Optimistic Knowledge Gradient Policy for Optimal Budget Allocation in Crowdsourcing

Presentation:
Kamran Ghasedi
Spring 2015
Outline

- Crowdsourcing
- Problem definition
- K-coin tossing problem
- Markov Decision Process (MDP) and Optimal Policy
- Bayesian Setup
- Accuracy Maximization
- Optimization problem
- Stage-wise expected reward
- Optimistic Knowledge Gradient
- Experiment on simulated data
- Experiment on real dataset
- Conclusion
Optimistic Knowledge Gradient Policy for Optimal Budget Allocation in Crowdsourcing

Xi Chen
Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
XICHEN@CS.CMU.EDU

Qihang Lin
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, USA
QIHANGL@ANDREW.CMU.EDU

Dengyong Zhou
Machine Learning Group, Microsoft Research, Redmond, WA 98052, USA
DENGYONG.ZHOU@MICROSOFT.COM

Abstract

We consider the budget allocation problem in binary/multi-class crowd labeling where each label from the crowd has a certain cost. Since different instances have different ambiguities and different workers have different reliabilities, a fundamental challenge is how to allocate a pre-fixed amount of budget among instance-worker pairs so that the overall accuracy can be maximized. We start with a simple setting where all workers are of the same quality and then extend results to more general settings.
Crowdsourcing

• Need training labels for building good classifiers

• Crowdsourcing services (e.g., Amazon Mechanical Turk)

• Collecting training labels from crowd of workers to label training instances

• Noisy labels

• Inferring true labels
Problem definition

Given the limited amount of budget, it is important to wisely allocate the budget among instances so that the overall accuracy is maximized.

1. How to accurately estimate the labeling difficulty for each instance?
2. Whether to spend more budget on ambiguous instances?
3. How to estimate the reliability of workers?
K-coin tossing problem

• Binary labeling task
• workers are identical and noiseless (perfectly reliable)

• We are allowed to sequentially specify a coin to toss.
• Then observe the outcome of the toss.
• We note that each coin can be chosen multiple times.
• After the coin toss budget T runs out, we decide whether a coin is biased more towards the head or the tail for each coin.
Markov Decision Process (MDP) and Optimal Policy

- $K$ instances
- True label $Z_i \in \{-1, 1\}$ for $1 \leq i \leq K$.
- Positive set $H^* = \{i : Z_i = 1\} = \{i : \theta_i \geq 0.5\}$
- Labeling difficulty of each instance $\theta_i \in [0, 1]$
- Total budget $T$,
- Action set $i_t \in A = \{1, \ldots, K\}$
- Predicted label $y_{i_t}$
- Optimal allocation sequence $(i_0, \ldots, i_{T-1})$
Illustration Example

Table 1. Labeling matrix

<table>
<thead>
<tr>
<th>Acc</th>
<th>Cur.</th>
<th>$y = 1$</th>
<th>$y = -1$</th>
<th>Expected Acc</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_1 &gt; 0.5$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$\theta_1 &lt; 0.5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\theta_2 &gt; 0.5$</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td>$\theta_2$</td>
<td>$\theta_2 - 0.5 &gt; 0$</td>
</tr>
<tr>
<td>$\theta_2 &lt; 0.5$</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
<td>$1 - \theta_2$</td>
<td>$0.5 - \theta_2 &gt; 0$</td>
</tr>
<tr>
<td>$\theta_3 &gt; 0.5$</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>$\theta_3 + 0.5(1 - \theta_3)$</td>
<td>$0.5(\theta_3 - 1) &lt; 0$</td>
</tr>
<tr>
<td>$\theta_3 &lt; 0.5$</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>$0.5(1 - \theta_3)$</td>
<td>$0.5(1 - \theta_3) &gt; 0$</td>
</tr>
</tbody>
</table>

Table 2. Calculation of the expected improvement. The 2nd column is the current accuracy. The 3rd and 4th are accuracies if the next label is 1 and -1.

