

# On Top-k Selection in Multi-Armed Bandits and Hidden Bipartite Graphs

**Authored by:**

Jian Li  
Wei Cao  
Yufei Tao  
Zhize Li

## **Abstract**

This paper discusses how to efficiently choose from  $n$  unknown distributions the  $k$  ones whose means are the greatest by a certain metric, up to a small relative error. We study the topic under two standard settings—multi-armed bandits and hidden bipartite graphs—which differ in the nature of the input distributions. In the former setting, each distribution can be sampled (in the i.i.d. manner) an arbitrary number of times, whereas in the latter, each distribution is defined on a population of a finite size  $m$  (and hence, is fully revealed after  $m$  samples). For both settings, we prove lower bounds on the total number of samples needed, and propose optimal algorithms whose sample complexities match those lower bounds.

## **1 Paper Body**

This paper studies a class of problems that share a common high-level objective: from a number  $n$  of probabilistic distributions, find the  $k$  ones whose means are the greatest by a certain metric. Crowdsourcing. A crowdsourcing algorithm (see recent works [1, 13] and the references therein) summons a certain number, say  $k$ , of individuals, called workers, to collaboratively accomplish a complex task. Typically, the algorithm breaks the task into a potentially very large number of micro-tasks, each of which makes a binary decision (yes or no) by taking the majority vote from the participating workers. Each worker is given an (often monetary) reward for every micro-task that s/he participates in. It is therefore crucial to identify the most reliable workers that have the highest rates of making correct decisions. Because of this, a crowdsourcing algorithm should ideally be preceded by an exploration phase, which selects the best  $k$  workers from  $n$  candidates by a series of “control questions”. Every control-question must be paid for in the same way as a micro-task. The challenge is to find the

best workers with the least amount of money. Frequent Pattern Discovery. Let  $B$  and  $W$  be two relations. Given a join predicate  $Q(b, w)$ , the joining power of a tuple  $b \in B$  equals the number of tuples  $w \in W$  such that  $b$  and  $w$  satisfy  $Q$ . A top- $k$  semi-join [14, 17] returns the  $k$  tuples in  $B$  with the greatest joining power. This type of semijoins is notoriously difficult to process when the evaluation of  $Q$  is complicated, and thus unfriendly to tailored-made optimization. A well-known example from graph databases is the discovery of frequent patterns [14], where  $B$  is a set of graph patterns,  $W$  a set of data graphs, and  $Q(b, w)$  decides if a pattern  $b$  is a subgraph of a data graph  $w$ . In this case, top- $k$  semi-join essentially returns the set of  $k$  graph patterns most frequently found in the data graphs. Given a black box for resolving subgraph isomorphism  $Q(b, w)$ , the challenge is to minimize the number of calls to the black box. We refer to the reader to [14, 15] for more examples of difficult top- $k$  semi-joins of this sort.

#### 1.1 Problem Formulation

The paper studies four problems that capture the essence of the above applications. Multi-Armed Bandit. We consider a standard setting of stochastic multi-armed bandit selection. Specifically, there is a bandit with a set  $B$  of  $n$  arms, where the  $i$ -th arm is associated with a Bernoulli 1

distribution with an unknown mean  $\mu_i \in (0, 1]$ . In each round, we choose an arm, pull it, and then collect a reward, which is an i.i.d. sample from the arm's reward distribution. Given a subset  $V \subseteq B$  of arms, we denote by  $\mu_i(V)$  the arm with the  $i$ -th largest mean in  $V$ , and  $\mu_k(V)$  the mean of  $\mu_i(V)$ . Define  $\mu_{\text{avg}}(V) = \frac{1}{k} \sum_{i=1}^k \mu_i(V)$ , namely, the average of the means of the top- $k$  arms in  $V$ . Our first two problems aim to identify  $k$  arms whose means are the greatest either individually or aggregatively:

