Journal of Combinatorial Optimization (2022) 44:2443–2459
https://doi.org/10.1007/s10878-021-00760-y

Data source selection for approximate query

Hongjie Guo[1] · **Jianzhong Li**[1] · **Hong Gao**[1]

Accepted: 15 May 2021 / Published online: 24 May 2021

Abstract
Exact query on big data is a challenging task due to the large numbers of autonomous data sources. In this paper, an efficient method is proposed to select sources on big data for approximate query. A gain model is presented for source selection by considering information coverage and quality provided by sources. Under this model, the source selection problem is formalized into two optimization problems. Because of the NP-hardness of proposed problems, two approximate algorithms are devised to solve them respectively, and their approximate ratios and complexities are analyzed. To further improve efficiency, a randomized method is developed for gain estimation. Based on it, the time complexities of improved algorithms are sub-linear in the number of data item. Experimental results show high efficiency and scalability of proposed algorithms.

Keywords Big data · Data quality · Source selection · Query approximation

1 Introduction

In the era of big data, with the rapid growth of useful information, data sets can be collected from a variety of sources. In the application with a large number of heterogeneous and autonomous data sources, it is infeasible and unnecessary to provide exact query answers on big data for the following reasons: (1).With the limited resources, querying on big data cannot be answered within an acceptable time bound; (2).Due to the autonomy of data sources, data sources are likely to contain overlap information,

A preliminary version of this paper appeared in Proceedings of 14th International Conference on Combinatorial Optimization and Applications, pp 61-75, 2020 (Guo et al. 2020).

✉ Hong Gao
honggao@hit.edu.cn

Hongjie Guo
hjguo@stu.hit.edu.cn

Jianzhong Li
lijzh@hit.edu.cn

[1] Harbin Institute of Technology, Harbin, China

querying redundant sources may bring some gain, but it will bring high additional costs; (3).Data sources are often of low quality, and querying low-quality data sources will even deteriorate the quality of query results.

Data source selection has received much attention in the literature. Dong et al. (2012) selects a subset of sources for data integration. This work formalizes several optimization goals for source selection and efficiently estimates resulting accuracy. A heuristic randomized approach is presented to approximate the optimal selection. However, this paper does not take into account data self-conflicting and incompleteness. Salloum et al. (2013) provides a method to determine the online query order on data sources. This methods is based on estimating overlaps between sources according to available statistics. It requires some prior statistics and overlooks data quality. Rekatsinas et al. (2014) selects sources by using freshness as a quality measure, which without comprehensively considering the quality of data sources, such as functional dependency and completeness. Lin et al. (2019) incorporates truth discovery techniques into source selection. It presents a probabilistic coverage model to measure the quality of sources. Nevertheless, this work needs a prior statistics of each attribute value. Li et al. (2018) attempts to find a subset of sources that maximizes the coverage with a bounded number of sources. A randomized approach for intersection set size estimation is leveraged to estimate the coverage without accessing the data sources. This work does not take into consideration data quality and cost.

In summary, currently, there is no efficient method to select data sources by considering data coverage, overlap and quality. We dedicate this paper to the development of an efficient approach that selects proper data sources before querying. Given an upper bound on the amount of data sources or data items that can be used, efficient methods are proposed for source selection. We present a gain model to evaluate sources by considering their data coverage, overlap and quality. Based on presented gain and cost models, the source selection problem is formulated into two optimization problems according to different application scenarios. Because of the NP-hardness of problems, two heuristic approximate algorithms is leveraged to solve them. These two approximate algorithms are proved can obtain the best possible approximate factor. To further improve the efficiency, a Monte-Carlo sampling based algorithm is devised to estimate the sources coverage, in this way, the time complexities of algorithms are sub-linear in the number of data item.

In this paper, we make the following contributions.

First, a gain model is proposed for data source selection. This model take into account data coverage, overlap and quality.

Second, two optimization problems for source selection, called BNMG and BCMG, are presented based on gain model. They all proved to be NP-hard.

Third, a simple greedy algorithm is used for BNMG. An enumerate embedded greedy algorithm is devised for BCMG to achieve a bounded approximation factor.

Finally, a randomized approach is developed for coverage estimation. Such estimation has a theoretical guarantee and without extra space cost.

2 Problem definition

Section 2.1 defines the basic notions used in this paper and quality metric of data source, Sect. 2.2 formulates the problems of source selection, Sect. 2.3 analyzes the complexities of proposed problems.

2.1 Basic notions and quality metric

Definition 1 (Data source) A dataset consists of a set of data sources $\mathbb{S} = \{S_i | 1 \leq i \leq m\}$. Each source S_i contains a set of tuples. Each tuple consists of a set of attributes value.

Definition 2 (Functional dependency(FD) (Codd et al. 1972)) An FD φ: $[A_1, \cdots, A_l] \rightarrow [B]$, where A_i and B are attributes in data source. The semantic of φ is that any two tuples are equal on the left-hand side attribute of φ, they should also be equal on the right-hand side, otherwise, we say such tuples violate the FD.

Definition 3 (Data item) Consider a data item domain $\mathcal{D}$. A data item $D \in \mathcal{D}$ denotes a particular aspect of a real-world entity. $\mathcal{D}_S$ denotes that a set of data items provided by S.

Definition 4 (Claim (Sun et al. 2018)) Each claim is a triple $< S, key, v >$, meaning that the source, S, claims a value, v, on a data item with key, key.

