Hear the Whole Story: Towards the Diversity of Opinion in Crowdsourcing Markets

Ting Wu Lei Chen Pan Hui Chen Jason Zhang Weikai Li
Hong Kong University of Science and Technology, Hong Kong, China
{twuad,leichen,panhui,czhangad,weikaili}@cse.ust.hk

ABSTRACT

The recent surge in popularity of crowdsourcing has brought with it a new opportunity for engaging human intelligence in the process of data analysis. Crowdsourcing provides a fundamental mechanism for enabling online workers to participate in tasks that are either too difficult to be solved solely by a computer or too expensive to employ experts to perform. In the field of social science, four elements are required to form a wise crowd - Diversity of Opinion, Independence, Decentralization and Aggregation. However, while the other three elements are already studied and implemented in current crowdsourcing platforms, the 'Diversity of Opinion' has not been functionally enabled. In this paper, we address the algorithmic optimizations towards the *diversity of opinion* of crowdsourcing marketplaces.

From a computational perspective, in order to build a wise crowd, we need to quantitatively modeling the diversity, and take it into consideration for constructing the crowd. In a crowdsourcing marketplace, we usually encounter two basic paradigms for worker selection: building a crowd to wait for tasks to come and selecting workers for a given task. Therefore, we propose our Similarity-driven Model (S-Model) and Task-driven Model (T-Model) for both of the paradigms. Under both of the models, we propose efficient and effective algorithms to enlist a budgeted number of workers, which have the optimal diversity. We have verified our solutions with extensive experiments on both synthetic datasets and real data sets.

1. INTRODUCTION

Recently, with the emergence of crowdsourcing platforms, such as Amazon Mechanical Turk [3] and CrowdFlower [4], more and more applications are utilizing human intelligence in processing various tasks that are either too difficult to be solved only by computers alone or too expensive to employ experts to perform. For example, data gathering can be done implicitly, through crowdsourced sensing and on-line behaviour collection, or explicitly, by sending targeted information requests to the crowd. Given another example from an analytical perspective, human input can be used to address computationally difficult tasks such as entity resolution [34], schema matching [35] and the like.

Proceedings of the VLDB Endowment, Vol. 8, No. 5

Though humankind is intelligent, meanwhile, they are also erroneous and greedy, which makes the quality of crowdsourcing results quite questionable. Therefore, it is important to select the "right" workers to build a wise crowd to guarantee the quality. Then one crucial question to address is "What are the elements of a wise crowd?". Fortunately, this question has been thoroughly studied in the field of social science and many detailed answers have been given. One of the most recognized answers, from [31] with over 5,000 citations, points out that four elements are essential to form a wise crowd, which are:

1. Diversity of Opinion - Each person should have private information even if it's just an eccentric interpretation of the known facts.
2. Independence - People's opinions aren't determined by the opinions of those around them.
3. Decentralization - People are able to specialize and draw on local knowledge.
4. Aggregation - Some mechanism exists for turning private judgements into a collective decision.

Therefore, in order to construct a wise crowd, we need to make sure that the constructed crowd satisfies the above four elements. From the perspective of crowdsourcing systems, *independence* and *decentralization* are easy to achieve, by providing a free and independent channel for each individual worker, that is, a means to enable each worker to answer questions based on personal specialism without being aware of other workers. Existing crowdsourcing platforms, such as AMT and CrowdFlower, work precisely in this way. Concerning *aggregation*, various mechanisms have been proposed already, such as majority voting [10], to achieve a target overall reliability. However, to the best of our knowledge, how to ensure the *diversity of opinion* in constructing a wise crowd has not been studied from algorithmic perspectives before. Thus, in this paper, we address the algorithmic optimizations towards the *diversity of opinion* for crowd construction.

1.1 When Diversity Trumps Ability

The effect of diversity differs depending on the corresponding crowdsourced tasks, as pointed out in [23]. In particular, for *problem-solving tasks*, diversity is the essential factor affecting the performance of a crowd, and it is even much more important than the average ability of individuals. This phenomenon was discovered and verified in [24], and referred to the 'Diversity Trumps Ability Theorem', which makes the observation that diverse groups of problem solvers - groups of people with diverse tools consistently outperformed groups of the best and the brightest. People with high abilities are often trained in the same institutions, tend to possess similar perspectives and apply similar problem-solving techniques, or heuristics. Many problems do not succumb to a single heuristic, or even a set of similar ones. This is why a diverse crowd functions better than a few experts. Intuitively, if two groups are formed, one random (and therefore diverse) and one consisting of the best individual performers, the first group almost always did better.

This theorem ends up indirectly providing convincing arguments as to why - under certain conditions - citizens may outperform elected officials and experts [23].

1.2 Two Basic Models for Diversity of Opinion

From a computational perspective, in order to build a wise crowd, we are interested in quantitatively modeling the diversity, and take it into consideration for constructing a crowd. In a crowdsourcing marketplace, we usually encounter two basic paradigms for worker selection: building a crowd that will wait for tasks to come or selecting workers for a given task. We propose models for both of the paradigms.

1.2.1 Similarity-driven Model (S-Model)

When there is no explicit query, we resort to the pairwise similarity of workers to model the diversity of opinion. In particular, we model the similarity of a pair of workers as a similarity score value (high value indicates high similarity), and use the negative value of average pairwise similarity to quantify the overall diversity. Intuitively, the lower the average similarity, the higher the diversity.

S-Model can be applied to crowdsourcing scenarios which do not have explicit queries when constructing a crowd and require quick responses when a query arrives. For example, diners may comment on a restaurant through Foursquare [1], whereas iPhone users may post ratings of the applications that they have downloaded from the Apple Store. Such data is highly valuable for product creators (usually a company) : as ratings and reviews have a significant impact on sales; and companies can analyze ratings and review trends to adjust overall marketing strategies, improve customer service, and fine-tune merchandising and so on. However, in current web-based commenting systems, product creators must passively wait for reviewers to visit the commenting systems to provide their comments and ratings. Hence, product creators may have to wait a long time to receive a satisfactory number of reviews. These drawbacks with existing commenting systems motivate the quest for effective methods to actively invite a group of reviewers prior to the arrival of the query.

1.2.2 Task-driven Model (T-Model)

Another common scenario is that a requester has a specific query, and enlists workers to join the crowd to answer it. In such a paradigm, we are able to analyze the diversity of workers according to the content of the query. Regarding the given query, we model the opinion of each worker as a probability ranging from 0 to 1, which indicates opinions from negative to positive, respectively. To guarantee the desirable diversity of opinion, we allow a user to set up the demand on the number of workers with positive (negative) opinions. Therefore, the optimization issue is to maximize the probability that the user's demand is satisfied.

T-model captures essence of diversity for a wide class of crowdsourcing scenarios. A typical example application, which is initiated and currently operated by the US government [2], is an *online petitioning system* enabling participants to propose, discuss and sign political petitions. To determine whether a petition is significant enough to get a response from the White House, the current mechanism is simply a threshold of the number of signatures (currently 100,000), indicating the number of people who support the petition. However, to analyze a particular petition fairly, it would be more constructive if opinions from both the proposition and the opposition are taken into consideration. So guided by the T-model, the government may actively collect online comments on both sides of the petition, which is more constructive for further governmental processing.

Table 1: MEANINGS OF SYMBOLS USED

Notation	Description
$w(w_i)$	a crowdsourcing worker
$Sim(w_i, w_j)$	the pairwise similarity between w_i and w_j
$Div(C)$	the diversity of a crowd C of workers
$\theta_1(\theta_0)$	the number of positive (negative) workers to be enlisted with positive (negative)
t_i	the opinion of worker w_i
$Pr(t = 1 \; or \; 0)$	the probability of t satisfying or dissatisfying P
N	the set of candidate workers to be selected
k	the number of workers to be enlisted
S	the set of workers to be selected, $\|S\| = k$
θ_2	$\theta_2 = k - \theta_0$
$\tau(S)$	the probability of at least θ_1 (θ_0) workers existing in S
T_0	$T_0 = \sum_{t \in S} t$, following Poisson Binomial distribution

1.3 Challenges and Contributions

As *diversity* is a loosely defined concept, the first main challenge is quantitatively measuring the diversity among candidate workers. Another main challenge to be addressed is to design effective and efficient algorithms for worker selection with the consideration of the diversity of opinions. To address these two challenges, we propose effective measures to estimate the diversity of the crowd under two common scenarios, S-Model and T-Model, respectively, and propose effective approximation algorithms for crowd selection. To summarize, this paper has made the following contributions

1. In Section 2, we study the crowd selection problem under S-model, and propose an efficient $(1 + \epsilon)$ approximation algorithm for finding a crowd with the highest diversity.

2. In Section 3, we study the crowd selection problem under the T-model, prove its NP-hardness, and provide a solution based on distribution approximations.

3. In Sections 4 and 5, we discuss related works and conclude the paper.

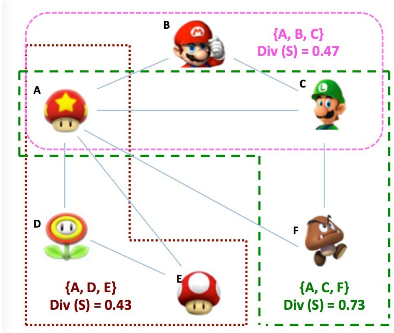
2. SIMILARITY-DRIVEN MODEL

In this section, we formally introduce the model, and propose efficient algorithms to enlist workers.

