Crowdsourcing & Optimal Budget Allocation in Crowd Labeling

Madhav Mohandas, Richard Zhu, Vincent Zhuang

May 5, 2016

Table of Contents

Active learning

Semi-supervised learning!

- Learner makes queries and receives outputs, choosing which examples to learn from
- Effective when labels are expensive or difficult to obtain
- Examples:
  - Streaming-based active learning (IWAL, last Thursday)
  - Near-Optimal Bayesian Active Learning for Decision Making (NOBAL, this Tuesday)
- Fundamental assumption: $\exists$ "oracle" or perfect labeler
- Abstract away the process of actually obtaining data

Crowdsourcing

- In real life, no oracle exists
- Both passive/active learning require labeled data
- Data is naturally unlabeled
- Obtaining accurate labels is difficult and expensive
- Getting expert annotations is often infeasible

Crowdsourcing

- Query labels from the crowd (e.g. Amazon Mechanical Turk)
- Introduces further challenges
- Workers are unreliable...
- Maximize overall accuracy with the knowledge that workers are unreliable

Examples

Object labeling

Crowdsourcing issues

- Low-paid labeling ($\sim$\$0.01 / instance, \$1 - \$5 / hour) usually very noisy
- Get an instance labeled several times by different workers
- Infer the true label by some clever aggregation of labels (oftentimes domain-specific techniques)
- How do you accurately estimate labeling difficulty for a given instance?

Varying label quality

- How can we deal with varying quality of crowdsourced labels in algorithms?
- Use redundancy and aggregation
- Naive: Majority voting heuristic (error-prone, all annotators are not equal)
- 2nd level: Generative probabilistic models, inference (EM algorithm)
- Modern day: Variational inference approaches
    - Belief propagation (BP)
    - Mean field (MF)
- Efficient and possess strong optimality guarantees

Budget Allocation vs. Active Learning

Both amenable to crowdsourcing, but seek to answer very different questions:

- Active learning: how many samples does it take for our algorithm to produce a good model?
- Budget allocation: given a fixed budget, how do we allocate it so that we get the best data/model possible?

Budget Allocation Problem

- Budget requires you to decide whether to spend more on ambiguous instances, or instead save money and label other instances.
- Balance exploration (labeling other instances) and exploitation (continue labeling ambiguous instances)
- Previous work on crowdsourcing addressed parts independently
- This paper addresses both budget allocation (selection of instances) and the label aggregation (inference of true label) in a general framework

Where we're going

- Bayesian statistics setup
- Problem formulated $\rightarrow$ finite-horizon Markov Decision Process
- Optimal budget allocation π obtained using DP
- Computationally intractable! State space $\propto \exp(T)$
- Introduce efficient approximate algorithm - optimistic knowledge gradient
- Experiments and applications

Problem Setting

- Binary labeling task
- K “coins” (instances) with true labels $Z_i \in \{-1, 1\}$, $1 \leq i \leq K$
- Positive set: $H^* = \{i : Z_i = 1\}$
- Goal is to predict true label of each instance using labels generated by crowd members

Problem Setting

- All workers are identical and noiseless (perfectly reliable)
- Accuracy of the worker label only depends on the inherent difficulty of categorizing the instance
- Ambiguity of instance i parameterized by $\theta_i \in [0, 1]$
- Equivalently, θ_i is the fraction of workers that will label instance i as $+1$ (ensemble average)
- In coin-flipping interpretation, θ_i is bias of the coin
- Assume soft-label is consistent with true label, i.e.

$$Z_i = 1 \iff \theta_i \geq 0.5$$

Problem Setting

Interpretation of θ_i. Adult $+1$, not adult -1:

Budget Allocation Challenge

- Total budget of T
- Costs 1 unit to get a worker to label some instance i
- How to spend money in a way that maximizes overall accuracy?