For example, in Table 1 there are two sources, S_1 and S_2. S_1 provides 4 data items: the name of book with ISBN=02010, the author of book with ISBN=02010, the name of book with ISBN=02011, the author of book with ISBN=02011. Similarly, S_2 provides 6 data items. Note that a source can provide conflicting claims for a data item, for instance, consider two claims: $< S_1, ISBN = 02010.Name, Java >$ and $< S_1, ISBN = 02010.Name, C++ >$, which means S_1 claims that the name of ISBN=02010 are Java and C++, respectively. However, only one of the conflicting values is true. S_2 also miss the value of the author of the book with ISBN=02010.

Next, we consider quality metrics. Three aspects should be considered when selecting sources. First, we prefer to select a source with high *coverage* and low *overlap*: such a source would return more new answers. Second, the answers returned by a high-quality source are of high *reliability*. Third, a source with high querying *cost* will yield worse performance.

Definition 5 (Coverage) The *coverage* of source S is the number of its provided data items, denoted by $V(S)$. Formally,

$$V(S) = |\mathcal{D}_S| \tag{1}$$

For source set $\mathcal{S}$ (a subset of $\mathbb{S}$), we have

$$V(\mathcal{S}) = |\cup_{S \in \mathcal{S}} \mathcal{D}_S| \tag{2}$$

Table 1 A running example of bookstores dataset

(a) Source S_1

Id	ISBN	Name	Author
t_1	02010	Java	Robert
t_2	02010	C++	Robert
t_3	02011	Algorithms	Thomas
t_4	02011	Algorithms	Thomas
t_5	02011	Algorithms	Thomas

(b) Source S_2

Id	ISBN	Name	Author
t_1	02010	Java	Robert
t_2	02010	Java	
t_3	02011	Algorithms	Thomas
t_4	02012	Discrete mathematics	Oscar Levin

Coverage of the source S represents the expected number of answers returned by S. *coverage* of the source set $\mathcal{S}$ reflects the total distinct data items provided by sources, which has already eliminated the *overlap* information.

As shown in in Table 1, a source may provide self-conflicting or incomplete data, which means that the source has low reliability, querying low reliability may lead to a bad result. Therefor, it is necessary to select data sources based on their reliability. In this paper, we measure source reliability as the maximum correct number of claims provided by source. The $reliability$ of source S is denoted by $R(S)$.

In Table 1, S_1 claims two different values for the name of book with ISBN=02010, and only one of these can be true. The upper bound of correct claims number provided by source S over key k is

$$u_{S,k} = \max_{v}(N_{S,k,v}) \tag{3}$$

where $N_{S,k,v}$ is the number of claims provided by S for k with a value v.

In Table 1. $N_{S_1,ISBN=02010.Name,Java}$ and $N_{S_1,ISBN=02010.Name,C++}$ are both equal to 1, only one of the conflicting values is true. $u_{S_1,ISBN=02010.Name}$ provided by Source S_1 over key $ISBN = 02010.Name$ is 1. While the upper bound of correct claims number provided by Source S_1 over key $ISBN = 02010.Author$: $u_{S_1,ISBN=02010.Author}$ is 2.

Definition 6 (Reliability) The $reliability$ of source S is the upper bound of correct number of claims provided by S.

$$R(S) = \sum_{k} u_{S,k} \tag{4}$$

For source set $\mathcal{S}$, we have

$$R(\mathcal{S}) = \sum_{S \in \mathcal{S}, k} u_{S,k} \tag{5}$$

Now, we define the $gain$ model of source selection. The gain model is a trade-off between information coverage and reliability of selected sources.

Definition 7 (Gain).

$$G(\mathcal{S}) = \alpha V(\mathcal{S}) + (1 - \alpha) R(\mathcal{S}) \tag{6}$$

where $\alpha \in [0, 1]$ is a parameter to control how much $coverage$ and $reliability$ to take into consideration.

Since collecting and integrating data sources requires resources and time. There comes a *cost* to collecting sources for querying.

Definition 8 (Cost) The $cost$ of source S is the total number of its provided claims, denoted by $C(S)$.

$$C(S) = \sum_{k,v} N_{S,k,v} \tag{7}$$

Similarly, for source set $\mathcal{S}$, we have

$$C(\mathcal{S}) = \sum_{S \in \mathcal{S}} C(S) \tag{8}$$

In the running example shown of Table 1. The $coverage$ of S_1 and S_2 is $V(S_1) = 4$, and $V(S_2) = 6$. The $reliability$ of S_1 and S_2 is $R(S_1) = 9$, and $R(S_2) = 7$. The $cost$ of S_1 and S_2 is $C(S_1) = 10$, and $C(S_2) = 8$. Therefor, the answer returned by S_1 is more reliability than S_2, while querying S_2 brings more information with less cost.

2.2 Problems

In this subsection we formulate our problems.

For a data source S, it comes with a $cost$ and owns a $gain$. It is impractical to maximize the $gain$ while minimizing the $cost$. Thus, we define the following two optimization problems for source selection.

In some scenarios, query system gives an upper **B**ound on the **N**umber of data sources can be used, and wishes to **M**aximize the **G**ain. We define this problem as BNMG. Formally, the problem is defined as follows.

