Where To: Crowd-Aided Path Selection by Selective Bayesian Network

Chen Zhang, *Member, IEEE,* Haodi Zhang, Weiteng Xie, Nan Liu, Kaishun Wu, *Member, IEEE,* and Lei Chen *Fellow, IEEE*

Abstract—With the wide usage of geo-positioning services (GPS), GPS-based navigation systems have become more and more of an integral part of people's daily lives. GPS-based navigation systems usually suggest multiple paths for a pair of given source and target. Therefore, users become perplexed when trying to select the best one among them, namely the problem of *best path selection*. Too many suggested paths may jeopardize the usability of the recommendation data, and decrease user satisfaction. Although the existing studies have already partially relieved this problem through integrating historical traffic logs or updating traffic conditions periodically, their solutions neglect the potential contribution of human experiences. In this paper, we resort to crowdsourcing to ease the pain of best path selection. However, the first step of using the crowd is to ask the right questions. For best path selection problem, the simple questions (e.g. binary voting) on crowdsourcing platforms cannot be directly applied to road networks. Thus, in this paper, we have made the first contribution by designing two right types of questions, namely Routing Query (RQ) to ask the crowd to decide the direction at each road intersection. Secondly, we propose a series of efficient algorithms to dynamically manage the questions in order to reduce the selection hardness within a limited budget. In particular, we show that there are two factors affecting the informativeness of a question: the randomness (entropy) of the question and the structural position of the road intersection. Furthermore, we extend the framework to enable multiple RQs per round. To ease the pain of the sample sensitiveness, we propose a new approach to reduce the selection hardness by reasoning on a so-called *Selective Bayesian network*. We compare our approach against several baselines, and the effectiveness and efficiency of our proposal are verified by the results in simulations and experiments on real-world datasets. The experimental results show that, even the Selective Bayesian Network provides only partial information of causality, the performance on the reduction of the selection hardness are dramatically improved, especially when the size of samples are relatively small.

Index Terms—Crowdsourcing, Approximation Algorithm, Path Selection

1 Introduction

With the rapid development of information technology and data science, the scale of diverse data is getting larger and larger. For instance, the global positioning systems makes the real-time navigation systems commonly used in daily life. With the data collected from mobile devices, a good navigation system gives optimal routes between given locations. In realistic applications, however, the selection of the best path can be very challenging, especially in those large-scaled domains. For a road network with a large amount of paths and crossroads, it might be difficult to maintain precise information of the entire map all the time. To address the problem, quite a few existing studies integrate historical traffic logs or periodically update traffic conditions. However they usually neglect the potential contribution of human experience.

In our previous work [1], a crowd-aided path selection framework is proposed to resort to crowdsourcing for best route selection. The framework successfully leverages the human expertise for the task. The main idea is to first design questions in suitable form for crowdsourcing workers, and then decide the best path with help of the crowdsourced answers. As the query budget is usually limited, the selection of a proper set of Human Intelligence Tasks (HITs) is very important. A series of efficient algorithms are also proposed in [1] to dynamically manage the questions in order to reduce the selection hardness within a limited budget. The sampling-based approach works for best route selection tasks but is very sensitive to the quality and the quantity of the samples. The main reason is that, without considering the probabilistic causalities embedded in the spacial topology, it is actually difficult to precisely estimate the underlying relations merely by sampling.

Therefore, this paper makes the first contribution by proposing a natural way to build up a so-called Selective Bayesian Network as a reasoning tool. The spacial causalities embedded in the network can remarkably control the influence of the noises and sampling bias. Secondly, we propose an effective and efficient algorithm to select the most valuable set of queries for the crowd, with the help of the Selective Bayesian Network. Finally we compare our proposal with several baselines with varying sample size, query batch size, error rate and budget. The experimental result shows that our method dominates others both on simulations and on real datasets.

1.1 Candidate Routes and Measurement

As a motivating example, some PhD student who is new to some city needs to go to the university from her apartment every

- *C. Zhang is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, SAR China. E-mail: c4zhang@comp.polyu.edu.hk.*
- *H. Zhang is with the Shanghai Research Center for Brain Science and Brain-Inspired Intelligence, Shanghai, China, and the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. E-mail: zhanghd.ustc@gmail.com.*
- *W. Xie and K. Wu are with the College of Computer Science and Software Engineering, Shenzhen University and Guangdong Laboratory of Artificial Intelligence and Digital Economy, Shenzhen, China. E-mail: {wtxie, wu}@szu.edu.cn.*
- *N. Liu is with the College of Engineering, University of Michigan, US. E-mail: liunan@umich.edu.*
- *L. Chen is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, SAR China. E-mail: leichen@cse.ust.hk.*

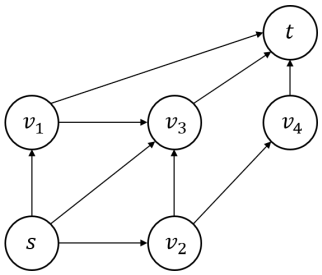


Fig. 1. Candidate Routes

TABLE 1
Route Distribution and Routing Queries

Candidate Routes	probability
R_1=$\big((s,v_1),(v_1,t)\big)$	0.1
R_2=$\big((s,v_1),(v_1,v_3),(v_3,t)\big)$	0.1
R_3=$\big((s,v_2),(v_2,v_3),(v_3,t)\big)$	0.4
R_4=$\big((s,v_2),(v_2,v_4),(v_4,t)\big)$	0.1
R_5=$\big((s,v_3),(v_3,t)\big)$	0.3
Routing Queries	**pmf over C_i**
$Q_1:(s,C=\{v_1,v_2,v_3\},t)$	$0.2, 0.5, 0.3$
$Q_2:(v_1,C=\{v_3,t\},t)$	$0.5, 0.5$
$Q_3:(v_2,C=\{v_3,v_4\},t)$	$0.8, 0.2$
Binary Routing Query	**pmf over $\{yes,no\}$**
$BQ_1:(s,C=\{v_1\},t)$	$0.2, 0.8$
$BQ_2:(s,C=\{v_2\},t)$	$0.5, 0.5$
$BQ_3:(s,C=\{v_3\},t)$	$0.3, 0.7$
$BQ_4:(v_1,C=\{v_3\},t)$	$0.5, 0.5$
$BQ_5:(v_1,C=\{t\},t)$	$0.5, 0.5$
$BQ_6:(v_2,C=\{v_3\},t)$	$0.8, 0.2$
$BQ_7:(v_2,C=\{v_4\},t)$	$0.2, 0.8$

morning. A navigation service usually suggests five different paths to her, namely by taxi, Uber, bus, subway and ferry. Taxi and Uber are convenient and comfortable, but quite expensive; while buses and the subway are fairly affordable, but usually slow and crowded; and the ferry is cheapest but also slowest. If the cost of all paths above can be perfectly evaluated by some value function or model, the best path is simply the shortest one. However, such a value function or model is usually absent, as the cost may be influenced by many dynamic factors that are difficult, sometimes impossible, to be quantitatively modeled. Instead, there are only statistics or observations with noises available for the estimation of the best route. To integrate and analyze the complex, multi-sourced statistics via a completely explicit model is quite challenging. However, for humans, experienced drivers for instance, it might be relatively easy to make a quick yet acceptable assessment in such circumstances. Consequently, many systems consider the paths preferred by humans [2], and produce not just one best path, but rather a set of paths. However, to address the 'Painful Options' problem [3], a most likely best route needs to be further selected. In some existing work [4], the task is formalized as predicting the spatial transition patterns of the trips. Several proposals [5], [6] have revealed that the transition patterns of traffic are usually highly skewed and unbalanced: some paths are more likely to be traveled than others. Thus, we follow the problem formalism in [1], [4] and presume a set of candidate routes with a given distribution as the input. It worth mentioning that the payment for the help from the crowd is very important. If the cost is too high, She should simply take a taxi without worrying about the best path. However, the taxi drivers may also be faced with multiple choices of routes, and also potentially need some help from some crowd, for instance other drivers.

The motivating example above can be specifically demonstrated by the distribution in Table 1, over a road network showed in Figure 1. There are totally 5 candidate routes from s to t, with the probabilities of being the best route 0.1, 0.1, 0.4, 0.1, 0.3, respectively. We first have to give a measurement to quantify the hardness selecting the best one from these candidates. As in many previous research work [1], [7], [8], we consider the best route as a discrete random variable defined over the set of the candidate paths, and use the Shannon entropy to measure the selection hardness. For a given candidate value set S for discrete random variable x, the entropy of x is $H = -\sum_{s\in S} Pr(x = s)\log Pr(x = s)$. So for a given set of candidate routes $\mathbb{R}$, the selection hardness of the best route BR, denoted by $H(BR)$, is

$$H(BR) = -\sum_{R\in\mathbb{R}} Pr(BR = R)\cdot\log Pr(BR = R) \qquad (1)$$

In the rest of the paper, we use $Pr(R)$ as the abbreviation of $Pr(BR = R)$. If the distribution over the candidate paths is relatively skewed, i.e. there is some route with a dominant probability to be the best route, then the hardness of the selection is quite low, as well as the entropy. In particular, if the candidate set is a singleton, the entropy is $1\cdot log1 = 0$, i.e. there is no difficulty at all to select the best route. When the distribution over the candidates is quite balanced, the selection is quite difficult, and the entropy value is the highest given a uniform distribution. In the example above in Figure 1, the hardness of selecting the best route is $H(BR) = -(3\cdot 0.1\cdot\log 0.1+0.4\cdot\log 0.4+0.3\cdot\log 0.3) = 0.616$.

It worth mentioning that the distribution of the candidate paths can be obtained in multiple ways. There have been many related research works. A straightforward idea is to use the historical trajectory data. In [2], the distribution is inferred by mining the frequent paths chosen by experienced drivers. It is also reasonable to let the user to initialize the distribution according to personal preference [9]. For a recommendation system integrated with multiple routing algorithms, a possible way is to train and test the algorithms on a large number of queries, and each of the methods is assigned with a probability based on its average performance. For some learning-based method, e.g. deep reinforcement learning [10], the output is already a distribution of candidate choices. For instance in [4], a deep probabilistic model is proposed to predict the most likely traveling route on the road network, which unifies three key explanatory factors. To enable effectively sharing the statistical strength, they also proposed an adjoin generative model to learn representations of k-destination proxies. In this paper, we assume that the distribution of the routes has already been given by some of the above methods.

1.2 Crowd-Aided Best Path Selection

To leverage human expertise for the route selection, we need to first determine a HIT (Human Intelligent Task) design. We follow the design of the crowdsourcing task and the queries in our previous work [1]. A Human Intelligence Task (HIT) is in the form of a Routing Query, which is a tuple (v_{st}, C, v_{tg}), where v_{st} is the *starting vertex* of the query, C is the set of candidate directions, and v_{tg} is the *target vertex*. Intuitively, such a query Q is asking that, if the current position is at the starting vertex v_{st}, in order to reach the target vertex v_{tg}, which direction in the candidate set C should be chosen. In Table 1, there are three

routing queries: Q_1, Q_2 and Q_3, each of which contains a starting vertex, a target vertex, and a candidate set of next directions. For instance, Q_2 is a query about which direction to choose to get to t, if one is currently at v_1. There are two choices available, either to go through v_3 and then to t, or directly go to t. The crowdsourcing task is to give an answer, denoted by A_{Q_2}, which is either v_3 or t.

Please notice that there is a probability mass function (pmf) available in the table, which decides the probabilities of different crowdsourced answers for some given Q. The probability mass function needs to submit to the distribution of the candidate routes RS. Once the distribution is given, the pmf can be determined. The distribution and the pmf can be regarded as the long-term statistics. For example, without any observation at a specific time, the probability of the best route in Figure 1 to be R_1 is statistically 0.1, and the probability of $A(Q_1) = v_1$ is statistically 0.2. However, if a crowdsourcing worker has been placed in a specific environment where the best route is already deterministic, which is R_3 for instance, i.e. with the observation $BR = R_2$, the probability of $A(Q_1) = v_1$ then becomes 0, and $Pr(A(Q_1) = v_2)$ is 1, if noises are not concerned. It is commonly accepted that crowdsourcing works best when the human intelligent tasks can be decomposed into very simple pieces [11]. So asking which closely next direction is the best choice, as what we did, is much better than generally asking which route is the best one. Moreover, the knowledge about the traffic condition sometimes comes from real-time observation, which is usually partial - a human driver despite his/her rich experience is only able to observe the traffic condition in a limited scope. With the simple pieces of the tasks, the crowdsourcing workers can make use of their expertise or real-time observation easily.

The crowdsourced answers for the simple tasks are then collected and integrated for updating the distribution of the routes. In [1], the query above can be further decomposed into smaller pieces, namely binary routing queries (BRQ). Each $Q = (v_{st}, C, v_{tg})$ can be broken down into BQs, each of which contains a singleton candidate direction set, and the answer is either yes or no. For instance, Q_1 in Table 1 is decomposed into BQ_1, BQ_2 and BQ_3, and Q_2 is decomposed into BQ_4 and BQ_5. A binary routing query set is easier for the crowd to answer, yet in expressiveness and efficiency it is equivalent with the corresponding routing query set. The framework of crowd-aided route selection [1] is as follows: given a route set $\mathbb{R}$ and a budget limit B of RQ numbers,

1) select k queries (denoted as S_k) to ask the crowd, to reduce the selection hardness as much as possible,
2) update the probabilities of all routes in $\mathbb{R}$ according to the crowdsourced k answers (denoted as A_{S_k}),
3) repeat 1 and 2 until budget of B is used up, and then report the most likely best route.

