Report Number:

2020-05-07

Elicitation and aggregation of data in knowledge intensive crowdsourcing

Dohoon Kim

With the significant advance of internet and connectivity, crowdsourcing gained more popularity and various crowdsourcing platforms emerged. This project focuses on knowledge-intensive crowdsourcing, in which agents are presented with the tasks that require certain knowledge in domain. Knowledge-intensive crowdsourcing requires agents to have experiences on the specific domain. With the constraint of resources and its trait as sourcing from crowd, platform is likely to draw agents with different levels of expertise and knowledge and asking same task can result in bad performance. Some agents can give better information when they are asked with more general question or more knowledge-specific... **Read complete abstract on page 2.**

Elicitation and aggregation of data in knowledge intensive crowdsourcing

Dohoon Kim

Complete Abstract:

With the significant advance of internet and connectivity, crowdsourcing gained more popularity and various crowdsourcing platforms emerged. This project focuses on knowledge-intensive crowdsourcing, in which agents are presented with the tasks that require certain knowledge in domain. Knowledge-intensive crowdsourcing requires agents to have experiences on the specific domain. With the constraint of resources and its trait as sourcing from crowd, platform is likely to draw agents with different levels of expertise and knowledge and asking same task can result in bad performance. Some agents can give better information when they are asked with more general question or more knowledge-specific task or even other task in the same domain. With this intuition of hierarchy, this project depicts knowledge-structure in domain as tree structure and aims to propose methods on how to assign tasks to the agents to realize the ground truth of the data they are presented.

Elicitation and aggregation of data in knowledge intensive crowdsourcing

Dohoon Kim
Computer Science & Engineering
McKelvey School of Engineering
Washington University in St. Louis
dohoon.kim@wustl.edu

Abstract

With the significant advance of internet and connectivity, crowdsourcing gained more popularity and various crowdsourcing platforms emerged. This project focuses on knowledge-intensive crowdsourcing, in which agents are presented with the tasks that require certain knowledge in domain. Knowledge-intensive crowdsourcing requires agents to have experiences on the specific domain. With the constraint of resources and its trait as sourcing from crowd, platform is likely to draw agents with different levels of expertise and knowledge and asking same task can result in bad performance. Some agents can give better information when they are asked with more general question or more knowledge-specific task or even other task in the same domain. With this intuition of hierarchy, this project depicts knowledge-structure in domain as tree structure and aims to propose methods on how to assign tasks to the agents to realize the ground truth of the data they are presented.

Introduction

Crowdsourcing has been recognized as an efficient and innovative method to elicit and aggregate the data from the individuals, as it works in a collaborative manner with the out-sourced individuals contributing to the given tasks and platform aggregates the information from the agents. General crowdsourcing platform presents simple tasks to the individual agents and aggregate the information from those agents to achieve the results for platform's objective.
Crowdsourcing can be applied in wide range of industry and levels as it is flexible on gathering agents from various backgrounds. And, it leads to the knowledge intensive crowdsourcing, in which agents can be asked with more knowledge requiring tasks.[1]
We assume the platform's objective is to gain high prediction on the ground truth of the categorical data presented to the agents. We know that in real world, agents have different levels of knowledge or expertise, and tasks on the data can vary with the particular knowledge required for particular tasks. But we can intuitively think that those tasks are bounded to certain knowledge domain such as medical domain. For the research, we formulate these knowledge-particular tasks as in tree model since questions at the bottom of the tree involve more knowledge intensive tasks. To maintain the hierarchical tree structure, the parent question involves relatively "general" question than the child questions. Agents are different in their expertise and knowledge and they may be more familiar with some tasks while others may be more familiar with other tasks.
For example, in medical domain, we say the agents are comprised of general doctors, cardiovascular doctors, and neurologists. Platform presents the image with heart disease to the agents in this domain and wishes to realize the ground truth. Cardiovascular doctors can better answer whether there is disease in a heart while general doctors can better answer more general task whether the organ is heart or not.
Goal of the research is to test on methods for platform to assign task to the agent to achieve high prediction on the ground truth in knowledge-intensive crowdsourcing.

