt{M}$-fold improvement.

Regret bound

Theorem

The weak regret of user i under U_full is upper bounded by

$$R^i_{U\_full}(T) \leq \sum_{k \in \overline{N}^i_K} \left\lceil \frac{4(\sqrt{2} + \kappa M^\gamma)^2 \log T}{M \cdot \Delta^i_k} \right\rceil + const.$$

where $\Delta^i_k = \mu^i_K - \mu^i_k$, *assuming* $\min_j \{E[\log r^j_k] - \delta^{i,j}_k\} > 0$, *and* $r^i_k = (r^j_k)^{\delta^{i,j}_k}$.

Compared with UCB1: $R_{\text{ucb1}}(T) \leq \sum_{k \in \overline{N}_K} \lceil \frac{8 \log T}{\Delta_k} \rceil + \text{const.}$

- When M large roughly $\sqrt{M}$-fold improvement.

Partial information exchange

Only sees others' choices, not rewards

- Will further distinguish users by their *preference groups*.
- Within the same preference group users have the same preference ordering among all choices: $\mu_1^i > \mu_2^i > \cdots > \mu_N^i$ for all i in the group.

Uniform group preference

Keep track of sample frequency:

- Track $n_k(t) = \sum_i n_k^i(t)$; compute frequency

$$\beta_k(t) := \frac{n_k(t)}{\sum_l n_l(t)}$$

Modified index:

$$\text{(U\_part)}: \;\; \bar{r}_k^i(t) - \underbrace{\alpha(1 - \beta_k(t))\sqrt{\frac{\log t}{n_k^i(t)}}}_{\text{Group recommendation}} + \sqrt{\frac{2 \log t}{n_k^i(t)}}$$

- α: the weight given to others' choices.

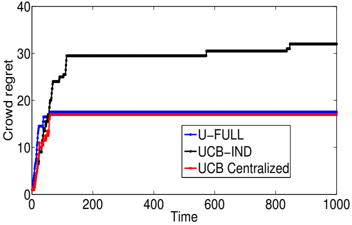
Non-uniform group preferences

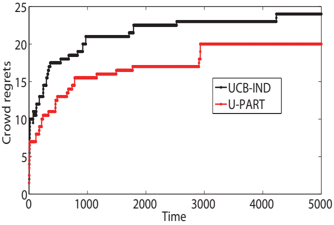
Additional technical hurdles:

- Assume known set of preferences but unknown group affiliation.
- Need to perform group identification
  - Keep track of the number of times user j chooses option k.
  - Estimate j's preference by ordering the sample frequency.
  - Place j in the group with best match in preference ordering.
  - Discount choices made by members of a different group.
- Similar results can be obtained.

Experiment I: $M = 10, N = 5, K = 3$

Uniform preference; rewards exp rv; distortion Gaussian





(L) comparing full information exchange with UCB1 applied individually, and applied centrally with known distortion.

(R) comparing partial information exchange with UCB1 applied individually.

Experiment II: MovieLens data

A good dataset though not ideal for our intended use

- Collected via a movie recommendation system
- We will use MovieLens-1M dataset: containing 1M rating records provided by 6040 users on 3952 movies from 18 genres, from April 25, 2000 to February 28, 2003.
- Each rating on a scale of 1-5.
- In general, each reviewer contributes to multiple reviews: $\sim$70% have more than 50 reviews.

Can we provide better recommendation?

- Predict how a user is going to rate movies given his and other users' reviews in the past.
- The decision aspect of the learning algorithm is not captured.

MovieLens: methodology

- Discrete time steps clocked by the review arrivals.
- Bundle movies into 18 genres (action, adventure, comedy, etc), each representing an option/arm:
  - ensure that each option remains available for each reviewer
  - lose the finer distinction between movies of the same genre
  - prediction is thus for a whole genre, used as a proxy for a specific movie within that genre.
- Use full information exchange index
  - we will only utilize users estimated to be in the same group (same preference ordering).
- Prediction performance measured by error and squared error averaged over the total number of reviews received by time t.