Budget Allocation Challenge

Can formulate as a finite horizon decision process:

- At each timestep $0 \leq t \leq T - 1$, choose an instance $i_t \in \mathcal{A} = \{1, \ldots, K\}$ to label, and receive label y_{i_t}. (Budget Allocation Phase)
- After final step make inference of true label, estimate positive set H and gain reward $|H \cap H^*| + |H^c \cap (H^*)^c|$. (Label Aggregation Phase)
- Best prediction strategy is majority vote (proof in paper), since all workers are assumed to be identical.
- This assumption relaxed later on
- Goal: determine optimal (adaptive) budget allocation policy π to label instances

Simple Example

Here we assume no prior knowledge about the θ_is

Instance	1st round label	2nd round label
Instance 1 (θ_1)	1	1
Instance 2 (θ_2)	1	-1
Instance 3 (θ_3)	1	

What instance should we label next?

Simple Example

Instance	1st round label	2nd round label
Instance 1 (θ_1)	1	1
Instance 2 (θ_2)	1	-1
Instance 3 (θ_3)	1	

Consider instance 1.

- If we label instance 1, majority vote guarantees us to predict $\hat{Z}_i \to +1$.
- No improvement in expected accuracy, regardless of whether $\theta_i > 0.5$ or $\theta_i < 0.5$

Simple Example

Instance	1st round label	2nd round label
Instance 1 (θ_1)	1	1
Instance 2 (θ_2)	1	-1
Instance 3 (θ_3)	1	

Consider instance 2.

- If $\theta_2 > .5$, then true label for instance 2 will be 1
- Current expected accuracy: .5
- If received label is 1 ($P(y_2 = 1) = \theta_2$), accuracy will be 1. If received label is -1, accuracy will be 0.
- So, expected accuracy after receiving label is θ_2. Improvement is $\theta_2 - .5 > 0$

Simple Example

Instance	1st round label	2nd round label
Instance 1 (θ_1)	1	1
Instance 2 (θ_2)	1	-1
Instance 3 (θ_3)	1	

Consider instance 2.

- If $\theta_2 < .5$, then true label for instance 2 will be -1
- Current expected accuracy: .5
- If received label is 1 ($P(y_2 = 1) = \theta_2$), accuracy will be 0. If received label is -1, accuracy will be 1.
- So, expected accuracy after receiving label is $1 - \theta_2$. Improvement is $0.5 - \theta_2 > 0$

Simple Example

Instance	1st round label	2nd round label
Instance 1 (θ_1)	1	1
Instance 2 (θ_2)	1	-1
Instance 3 (θ_3)	1	

Consider instance 3.

- If $\theta_3 > .5$, then true label for instance 3 will be $+1$
- Current expected accuracy: 1
- If received label is 1 ($P(y_3 = 1) = \theta_3$), accuracy will be 1. If received label is -1, accuracy will be 0.5
- So, expected accuracy after receiving label is $\theta_3 + 0.5(1 - \theta_3)$. Improvement is $0.5(\theta_3 - 1) < 0$

Simple Example

Instance	1st round label	2nd round label
Instance 1 (θ_1)	1	1
Instance 2 (θ_2)	1	-1
Instance 3 (θ_3)	1	

Consider instance 3.

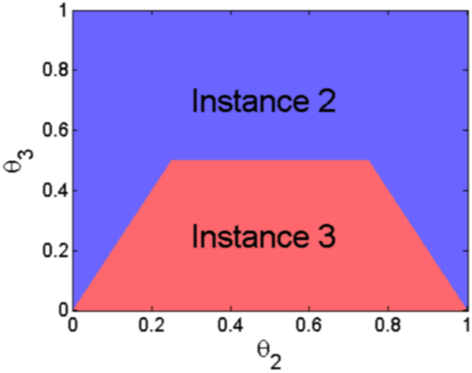
- If $\theta_3 < .5$, then true label for instance 3 will be -1
- Current expected accuracy: 0
- If received label is 1 ($P(y_3 = 1) = \theta_3$), accuracy will be 0. If received label is -1, accuracy will be 0.5
- So, expected accuracy after receiving label is $0.5(1 - \theta_3)$. Improvement is $0.5(1 - \theta_3) > 0$

Simple Example

	Current Accuracy	$y=1$	$y=-1$	Expected Accuracy	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	$0.5(1 - \theta_3)$	$0.5(1 - \theta_3) > 0$

- Expected increase in accuracy for instance *i* is a function of the true value of θ_i
- No optimal choice in frequentist setting!

