Optimistic Knowledge Gradient Policy for Optimal Budget Allocation in Crowdsourcing

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

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

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

Abstract

In real crowdsourcing applications, each label from a crowd usually comes with a certain cost. Given a pre-fixed amount of budget, since different tasks have different ambiguities and different workers have different expertises, we want to find an optimal way to allocate the budget among instance-worker pairs such that the overall label quality can be maximized. To address this issue, we start from the simplest setting in which all workers are assumed to be perfect. We formulate the problem as a Bayesian Markov Decision Process (MDP). Using the dynamic programming (DP) algorithm, one can obtain the optimal allocation policy for a given budget. However, DP is computationally intractable. To solve the computational challenge, we propose a novel approximate policy which is called optimistic knowledge gradient. It is practically efficient while theoretically its consistency can be guaranteed. We then extend the MDP framework to deal with inhomogeneous workers and tasks with contextual information available. The experiments on both simulated and real data demonstrate the superiority of our method.

1. Introduction

In many real-world machine learning applications, obtaining sufficient training labels is often the major obstacle for good performance. Due to the flourish of many online crowdsourcing services (e.g., Amazon Mechanical Turk), an effective way of collecting training labels is to ask a crowd of low-paid nonexpert workers for labeling. The class labels provided by the crowd could be highly noisy, so each instance has to be labeled several times by different workers such that that we have a large chance to correctly estimate the underlying true label from those noisy labels. Each label from the crowd usually has a certain cost (e.g., 10 cents). Given a limited amount of budget, it is important to wisely allocate the budget among instances and workers so that the overall accuracy is maximized. To tackle this problem, there are the following challenges. We need to estimate the labeling ambiguity for each instance on the fly and avoid spending much budget on fairly easy instances. On the other hand, however, we also need to avoid spending much budget on few highly ambiguous instances. Our goal is to maximize the *overall labeling accuracy*. Ideally, we should simply put those few highly ambiguous instances aside to save budget for labeling many other relatively easy instances. In addition, we also need to estimate the reliability of each worker on the fly and allocate as many labeling tasks to reliable workers as possible.

To address these challenges in budget-optimal crowdsourcing, we start from the binary labeling task and assume that: (1) each instance is associated with an unknown probability of being positive; (2) for any given instance and worker pair, the label provided by the worker is drawn from the underlying label distribution of the instance. So it means that each worker is perfectly reliable. Later we will relax (2) to consider inhomogeneous workers. At a first glance, such an assumption may seem oversimplified and thus naive. In fact, it turns out that the budget-optimal crowdsourc-

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

ing problem with such an assumption has been highly non-trivial. We can imagine it as a K-coin tossing problem. Each coin has a unknown head probability. We have a budget of T tosses. We sequentially choose a coin to toss according to some policy. We then observe the outcome. A coin may be chosen multiple times. After the tossing budget runs out, we predict for each coin if it is biased to head or tail using all the observed outcomes. Our goal is to find a policy such that the overall prediction accuracy is maximized.

To search for the optimal allocation policy, we adopt the Bayesian setting and formulate the problem into a finite-horizon Markov Decision Process (MDP). Here the Bayesian setting is necessary. We shall show that an optimal policy only exists in the Bayesian setting. Using the MDP formulation, the optimal budget allocation policy for any finite budget T can be readily obtained via the dynamic programming (DP). However, DP is computationally intractable for large-scale problems since the size of the state space grows exponentially in T. The existing widely-used approximate policies, such as approximate Gittins index rule (Gittins, 1989) or knowledge gradient (KG) (Gupta & Miescke, 1996; Frazier et al., 2008), either has a high computational cost or poor performance in our problem. In this paper, we propose a new policy, called *optimistic knowledge gradient*, which combines the KG and the conditional value-at-risk measure (Rockafellar & Uryasev, 2002). The optimistic KG is computationally efficient and achieves superior empirical performance. Theoretically, we prove that it is consistent, that is, when the budget T goes to infinity, the accuracy converges to 100% almost surely.

It is easy to extend the MDP formulation to deal with inhomogeneous workers. We introduce one parameter to characterize worker reliability and update the joint distribution of instance labeling difficulty and worker reliability on the fly using the variational approximation. Then our decision process simultaneously selects the next instance to label and the next worker for labeling the instance. The MDP framework is so flexible that we can further easily extend it to incorporate instance contextual information whenever they are available and to handle multi-class labeling.

In summary, the main contribution of the paper consists of the three folds: (1) we formulate the budget allocation in crowdsourcing into a MDP and characterize the *optimal* policy using DP; (2) computationally, we propose an efficient approximate policy, optimistic knowledge gradient; (3) the MDP frameowrk can be used as a general framework to address various budget allocation problems in crowdsourcing.

2. MDP and Optimal Policy

To better illustrate our model, we first introduce a simplified *homogeneous worker setting* for binary-labeling task. We note that such a simplification is important for investigating this problem, since the incorporation of workers' reliability becomes rather straightforward once this simplified problem is correctly modeled (see Section 4).