1 Problem 1 [Top- $k$  Arm Selection (k-AS)] Given parameters  $\epsilon \in (0, 1]$ ,  $\delta \in (0, 1]$ , and  $k \in \mathbb{N}$ , we want to select a  $k$ -sized subset  $V$  of  $B$  such that, with probability at least  $1 - \delta$ , it holds that  $\mu_i(V) \geq (1 - \epsilon)\mu_i(B)$ ,  $\forall i \in [k]$ . We further study a variation of k-AS where we change the multiplicative guarantee  $\mu_i(V) \geq (1 - \epsilon)\mu_i(B)$  to an additive guarantee  $\mu_i(V) \geq \mu_i(B) - \epsilon$ . We refer to the modified problem as Topkadd Arm Selection(kadd-AS). Due to the space constraint, we present all the details of kadd-AS in Appendix C. Problem 2 [Top-kavg Arm Selection (kavg-AS)] Given the same parameters as in k-AS, we want to select a  $k$ -sized subset  $V$  of  $B$  such that, with probability at least  $1 - \delta$ , it holds that  $\mu_{\text{avg}}(V) \geq (1 - \epsilon)\mu_{\text{avg}}(B)$ . For both problems, the cost of an algorithm is the total number of arms pulled, or equivalently, the total number of samples drawn from the arms' distributions. For this reason, we refer to the cost as the algorithm's sample complexity. It is easy to see that k-AS is more stringent than kavg-AS; hence, a feasible solution to the former is also a feasible solution to the latter, but not the vice versa. Hidden Bipartite Graph. The second main focus of the paper is the exploration of hidden bipartite graphs. Let  $G = (B, W, E)$  be a bipartite graph, where the nodes in  $B$  are colored black, and those in  $W$  colored white. Set  $n = |B|$  and  $m = |W|$ . The edge set  $E$  is hidden in the sense that an algorithm does not see any edge at the beginning. To find out whether an edge exists between a black vertex  $b$  and a white vertex  $w$ , the algorithm must perform a probe operation. The cost of the algorithm

equals the number of such operations performed. If an edge exists between  $b$  and  $w$ , we say that there is a solid edge between them; otherwise, we say that they have an empty edge. Let  $\deg(b)$  be the degree of a black vertex  $b$ , namely, the number of solid edges of  $b$ . Given a subset of black vertices  $V \subseteq B$ , we denote by  $b_i(V)$  the black vertex with  $i$ -th largest degree in  $V$ , and by  $\deg_i(V)$  the degree of  $b_i(V)$ . Furthermore, define  $\text{degavg}(V) = \frac{1}{k} \sum_{i=1}^k \deg_i(V)$ . We now state the other two problems studied in this work, which aim to identify  $k$  black vertices whose degrees are the greatest either individually or aggregatively:

**Problem 3 [k-Most Connected Vertex [14] (k-MCV)]** Given parameters  $\epsilon \in (0, 1/4]$ ,  $\delta \in (0, 1/4]$ , and  $k \leq n/2$ , we want to select a  $k$ -sized subset  $V$  of  $B$  such that, with probability at least  $1 - \delta$ , it holds that  $\deg_i(V) \geq (1 - \epsilon) \deg_i(B)$ ,  $\forall i \leq k$ . **Problem 4 [kavg-Most Connected Vertex (kavg-MCV)]** Given the same parameters as in k-MCV, we want to select a  $k$ -sized subset  $V$  of  $B$  such that, with probability at least  $1 - \delta$ , it holds that  $\text{degavg}(V) \geq (1 - \epsilon) \text{degavg}(B)$ . A feasible solution to k-MCV is also feasible for kavg-MCV, but not the vice versa. We will refer to the cost of an algorithm also as its sample complexity, by regarding a probe operation as "sampling" the edge probed. For any deterministic algorithm, the adversary can force the algorithm to always probe  $\Theta(nm)$  edges. Hence, we only consider randomized algorithms. k-MCV can be reduced to k-AS. Given a hidden bipartite graph  $(B, W, E)$ , we can treat every black vertex  $b \in B$  as an "arm" associated with a Bernoulli reward distribution: the reward is 1 with probability  $\deg(b)/m$  (recall  $m = |W|$ ), and 0 with probability  $1 - \deg(b)/m$ . Any algorithm  $A$  for k-AS can be deployed to solve k-MCV as follows. Whenever  $A$  samples from arm  $b$ , we randomly choose a white vertex  $w \in W$ , and probe the edge between  $b$  and  $w$ . A reward of 1 is returned to  $A$  if and only if the edge exists. <sup>2</sup>