Simple Example



Decisi

Boundary

Simple Example

	Current Accuracy	$y = 1$	$y = -1$	Expected Accuracy	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	$0.5(1 - \theta_3)$	$0.5(1 - \theta_3) > 0$

- Originally made no assumptions on distributions of θ_i
- But if we assume θ_i are sampled from some prior distribution, we can compute the expected value of this improvement
- Optimal choice exists in Bayesian setting!

Bayesian Modeling

- K coin-flipping problem!
- The label $y_{i_t} \in \{-1, 1\}$ must follow $y_{i_t} \sim Bernoulli(\theta_{i_t})$
- We assumed all workers to be identical, so y_{i_t} depends only on θ_{i_t}
- Prior to any labels being collected, we assume that $\theta_i \sim Unif(0, 1) = Beta(1, 1)$

- Define a_i^t / b_i^t as $1+$ the number of positive/negative labels we have gathered for instance i before time t. $a_i^0 = b_i^0 = 1$

Bayesian Modeling

- Define state of the decision process at (the start of) timestep t as $S^t = \{a_i^t, b_i^t\}_{i=1}^K$ (K x 2 matrix)
- If we request a label for instance i_t in state S^t, our state will be updated according to:

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

- e_{i_t} is a K x 1 vector with 1 at the i_t-th entry and 0 at all other entries

Bayesian Modeling

Recall:
$y_i \sim \textit{Bernoulli}(\theta_i)$
$\theta_i \sim \textit{Unif}(0, 1)$

So:

$$\begin{aligned} P(\theta_i | S^t) &= \frac{P(S^t|\theta_i) * P(\theta_i)}{P(S^t)} \\ &\propto P(S^t|\theta_i) * P(\theta_i) \\ &\propto \theta_i^{a_i^t - 1} * (1 - \theta_i)^{b_i^t - 1} \\ &= \textit{Beta}(a_i^t, b_i^t) \end{aligned}$$

Bayesian Modeling

Given state S^t and instance choice i_t, can transition to two different states S^{t+1} according to:

$$\Pr(y_{i_t} = 1 | S^t, i_t) = \mathbb{E}(\theta_{i_t} | S^t) = \frac{a^t_{i_t}}{a^t_{i_t} + b^t_{i_t}} \quad \text{and} \quad \Pr(y_{i_t} = -1 | S^t, i_t) = \frac{b^t_{i_t}}{a^t_{i_t} + b^t_{i_t}}.$$

7

Markov Decision Process

Starting to sound like a MDP:

- State: S^t, Action: instance i_t to label
- Next state is determined completely from previous state and action, according to transition probabilities
- Must develop notion of reward of taking action i_t in state S^t

Reward function

Define:
$P_i^t = P(\theta_i \geq .5 | S_i^t) = P(\theta_i \geq .5 | a_i^t, b_i^t)$
$h(x) = max(x, 1 - x)$

Then:
$h(P_i^t) =$ expected accuracy of the prediction (majority vote) for the label of instance i at time t

Reward function

$$R(S^t, i_t) = \mathbf{E}(h(P_{i_t}^{t+1}) - h(P_{i_t}^t)|S^t, i_t)$$

Reward for choosing action i_t in state S_t is the expected gain in accuracy of classifying instance i after receiving y_{i_t}.

Value function

$$V_{t_0}(S^{t_0}) = \sup_{\pi} \mathbf{E}^{\pi}\left(\sum_{t=t_0}^{T-1} R(S^t, i_t)\right)$$

Value function

$$V(S^{t_0}) = \sup_{\pi} \mathbf{E}^{\pi}\left(\sum_{t=t_0}^{T-1} R(S^t, i_t)\right)$$

$$\begin{aligned} & V_t(S^t) \\ = & \max_i \Big(R(S^t, i) + \Pr(y_i = 1 | S^t, i) V_{t+1}\left(S^t + (\mathbf{e}_i, \mathbf{0})\right) + \Pr(y_i = -1 | S^t, i) V_{t+1}\left(S^t + (\mathbf{0}, \mathbf{e}_i)\right) \Big), \end{aligned}$$