Blue region:

$max(\theta_2 - 0.5, 0.5 - \theta_2) > 0.5(1 - \theta_3)$ or $\theta_3 > 0.5$
Beta is the conjugate prior of the Bernoulli:

\( y_{i_t} \sim \text{Bernoulli}(\theta_{i_t}) \)

\( \theta_i \) is drawn from a known Beta prior distribution \( \text{Beta}(a_i^0, b_i^0) \)

Mathematical expression:

\[
f(x; \alpha, \beta) = \text{constant} \cdot x^{\alpha-1}(1-x)^{\beta-1}
\]

\[
= \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1} \, du}
\]

\[
= \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}
\]
$y_{it} \sim \text{Bernoulli}(\theta_{it})$

Beta is the conjugate prior of the Bernoulli:

$\theta_{it}$ is drawn from a known Beta prior distribution $\text{Beta}(a^0_{it}, b^0_{it})$

\[
\begin{align*}
\text{Beta}(a^{t+1}_{it}, b^{t+1}_{it}) &= \text{Beta}(a^t_{it} + 1, b^t_{it}) \text{ if } y_{it} = 1 \\
\text{Beta}(a^{t+1}_{it}, b^{t+1}_{it}) &= \text{Beta}(a^t_{it}, b^t_{it} + 1) \text{ if } y_{it} = -1
\end{align*}
\]

$S^t = \{a^t_{i_1}, b^t_{i_1}\}_{i=1}^K$ \hspace{2cm} $K \times 2$ matrix

$S^{t+1} = \begin{cases} 
S^t + (e_{it}, 0) & \text{if } y_{it} = 1; \\
S^t + (0, e_{it}) & \text{if } y_{it} = -1,
\end{cases}$

$\Pr(y_{it} = 1|S^t, i_t) = \mathbb{E}(\theta_{it} | S^t) = \frac{a^t_{it}}{a^t_{it} + b^t_{it}}$, \\
$\Pr(y_{it} = -1|S^t, i_t) = 1 - \Pr(y_{it} = 1|S^t, i_t)$
Conditional distribution $\theta_i | \mathcal{F}_t$ \quad \longrightarrow \quad Posterior distribution $\text{Beta}(a_i^t, b_i^t)$

$I(a, b) = \Pr(\theta \geq 0.5 | \theta \sim \text{Beta}(a, b))$,
$P_i^t = \Pr(i \in H^* | \mathcal{F}_t) = \Pr(\theta \geq 0.5 | S_i^t) = I(a_i^t, b_i^t),$
Optimization problem

\{ F_t \}_{t=0}^T \leftarrow (i_0, y_{i_0}, \ldots, i_{t-1}, y_{i_{t-1}})

H_T = \arg \max_{H \subseteq \{1, \ldots, K\}} E \left( \sum_{i \in H} 1(i \in H^*) + \sum_{i \notin H} 1(i \notin H^*) \middle| F_T \right)

Conditional distribution \( \theta_i | F_t \) \quad \longrightarrow \quad Posterior distribution \( \text{Beta}(a_i^t, b_i^t) \)

\[
I(a, b) = \Pr(\theta \geq 0.5|\theta \sim \text{Beta}(a, b)),
\]

\[
P_i^t = \Pr(i \in H^*|F_t) = \Pr(\theta \geq 0.5|S_i^t) = I(a_i^t, b_i^t),
\]

**Proposition 2.1** \( H_T = \{i : P_i^T \geq 0.5\} \) solves (3) and the expected accuracy on RHS of (3) can be written as \( \sum_{i=1}^K h(P_i^T) \), where \( h(x) = \max(x, 1 - x) \).

\[
V(S^0) \doteq \sup_{\pi} E^\pi \left[ E \left( \sum_{i \in H_T} 1(i \in H^*) + \sum_{i \notin H_T} 1(i \notin H^*) \middle| F_T \right) \right]
\]

\[
= \sup_{\pi} E^\pi \left( \sum_{i=1}^K h(P_i^T) \right), \quad (6)
\]
Stage-wise expected reward