Problem 1: (BNMG). Given a set of data sources $\mathbb{S}$ and a integer k, the BNMG problem is to find a subset $\mathcal{S}$ of $\mathbb{S}$, such that

$$\begin{aligned} &\max G(\mathcal{S}) \\ &Subject\ to: |\mathcal{S}| \leq k \end{aligned} \tag{9}$$

In some situations, query system gives an upper **B**ound on the **C**ost as the constraint, and the optimization goal is to obtain the **M**aximum **G**ain. We call this problem as BCMG.

Problem 2: (BCMG). Given a source set $\mathbb{S}$ and τ_c be a budget on cost, the BCMG problem is to find a subset $\mathcal{S}$ of $\mathbb{S}$, such that

$$\begin{aligned} &\max G(\mathcal{S}) \\ &Subject\ to: C(\mathcal{S}) \leq \tau_c \end{aligned} \tag{10}$$

2.3 Complexity results

Lemma 1 *The gain function (6) is non-negative, monotone and submodular.*

Proof **non-negative**. Obviously.

monotone.

For $x \in \mathbb{S} - \mathcal{S}$, if $G(\mathcal{S} \cup x) \geq G(\mathcal{S})$, the $gain$ model is monotone.

$$\alpha V(\mathcal{S} \cup x) \geq \alpha V(\mathcal{S}), obviously \tag{11}$$

$$(1-\alpha)R(\mathcal{S} \cup x) = (1-\alpha)R(\mathcal{S}) + (1-\alpha)R(x) \geq (1-\alpha)R(\mathcal{S}) \tag{12}$$

Combining Eqs.(11) and (12), then

$$G(\mathcal{S} \cup x) = \alpha V(\mathcal{S} \cup x) + (1-\alpha)R(\mathcal{S} \cup x) \geq \alpha V(\mathcal{S}) + (1-\alpha)R(\mathcal{S}) = G(\mathcal{S}) \tag{13}$$

submodular.

For $\mathcal{R} \subset \mathcal{S}$ and $x \in \mathbb{S} - \mathcal{S}$, if $G(\mathcal{S} \cup x) - G(\mathcal{S}) \leq G(\mathcal{R} \cup x) - G(\mathcal{R})$, the $gain$ model is submodular.

$$\begin{aligned} V(\mathcal{S} \cup x) - V(\mathcal{S}) &= V(\mathcal{S}) + V(x) - V(\mathcal{S} \cap x) - V(\mathcal{S}) \\ &= V(x) - V(\mathcal{S} \cap x) \end{aligned} \tag{14}$$

Similarly,

$$V(\mathcal{R} \cup x) - V(\mathcal{R}) = V(x) - V(\mathcal{R} \cap x) \tag{15}$$

As $\mathcal{R} \subset \mathcal{S}$, $V(\mathcal{S} \cap x) \geq V(\mathcal{R} \cap x)$. Hence

$$\alpha V(\mathcal{S} \cup x) - \alpha V(\mathcal{S}) \leq \alpha V(\mathcal{R} \cup x) - \alpha V(\mathcal{R}) \tag{16}$$

And

$$(1-\alpha)R(\mathcal{S}\cup x)-(1-\alpha)R(\mathcal{S})=(1-\alpha)R(\mathcal{S})+(1-\alpha)R(x)-(1-\alpha)R(\mathcal{S})\\ =(1-\alpha)R(x) \tag{17}$$

Similarly, $(1-\alpha)R(\mathcal{R}\cup x)-(1-\alpha)R(\mathcal{R})=(1-\alpha)R(x)$, we have

$$(1-\alpha)R(\mathcal{S}\cup x)-(1-\alpha)R(\mathcal{S})=(1-\alpha)R(\mathcal{R}\cup x)-(1-\alpha)R(\mathcal{R}) \tag{18}$$

Combining Eqs.(16) and (18), we get

$$G(\mathcal{S}\cup x)-G(\mathcal{S})\leq G(\mathcal{R}\cup x)-G(\mathcal{R}) \tag{19}$$

□

Lemma 2 *For a function f, if f is submodular, non-negative, and monotone. Selecting a k-element set $\mathcal{S}$ to maximize $f(\mathcal{S})$ is an NP-hard problem (Nemhauser et al. 1978).*

Theorem 1 *Both BNMG and BCMG are NP-hard problems.*

Proof For BNMG problem, based on Lemma 1 and 2. BNMG is NP-hard.

We prove the NP-hardness of BCMG problem by reducing Budgeted Maximum Coverage Problem (BMC) (Khuller et al. 1999) to it.

Given an instance of BMC Problem: A collection of sets $\Omega=\{S_1,S_2,\cdots,S_m\}$ with associated costs $\{C_i\}_{i=1}^m$ is defined over a domain of elements U with associated equivalent-weights. The goal is to find a collection of sets $\mathcal{S}\subseteq\Omega$, such that the total cost of elements in $\mathcal{S}$ does not exceed a given budget L, and the total weight of elements covered by $\mathcal{S}$ is maximized. BMC can be captured by the BCMG problem in the following way:

(1) the sources in BCMG represent the sets in BMC;
(2) the data items in BCMG represent the elements in BMC;
(3) the costs of sources in BCMG represent the costs of sets in BMC;
(4) the parameter α of the $gain$ model in BCMG is equal to 1.

BCMG is a generalization of BMC. BMC is NP-hard, and therefore BCMG is NP-hard.
□

3 Algorithms for source selection

Since both BNMG and BCMG are NP-hard. In this section, we first present a greedy approximation algorithm for BNMG. Then, we prove that a simple greedy algorithm is insufficient for solving the BCMG problem. Therefore, we propose a enumerate embedded greedy algorithm for BCMG.