Suppose that there are K instances and each one is associated with a true label $Z_i \in \{-1, 1\}$ for $1 \le i \le K$. We denote the positive set by $H^* = \{i : Z_i = 1\}$. Moreover, we characterize the labeling difficulty of each instance by $\theta_i \in [0, 1]$. More precisely, θ_i can be interpreted as the percentage of the workers labels the i-th instance as positive if a large number of *noiseless* (perfectly reliable) workers are asked for the labeling task. Let's consider a concrete example of identifying whether an individual is an adult (positive) or not (negative) by presenting his/her photo to workers. For an individual above 25 years old, the corresponding θ_i is close to 1; and for one below 15, θ_i is close to 0. For these individuals, it is easy to get consensus labels and thus only a few labels for each photo are enough. On the other hand, for an individual between 15 and 25, θ_i will be close to 0.5 and the corresponding labeling task is difficult. We assume that the soft-label θ_i is consistent with the true label in the sense that $Z_i = 1$ if and only if $\theta_i \ge 0.5$ and hence $H^* = \{i : \theta_i \ge 0.5\}$.

As described in the introduction, we can model the sequential labeling process as a coin-tossing problem. Given the total budget T, at each stage $0 \le t \le T-1$, we choose an instance $i_t \in \mathcal{A} = \{1, \ldots, K\}$ to acquire its label from a random *noiseless* worker. According to the meaning of θ_{i_t}, the label we obtained, $y_{i_t} \in \{-1, 1\}$, will follow the Bernoulli distribution with the parameter θ_{i_t}. In fact, θ_{i_t} can be viewed as the head probability of the i_t-th coin/instance and the label y_{i_t} is the outcome of the coin toss at the stage t. We note that, at this moment, all workers are assumed to be identical so that y_{i_t} only depends on θ_{i_t} but not on which worker gives the label. When the budget is exhausted, we need to make an inference about the true label and estimate the positive set $\widehat{H}$. Since workers are assumed to be identical, the most straightforward way is by the *majority vote*. Our goal is to determine the *optimal allocation sequence* (a.k.a. optimal allocation policy) $(i_0, \ldots, i_{T-1})$ so that overall accuracy is maximized. Here, a natural question to ask is whether the *optimal* allocation policy exists and what assumptions do we need for the existence of the optimal policy. To answer this question, we provide a concrete example and motivates our Bayesian setting.

θ_1	1	1	
θ_2	1	-1	?
θ_3	1	?	

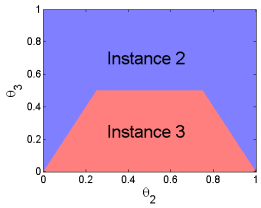


Figure 1. Left: label matrix. Right: decision boundary.

Table 1. Expected improvements. The current accuracies are in the 2nd column. The 3rd and 4th contain the accuracies with the next labels being 1 and -1.

Acc	Cur.	$y=1$	$y=-1$	Expected Acc	Improvement
$\theta_1 > 0.5$	1	1	1	1	0
$\theta_1 < 0.5$	0	0	0	0	0
$\theta_2 > 0.5$	0.5	1	0	θ_2	$\theta_2 - 0.5 > 0$
$\theta_2 < 0.5$	0.5	0	1	$1-\theta_2$	$0.5 - \theta_2 > 0$
$\theta_3 > 0.5$	1	1	0.5	$\theta_3 + 0.5(1-\theta_3)$	$0.5(\theta_3 - 1) < 0$
$\theta_3 < 0.5$	0	0	0.5	$0.5(1-\theta_3)$	$0.5(1-\theta_3) > 0$

2.1. Illustration Example

Let us check a toy example with 3 instances and 5 collected labels (Figure 1). *If we only have the budget to get one more label, which instance should be chosen to label?* It might be obvious that we do not have to put the remaining budget on the first instance since we are relatively more confident on what its true label should be. Thus, the problem becomes how to choose between the second and third instances. In what follows, we shall show that *there is no uniformly optimal policy under the frequentist setting*. A uniformly optimal policy can only exist in the Bayesian setting.

Let us compute the expected improvement in accuracy in terms of the frequentist risk (Table 1). We assume that $\theta_i \neq 0.5$ and if the number of 1 and -1 labels are the same for an instance, the accuracy is 0.5 based on a random guess. From Table 1, we should not label the first instance since the improvement is always 0. This coincides with our intuition. When $\max(\theta_2 - 0.5, 0.5 - \theta_2) > 0.5(1-\theta_3)$ or $\theta_3 > 0.5$, which corresponds to the blue region in Figure 1, we should choose to label the second instance. Otherwise, we should ask the label for the third one. Since the true value of θ_2 and θ_3 are unknown, a uniformly optimal policy does not exist. In constrast, if we choose the Bayesian setting with prior distribution on each θ_i, we could determine the next instance for labeling by taking another expectation over the distribution of θ_i. Therefore, we adopt the Bayesian setting to formulate the budget allocation problem in crowdsourcing.

2.2. Bayesian Setup

We assume that each θ_i is drawn from a known Beta prior distribution $\text{Beta}(a_i^0, b_i^0)$. This can be interpreted as having a_i^0 positive and b_i^0 negative pseudo-labels for the i-th instance at the initial stage. In practice when there is no prior knowledge, we can simply assume $a_i^0 = b_i^0 = 1$ so that the prior is a uniform distribution. Other objective priors (e.g., Jeffreys prior or reference prior) can also be adopted (Robert, 2007).