k-AS and k-MCV differ, however, in the size of the population that a reward distribution is defined on. For k-AS, the reward of each arm is sampled from a population of an indefinite size, which can even be infinite. Consequently, k-AS nicely models situations such as the crowdsourcing application mentioned earlier. For k-MCV, the reward distribution of each "arm" (i.e., a black vertex  $b$ ) is defined on a population of size  $m = |W|$  (i.e., the edges of  $b$ ). This has three implications. First, k-MCV is a better modeling of applications like top- $k$  semi-join (where an edge exists between  $b \in B$  and  $w \in W$  if and only if  $Q(b, w)$  is true). Second, the problem admits an obvious algorithm with cost  $O(nm)$  (recall  $n = |B|$ ): simply probe all the hidden edges. Third, an algorithm never needs to probe the same edge between  $b$  and  $w$  twice—once probed, whether the edge is solid or empty is perpetually revealed. We refer to the last implication as the history-awareness property. The above discussion on k-AS and k-MCV also applies to kavg-AS and kavg-MCV. For each of above problems, we refer to an algorithm which achieves the precision and failure requirements prescribed by  $\epsilon$  and  $\delta$  as an  $(\epsilon, \delta)$ -approximate algorithm. <sup>1,2</sup>

#### Previous Results

**Problem 1.** Sheng et al. [14] presented an algorithm  $\mathcal{A}_1$  that solves k-AS with expected cost  $O(n^2 k(B) \log n)$ . No lower bound is known on the sample

complexity of  $k$ -AS. The closest work is due to Kalyanakrishnan et al. [11]. They considered the  $E$  XPLORE- $k$  problem, where the goal is to return a set  $V$  of  $k$  arms such that, with probability at least  $1 - \epsilon$ , the mean of each arm in  $V$  is at least  $\epsilon k(B) - \epsilon$ . They showed an algorithm with sample complexity  $O(n^2 \log k)$  in expectation and establish a matching lower bound. Note that  $E$  XPLORE- $k$  ensures an absolute-error guarantee, which is weaker than the individually relative-error guarantee of  $k$ -AS. Therefore, the same  $E$  XPLORE- $k$  lower bound also applies to  $k$ -AS. The readers may be tempted to set  $\epsilon = \epsilon k(B)$  to derive a lower bound of  $O(n^2 (\epsilon k(B))^2 \log k)$  for  $k$ -AS. This, however, is clearly wrong because when  $\epsilon k(B) = o(1)$  (a typical case in practice) this lower bound may be even higher than the upper bound of [14] mentioned earlier. The cause of the error lies in that the hard instance constructed in [11] requires  $\epsilon k(B) = \Omega(1)$ . Problem 2. The  $O(n^2 \epsilon k(B) \log n)$  upper bound of [14] on  $k$ -AS carries over to  $k$ avg-AS (which, as mentioned before, can be solved by any  $k$ -AS algorithm). Zhou et al. [16] considered an OPT MAI problem whose goal is to find a  $k$ -sized subset  $V$  such that  $\epsilon \text{avg}(V) \geq \epsilon \text{avg}(B) - \epsilon$  holds with probability at least  $1 - \epsilon$ . Note, once again, that this is an absolute-error guarantee, as opposed to the relative-error guarantee of  $k$ avg-AS. For OPT MAI, Zhou et al. presented an algorithm with sample complexity  $O(n^2 (1 + \log(1/\epsilon)))$  in expectation. Observe that if  $\epsilon \text{avg}(B)$  is available magically in  $k$  advance, we can immediately apply the OPT MAI algorithm of [16] to settle  $k$ avg-AS by setting  $\log(1/\epsilon) - 1 = \epsilon \text{avg}(B)$ . The expected cost of the algorithm becomes  $O(n^2 (\epsilon \text{avg}(B)))$  (which  $2(1 + k)$  is suboptimal. See the table).

No lower bound is known on the sample complexity of  $k$ avg-AS. For OPT MAI, Zhou et al. [16]), which directly applies to  $k$ avg-AS due to its stronger proved a lower bound of  $O(n^2 (1 + \log(1/\epsilon)) k)$  quality guarantee. Problems 3 and 4. Both problems can be trivially solved with cost  $O(nm)$ . Furthermore, as explained in Section 1.1,  $k$ -MCV and  $k$ avg-MCV can be reduced to  $k$ -AS and  $k$ avg-AS respectively. Indeed, the best existing  $k$ -AS and  $k$ avg-AS algorithms (surveyed in the above) serve as the state of the art for  $k$ -MCV and  $k$ avg-MCV, respectively. Prior to this work, no lower bound results were known for  $k$ -MCV and  $k$ avg-MCV. Note that none of the lower bounds for  $k$ -AS (or  $k$ avg-AS) is applicable to  $k$ -MCV (or  $k$ avg-MCV, resp.), because there is no reduction from the former problem to the latter. 1.3 Our Results We obtain tight upper and lower bounds for all of the problems defined in Section 1.1. Our main results are summarized in Table 1 (all bounds are in expectation). Next, we explain several highlights, and provide an overview into our techniques. 1

The algorithm was designed for  $k$ -MCV, but it can be adapted to  $k$ -AS as well.