3.1 Algorithm for BNMG

For a submodular and non-decreasing function f, f satisfies a property: The marginal gain of adding a source to a set of sources $\mathcal{S}$ is at least as high as the marginal gain of adding the same source to a superset of $\mathcal{S}$. Here, the marginal gain ($G(\mathcal{S} \cup S_i) - G(\mathcal{S})$ in this algorithm) is the difference between the gain after and before selecting the new source. Such problem can be solved well by a simple greedy algorithm, denoted by Greedy (shown in Algorithm 1), which selects k sources by iteratively choosing the source that provides the largest marginal gain (line 6).

Algorithm 1 Greedy

Require: $\mathbb{S}$, k
Ensure: a subset $\mathcal{S}$ of $\mathbb{S}$ with $|\mathcal{S}| \leq k$
1: Initialize $\mathcal{S} \leftarrow \emptyset$
2: **while** $|\mathcal{S}| < k$ **do**
3: **for all** $S_i \in \mathbb{S}$ **do**
4: $G(\mathcal{S} \cup S_i) \leftarrow \text{CompGain}(\mathcal{S} \cup S_i)$;
5: **end for**
6: $S_{opt} \leftarrow \arg\max_{S_i \in \mathbb{S}} G(\mathcal{S} \cup S_i) - G(\mathcal{S})$;
7: $\mathcal{S} \leftarrow \mathcal{S} \cup S_{opt}$;
8: $\mathbb{S} \leftarrow \mathbb{S} \backslash S_{opt}$;
9: **end while**

Time Complexity Analysis. The time complexity of Algorithm 1 depends on the complexity to compute the $gain$ of $(\mathcal{S} \cup S_i)$, this complexity is $O(n)$, where n represents the maximal number of data items in S_i. Clearly, the complexity of Algorithm 1 is $O(k * n * m)$, where k denotes the number of selected sources, and m denotes the number of sources in $\mathbb{S}$.

Theorem 2 *Algorithm 1 is a* $(1 - 1/e) - approximation\ algorithm.$

Proof The greedy algorithm get $(1 - 1/e)$ approximation ratio for a submodular and monotone function with a cardinality constraint (Nemhauser et al. 1978). □

3.2 Algorithm for BCMG

The greedy algorithm that selects at each step a source maximizing the ratio $\frac{G(\mathcal{S} \cup S_i) - G(\mathcal{S})}{C(S_i)}$ has an unbounded approximation factor. That is, the behavior of the worst case might be very far from that of the optimal solution. In Table 2 for example, two sources S_1 and S_2 are subjected to an FD: $key \rightarrow value$. According to our problem definition, S_1 has $V(S_1) = 1$, $R(S_1) = 1$, $C(S_1) = 1$; S_2 has $V(S_2) = p$, $R(S_2) = p$, $C(S_2) = p + 1$. Let $\mathbb{S} = \{S_1, S_2\}$, $\alpha = 0.5$, and the budget of cost $\tau_c = p + 1$. The optimal solution is the source S_2 and has $gain = p$, while the result selected by the greedy algorithm contains the source S_1 and has $gain = 1$. The approximation factor of this instance is p, and is therefore unbounded (p is not a constant).

Table 2 Two sources for an example

Source (a) S_1		
id	key	value
t_1	1	1

Source (b) S_2		
id	key	value
t_1	1	1
...	...	...
t_p	p	p
t_{p+1}	p	m

We improve the heuristic algorithm by leveraging the enumeration technique to achieve a constant approximation factor. The main idea is to utilize the partial enumeration technique (Shachnai and Tamir 2005) before calling greedy algorithm. The improved algorithm is denoted by EnumGreedy (shown in Algorithm 2). Let l be a fixed integer. Firstly, we enumerate all subsets of $\mathbb{S}$ that their cardinality are less than l and have cost at most τ_c, and select the subset that has the maximal $gain$ as the candidate solution (line 2). Then, we consider all subsets of $\mathbb{S}$ that their cardinality are l and have cost at most τ_c, and we complete each subset to a candidate solution using the greedy algorithm (line3-17). The algorithm outputs the candidate solution having the greatest $gain$ (line18-22).

Time Complexity Analysis. The running time of Algorithm 2 is $O(n \cdot m^{(l-1)})$ executions of enumeration and $O(n \cdot m^{l+2})$ executions of greedy, where m is the number of sources, n is the maximal number of data items in S_i. Therefore, for every fixed l, the running time is polynomial in $n \cdot m$.

Theorem 3 *For $l = 2$, Algorithm 2 achieves an approximation factor of $\frac{1}{2}(1 - \frac{1}{e})$ for the BCMG problem. For $l \geq 3$, Algorithm 2 achieves a $(1 - \frac{1}{e})$ approximation ratio for the BCMG, and this approximation factor is the best possible.*

Proof The proof is by generalized the proof of approximation factor for the BMC problem, presented in (Khuller et al. 1999). The detail is omitted here. □

4 Coverage estimation

The time complexities of proposed algorithms are determined by computing $gain$. In fact, the time complexities are dominated by the computing of $coverage$ since $reliability$ and $cost$ can be computed in constant time. To reduce the computation time, we introduce a method for coverage estimation based on a Monte-Carlo algorithm proposed in (Karp et al. 1989). The coverage estimation method is a sampling based randomized algorithm which returns the approximate coverage result with a failure probability (Sun et al. 2000). Given the error bound $0 < \varepsilon < 1$, and the failure probability $0 < \delta < 1$, the probability of the relative error of the result being less than