Tree Knowledge Structure

We assume the hierarchical knowledge tree model is general for both the agents and platform. Parent node involves the less knowledge requiring task,

while subsequent nodes follow more knowledge intensive tasks. We adopt the tree model since it is widely used in many other fields, and it conserves high accuracy for predictive model as we aim to get maximal posterior belief on ground truth for platform. Tree model also captures the relationship between the questions, as questions on lower nodes involve more "knowledge intensive" question.
Following the previous examples, root node contains the question for distinguishing whether it is heart or brain, and child nodes will have the question for distinguishing the heart diseases, and brain diseases as well.
Basic design of the tree structure is shown below,

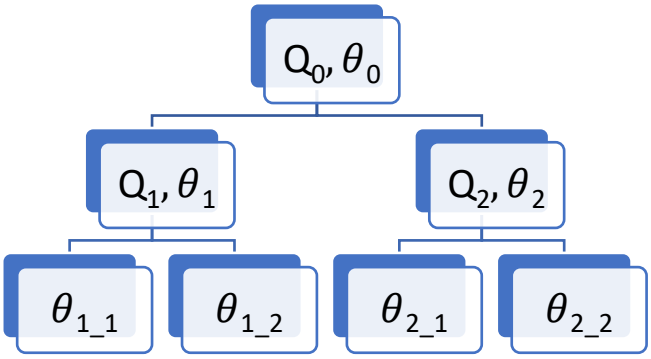

Fig. 1 Tree Model

Fig.1 grasps the examples dealt in motivation above, and this smaller model will work as a proof of concept for bigger tree model to show the feasibility of the method proposed.
Every parent node contains the categorical question, and child nodes contain the answers for the question.

E.g.)
Q_0 : Is it heart or brain?
Q_1 : Is it heart disease or not?
Q_2 : Is it brain disease or not?

Model

Our model is based on Bayesian model and follow the model setup in [2]. Platform elicits the confidence value for categorical data as an answer.
Each question node has the *k* finite number of child nodes (possible answers),
That is, for each tree with question node as a root node, there is a finite number of possible answers χ, and $|\chi| = k$.
With the tree structure, subsequent subtree that has the child node as the root node has a finite number of possible answers χ', and we say $\chi' \in \chi$ since answers in χ' are sub-family of the answer $\theta \in \chi$.
The ground truth (correct answer), θ^* is drawn from the prior distribution $p(\theta)$ with realized value in χ, and it always exists in the most bottom of the whole tree model.
Ground truth, θ^* is unknown to both agents and platform, and platform aims to aggregate the answers from agents to update the posteriors on the nodes to realize the ground truth at the end.
To model reasonable and related workers, we assume that workers have encountered the independent noisy samples related to ground truth and their abilities are designed by the number of samples they encountered for each node in the tree.
That is, each sample x, with $x \in \chi$, is drawn from $p(x|\theta^*)$. We adopt the symmetric noise distribution for $p(x|\theta^*)$.[2]

$$p(x|\theta^*) = (1-\varepsilon)\,\mathbb{1}\{\theta = x\} + \varepsilon * 1/k$$

Agent Update

To model the agents' ability, we assume each worker has encountered n number of samples and we use the counts, C_θ which is number of sample counts for the possible answer θ.
Now, with the Bayesian update, the posterior belief of worker on θ is,

$$p(\theta|x_1, \dots, x_{n'}) = p(\theta_{parent}|x_1, \dots, x_n) * \frac{\prod_{j=1}^{n} p(x_j|\theta)p(\theta)}{p(x_1,\dots x_{n'})} = p(\theta_{parent}|x_1, \dots, x_n) * \frac{\alpha^{c'}\beta^{n_{parent}-c'}p(\theta')}{\sum_{\theta' \in \chi}\alpha^{c'}\beta^{n\_parent-c'}p(\theta')}$$

, where

c' = number of signals for θ',

$\theta \in \theta_{parent}$,

n' = number of signals for parent node,

$[x_1, \dots x_{n'}] \in [x_1, \dots x_n]$

$\alpha = 1 - \varepsilon_i + \varepsilon_i/k$

$\beta = \varepsilon_i/k$

ε_i is epsilon value of agent on task i

This formula holds valid in that the samples $[x_1, \dots, x_{n'}]$ are extracted from the bigger set of samples $[x_1, \dots, x_n]$ and they are conditional on θ, which is the root node of the subtree.