At each stage t with $\text{Beta}(a_i^t, b_i^t)$ as the current posterior distribution for θ_i, we choose an instance i_t and acquire its label $y_{i_t} \sim \text{Bernoulli}(\theta_{i_t})$. By the fact that Beta is the conjugate prior of the Bernoulli, the posterior of θ_{i_t} in the stage $t+1$ will be updated as $\text{Beta}(a_{i_t}^{t+1}, b_{i_t}^{t+1}) = \text{Beta}(a_{i_t}^t + 1, b_{i_t}^t)$ if $y_{i_t} = 1$ and $\text{Beta}(a_{i_t}^{t+1}, b_{i_t}^{t+1}) = \text{Beta}(a_{i_t}^t, b_{i_t}^t + 1)$ if $y_{i_t} = -1$. We put $\{a_i^t, b_i^t\}_{i=1}^K$ into a $K \times 2$ matrix S^t, called a state matrix, and let $S_i^t = (a_i^t, b_i^t)$ be the i-th row of S^t. The update of the state matrix can be written in a more compact form:

$$S^{t+1} = \begin{cases} S^t + (\mathbf{e}_{i_t}, \mathbf{0}) & \text{if } y_{i_t} = 1; \\ S^t + (\mathbf{0}, \mathbf{e}_{i_t}) & \text{if } y_{i_t} = -1, \end{cases} \tag{1}$$

where $\mathbf{e}_{i_t}$ is a $K \times 1$ vector with 1 at the i_t-th entry and 0 at all other entries. As we can see, $\{S^t\}$ is a Markovian process because S^{t+1} is completely determined by the current state S^t, the action i_t and the obtained label y_{i_t}. It is easy to calculate the state transition probability $\Pr(y_{i_t}|S^t, i_t)$, which is the posterior probability that we are in the next state S^{t+1} if we choose i_t to be label in the current state S^t:

$$\Pr(y_{i_t} = 1|S^t, i_t) = \mathbb{E}(\theta_{i_t}|S^t) = \frac{a_{i_t}^t}{a_{i_t}^t + b_{i_t}^t}, \tag{2}$$

and $\Pr(y_{i_t} = -1|S^t, i_t) = 1 - \Pr(y_{i_t} = 1|S^t, i_t)$. Given this labeling process, we further define a filtration $\{\mathcal{F}_t\}_{t=0}^T$, where $\mathcal{F}_t$ is the σ-algebra generated by the sample path $(i_0, y_{i_0}, \ldots, i_{t-1}, y_{i_{t-1}})$. We choose the action i_t, i.e., the instance to label, after we observe the historical labeling results up to the stage $t-1$. Hence, i_t is $\mathcal{F}_t$-measurable. The budget allocation policy is defined as a sequence of decisions: $\pi = (i_0, \ldots, i_{T-1})$.

2.3. Accuracy Maximization

At the stage T when the budget is exhausted, we need to infer the true label of each instance based on the collected labels. In particular, we need to determine a positive set H_T which maximizes the *conditional* expected accuracy conditioning on $\mathcal{F}_T$ (i.e., minimizing the posterior risk):

$$H_T = \underset{H \subset \{1,\ldots,K\}}{\arg\max}\ \mathbb{E}\left(\sum_{i \in H} \mathbf{1}(i \in H^*) + \sum_{i \notin H} \mathbf{1}(i \notin H^*) \middle| \mathcal{F}_T \right), \tag{3}$$

where $\mathbf{1}(\cdot)$ is the indicator function. We first observe that, for $0 \le t \le T$, the conditional distribution $\theta_i|\mathcal{F}_t$ is exactly the posterior distribution $\text{Beta}(a_i^t, b_i^t)$, which depends on the historical sampling results only through $S_i^t = (a_i^t, b_i^t)$. Hence, we define

$$I(a,b) = \Pr(\theta \ge 0.5 | \theta \sim \text{Beta}(a,b)), \tag{4}$$

$$P_i^t = \Pr(i \in H^* | \mathcal{F}_t) = \Pr(\theta \ge 0.5 | S_i^t) = I(a_i^t, b_i^t), \tag{5}$$

As shown in (Xie & Frazier, 2012), the final positive set H_T can be determined by the Bayes decision rule.

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

According to the next corollary with the proof in Appendix, we show that the construction of H_T is based on the *majority vote*.

Corollary 2.2 *$I(a,b) > 0.5$ if and only if $a > b$ and $I(a,b) = 0.5$ if and only if $a = b$. Therefore, $H_T = \{i : a_i^T \ge b_i^T\}$ solves (3).*

By viewing a_i^0 and b_i^0 as pseudo-counts of 1s and -1s, a_i^T and b_i^T are the total counts of 1s and -1s. The estimated positive set $H_T = \{i : a_i^T \ge b_i^T\}$ consists of instances with more (or equal) counts of 1s than that of -1s. When $a_i^0 = b_i^0$, H_T is constructed exactly according to the *majority vote* rule.

To find the optimal allocation policy which maximizes the expected accuracy, we need to solve the following optimization problem:

$$\begin{aligned} V(S^0) &\doteq \sup_\pi \mathbb{E}^\pi \left[\mathbb{E}\left(\sum_{i \in H_T} \mathbf{1}(i \in H^*) + \sum_{i \notin H_T} \mathbf{1}(i \notin H^*) \Big| \mathcal{F}_T \right) \right] \\ &= \sup_\pi \mathbb{E}^\pi \left(\sum_{i=1}^K h(P_i^T) \right), \end{aligned} \tag{6}$$

where $\mathbb{E}^\pi$ represents the expectation taken over the sample paths $(i_0, y_{i_0}, \ldots, i_{T-1}, y_{i_{T-1}})$ generated by a policy π. The second equality is due to Proposition 2.1 and $V(S^0)$ is called value function at the initial state S^0. The optimal policy π^* is any policy π that attains the supremum in (6).