Algorithm 2 EnumGreedy

Input: $\mathbb{S}, \tau_c, l$
Output: a subset $\mathcal{S}$ of $\mathbb{S}$ with $C(\mathcal{S}) \leq \tau_c$
Initialize $\mathcal{S} \leftarrow \emptyset, \overline{\mathcal{S}'} \leftarrow \emptyset, \overline{\mathcal{S}''} \leftarrow \emptyset$
$\overline{\mathcal{S}'} \leftarrow \arg\max_{\overline{\mathcal{S}'} \subseteq \mathbb{S}} \{G(\overline{\mathcal{S}'}) | \, C(\overline{\mathcal{S}'}) \leq \tau_c, |\overline{\mathcal{S}'}| < l\}$
for all $\mathcal{S}'' \subseteq \mathbb{S}, |\mathcal{S}''| = l, C(\mathcal{S}'') \leq \tau_c$ **do**
  $\mathbb{S} \leftarrow \mathbb{S} \backslash \mathcal{S}''$
  for all $S_i \in \mathbb{S}$ **do**
    $G(\mathcal{S}'' \cup S_i) \leftarrow \text{CompGain}(\mathcal{S}'' \cup S_i)$;
    $C(S_i) \leftarrow \text{CompCost}(S_i)$;
  end for
  $S_{opt} \leftarrow \arg\max_{S_i} \frac{G(\mathcal{S}'' \cup S_i) - G(\mathcal{S}'')}{C(S_i)}$;
  if $C(\mathcal{S}'') + C(S_i) \leq \tau_c$ **then**
    $\mathcal{S}'' \leftarrow \mathcal{S}'' \cup S_{opt}$;
    $\mathbb{S} \leftarrow \mathbb{S} \backslash S_{opt}$;
  end if
  if $G(\mathcal{S}'') > G(\overline{\mathcal{S}''})$ **then**
    $\overline{\mathcal{S}''} \leftarrow \mathcal{S}''$;
  end if
end for
if $G(\overline{\mathcal{S}'}) \geq G(\overline{\mathcal{S}''})$ **then**
  $\mathcal{S} \leftarrow \overline{\mathcal{S}'}$;
else
  $\mathcal{S} \leftarrow \overline{\mathcal{S}''}$;
end if

ε is large than $1 - \delta$. The framework for coverage estimation is shown in Algorithm 3. Note that, this algorithm is embedded into our source selection algorithms (Algorithm 1 and 2). We call the source selection with Algorithm 3 embedded SB-Greedy algorithm and SB-EnumGreedy algorithm, respectively.

Algorithm 3 CompCoverage

Require: $\mathcal{S} = \{S_1, \cdots, S_t\}, \varepsilon, \delta,$
Ensure: $\hat{V}(\mathcal{S})$
Initialize $r = 0$;
while $r < 4t \ln(2/\delta)/\varepsilon^2$ **do**
  Randomly choose a source S_i with probability $|\mathcal{D}_{S_i}| / \sum_{j=1}^{t} |\mathcal{D}_{S_j}|$;
  Uniformly and randomly drawn a data item $D \in \mathcal{D}_{S_i}$;
  $cov(D) = 0$;
  for all $S_j \in \mathcal{S}$ **do**
    if $D \in \mathcal{D}_{S_j}$ **then**
      $cov(D) \leftarrow cov(D) + 1$;
    end if
  end for
  $X_r \leftarrow \sum_{j=1}^{t} |\mathcal{D}_{S_j}| / cov(D)$;
  $r \leftarrow r + 1$;
end while
$\hat{V}(\mathcal{S}) = \frac{\sum_{i=0}^{r} X_i}{r}$;

Lemma 3 *The expected time of Algorithm 3 is* $O(t^2 \log n \ln(1/\delta)(1/\varepsilon^2))$.

Proof In each round, the time complexity to test whether a data item lies in $\mathcal{D}_{S_j}$ is $O(\log|\mathcal{D}_{S_j}|)$. So the time bound of algorithm 3 is $O(t^2 \log n \ln(1/\delta)(1/\varepsilon^2))$, where n is the maximal number of data item in S_j. □

For problem BNMG, $|\mathcal{S}| \leq k$, we can deduce theorem 4.

Theorem 4 *The expected time of SB-Greedy for BNMG is* $O(k^3 m \log n \ln(1/\delta)(1/\varepsilon^2))$.

Definition 9 ((ε,δ)-approximation algorithm) Given $0 < \varepsilon < 1$, $0 < \delta < 1$, an algorithm A for coverage estimation problem is called an (ε,δ)-approximation algorithm if for every instance with the correct solution $V(\mathcal{S})$, the algorithm returns a solution $\hat{V}(\mathcal{S})$ such that

$$Pr[|\frac{V(\mathcal{S}) - \hat{V}(\mathcal{S})}{V(\mathcal{S})}| \leq \varepsilon] \geq 1 - \delta \tag{20}$$

Lemma 4 $\forall \mathcal{S} \subseteq \mathbb{S}$, *Algorithm 3 is a* (ε,δ)*-approximation algorithm.*

We first state some lemmas needs for the proof of Lemma 4.