3

Table 1: Comparison of our and previous results (all bounds are in expectation) problem sample complexity source

$k$ -AS

$O$

upper bound



proof of the lower bound in [16, Theorem 5.5] .  $k$ -MCV and  $k$ avg-MCV. The stated upper bounds for  $k$ -MCV and  $k$ avg-MCV in Table 1 can be obtained directly from our  $k$ -AS and  $k$ avg-AS algorithms. In contrast, all the lower-bound arguments for  $k$ -AS and  $k$ avg-AS which crucially rely on the samples being i.i.d. break down for the two MCV problems, due to the history-awareness property explained in Section 1.1. For  $k$ -MCV, we remedy the issue by (i) (when  $\deg_k(B)$  is large) a reduction from  $k$ -AS, and (ii) (when  $\deg_k(B)$  is small) a reduction from a sampling lower bound for distinguishing two extremely similar distributions (Theorem 3). Analogous ideas are deployed for  $k$ avg-MCV (Theorem 6). Note that for a small range of  $\deg_k(B)$  (i.e.,  $\Theta(1) \leq \deg_k(B) \leq O(12 \log n)$ ), we do not have the optimal lower bounds yet for  $k$ -MCV and  $k$ avg-MCV. Closing the gap is left as an interesting open problem.

4

Algorithm 1: ME-AS 1 input:  $B, \epsilon, k$  2 for  $\epsilon = 1/2, 1/4, \dots$  do 3  $S = \text{ME}(B, \epsilon, \epsilon, k)$ ; 4  $\{(a_i, \text{US}(a_i)) \mid 1 \leq i \leq k\} = \text{US}(S, \epsilon, (1 - \epsilon/2)\epsilon, k)$ ; 5 if  $\text{US}(a_k) \geq \epsilon$  then 6 return  $\{a_1, \dots, a_k\}$ ; Algorithm 2: Median Elimination (ME) 1 input:  $B, \epsilon, \epsilon, k$  2  $S_1 = B, \epsilon_1 = \epsilon/16, \epsilon_1 = \epsilon/8, \epsilon_1 = \epsilon$ , and  $\epsilon' = 1$ ; 3 while  $|S'| \geq 4k$  do 4 sample every arm  $a \in S'$  for  $Q' = (12/2^{\epsilon'})/(1/\epsilon') \log(6k/\epsilon')$  times; 5 for each arm  $a \in S'$  do  $\bar{a} =$  the average of the  $Q'$  samples from  $a$ ; 6 its empirical value  $\bar{a}$ ; 7  $a_1, \dots, a_{|S'|}$  = the arms sorted in non-increasing order of their empirical values; 8  $S' + 1 = \{a_1, \dots, a_{|S'|/2}\}$ ; 9  $\epsilon' + 1 = 3\epsilon'/4, \epsilon' + 1 = \epsilon'/2, \epsilon' + 1 = (1 - \epsilon')\epsilon'$ , and  $\epsilon' = \epsilon' + 1$ ; 10 return  $S'$ ; Algorithm 3: Uniform Sampling (US) 1 input:  $S, \epsilon, \epsilon_s, k$  2 sample every arm  $a \in S$  for  $Q = (96/2^{\epsilon})(1/\epsilon_s) \log(4|S|/\epsilon)$  times; 3 for each arm  $a \in S$  do 4 its US-empirical value  $\text{US}(a)$  = the average of the  $Q$  samples from  $a$ ; 5  $a_1, \dots, a_{|S|}$  = the arms sorted in non-increasing order of their US-empirical values; 6 return  $\{(a_1, \text{US}(a_1)), \dots, (a_k, \text{US}(a_k))\}$