2.1 Model and Definitions

We first need to design a computational model to depict the crowd diversity for the worker selection problem. Under the similarity-driven model, each pair of workers is associated with a value which describes their pairwise similarity. We aim to select k workers out of n candidates, such that the average pairwise distance is maximized (i.e. the average similarity is minimized).

We formally present the model with the following definitions.



Distances of workers

	A	B	C	D	E	F
A		0.5	0.6	0.7	0.2	0.9
B	0.5		0.3	0.9	0.7	0.8
C	0.6	0.3		0.6	0.8	0.7
D	0.7	0.9	0.6		0.4	0.9
E	0.2	0.7	0.8	0.4		0.6
F	0.9	0.8	0.7	0.9	0.6	

Figure 1: Find 3 workers with highest diversity

DEFINITION 2.1 (PAIRWISE SIMILARITY). *For a given set of potential crowdsourcing workers W, the diversity of any two workers is computed by a pairwise similarity function $Sim(w_i, w_j)$ where $w_i, w_j \in W$.*

DEFINITION 2.2 (CROWD DIVERSITY). *Given a crowd of workers $C = \{w_1, w_2, ..., w_{|C|}\}$, a pairwise similarity function $Sim(.)$, the diversity of the crowd is defined as the negative value averaged pairwise similarity, that is,*

$$Div(C) = -\frac{\sum_{w_i,w_j \in C \wedge i \neq j} Sim(w_i, w_j)}{|C|}$$

Remark: For the sake of generality, we consider $Sim(.)$ here as an abstract function, which measures the *similarity* between two workers. In the appendix, we list a number of popular methods to quantify $Sim(.)$. Aside from these measurements, we can also plug in any reasonable diversity measurements. In our model, users may also design appropriate similarity functions depending on the data structure and application requirements.

Essentially, we are interested in finding a subset of candidate workers with the maximal diversity, using the cardinality constraint. We formally define this optimization problem as follows.

PROBLEM STATEMENT 1 (DIVERSITY MAXIMIZATION). *For a given set of potential crowdsourcing workers W, each worker $w_i \in W$, an integer k, we aim to find a subset $C \subseteq W$ such that $|C| = k$ and $Div(C)$ is maximized, that is,*

$$\arg\max_{C \subseteq W, |C|=k} Div(C)$$

Running Example: Figure 1 illustrates an example with 6 workers and their pairwise similarity values. We aim to select three of them, to maximize the crowd diversity. All the possible selections are enumerated as follows and the associated crowd diversity.

Crowd	$Div(S)$	Crowd	$Div(S)$	Crowd	$Div(S)$
A, B, C	-0.467	A, B, D	-0.7	A, B, E	-0.467
A, B, F	-0.733	A, C, D	-0.633	A, C, E	-0.533
A, C, F	-0.733	**A, D, E**	**-0.433**	A, D, F	-0.833
A, E, F	-0.567	B, C, D	-0.6	B, C, E	-0.6
B, C, F	-0.6	B, D, E	-0.667	B, D, F	-0.867
B, E, F	-0.7	C, D, E	-0.6	C, D, F	-0.73
C, E, F	-0.7	D, E, F	-0.633		

Clearly, the optimal selection is $< A, D, E >$, with the highest diversity -0.433.

2.2 NP-Hardness

Unfortunately, the diversity maximization problem under S-Model is NP-hard, as stated in the following theorem.

THEOREM 2.1. *The diversity maximization problem is NP-hard.*

PROOF. First, we reduce the diversity maximization problem to a subset version: relaxing the constant from $|S| = k$ to be $|S| \leq k$. The reduction is correct because, if a polynomial algorithm A solves the crowd selection problem, then we can solve this by calling A k times, setting $|S| = 1, 2, ..., k$.

Next, we construct a special case of the diversity maximization problem, namely the crowd selection problem. We reach the NP-hardness of crowd selection problem by proving the crowd selection problem is NP-hard. With a trivial reduction, the crowd selection problem becomes an nth-order Knapsack Problem according to Formula 6. Following the proof by H. Kellerer, et al in [19], we prove the hardness of nOKP.

An nth-order Knapsack Problem(nOKP) is a Knapsack problem whose objective function has the form as follows:

$$optimize \sum_{i_1 \in n} \sum_{i_2 \in n} \cdots \sum_{i_n \in n} V[i_1, i_2, \cdots, i_n] \cdot x_1 x_2 \cdots x_n$$

where $V[i_1, i_2, \cdots, i_n]$ is an n-dimensional vector indicating the profit achieved if objects $[i_1, i_2, \cdots, i_n]$ are concurrently selected. Given an instance of a traditional KP, we can construct an nOKP instance by defining the profit n-dimensional vector as $V[i, i, ...i] = p_i$ and $V[otherwise] = 0$ for all i, where p_i is the profit in a traditional KP. The weight vector and objective value remain the same. □

2.3 Approximation Algorithm

In the previous section, we show that the diversity maximization problem is NP-hard. Therefore, we are interested in developing fast approximation algorithms.

Now we revisit the optimization function defined in Definition 2.2: $Div(C) = -\frac{\sum_{w_i,w_j \in C \wedge i \neq j} Sim(w_i, w_j)}{|C|}$, in which $|C|$ is a fixed value, indicating the number of workers to be selected. Hence, the goal is actually to maximize $-\sum_{w_i,w_j \in C \wedge i \neq j} Sim(w_i, w_j)$, which we use $Sum(C)$ to denote. As a result, we have

$$Sum(C) = -\sum_{w_i,w_j \in C \wedge i \neq j} Sim(w_i, w_j)$$

Then, the optimization is equivalently transformed as

$$\arg\max_{C \subseteq W, |C|=k} Sum(C)$$

Furthermore, we discover that the optimization function $Sum(.)$ is a submodular function of the set of candidate workers W.

A function f is submodular if

$$f(A \cup \{a_1\}) + f(A \cup \{a_2\}) \geq f(A \cup \{a_1, a_2\}) + f(A)$$

for any A and $a_1, a_2 \notin A$. Submodularity implies the property of diminishing marginal returns. Intuitively, in our problem, this says that adding a new worker would lead to an enhanced improvement if there were less workers already in the crowd. The problem of selecting a k-element subset maximizing a sub-modular function can be approximated with a performance guarantee of $(1 - 1/e)$, by iteratively selecting the best element given the ones selected so far.

With theorem 2.2, we indicate that function $Sum(.)$ is submodular.

Input: $C \leftarrow \emptyset$
Output: Find C s.t. $|C| = k$ and $Div(C)$ is maximized.
$C \leftarrow \{w_0, w_1\}$
while $|C| \leq k$ **do**

$$x = \arg \max_{w_x \in W} Div(C \cup \{w_x\})$$

$C \leftarrow C \cup \{w_x\}$
end
return C

Algorithm 1: Diversity Maximization

THEOREM 2.2. *For an arbitrary instance of the diversity maximization problem, the resulting optimization function $Sum(.)$ is submodular.*

PROOF. In order to establish this result, we need to prove that $\forall C, w_0, w_1$, we have

$$Sum(C \cup \{w_0\}) + Sum(C \cup \{w_1\}) \geq Sum(C \cup \{w_0, w_1\}) + Sum(C)$$

where $C \subseteq W$, $w_0, w_1 \in W - C$, By definition 2.2, we express the left-hand-side and right-hand-side as follows

$$\begin{aligned} LHS = &- \sum_{w_i, w_j \in C \wedge i \neq j} Sim(w_i, w_j) - \sum_{w \in C} Sim(w, w_0) \\ &- \sum_{w_i, w_j \in C \wedge i \neq j} Sim(w_i, w_j) - \sum_{w \in C} Sim(w, w_1) \end{aligned} \quad (1)$$

$$\begin{aligned} RHS = &- \sum_{w_i, w_j \in C \wedge i \neq j} Sim(w_i, w_j) - \sum_{w \in C} Sim(w, w_0) \\ &- \sum_{w_i, w_j \in C \wedge i \neq j} Sim(w_i, w_j) - \sum_{w \in C} Sim(w, w_1) - Sim(w_0, w_1) \end{aligned} \quad (2)$$

Therefore, we have

$$LHS - RHS = Sim(w_0, w_1) \geq 0$$

which competes the proof. □

Facilitated by Theorem 2.2, our first main result is that the optimal solution for *diversity maximization* can be efficiently approximated within a factor of $(1 - 1/e - \epsilon)$ [7]. Here e is the base of the natural logarithm and ϵ is any arbitrary small positive real number. Thus, this is a performance guarantee slightly better than $(1 - 1/e) = 63\%$.

Algorithm 1 lists the detailed steps of this approximation algorithm. This algorithm, which achieves the performance guarantee, is a natural greedy hill-climbing strategy related to the approach considered in [7]. Thus the main content of this result is the analysis framework needed for obtaining a provable performance guarantee, and the fairly surprising fact that hill-climbing is always within a factor of at least 63% of the optimal for this problem.

3. TASK-DRIVEN MODEL

Under the task-driven model, each worker is associated with a probability, describing his/her opinion about the given task. We aim to select k workers out of n candidates, such that the numbers of positive and negative workers satisfy a user's demand.

We formally define the optimization problem and related important notations in this section.