Value function

$$V(S^{t_0}) = \sup_{\pi} \mathbf{E}^{\pi}\left(\sum_{t=t_0}^{T-1} R(S^t, i_t)\right)$$

$$V_t(S^t) = \max_i \Big(R(S^t, i) + \Pr(y_i = 1 | S^t, i) V_{t+1}\left(S^t + (\mathbf{e}_i, \mathbf{0})\right) + \Pr(y_i = -1 | S^t, i) V_{t+1}\left(S^t + (\mathbf{0}, \mathbf{e}_i)\right) \Big),$$

- If we know the value of every state, optimal policy follows
- Can derive value function using dynamic programming

Dynamic programming

- Downside: Number of possible states $|\mathcal{S}^t|$ grows exponentially in t. (State is reachable at time t if $\sum_{i=1}^{K}(a_i^t + b_i^t) - (a_i^0 + b_i^0) = t$)
- Will develop computationally efficient approximate algorithm

Notation: $R(S^t, i_t) = R(a_{i_t}^t, b_{i_t}^t)$
$R(a_{i_t}^t, b_{i_t}^t) = p_1 R_1(a_{i_t}^t, b_{i_t}^t) + p_2 R_2(a_{i_t}^t, b_{i_t}^t)$

R_1/R_2 is the reward if the received label y_{i_t} is $1/-1$
p_1 and p_2 are the transition probabilities

MAB Approximation

Reformulate as finite-horizon Bayesian MAB problem:

- Arms are K instances
- Rewards R_1, R_2 are provided i.i.d from a fixed set of Bernoulli distributions

Bandit algorithms offer approximate solutions in the finite horizon

- e.g. Gittins index

Our ideal algorithm

Ideally, we want an approximation algorithm that is:

- computationally efficient
  - Gittins index is either $O(T^3)$ or $O(T^6)$, depending on if we want the exact index or not
- consistent
  - i.e. obtains 100% accuracy a.s. as $T \to \infty$

Knowledge Gradient

- Recall that we have two possible rewards for any given instance at each iteration:

$$R_1(a,b) = h(I(a+1,b)) - h(I(a,b))$$
$$R_2(a,b) = h(I(a,b+1)) - h(I(a,b))$$

- Greedily picks instance with largest expected reward:

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

- What happens if there is a tie?
    - Deterministic KG: pick smallest i
    - Randomized KG: pick randomly

Behavior of KG

- Deterministic KG is inconsistent
- Randomized KG performs badly empirically (behaves similarly to uniform sampling)

Optimistic Knowledge Gradient

- Optimistic KG greedily selects largest *optimistic* reward

$$i_t = \arg\max_i \left[R^+(a_i, b_i) = \max\left(R_1(a_i, b_i), R_2(a_i, b_i)\right) \right]$$

- Runtime $O(KT)$
- Consistent!

Algorithm

Init T and prior parameters $\{a_i^0, b_i^0\}_{i=1}^K$

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

  Select the next instance i_t to label according to

$$i_t = \arg\max_i \left[R^+(a_i, b_i) = \max\left(R_1(a_i, b_i), R_2(a_i, b_i)\right)\right]$$

  Acquire label $y_i \in \{-1, 1\}$

  if $y_{i_t} = 1$ **then**

    $a_{i_t}^{t+1} = a_{i_t}^t + 1$, $b_{i_t}^{t+1} = b_{i_t}^t$

  else

    $a_{i_t}^{t+1} = a_{i_t}^t$, $b_{i_t}^{t+1} = b_{i_t}^t + 1$

  end if

end for

Consistency

Theorem

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.

Properties of R

Can write out R, R^+ explicitly using Beta distribution. For example, if $a > b$, then $R^+(a, b) = R_1(a, b) = \frac{0.5^{a+b}}{aB(a,b)}$.