In [1], the selection of the best set of queries is done by a sampling-based approach, which estimates the mutual information between BR and A_{S_k} by sampling. It neglects spacial relationship between BR and A_{S_k}, resulting that the performance is very sensitive to the quality and size of the sample set. In this paper we propose a method to address the problem.

1.3 Challenges and Contributions

We present the challenges in the following aspects.

- **Idleness of topological information:** a main challenge is how to utilize the spacial information in the map, to accelerate the hardness reduction of selecting the best route. Generally, the routing queries that are given to the crowd are not independent with each other, which makes it very difficult to precisely evaluate different combinations of queries. It is crucial to find an efficient way to make use of the topological causalities among the queries when we compute the mutual information between the selection hardness and the crowdsourced answers.
- **Uncertainty from noises and sampling bias:** existing sampling-based algorithm suffers from the possible noises in the crowdsourced answers and sample insufficiency. As the answers from the crowd are not always correct, the bias in the samples might be magnified by the noises, in particular when the sample size is not large enough. Thus, we need to find an effective mechanism to control the uncertainty brought about by the error of the crowd.
- **Computational complexity on large maps:** suppose that we have already known how to leverage the spacial information for the route selection, the efficiency is another important concern. As the given map might be quite large in reality, the designed algorithm has to be highly efficient for online usage.

To address the problems above, we propose a new approach for crowd-aided route selection. We summarize our new contributions as follows:

- Firstly, we give the exact solution for selecting the optimal k RQs in the framework of crowd-aided route selection. We propose a so-called Selective Bayesian Network for representing and leveraging the knowledge of spacial causalities, and propose a natural method to build the networks, as described in detail in Section 4.2.
- Secondly, we propose an efficient algorithm that selects most valuable routing queries and then suggests best routes, by reasoning on the Selective Bayesian Network, which successfully reduces the uncertainty yielded by the sampling bias and the noises in the crowdsourced answers, as presented in Sections 4.3 and 4.4.
- Thirdly, we modularize the algorithm to enable topological information reuse, thus improve the efficiency of the hardness reduction, as illustrated in Section 4.4.
- Finally, we compare different algorithms both on synthetic data and real-world road networks with varying sample size, query batch size, error rate and budget. It turns out that our approach dominates others, as shown in Section 5.

2 Definitions and Problem Statement

In this section, we present the core definitions and related notations, then formally state the problem.

Definition 2.1. *Given a source vertex s and a target vertex t over a directed graph G, a* candidate route *is a sequence of edges $R = (e_1, \ldots, e_n)$, such that s is the head of e_1, t is the tail of e_n, and the sequence $e_1, \ldots, e_n$ is a directed path in G.*

For a given vertex v and a candidate path R, if R *goes through* v, we denote it as $R \to v$, and $R \not\to v$ denotes that R does not go through v. For an edge $e = (v_i, v_j)$ in G such that $R \to v_i$ and $R \to v_j$, i.e. R goes through edge e, we simply denote it as $e \in R$.

Definition 2.2. *Given a source vertex s and a target vertex t, $\mathbb{R}$ denotes the set of all candidate routes from s to t. The* best route*, denoted by BR, is defined as a discrete random variable with sample space $\mathbb{R}$. Each candidate route $R \in \mathbb{R}$ has a probability $Pr(R)$ of being the best one, and $\sum_{R\in\mathbb{R}} Pr(R) = 1$.*

Definition 2.3. *Given a Route Set RS with the source vertex s and the target vertex t, a* Routing Query *Q is defined as a triple (v_{st}, C, t), where v_{st} is the starting vertex of the query, indicating an intersection, and $C = \{v_1, ..., v_{|C|}\}$ is the set of all successors of v_{st} in $\mathbb{R}$, namely, all possible directions of moving from v_{st} towards t. In the rest of the paper, by default $|C| \geq 2$, i.e. only those vertices with at least two successors are worth querying about.*

Given a route set $\mathbb{R}$, we use $\mathbb{Q}$ to denote the set of all queries in $\mathbb{R}$. For a set of queries $S_k \in \mathbb{R}$ that is selected to ask, suppose that $S_k = \{Q_1, Q_2, \ldots, Q_k\}$ the crowdsourced answer set $A_{S_k} \in C_{S_k}$, where $C_{S_k} = C_{Q_1} \times \ldots \times C_{Q_k}$. The error rate of the crowdsourcing workers is denoted by ε. Table 2 summarizes the notations. In Figure 1 and Table 1, $\mathbb{R} = \{R_1, R_2, R_3\}$, and suppose that $BR = R_4$. Let $k = 2$ and $S_k = \{Q_1, Q_3\}$. As v_1 is the starting vertex of Q_2 and $BR \not\to v_1$, the crowdsourcing worker gives random answer according to the probability mass function over C_2, namely, v_3 with probability 0.5, or otherwise v_4. For the starting vertex v_2 of Q_3, $BR \to v_2$ and $(v_2, v_4) \in BR$, a worker with error rate ε answers v_4 with probability $(1 - \varepsilon)$, or answers t with probability ε.

Similar as in [1], we use the Shannon Entropy, which is a non-parametric measurement that requires no assumption about external factors, to measure the hardness of selecting BR from $\mathbb{R}$.

Definition 2.4. *Given a path set $\mathbb{R} = \{R_1, R_2, ..., R_{|\mathbb{R}|}\}$, the hardness of selecting the best path BR, denoted by $H(BR)$, is defined as the Shannon Entropy of BR,*

$$H(BR) = -\sum_{R\in\mathbb{R}} Pr(R)\log(Pr(R))$$

We formally state the problem definition as follows.

Definition 2.5. *(*Problem definition*) Given a path set $\mathbb{R}$ and a budget B of the number of queries, without exceeding the budget, we aim to design strategies to crowdsource routing queries in order to maximally reduce the selection hardness $H(BR)$.*

3 Basic RQ-based Method

In this section, we present a complete solution to select and crowdsource RQs in order to reduce the selection hardness. First, we use the expected reduction of selection hardness as the metric to evaluate RQs, and derive necessary formulas to enable the computation. Second, we study how to efficiently select the best RQ. Third, we present how to utilize conflicting crowdsourced answers. Lastly, we put these together to develop the framework of the RQ-based method, which reduces the selection hardness using a sequence of RQs.

3.1 RQ Selection Metric

In order to design an effective strategy for selecting RQs, it is essential to define a metric to estimate the importance of RQs before they are answered. Since the final objective is to reduce the selection hardness, we use the *probabilistic expectation* of selection hardness conditioned on individual RQs as the metric.

TABLE 2
Summary of Notations

Notation	Meaning
R or R_i	a candidate route
$R \to v$	R goes through vertex v
$R \not\to v$	R does not go through vertex v
$\mathbb{R}$	Route Set: the set of all candidate routes for a pair of source and target
BR	the best route for a given $\mathbb{R}$
$Q = (v_{st}, C, t)$	a routing query with starting vertex v_{st}, constant target t and direction set C
A_Q	the correct answer of query Q
$\mathbb{Q}$	the set of all queries for a given $\mathbb{R}$
S_k	a subset of $\mathbb{Q}$ containing k queries
A_{S_k}	the k answers of queries in S_k
C_{S_k}	the Cartesian product of direction sets of the queries in S_k: $C_{Q_1} \times \cdots \times C_{Q_k}$
$BQ = (v_{st}, C, t)$	a Binary routing query with $\|C\|$=1
$H(BR)$	the selection hardness among candidate routes
ΔH_Q	the expected reduction of selection hardness by asking the crowd with query Q
ΔH_{S_k}	the expected reduction of selection hardness by asking the crowd all the queries in S_k
ε	the error rate of a crowdsourcing worker
$X \perp Y \| Z$	X is independent of Y given Z, where X, Y, Z are random variables

For an arbitrary routing query $Q :=< v_{st}, D, t >$, let A_Q be the ground truth answer of the Q. Probabilistically, A_Q is a discrete random variable with sample space D. Therefore, the expectation of selection hardness after receiving A_Q, denoted as $\mathbb{E}H(BR|A_Q)$, is that

$$\begin{aligned}
&\mathbb{E}H(BR|A_Q)\\
&= \sum_{v_i\in D} Pr(A_Q = v_i)H(BR = R|A_Q = v_i)\\
&= \sum_{v_i\in D} Pr(A_Q = v_i) \sum_{R_j\in\mathbb{R}} (Pr(BR = R_j|A_Q = v_i)\\
&\qquad \log Pr(BR = R_j|A_Q = v_i))
\end{aligned} \tag{2}$$

There are two parameters used in Equation 25 : $Pr(A_Q = v_i)$ (i.e. the probability that v_i is the correct answer of Q) and $Pr(BR = R_j|A_Q = v_i)$ (i.e. the probability that R_j is the best path, given that v_i is the correct answer of Q). Now we derive formulas to compute these two parameters.

Computation of $\mathbf{Pr(A_Q = v_i)}$: Recall that RQ is a question asking how to move forward starting from v_{st}. Hence, $A_Q = v_i$ indicates $e := (v_{st}, v_i) \in BR$, given BR goes through v_{st}. Then we have

$$\begin{aligned}
Pr(A_Q = v_i) &= Pr(e \in BR|BR \to v_{st})\\
&= \frac{Pr(BR \to v_{st}|e \in BR)Pr(e \in BR)}{Pr(BR \to v_{st})}
\end{aligned}$$

Please note that v_{st} is the head of edge e, so given the condition that e is on the best path (i.e. $e \in BR$), the best path must go through v_{st}, that is, $Pr(BR \to v_{st}|e \in BR) = 1$. Hence, we have

$$Pr(A_Q = v_i) = \frac{Pr(e \in BR)}{Pr(BR \to v_{st})}$$

Following the Law of Total Probability [12], we have $Pr(e \in BR) = \sum_{R\in\mathbb{R}} Pr(e \in BR \cap BR = R) = \sum_{R\in\mathbb{R}\wedge e\in R} Pr(R)$

and $Pr(BR \rightarrow v_{st}) = \sum_{R' \in \mathbb{R}} Pr(R' \rightarrow v_{st} \cap BR = R') = \sum_{R' \in \mathbb{R} \wedge R' \rightarrow v_{st}} Pr(R')$. Finally, we have

$$Pr(A_Q = v_i) = \frac{\sum_{R \in \mathbb{R} \wedge e \in R} Pr(R)}{\sum_{R' \in \mathbb{R} \wedge R' \rightarrow v_{st}} Pr(R')} \tag{3}$$

Equation 3 computes the probability that A_Q taking each element of D, hereby we have the probability mass function (pmf) [12] of A_Q.

Computation of $\mathbf{Pr(BR = R_j | A_Q = v_i)}$: The main difficulty of deriving $Pr(BR = R_j | A_Q = v_i)$ is to determine the correlation between '$BR = R_j$' and '$A_Q = v_i$'. We observe that this correlation is closely related to '$BR \rightarrow v_{st}$', i.e. whether the best path goes through the starting point of RQ. Therefore, we expand $Pr(BR = R_j | A_Q = v_i)$ with the Law of Total Probability as follows:

$$\begin{aligned} &Pr(BR = R_j | A_Q = v_i) = \\ &Pr(BR \rightarrow v_{st}) Pr(BR = R_j | A_Q = v_i, BR \rightarrow v_{st}) \\ &+ (1 - Pr(BR \rightarrow v_{st})) Pr(BR = R_j | A_Q = v_i, BR \nrightarrow v_{st}) \end{aligned} \tag{4}$$

where we have $Pr(BR \rightarrow v_{st}) = \sum_{R \in \mathbb{R} \wedge R \rightarrow v_{st}} Pr(R)$.

We derive $Pr(BR = R_j | A_Q = v_i, BR \rightarrow v_{st})$ and $Pr(BR = R_j | A_Q = v_i, BR \nrightarrow v_{st})$ by respectively analyzing two exclusive conditions - $BR \rightarrow v_{st}$ and $BR \nrightarrow v_{st}$.

Condition $BR \rightarrow v_{st}$: First, we analyze the situation that v_{st} is on the best path BR. For each $v_i \in D$, if edge $e := (v_{st}, v_i)$ is on the best path, then v_i must be the best direction going from v_{st} to t, i.e. the ground truth answer A_Q should be v_i. Therefore, we have $e \in BR \Rightarrow A_Q = v_i$.