Lemma 5 *For* X_r*, the random variable obtained by Algorithm 3 (line 11).* $E[X_r]$ *and* $\sigma[X_r]$ *represent the expectation and variance of* X_r*, respectively. Then* $E[X_r] = V(\mathcal{S})$.

Proof Suppose $\cup_{S_i \in \mathcal{S}} \mathcal{D}_{S_i}$ has n distinct data items $D_1, D_2 \cdots, D_n$. Then

$$E[X_r] = \sum_{i=1}^{n} Pr[D_i \ is \ chosen](\sum_{j=1}^{t} |\mathcal{D}_{S_j}|/cov(D_i)) \tag{21}$$

Since $Pr[D_i \ is \ chosen]$ is

$$\sum_{l=1}^{cov(D_i)} (|\mathcal{D}_{S_{i_l}}|/\sum_{j=1}^{t} |\mathcal{D}_{S_j}|)\frac{1}{|\mathcal{D}_{S_{i_l}}|} = cov(D_i)/\sum_{j=1}^{t} |\mathcal{D}_{S_j}| \tag{22}$$

where $S_{i_l}(l = 1, 2, \cdots, cov(D_i))$ are the sources which contain D_i, we obtain $E[X_r] = n = |\cup_{S_i \in \mathcal{S}} \mathcal{D}_{S_i}| = V(\mathcal{S})$. □

Lemma 6 *Let* $Y_1, Y_2, \cdots, Y_r$ *be independent random variables over interval* $[0, 1]$*, such that, for* $1 \leq i \leq r$, $E[Y_i] = p_i$*, where* $0 < p_i < 1$*. Then for* $Y = \sum_{i=1}^{r} Y_i$, $\mu = E[Y] = \sum_{i=1}^{r} p_i$*, and any* $\delta > 0$*,*

$$Pr[Y > (1+\delta)\mu] < [\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}]^{\mu} \tag{23}$$

and

$$Pr[Y < (1-\delta)\mu] < \exp(\frac{-\mu\delta^2}{2}) \tag{24}$$

Proof Lemma 6 is a generalization of the Chernoff bound, refer to (Motwani and Raghavan 1995). □

Proof of Lemma 4 We now apply the above lemmas to our problem. Let $X_i^{'} = X_i / \sum_{j=1}^{t} |\mathcal{D}_{S_j}|$. It is obvious that $X_i^{'}$s are independent random variables over [0, 1] and for all $1 \leq i \leq r$, $\mu = E[X_i^{'}] = V(\mathcal{S}) / \sum_{j=1}^{t} |\mathcal{D}_{S_j}| \in [1/t, 1]$. Without loss of generality, suppose $\varepsilon < 1$, then by Lemma 6

$$\begin{aligned} Pr[|\hat{V}(S) - V(S)| > \varepsilon V(S)] &= Pr[|\sum_{i=1}^{r} X_i^{'} - r\mu| > \varepsilon r\mu] \\ &\leq [\frac{e^{\varepsilon}}{(1+\varepsilon)^{(1+\varepsilon)}}]^{r\mu} + \exp(\frac{-r\mu\varepsilon^2}{2}) < 2\exp(\frac{-r\mu\varepsilon^2}{4}) \\ &\leq 2\exp(\frac{-r\varepsilon^2}{4t}) \end{aligned} \tag{25}$$

Since r=4t $\ln(2/\delta)/\varepsilon^2$ is sufficient to guarantee $2\exp(\frac{-r\varepsilon^2}{4t}) < \delta$, the proof is complete. □

Theorem 5 *SB-Greedy algorithm achieves a* $\frac{(1-\varepsilon)(1-1/e)}{1+\varepsilon}$*-approximation ratio for the BNMG problem with a probability larger than* $1 - \delta$.

Proof We denote the optimal solution of BNMG under the true values of coverage as β, the optimal solution under the estimated values of coverage based on algorithm 3 as γ, our solution is θ. Let G ($\hat{G}$) denote the gain of solution under true coverage (estimated coverage based on algorithm 3). Based on Lemma 4, for any solution $x \subseteq \mathbb{S}$, we have

$$Pr[|\frac{V(x) - \hat{V}(x)}{V(x)}| \leq \varepsilon] \geq 1 - \delta \tag{26}$$

We assume that

$$|\frac{V(x) - \hat{V}(x)}{V(x)}| \leq \varepsilon \tag{27}$$

Due to $G(x) = \alpha V(x) + (1-\alpha)R(x)$, $\hat{G}(x) = \alpha \hat{V}(x) + (1-\alpha)R(x)$, then

$$|\frac{G(x) - \hat{G}(x)}{G(x)}| = |\frac{\alpha(V(x) - \hat{V}(x))}{\alpha V(x) + (1-\alpha)R(x)}| \leq \varepsilon \tag{28}$$

Then we get

$$(1-\varepsilon)G(\beta) \leq \hat{G}(\beta) \leq (1+\varepsilon)G(\beta) \tag{29}$$

$$(1-\varepsilon)G(\theta) \leq \hat{G}(\theta) \leq (1+\varepsilon)G(\theta) \tag{30}$$

According to Theorem 2, θ achieves a $(1-1/e)$-ratio under the estimated coverage, then

$$\hat{G}(\theta) \geq (1-1/e)\hat{G}(\gamma) \tag{31}$$

Since γ is the optimal solution under the estimated coverage, we have

$$\hat{G}(\gamma) \geq \hat{G}(\beta) \tag{32}$$

Combining (15)-(18), we then give

$$G(\theta) \geq \frac{\hat{G}(\theta)}{1+\varepsilon} \geq \frac{(1-1/e)}{1+\varepsilon}\hat{G}(\gamma) \geq \frac{(1-1/e)}{1+\varepsilon}\hat{G}(\beta) \geq \frac{(1-\varepsilon)(1-1/e)}{1+\varepsilon}G(\beta) \tag{33}$$

According to Eqs.(12) and (19), we get $Pr[\frac{G(\theta)}{G(\beta)} \geq \frac{(1-\varepsilon)(1-1/e)}{1+\varepsilon}] \geq 1-\delta$.