Lemma

1. *$R(a, b)$ is symmetric, i.e. $R^+(a, b) = R^+(b, a)$*
2. $\lim_{a\to\infty} R^+(a, a) = 0$
3. *For any fixed $a \geq 1$, $R^+(a + k, a - k) = R^+(a - k, a + k)$ is monotonically decreasing in k for $k = 0, \ldots, a - 1$*
4. *When $a \geq b$, for any fixed b, $R^+(a, b)$ is monotonically decreasing in a. By the symmetry of $R^+(a, b)$, when $b \geq a$, for any fixed a, $R^+(a, b)$ is monotonically decreasing in b.*

Every instance labelled infinitely many times as $T \to \infty$!

- Let $\mathcal{I}(\omega)$ be set of all instances labelled only finitely many times for some infinite sample path ω
- Must exist some T' after which no instances in $\mathcal{I}$ will be labelled
- For any instance $j \notin \mathcal{I}$, $R^+(a_j, b_j) \to 0$ (from lemma)
- Optimistic KG will select an instance in $\mathcal{I}$ next, contradiction!

- Let $\eta_i(T) = a_i^T + b_i^T$ be number of times instance i has been picked before step T.
- Since each instance will be labelled infinitely many times as $T \to \infty$, by the Strong Law of Large Numbers,

$$\lim_{T \to \infty} \frac{a_i^T - b_i^T}{\eta_i(T)} = 2\theta_i - 1$$

Recall that accuracy Acc(T) is defined as

$$\frac{1}{K}|H \cap H^*| + |H^c \cap (H^*)^c|$$

where $H_T = \{i : a_i^T \geq b_i^T\}$ and $H^* = \{i : \theta_i \geq 0.5\}$.
Then:

$$
\begin{aligned}
&Pr\left(\lim_{T\to\infty} \text{Acc}(T) = 1 \mid \{\theta_i\}_{i=1}^K\right) \\
=&Pr\left(\lim_{T\to\infty} \frac{1}{K}|H \cap H^*| + |H^c \cap (H^*)^c| = 1 \mid \{\theta_i\}_{i=1}^K\right) \\
\geq&Pr\left(\lim_{T\to\infty} \frac{a_i^T - b_i^T}{\eta_i(T)} = 2\theta - 1 \mid \{\theta_i\}_{i=1}^K\right) \\
=&1
\end{aligned}
$$

Take expectation over all $\theta \neq 0.5$, since $\{\theta_i | \theta_i = 0.5\}$ has measure zero:

$$
\begin{aligned}
& Pr\left(\text{Acc}(T) = 1\right) \\
=& \mathbb{E}_{\{\theta_i : \theta_i \neq 0.5\}_{i=1}^K} \left[Pr\left(\lim_{T \to \infty} \text{Acc}(T) = 1 \mid \{\theta_i\}_{i=1}^K \right) \right] \\
=& \mathbb{E}_{\{\theta_i : \theta_i \neq 0.5\}_{i=1}^K} [1] = 1
\end{aligned}
$$

Recap

Basic intuition:

- Optimistic KG adequately explores all instances
- Given enough samples, we converge to the true θ's

Worker Reliability

- Suppose we have *M* unreliable workers
- Let ρ_j be the probability of worker *j* getting the same label as a perfectly reliable noiseless worker and Z_{ij} be the label we receive from worker *j* on instance *i*
- Then we can parameterize the probability by ρ and θ:

$$Pr(Z_{ij} = 1) = \rho_j \theta_i + (1 - \rho_j)(1 - \theta_i)$$

- Assume Beta prior for ρ as well

- Note that the posterior is no longer a product of beta distributions
- Assume posterior factorizes (conditional independence):

$$\begin{aligned} p(\theta_i, \rho_j | Z_{ij} = z) &\approx p(\theta_i | Z_{ij} = z) p(\rho_j | Z_{ij} = z) \\ &\approx Beta(a_i(z), b_i(z)) Beta(c_j(z), d_j(z)) \end{aligned}$$

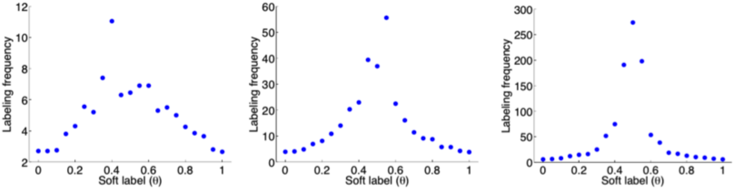
- Must find $a_i(z), b_i(z), c_j(z), d_j(z)$ using variational approximation (moment matching on marginals)