Similarly, if $A_Q = v_i$ and $BR \rightarrow v_{st}$, we can ensure that $e \in BR$. So $(A_Q = v_i \wedge BR \rightarrow v_{st}) \Rightarrow e \in BR$. Overall, we conclude that $e \in BR$ if and only if $(A_Q = v_i \wedge BR \rightarrow v_{st})$, i.e.

$$(A_Q = v_i \wedge BR \rightarrow v_{st}) \Leftrightarrow e := (v_{st}, v_i) \in BR \tag{5}$$

Therefore, we have

$$\begin{aligned} &Pr(BR = R_j | A_Q = v_i, BR \rightarrow v_{st}) \\ &= Pr(BR = R_j | e := (v_{st}, v_i) \in BR) \\ &= \frac{Pr(e \in BR | BR = R_j) Pr(R_j)}{Pr(e \in BR)} \\ &= \begin{cases} 0 & e \notin R_j \\ \dfrac{Pr(R_j)}{\sum_{R \in \mathbb{R} \wedge e \in R} Pr(R)} & otherwise \end{cases} \end{aligned} \tag{6}$$

Condition $BR \nrightarrow v_{st}$: Second, we consider the condition when the best path does not go through v_{st}. Note each vertex in D indicates a path that is possibly the best direction going from v_{st} to t, and we are interested in the best path from the source vertex s to t. Therefore, the answer to RQ gives us useful information only if v_{st} is known to be on the best path. In other words, if v_{st} is not on BR, how to move from v_{st} towards the target does not affect the distribution of BR, since one will not even go to v_{st} in the first place. Probabilistically, BR and A_Q are independent given that 'BR does not go through v_{st}'. Formally, we have

$$A_Q \perp BR | BR \nrightarrow v_{st} \tag{7}$$

where we adopt $\perp$ to denote the operator indicating two random variables are conditionally independent [13].

From Formula 7, we have

$$\begin{aligned} &Pr(BR = R_j | A_Q = v_i, BR \nrightarrow v_{st}) \\ &= \begin{cases} 0 & R_j \rightarrow v_{st} \\ \dfrac{Pr(R_j)}{\sum_{R \in \mathbb{R} \wedge R \nrightarrow v_{st}} Pr(R)} & otherwise \end{cases} \end{aligned} \tag{8}$$

Then, equipped with Equation 8 and 6, we have completed the derivation of parameters used in Equation 30.

Finally, by substituting Equation 30 and 3 into Equation 25, we can compute the expectation of selection hardness for asking each RQ.

3.2 Choosing the best RQ

A naive approach of selecting the best RQ is to traverse all the RQs. However, this is very costly since the computation w.r.t. RQ requires accessing all the paths in $\mathbb{R}$. When the number of candidate paths is large, the computational cost will be higher. Fortunately, we found that the expected reduction of selection hardness for $RQ :=< v_{st}, D, t >$ is only related to the paths going through v_{st}. We conclude this discovery with the following theorem.

Theorem 3.1. *For a given path set $\mathbb{R}$ and a given $RQ :=< v_{st}, D, t >$, let ΔH_{RQ} be the expected reduction of selection hardness by asking RQ to the crowd, we have that ΔH_{RQ} is equivalent to 'the entropy of RQ' multiplying 'the probability of the best path going through v_{st}', i.e.*

$$\begin{aligned} &\Delta H_{RQ} = H(BR) - \mathbb{E}H(BR | A_Q) \\ &= -(\sum_{R \rightarrow v_{st}} Pr(R)) \sum_{v_i \in D} Pr(A_Q = v_i) \log Pr(A_Q = v_i) \end{aligned} \tag{9}$$

Proof. Please see the appendix in [1]. □

Theorem 3.1 reflects two factors influencing the importance of a RQ - 'the entropy of the RQ' and 'the probability of the best path going through v_{st}'. Intuitively, the former indicates the amount of information gain by asking this question, so the higher the entropy, the more important the question; the latter indicates the structural position of the question, representing how useful the information gain is for determining the best path. It worth noticing that, the common practice 'asking the most uncertain question' does NOT apply in our problem, as shown in the following example.

3.3 Utilization of Conflicting Crowdsourced Answers

The essential objective of crowdsourcing is to use the answers to adjust the probability distribution of the best path. However, crowdsourced answers may be mistaken or subjective. As a result, different workers may return conflicting answers for the same question. To handle this issue, we must allow each crowdsourced answer to be wrong with a probability. This probability can be estimated by the error rate of the worker. For a $RQ :=< v_{st}, D, t >$, let v_C be the result returned by a crowdsourcing worker with error rate ϵ.

Now we present how to use crowdsourced answers to adjust the probability of each candidate path R_i. That is to derive the formula to compute $Pr(BR = R_i | v_C \; returned \; by \; the \; crowd)$. To do this, we need to consider three exclusive cases: 1) $R_i \nrightarrow v_{st}$,

i.e. R_i does not go through v_{st}, so R_i is not affected by the answer of the RQ; 2) $(v_{st}, v_C) \in R_i$, i.e. R_i goes through v_{st} and v_C, which indicates that the crowdsourced answer is supportive for R_i; 3) $R_i \rightarrow v_{st} \wedge (v_{st}, v_C) \notin R_i$, i.e. R_i goes through v_{st} but not v_C, which indicates that the crowdsourced answer is against for R_i.

We list the details for all three cases as follows.

Case 1) $\underline{BR \nrightarrow v_{st}}$: According to Equation 7, the answer of RQ is independent of BR given $BR \nrightarrow v_{st}$, so we have $Pr(BR = R_i|v_C \ returned \ by \ the \ crowd) = Pr(BR = R_i) = Pr(R_i)$;

Case 2) $\underline{(v_{st}, v_C) \in R_i}$: According to Bayes' theorem

$$\begin{aligned} &Pr(BR = R_i|v_C \ returned \ by \ the \ crowd) \\ &= \frac{Pr(R_i)Pr(v_C \ returned \ by \ the \ crowd|BR = R_i)}{Pr(v_C \ returned \ by \ the \ crowd)} \end{aligned} \tag{10}$$

We have

$$\begin{aligned} &Pr(v_C \ returned \ by \ the \ crowd) = \\ &\quad Pr(A_Q = v_C)(1-\epsilon) + (1 - Pr(A_Q = v_C))\epsilon \\ &Pr(v_C \ returned \ by \ the \ crowd|BR = R_i) = \\ &\quad Pr(crowd \ answers \ the \ RQ \ correctly) = 1 - \epsilon \end{aligned} \tag{11}$$

So, in case of $(v_{st}, v_C) \in BR$, we have

$$\begin{aligned} &Pr(BR = R_i|v_C \ returned \ by \ the \ crowd) = \\ &\quad \frac{Pr(R_i)(1-\epsilon)}{Pr(A_Q = v_C)(1-\epsilon) + (1 - Pr(A_Q = v_C))\epsilon} \end{aligned} \tag{12}$$

where $Pr(A_Q = v_C)$ is derived in Equation 3.

Case 3) $\underline{R_i \rightarrow v_{st} \wedge (v_{st}, v_C) \notin R_i}$: Analogous to case 2), since $(v_{st}, v_C) \notin R_i$ and $(v_{st}, v_C) \notin BR$, we know that v_C is an incorrect answer of RQ conditioning on $BR = R_i$, i.e. the crowd answers RQ correctly. So,

$$\begin{aligned} &Pr(v_C \ returned \ by \ the \ crowd|BR = R_i) = \\ &\quad Pr(crowd \ answers \ the \ RQ \ incorrectly) = \epsilon \end{aligned}$$

Then we have

$$\begin{aligned} &Pr(BR = R_i|v_C \ returned \ by \ the \ crowd) = \\ &\quad \frac{Pr(R_i)\epsilon}{Pr(A_Q = v_C)(1-\epsilon) + (1 - Pr(A_Q = v_C))\epsilon} \end{aligned} \tag{13}$$

To conclude the above analysis, we achieve the following close-form formula for using crowdsourced answer to adjust the probability distribution of the best path:

$$\begin{aligned} &Pr(BR = R_i|v_C \ returned \ by \ the \ crowd) = \\ &\begin{cases} Pr(R_i) & R_i \nrightarrow v_{st} \\ \dfrac{Pr(R_i)(1-\epsilon)}{Pr(A_Q = v_C)(1-\epsilon) + (1 - Pr(A_Q = v_C))\epsilon} & (v_{st}, v_C) \in R_i \\ \dfrac{Pr(R_i)\epsilon}{Pr(A_Q = v_C)(1-\epsilon) + (1 - Pr(A_Q = v_C))\epsilon} & otherwise \end{cases} \end{aligned} \tag{14}$$

Actually, by considering R_i as a binary random variable, $Pr(R_i|v_C)$ is the probability of $BR = R_i$ conditioning on event "v_C is answered by the crowd". Therefore, when more answers are received, the probability of $BR = R_i$ would be recursively adjusted by Equation 14, conditioning on each received answer and error rate of the corresponding worker. Please note that different workers may have different error rates. Furthermore, after the probabilities of candidate paths are adjusted by one answer, the probability distribution of each A_Q is also updated by recomputing Equation 3. So, when the next answer is received, the adjustment is conducted with the updated probability of each R_i.

It is easy to perform the algebraic manipulations to show that, for any two answers v_C and v'_C, we have

$$\begin{aligned} &Pr(BR = R_i|v_C \ returned \ by \ the \ crowd, \\ &\qquad and \ then \ v'_C \ returned \ by \ the \ crowd) \\ =&Pr(BR = R_i|v'_C \ returned \ by \ the \ crowd, \\ &\qquad and \ then \ v_C \ returned \ by \ the \ crowd) \\ =&Pr(BR = R_i|v_C \ and \ v'_C \ are \ returned \ by \ the \ crowd) \end{aligned} \tag{15}$$

The above equation resolves three issues of concern. The first is the sequence of answers received from workers. Equation 15 indicates that, given two crowdsourced answers, the final result of $\mathbb{R}$ is independent of the sequence of the answers being utilized. In other words, the final result of R_i is the probability of $BR = R_i$ conditioning on the event that "both answers are received". The second issue is that, the same RQ may be answered differently by multiple workers. Particularly, in Equation 15, v'_C and v_C may be conflicting answers for the same RQ from two workers. In this case, by recursively executing Equation 14 twice, the effect of v_C and v'_C are gracefully aggregated based on different error rates of workers. Third, after the utilization of crowdsourced answers, the sum of probabilities of all candidate paths should always be one. As follows, we show how to use a crowdsourced answer with a running example.

3.4 The Framework of RQ-based Method

In this subsection, we provide the complete framework of our proposed RQ-based method. Algorithm 1 illustrates this framework, which consists of two iterative phases:

- *Choosing the best RQ* - select the best RQ based on the current probabilities of candidate paths, and post it to the crowd;
- *Utilization of Conflicting Crowdsourced Answers* - adjust the probabilities of all candidate paths according to the crowdsourced answers.

In Algorithm 1, these two phases are iteratively performed B times due to the given budget. In each iteration, we firstly calculate the expected reduction of selection hardness, ΔH_{RQ}, for each RQ via Theorem 3.1. Then, the one with maximum ΔH_{RQ} is selected and published to the crowd. Second, we receive the answer v_C, and adjust the probabilities of all candidate paths through Formula 14, hereby reduce the selection hardness.

Input: A path set $\mathbb{R}$, U_{RQ}, a total budget B
while $B \neq 0$ **do**
  for *each* $RQ_i \in U_{RQ}$ **do**
    calculate ΔH_{RQ_i} via Theorem 3.1;
  end
  $RQ_{max} \leftarrow \text{argmax}_{RQ_i \in U_{RQ}} \Delta H_{RQ_i}$;
  Ask RQ_{max} to crowd and receive the corresponding answer v_C;
  for *each* $R_j \in \mathbb{R}$ **do**
    $Pr(R_j) \leftarrow Pr(BR = R_j|v_C)$ via Formula 14);
  end
  $B \leftarrow B - 1$;
end

Algorithm 1: The Framework of RQ-based Method

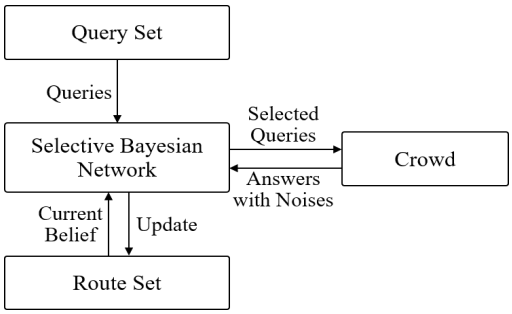


Fig. 2. Crowd-aided route selection with Selective Bayesian Network

3.5 BRQ: A Different Question Type

In the RQ-based method, each RQ is a multiple choice question. For a high-degree vertex, it would be constructed into a multiple choice with too many options to be answered by a crowdsourcing worker. As suggested in [14], [15], crowds are good at tasks broken down into small pieces (often with a YES/NO answer). Motivated by this, we consider an extension that uses an easier type of questions, namely BRQ, as defined by the following Definition 3.1.

Definition 3.1 (Binary Routing Query (BRQ))**.** *For a given* $RQ :=< v_{st}, D = \{v_0, ..., v_{|D|}\}, t >$*, a Binary Routing Query BRQ is triple* $< v_{st}, v_d, t >$*, where* $v_d \in D$*.*

From the perspective of a crowdsourcing worker, a BRQ is a question of the form "From v_{st} to t, should I go to the direction of v_d?" The bottom part of Table 1 lists all the BRQs for the $\mathbb{R}$. It is obvious that each RQ can be easily decomposed in to $|D|$ distinct BRQs. As a new type of questions, BRQs can easily fit into the Algorithm 1. Analogous to the RQ-based method, we also focus on studying how to select the best BRQ in this extension.