2.4. MDP and Optimal Policy

To solve the optimization problem in (6), we formulate it into a Markov Decision Process (MDP). To do so, we use the technique from (Xie & Frazier, 2012) to decompose the final expected accuracy as a sum of *stage-wise rewards* as shown in the next proposition. Note that the problem in (Xie & Frazier, 2012) is an infinite-horizon one which optimizes the stopping time while our problem is *finite-horizon* since the labeling procedure must be stopped at the stage T.

Proposition 2.3 *Define the stage-wise expected reward as:*

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

then the value function (6) becomes:

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

where $G_0(S^0) = \sum_{i=1}^K h(P_i^0)$ and the optimal policy π^* is any policy π that attains the supremum.

Since the expected reward (7) only depends on $S_{i_t}^t = (a_{i_t}^t, b_{i_t}^t) \in \mathbb{R}_+^2$, we define $R(a_{i_t}^t, b_{i_t}^t) = R(S^t, i_t)$ and use them interchangeably. As a function on $\mathbb{R}_+^2$, $R(a,b)$ has an analytical representation. In fact, for any state (a,b) of a single instance, the reward of getting a label 1 and a label -1 are:

$$R_1(a,b) = h(I(a+1,b)) - h(I(a,b)), \tag{9}$$

$$R_2(a,b) = h(I(a,b+1)) - h(I(a,b)). \tag{10}$$

The expected reward $R(a,b) = p_1 R_1 + p_2 R_2$ with $p_1 = \frac{a}{a+b}$ and $p_2 = \frac{b}{a+b}$ are transition probabilities in (2).

With Proposition 2.3, the maximization problem (6) is formulated as a T-stage MDP (8), which is associated with a tuple $\{T, \{\mathcal{S}^t\}, \mathcal{A}, \Pr(y_{i_t}|S^t, i_t), R(S^t, i_t)\}$. Here, the state space at the stage t, $\mathcal{S}^t$, is all possible states that can be reached at t. Once we collect a label y_{i_t}, one element in S^t (either $a_{i_t}^t$ or $b_{i_t}^t$) will add one. Therefore, we have

$$\mathcal{S}^t = \left\{ \{a_i^t, b_i^t\}_{i=1}^K : a_i^t \ge a_i^0, b_i^t \ge b_i^0, \sum_{i=1}^K (a_i^t - a_i^0) + (b_i^t - b_i^0) = t \right\}. \tag{11}$$

The action space is the set of instances that could be labeled next: $\mathcal{A} = \{1, \ldots, K\}$. The transition probability $\Pr(y_{i_t}|S^t, i_t)$ is defined in (2) and the expected reward at each stage $R(S^t, i_t)$ is defined in (7). Moreover, due to the Markovian property of $\{S^t\}$, it is enough to consider a Markovian policy (Powell, 2007) where i_t is chosen only based on the state S^t.

With the MDP in place, we can apply dynamic programming (DP) algorithm (Puterman, 2005; Powell, 2007) (a.k.a. backward induction) to to compute the *optimal* policy according to the Bellman equation. Although DP can identify the optimal policy, its computation is intractable since the size of the state space $|\mathcal{S}^t|$ grows exponentially in t according to (11). Therefore, we need some computationally efficient approximate policies.

3. Optimistic Knowledge Gradient

In this section, we first review some existing approximate policies to better motivate the proposed new policy named *optimistic knowledge gradient*.

3.1. Existing Approximate Policies

The simplest approximate policy is the uniform sampling (a.k.a, pure exploration), i.e., we choose the next instance to label uniformly and independently at random: $i_t \sim \text{Uniform}(1, \ldots, K)$. However, this policy does not explore the structure of the problem.

With the decomposed reward function, our problem is essentially a finite-horizon Bayesian multi-armed bandit (MAB) problem. Gittins (1989) showed that Gittins index is an optimal policy for infinite-horizon MAB with the discounted reward. Since our problem is finite-horizon, Gittins index is no longer optimal while it can still provide us a good heuristic index rule. However, the computational cost of Gittins index is very high. To compute finite-horizon Gittins index for our problem, the *approximate* method (i.e., calibration method (Gittins, 1989; Nino-Mora, 2011)) requires $O(T^3)$ time and space complexity; while the state-of-the-art exact method (Nino-Mora, 2011) requires $O(T^6)$ time and space complexity.

A computationally more attractive policy is the knowledge gradient (KG) (Frazier et al., 2008). It is essentially a single-step look-ahead policy, which greedily selects the next instance with the largest expected reward:

$$i_t = \arg\max_i \left(R(a_i^t, b_i^t) \doteq \frac{a_i^t}{a_i^t + b_i^t} R_1(a_i^t, b_i^t) + \frac{b_i^t}{a_i^t + b_i^t} R_2(a_i^t, b_i^t) \right). \tag{12}$$

As we can see, this policy corresponds to the first step in DP algorithm and hence KG policy is optimal if only one labeling chance is remaining.

When there is a tie, if we select the one with the smallest index, the policy is referred to *deterministic KG*; while if we randomly break the tie, the policy is referred to *randomized KG*. Although KG has been successfully applied to many MDP problems (Powell, 2007), it will fail in our problem as shown in the next proposition with the proof in Appendix.

Proposition 3.1 *Assuming that a_i^0 and b_i^0 are positive integers and letting $\mathcal{E} = \{i : a_i^0 = b_i^0\}$, then the deterministic KG policy will acquire one label for each item in $\mathcal{E}$ and then consistently obtain the label for the first item even if the budget T goes to infinity.*

According to Proposition 3.1, the 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. We note that randomized KG policy can address this problem. However, from the proof of Proposition 3.1, randomized KG behaves similar to the uniform sampling policy in many cases and its empirical performance is undesirable. In the next subsection, we will propose a new approximate allocation policy.