Thus, it should maintain that

$p(\theta|x_1, \ldots, x_n) = p(\theta'|x_1, \ldots, x_n') + p(\theta''|x_1, \ldots, x_n')$, , where θ' and $\theta'' \in \theta$

Platform update

To model the update in the platform, we assume she knows about the prior distribution of the node and epsilon values of the worker.
For each elicitation step, platform asks one question to the agent and elicits the confidence value on the nodes in the subtree for that question.
Since platform has no discerning ability, we give the uniform value of epsilon for all the question nodes. For each question, it involves the "None" node that takes account on all the other nodes than the ones in subtree and we update the posteriors of other nodes based on that.

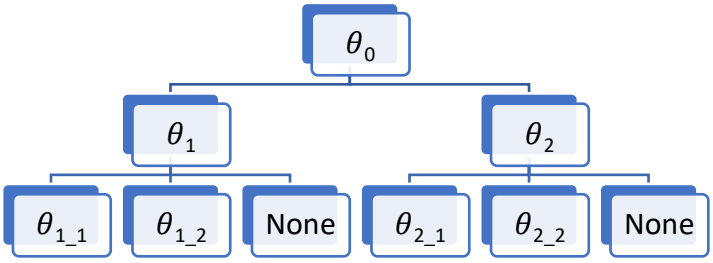

Fig. 2 Platform Tree Model

We uniformly distribute the confidence of "None" node in Fig. 2 among all the other nodes corresponding to the prior probability of the nodes, and it will be discussed below.
After eliciting the confidence from the agent, platform has updated posteriors of agent on the nodes, then converts those posteriors to the sample difference counts because they are one-to-one mapping and further use these to update the posteriors of the platform in the aggregation step.

Difference counts are obtained with, [2]

$$\text{diff}_\theta = c_{\theta'} - c_\theta = \frac{\log\left(\frac{q'}{q}\right) - \log\left(\frac{p'}{p}\right)}{\log\left(\frac{\alpha}{\beta}\right)},$$

where $\alpha = 1 - \varepsilon + \varepsilon/k$, $\beta = \varepsilon/k$, and $\varepsilon = \varepsilon^p$

(ε^p stands for platform epsilon)

From the formula, we can figure that the ratio of q' and q, and the ratio of p' and p are same if the difference is 0. That is, it holds true to distribute the confidence value of None node corresponding to prior distribution as it uniformly distributes the signal counts.
For each worker t from 1,…, T, difference counts vectors are updated and platform aggregates the signal counts and updates its posteriors on the nodes.
To update the posterior of the platform, we use the same formula for the agent.

$$p(\theta|x_1, \ldots, x_{n'}) = p(\theta_{parent}|x_1, \ldots, x_n) * \frac{\prod_{j=1}^{n} p(x_j|\theta)p(\theta)}{p(x_1, \ldots x_{n'})} = p(\theta_{parent}|x_1, \ldots, x_n) * \frac{\alpha^{c'}\beta^{n_{parent}-c'}p(\theta')}{\sum_{\theta' \in \chi}\alpha^{c'}\beta^{n\_parent-c'}p(\theta')}$$

,where
c' = number of signals for θ',
$\theta \in \theta_{parent}$,
n' = number of signals for parent node,
$[x_1, \ldots, x_{n'}] \in [x_1, \ldots, x_n]$
$\alpha = 1 - \varepsilon_p + \varepsilon_p/k$
$\beta = \varepsilon_p/k$
ε_p is uniform epsilon value of platform

Methods

We worked with 3 methods for platform to decide on the question asked. We further test these methods with different types of distribution.

Random Method
We ask random question to the agent and updates the signal counts and posteriors of platform.

Greedy Method
As we assumed platform knows about the epsilon values of agents, platform asks the question that has smallest epsilon value since agent is most discerning on that node.

Heuristic Method
For Heuristic method, for each elicitation step, platform calculates the expected entropy of the "leaf nodes" at the bottom of the tree assuming platform

hypothetically asks each of 3 questions and get hypothetical updates on the signal counts.
Then, with the hypothetical posterior sets, platform figures the expected entropy of the "leaf nodes" at the bottom layer, and selects the question based on the lowest entropy. The reason we use the lowest entropy value to choose the question is since we know the ground truth exists in the leaf nodes and entropy gets lower when the distribution of posteriors gets less uniform.
Thus, with lowest entropy, we can find the set with least uniformly distributed and we need the least uniform distribution of posteriors to achieve maximal posterior belief of platform on ground truth.