□

According to Theorem 3, 4 and 5, we can easily deduce the following theorems.

Theorem 6 *For $l \geq 3$, SB-EnumGreedy algorithm achieves a $\frac{(1-\varepsilon)(1-1/e)}{1+\varepsilon}$-approximation ratio for the BCMG problem with a probability larger than $1-\delta$.*

Theorem 7 *The expected time of SB-EnumGreedy is $O(m^{l+3}\log n \ln(1/\delta)(1/\varepsilon^2))$.*

5 Experimental results

This section conducts some experiments to evaluate the proposed algorithms. We focus on two aspects: (1) the comparison between Greedy and SB-Greedy for BNMG problem, as well as EnumGreedy and SB-EnumGreedy for BCMG problem, and (2) how SB-Greedy and SB-EnumGreedy perform in terms of efficiency and scalability.

5.1 Experiment setup

5.1.1 Dataset

We conducted our comparison experiments on a synthetic dataset. The *Syn. Data* is a synthetic dataset with various data sources number(m) and data items number(n). We used 10 attributes $A1-A10$ and 8 FDs: $A1 \rightarrow A8$, $A1 \rightarrow A9$, $A1 \rightarrow A10$, $A2 \rightarrow A6$, $A2 \rightarrow A7$, $A3 \rightarrow A6$, $A3 \rightarrow A7$, $[A4, A5] \rightarrow A8$. Each data source randomly chose an attribute with 20% probability, and each source contains at least one of the FDs. The size of each data source is a random number in the range of [200000, 1000000].

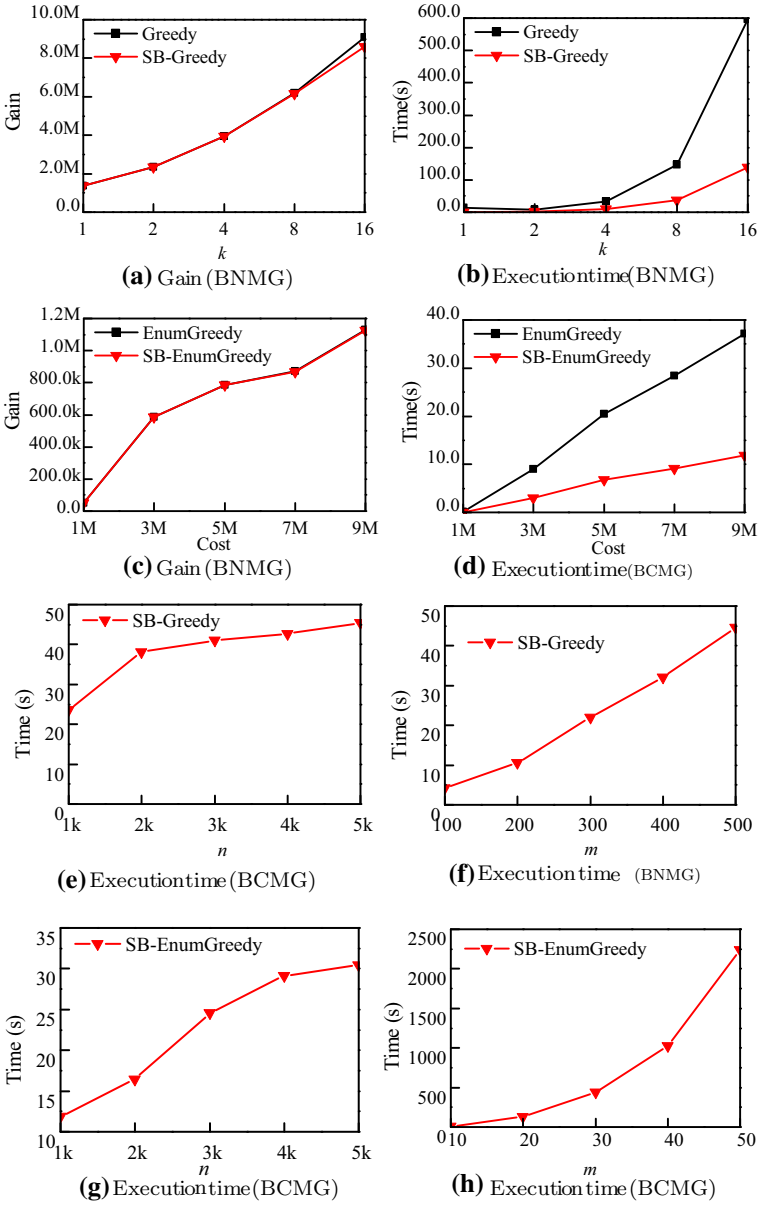

Fig. 1 Experimental results

5.1.2 Parameters

In this paper, we focus more on selecting data sources with high coverage, hence, we set $\alpha = 0.9$. Users can set different values for α according to their preferences.