Proposition 2.3 Define the stage-wise expected reward as:

$$R(S^t, i_t) = \mathbb{E} \left( h(P^t_{i_t} + 1) - h(P^t_{i_t}) | S^t, i_t \right), \quad (7)$$

$$V(S^0) \doteq \sup_\pi \mathbb{E}^\pi \left[ \mathbb{E} \left( \sum_{i \in H_T} 1(i \in H^*) + \sum_{i \notin H_T} 1(i \notin H^*) | F_T \right) \right]$$

$$= \sup_\pi \mathbb{E}^\pi \left( \sum_{i=1}^K h(P^T_i) \right), \quad (6)$$

$$V(S^0) = G_0(S^0) + \sup_\pi \mathbb{E}^\pi \left( \sum_{t=0}^{T-1} R(S^t, i_t) \right), \quad (8)$$

where $$G_0(S^0) = \sum_{i=1}^K h(P^0_i)$$

With Proposition 2.3, the maximization problem (6) is formulated as a T-stage MDP (8).
Stage-wise expected reward

\[ V(S^0) = G_0(S^0) + \sup_\pi \mathbb{E}^\pi \left( \sum_{t=0}^{T-1} R(S^t, i_t) \right) \tag{8} \]

since \( S_{i_t}^t = (a_{i_t}^t, b_{i_t}^t) \in \mathbb{R}_{+}^2 \quad \rightarrow \quad R(a_{i_t}^t, b_{i_t}^t) = R(S^t, i_t) \)

\[
R_1(a, b) = h(I(a + 1, b)) - h(I(a, b)), \\
R_2(a, b) = h(I(a, b + 1)) - h(I(a, b)).
\]

Expected reward:

\[
R(a, b) = p_1 R_1 + p_2 R_2 \\
p_1 = \frac{a}{a+b} \quad p_2 = \frac{b}{a+b}
\]
Dynamic programming (DP) algorithm (Puterman, 2005; Powell, 2007) (a.k.a. backward induction)

computation is intractable, since size of the state space grows exponentially in $t$

need some computationally efficient approximate policies
Problem: finite-horizon Bayesian multi-armed bandit (MAB)

- calibration method (Gittins, 1989; Nino-Mora, 2011) $O(T^3)$ time and space complexity

- the state-of-the-art exact method (Nino-Mora, 2011) $O(T^6)$ time and space complexity

Knowledge gradient (KG) method:
Knowledge Gradient:

\[ i_t = \arg \max_i \left( R(a_t^i, b_t^i) \triangleq \frac{a_t^i}{a_t^i + b_t^i} R_1(a_t^i, b_t^i) + \frac{b_t^i}{a_t^i + b_t^i} R_2(a_t^i, b_t^i) \right). \] (12)

Deterministic KG is NOT a consistent policy, where the consistent policy refers to the policy that will achieve 100% accuracy almost surely when T goes to infinity.

randomized KG policy’s empirical performance is undesirable.
Conditional Value-at-Risk (CVaR)

\[ \text{CVaR}_\alpha(X) = \max_{\{q_1 \geq 0, q_2 \geq 0\}} q_1 R_1 + q_2 R_2, \]

s.t. \( q_1 \leq \frac{1}{\alpha} p_1, \ q_2 \leq \frac{1}{\alpha} p_2, \ q_1 + q_2 = 1. \)

when \( \alpha = 1 \quad \rightarrow \quad \text{CVaR}_\alpha(X) = p_1 R_1 + p_2 R_2 \)

when \( \alpha \rightarrow 0, \quad \rightarrow \quad \text{CVaR}_\alpha(X) = \max(R_1, R_2) \)
The stage-wise reward $R(a, b)$ can be viewed as a random variable with a two point distribution:

$$
\begin{align*}
\text{probability } p_1 &= \frac{a}{a+b} \text{ of being } R_1(a, b) \\
\text{probability } p_2 &= \frac{b}{a+b} \text{ of being } R_2(a, b)
\end{align*}
$$