Algorithm 1 Optimistic Knowledge Gradient

Input: Parameters of prior distributions for instances $\{a_i^0, b_i^0\}_{i=1}^K$ and the total budget T.
for $t = 0, \ldots, T-1$ **do**
  Select the next instance i_t to label according to:

$$i_t = \arg\max_{i \in \{1,\ldots,K\}} \left(R^+(a_i^t, b_i^t) \doteq \max(R_1(a_i^t, b_i^t), R_2(a_i^t, b_i^t)) \right).$$

  Acquire the label $y_{i_t} \in \{-1, 1\}$.
  if $y_{i_t} = 1$ **then**
    $a_{i_t}^{t+1} = a_{i_t}^t + 1, b_{i_t}^{t+1} = b_{i_t}^t$; $a_i^{t+1} = a_i^t, b_i^{t+1} = b_i^t$ for all $i \neq i_t$.
  else
    $a_{i_t}^{t+1} = a_{i_t}^t, b_{i_t}^{t+1} = b_{i_t}^t + 1$; $a_i^{t+1} = a_i^t, b_i^{t+1} = b_i^t$ for all $i \neq i_t$.
  end if
end for
Output: The positive set $H_T = \{i : a_i^T \geq b_i^T\}$.

3.2. Optimistic Knowledge Gradient

The stage-wise reward $R(a, b)$ can be viewed as a random variable with a two point distribution, i.e., with the probability $p_1 = \frac{a}{a+b}$ of being $R_1(a, b)$ and the probability $p_2 = \frac{b}{a+b}$ of being $R_2(a, b)$. The KG policy selects the instance with the largest *expected* reward. However, it is not consistent. A simple idea is to select the instance based on the optimistic outcome of the reward, i.e., the instance with the largest $R^+(a, b) = \max(R_1(a, b), R_2(a, b))$. The policy, which is named as *optimistic knowledge gradient*, is presented in Algorithm 1.

Theoretically, the optimistic KG policy is consistent in our problem as shown in the next theorem with the proof in Appendix.

Theorem 3.2 *Assuming that a_i^0 and b_i^0 are positive integers, the optimistic KG is a consistent policy, i.e, as T goes to infinity, the accuracy will be* 100% *(i.e., $H_T = H^*$) almost surely.*

Computationally, the optimistic KG has the time complexity $\mathcal{O}(KT)$ and space complexity $\mathcal{O}(K)$, both of which are smaller in magnitude than that of the approximate Gittins index rule.

The proposed optimistic KG is motivated by a more general framework, called conditional value-at-risk (CVaR) (Rockafellar & Uryasev, 2002). In particular, for a random variable X with the support $\mathcal{X}$ (e.g., our random reward with the two point distribution), let α-

quantile function be denoted as $Q_X(\alpha) = \inf\{x \in \mathcal{X} : \alpha \leq F_X(x)\}$, where $F_X(\cdot)$ is the CDF of X. The value-at-risk is defined as: $\text{VaR}_\alpha(X) = Q_X(1-\alpha)$ and the conditional value-at-risk, $\text{CVaR}_\alpha(X)$, is the expected reward exceeding (or equal to) $\text{VaR}_\alpha(X)$. As shown in (Rockafellar & Uryasev, 2002), $\text{CVaR}_\alpha(X)$ can be expressed as:

$$\text{CVaR}_\alpha(X) = \max_{\{q_1 \geq 0, q_2 \geq 0\}} q_1 R_1 + q_2 R_2,$$
$$\text{s.t.} \quad q_1 \leq \frac{1}{\alpha} p_1,\ q_2 \leq \frac{1}{\alpha} p_2,\ q_1 + q_2 = 1.$$

In our problem, when $\alpha = 1$, $\text{CVaR}_\alpha(X) = p_1 R_1 + p_2 R_2$, which is the expected reward; when $\alpha \to 0$, $\text{CVaR}_\alpha(X) = \max(R_1, R_2)$, which is used as the selection criterion in optimistic KG. In fact, a more general policy could be selecting the next instance with the largest $\text{CVaR}_\alpha(X)$ with a tuning parameter $\alpha \in [0, 1]$. We can extend Theorem 3.2 to prove that the policy based on $\text{CVaR}_\alpha(X)$ is consistent for any $\alpha < 1$. Since our MDP formulation is essentially a multi-armed Bayesian bandit (MAB), the proposed optimistic KG and CVaR based KG could be adopted as general index policies for solving Bayesian MAB.

4. Incorporating Worker Reliability

In many crowdsourcing applications, it is important to model worker reliability. Assuming that there are M workers, we can capture the reliability of the j-th worker by introducing an extra parameter $\rho_j \in [0, 1]$ as in (Dawid & Skene, 1979; Raykar et al., 2010; Karger et al., 2012), which is defined as the probability of getting the same label as the one from a random noiseless (perfectly reliable) worker. Let Y_i be the label for the i-th instance from a random noiseless worker and Z_{ij} be the label provided by the j-th worker for the i-th instance. Then $\rho_j = \Pr(Z_{ij} = Y_i | Y_i)$ and

$$\begin{aligned}\Pr(Z_{ij} = 1) =& \Pr(Z_{ij} = 1 | Y_i = 1) \Pr(Y_i = 1) + \\ & \Pr(Z_{ij} = 1 | Y_i = -1) \Pr(Y_i = -1) \\ =& \rho_j \theta_i + (1 - \rho_j)(1 - \theta_i). \end{aligned} \tag{13}$$

This model is often called *one-coin* model. We note that the previous simplified model is a special case of the one-coin model with $\rho_j = 1$ for all j, i.e., assuming that every worker is *perfectly reliable*.