Heuristic Steps

1. Before platform asks the question, she hypothetically tests worker on each of 3 questions and get corresponding posterior set of nodes for 3 cases.
2. With 3 posterior sets, calculate the expected entropy for leaf nodes.
3. Find out the set with least entropy among 3 sets.
4. Ask the question corresponding to the set we found.

Simulation

For simulation, we tested with 18 hypothetical workers and with 2-layer tree model.
The worker knowledge tree structure is,

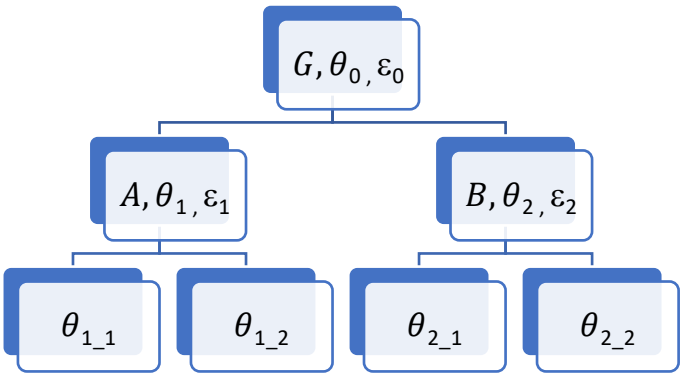

Fig. 3 Simulation Tree Model

To use Fig. 3, we restrict the setting for the simulation that knowledge domain existing is [θ_1, θ_2, θ_{1_1}, θ_{1_2}, θ_{2_1}, θ_{2_2}].

To generate the reasonable worker type, we maintain worker to be discerning for certain task, corresponding to his/her type. Lower epsilons infer that agent has higher signal ratio on the task and further agent's posterior difference is higher for the task.

G-type worker
ε_0 : [0.9, 0.92], ε_1 : [0.94, 0.96], ε_2 : [0.94, 0.96]
Relatively general worker with more discerning for general question G.

A-type worker
ε_0 : [0.93, 0.94], ε_1 : [0.85, 0.87], ε_2 : [0.93, 0.94]
Worker with relatively more discerning for question A

B-type worker
ε_0 : [0.93, 0.94], ε_1 : [0.93, 0.94]. ε_2: [0.85, 0.87]
Worker with relatively more discerning for question B

To test on the different eliciting methods of platform on agents, set

$$\varepsilon^p = 0.95,\ N = 25,\ k = 2,\ \theta^* = \theta_{1\_1},$$
$$P(\theta) = [1.0, 0.5, 0.5, 0.25, 0.25, 0.25, 0.25]$$

For each time t = 1, …., T, worker t comes in,
Elicitation Step:

1. Platform asks a single task.
2. Based on the confidence value(answer), platform updates the posteriors of all the nodes in the tree for the agents.
3. Platform converts the posteriors of agents into the signal differences.

Aggregation Step:

1. Platform updates its signal counts for each node using the signal differences.
2. Platform updates its own posteriors on the nodes using aggregated signal counts.

Result

1. G-type worker distribution

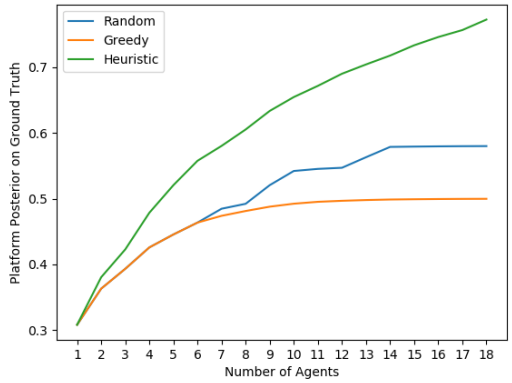

Fig. 4 G distribution graph

- Greedy approach is poor in performance for this distribution because as we only ask the Q_0, which means the signal counts are uniformly distributed among the leaf nodes, thus it is not possible to have high posterior belief on ground truth. Also as the belief on θ_1 gets higher to 1.0 as updated, the belief in ground truth goes to 0.5.
- Random approach shows better performance than greedy approach because it has higher chance of asking question that has ground truth as an answer. And, even with small number of cases where agents are asked Q_1, we can gather more information on the ground truth than greedy method.
- Heuristic method shows high performance even with this distribution because it always tries to select the question with least entropy, and asking Q_0 will incur high entropy with the randomness of the leaf nodes. And, asking Q_1 updates that the posterior belief on θ_1 goes high and, high posterior belief on θ_1 and small signal ratio of θ_{1_1} and θ_{1_2} of the agent affect the heuristic method to ask on Q_1.