Finding the best BRQ: As shown in Theorem 3.1, the significance of an RQ is determined by its information gain and topological position. For BRQs, we reach similar result, as shown with the following theorem.

Theorem 3.2. *For a given path set* $\mathbb{R}$ *and a given* $BQ :=< v_{st}, v_d, t >$*, let* ΔH_{BQ} *be the expected reduction of selection hardness by asking the BRQ to the crowd, we find that* ΔH_{BQ} *is equivalent to 'the entropy of the BRQ' multiplying 'the probability of the best path going through* v_{st}*', that is*

$$\begin{aligned}\Delta H_{BRQ} &= H(BR) - \mathbb{E}H(BR|A_{BQ}) \\ &= -(\sum_{R \to v_{st}} Pr(R))[Pr(A_{BQ} = v_d) \log Pr(A_{BQ} = v_d) \\ &+ (1 - Pr(A_{BQ} = v_d)) \log (1 - Pr(A_{BQ} = v_d))]\end{aligned} \tag{16}$$

Proof. Please see appendix in [1]. □

4 Select the Best Set of RQs with Selective Bayesian Network

In this section, we present the crowd-aided route selection with Selective Bayesian Network. Figure 2 shows the architecture of the system. As a centric part of the framework, the Selective Bayesian Network repeatedly selects the best set S_k of k routing queries from the query set $\mathbb{Q}$, according to current probabilities of the routes in $\mathbb{R}$. Each selected S_k is given to the crowd, and the k answers with noises are then collected from the crowd to update the probabilities and the network. When the budget runs out, the system returns the most likely best route.

4.1 RQ Set Selection Metric

Before we formally introduce our approach, we first give the metric. The final objective is to reduce the selection hardness, so we still use the probabilistic expectation of selection hardness conditioned on given set of routing queries as our metric. If the queries are given to the crowd one by one, for a routing query $Q = (v_{st}, C, t)$, suppose that the ground truth answer for Q is A_Q. The answer A_Q is actually a discrete random variable with sample space C. The expectation of selection hardness after receiving A_Q, denoted as $\mathbb{E}H(BR|A_Q)$, is that

$$\begin{aligned}&\mathbb{E}H(BR|A_Q) \\ &= \sum_{v \in D} Pr(A_Q = v) H(BR|A_Q = v) \\ &= \sum_{v \in D} Pr(A_Q = v) \sum_{R \in \mathbb{R}} (Pr(BR = R|A_Q = v) \\ &\qquad \log Pr(BR = R|A_Q = v))\end{aligned} \tag{17}$$

If the problem is to select the best single query Q from $\mathbb{Q}$, such that the expected selection hardness is maximally reduced, the expected reduction of the selection hardness ΔH_{S_k} is

$$\begin{aligned}\Delta H_Q &= H(BR) - \mathbb{E}H(BR|A_Q) \\ &= -\sum_{R \in \mathbb{R}} Pr(R) \log(Pr(R)) - \sum_{v \in D} Pr(A_Q = v) H(BR|A_Q = v)\end{aligned} \tag{18}$$

and we have the optimization problem $\text{argmax}_{Q \in \mathbb{Q}} \Delta H_Q$. As a main conclusion of our work in [1], ΔH_Q can be characterized by two features, namely the entropy of Q, and the probability of the best path going through v_{st}, where v_{st} is the starting vertex of Q.

$$\begin{aligned}\Delta H_Q &= H(BR) - \mathbb{E}H(BR|A_Q) \\ &= -(\sum_{R \to v_{st}} Pr(R)) \sum_{v \in C} Pr(A_Q = v) \log Pr(A_Q = v)\end{aligned} \tag{19}$$

The former feature above indicates the information gain by asking the query Q. The higher entropy the query Q has, the more information the answer A_Q gives. The latter feature represents the structural importance of the query, indicating how important the information gain is for determining the best route. The two features well captures the essence of crowd-aided route selection with a single best routing query - to select the query with high uncertainty and high spacial importance. However, in a crowdsourcing environment, we usually need to ask multiple questions each round to reduce latency. If we consider giving k queries per round to the crowd, the problem becomes to select the best combination of k routing queries, S_k, from $\mathbb{Q}$, such that the expected selection hardness is maximally reduced, i.e. the optimization problem

$$\underset{S_k \subseteq \mathbb{Q}, |S_k| \leq k}{\text{argmax}} \quad \Delta H_{S_k} \tag{20}$$

where

$$\Delta H_{S_k} = H(BR) - \mathbb{E}H(BR|A_{S_k}) \tag{21}$$

Now the key problem is how to precisely estimate and efficiently compute the expected reduction of selection hardness

after receiving a set of crowdsourced answers. Generally, selecting S_k from $\mathbb{Q}$ that maximizes ΔH_{S_k} is NP-Hard, but we can approximate it. From the perspective of information theory [16], ΔH_{S_k} can be considered as the mutual information between BR and A_{S_k}. The existing work did not give an explicit solution for ΔH_{S_k} above, and the metric was roughly estimated by sampling and the following formula,

$$\Delta H_{S_k} \approx \Delta \hat{H}_{S_k} = \sum_{BR, A_{S_k}} fq(BR, A_{S_k}) \log \frac{fq(BR, A_{S_k})}{fq(BR) \cdot fq(A_{S_k})} \tag{22}$$

where $fq(BR) = \sum_{A_{S_k}} fq(BR, A_{S_k})$ and $fq(A_{S_k}) = \sum_{BR} fq(BR, A_{S_k})$. To find a better approximation and finally solve the problem, we introduce the Selective Bayesian Network.

4.2 Selective Bayesian Network

The baseline sampling-based method in [1] uses two relaxations, one of which calculates approximate solution by greedy strategy, and the other one uses random sampling to estimate the probabilities. If the sampling rate is high enough, the algorithm will give acceptable result in the sense of precision. However, if the sizes of the route set $\mathbb{R}$ and corresponding query set $\mathbb{Q}$ are relatively large, the performance of the sampling-based algorithm dramatically declines due to sampling insufficiency. In the following, we give an alternative approach based on Selective Bayesian Network.

Given a set of candidate routes $\mathbb{R}$, we can construct a corresponding Selective Bayesian Network for the computation of selection hardness reduction ΔH_{S_k}. It is obvious that any route that contains a directed loop is not the best route. For instance, for a route $R = v_1 v_2 \dots v_t$ where $v_i = v_j$ for some $i, j \in [1, t], i \neq j$, R contains a loop $v_i v_{i+1} \dots v_j$. Obviously, R is dominated by another route $R' = v_1 v_2 \dots v_i v_{j+1} \dots v_t$. For any given route $R \in \mathbb{R}$, checking the existence of a loop is an linear-time task, and it's trivial to improve the paths with loops by simply eliminating all loops from them. In the following, we assume that the graph constructed by $\mathbb{R}$ is a directed acyclic graph (DAG), and call $\mathbb{R}$ a directed acyclic graph for convenience.

Definition 4.1. *Given a set of routing queries $\mathbb{R}$ that is acyclic, its **Selective Bayesian Network**, denoted by $\mathcal{N}(\mathbb{R})$, is a DAG that consists of:*

- Nodes *There is a node v_i in $\mathcal{N}(\mathbb{R})$ for each vertex v_i in $\mathbb{R}$, annotated with the probability of $Pr(v_i) = Pr(BR \to v_i)$, where BR is the best route. In the rest of the paper, we also use v_i as a random variable standing for $BR \to v_i$, and $\neg v_i$ for $BR \not\to v_i$, with probabilities $Pr(v_i)$ and $1 - Pr(v_i)$ respectively. Each node v_i (except the source node s) is labeled by a conditional probability table $PT(v_i)$. The computation of the probability table PT will be introduced later.*
- Edges *There is a directed edge $(v_i, v_j) \in \mathcal{N}(\mathbb{R})$ for each directed edge $(v_i, v_j) \in \mathbb{R}$. In the following, we call v_i a parent of v_j, and v_j a child of v_i. Notice that it is possible for a node to have multiple parents and multiple children.*

Definition 4.2. *For a route set $\mathbb{R}$, its Selective Bayesian Network $\mathcal{N}(\mathbb{R})$, and a node $v \in \mathcal{N}(\mathbb{R})$, we call a set of literals o a **priori observation** of v if o is in following form:*

$$o = \{l_i \mid (v_i, v) \in \mathcal{N}(\mathbb{R}), l_i \in \{v_i, \neg v_i\}\},$$

and define the positive part of the observation o as $o^+ = \{v_i \mid v_i \in o\}$, and the negative part of the observation o as $o^- = \{\neg v_i \mid \neg v_i \in o\}$.

We denote the set of all priori observations of v by $O(v)$. In Figure 1, for instance,

- $o_1 = \{\neg v_1, v_3, \neg v_4\}$ is a priori observation of vertex t, i.e. $o_1 \in O(t)$, with the positive part $o_1^+ = \{v_3\}$ and the negative part $o_1^- = \{\neg v_1, \neg v_4\}$,
- $o_2 = \{s, v_2\}$ is a priori observation of vertex v_3, i.e. $o_2 \in O(v_3)$, with the positive part $o_2^+ = \{s, v_2\}$ and negative part $o_2^- = \emptyset$.

Definition 4.3. *For a node v in $\mathcal{N}(\mathbb{R})$, an observation o of v, a set of answers A_{S_k} for routing query set $S_k \subset \mathbb{Q}$, and a route $R \in \mathbb{R}$, we say R **submits to** o, denoted as $R \models o$, if*

$$(\forall v_x \in o^+ . R \to v_x) \wedge (\forall v_y \in o^- . R \not\to v_y),$$

*We say R **submits to** A_{S_k}, also denoted as $R \models A_{S_k}$, if for each query $Q = (v_{st}, C, t)$ in S_k, suppose v_{ans} is the answer of Q given by A_{S_k}, we have $R \to v_{st} \supset R \to v_{ans}$.*

The "$\supset$" above is the implication in classic logic, indicating if the route R goes through v_{st}, then it has to goes through v_{ans}. Now we give the computation of the probability table, denoted by PT, of $\mathcal{N}(\mathbb{R})$.

Lemma 4.1 (Computation of probability table PT)**.** *For each node $v_i \in \mathcal{N}(\mathbb{R})$,*

1) *Initialization: for each observation $o \in O(v_i)$, set*

$$Pr(v_i|o) = \frac{\sum_{R \in \mathbb{R} \wedge R \to v_i \wedge R \models o} Pr(R)}{\sum_{R \in \mathbb{R} \wedge R \models o} Pr(R)} \tag{23}$$
$$Pr(\neg v_i|o) = 1 - Pr(v_i|o)$$

2) *Completion: for those observation $o \in O(v_i)$ such that $\nexists R \in \mathbb{R}. R \models o$, set $Pr(v_i|o) = 0$ and $Pr(\neg v_i|o) = 0$.*

Note that after the initialization step in Lemma 4.1, the probability table PT is not complete. For instance, Figure 3 shows the the Selective Bayesian Network for the example in Figure 1 with the routing queries in Table 1 (with negative conditional probabilities omitted). In the figure, for node v_3, the conditional probability $Pr(v_3|sv_1v_2)$ is still undefined after initialization, since there is no path that submits to the observation of v_3: $o = \{s, v_1, v_2\}$. So, in the completion step, we complete the probability table $PT(v_3)$ by simply setting $Pr(v_3|s, v_1, v_2) = 0$ and $Pr(\neg v_3|s, v_1, v_2) = 1$, and other necessary conditional probabilities.