2

#### Top-k Arm Selection

In this section, we describe a new algorithm for the  $k$ -AS problem. We present the detailed analysis in Appendix B. Our  $k$ -AS algorithm consists of three components: ME-AS, Median Elimination (ME), and Uniform Sampling (US), as shown in Algorithms 1, 2, and 3, respectively. Given parameters  $B, \epsilon, \epsilon, k$  (as in Problem 1), ME-AS takes a “guess”  $\epsilon$  (Line 2) on the value of  $\epsilon_k(B)$ , and then applies ME (Line 3) to prune  $B$  down to a set  $S$  of at most  $4k$  arms. Then, at Line 4, US is invoked to process  $S$ . At Line 5, (as will be clear shortly) the value of  $\text{US}(a_k)$  is what ME-AS thinks should be the value of  $\epsilon_k(B)$ ; thus, the algorithm performs a quality check to see whether  $\text{US}(a_k)$  is larger than but close to  $\epsilon$ . If the check fails, ME-AS halves its guess  $\epsilon$  (Line 2), and repeats the above steps; otherwise, the output of US from Line 4 is returned as the final result. ME runs in rounds. Round  $\epsilon'$  ( $= 1, 2, \dots$ ) is controlled by parameters  $S', \epsilon', \epsilon$ , and  $\epsilon$  (their values for Round 1 are given at Line 1). In general,  $S'$  is the set of arms from which we still want to sample. For each arm  $a \in S'$ , ME takes  $Q'$  (Line 4) samples from  $a$ , and calculates its empirical value  $\bar{a}$  (Lines 5 and 6). ME drops (at Lines 7 and 8) half of the arms in  $S'$  with the smallest empirical values, and then (at Line 9) sets the parameters of

the next round. ME terminates by returning  $S'$  as soon as  $|S'|$  is at most  $4k$  (Lines 3 and 10). US simply takes  $Q$  samples from each arm  $a \in S$  (Line 2), and calculates its US-empirical value  $\hat{\mu}_{US}(a)$  (Lines 3 and 4). Finally, US returns the  $k$  arms in  $S$  with the largest US-empirical values (Lines 5 and 6). Remark. If we ignore Line 3 of Algorithm 1 and simply set  $S = B$ , then ME-AS degenerates into the algorithm in [14].

5

Theorem 1 ME-AS solves the  $k$ -AS problem with expected cost  $O$

$n^{1/2} k \log(B)$

$\log$

$k$

.

We extend the proof in [11] and establish the lower bound for  $k$ -AS as shown in Theorem 2.

Theorem 2 For any  $\epsilon \in (0, 1/4]$  and  $\delta \in (0, 1/48]$ , given any algorithm, there is an instance of the  $k$ -AS problem on which the algorithm must entail  $(n^{1/2} k \log(B) \log k)^{1-\epsilon}$  cost in expectation.

3

### $k$ -MOST CONNECTED VERTEX

This section is devoted to the  $k$ -MCV problem (Problem 3). We will focus on lower bounds because our  $k$ -AS algorithm in the previous section also settles  $k$ -MCV with the cost claimed in Table 1 by applying the reduction described in Section 1.1. We establish matching lower bounds below:

Theorem 3 For any  $\epsilon \in (0, 1/2]$  and  $\delta \in (0, 1/48]$ , the following statements are true about any  $k$ -MCV algorithm:

• when  $\deg_k(B) \leq 12 \log n$ , there is an instance on which the algorithm must probe  $(n^{1/2} \deg_k(B) \log k)^{1-\epsilon}$  edges in expectation.

• when  $\deg_k(B) \geq O(1)$ , there is an instance on which the algorithm must probe  $(nm)^{1-\epsilon}$  edges in expectation. For large  $\deg_k(B)$  in Theorem 3, we utilize an instance for  $k$ -AS to construct a random hidden bipartite graph and feed it to any algorithm solves  $k$ -MCV. By doing this, we reduce  $k$ -AS to  $k$ -MCV and thus, establish our first lower bound. For small  $\deg_k(B)$ , we define the single-vertex problem where the goal is to distinguish two extremely distributions. We prove the lower bound of single-vertex problem and reduce it to  $k$ -MCV. Thus, we establish our second lower bound. The details are presented in Appendix D.

4

### Top- $k$ avg Arm Selection

Our  $k$ avg-AS algorithm QE-AS is similar to ME-AS described in Section 2, except that the parameters are adjusted appropriately, as shown in Algorithm 4, 5, 6 respectively. We present the details in Appendix E.