Extensions

- features: apply Bayesian updates onto weights
- multiclass labelling: Dirichlet prior

Experiments

First, consider identical, noiseless workers
$K = 21$ instances, $(\theta_1, \theta_2, \theta_3 ... \theta_K) = (0, .05, .1, ...1)$



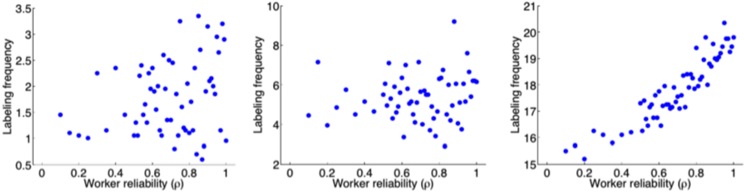
(a) $T = 5K = 105$ (b) $T = 15K = 315$ (c) $T = 50K = 1050$

Figure 4: Labeling counts for instances with different levels of ambiguity.

- Averaged over 20 runs
- In general, more ambiguous instances get labeled more, but most ambiguous instance may not get most labels
- As budget grows, algorithm considers increasingly ambiguous instances

Experiments

Now add $M = 59$ workers with:
$(\rho_1 ... \rho_M) = (.1 ..., .995)$



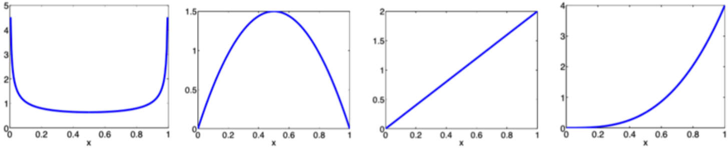
(a) $T = 5K = 105$ (b) $T = 15K = 315$ (c) $T = 50K = 1050$

Figure 5: Labeling counts for workers with different levels of reliability.

As the budget increases, more reliable workers get more assignments.

Experiments

How robust is Opt-KG under incorrect assumption of prior for θ?



(a) Beta(0.5, 0.5) (b) Beta(2, 2) (c) Beta(2, 1) (d) Beta(4, 1)

Figure 6: Density plot for different Beta distributions for generating each θ_i.

Experiments

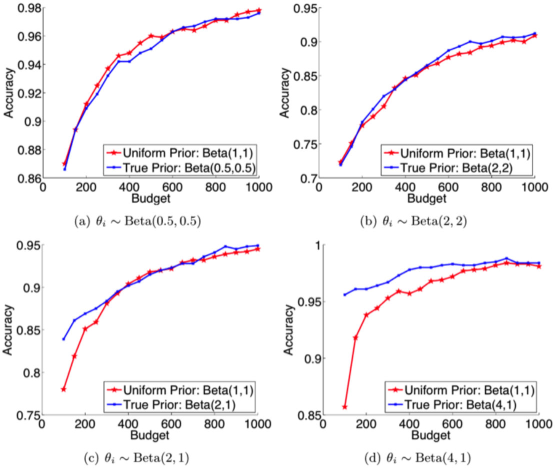


Figure 7: Comparison between Opt-KG using the uniform distribution and true generating distribution as the prior.

Experiments

How does the accuracy of Opt-KG compare to the following algorithms:

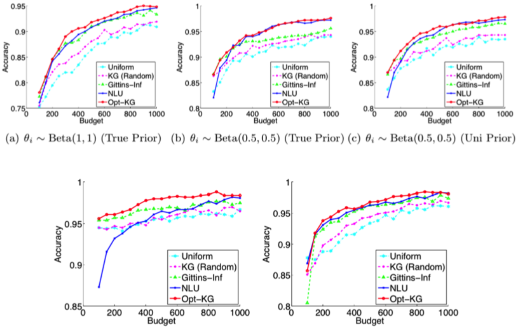
- **Uniform**: Uniform sampling
- **KG(Random)**: Randomized knowledge gradient (Frazier et al., 2008)
- **Gittins-Inf**: Gittins-indexed based policy for solivng infinite-horizon MAB problem with reward discounted by δ (Xie and Frazier, 2013)
- **NLU**: The ”new labeling uncertainty” method (Ipeirotis et al., 2013)