We give some more examples of the supportive probabilities:

- For a node v_i in $\mathbb{R}$, there is only one outgoing edge from v_i. Namely, in the routing query $RQ =< v_i, D_i, t >$, D_i is a singleton, say $D_i = \{v_{out}\}$, we have $Pr(A_Q = v_{out}) = 1$, though such a query is usually omitted in U_{RQ}.
- For a node v_i in $\mathbb{R}$, there is only one incoming edge towards v_i, say v_{st}. The corresponding supportive probabilities will be,

$$Pr(v_i|\{v_{st}\}) = Pr(A_{<v_{st}, D, t>} = v_i)$$
$$Pr(v_i|\{\neg v_{st}\}) = 0$$
$$Pr(\neg v_i|\{v_{st}\}) = 1 - Pr(A_{<v_{st}, D, t>} = v_i)$$
$$Pr(\neg v_i|\{\neg v_{st}\}) = 1$$

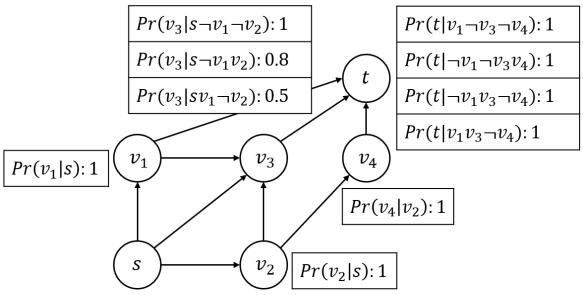


Fig. 3. Initialization of the Selective Bayesian Network

TABLE 3
Supportive probabilities

Example supportive probabilities			
$Pr(v_1\|\{s\})$	0.1	$Pr(\neg v_1\|\{s\})$	0.9
$Pr(v_1\|\{\neg s\})$	0	$Pr(\neg v_1\|\{\neg s\})$	1
$Pr(v_2\|\{s\})$	0.1	$Pr(\neg v_2\|\{s\})$	0.9
$Pr(v_2\|\{\neg s\})$	0	$Pr(\neg v_2\|\{\neg s\})$	1
$Pr(v_3\|\{s\})$	0.8	$Pr(\neg v_3\|\{s\})$	0.2
$Pr(v_3\|\{\neg s\})$	0	$Pr(\neg v_3\|\{\neg s\})$	1
$Pr(v_4\|\{v_2, v_3\})$	0	$Pr(\neg v_4\|\{v_2, v_3\})$	1
$Pr(v_4\|\{v_2, \neg v_3\})$	1	$Pr(\neg v_4\|\{v_2, \neg v_3\})$	0
$Pr(v_4\|\{\neg v_2, v_3\})$	0.5	$Pr(\neg v_4\|\{\neg v_2, v_3\})$	0.5
$Pr(v_4\|\{\neg v_2, \neg v_3\})$	0	$Pr(\neg v_4\|\{\neg v_2, \neg v_3\})$	1
$Pr(t\|\{v_1, v_3, v_4\})$	0	$Pr(\neg t\|\{v_1, v_3, v_4\})$	1
$Pr(t\|\{v_1, v_3, \neg v_4\})$	0	$Pr(\neg t\|\{v_1, v_3, \neg v_4\})$	1
$Pr(t\|\{v_1, \neg v_3, v_4\})$	0	$Pr(\neg t\|\{v_1, \neg v_3, v_4\})$	1
$Pr(t\|\{v_1, \neg v_3, \neg v_4\})$	1	$Pr(\neg t\|\{v_1, \neg v_3, \neg v_4\})$	0
$Pr(t\|\{\neg v_1, v_3, v_4\})$	1	$Pr(\neg t\|\{\neg v_1, v_3, v_4\})$	0
$Pr(t\|\{\neg v_1, v_3, \neg v_4\})$	1	$Pr(\neg t\|\{\neg v_1, v_3, \neg v_4\})$	0
$Pr(t\|\{\neg v_1, \neg v_3, v_4\})$	1	$Pr(\neg t\|\{\neg v_1, \neg v_3, v_4\})$	0
$Pr(t\|\{\neg v_1, \neg v_3, \neg v_4\})$	0	$Pr(\neg t\|\{\neg v_1, \neg v_3, \neg v_4\})$	1
The set of all predecessors $pre(v)$ for each $v \in \mathbb{R}$			
$pre(s) : \emptyset$	$pre(v_1) : \{s\}$	$pre(v_2) : \{s\}$	
$pre(v_3) : \{s\}$	$pre(v_4) : \{v_2, v_3\}$	$pre(t) : \{v_1, v_3, v_4\}$	

- For the example in Table 1, the corresponding conditional probabilities is shown in Table 3.

4.3 Select the Best Routing Query Set

A straightforward heuristic solution to select the best query set is to select the top-k questions which have the highest expected reduction of selection hardness in Equation 19. However, such a solution neglects the correlation among the queries in S_k. The correlation can be estimated by sampling, but the precision depends on the sample qualification and quantity. With the Selective Bayesian Network, we have the spacial casual information about the nodes, so we can make use of the spacial causalities to compute the expected uncertainty reduction.

Let S_k be a set of k routing queries, $S_k = \{Q_1, \ldots, Q_k\}$, where each query is in the form $Q_i = (v^i_{st}, C_i, t)$, Let A_{S_k} be the k ground-truth answers of the queries in S_k, which is actually a discrete random variable with sample space $C_1 \times \ldots \times C_k$. The probability of A_{S_k} is is actually the joint probability

$$Pr(A_{S_k} = v_{S_k}) = Pr(A_{Q_1} = v^1_{S_k}, \ldots, A_{Q_k} = v^k_{S_k}) \quad (24)$$

where the vector $v_{S_k} \in C_1 \times \ldots \times C_k$ and $v^i_{S_k}$ is the i-th coordinate of v_{S_k}, which is the ground-truth answer for Q_i. Therefore, the expectation of selection hardness after receiving A_{S_k} is

$$\begin{aligned}
&\mathbb{E}H(BR|A_{S_k}) \\
&= \sum_{v_{S_k} \in C_1 \times \ldots \times C_k} Pr(A_{S_k} = v_{S_k}) H(BR = R|A_{S_k} = v_{S_k}) \\
&= \sum_{v_{S_k} \in C_1 \times \ldots \times C_k} Pr(A_{S_k} = v_{S_k}) \sum_{R \in RS} (Pr(BR = R|A_{S_k} = v_{S_k}) \\
&\qquad \log Pr(BR = R|A_{S_k} = v_{S_k})) \\
&= \sum_{v_{S_k} \in C_1 \times \ldots \times C_k} Pr(A_{Q_1} = v^1_{S_k}, \ldots, A_{Q_k} = v^k_{S_k}) \\
&\qquad \sum_{R \in RS} (Pr(BR = R|A_{Q_1} = v^1_{S_k}, \ldots, A_{Q_k} = v^k_{S_k}) \\
&\qquad\qquad \log Pr(BR = R|A_{Q_1} = v^1_{S_k}, \ldots, A_{Q_k} = v^k_{S_k})
\end{aligned} \quad (25)$$

As stated previously, ΔH_{S_k} can be regarded as the mutual information between BR and A_{S_k}. Actually, for a set of queries $S_k = \{Q_1, Q_2, \ldots, Q_k\}$, some of the queries are independent with BR, and with other queries, given the ground truth and the spacial information. Suppose that $Q_i = (v^i_{st}, C^i, t)$, if exists some $1 \le i, j \le k$ such that $BR \rightarrow v^i_{st}$ and $BR \not\rightarrow v^j_{st}$, then $Pr(A_{Q_j})$ is conditionally independent with $Pr(BR)$ and $Pr(A_{Q_i})$, i.e.

$$\begin{aligned}
&\forall v_p \in C^i, \forall v_q \in C^j \\
&Pr(A_{Q_i}, A_{Q_j}|BR \rightarrow v^i_{st}, BR \not\rightarrow v^j_{st}) \\
&= Pr(A_{Q_i}|BR \rightarrow v^i_{st}, BR \not\rightarrow v^j_{st}) \\
&\qquad\qquad \cdot Pr(A_{Q_j}|BR \rightarrow v^i_{st}, BR \not\rightarrow v^j_{st}) \\
&= Pr(A_{Q_i}|BR \rightarrow v^i_{st}, BR \not\rightarrow v^j_{st}) \cdot Pr(A_{Q_j})
\end{aligned} \quad (26)$$

Now, for a candidate route set $\mathbb{R}$, we can finally calculate $\Delta(H_{S_k})$ for a given set S_k of queries, by using the Selective Bayesian network $\mathcal{N}(\mathbb{R})$ and the following theorem.

Theorem 4.2. *For a candidate route set $\mathbb{R}$, its Selective Bayesian Network $\mathcal{N}(\mathbb{R})$ and a given query set S_k, the expected reduction of selection hardness ΔH_{S_k} is*

$$\Delta H_{S_k} = -\sum_{S^+_k(R) \neq \emptyset} Pr(R) \log Pr(v^1_R, v^2_R, \ldots, v^{|S^+_k(R)|}_R) \quad (27)$$

where $S^+_k(R) = \{Q = (v_{st}, C, t) \mid Q \in S_k, R \rightarrow v_{st}\}$ and $v^i_R = R \cap C_i$ by supposing that $S^+_k(R) = \{Q_1, Q_2, \ldots, Q_{|S^+_k(R)|}\}$ and $Q_i = (v_{st}, C_i, t)$.

Proof. Please see in appendix in supplemental file. □

The joint probability $Pr(v^1_R, v^2_R, \ldots, v^{|S^+_k(R)|}_R)$ in Theorem 4.2 can be calculated by reasoning on $\mathcal{N}(\mathbb{R})$:

$$Pr(v^1_R, v^2_R, \ldots, v^{|S^+_k(R)|}_R) = \prod_{i=1}^{|S^+_k(R)|} Pr(v^i_R|parent(v^i_R)) \quad (28)$$

where $parent(v^i_R) = \{v \mid (v, v^i_R) \in \mathcal{N}(\mathbb{R})\}$, namely, the set of all parents of node v^i_R in the network $\mathcal{N}(\mathbb{R})$.

Thus, instead of estimating ΔH_{S_k} by sampling, the Selective Bayesian Network $\mathcal{N}(\mathbb{R})$ is capable to select the best S_k with Equations 27 and 28. In practice, however, exploring all possible k combinations and reasoning on them is quite expensive. So in our

algorithm, we also use two relaxations to improve the efficiency, which will be discussed in Section 4.4.

4.4 The Framework of the Algorithm

The best routing query set selected by the Selective Bayesian Network is given to the crowd. The probability of the ground-truth answer of a routing query Q is computed by

$$\begin{aligned} &Pr(A_Q = v) = Pr((v_{st}, v) \in BR | BR \to v_{st}) \\ =&\frac{Pr(BR \to v_{st} | (v_{st}, v) \in BR) Pr((v_{st}, v) \in BR)}{Pr(BR \to v_{st})} \\ =&\frac{\sum_{R \in \mathbb{R} \wedge (v_{st}, v) \in R} Pr(R)}{\sum_{R' \in \mathbb{R} \wedge R' \to v_{st}} Pr(R')} \end{aligned} \tag{29}$$

The above equation can be used to compute the pmf for a given set of route $\mathbb{R}$. For instance, in Figure 1, for the routing query Q_3, we have $pmf(Q_3, v_3) = Pr(A_{Q_3} = v_3) = Pr(R_3)/(Pr(R_3) + Pr(R_4)) = 0.4/(0.4 + 0.1) = 0.8$.

With the answers A_{S_k} collected from the crowd, the probabilities of the routes in $\mathbb{R}$ can be updated. According to the conclusion in [1], the final result of the best route distribution is independent of the sequence of the answers being utilized. So the answers in A_{S_k} can be used to update the distribution one by one. With each answer $A_Q = v$ in A_{S_k}, suppose that $Q = (v_{st}, C, t)$, the probability of $R \in \mathbb{R}$ being the best route is updated by following equation

$$\begin{aligned} &Pr(BR = R | A_Q = v) = \\ &Pr(BR \to v_{st}) Pr(BR = R | A_Q = v, BR \to v_{st}) \\ &+ (1 - Pr(BR \to v_{st})) Pr(BR = R | A_Q = v, BR \not\to v_{st}) \end{aligned} \tag{30}$$

where $Pr(BR \to v_{st}) = \sum_{R \in \mathbb{R} \wedge R \to v_{st}} Pr(R)$. In practice, we reasonably assume that the crowd workers do not always give correct answers, so let ε be the error rate of the crowd. The utilization needs to be replaced by

$$Pr(BR = R | v \ returned \ by \ the \ crowd) = \begin{cases} Pr(R) & R \not\to v_{st} \\ \dfrac{Pr(R)(1-\varepsilon)}{Pr(A_Q = v)(1-\varepsilon) + (1 - Pr(A_Q = v))\varepsilon} & (v_{st}, v) \in R \\ \dfrac{Pr(R)\varepsilon}{Pr(A_Q = v)(1-\varepsilon) + (1 - Pr(A_Q = v))\varepsilon} & otherwise \end{cases} \tag{31}$$

We formally introduce our algorithm. The framework of the algorithm is as follows, given a route set $\mathbb{R}$ and a budget limit B,

1) Build the Selective Bayesian Network $\mathcal{N}(\mathbb{R})$.
2) Repeatedly use $\mathcal{N}(\mathbb{R})$ to select query set S_k to ask the crowd and receive k answers.
3) Update the probabilities of all routes in $\mathbb{R}$ with the crowdsourced k answers.
4) Repeat 2 and 3 until the budget B is used up, and then report the most likely best route.

Please note that in implementation we modularize the algorithm into offline part and online part. The Selective Bayesian Network is built offline for efficiency. The spacial relations between the routes and the vertices are also computed offline, and then stored in a spacial information table $T_{si}(\mathbb{R})$, which supports queries of form "$R \to v$?" within linear time. As stated above, precise reasoning on all possible combinations of S_k in each round is too expensive, so we use two relaxations. The first one

Input: A path set $\mathbb{R}$ and its query set $\mathbb{Q}$, the number k of queries per round and the total budget B
Output: The most likely best route
Build the Selective Bayesian network $\mathcal{N}(\mathbb{R})$ according to Lemma 4.1
while $B \neq 0$ **do**
  Set $S_k = \emptyset$
  while $|S_k| < k$ **do**
    for *each* $Q \in \mathbb{Q}$ **do**
      Calculate $\Delta H_{S_k \cup Q}$ with $\mathcal{N}(\mathbb{R})$ and Theorem 4.2
    end
    Set $Q_{max} = \text{argmax}_{Q \in \mathbb{Q}} \Delta H_{S_k \cup Q}$
    Add Q_{max} into S_k and remove it from $\mathbb{Q}$
  end
  Ask queries in S_k to crowd and receive the corresponding answers A_{S_k}
  for *each answer* $v \in A_{S_k}$ **do**
    for *each* $R \in \mathbb{R}$ **do**
      Set $Pr(R) = Pr(BR = R | v)$ with Formula 31
    end
  end
  Set $B = B - k$
  if $B < k$ **then**
    $k = B$
  end
end
return the route R with the maximum $Pr(R)$

Algorithm 2: k-selection with Selective Bayesian Net

is the same with the sampling-based approach in [1], namely, to approximate the best S_k by incrementally selecting queries, as in Lines 5 - 9 in Algorithm 2. The second relaxation is use sampling-based reasoning instead of precise reasoning, when calculating ΔH_{S_k} in Line 6. Although the joint probabilities needed in the calculation is also estimated by sampling, the precision is higher with the help of the spacial causalities embedded in $\mathcal{N}(\mathbb{R})$. There are lots of samplers available, including Gibbs [17], Hamiltonian Monte Carlo [18], Metropolis-Hastings [19], etc. In our work, we use the basic Gibbs sampling for the reasoning on $\mathcal{N}(\mathbb{R})$.