We assume that ρ_j is also drawn from a Beta prior distribution: $\rho_j \sim \text{Beta}(c_j^0, d_j^0)$. At each stage t, we need to make the decision on both the next instance i to be labeled and the next worker j to label the instance i (we omit t in i, j here for notation simplicity). In other words, the action space $\mathcal{A} = \{(i, j) : (i, j) \in \{1, \ldots, K\} \times \{1, \ldots, M\}\}$. Once the decision is made, we observe the label 1 with the probability $\Pr(Z_{ij} = 1 | \theta_i, \rho_j) = \theta_i \rho_j + (1 - \theta_i)(1 - \rho_j)$ and -1 with $\Pr(Z_{ij} = -1 | \theta_i, \rho_j) = (1 - \theta_i)\rho_j + \theta_i(1 - \rho_j)$, which is the transition probability. Although the likelihood $\Pr(Z_{ij} = z | \theta_i, \rho_j)$ $(z \in \{-1, 1\})$ can be explicitly written out, the product of the Beta priors of θ_i and ρ_j is no longer the conjugate prior of our likelihood and we need to approximate posterior distribution. In particular, we adopt the variational approximation by assuming the conditional independence of θ_i and ρ_j: $p(\theta_i, \rho_j | Z_{ij} = z) \approx p(\theta_i | Z_{ij} = z) p(\rho_j | Z_{ij} = z)$. We further approximate $p(\theta_i | Z_{ij} = z)$ and $p(\rho_j | Z_{ij} = z)$ by two Beta distributions whose parameters are computed using the moment matching. Due to the Beta distribution approximation of $p(\theta_i | Z_{ij} = z)$, the reward function takes a similar form as in the previous setting and the corresponding approximate policies (e.g., KG, optimistic KG) can be directly applied. The detailed derivations and the optimistic KG algorithm with worker reliability are provided in Appendix.

We can further extend it to a more complex *two-coin* model (Dawid & Skene, 1979; Raykar et al., 2010) by introducing a pair of parameters (ρ_{j1}, ρ_{j2}) to model the j-th worker's reliability: $\rho_{j1} = \Pr(Z_{ij} = Y_i | Y_i = 1)$ and $\rho_{j2} = \Pr(Z_{ij} = Y_i | Y_i = -1)$.

5. Extensions

Our MDP formulation is a general framework to address many complex settings of sequential budget allocation problems in crowdsourcing. In particular, when the feature information $\mathbf{x}_i$ for each instance i is available, we could utilize it by assuming $\theta_i = \sigma(\langle \mathbf{w}, \mathbf{x}_i \rangle) \doteq \frac{\exp\{\langle \mathbf{w}, \mathbf{x}_i \rangle\}}{1 + \exp\{\langle \mathbf{w}, \mathbf{x}_i \rangle\}}$, where $\mathbf{w}$ is drawn from a Gaussian prior $N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$. The posterior $\boldsymbol{\mu}_{t+1}$ and $\boldsymbol{\Sigma}_{t+1}$ can be updated using the Laplace method as in Bayesian logistic regression (Bishop, 2007).

In multi-class labeling problems, the i-th instance is associated with a probability vector $\boldsymbol{\theta}_i = (\theta_{i1}, \ldots \theta_{iC})$, where θ_{ic} is the probability that the i-th instance belongs to the class c and $\sum_{i=1}^{C} \theta_{ic} = 1$. By generalizing Beta distribution to the multivariate case, we assume that $\boldsymbol{\theta}_i$ has a Dirichlet prior $\boldsymbol{\theta}_i \sim \text{Dir}(\boldsymbol{\alpha}_i^0)$. Then we can formulate the problem into a Bayesian MDP and apply the optimistic KG. We can further use Dirichlet distribution to model worker reliability as in (Liu & Wang, 2012). The detailed derivations of these extensions are presented in Appendix.

6. Related Work

To address new challenges in crowdsourcing problems, many research work has been done. Most of them are solving a static problem, i.e., inferring true la-

bels and worker reliability based on a static labeled dataset (Dawid & Skene, 1979; Raykar et al., 2010; Liu & Wang, 2012; Welinder et al., 2010; Whitehill et al., 2009; Bachrach et al., 2012; Zhou et al., 2012; Liu et al., 2012). The first work that incorporates diversity of worker reliability is (Dawid & Skene, 1979), which uses EM to perform the point estimation on both worker reliability and true class labels. Based on that, Raykar et al. (2010) extended (Dawid & Skene, 1979) into Bayesian framework and Liu & Wang (2012) further introduced Dirichlet prior for modeling worker reliability in multi-class settings. Our work utilizes the modeling techniques in these two static models as basic building blocks but extends to dynamic budget allocation settings.

In recent years, there are several works that have been devoted into online learning or budget allocation in crowdsourcing (Karger et al., 2012; Ertekin et al., 2012; Yan et al., 2011; Pfeiffer et al., 2012). The method proposed in (Karger et al., 2012) is based on the one-coin model. In particular, it assigns instances to workers according to a random bipartite (l, r)-regular graph. Although the error rate method is proved to achieve the minimax rate, its analysis is asymptotic and method is not optimal when the budget is limited. For other methods, Pfeiffer et al. (2012) failed to model the worker reliability in the allocation process. Yan et al. (2011) required the feature information for the decision problem. Basically, none of the existing methods has characterized the *optimal* allocation policy for finite budget.