Parameter l in Algorithm 2 is set to 2. The error bound ε and the failure probability bound δ are both set to 0.1.

5.1.3 Platforms

All experiments are implemented in Java and executed on a PC with Windows 7, a 16GB of RAM and a 3.6 GHz Intel i7-7700U CPU.

5.2 Comparison

For BNMG problem, we firstly compare the effectiveness of Greedy and SB-Greedy with k varying from 1 to 16 ($m = 100$, $n = 2000$), shown in Fig. 1a and b.

For BCMG problem, we compare EnumGreedy and SB-EnumGreedy with Bounded cost varying from 1 to 9 M ($m = 10$, $n = 1000$). The results are shown in Fig. 1c, d.

We have the following observations. (1) For BNMG problem, as the value of k increases, the runtimes of Greedy and SB-Greedy also increase, among which the running time of algorithm SB-Greedy changes more obviously. SB-Greedy achieves a significant runtime saving at the expense of very small gain. (2) BCMG problem gets a similar result. With the value of m increases, the runtimes of EunmGreedy and SB-EunmGreedy both increase, and SB-EunmGreedy is more sensitive to m. In this set of experiments, SB-EunmGreedy has a significantly lower running time with almost no gain sacrifice.

5.3 Efficiency and scalability

To further test how the data items and number of sources affect efficiency and scalability, we conduct more experiments on synthetic datasets. (1) Fig. 1(e, g) report the runtimes of both algorithms with varying the data items from 1000 to 5000 ($k = 8$ for BNMG, $Cost = 10M$ for BCMG). We observe that the runtimes of SB-Greedy and SB-EnumGreedy are grow sub-linear in data items. Meanwhile, the time complexity is independent to the data size, which demonstrates that the high efficiency and scalability of our algorithms to the data items and data size. Fig. 1f plots the running time of SB-Greedy, as we vary the number of data sources from 100 to 500 ($k = 8$ and $n = 500$). The runtime of SB-Greedy increases nearly linearly with data sources, which also indicates SB-Greedy has good scalability to the number of data sources. Fig. 1h manifests the runtimes of SB-EunmGreedy when the data sources varies from 10 to 50 ($n = 1000$ and $Cost = 10M$). As we can see, the runtime of SB-EunmGreedy is grows exponentially in data source. This illustrates that SB-EunmGreedy has poor scalability towards the number of data sources.

5.4 Summary

(1) The gain of SB-Greedy and SB-EnumGreedy are very close to the gain of Greedy and EnumGreedy. (2) The algorithms, which embedded a Monte-Carlo algorithm

to estimate coverage, outperform original methods both on efficiency and scalability significantly. (3) Our algorithms scale well on both the data size and the data items. (4) SB-Greedy has good scalability to the number of data sources while SB-EnumGreedy does not.

6 Conclusion

This paper studies source selection problem for query approximation. We first propose a gain model to evaluate the coverage and quality of data source. Based on the proposed model, we formulate source selection problem into two optimization problems, which are both proven to be NP-hard. Then we develop a greedy algorithm for bounded source number problem and an enumerate embedded greedy algorithm for bounded cost problem, both algorithms come with rigorous theoretical guarantees on their performance. Finally, we devise a randomized method to estimate coverage to further improve efficiency. Under this method, our propose algorithms get sub-linear complexities in the number of data item. Experimental results show our algorithms can select sources efficiently.

Funding This study was funded by the National Natural Science Foundation of China under grants 61732003 and 61832003.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Codd EF et al (1972) Relational completeness of data base sublanguages

Dong XL, Saha B, Srivastava D (2012) Less is more: selecting sources wisely for integration. Proc VLDB Endow 6(2):37–48

Guo H, Li J, Gao H (2020)Selecting sources for query approximation with bounded resources. In: International conference on combinatorial optimization and applications. Springer, pp. 61–75

Karp RM, Luby M, Madras N (1989) Monte-Carlo approximation algorithms for enumeration problems. J Algorithms 10(3):429–448

Khuller S, Moss A, Naor JS (1999) The budgeted maximum coverage problem. Inf Process Lett 70(1):39–45

Li L, Feng X, Shao H, Li J (2018) Source selection for inconsistency detection. In: International conference on database systems for advanced applications. Springer, pp. 370–385

Lin Y, Wang H, Li J, Gao H (2019) Data source selection for information integration in big data era. Info Sci 479:197–213

Motwani R, Raghavan P (1995) Randomized algorithms. Cambridge University Press, Cambridge

Nemhauser GL, Wolsey LA, Fisher ML (1978) An analysis of approximations for maximizing submodular set functions-i. Math Program 14(1):265–294

Rekatsinas T, Dong XL, Srivastava D (2014) Characterizing and selecting fresh data sources. In: Proceedings of the 2014 ACM SIGMOD international conference on Management of data. pp 919–930

Salloum M, Dong XL, Srivastava D, Tsotras VJ (2013) Online ordering of overlapping data sources. Proc VLDB Endow 7(3):133–144

Shachnai H, Tamir T (2018) Polynomial time approximation schemes. In: Handbook of approximation algorithms and Metaheuristics, 2nd edn. Chapman and Hall/CRC, pp. 125–156

Sun J, Li J, Gao H, Wang H (2018) Truth discovery on inconsistent relational data. Tsinghua Sci Technol 23(3):288–302

Sun L, Hong Z, Pixing Z (2000) A randomized algorithm for the union of sets problem. J Softw 11(12):1587–1593