Theorem 4 QE-AS solves the  $k$ avg-AS problem with expected cost  $O(n^{1/2} \text{avg}_1(B) (1 + \log(1/\delta))^{1-\epsilon} k)$ . We establish the lower bound for  $k$ avg-AS as shown in Theorem 5.

Theorem 5 For any  $\epsilon \in (0, 1/2]$  and  $\delta \in (0, 1/48]$ , given any  $(\epsilon, \delta)$ -approximate algorithm, there is an instance of the  $k$ avg-AS problem on which the algorithm must entail

$\frac{1}{\epsilon}$   
 $\frac{1}{\epsilon} \log(1/\epsilon) \log(1/\epsilon)$

cost in expectation.

We show that the lower bound of  $\text{kavg}$ -AS is the maximum of  $\frac{1}{\epsilon} \log(1/\epsilon)$  and  $k$

$\frac{1}{\epsilon} \log(1/\epsilon) \log(1/\epsilon)$ . Our proof of the first lower bound is based on a novel reduction from 1-AS. We stress that our reduction can be used to simplify the proof of the lower bound in [16, Theorem 5.5].

5  
 $\text{kavg}$ -MOST CONNECTED VERTEX

Our  $\text{kavg}$ -AS algorithm, combined with the reduction described in Section 1.1, already settles  $\text{kavg}$  MCV with the sample complexity given in Table 1. We establish the following lower bound and prove it in Appendix F.

1 1 Theorem 6 For any  $\epsilon \in (0, 1/2]$  and  $k \in \mathbb{N}$ , the following statements are true about any  $\text{kavg}$ -MCV algorithm:

1 when  $\deg_{\text{avg}}(B) \geq \frac{1}{\epsilon} \log(1/\epsilon)$ , there is an instance on which the algorithm must probe

$\frac{1}{\epsilon} \log(1/\epsilon) \log(1/\epsilon)$  edges in expectation. 2 when  $\deg_k(B) \leq \frac{1}{\epsilon} \log(1/\epsilon)$ , there is an instance on which the algorithm must probe  $\Omega(k \log(1/\epsilon))$  edges in expectation.

6

Algorithm 4: QE-AS 1 input:  $B, \epsilon, k$  2 for  $\epsilon = 1/2, 1/4, \dots$  do 3  $S = \text{QE}(B, \epsilon, k)$ ; 4  $\{(a_i, \text{val}_i)\}_{i=1}^k = \text{US}(S, \epsilon, (1/\epsilon) \log(1/\epsilon), k)$ ; 5 if  $\text{val}_i \geq \frac{1}{\epsilon} \log(1/\epsilon)$  then 6 return  $\{a_1, \dots, a_k\}$ ; Algorithm 5: Quartile Elimination (QE) 1 input:  $B, \epsilon, k$  2  $S_1 = B$ ,  $\epsilon_1 = \epsilon/8$ ,  $\epsilon_1 = \epsilon$ , and  $\epsilon_1 = 1$ ; 3 while  $|S_1| \geq 4k$  do

4 sample every arm  $a \in S_1$  for  $Q_1 = (48/\epsilon_1) \log(1/\epsilon_1)$  times;

5  $\epsilon_1 = \epsilon_1/2$ ;

6 for each arm  $a \in S_1$  do 7  $\bar{a} = \text{average of the } Q_1 \text{ samples from } a$ ; 8 its empirical value  $\bar{a}$

9  $a_1, \dots, a_{|S_1|} = \text{arms sorted in non-increasing order of their empirical values}$ ; 10  $S_2 = \{a_1, \dots, a_{|S_1|/4}\}$ ; 11  $\epsilon_2 = \epsilon_1/8$ ,  $\epsilon_2 = \epsilon_1/2$ ,  $\epsilon_2 = (1/\epsilon_2) \log(1/\epsilon_2)$ , and  $\epsilon_2 = \epsilon_2 + 1$ ; 12 return  $S_2$ ; 13 8 9

Algorithm 6: Uniform Sampling (US) 1 input:  $S, \epsilon, \epsilon_s, k$

2 sample every arm  $a \in S$  for  $Q = (120/\epsilon_s) \log(1/\epsilon_s)$  times;

3  $\epsilon_s = \epsilon_s/2$ ;