The recommendation algorithm

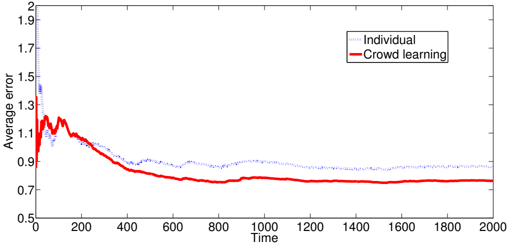
At time t, given i's review $r_k^i(t)$ for movie k:

- update i's preference ranking over options/genres;
- update i's similarity group: reviewers that share the same set of top K preferred options as i;
- estimate the distortion between i and those in its similarity group;
- update i's rating for each option by including rating from those in its similarity group corrected by the estimated distortion;
- repeat for all reviews arriving at time t.

At the end of step t, obtain estimated rating for all reviewers and all genres.

Algorithm used online

Prediction at each time step



- Prediction becomes more accurate with more past samples.
- Group learning outperforms individual learning.
- Downward trend not monotonic due to arrivals of new movies.

Algorithm used offline

Offline estimation result; comparison with the following

- SoCo, a social network and contextual information aided recommendation system. A random decision tree is adopted to partition the original user-item-rating matrix (user-movie-rating matrix in our context) so that items with similar contexts are grouped.
- RPMF, Random Partition Matrix Factorization, a contextual collaborative filtering method based on a tree constructed by using random partition techniques.
- MF, basic matrix factorization technique over the user-item matrix.

Algorithm	Crowd	Ind.	SoCo	RPMF	MF
Avg. Error	**0.6880**	0.8145	0.7066	0.7223	0.7668
RMSE	0.9054	1.0279	**0.8722**	0.8956	0.9374

Discussion

Combination of learning from direct and indirect experience

- Estimation of similarity groups and pairwise distortion effectively allows us to utilize a larger set of samples.

Our algorithm does not rely on exogenous social or contextual information

- However, the estimation of similarity groups introduces a type of social connectivity among users.

In settings where it's unclear whether preferences are uniform or non-uniform:

- can simply assume it to be the latter and do as we did in the MovieLens experiment.

Outline of the talk

The recommender system problem

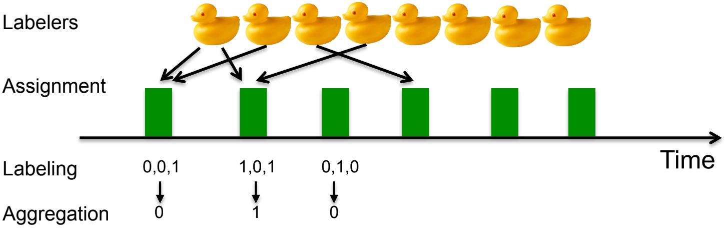
- Formulation and main results
- Experiments using MovieLens data

The labeler section problem

- Formulation and main results
- Experiments using a set of AMT data

Discussion and conclusion

Labeler selection



- M labelers; labelers i has accuracy p_i (can be task-dependent).
  - No two exactly the same: $p_i \neq p_j$ for $i \neq j$, and $0 < p_i < 1$, $\forall i$.
  - Collective quality: $\bar{p} := \sum_i p_i / M > 1/2$.
- Unlabeled tasks arrive at $t = 1, 2, \cdots$.
  - User selects a subset S_t of labelers for task at t.
  - Labeling payment of c_i for each task performed by labeler i.

Labeling outcome/Information aggregation

Aggregating results from multiple labelers:

- A task receives a set of labels: $\{L_i(t)\}_{i \in S_t}$.
  - Use simple majority voting or weighted majority voting to compute the label output: $L^*(t)$.
- Probability of correct labeling outcome: $\pi(S_t)$; well defined function of p_is.
  - Optimal set of labelers: S^* that maximizes $\pi(S)$.

Accuracy of labeling outcome:

- Probability that a simple majority vote over all M labelers is correct: $a_{\min} := P(\sum_i X_i/M > 1/2)$.
  - If $\bar{p} > 1/2$ and $M > \frac{\log 2}{\bar{p}-1/2}$, then $a_{\min} > 1/2$.