We also note that the budget allocation in crowdsourcing is fundamentally different from noisy active learning (Settles, 2009; Nowak, 2009). Active learning usually does not model the variability of labeling difficulties and assumes a single (noisy) oracle; while in crowdsourcing, we need to model both labeling difficulties for instances and different worker reliability. Secondly, active learning requires the feature vectors for the decision, which could be unavailable for crowdsourcing. Finally, the goal of the active learning is to label as few instances as possible to learn a good classifier. In contrast, for budget allocation in crowdsourcing, the goal is to infer the true labels for as many instances as possible.

7. Experiments

We conduct empirical study to compare our optimistic KG (Opt-KG) policy with several competitors as follows. For all experiments, we start from the uniform prior $\text{Beta}(1, 1) = \text{Unif}[0, 1]$ for each θ_i.

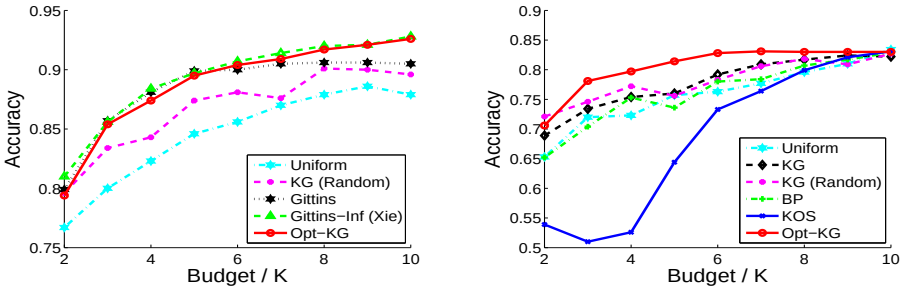
1. Uniform: Uniform sampling.
2. Gittins: Approximate finite-horizon Gittins index rule computed using the calibration method (Nino-Mora, 2011).
3. Gittins-Inf (Xie): Another policy proposed in (Xie & Frazier, 2012) for solving the infinite-horizon problem where the reward is discounted by α. Although it solves a different problem, we apply it as a heuristic by choosing α such that $T = 1/(1-\alpha)$.
4. KG / KG(Random): Deterministic KG or randomized KG (Frazier et al., 2008).
5. KOS: The randomized budget allocation algorithm by (Karger et al., 2012). Unlike the original algorithm in their paper, we normalized the messages. Without the normalization, KOS could preform incredibly poor (Liu et al., 2012).
6. BP: Random sampling with the labeling aggregation method based on the belief propagation (BP) in (Liu et al., 2012).

We note that both Gittins and Gittins-Inf polices can not be applied when the worker reliability is incorporated as in Section 4.

7.1. Simulated Study

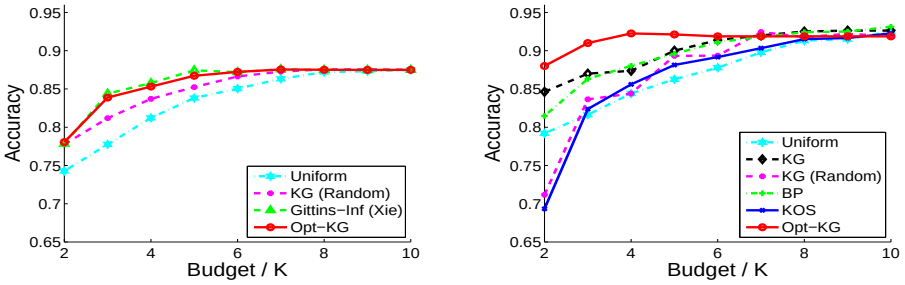
We first test the Opt-KG policy for the basic setting where labels are aggregated via majority vote without incorporating worker reliability. In particular, we assume $K = 50$ and generate 20 different sets of $\{\theta_i\}_{i=1}^K$. We vary the total budget $T = 2K, 3K, \ldots, 10K$, and report the mean and standard deviation of the accuracy over different sets of $\{\theta_i\}$ in Figure 2(a). The x-axis is the ratio between the budget T and the number of instances K. We note that since each θ_i is generated from uniform prior, the variance of the accuracy is quite large. For better visualization, the deviation in Figure 2 is 0.2 standard deviation. For KG policy, we only plot the accuracy for the randomized policy since we have proved that the deterministic KG will consistently sample one instance in Proposition 3.1. As we can see from Figure 2(a), our method and infinite-horizon Gittins perform better than the other three policies as the budget level increases. Although the infinite-horizon Gittins index performs slightly better than our method, it requires solving a linear system with $\mathcal{O}(T^2)$ variables at each stage, which could be too expensive for large-scale applications. While our Opt-KG policy has a time complexity linear in KT and space complexity linear in K, which is much more efficient when a quick online decision is required.

We also simulate worker reliability $\rho_j \sim \text{Beta}(4, 1)$ for $j = 1, \ldots, 10$ and compare different policies in Figure 2(b) over 20 simulations. As we can see, Opt-KG still performs the best. We also point out that, under the one-coin model, the deterministic KG policy will no



(a) Majority Vote (b) One-Coin Model

Figure 2. Performance comparison on simulated data.