4 for each arm  $a \in S$  do 5 its US-empirical value  $\bar{a} = \text{average of the } Q \text{ samples from } a$ ; 6  $a_1, \dots, a_{|S|} = \text{arms sorted in non-increasing order of their US-empirical values}$ ; 7 US 8 return  $\{(a_1, \bar{a}_1), \dots, (a_k, \bar{a}_k)\}$ ; 9  $\bar{a} = \text{average of the } Q \text{ samples from } a$

6

1 k

Pk



$i=1$   
 $\{US(a_i)\}$

#### Experiment Evaluation

Due to the space constraint, we show only the experiments that compare ME-AS and AMCV [14] for k-MCV problem. Additional experiments can be found in Appendix G. We use two synthetic data sets and one real world data set to evaluate the algorithms. Each dataset is represented as a bipartite graph with  $n = m = 5000$ . For the synthetic data, the degrees of the black vertices follow a power law distribution. For each black vertex  $b \in B$ , its degree equals  $d$  with probability  $c(d + 1)^{-\alpha}$  where  $\alpha$  is the parameter to be set and  $c$  is the normalizing factor. Furthermore, for each black vertex with degree  $d$ , we connected it to  $d$  randomly selected white vertices. Thus, we build two bipartite graphs by setting the proper parameters in order to control the average degrees of the black vertices to be 50 and 3000 respectively. For the real world data, we crawl 5000 active users from twitter with their corresponding relationships. We construct a bipartite graph  $G = (B, W, E)$  where each of  $B$  and  $W$  represents all the users and  $E$  represents the 2-hop relationships. We say two users  $b \in B$  and  $w \in W$  have a 2-hop relationship if they share at least one common friend. As the theoretical analysis is rather pessimistic due to the extensive usage of the union bound, to make a fair comparison, we adopt the same strategy as in [14], i.e., to divide the sample cost in theory by a heuristic constant  $\gamma$ . We use the same parameter  $\gamma = 2000$  for AMCV as in [14]. For ME-AS, we first take  $\gamma = 107$  for each round of the median elimination step and then we use the previous sample cost dividing 250 as the samples of the uniform sampling step. Notice that it does not conflict the theoretical sample complexity since the median elimination step dominates the sample complexity of the algorithm. We fix the parameters  $\alpha = 0.1$ ,  $k = 20$  and enumerate from 0.01 to 0.1. We then calculate the actual failure probability by counting the successful runs in 100 repeats. Recall that due to the heuristic nature, the algorithm may not achieve the theoretical guarantees prescribed by  $(\gamma, \alpha)$ .

7

Whenever this happens, we label the percentage of actual error  $a$  it achieves according to the failure probability  $\alpha$ . For example 2.9 means the algorithm actually achieves an error  $a = 0.029$  with failure probability  $\alpha$ . The experiment result is shown in Fig 1.

107
5.6
8.5
11.0
106
16.3
19.2 18.9
28.0
26.2 15.5 11.0 12.1 13.9
105 0.01
0.03

0.05  
 0.07  
 0.09  
 108  
 AMCV ME-AS  
 107  
 106  
 11.8  
 105 0.01  
 sample cost  
 AMCV ME-AS sample cost  
 sample cost  
 108  
 107  
 9.3  
 0.03  
 0.05  
 0.07  
 0.09  
 12.1  
 106  
 0.01  
 ? = 3000 (b) Power law with deg  
 16.3  
 5.7  
 105  
 ? = 50 (a) Power law with deg  
 AMCV ME-AS  
 3.7  
 0.03  
 18.0  
 22.7 23.1  
 29.5  
 27.6 12.8 7.4 8.6 9.9 11.2  
 0.05  
 0.07  
 0.09  
 (c) 2-hop

Figure 1: Performance comparison for k-MCV vs. As we can see, ME-AS outperforms AMCV in both sample complexity and the actual error in all data sets. We stress that in the worst case, it seems ME-AS only shows a difference when  $n \approx k$ . However for the most of the real world data, the degrees of the vertices usually follow a power law distribution or a Gaussian distribution. For such cases, our algorithm only needs to take a few samples in each round of the elimination step and drops half of vertices with high confidence. Therefore,

the experimental result shows that the sample cost of ME-AS is much less than AMCV.