Obtaining S^*

Assuming we know $\{p_i\}$, S^* can be obtained using a simple linear search

Theorem

Under the simple majority voting rule, $|S^*|$ is an odd number. Furthermore, S^* is monotonic: if $i \in S^*$ and $j \notin S^*$, then we must have $p_i > p_j$.

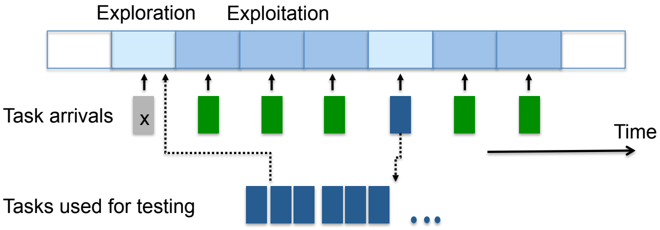
Obtaining S^*

Assuming we know $\{p_i\}$, S^* can be obtained using a simple linear search

Theorem

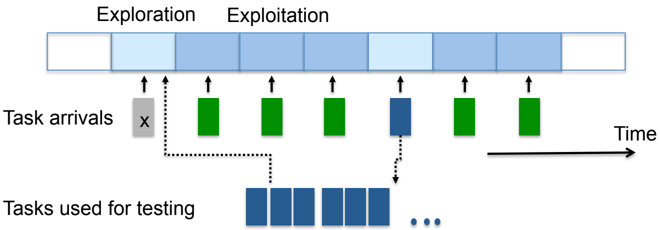
Under the simple majority voting rule, $|S^*|$ is an odd number. Furthermore, S^* is monotonic: if $i \in S^*$ and $j \notin S^*$, then we must have $p_i > p_j$.

An online learning algorithm



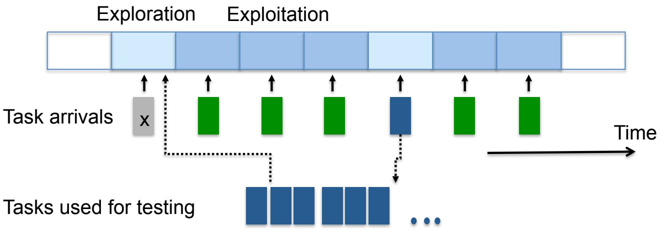
There is a set of tasks $E(t)$ ($\sim \log t$) used for *testing* purposes.

- These or their independent and identical variants are repeatedly assigned to the labelers ($\sim \log t$).



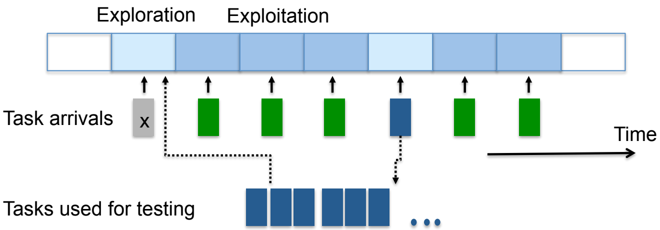
Two types of time steps:

- Exploration: all M labelers are used. Exploration is entered if (1) the number of testers falls below a threshold ($\sim \log t$), or if (2) the number of times a tester has been tested falls below a threshold ($\sim \log t$).
- Exploitation: the estimated $\tilde{S}^*$ is used to label the arriving task based on the current estimated $\{\tilde{p}_i\}$.



Three types of tasks:

- Testers: those arriving to find (1) true and (2) false. These are added to $E(t)$ and are repeatedly used to collect independent labels whenever (2) is true subsequently.
- Throw-aways: those arriving to find (2) true. These are given a random label.
- Keepers: those arriving to find both (1) and (2) false. These are given a label outcome using the best estimated set of labelers.



Accuracy update

- Estimated label on tester k at time t: majority label over all test outcomes up to time t.
- $\tilde{p}_i$ at time t: the % of times i's label matches the majority vote known at t out of all tests on all testers.