(a) RTE: Majority Vote (b) RTE: One-Coin Model

Figure 3. Performance comparison on real datasets.

longer only sample one instance as T goes large and becomes a reasonably good policy.

7.2. Real Data

We compare different policies on a standard real dataset for recognizing textual entailment (RTE) (Section 4.3 in (Snow et al., 2008)). There are 800 instances and each instance is a sentence pair. Each sentence pair is presented to 10 different workers to acquire binary choices of whether the second hypothesis sentence can be inferred from the first one. There are in total 164 different workers. We first consider our simpler setting without incorporating the diversity of workers. Therefore, once we decide to label an instance, we randomly choose an worker (who provides the label in the full dataset) to acquire the label. Due to this randomness, we run each policy 20 times and report the errorbar of the accuracy in Figure 3(a). As we can see, Opt-KG and infinite-horizon Gittins index policy still perform better than the others. Note that, different from the simulated study, the deviation in error bar here is the standard deviation. We omit the finite-horizon Gittins index rule due to its unaffordable computational complexity.

When the worker reliability is incorporated, we compare different policies in Figure 3(b). We put Beta$(4, 1)$ prior distribution for each ρ_j which indicates that we have the prior belief that most workers perform reasonably well and the averaged accuracy is $4/5 = 80\%$. As we can see, the accuracy of Opt-KG is much higher than that of other policies when T is small. It achieves the highest accuracy of 92.25% only using 40% of the total budget (i.e., on average, each instance is only labeled by 4 times). Another interesting observation is that when the budget is very large (e.g., more than 8K), other policies (e.g., KG, Random+BP) achieve slightly higher accuracy than Opt-KG. This is mainly due to the restrictiveness of the real experimental setting. In particular, since the experiment is conducted on a fixed dataset, the Opt-KG cannot freely choose instance-worker pairs especially when the budget goes up (i.e., the action set is greatly restricted). Comparing Figure 3(b) to 3(a), we also observe that the Opt-KG policy under the one-coin model performs much better than the Opt-KG with the majority vote, which indicates that it is beneficial to incorporate worker reliability.

Acknowledgements

The authors would like to thank Qiang Liu for sharing the code of KOS and BP; and Jing Xie and Peter Frazier for sharing their code for infinite-horizon Gittins index. The authors would also like to thank Leon Bottou, Jenn Wortman Vaughan and Chien-Ju Ho for helpful discussions. This work was done when the first two authors were interned at Microsoft Research.

References

Bachrach, Y., Minka, T., Guiver, J., and Graepel, T. How to grade a test without knowing the answers - a Bayesian graphical model for adaptive crowdsourcing and aptitude testing,. In *ICML*, 2012.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2007.

Dawid, A. P. and Skene, A. M. Maximum likelihood estimation of observer error-rates using the em algorithm. *JRSS-C*, 28:20–28, 1979.

Ertekin, S., Hirsh, H., and Rudin, C. Wisely using a budget for crowdsourcing. Technical report, MIT, 2012.

Frazier, P., Powell, W. B., and Dayanik, S. A knowledge-gradient policy for sequential information collection. *SIAM J. Control Optim.*, 47(5): 2410–2439, 2008.

Gittins, J. C. *Multi-armed Bandit Allocation Indices*. John Wiley & Sons, 1989.

Gupta, S. S. and Miescke, K.J. Bayesian look ahead one stage sampling allocations for selection the largest normal mean. *J. of Stat. Planning and Inference*, 54(2):229–244, 1996.

Karger, D. R., Oh, S., and Shah, D. Budget-optimal task allocation for reliable crowdsourcing systems. arXiv:1110.3564v3, 11 2012.

Liu, C. and Wang, Y. M. Truelabel + confusions: A spectrum of probabilistic models in analyzing multiple ratings. In *ICML*, 2012.

Liu, Q., Peng, J., and Ihler, A. Variational inference for crowdsourcing. In *NIPS*, 2012.

Nino-Mora, J. Computing a classic index for finite-horizon bandits. *INFORMS Journal on Computing*, 23(2):254–267, 2011.

Nowak, R. D. Noisy generalized binary search. In *NIPS*, 2009.

Pfeiffer, T., Gao, X. A., Mao, A., Chen, Y., and Rand, D. G. Adaptive polling for information aggregation. In *AAAI*, 2012.

Powell, W. B. *Approximate Dynamic Programming: solving the curses of dimensionality*. John Wiley & Sons, 2007.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 2005.

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., and Moy, L. Learning from crowds. *JMLR*, 11:1297–1322, 2010.

Robert, Christian P. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer Verlag, 2007.

Rockafellar, R. T. and Uryasev, S. Conditional value-at-risk for general loss distributions. *J. of Banking and Finance*, 26:1443–1471, 2002.

Settles, B. Active learning literature survey. Technical report, University of Wisconsin–Madison, 2009.

Snow, R., Connor, B. O., Jurafsky, D., and Ng., A. Y. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP*, 2008.

Welinder, P., Branson, S., Belongie, S., and Perona, P. The multidimensional wisdom of crowds. In *NIPS*, 2010.

Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., and Movellan, J. R. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, 2009.

Xie, J. and Frazier, P. I. Sequential bayes-optimal policies for multiple comparions with a control. Technical report, Cornell University, 2012.

Yan, Y., Rosales, R., Fung, G., and Dy, J. Active learning from crowds. In *ICML*, 2011.

Zhou, D., Basu, S., Mao, Y., and Platt, J. Learning from the wisdom of crowds by minimax conditional entropy. In *NIPS*, 2012.