DEFINITION 3.1 (WORKER OPINION). *A crowdsourcing worker w_i is associated with an opinion t_i about the given task, which is a Bernoulli random variable. We denote the probability $Pr(t_i = 1) = 1 - Pr(t_i = 0)$, where $Pr(t_i = 1)$ ($Pr(t_i = 0)$) is the probability of w_i having a positive (negative) opinion about the task. We assume that the opinions of all the workers are independent.*

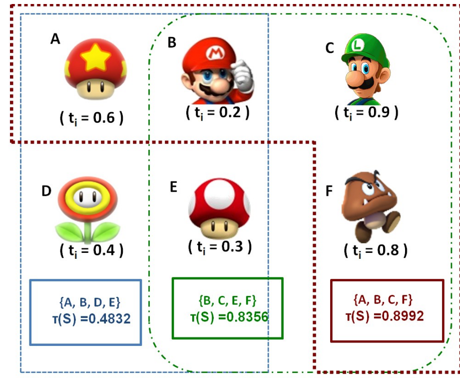


Figure 2: Find 4 workers including 1 supporter and 1 objector

There are two possible ways to obtain the probabilities for the workers. Firstly, when a crowdsourcing platform is implemented on a public online community (e.g. social networks, online forums), we can analyze the historical data and profile information of a given user. Any of the current techniques can be used as a plug-in for our system to detect relevance of a worker to a subject of interest. Secondly, before selecting a worker to participate in a crowd, we may simply ask individual workers for their opinions towards the given subject. On common crowdsourcing platforms, such questions can be designed as so-called *Qualification Tests*, which are prerequisites for workers to answer any questions thereafter.

3.1 Crowd Selection with T-Model

Now we illustrate how to optimize the process of worker selection under T-model. Before providing the formal definition, we introduce the rationale of the optimization. Since each worker's opinion is probabilistic, the total number of workers with positive (negative) opinions is also a probabilistic distribution. We assume that we have the user's demand of the number of workers with positive (negative) opinions, and the optimization is to select the best subset of workers such that the user's demand is satisfied.

As follows, we define the optimization problem under T-model.

DEFINITION 3.2 (K-BEST WORKERS SELECTION). *Given a set of $|N|$ workers $w_1, w_2, ..., w_{|N|}$ with opinions $N = \{t_1, t_2, ..., t_{|N|}\}$. Let θ_1 and θ_0 be the user's demand on the numbers of workers being supportive or opposing with respect to the given task, respectively. We aim to select k workers, so that the probability of the user's demand being fulfilled is maximized. To ensure this probability is positive for any $k \geq 1$, we assume $\theta_0 + \theta_1 <= k$. Formally, let S be the subset of N, and let τ be the probability that at least θ_1 (θ_0) workers existing in S supporting (opposing) the given task,*

$$\tau(S) = Pr\left\{\sum_{t \in S} t \geq \theta_1 \wedge \sum_{t \in S} (1 - t) \geq \theta_0\right\} \quad (3)$$

we have the optimization problem as follows:

$$S := \arg \max_{|S| = k} \tau(S) \quad (4)$$

By taking a closer look at Formula 3, we have $\sum_{t \in S} t + \sum_{t \in S}(1-t) = k$. For the sake of presentation, we denote $T = \sum_{t \in S} t$, $\theta_2 = k - \theta_0$. Then, Formula 3 can be rewritten as

$$\begin{aligned} \tau(S) =& Pr(\theta_1 \leq T \leq \theta_2) \\ =& \sum_{i=\theta_1}^{\theta_2} Pr(T = i) \end{aligned} \tag{5}$$

Since each worker can be treated as a random variable following Bernoulli distributions, T follows a standard Poison Binomial distribution (PBD). Therefore, by adopting the probability mass function (pmf) of PBD, we have

$$\tau(S) = \sum_{i=\theta_1}^{\theta_2} \sum_{A \in F_t} \prod_{t_\alpha \in A} Pr(t_\alpha = 1) \prod_{t_\beta \in A^c} Pr(t_\beta = 0) \tag{6}$$

where F_t is the set of all the subsets of S.

Running Example: A concrete example to illustrate the optimization problem as illustrated in Figure 2. Assume we have a set of candidate workers, with worker opinions $0.2, 0.3, 0.4, 0.6, 0.8$ and 0.9, respectively. We further assume that a user wants to select 4 of them, and one of them has a positive opinion and one of them has a negative opinion. Hence, we have $\theta_1 = 1, \theta_0 = 1, k = 4$, then $\theta_2 = 4 - 1 = 3$. There are totally C_6^4 possible combinations, each of which indicates a PBD. We present all the possible size-4 combinations, and compute $\tau(S)$ for each of them. Figure 3 illustrates the PBD of the number of workers with positive opinions, and indicates the range of probabilities we aim to maximize.

Crowd	$\tau(S)$	Crowd	$\tau(S)$
A, B, C, D	0.7616	A, B, C, E	0.7272
A, B, C, F	0.8992	A, B, D, E	0.4832
A, B, D, F	0.7152	A, B, E, F	0.6784
A, C, D, E	0.7884	**A, C, D, F**	**0.9224**
A, C, E, F	0.9108	A, D, E, F	0.7448
B, C, D, E	0.6188	B, C, D, F	0.8568
B, C, E, F	0.8356	B, D, E, F	0.5736
C, D, E, F	0.8732		

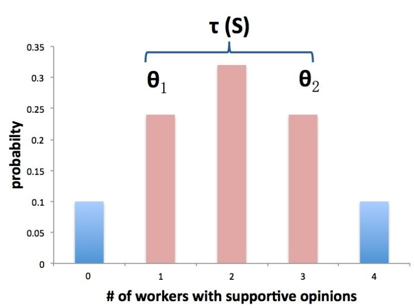


Figure 3: The Poisson-Binomial Distribution

One can see that $< A, C, D, F >$ is the optimal choice, since it maximizes the probability that the user's demand is satisfied.

3.2 Method with Poisson Approximation

To select the exact optimal combination of k workers, we have to enumerate all $O(n^k)$ PBDs, and output the one with the highest $\tau(S)$. However, this naive method leads to very high computational cost. In this subsection, we consider each PBD as a Poisson distribution, and conduct the selection among the approximated Poisson distributions. By aborting the bounded imprecision introduced by the approximation, we significantly improve the efficiency.

A Poisson binomial distribution can be well approximated by a Poisson distribution. Then, we consider T approximately following a Poisson distribution, with parameter $\lambda = \sum_{t \in S} Pr(t = 1)$.

Then, we have

$$Pr(\theta_1 \leq T \leq \theta_2) \approx F_P(\theta_2, \lambda) - F_P(\theta_1, \lambda)$$

where F_P is the cumulative mass function (CMF) of the Poisson distribution. As a result, we find S' to maximize

$$G_P(\lambda) := F_P(\theta_2, \lambda) - F_P(\theta_1, \lambda)$$

and return S' as the approximate answer. In the reminder of this subsection, we first analyze the monotonicity of $G_P(\lambda)$, and then provide two algorithmic solutions.

3.2.1 Monotonicity Analysis

In the following, we first analyze the monotonicity of $G_P(\lambda)$. We discover that $G_P(\lambda)$ has a nice monotonic property, which is algorithmically useful. This discovery is concluded with the following theorem.

THEOREM 3.1. *Considering λ as a continues independent variable with range $(0, k)$, $G_P(\lambda)$ monotonously increases and decreases on $[0, (\frac{\theta_2!}{\theta_1!})^{\frac{1}{\theta_2 - \theta_1}}]$ and $[(\frac{\theta_2!}{\theta_1!})^{\frac{1}{\theta_2 - \theta_1}}, k]$, respectively.*

PROOF. First, we expand F_P, the CMF of Poisson distribution, and rewrite $G_P(\lambda)$ as

$$G_P(\lambda) = e^{-\lambda} \sum_{i=0}^{\theta_2} \frac{\lambda^i}{i!} - e^{-\lambda} \sum_{j=0}^{\theta_1} \frac{\lambda^j}{j!}$$

$$= \sum_{i=\theta_1+1}^{\theta_2} \frac{e^{-\lambda}\lambda^i}{i!}$$

Then, we take the partial derivative of $G_P(\lambda)$ w.r.t λ:

$$\begin{aligned} \frac{\partial G_P(\lambda)}{\partial \lambda} &= \sum_{i=\theta_1+1}^{\theta_2} \frac{\partial(\frac{e^{-\lambda}\lambda^i}{i!})}{\partial \lambda} = \sum_{i=\theta_1+1}^{\theta_2} \frac{e^{-\lambda}(i\lambda^{i-1} - \lambda^i)}{i!} \\ &= e^{-\lambda} \sum_{i=\theta_1+1}^{\theta_2} \frac{(i\lambda^{i-1} - \lambda^i)}{i!} = e^{-\lambda} \sum_{i=\theta_1+1}^{\theta_2} \{\frac{\lambda^{i-1}}{(i-1)!} - \frac{\lambda^i}{i!}\} \\ &= e^{-\lambda}\{\sum_{i=\theta_1+1}^{\theta_2} \frac{\lambda^{i-1}}{(i-1)!} - \sum_{i=\theta_1+1}^{\theta_2} \frac{\lambda^i}{i!}\} = e^{-\lambda}\{\frac{\lambda^{\theta_1}}{\theta_1!} - \frac{\lambda^{\theta_2}}{\theta_2!}\} \\ &= e^{-\lambda}\lambda^{\theta_1}\{\frac{1}{\theta_1!} - \frac{\lambda^{\theta_2 - \theta_1}}{\theta_2!}\} \end{aligned} \tag{7}$$

To analyze the monotonicity of $G_P(\lambda)$, we solve λ for inequation $\frac{\partial G_P(\lambda)}{\partial \lambda} > 0$. Note that, in Eq 7, we have $e^{-\lambda}\lambda^{\theta_1} > 0$, and $\theta_2 > \theta_1$, so

$$\begin{aligned} \frac{\partial G_P(\lambda)}{\partial \lambda} &= e^{-\lambda}\lambda^{\theta_1}\{\frac{1}{\theta_1!} - \frac{\lambda^{\theta_2 - \theta_1}}{\theta_2!}\} > 0 \\ &\Leftrightarrow \lambda^{\theta_2 - \theta_1} < \frac{\theta_2!}{\theta_1!} \Leftrightarrow \lambda < (\frac{\theta_2!}{\theta_1!})^{\frac{1}{\theta_2 - \theta_1}} \end{aligned} \tag{8}$$

Similarly, we have $\frac{\partial G_P(\lambda)}{\partial \lambda} < 0 \Leftrightarrow \lambda > (\frac{\theta_2!}{\theta_1!})^{\frac{1}{\theta_2 - \theta_1}}$, which completes the proof. □

3.2.2 *Transformation to Exact k-item Knapsack Problem (E-kKP)*

Based on the discovered monotonicity property, we show that maximizing $G(\lambda)$ is equivalent to the classical "Exact k-object Knapsack (E-kKP)" problem as shown by the following Theorem.