5 Experimental Evaluation

In this section, we report the experimental study to validate the effectiveness and efficiency of our proposals. First, we use synthetic data and a simulated crowd to explore wide rages of values for the parameters. Second, we conduct an experiment with real-world datasets to verify our conclusions on the synthetic data.

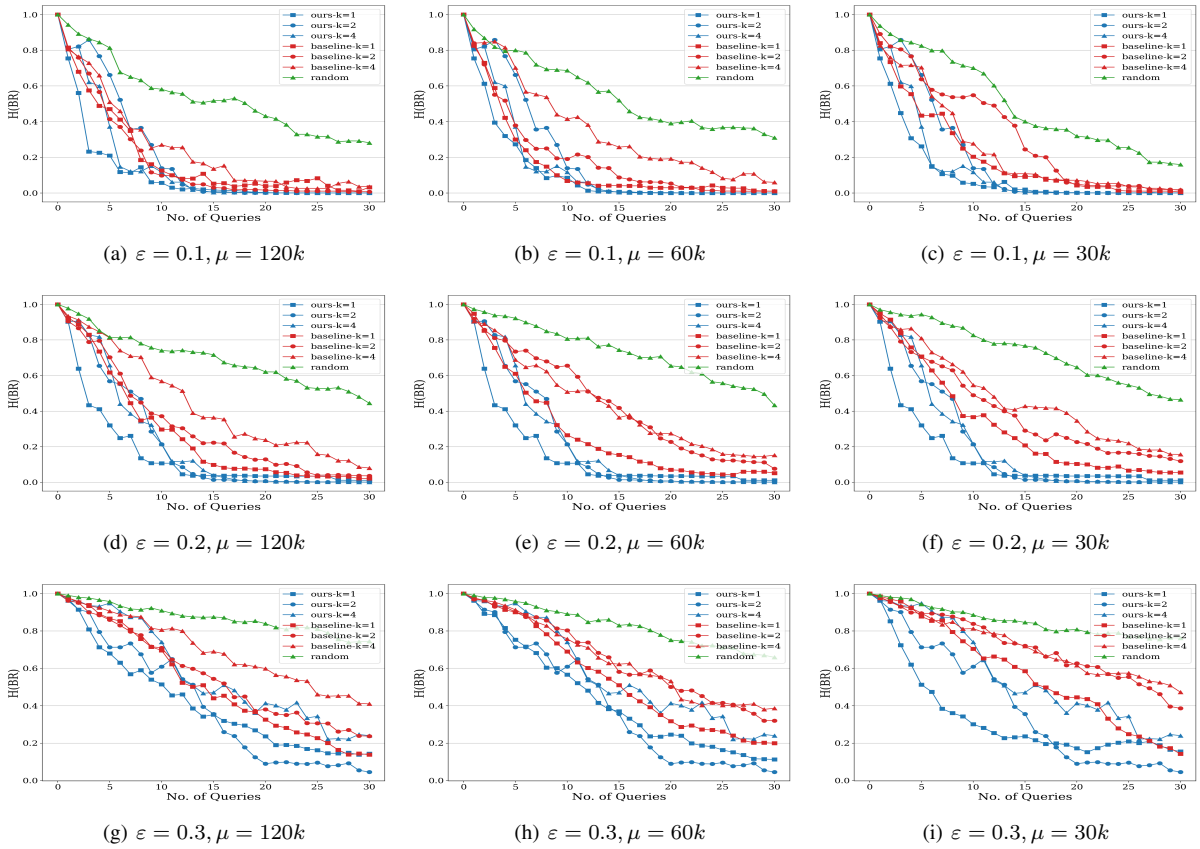
In the experiments, we compare the performances of the following three categories of algorithms.

1) (**ours-k=x**) k-selection with Selective Bayesian Network, with $k = 1, 2, 4$, our method proposed in Section 3.
2) (**baseline-k=x**) sampling-based algorithm proposed in [1], with $k = 1, 2, 4$.
3) (**random**) a naive algorithm - to select a random set of queries, with $k = 1, 2, 4$, to ask the crowd in each round.

In the rest of the paper, for the random selection, we only plot $k = 1$, and omit $k = 2, 4$, because the three curves almost coincides in all settings - selecting one random query is basically the same with selecting multiple random queries per round.

5.1 Simulation on Synthetic Data

We compare the performances of the above algorithms with the error rate of the crowd $\varepsilon = 0.1, 0.2, 0.3$, and the total sample size



(a) $\varepsilon = 0.1, \mu = 120k$ (b) $\varepsilon = 0.1, \mu = 60k$ (c) $\varepsilon = 0.1, \mu = 30k$

(d) $\varepsilon = 0.2, \mu = 120k$ (e) $\varepsilon = 0.2, \mu = 60k$ (f) $\varepsilon = 0.2, \mu = 30k$

(g) $\varepsilon = 0.3, \mu = 120k$ (h) $\varepsilon = 0.3, \mu = 60k$ (i) $\varepsilon = 0.3, \mu = 30k$

Fig. 4. Error rate $\boldsymbol{\varepsilon = 0.1, 0.2, 0.3}$ and sample size $\boldsymbol{\mu = 30k, 60k, 120k}$

$\mu = 30k, 60k, 120k$. The route sets with a best route distributions are randomly generated, and for each of them, the corresponding query set and pmf are then calculated. Then several best routes are sampled according to the distribution, and for each best route, we run the competing algorithms separately and finally plot the results, as shown in Figure 4. We summarize the experimental result in following aspects.

Varying sample size. The sample size for probability estimation, μ, is set to 30k, 60k and 12k, respectively. The performances are directly effected by the sample size, as during the computation, some required probabilities are estimated with the samples. In the result in Figure 4, it is obvious that our method is much less sensitive about the sample size compared with the baseline. With the decrease of the sample size, the advantage of our proposal becomes more significant.

Varying number of queries per round. The number of routing queries to select each round, k, is set to 1, 2 and 4. As shown in Figure 4, smaller k tends to be more effective on reducing the selection hardness, for both our proposal and the baseline. This is partially because the bigger k is, the more complex the correlation among the queries is, thus the less precisely the sample estimates the joint probabilities, if we fix the sample size. Another reason is that each routing query is selected based on the previously crowdsourced answers. The bigger k is, the less frequently the best route distribution is updated by the crowd. Interestingly, for our method, when k is increasing, the performance degradation is relatively small, in terms of the descent rate and the minimum number of queries to bring $H(BR)$ down to 0. The only exception is in the roughest setting, $\varepsilon = 0.3, \mu = 30k$, yet our approach even with $k = 4$ is still better than the baseline with $k = 1$.

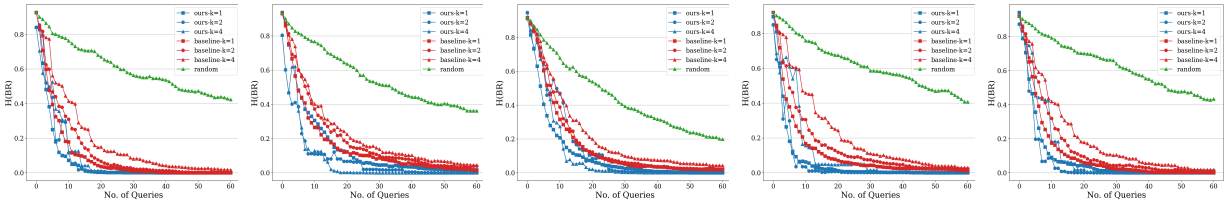
Varying error rate. The error rate of the crowd, ε, is set to 0.1, 0.2 and 0.3. For all the algorithms, the lower ε is, the faster the performance converges. For a crowd with higher accuracy, smaller amount of the queries are necessary to suggest a best route.

Different algorithms. The result in Figure 4 shows that for any combination of ε, μ, k, our approach dominates the baseline and the random algorithms. In most cases, our approach with $k = 4$ shows even better performance than the baseline with any k.

We can conclude that the k-selection with the Selective Bayesian Network is much less sensitive about the sample size μ, the query batch size k and the error rate of the crowd ε. The spacial causalities provided by the Selective Bayesian Network make the routing query selection much more stable and robust.

5.2 Verification on Real Data

We use five real-world road-network datasets, namely, California Road Network (CA), San Francisco Road Network (SF), Road Network of North America (NA), City of San Joaquin County Road Network (TG) and City of Oldenburg Road Network (OL) [20], [21]. The road networks in these datasets are obtained from Digital Chart of the World Server. Although each of the above datasets contains a lot of nodes and paths, the data is actually not dense enough in terms of routing queries. For a given pair of locations, the number of intersections of different routes is usually small, which makes the total number of routing queries small. So in the experiment we use a data augmentation to supplement



(a) CA (b) SF (c) NA (d) TG (e) OL

Fig. 5. Comparison on real datasets: CA, SF, NA, TG and OL

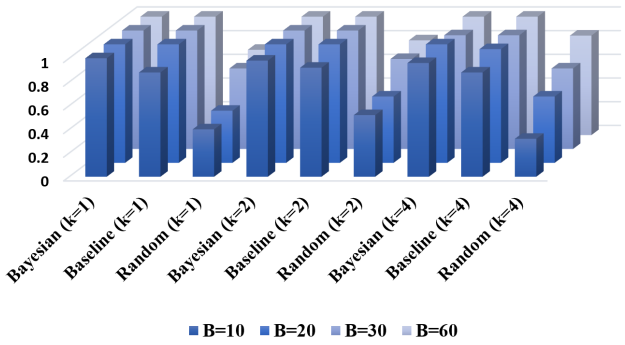


Fig. 6. Precision with budget B=10, 20, 30, 60

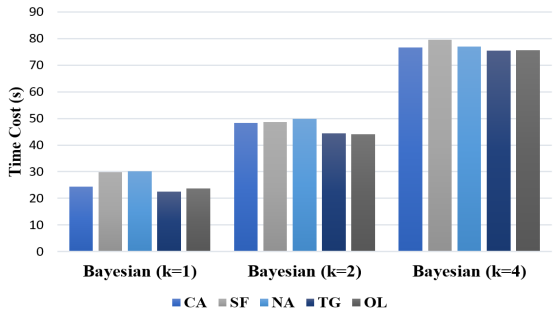


Fig. 7. Average time cost

necessary routing queries to deal with the query sparsity of the data. With the data augmentation, we construct 25 route sets (5 from each dataset), and each route set contains more than 60 routing queries. For each route set, we calculate the distribution of the best route according to the normalized costs of the routes, and then sample the best route according to the distribution for 10 runs of algorithms, with k=1,2 and 4 respectively. We test totally 750 runs on the real-world road networks for each algorithm. As shown in Figure 5, the result is consistent with synthetic data. Our best route selection with the Selective Bayesian Network performs better on all datasets.

5.3 Effectiveness and Time Cost

Below we conduct experiments to exhibit the goodness of paths selected by the crowd. It worth mentioning that for these 25 route sets from real-world data, the precision of each algorithms except random is quite high, after asking 60 routing queries. So we vary the budget B=10, 20, 30, 60, to test the performances of the algorithms when the budget is limited. As shown in Figure 6, our method performs better than others in terms of precision. In most cases, the precision of our method is close to 100% even with a budget of only 10 queries.

Moreover, we also test the average time cost of our algorithm with k=1, 2 and 4, on all the datasets. As shown in Figure 7, for all the route sets from the real data, our algorithm successfully selects 60 best routing queries and then suggests a best route with nearly 100% precision within 90 seconds, which is impressive for a crowdsourcing framework.

6 Related Works

6.1 Crowdsourcing

The recent development of crowdsourcing brings us a brand new opportunity to engage human intelligence in the process of answering queries (see [22] as a survey). Crowdsourcing provides a new problem-solving paradigm [23], [24], which has been blended into several research communities. In particular, crowdsourcing-based data management techniques have recently attracted much attention in the database and data mining communities. From a practical viewpoint, [25] proposed and developed 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, from the viewpoint of theoretical study, many fundamental queries have been extensively studied, including filtering [27], max [28], sorting [29], join [29], [30], and so on. Besides, crowdsourcing-based solutions of many complex algorithms have also been developed, such as categorization based on graph search [15], clustering [31], entity resolution [32], [33], analysis over social media [34], and tagging in social networks [35], trip planning [36], pattern mining [37] etc.