2. A-type worker distribution

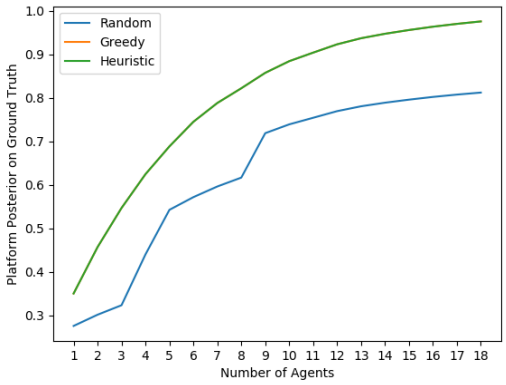

Fig. 5 A distribution graph

- Random approach has poor performance since asking other questions cannot get more information for the ground truth.
- Both greedy method and heuristic method achieve high prediction and convergence for ground truth as they both ask the right question to the agent and gets more information on the ground truth.

3. B-type worker distribution

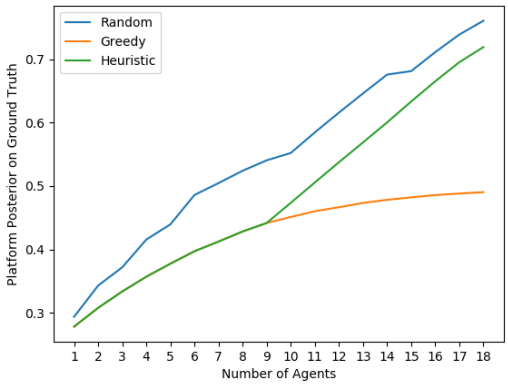

Fig. 6 B distribution graph

- Greedy approach is converging to 0.5 because as we only ask the Q_2, which means the signal counts are uniformly distributed among the leaf nodes, thus it is not possible to have high posterior belief on ground truth. Also as the belief on θ_1 gets higher to 1.0 as updated, the belief in ground truth goes to 0.5.
- Heuristic approach performs better than greedy method since as asking Q_2 updates that the posterior belief on θ_1 goes higher because of the None node and, high posterior belief on θ_1 and small signal ratio

of θ_{1_1} and θ_{1_2} of the agent affect the heuristic method to ask on Q_1.

- Random approach performs better than other two methods since it has higher chance of gathering information for ground truth as it can ask Q_1 from the beginning and more often.

4. Heterogeneous distribution (6G, 6A, 6B)

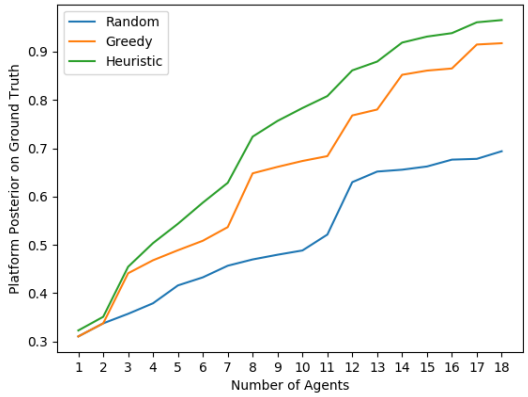

Fig. 7 Hetero distribution graph

- Random approach is poor in performance for heterogeneous distribution as is a constraint on getting more information for ground truth since we just randomly ask the question.
- Greedy approach is better in performance than random approach because it can tightly earn information from the A-type worker on ground truth, and the information from G-type and B-type workers do not directly update on ground truth but still they give indirect information on where ground truth may exist in.
- Heuristic approach shows high performance as the platform uses both currently updated beliefs and worker's ability simultaneously to decide on the question asked to get more information on the ground truth. Thus, during the process of elicitation, platform avoid asking question outside of ground truth and the graph maintains converging to high confidence for the ground truth.

References

1. Ghezzi et al. (2017). Crowdsourcing: A Review and Suggestions for Future Research. International Journal of Management Reviews. DOI: 10.1111/ijmr.12135.
2. Chien-Ju Ho, Rafael Frongillo, and Yiling Chen. 2016. Eliciting categorical data for optimal aggregation. In NIPS 2016. Curran Associates, Inc., 2450–2458

Archives

1. Github: https://github.com/andykim123/Masters_Research