THEOREM 3.2. *By considering each PBD approximately as a Poisson distribution, the k-best workers selection problem can be solved by any algorithm for the Exact k-item Knapsack Problem (E-kKP).*

PROOF. *Facilitated with theorem 3.1, our optimization is revised to select S such that $\lambda = \sum_{t \in S} Pr(t = 1)$ approaches $(\frac{\theta_2!}{\theta_1!})^{\frac{1}{\theta_2 - \theta_1}}$, which is a constant number. Furthermore, we have $\lambda = \sum_{t \in S} Pr(t = 1)$, then by defining*

$$\Omega_P := (\frac{\theta_2!}{\theta_1!})^{\frac{1}{\theta_2 - \theta_1}}$$

our optimization is further revised as selecting S such that $\sum_{t \in S} Pr(t = 1)$ approaches Ω_P. Despite having the nice property of monotonicity, $G_P(\lambda)$ may not be symmetric, and $\lambda = \sum_{t \in S} Pr(t = 1)$ is a discrete variable. This indicates, we need to find λ_l and λ_r, which achieve maximums of G_P on $[0, (\frac{\theta_2!}{\theta_1!})^{\frac{1}{\theta_2 - \theta_1}}]$ and $[(\frac{\theta_2!}{\theta_1!})^{\frac{1}{\theta_2 - \theta_1}}, k]$, respectively. Then we choose between them by comparing $G_P(\lambda_l)$ and $G_P(\lambda_r)$. Consequently, we aim to find two size-k subsets S_l and S_r of the given N, such that $\sum_{t \in S_l} Pr(t = 1)$ is largest to but no larger than Ω_P, and $\sum_{t \in S_r} Pr(t = 1)$ is smallest to but smaller than Ω_P. Actually, algorithmically speaking, finding S_l is the same as finding S_r. This is because finding S_r is equivalent to finding $N - S_r$, which is $|N| - k$ sized, such that $\sum_{t \in N - S_r} Pr(t = 1)$ is the largest but no larger than $\sum_{t \in N} Pr(t = 1) - \Omega_P$. Therefore, the remaining optimization problem is: finding S_l, which is a size-k subset of N, and we want to maximize the sum of values in S_l without exceeding Ω_P. This is a typical E-kKP problem. □

It is known that E-kKP can be solved by
(1) a backtracking approach with $O(|N|^k/k!)$ time;
(2) dynamic programming with $O(\gamma|N|)$;
(3) 1/2-approximation algorithm by linear programming with $O(|N|)$.

These three algorithms are proposed in [11]. For showing how to adopt these algorithms, we only demonstrate (1), that is, the backtracking algorithm with Algorithm 2. The other two algorithms are analogous.

With Algorithm 2, we find S_l and S_r by $Bt(k, \Omega_P, N)$ and $N - Bt(|N| - k, \sum_{t \in N} Pr(t = 1) - \Omega_P, N)$, receptively. Note $\lambda_l = \sum_{S_l} Pr(t = 1)$ and $\lambda_r = \sum_{S_r} Pr(t = 1)$, we set the output $S' = S_l$ as the final result if $G(\lambda_l) > G(\lambda_r)$; otherwise $S' = S_r$ is returned.

3.3 Method with Binomial Approximation

It is known that Binomial approximation is also an effective method to deal with the high complexity of the Poisson Binomial distribution. Similar to the Poisson approximation, we have

$$Pr(\theta_1 \le T \le \theta_2) \approx F_B(\theta_2; n, p) - F_B(\theta_1; n, p)$$

where F_B is the CMF of Binomial Distribution with parameter $n = k$ and $p = \frac{\sum_{t \in S} Pr(t = 1)}{k}$. Then, the optimization is to maximize:

$$G_B(p) := F_B(\theta_2; n, p) - F_B(\theta_1; n, p)$$

Input: $k, \Omega, N = \{t_0, t_1, ..., t_{|N|}\}$
Output: A size-k subset of N
Fuction $Bt(k, \Omega, N)$
if $|N| = k$ **then**
  return N;
end
else if $\sum_{i=0}^{k-1} Pr(t_i = 1) > \Omega$ **then**
  return null;
end
else if $Bt(k, \Omega, N - t_{|N|}) > Bt(k - 1, \Omega - Pr(t_{|N|} = 1), N - t_{|N|}) + Pr(t_{|N|} = 1)$ **then**
  return $Bt(k, \Omega, N - t_{|N|})$;
end
else
  return $Bt(k - 1, \Omega - Pr(t_{|N|} = 1), N - t_{|N|}) \cup t_{|N|}$
end

Algorithm 2: Backtracking Algorithm (Bt)

Please note n is a fixed parameter since k is a constant in K-best workers selection problem. Therefore, what we can do is to simply adjust p with different selections of S. Analogous to the Poisson Approximation in Section 3.2, we first analyze the monotonicity, and then discuss the algorithm.

Monotonicity Analysis:

With theorem 3.3, we show that $G_B(p)$ also has a useful monotonic feature, which is similar to the Poisson approximation.

THEOREM 3.3. *Considering p as a continues independent variable with range $(0, n)$, $G_B(p)$ monotonously increases and decreases on $[0, \frac{1}{1 + (\frac{(n-\theta_2)C_n^{\theta_2}}{(n-\theta_1)C_n^{\theta_1}})^{\frac{1}{\theta_2 - \theta_1}}}]$ and $[\frac{1}{1 + (\frac{(n-\theta_2)C_n^{\theta_2}}{(n-\theta_1)C_n^{\theta_1}})^{\frac{1}{\theta_2 - \theta_1}}}, n]$, respectively*

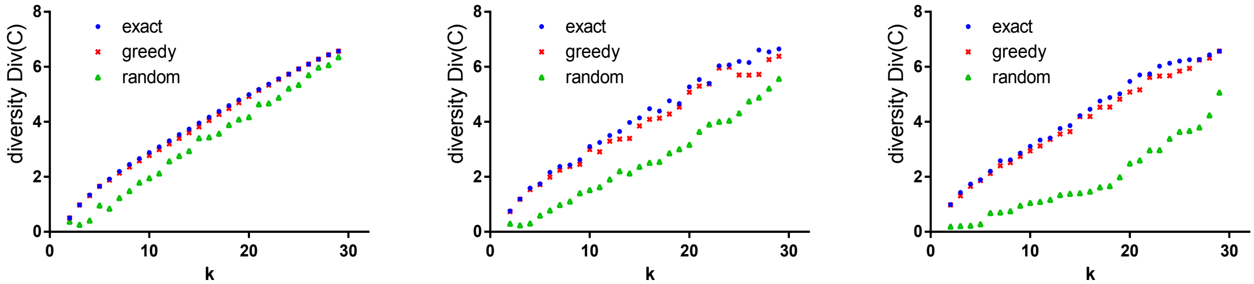
PROOF. The CMF of a Binomial distribution, F_B, can be represented in terms of the regularized incomplete beta function:

$$F_B(\theta; n, p) = (n - \theta)C_n^{\theta} \int_0^{1-p} t^{n-\theta-1}(1-t)^{\theta} dt \quad (9)$$

Facilitated with formula 9, we compute the partial derivative of $G_B(p)$ w.r.t p:

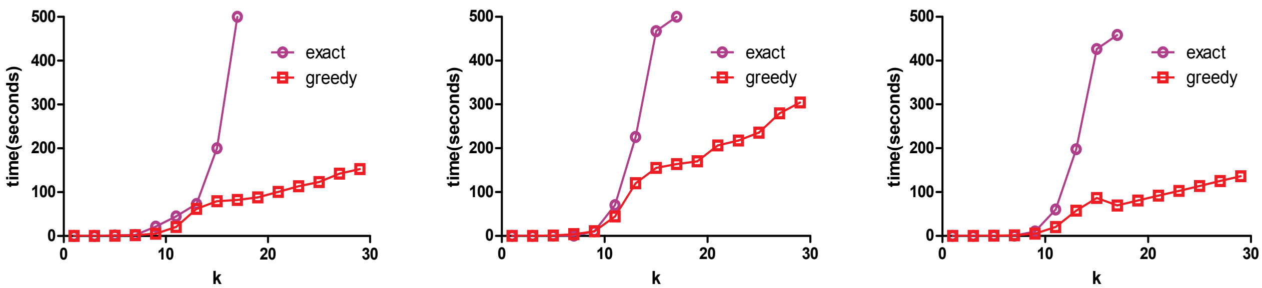
$$\begin{aligned}\frac{\partial G_B(p)}{\partial p} =& (n - \theta_2)C_n^{\theta_2} \frac{\partial \int_0^{1-p} t^{n-\theta_2-1}(1-t)^{\theta_2} dt}{\partial p} \\ & - (n - \theta_1)C_n^{\theta_1} \frac{\partial \int_0^{1-p} t^{n-\theta_1-1}(1-t)^{\theta_1} dt}{\partial p} \\ =& (n - \theta_2)C_n^{\theta_2}\{-(1-p)^{n-\theta_2-1}p^{\theta_2}\} \\ & - (n - \theta_1)C_n^{\theta_1}\{-(1-p)^{n-\theta_1-1}p^{\theta_1}\} \\ =& p^{\theta_1}(1-p)^{n-\theta_2-1}\{(n-\theta_1)C_n^{\theta_1}(1-p)^{\theta_2-\theta_1} \\ & - (n-\theta_2)C_n^{\theta_2}p^{\theta_2-\theta_1}\}\end{aligned} \quad (10)$$