Regret

Comparing with the optimal selection (static):

$$R(T) \quad = \quad T\pi(S^*) - E[\sum_{t=1}^{T} \pi(S_t)]$$

Main result:

$$R(T) \quad \leq \quad \text{Const}(S^*, \Delta_{\max}, \Delta_{\min}, \delta_{\max}, \delta_{\min}, a_{\min}) \log^2(T) + \text{Const}$$

- $\Delta_{\max} = \max_{S \neq S^*} \pi(S^*) - \pi(S)$, $\delta_{\max} = \max_{i \neq j} |p_i - p_j|$.
- First term due to exploration; second due to exploitation.
- Can obtain similar result on the cost $C(T)$.

Discussion

Relaxing some assumptions

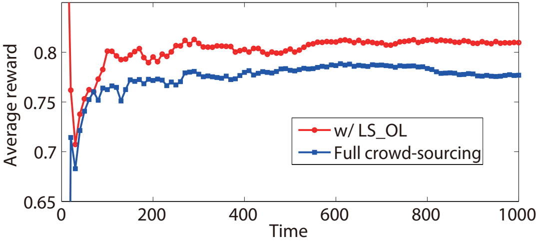
- Re-assignment of the testers after random delay
- Improve the bound by improving $a_{\min}$: weed out bad labelers.

Weighted majority voting rule

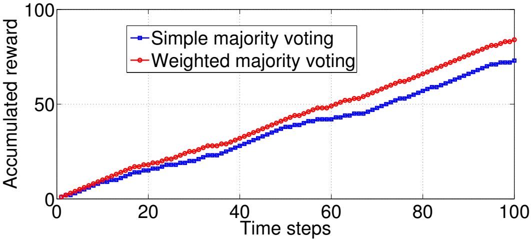
- Each labeler i's decision is weighed by $\log \frac{p_i}{1-p_i}$.
- Have to account for additional error in estimating the weights when determining label outcome.
- A larger constant: slower convergence to a better target.

Experiment I: simulation with $M = 5$

Performance comparison: labeler selection v.s. full crowd-sourcing (simple majority vote)



Comparing weighted and simple majority vote



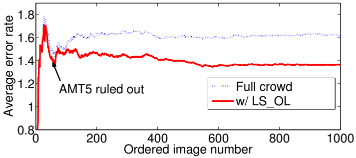
Experiment II: on a real AMT dataset

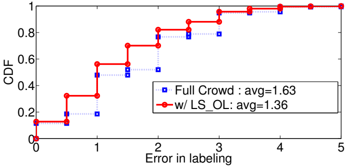
- Contains 1,000 images each labeled by the same set of 5 AMTs.
- Labels are on a scale from 0 to 5, indicating how many scenes are seen from each image.
- A second dataset summarizing keywords for scenes of each image: use this count as the ground truth.

	AMT1	AMT2	AMT3	AMT4	AMT5
# of disagree	348	353	376	338	441

Table : Total number of disagreement each AMT has

Performance comparison





(L) AMT 5 was quickly weeded out; eventually settled on the optimal set of AMTs 1, 2, and 4.

(R) CDF of all images' labeling error at the end of this process.

Conclusion

We discussed two problems

- How to make better recommendation for a user by considering more heavily opinions of other like-minded users.
  - UCB1-like group learning algorithms.
  - Outperforms individual learning.
- How to select the best set of labelers over a sequence of tasks.
  - An algorithm that estimates labeler's quality by comparing against (weighted) majority vote.
  - New regret bound.

Currently under investigation

- Lower bound on the regret in the labeler selection problem.
- Generalization to sequential classifier design.

References

- Y. Liu and M. Liu, "An Online Learning Approach to Improving the Quality of Crowd-sourcing, to appear in *ACM SIGMETRICS*, June 2015, Portland, OR.
- Y. Liu and M. Liu, "Group Learning and Opinion Diffusion in a Broadcast Network," Annual Allerton Conference on Control, Communication, and Computing (Allerton), October 2013, Allerton, IL.