7

#### Related Work

Multi-armed bandit problems are classical decision problems with exploration-exploitation tradeoffs, and have been extensively studied for several decades (dating back to 1930s). In this line of research, k-AS and kavg-AS fit into the pure exploration category, which has attracted significant attentions in recent years due to its abundant applications such as online advertisement placement [6], channel allocation for mobile communications [2], crowdsourcing [16], etc. We mention some closely related work below, and refer the interested readers to a recent survey [4]. Even-Dar et al. [8] proposed an optimal algorithm for selecting a single arm which approximates the best arm with an additive error at most (a matching lower bound was established by Mannor et al. [12]). Kalyanakrishnan et al. [10, 11] considered the EXPLORE-k problem which we mentioned in Section 1.2. They provided an algorithm with the sample complexity  $O(n^2 \log k)$ . Similarly, Zhou et al. [16] studied the OPT-MAI problem which, again as mentioned in Section 1.2, is the absolute-error version of kavg-AS. Audibert et al. [2] and Bubeck et al. [4] investigated the fixed budget setting where, given a fixed number of samples, we want to minimize the so-called misidentification probability (informally, the probability that the solution is not optimal). Bubeck et al. [5] also showed the links between the simple regret (the gap between the arm we obtain and the best arm) and the cumulative regret (the gap between the reward we obtained and the expected reward of the best arm). Gabillon et al. [9] provide a unified approach UGapE for EXPLORE-k in both the fixed budget and the fixed confidence settings. They derived the algorithms based on lower and upper confidence bound (LUCB) where the time complexity depends on the gap between  $\mu_k$  (B) and the other arms. Note that each time LUCB samples the two arms that are most difficult to distinguish. Since our problem ensures an individually guarantee, it is unclear whether only sampling the most difficult-to-distinguish arms would be enough. We leave it as an intriguing direction for future work. Chen et al. [6] studied how to select the best arms under various combinatorial constraints. Acknowledgements. Jian Li, Wei Cao, Zhize Li were supported in part by the National Basic Research Program of China grants 2015CB358700, 2011CBA00300, 2011CBA00301, and the National NSFC grants 61202009, 61033001, 61361136003. Yufei Tao was supported in part by projects GRF 4168/13 and GRF 142072/14 from HKRGC.

8

## 2 References

- [1] Y. Amsterdamer, S. B. Davidson, T. Milo, S. Novgorodov, and A. Somech. OASSIS: query driven crowd mining. In SIGMOD, pages 589-600, 2014.
- [2] J.-Y. Audibert, S. Bubeck, et al. Best arm identification in multi-armed bandits. COLT, 2010.
- [3] Z. Bar-Yossef. The complexity of massive data set

computations. PhD thesis, University of California, 2002. [4] S. Bubeck, N. Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and trends in machine learning*, 5(1):1?122, 2012. [5] S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19):1832?1852, 2011. [6] S. Chen, T. Lin, I. King, M. R. Lyu, and W. Chen. Combinatorial pure exploration of multiarmed bandits. In *Advances in Neural Information Processing Systems*, pages 379?387, 2014. [7] D. P. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009. [8] E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *The Journal of Machine Learning Research*, 7:1079?1105, 2006. [9] V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in Neural Information Processing Systems*, pages 3212?3220, 2012. [10] S. Kalyanakrishnan and P. Stone. Efficient selection of multiple bandit arms: Theory and practice. In *ICML*, pages 511?518, 2010. [11] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. PAC subset selection in stochastic multiarmed bandits. In *ICML*, pages 655?662, 2012. [12] S. Mannor and J. N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *The Journal of Machine Learning Research*, 5:623?648, 2004. [13] A. G. Parameswaran, S. Boyd, H. Garcia-Molina, A. Gupta, N. Polyzotis, and J. Widom. Optimal crowd-powered rating and filtering algorithms. *PVLDB*, 7(9):685?696, 2014. [14] C. Sheng, Y. Tao, and J. Li. Exact and approximate algorithms for the most connected vertex problem. *TODS*, 37(2):12, 2012. [15] J. Wang, E. Lo, and M. L. Yiu. Identifying the most connected vertices in hidden bipartite graphs using group testing. *TKDE*, 25(10):2245?2256, 2013. [16] Y. Zhou, X. Chen, and J. Li. Optimal PAC multiple arm identification with applications to crowdsourcing. In *ICML*, pages 217?225, 2014. [17] M. Zhu, D. Papadias, J. Zhang, and D. L. Lee. Top-k spatial joins. *TKDE*, 17(4):567?579, 2005.