Then, by solving equations $\frac{\partial G_B(p)}{\partial p} >= 0$ and $\frac{\partial G_B(p)}{\partial p} <= 0$, we have results $p <= \frac{1}{1 + (\frac{(n-\theta_2)C_n^{\theta_2}}{(n-\theta_1)C_n^{\theta_1}})^{\frac{1}{\theta_2 - \theta_1}}}$ and $p >=$



(a) Uniform Distribution (b) Normal Distribution (c) Zipf Distribution

Figure 4: Effectiveness of Methods for S-model with Various Distributions



(a) Uniform Distribution (b) Normal Distribution (c) Zipf Distribution

Figure 5: Efficiency of Methods for S-model with Various Distributions

$\frac{1}{1+(\frac{(n-\theta_2)C_n^{\theta_2}}{(n-\theta_1)C_n^{\theta_1}})^{\frac{1}{\theta_2-\theta_1}}}$, respectively, which completes the proof. □

Algorithms

Algorithm 2 (and other algorithms for E-kKP problem) can be reused for finding the approximate solution based on binomial approximation. Specifically, we define

$$\Omega_B := \frac{|N|}{(1+(\frac{(n-\theta_2)C_n^{\theta_2}}{(n-\theta_1)C_n^{\theta_1}})^{\frac{1}{\theta_2-\theta_1}}}$$

and the solution subset is between $S'_l = Bt(k, \Omega_B, N)$ and $S'_r = N - Bt(|N| - k, \sum_{t\in N} Pr(t = 1) - \Omega_B, N)$. Here, let $p_l = \sum_{t\in S'_l} Pr(t = 1)$ and $p_r = \sum_{t\in S'_r} Pr(t = 1)$, then we return S'_l as result if $G_B(p_l) > G_B(p_r)$; otherwise return S'_r.

4. EXPERIMENTAL EVALUATION

In this section, we present our experimental evaluation of the performances of T-model and S-model, as well as an experimental study of the crowd selection problem, namely finding the optimal set of workers with a given budget. The goal of our experiments is twofold: first, we study the effect of different parameters for the proposed algorithms; second, we compare the two proposed algorithms with a baseline algorithm, that is, selecting the workers randomly. In order to explore the various settings of parameter values in our methods, we have used synthetic data for the testing. In addition, we verify the effectiveness of our methods on data from the Foursquare [1], a very popular social network. Specifically, we used the Foursquare API to gather sample data of the existing venues and the tips posted on them. In particular, for each collected venue, the crawler collects all its tips, the identifications of the users who posted each of them. Our crawler ran from March 15th to May 19th, which collected data from 69,423 users. Additionally, to evaluate the practicability of the proposed models, we conducted a case study on Amazon Mechanical Turk (AMT).

All the experiments are conducted on a server equipped with Intel(R) Core(TM)i7 3.40GHz PC and 16GB memory, running on Microsoft Windows 7.

4.1 Experiments on S-model

We first conducted evaluation on S-model. In particular, we compared the proposed greedy algorithm, namely *greedy*, with two alternative methods- (1) *exact*: a brute-force algorithm, which computes the exact optimal solution; (2) *random*: the workers are selected randomly. Due to the high computational cost for the exact algorithm, we only generate a small data set with 30 workers. Each pair of workers is assigned a similarity ranging from -1 to 0 (so $Div(C) > 0$), following three different distributions - Uniform, Normal and Zipf.

Effectiveness: We generated 100 such data sets, and reported their average performance in Figure 4. Note the x-axis denotes the budget number of workers to be enlisted, and y-axis indicates the diversity of the selected crowd.

It is straightforward to interpret our findings: from the experimental results, we can see that *greedy* well approximates the performance of the *exact*. This is consistent with our theoretical analysis that *greedy* performs an approximation guarantee of 63%, as shown in Section 2.3. In addition, *greedy* outperforms *random* for all three distributions. We also find that the diversity grows with the increasing number of k for all three algorithms, which confirms the

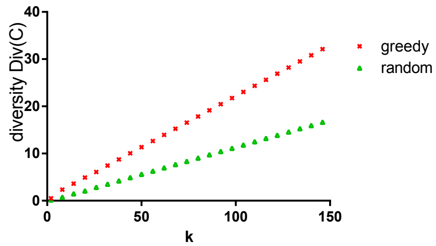


Figure 6: Effectiveness of S-model on Foursquare (Real) data

.

fact that large crowds tend to have high diversity. Another interesting finding is that, by comparing it with *random*, the advantages of *greedy* are more evident in Normal/Zipf distributions than in Uniform distributions. This is because Normal/Zipf distributions are skewed, thereby *random* is very likely to select the values around the mean, which leads to low diversity.

On the real data set, the exact algorithm cannot be performed due to its factorial time cost. So we only plotted the performance of *random* and *greedy*, as demonstrated in Figure 6. The result is basically consistent with the synthetic data.

Efficiency: In this subsection, we empirically examine the time-efficiency of the proposed algorithm for S-model. In particular, we compare the greedy algorithm (Algorithm 2) with the exact algorithm (Brute-force enumeration). As shown in Figure 5, the exact algorithm (denoted by *exact*) entails exponential computation time, and the greedy algorithm (*greedy*) is much more effective than *exact*. Please note that we stop *exact* after running it over 500 seconds.

4.2 Experiments on T-model

4.2.1 Synthetic Data

In this subsection, we demonstrate a series of experimental results on synthetic data. To simulate individual opinions without bias, in this section we produced synthetic datasets following three different distributions - normal, uniform and Zipf, each of which has varying mean values and variance values. The characteristics of K-Best selection are investigated with both Poisson Approximation and Binomial Approximation. Then we evaluate the efficiency and effectiveness of both methods.

The synthetic dataset is generated as follows: we generated 100 data sets, each including 30 candidate workers. The number of candidate workers is small because we want to use a brute-force algorithm to traverse the searching space, and find the absolute optimal solution. Then, we can evaluate how far the proposed approximation algorithm is from this optimum. The setting of parameters is: $k = 10, \theta_1 = 3, \theta_0 = 3$, $k = 15, \theta_1 = 5, \theta_0 = 5$ and $k = 20, \theta_1 = 6, \theta_0 = 6$.

The results of effectiveness are reported in Figure 7. In each subfigure of Figure 7, x-axis indicates the index of the 100 data sets, and y-axis denotes the value of $\tau(S)$, which is the function we try to maximize. The methods with poisson and binomial approximations are named 'poisson' and 'binomial', respectively. To better illustrate the advantage of the proposed methods, we also compare them with a baseline method, which randomly select workers, denoted by 'random'. From the experimental results, we can see that the performance of 'random' can be arbitrarily bad, while the 'poisson' and 'binomial' have similar performance, and well approximate the optimum. In addition, we present the comparison of efficiency in Figure 8. One can see that the approximation techniques are much more efficient than computing the exact solutions. Moreover, we observe that 'poisson' and 'binomial' have similar performance in terms of efficiency.

4.2.2 Real Data

In this subsection, we evaluated the proposed methods on real data sets from Foursquare. In particular, we select 10000 active workers (i.e. Foursquare users) from all the data collected. We evaluate sentiment of all the historical comments for each worker, and use average opinion sentiment value for this experiment. With this large data set, we examine the performance of the proposed algorithms with different settings of θ_0, θ_1 and k. In Figure 9, we use x-axis to denote the value of k, whereas θ_0 and θ_1 are set to be different portions of k.