6.2 Path Recommendation with Crowd

Finding the most desirable path has now been receiving tremendous research interest for decades [38], [39], [40]. The most popular topic in this area is shortest path finding, which has been extensively studied for over fifty years. If the weight on each edge represents travel time, shortest path finding becomes fastest path finding. Such as, the authors in [38] introduce a favorite route proposal scheme to provide route recommendations, and the scheme they proposed can generate better recommendations than alternative learning algorithms. Specifically, the work in [41] considers the availability of routes for different users. Two efficient algorithms are proposed in [42], which uses minimum on-road travel cost function. The authors in [39] improve the performance of batch shortest path algorithms they proposed by revisiting the problem of query clustering, and three query decomposition methods are proposed for fast query clustering.

In addition, approaches in [43], [44], [45], [46] predict the future trajectory by introducing deep learning models. To help route decision, the approach in [43] uses RNN to build a trajectory model, assuming that the itinerary of the destination link is a known parameter. In [4], a deep probabilistic model is presented, which unifies three key explanatory factors for most likely route prediction. Moreover, in order to effectively share the statistical strength, an adjoint generative model is proposed to learn representations of k-destination proxies.

To the best of our knowledge, this is the first work studying the path selection problem with the help of crowd. The essential objective is to make it easier for users to select the best path among a number of candidates. The authors in [47] propose a system to leverage crowds' knowledge to improve the quality of recommended routes. This paper distinguishes itself with [47] from the following aspects: first, we ask the crowd to identify the direction at each road intersection; second, we adjust the distribution of recommended paths, rather than identify the very best one.

7 Conclusion and Future Work

A GPS-based navigation system usually suggests multiple paths for a pair of given source and target. Therefore, a struggling problem for users is to select the best one among them, namely *the best path selection* problem. Too many suggested paths may jeopardize the usability of recommendation data, and decrease user satisfaction. Although existing studies have partially solved this problem through integrating historical traffic logs or updating traffic conditions periodically, their solutions neglect the potential contribution of human experiences. In this paper, we resort to crowdsourcing to ease the pain of best path selection. In particular, we design two types of questions, namely Routing Query (RQ) and Binary Routing Query (BRQ), to ask the crowd to decide the direction at each road intersection. We consider the problem of selecting the best k RQs. Furthermore, we propose a series of efficient algorithms, which dynamically manage the questions in order to reduce the selection hardness with a limited budget of questions. Finally, we verified the effectiveness and efficiency of our proposed approaches through experiments with synthetic and real-world datasets.

There are many further research directions to explore. First, an immediate interesting topic is how to create one HIT with multiple RQ/BRQ questions. To do this, our exact formulation has to be modified since some questions would have been answered by the same worker, so the assumption that each question is independently answered does not hold. In future works, we would be interested in examining the trade-off between this decrease in data quality and the cost savings due to larger HITs. Second, although we proposed in this paper an efficient approach to recommend the best path by k-RQ selection, we omitted the take into account manpower scheduling problem in crowdsourcing and the delay to get the answers of the crowd. It will be more practical and interesting to design and conduct some real-world experiments to further verify our framework. Last, in our approach, we did not consider the cases that some workers failed to return their answers. In our current framework, we can simply ignore a crowdsourcing worker if no answer is returned. In other words, the worker fails to provide any useful observation to lower the uncertainty of the best path. But it is more interesting to model the failure rate of returning answers to see what happens. We leave these interesting topics for future work.

Acknowledgments

Haodi Zhang is the corresponding author of this work. Chen Zhang and Haodi Zhang have contributed equally to this work. The authors are grateful to the crowd workers for their efforts to complete the experiment. The work is partially supported by the National Natural Science Foundation of China (Grant No. NSFC-61806132 and No. 61729201), Tencent Rhino-Bird Open Fund, Hong Kong RGC GRF Project 16202218 CRF Projects C6030-18G, C1031-18G, C5026-18G, AOE Project AoE/E-603/18, Guangdong Basic and Applied Basic Research Foundation 2019B151530001, Hong Kong ITC ITF grants ITS/044/18FX and ITS/470/18FX, Microsoft Research Asia Collaborative Research Grant, Didi-HKUST joint research lab project, and Wechat and Webank Research Grants.

References

[1] C. J. Zhang, Y. Tong, and L. Chen, "Where to: Crowd-aided path selection," *Proc. VLDB Endow.*, vol. 7, no. 14, pp. 2005–2016, 2014.

[2] W. Luo, H. Tan, L. Chen, and L. M. Ni, "Finding time period-based most frequent path in big trajectory data," in *SIGMOD Conference*, 2013, pp. 713–724.

[3] H. V. Jagadish, A. Chapman, A. Elkiss, M. Jayapandian, Y. Li, A. Nandi, and C. Yu, "Making database systems usable," in *SIGMOD Conference*, 2007, pp. 13–24.

[4] X. Li, G. Cong, and Y. Cheng, "Spatial transition learning on road networks with deep probabilistic models," *ICDE. IEEE*, 2020.

[5] K. Zheng, Y. Zheng, X. Xie, and X. Zhou, "Reducing uncertainty of low-sampling-rate trajectories," in *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012*, A. Kementsietsidis and M. A. V. Salles, Eds. IEEE Computer Society, 2012, pp. 1144–1155.

[6] H. Su, K. Zheng, H. Wang, J. Huang, and X. Zhou, "Calibrating trajectory data for similarity-based analysis," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, K. A. Ross, D. Srivastava, and D. Papadias, Eds. ACM, 2013, pp. 833–844.

[7] C. J. Zhang, L. Chen, H. V. Jagadish, M. Zhang, and Y. Tong, "Reducing uncertainty of schema matching via crowdsourcing with accuracy rates," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 1, pp. 135–151, 2020.

[8] C. J. Zhang, L. Chen, Y. Tong, and Z. Liu, "Cleaning uncertain data with a noisy crowd," in *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015*, J. Gehrke, W. Lehner, K. Shim, S. K. Cha, and G. M. Lohman, Eds. IEEE Computer Society, 2015, pp. 6–17.

[9] L. Liu, J. Xu, S. S. Liao, and H. Chen, "A real-time personalized route recommendation system for self-drive tourists based on vehicle to vehicle communication," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3409–3417, 2014.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nat.*, vol. 518, no. 7540, pp. 529–533, 2015.

[11] H. Park and J. Widom, "Query optimization over crowdsourced data," *PVLDB*, vol. 6, no. 10, pp. 781–792, 2013.

[12] M. DeGroot and M. Schervish, *Probability and Statistics*, ser. Addison-Wesley series in statistics. Addison-Wesley, 2002.

[13] A. P. Dawid, "Conditional independence in statistical theory," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 41, no. 1, pp. 1–31, 1979.

[14] H. Su, J. Deng, and L. Fei-Fei, "Crowdsourcing annotations for visual object detection," in *AAAI Technical Report, 4th Human Computation Workshop*, 2012.

[15] 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*, vol. 4, no. 5, pp. 267–278, 2011.

[16] M. Cover Thomas and A. Thomas Joy, "Elements of information theory," *New York: Wiley*, vol. 3, pp. 37–38, 1991.

[17] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721–741, 1984.

[18] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid monte carlo," *Physics Letters B*, vol. 195, no. 2, pp. 216 – 222, 1987.

[19] W. D. Hastings, "Monte carlo sampling methods using markov chains and their applications," 1970.

[20] M. L. Yiu, D. Papadias, N. Mamoulis, and Y. Tao, "Reverse nearest neighbors in large graphs," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 4, pp. 540–553, 2006.

[21] X. Xiao, B. Yao, and F. Li, "Optimal location queries in road network databases," in *ICDE*, 2011, pp. 804–815.

[22] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing systems on the world-wide web," *Commun. ACM*, vol. 54, no. 4, pp. 86–96, 2011.

[23] D. C. Brabham, "Crowdsourcing as a model for problem solving an introduction and cases," *Convergence February 2008 vol. 14 no. 1 75-90*, 2008.

[24] T. Malone, R. Laubacher, and C. Dellarocas, "Harnessing crowds: Mapping the genome of collective intelligence," MIT, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, USA, Research Paper No. 4732-09, February 2009, sloan Research Paper No. 4732-09.

[25] 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*, vol. 4, no. 12, pp. 1387–1390, 2011.

[26] A. G. Parameswaran and N. Polyzotis, "Answering queries using humans, algorithms and databases," in *CIDR*, 2011, pp. 160–166.

[27] 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*, 2012, pp. 361–372.

[28] S. Guo, A. G. Parameswaran, and H. Garcia-Molina, "So who won?: dynamic max discovery with the crowd," in *SIGMOD Conference*, 2012, pp. 385–396.

[29] A. Marcus, E. Wu, D. R. Karger, S. Madden, and R. C. Miller, "Human-powered sorts and joins," *PVLDB*, vol. 5, no. 1, pp. 13–24, 2011.

[30] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng, "Leveraging transitive relations for crowdsourced joins," in *SIGMOD Conference*, 2013, pp. 229–240.

[31] R. Gomes, P. Welinder, A. Krause, and P. Perona, "Crowdclustering," in *NIPS*, 2011, pp. 558–566.

[32] J. Wang, T. Kraska, M. J. Franklin, and J. Feng, "Crowder: Crowdsourcing entity resolution," *PVLDB*, vol. 5, no. 11, pp. 1483–1494, 2012.

[33] S. E. Whang, P. Lofgren, and H. Garcia-Molina, "Question selection for crowd entity resolution," *PVLDB*, vol. 6, no. 6, pp. 349–360, 2013.

[34] C. C. Cao, J. She, Y. Tong, and L. Chen, "Whom to ask? jury selection for decision making tasks on micro-blog services," *PVLDB*, vol. 5, no. 11, pp. 1495–1506, 2012.

[35] M. Das, S. Thirumuruganathan, S. Amer-Yahia, G. Das, and C. Yu, "Who tags what? an analysis framework," *PVLDB*, vol. 5, no. 11, pp. 1567–1578, 2012.

[36] H. Kaplan, I. Lotosh, T. Milo, and S. Novgorodov, "Answering planning queries with the crowd," *PVLDB*, vol. 6, no. 9, pp. 697–708, 2013.

[37] Y. Amsterdamer, Y. Grossman, T. Milo, and P. Senellart, "Crowd mining," in *SIGMOD Conference*, 2013, pp. 241–252.

[38] P. Campigotto, C. Rudloff, M. Leodolter, and D. Bauer, "Personalized and situation-aware multimodal route recommendations: The FAVOUR algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 92–102, 2017.

[39] Y. Liu, K. Zhao, G. Cong, and Z. Bao, "Online anomalous trajectory detection with deep generative sequence modeling," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 2020, pp. 949–960.

[40] L. Li, S. Wang, and X. Zhou, "Time-dependent hop labeling on road network," in *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*. IEEE, 2019, pp. 902–913.

[41] M. S. Rahaman, Y. Mei, M. Hamilton, and F. D. Salim, "Capra: A contour-based accessible path routing algorithm," *Information Sciences*, vol. 385, pp. 157–173, 2017.

[42] L. Li, K. Zheng, S. Wang, W. Hua, and X. Zhou, "Go slow to go fast: minimal on-road time route scheduling with parking facilities using historical trajectory," *The VLDB Journal*, vol. 27, no. 3, pp. 321–345, 2018.

[43] H. Wu, Z. Chen, W. Sun, B. Zheng, and W. Wang, "Modeling trajectories with recurrent neural networks." IJCAI, 2017.

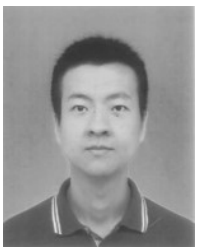
[44] J. Zhao, J. Xu, R. Zhou, P. Zhao, C. Liu, and F. Zhu, "On prediction of user destination by sub-trajectory understanding: A deep learning based approach," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1413–1422.

[45] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "Deepmove: Predicting human mobility with attentional recurrent networks," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1459–1468.

[46] Y. Chen, C. Long, G. Cong, and C. Li, "Context-aware deep model for joint mobility and time prediction," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 106–114.

[47] H. Su, K. Zheng, J. Huang, H. Jeung, L. Chen, and X. Zhou, "Crowdplanner: A crowd-based route recommendation system," in *ICDE*, 2014.

Chen Zhang received the PhD degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology in 2015. He is currently a research assistant professor of the Department of Computing, Hong Kong Polytechnic University, Hong Kong, SAR China. Before joining the Department, he worked as a senior manager of the Big Data Institute at HKUST. He is broadly interested in Crowdsourcing, Fintech and Machine Learning.

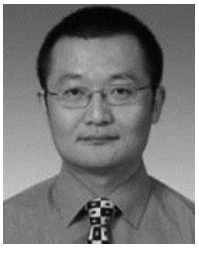


Haodi Zhang received the PhD degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, in 2016. He is currently a principal investigator in the Shanghai Research Center for Brain Science and Brain-Inspired Intelligence, Shanghai, China, and an assistant professor in the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

Weiteng Xie received the Bachelor degree from the College of Physics and Electronic Science, Hubei Normal University, in 2018. He is currently a Master student in the College of Computer Science and Software Engineering, Shenzhen University and Guangdong Laboratory of Artificial Intelligence and Digital Economy, Shenzhen, China.

Nan Liu is currently an undergraduate student in the Computer Science and Engineering Division, College of Engineering, University of Michigan, Ann Arbor, USA, and is currently visiting Shenzhen University. His research interests include crowdsourcing, crowd-aided uncertainty reduction, path selection, etc.