A simple idea is to select the instance based on the optimistic outcome of the reward, i.e., the instance with the largest:

$$
R^+(a, b) = \max(R_1(a, b), R_2(a, b)).
$$

Time complexity: $O(KT)$

Space complexity: $O(K)$
Algorithm 1 Optimistic Knowledge Gradient

Input: Parameters of prior distributions for instances $\{a^0_i, b^0_i\}_{i=1}^K$ and the total budget $T$.

for $t = 0, \ldots, T - 1$ do

Select the next instance $i_t$ to label according to:

$$i_t = \arg\max_{i \in \{1, \ldots, K\}} \left( R^+(a^t_i, b^t_i) = \max(R_1(a^t_i, b^t_i), R_2(a^t_i, b^t_i)) \right).$$

Acquire the label $y_{i_t} \in \{-1, 1\}$.

if $y_{i_t} = 1$ then

$a^{t+1}_{i_t} = a^t_{i_t} + 1, b^{t+1}_{i_t} = b^t_{i_t}; a^{t+1}_i = a^t_i, b^{t+1}_i = b^t_i$ for all $i \neq i_t$.

else

$a^{t+1}_{i_t} = a^t_{i_t}, b^{t+1}_{i_t} = b^t_{i_t} + 1; a^{t+1}_i = a^t_i, b^{t+1}_i = b^t_i$ for all $i \neq i_t$.

end if

end for

Output: The positive set $H_T = \{i : a^T_i \geq b^T_i\}$. 
Incorporate Workers' Reliability

$M$ workers

reliability of the $j$-th worker $\rho_j \in [0, 1]$

$$\rho_j = \Pr(Z_{ij} = Y_i|Y_i)$$

$$\Pr(Z_{ij} = 1) = \Pr(Z_{ij} = 1|Y_i = 1) \Pr(Y_i = 1) + \Pr(Z_{ij} = 1|Y_i = -1) \Pr(Y_i = -1)$$

$$= \rho_j \theta_i + (1 - \rho_j)(1 - \theta_i). \quad (13)$$

If we assume: $\rho_j \sim \text{Beta}(c_j^0, d_j^0)$

When we observe label 1: $\Pr(Z_{ij} = 1|\theta_i, \rho_j) = \theta_i \rho_j + (1 - \theta_i)(1 - \rho_j)$

When we observe label -1: $\Pr(Z_{ij} = -1|\theta_i, \rho_j) = (1 - \theta_i) \rho_j + \theta_i (1 - \rho_j)$

$$p(\theta_i, \rho_j|Z_{ij} = z) \approx p(\theta_i|Z_{ij} = z)p(\rho_j|Z_{ij} = z)$$
Extensions

- Incorporate feature information of instances
- Address multi-class labeling problems
Experiment on simulated data

\[ K = 50 \]
generate 20 different sets of \( \{\theta_i\}_{i=1}^K \).  
total budget \( T = 2K, 3K, \ldots, 10K \)

workers’ reliability \( \rho_j \sim \text{Beta}(4, 1) \)  
\( j = 1, \ldots, 10 \)

(a) Majority Vote

(b) One-Coin Model
Experiment on real dataset

Recognizing textual entailment (RTE)
800 instances and each instance is a sentence pair
10 workers
Whether the second hypothesis sentence can be inferred from the first one

Workers' reliability: $\text{Beta}(4, 1)$

(a) RTE: Majority Vote

(b) RTE: One-Coin Model
Conclusion

- Formulate the budget allocation in crowdsourcing into a MDP and characterize the optimal policy using DP.

- Computationally propose an approximate policy, optimistic knowledge gradient.

- MDP formulation can be used as a general framework to address various budget allocation problems in crowdsourcing.
Thanks for your attention