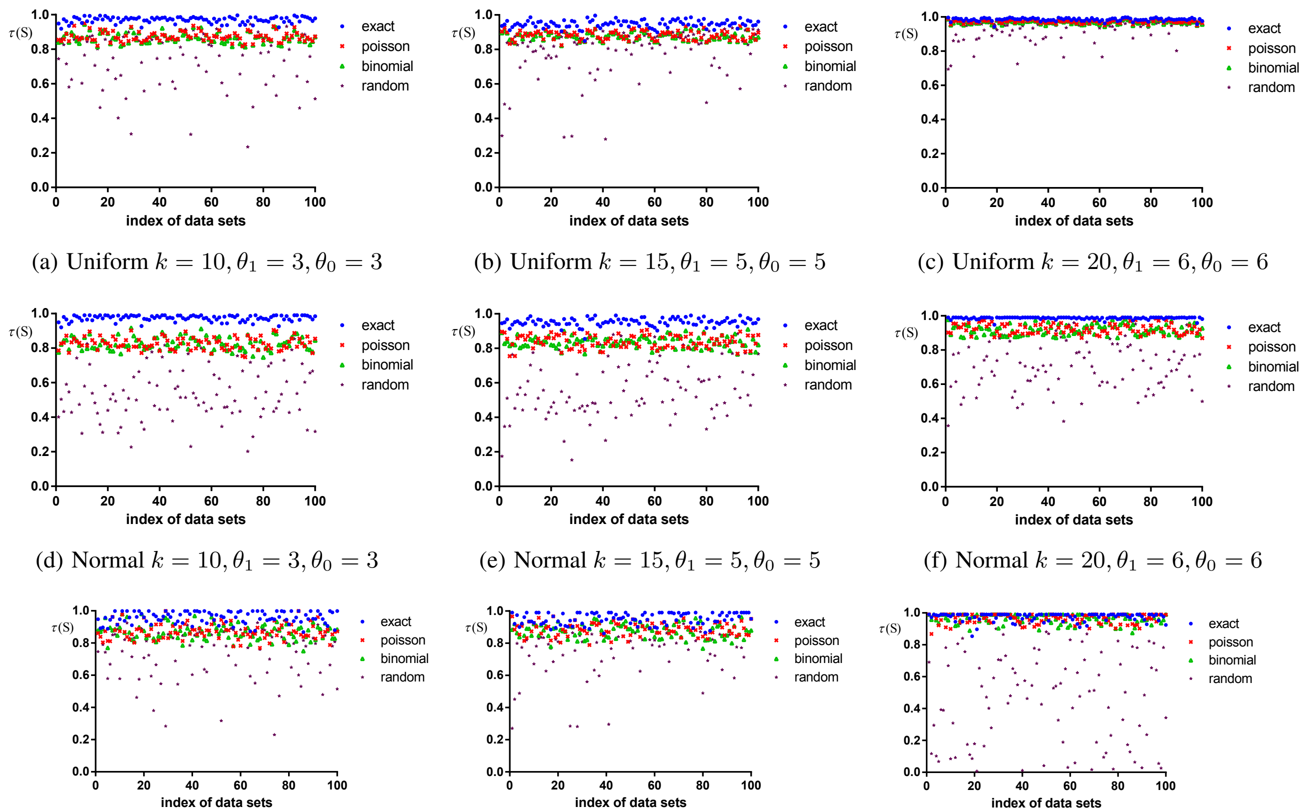
First, we can observe that the proposed approximation-based methods significantly outperforms the random baseline. In particular, the advantage of proposals is evident when θ_0 and θ_1 are far from the $k/2$, such as figures 9(a),9(d) and 9(h). Comparatively, when they are close to $k/2$, the performance of random baseline becomes better, but still worse than our proposals. This phenomenon can be explained by the Central Limit Theorem [29] - the sum of 0-1 random variables (i.e. a Poisson Binomial Distribution) is approximately a normal distribution, and the random baseline is more likely to pick the workers with probability close to the mean. So when the user's demand is also close to the mean, the random baseline would have a better performance. When the user's demand is far to the mean, randomly selecting workers is very unlikely to satisfy the user's demand. Overall speaking, our proposal demonstrates very stable and outstanding performance. Moreover, when k is fairly large, the user's demand can be almost 100% guaranteed.

4.3 Case Study

We conducted a case study to exhibit the *goodness* of crowds selected by our proposed models. In particular, we ask the crowds to produce pairwise comparisons for a number of restaurants. One thing worth noting is that the goodness of a crowdsourced result for restaurants is not *absolute*. Nevertheless, in order to present a fairly objective evaluation, we carefully select 40 pairs of restaurants, such that each of them is consistently ranked by three different third-party systems, namely *Yelp!* (http://www.yelp.com/), *Michelin* (http://www.michelin.com/), as well as *OpenRice* (http://www.openrice.com/). The pairwise comparisons agreed by all the systems are assumed to be the ground truth.

We publish questions on Amazon Mechanical Turk (AMT), which is a widely used crowdsourcing marketplace. Each question contains two restaurants, and requires a worker to provide comments (at least 200 words) on each restaurant and decide which one is better. We accept 100 workers for each question. We apply the S-model and T-model on the data obtained from AMT, and select a subset of workers out of the 100 for each pair of restaurants. Specifically, we adopt the distance function detailed in Appendix A.3 for S-model; and use the sentiment analysis tool from Natural Language Toolkit (NLTK [5]) for the T-model. To aggregate crowdsourced answers, we use the majority as the crowd's result. Moreover, for comparison, we randomly select the same number of workers, denoted by *rand*.

The size of the selected subset of workers is set to $11, 21, ..., 51$, and the proposed models consistently outperform *rand*. Due to the page limit, we demonstrate the precision and recall when the size is 21. In Figure 10, we use *rand*, *t-model* and *s-model* to denote the results for random selection, t-model and s-model, respectively. From the experimental results, we can see that the proposed models achieve fairly high precision and recall (70%+). Besides, we observe that *rand* has quite low precision and recall, which indicate that the diversity of opinion is very important for constructing a crowd.

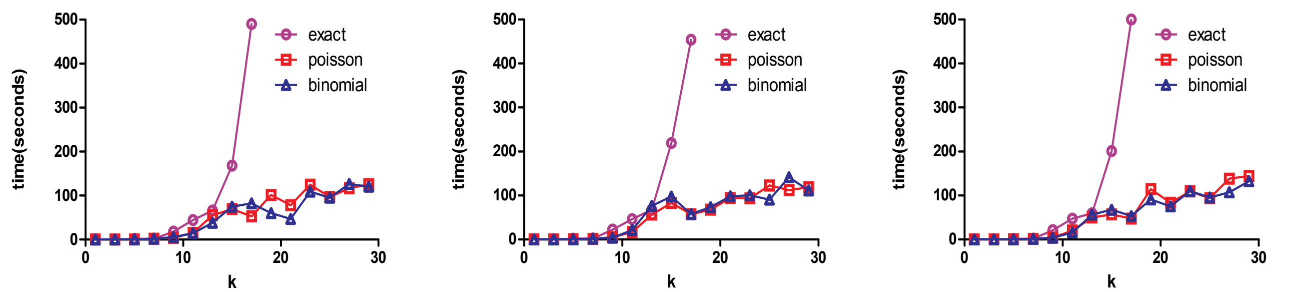


(a) Uniform $k = 10, \theta_1 = 3, \theta_0 = 3$ (b) Uniform $k = 15, \theta_1 = 5, \theta_0 = 5$ (c) Uniform $k = 20, \theta_1 = 6, \theta_0 = 6$

(d) Normal $k = 10, \theta_1 = 3, \theta_0 = 3$ (e) Normal $k = 15, \theta_1 = 5, \theta_0 = 5$ (f) Normal $k = 20, \theta_1 = 6, \theta_0 = 6$

(g) Zipf $k = 10, \theta_1 = 3, \theta_0 = 3$ (h) Zipf $k = 15, \theta_1 = 5, \theta_0 = 5$ (i) Zipf $k = 20, \theta_1 = 6, \theta_0 = 6$

Figure 7: Effectiveness of Methods with Poisson and Binomial Approximations



(a) Uniform Distribution (b) Normal Distribution (c) Zipf Distribution

Figure 8: Efficiency of Methods for T-model with Various Distributions

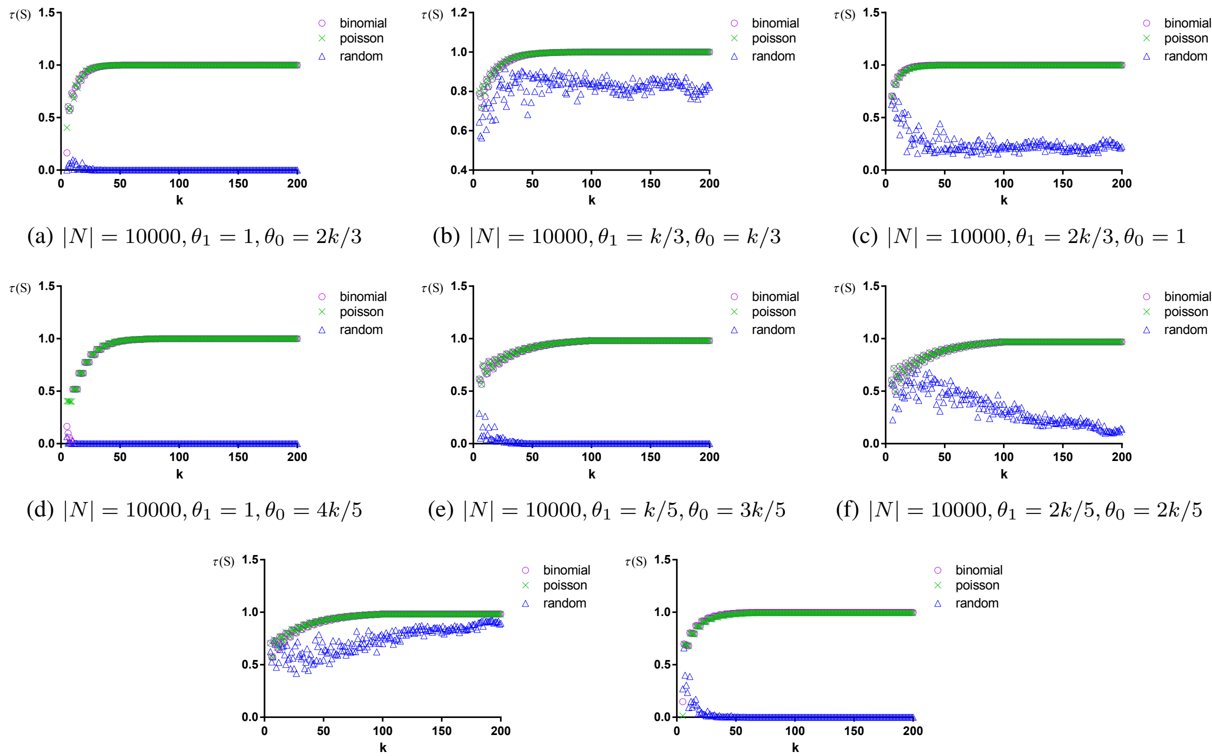
5. RELATED WORK

5.1 Crowd-based Queries

The recent development of crowdsourcing brings us a new opportunity to engage human intelligence into the process of answering queries (see [13] as a survey). Crowdsourcing provides a new problem-solving paradigm [8, 21], which has been blended into several research communities. In particular, crowdsourcing-based data management techniques have attracted many attentions in the database and data mining communities recently. In the practical viewpoint, [15] proposed and develop a query processing system using microtask-based crowdsourcing to answer queries. Moreover, in [26], a declarative query model is proposed to cooperate with standard relational database operators. In addition, in the viewpoint of theoretical study, many fundamental queries have been extensively studied, including filtering [25], max [17], sorting [22], join [22, 33], etc. Besides, crowdsourcing-based solutions of many complex algorithms are developed, such as categorization based on graph search [27], clustering [16], entity resolution [32, 34], analysis over social media [10], and tagging in social networks [12], trip planning [18], pattern mining [6] etc.