Kaishun Wu received the PhD degree in Department of Computer Science and Engineering, Hong Kong University of Science and Technology, in 2011. He is currently a distinguished professor in the College of Computer Science and Software Engineering, Shenzhen University and Guangdong Laboratory of Artificial Intelligence and Digital Economy, Shenzhen, China.



Lei Chen, IEEE Fellow, received the PhD degree in computer science from the University of Waterloo, Canada, in 2005. He is currently a professor in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His research interests include crowdsourcing over social media, social media analysis, probabilistic and uncertain databases, and privacy-preserved data publishing.

APPENDIX

Proof of Theorem 4.2

Proof. Suppose $S_k = \{(v^1_{st}, C_1, t), \ldots, (v^k_{st}, C_k, t)\}$, and $C_{S_k} = C_1 \times C_2 \times \ldots \times C_k$.

$$
\begin{aligned}
&\mathbb{E}H(BR|A_{S_k}) \\
&= \sum_{R\in\mathbb{R}} \sum_{v_{S_k}\in C_{S_k}} Pr(A_{S_k} = v_{S_k})(Pr(BR = R|A_{S_k} = v_{S_k}) \\
&\qquad \log Pr(BR = R|A_{S_k} = v_{S_k})) \\
&= \sum_{R\in\mathbb{R}} \sum_{v_{S_k}\in C_{S_k}} \Big[Pr(A_{S_k} = v_{S_k}) \\
&\qquad \frac{Pr(BR = R)Pr(A_{S_k} = v_{S_k}|BR = R)}{Pr(A_{S_k} = v_{S_k})} \\
&\qquad \log \frac{Pr(R)Pr(A_{S_k} = v_{S_k}|BR = R)}{Pr(A_{S_k} = v_{S_k})}\Big]
\end{aligned}
$$

Now, for a given $v_{S_k} \in C_{S_k}$, we calculate the above equation part by part according to different types of R.

- For those paths such that $\forall v^i_{st}.R \not\to v^i_{st}$, A_{S_k} is independent with $BR = R$, and $A_{Q_i} = v_i$ is also independent with each other for $i = 1..N$, i.e.

$$
\begin{aligned}
X_1 &= \sum_{\forall v^i_{st}.R\not\to v^i_{st}} \sum_{v_{S_k}\in C_{S_k}} \Big[Pr(A_{S_k} = v_{S_k}) \\
&\qquad \frac{Pr(BR = R)Pr(A_{S_k} = v_{S_k}|BR = R)}{Pr(A_{S_k} = v_{S_k})} \\
&\qquad \log \frac{Pr(BR = R)Pr(A_{S_k} = v_{S_k}|BR = R)}{Pr(A_{S_k} = v_{S_k})}\Big] \\
&= \sum_{\forall v^i_{st}.R\not\to v^i_{st}} \sum_{v_{S_k}\in C_{S_k}} \Big[Pr(A_{S_k} = v_{S_k}) \\
&\qquad \frac{Pr(BR = R)Pr(A_{S_k} = v_{S_k})}{Pr(A_{S_k} = v_{S_k})} \\
&\qquad \log \frac{Pr(BR = R)Pr(A_{S_k} = v_{S_k})}{Pr(A_{S_k} = v_{S_k})}\Big] \\
&= \sum_{\forall v^i_{st}.R\not\to v^i_{st}} \sum_{v_{S_k}\in C_{S_k}} Pr(A_{S_k} = v_{S_k})Pr(R)\log Pr(R) \\
&= \sum_{\forall v^i_{st}.R\not\to v^i_{st}} Pr(R)\log Pr(R)
\end{aligned}
$$

- For those paths such that $\forall v^i_{st} R \to v^i_{st}$, given that R goes through all starting vertices of the queries in S_k, the probability of $A_{S_k} = v_{S_k}$ is actually the joint probability, $Pr(A_{S_k} = v_{S_k}) = Pr(R \to A_1, R \to A_2, \ldots, R \to A_k) = Pr(A_1, A_2, \ldots, A_k)$. So, $Pr(A_{S_k} = v_{S_k}|BR = R) = Pr(A_1, A_2, \ldots, A_k|BR = R)$, where A_i is the node in $BN(G)$ corresponds the answer of RQ_i. Thus, we have,

$$
\begin{aligned}
X_2 &= \sum_{\forall v^i_{st}.R\to v^i_{st}} \sum_{v_{S_k}\in C_{S_k}} \Big[Pr(A_{S_k} = v_{S_k}) \\
&\qquad \frac{Pr(BR = R)Pr(A_{S_k} = v_{S_k}|BR = R)}{Pr(A_{S_k} = v_{S_k})} \\
&\qquad \log \frac{Pr(BR = R)Pr(A_{S_k} = v_{S_k}|BR = R)}{Pr(A_{S_k} = v_{S_k})}\Big] \\
&= \sum_{\forall v^i_{st}.R\to v^i_{st}} \sum_{v_{S_k}\in C_{S_k}} \Big[Pr(BR = R)Pr(A_{S_k} = v_{S_k}|BR = R) \\
&\qquad \log \frac{Pr(BR = R)Pr(A_{S_k} = v_{S_k}|BR = R)}{Pr(A_{S_k} = v_{S_k})}\Big] \\
&= \sum_{\forall v^i_{st}.R\to v^i_{st}} \Big[Pr(BR = R)Pr(A_{S_k} = R\cap C_{S_k}|BR = R) \\
&\qquad \log \frac{Pr(BR = R)Pr(A_{S_k} = R\cap C_{S_k}|BR = R)}{Pr(A_{S_k} = R\cap C_{S_k})}\Big] \\
&= \sum_{\forall v^i_{st}.R\to v^i_{st}} Pr(BR = R)\log \frac{Pr(BR = R)}{Pr(A_{S_k} = R\cap C_{S_k})}
\end{aligned}
$$

where

$$R \cap C_{S_k} = \{v^i_R \mid v^i_R \in C_i, (v^i_{st}, v^i_R) \in R\},$$

namely, the set of k answers for S_k which are consistent with R.
Suppose that $R \cap C_{S_k} = \{v^1_R, v^2_R, \ldots, v^k_R\}$, we have,

$$
\begin{aligned}
X_2 &= \sum_{\forall v^i_{st}.R\to v^i_{st}} Pr(BR = R)\log \frac{Pr(BR = R)}{Pr(A_{S_k} = R\cap C_{S_k})} \\
&= \sum_{\forall v^i_{st}.R\to v^i_{st}} Pr(R) log \frac{Pr(R)}{Pr(v^1_R, v^2_R, \ldots, v^k_R)}
\end{aligned}
$$

where $Pr(v^1_R, v^2_R, \ldots, v^k_R)$ can be calculated by the selection Bayesian network $\mathcal{N}(\mathbb{R})$.

- For those paths R such that $\exists v^i_{st}.R \to v^i_{st}$ and $\exists v^j_{st}.R \not\to v^j_{st}$, i.e. some of the starting vertices of S_k are in the path R, while the other are not. We denote the corresponding queries with these two categories of starting vertices by $S^+_k(R)$ and $S^-_k(R)$:

$$
\begin{aligned}
S^+_k(R) &= \{Q = (v_{st}, C, t) \mid Q \in S_k, R \to v_{st}\} \\
S^-_k(R) &= \{Q = (v_{st}, C, t) \mid Q \in S_k, R \not\to v_{st}\}
\end{aligned}
$$

We have,

$$
\begin{aligned}
X_3 &= \sum_{\substack{\exists v^i_{st}.R\to v^i_{st} \\ \exists v^j_{st}.R\not\to v^j_{st}}} \sum_{v_{S_k}\in C_{S_k}} \Big[Pr(A_{S_k} = v_{S_k}) \\
&\qquad \frac{Pr(BR = R)Pr(A_{S_k} = v_{S_k}|BR = R)}{Pr(A_{S_k} = v_{S_k})} \\
&\qquad \log \frac{Pr(BR = R)Pr(A_{S_k} = v_{S_k}|BR = R)}{Pr(A_{S_k} = v_{S_k})}\Big] \\
&= \sum_{\substack{\exists v^i_{st}.R\to v^i_{st} \\ \exists v^j_{st}.R\not\to v^j_{st}}} \sum_{v_{S_k}\in C_{S_k}} \Big[\\
&\qquad Pr(A_{S^+_k(R)} = v_{S^+_k(R)}) \prod_{RQ\in S^-_k(R)} Pr(A_Q = v^{RQ}_{S_k}) \\
&\qquad \frac{Pr(BR = R)Pr(A_{S^+_k(R)} = v_{S^+_k(R)}|BR = R)}{Pr(A_{S^+_k(R)} = v_{S^+_k(R)})} \\
&\qquad \log \frac{Pr(BR = R)Pr(A_{S^+_k(R)} = v_{S^+_k(R)}|BR = R)}{Pr(A_{S^+_k(R)} = v_{S^+_k(R)})}\Big]
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{\substack{\exists v^i_{st}.R\to v^i_{st}\\ \exists v^j_{st}.R\not\to v^j_{st}}} \Big[\sum_{v_{S^-_k}\in C_{S^-_k(R)}} \prod_{v\in v_{S^-_k}} Pr(A_Q = v)\Big] \\
&\quad \sum_{v_{S^+_k}\in C_{S^+_k(R)}} \Big[Pr(A_{S^+_k(R)} = v_{S^+_k(R)}) \\
&\quad \frac{Pr(BR=R)Pr(A_{S^+_k(R)} = v_{S^+_k(R)}|BR=R)}{Pr(A_{S^+_k(R)} = v_{S^+_k})} \\
&\quad \log \frac{Pr(BR=R)Pr(A_{S^+_k(R)} = v_{S^+_k(R)}|BR=R)}{Pr(A_{S^+_k(R)} = v_{S^+_k(R)})}\Big] \\
&= \sum_{\substack{\exists v^i_{st}.R\to v^i_{st}\\ \exists v^j_{st}.R\not\to v^j_{st}}} \Big[Pr(BR=R) \\
&\quad Pr(A_{S^+_k(R)} = R\cap C_{S^+_k(R)}|BR=R) \\
&\quad \log \frac{Pr(BR=R)Pr(A_{S^+_k(R)} = R\cap C_{S^+_k(R)}|BR=R)}{Pr(A_{S^+_k(R)} = R\cap C_{S^+_k(R)})}\Big] \\
&= \sum_{\substack{\exists v^i_{st}.R\to v^i_{st}\\ \exists v^j_{st}.R\not\to v^j_{st}}} Pr(R)\log\frac{Pr(R)}{Pr(A_{S^+_k(R)} = R\cap C_{S^+_k(R)})}
\end{aligned}
$$

where

$$
\begin{aligned}
R\cap C_{S^+_k(R)} = \{&v_p \mid \exists v_{st} \ s.t. \\
&< v_{st}, D, t > \in S_k, v_p\in D, (v_{st}, v_p)\in R\}
\end{aligned}
$$

Let $R\cap C_{S^+_k(R)} = \{v^1_R, v^2_R, \ldots, v^{|S^+_k(R)|}_R\}$, we have

$$
X_3 = \sum_{\substack{\exists v^i_{st}.R\to v^i_{st}\\ \exists v^j_{st}.R\not\to v^j_{st}}} Pr(R)\log\frac{Pr(R)}{Pr(v^1_R, v^2_R, \ldots, v^{|S^+_k(R)|}_R)}
$$

Finally, the expected reduction of selection hardness is

$$
\begin{aligned}
&\Delta H_{S_k} \\
=&H(BR) - \mathbb{E}H(BR|AS_k) \\
=&H(BR) + X_1 + X_2 + X_3 \\
=&-\sum_{R\in\mathbb{R}} Pr(R)\log(Pr(R)) + \sum_{\forall v^i_{st}.R\not\to v^i_{st}} Pr(R)\log Pr(R) \\
&+ \sum_{\forall v^i_{st}.R\to v^i_{st}} Pr(R) log\frac{Pr(R)}{Pr(v^1_R, v^2_R, \ldots, v^k_R)} \\
&+ \sum_{\substack{\exists v^i_{st}.R\to v^i_{st}\\ \exists v^j_{st}.R\not\to v^j_{st}}} Pr(R)\log\frac{Pr(R)}{Pr(v^1_R, v^2_R, \ldots, v^{|S^+_k(R)|}_R)} \\
=&-\sum_{\forall v^i_{st}.R\to v^i_{st}} Pr(R)\log Pr(v^1_R, v^2_R, \ldots, v^k_R) \\
&-\sum_{\substack{\exists v^i_{st}.R\to v^i_{st}\\ \exists v^j_{st}.R\not\to v^j_{st}}} Pr(R)\log Pr(v^1_R, v^2_R, \ldots, v^{|S^+_k(R)|}_R) \\
=&-\sum_{\exists v^i_{st}.R\to v^i_{st}} Pr(R)\log Pr(v^1_R, v^2_R, \ldots, v^{|S^+_k(R)|}_R) \\
=&-\sum_{S^+_k(R)\neq\emptyset} Pr(R)\log Pr(v^1_R, v^2_R, \ldots, v^{|S^+_k(R)|}_R)
\end{aligned}
$$

□