Experiments

Simulated Setting:

- Identical and noiseless workers
- $K = 50$ instances, with each $\theta_i \sim Beta(1, 1)$, $\theta_i \sim Beta(.5, .5)$, $\theta_i \sim Beta(4, 1)$
- Results show average of 20 runs on independently generated sets of $\{\theta_i\}_{i=1}^{K}$

Experiments



(a) $\theta_i \sim \text{Beta}(1,1)$ (True Prior) (b) $\theta_i \sim \text{Beta}(0.5,0.5)$ (True Prior) (c) $\theta_i \sim \text{Beta}(0.5,0.5)$ (Uni Prior)

(h) $\theta_i \sim \text{Beta}(4,1)$ (True Prior) (i) $\theta_i \sim \text{Beta}(4,1)$ (Uni Prior)

Figure 10: Performance comparison under the homogeneous noiseless worker setting.

Experiments

Test using real data set for recognizing textual entailment (RTE)

RTE task:

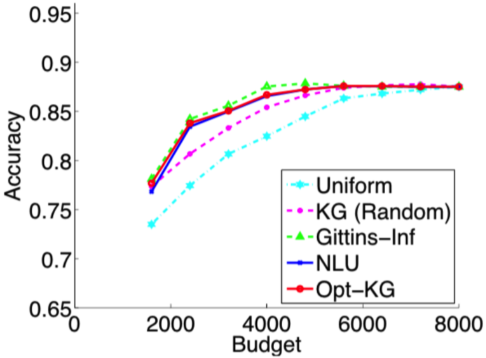
- Given ordered pair of sentences (s_1, s_2)
- Return 1 if s_2 can be inferred from s_1
- Example:
  - s_1: If you help the needy, God will reward you.
  - s_2: Giving money to a poor man has good consequences.

Experiments

Data set:

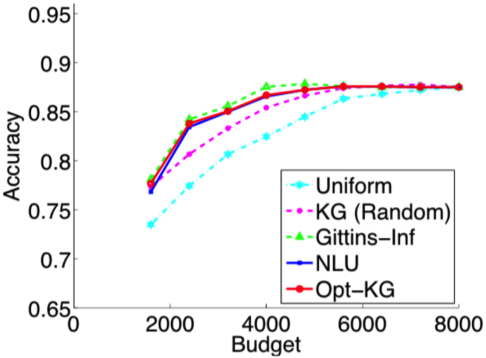
- 800 instances (sentence pairs)
- Each instance labeled by 10 different workers
- 164 different workers
- First, consider homogeneous noiseless workers
- Results show average of 20 independent trials for each policy

Experiments



(a) RTE: Homogeneous Noiseless Worker

Experiments



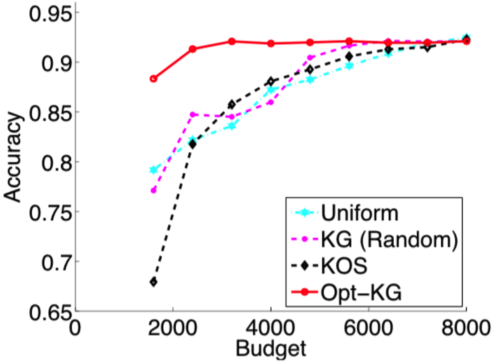
(a) RTE: Homogeneous Noiseless Worker

Budget T	$2K = 1,600$	$4K = 3,200$	$6K = 4,800$	$10K = 8,000$
Opt-KG	1.09	2.19	3.29	5.48
Gittins-inf	25.87	35.70	45.59	130.68

Table 4: Comparison in CPU time (seconds)

Experiments

Now incorporate worker reliabilities: $\rho_j \sim \mathit{Beta}(4,1)$ (i.e. workers behave reliably 80% of the time).



(b) RTE: Heterogeneous Worker

References

- Chen, Xi, Qihang Lin, and Dengyong Zhou. “Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing.” *Proceedings of the 30th International Conference on Machine Learning (ICML-13).* 2013.