5.2 Team Formation

Another related problem in the field of data mining is *Team Formation Problem* [20]. Before taking diversity into consideration, previous Team Formation problems focus on satisfying the specific requirements of given tasks for certain skills which are possessed by different candidates experts. Normally, the cost of choos-



(a) $|N| = 10000, \theta_1 = 1, \theta_0 = 2k/3$ (b) $|N| = 10000, \theta_1 = k/3, \theta_0 = k/3$ (c) $|N| = 10000, \theta_1 = 2k/3, \theta_0 = 1$

(d) $|N| = 10000, \theta_1 = 1, \theta_0 = 4k/5$ (e) $|N| = 10000, \theta_1 = k/5, \theta_0 = 3k/5$ (f) $|N| = 10000, \theta_1 = 2k/5, \theta_0 = 2k/5$

(g) $|N| = 10000, \theta_1 = 3k/5, \theta_0 = k/5$ (h) $|N| = 10000, \theta_1 = 4k/5, \theta_0 = 1$

Figure 9: Testing on real data

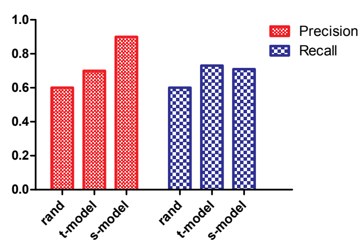


Figure 10: Precision and Recall on Case Study over AMT

ing one expert is also defined, e.g. influence on personal relationship and communication cost etc. Aside from using explicit graph constraints, some attempts of solving team formation problem are based on communication activities [9, 14].

The difference between Team Formation problem and ours is twofold. First, Team Formation mainly considers on individual capabilities, while we consider the crowd as a whole - the most capable workers may not make a wise crowd [31]. Second, we focus on the diversity of opinions of the crowd, which has not been addressed in the Team Formation problem.

5.3 Diversity of Opinions in Social Science

The importance of diversity of opinions for crowdsourcing is already well studied in the field of social science. In particular, [23] is known as one of the most representative book in the field. It highlights the importance of cognitive diversity for collective problem-solving (where diversity trumps ability), and takes a complex subject, moves beyond metaphor and mysticism and politics and places the claims of diversity's benefits on a solid intellectual foundation.

To our best knowledge, this is the first work of algorithmic study on a how to construct a wise crowd with the consideration of the diversity of opinion.

6. CONCLUSION AND FUTURE WORK

In this paper, we study how to construct a wise crowd with the consideration of diversity of opinions. In particular, two basic paradigms for worker selection is addressed - building a crowd waiting for tasks to come and selecting workers for a given task. Accordingly, we propose Similarity-driven (S-Model) and Task-driven Model (T-Model) for these two paradigms. Under both of the models, we propose efficient and effective algorithms to enlist workers with a budgeted constraint. We have verified the solutions with extensive experiments on both synthetic datasets and real data sets. The experimental studies demonstrate that the proposals are robust for varying parameters, and significantly outperform the baselines.

There are many further research directions to explore. One immediate future direction is how to consider the different influence of workers for the diversity of opinions. The influence may diminish the range of opinions, and polarize people's opinions making group feedback less reliable in guiding decision-makers. Influencers tend to improve people's confidence, but this so-called 'confidence effect' will boost an individual's confidence, while at the same time, decrease their accuracy. Another interesting dimension is to differentiate the cost for recruiting different workers, then the problem is to minimize the total cost while fulfilling the requirement of diversity. Besides, we are interested in designing better similarity/distance functions for our T-model.

7. ACKNOWLEDGEMENT

This work is supported in part by the Hong Kong RGC Project N_HKUST637/13, National Grand Fundamental Research 973 Program of China under Grant 2014CB340303, NSFC Grant No. 61328202, Microsoft Research Asia Gift Grant and Google Faculty Award 2013.

8. REFERENCES

[1] https://foursquare.com/.
[2] https://petitions.whitehouse.gov/.
[3] https://www.mturk.com/mturk/welcome.
[4] http://www.crowdflower.com/.
[5] http://www.nltk.org/.
[6] Y. Amsterdamer, Y. Grossman, T. Milo, and P. Senellart. Crowd mining. In *SIGMOD Conference*, pages 241–252, 2013.
[7] C. G. Andreas Krause. A note on the budgeted maximization of submodular functions. Technical report, School of Computer Science, Carnegie Mellon University, March 2005.
[8] D. C. Brabham. Crowdsourcing as a model for problem solving an introduction and cases. *Convergence February 2008 vol. 14 no. 1 75-90*, 2008.
[9] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom. Expertise identification using email communications. In *In CIKM 03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 528–531. ACM Press, 2003.
[10] C. C. Cao, J. She, Y. Tong, and L. Chen. Whom to ask? jury selection for decision making tasks on micro-blog services. *PVLDB*, 5(11):1495–1506, 2012.
[11] A. Caprara, H. Kellerer, U. Pferschy, and D. Pisinger. Approximation algorithms for knapsack problems with cardinality constraints. *European Journal of Operational Research*, 123(2):333 – 345, 2000.
[12] M. Das, S. Thirumuruganathan, S. Amer-Yahia, G. Das, and C. Yu. Who tags what? an analysis framework. *PVLDB*, 5(11):1567–1578, 2012.
[13] A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the world-wide web. *Commun. ACM*, 54(4):86–96, 2011.
[14] B. Dom, I. Eiron, A. Cozzi, and Y. Zhang. Graph-based ranking algorithms for e-mail expertise analysis. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '03, pages 42–48, New York, NY, USA, 2003. ACM.
[15] A. Feng, M. J. Franklin, D. Kossmann, T. Kraska, S. Madden, S. Ramesh, A. Wang, and R. Xin. Crowddb: Query processing with the vldb crowd. *PVLDB*, 4(12):1387–1390, 2011.
[16] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In *NIPS*, pages 558–566, 2011.
[17] S. Guo, A. G. Parameswaran, and H. Garcia-Molina. So who won?: dynamic max discovery with the crowd. In *SIGMOD Conference*, pages 385–396, 2012.
[18] H. Kaplan, I. Lotosh, T. Milo, and S. Novgorodov. Answering planning queries with the crowd. *PVLDB*, 6(9):697–708, 2013.
[19] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems*. Springer, 2004.
[20] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 467–476, New York, NY, USA, 2009. ACM.
[21] T. Malone, R. Laubacher, and C. Dellarocas. Harnessing crowds: Mapping the genome of collective intelligence. Research Paper No. 4732-09, MIT, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, USA, February 2009. Sloan Research Paper No. 4732-09.
[22] A. Marcus, E. Wu, D. R. Karger, S. Madden, and R. C. Miller. Human-powered sorts and joins. *PVLDB*, 5(1):13–24, 2011.
[23] S. Page. *The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies*. Princeton University Press, 2007.
[24] S. Page. Making the difference: Applying a logic of diversity. 2007.
[25] A. G. Parameswaran, H. Garcia-Molina, H. Park, N. Polyzotis, A. Ramesh, and J. Widom. Crowdscreen: algorithms for filtering data with humans. In *SIGMOD Conference*, pages 361–372, 2012.
[26] A. G. Parameswaran and N. Polyzotis. Answering queries using humans, algorithms and databases. In *CIDR*, pages 160–166, 2011.
[27] A. G. Parameswaran, A. D. Sarma, H. Garcia-Molina, N. Polyzotis, and J. Widom. Human-assisted graph search: it's okay to ask questions. *PVLDB*, 4(5):267–278, 2011.
[28] M. F. Porter. Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.
[29] B. Roos. Binomial approximation to the poisson binomial distribution: The krawtchouk expansion. *Theory of Probability and its Applications*, 45(2):258–272 (2000) and Teor. Veroyatn. Primen. 45, No. 2, 328–344, 2000.
[30] C. Stein. *Approximate Computation of Expectations.* Hayward, Calif.: Institute of Mathematical Statistics, 1986.
[31] J. Surowiecki. *The Wisdom of Crowds*. Anchor, 2005.
[32] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *PVLDB*, 5(11):1483–1494, 2012.
[33] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng. Leveraging transitive relations for crowdsourced joins. In *SIGMOD Conference*, pages 229–240, 2013.
[34] S. E. Whang, P. Lofgren, and H. Garcia-Molina. Question selection for crowd entity resolution. *PVLDB*, 6(6):349–360, 2013.
[35] C. J. Zhang, L. Chen, H. V. Jagadish, and C. C. Cao. Reducing uncertainty of schema matching via crowdsourcing. *PVLDB*, 6(9):757–768, 2013.

APPENDIX

A. SIMILARITY MEASUREMENT OF S-MODEL

We assume we are given a set $\mathcal{T}$ of historical tasks and a set $\mathcal{W}$ of workers. Each Task $t \in \mathcal{T}$ is associated with a unique identifier t_{id} and a set of workers $t_W \subseteq \mathcal{W}$ who have worked on t. A record e is a triple of the form $[t_{id}, w_{id}, features]$ where w_{id} is a unique identifier of the worker and *features* contain certain useful information(e.g. correctness, latency, submission time, etc.) which this record refers. The set of all records belonging to a worker w forms the experience of the worker denoted by *experience(w)*. Without loss of generality, we assume a worker has at most one record per task.

For each task t, we characterize it with a set of attributes such as category, complexity, workload, requester and nature (e.g. problem solving task, survey). Similarly, a worker w could carry demographic information such as gender, age, expertises, occupation and geographic location.

A.1 Pairwise Relevance

In a typical crowdsourcing environment, *relevance* between a task and a candidate worker serves as an important criterion to guarantee the quality of the crowdsourced results. Therefore, we first introduce the definition and measurement of relevance before formally defining the concept of diversity.

DEFINITION A.1 (PAIRWISE RELEVANCE). *For a given set of potential workers $\mathcal{W}$ and tasks $\mathcal{T}$, the relevance between any worker and task is computed by a given function $Rel(w_i, t_i) = 1/d_{rel}(w_i, t)$, where $w_i, \in \mathcal{W}$, $t_i \in \mathcal{T}$.*

Given a task t and a threshold radius r, we define the set of workers relevant to t as the set of workers $w_i \in \mathcal{W}$ within the relevant distance r from t, e.g. $Rel(w_i, t) \leq r$. For example, the distance between a task and a worker (represented by their sets of features x and y) could be computed by *Jaccard distance*, e.g. $d_{rel}(x, y) = 1 - Jaccard(x, y)$. In this paper, features are extracted from the descriptions of tasks and profile of workers by running Porter Algorithm [28].

A.2 Pairwise Profile-Based Diversity

Intuitively, we define diversity between two workers w_i and w_j as a function of entities extracted from their profiles.

DEFINITION A.2 (PAIRWISE SIMILARITY). *For a given set of potential crowdsourcing workers W, the diversity of any two workers is computed by the similarity function $Sim(w_i, w_j) = Jaccard(w_i, w_j)$, where $w_i, w_j \in W$.*

Thus, two workers maybe similar because they have the same gender and age, but still different(diverse), if one is living in Hong Kong and the other in New York.

A.3 Pairwise Experience-Based Diversity

For a more sophisticated measurement, we denote the *experience* $\mathcal{E}$ as a collection of historical records of each worker. Diversity between two workers w_i and w_j is defined as a function of experience engaged by works through their activities on the historic tasks. There are two steps for inferring pairwise experience-based diversity of two workers

A.3.1 Probabilistic Topic Model

We use a probabilistic model to model user's experience E_i as a unordered collection of words (a.k.a. bag of words). Such collection of words (i.e. task identifier, task features, etc.) can be extracted from the records of different tasks that the worker has been performed. Specifically, we use a mixture model in which each component corresponds to one of K different topics. Let π_k, for $k = 1, ..., K$, denote the prior probability that a collection contains topic T_k. For each topic, there is a corresponding multinomial distribution over the M distinct words in all collections. Let μ_{kj}, for $k = 1, ..., K, j = 1, ..., M$, denote the probability that topic T_k contains word ω_j in all collections. Suppose a collection U_i contains a total N_i words in which each word is generated i.i.d from the mixture model above. The number of occurrences of word ω_j in E_i is equal to n_{ij}, which follows that $\sum_{j=i}^{M} n_{ij} = N_i$. We assume there are N i.i.d collections denoted by $E_1, E_2, ..., E_N$ that associated with N users.

Let $\mathbf{\Phi} = (\pi_k, \mu_{kj})$ denote the model parameters. We estimate $\mathbf{\Phi}$ using EM, the E-step computes for each collection D_i the posterior probability that D_i belongs to topic T_k given the model parameters $\mathbf{\Phi}^t$ of the previous iteration. We can apply Bayes' rule to express $P(T_k|E_i, \mathbf{\Phi}^t)$ as

$$\begin{aligned} p(T_k|E_i, \mathbf{\Phi}^t) &= \frac{P(T_k)P(E_i|T_i, \mathbf{\Phi}^t)}{\sum_{l=1}^{K} P(T_l)P(E_i|T_l, \mathbf{\Phi}^t)} \\ &= \frac{\pi_k^t \prod_{j=1}^{M} (\mu_{kj})^{n_{ij}}}{\sum_{l=1}^{K} \pi_l^t \prod_{j=1}^{M} (\mu_{lj}^t)^{n_{ij}}} \end{aligned} \quad (11)$$

In the M-step, to maximize $\mathbf{\Psi}(\mathbf{\Phi}|\mathbf{\Phi}^t)$ w.r.t $\mathbf{\Phi}$ to obtain the next estimate $\mathbf{\Phi}^{t+1}$, we can obtain

$$\pi_k^{t+1} = \frac{1}{N} \sum_{i=1}^{N} h_k^{(i)} \quad (12)$$

We note that there are K constraints due to the multinomial distribution for the K topics:

$$\sum_{j=1}^{M} \mu_{kj} = 1 \qquad k = 1, ..., K \quad (13)$$

To solve a constrained optimization problem, we introduce K Lagrange multipliers.

$$\frac{\partial}{\partial \mu_{kj}} [\sum_{i=1}^{N} \sum_{k=1}^{K} h_k^{(i)} \log P(E_i|T_k, \mathbf{\Phi}) - \sum_{k=1}^{K} \lambda_k (\sum_{j=1}^{M} \mu_{kj} - 1)] = 0 \quad (14)$$

where $h_k^{(i)}$ denotes $P(T_k|E_i, \mathbf{\Phi}^t)$. This gives

$$\mu_{kj}^{t+1} = \frac{\sum_{i=1}^{N} h_k^{(i)} n_{ij}}{\sum_{j'=1}^{M} \sum_{i=1}^{N} h_k^{(i)} n_{ij'}} \quad (15)$$

The EM algorithm converges to a stationary point of the likelihood function. Then we obtain the probabilistic topic distribution, which is denote by $w_i.\varphi$, of each worker.

A.3.2 Worker Distance Function

Given two workers $w_i, w_j \in \mathcal{W}$, the topic distance between two workers is defined as

$$D(w_i, w_j) = KL(w_i.\varphi || w_j.\varphi)$$

where $KL(.)$ measures the distance between the topic distributions $w_i.\varphi$ and $w_j.\varphi$, i.e.

$$KL(w_i.\varphi || w_j.\varphi) = \sum_i Pr(w_i.\varphi(i)) \log \frac{Pr(w_i.\varphi(i))}{Pr(w_j.\varphi(i))}$$

.Then we have $Sim(w_i, w_j) = -D(w_i, w_j)$

B. POISSON AND BINOMIAL APPROXIMATION

In Section 3, we use Poisson distribution and Binomial distribution to approximate Poisson Binomial distribution. Here, we conclude the quality of approximation in [30, 29].

Let X_1,X_2,...,X_n be a set of Bernoulli trials such that $Pr(X_j = 1) = p_j$ and $X = \sum_{j=1}^{n} X_j$. Then X follows a Poisson binomial distribution. Suppose $\mu = E[X] = \sum_{j=1}^{n} p_j$. The probability of $X = i$ and $X \leq i$ can be approximated by the probability density function (PDF) and cumulative mass function (CMF) of Poisson distribution and Binomial distribution.

Poisson Approximation:

$$Pr(X \leq i) \approx F_P(i, u) = \frac{\Gamma(i + 1, u)}{i!} e^{-u}$$

[30] provides an upper bound of the error of the approximation:

$$|Pr(X \leq i) - F_p(i, u)| \leq min(\mu - 1 \wedge 1) \sum_{j=1}^{n} p_j^2$$

for $i = 0, 1, 2, ..., n$ Clearly, this upper bound of the error is greater than or equal to 0. When $\mu \in [0, 1]$

$$|Pr(X \leq i) - F_p(i, u)| = \sum_{j=1}^{n} p_j^2 \leq \sum_{j=1}^{n} p_j \leq 1$$

When $\mu \in [1, +\infty)$

$$|Pr(X \leq i) - F_p(i, u)| = \frac{\sum_{j=1}^{n} p_j^2}{\sum_{j=1}^{n} p_j} \leq \frac{\sum_{j=1}^{n} p_j}{\sum_{j=1}^{n} p_j} = 1$$

So, in ether case:

$$0 \leq |Pr(X \leq i) - F_p(i, u)| \leq 1$$

Binomial Approximation:

In [29], the metric of error is defined as

$$d_{err} = \frac{1}{2} \sum_{i \in \mathbb{Z}} |f_B(X = i) - Bi(i; n, p)|$$

By using binomial distribution $Bi(i; n, p)$ approximate the distribution of X, where $p = \mu/n$, we have

$$d_{err} \leq \frac{1 - p^{n+1} - (1 - p)^{n+1}}{(n + 1)p(1 - p)} \sum_{i=1}^{n} (p_i - p)^2$$