$sky(\{S[Q_1, Q_2, \ldots, Q_n], (R, U)\}) =$
sky($sky(\{S[Q_1], (R, U)\}) \bigoplus \ldots \bigoplus$ $sky(\{S[Q_n], (R, U)\})$**)**.
Where, (1) $sky(\{S[Q_i], (R, U)\})$ is a function return the skyline set over policy space $\{S[Q_i], (R, U)\}$ based on minimizing R and U, and $S[Q_i]$ is the set of all policies whose length is $r_i - 1$, with values of other attributes are mapping to the whole domain, respectively.
(2) $\bigoplus$ is a symbol indicates that the policy spaces are reproduced, and the length of new policy is l, $l = (r_1 - 1 + r_2 - 1 + \ldots + r_n - 1)$.

Proof: Following are the notations used in our proof: $F_i = sky(\{S[Q_i], (R, U)\})$, $i = 1, \ldots, n$ is the skyline computation over the policies which are generated by only considering partitions of attribute Q_i and other attributes are mapped to the whole domain. $F^i = sky(\{S[Q_1, \ldots, Q_i], (R, U)\})$, $i = 1, \ldots, n$, is the skyline over the policies which are generated by considering partitions of attribute $Q_1, \ldots, Q_i$ and other attributes $Q_{i+1}, \ldots, Q_n$ are mapped to the whole domain.

Here our proof begins. First, when n=2, we must prove:

$$F^2 \subset F_1 \bigoplus F_2, \tag{8}$$

Second, when n=n-1, with an assumption that the Lemma 1 holds, we must prove:

$$F^n \subset F^{n-1} \bigoplus F_n, \tag{9}$$

Finally, we can prove $F^n \subset F_1 \bigoplus \ldots \bigoplus F_n$ by applying the mathematical induction, then clearly our conclusion holds: $F^n = sky(F_1 \bigoplus \ldots \bigoplus F_n)$.

Next, we give the proof of (8) and (9). For (8), First, the definition of $\bigoplus$ guarantees that $F_1 \bigoplus F_2$ is the set of answering over the universal policy set S. That is, the F^2 and $F_1 \bigoplus F_2$ is in the same space $S[Q_1, Q_2]$. Second, for any policy p that is not in $F_1 \bigoplus F_2$ (p is in the complement set of $F_1 \bigoplus F_2$), and suppose p is composed of one policy $\alpha_1(R_1, U_1)$ from $S[Q_1]$ and another policy $\alpha_2(R_2, U_2)$ from $S[Q_2]$. Then we have two cases: a) α_1 is not in F_1 and α_2 is not in F_2, b) α_1 is not in F_1 and α_2 is in F_2. For the first case, there must exist one policy $\beta_1 \in F_1$, such that $\beta_1(R_1^{'}, U_1^{'}) \prec \alpha_1(R_1, U_1)$, and another policy $\beta_2 \in F_2$, such that $\beta_2(R_2^{'}, U_2^{'}) \prec \alpha_2(R_2, U_2)$, then we have the following inequalities:
$R_1^{'} * R_2^{'} < R_1 * R_2$, $U_1^{'} + U_2^{'} < U_1 + U_2$,
We can get $\beta_1 \bigoplus \beta_2 \prec \alpha_1 \bigoplus \alpha_2$ by applying Equations (4) and (5). Note that $\beta_1 \bigoplus \beta_2 \in S[Q_1, Q_2]$, then policy $p = \alpha_1 \bigoplus \alpha_2$ is not in F^2, and thus (8) holds. For the second case, the policy $\beta_1(R_1^{'}, U_1^{'}) \prec \alpha_1(R_1, U_1)$, $R_1^{'} * R_2^{'} < R_1 * R_2^{'}$, $U_1^{'} + U_2^{'} < U_1 + U_2^{'}$, that is $\beta_1 \bigoplus \alpha_2 \prec \alpha_1 \bigoplus \alpha_2$, similarly, policy p is not in F^2, thus (8) holds. To summarise, $F^2 \subset F_1 \bigoplus F_2$.

For (9), by the above deduction, for any policy p that is not in $F^{n-1} \bigoplus F_n$, it must not in F^n by applying Equations (6) and (7), then, we have $F^n \subset F^{n-1} \bigoplus F_n$.

□

By Lemma 1, we devise an effective approach to solve the policy generation problem for large l. The idea is that 1) counting up the tuples of every attribute respectively; 2) generating policies for each attribute, and calculating their risk and utility cost; 3) making skyline computation over alternative policy set for every attribute to get a new policy space; 4) synthesizing new policies with length $l = (r_1 - 1 + r_2 - 1 + \ldots + r_n - 1)$ by using the policy space generated above, and calculate their risk and utility cost for skyline computation.

Table 3 illustrates the benefits from this way based on independent property. The expected skyline cardinality of generated points in d dimensions is around $m = O((\ln n)^{d-1})$ according to the analysis of [39]. $|G_1| \ll |G_2|$, when $r_i > 2$, $i = 1, 2, 3, \ldots, n$. Here G_1 is the alternative policy set for our approach and G_2 is the alternative policy set for traditional approach.

TABLE 3: Two ways of policy generation

Attribute	Q_1	$\ldots$	Q_i	$\ldots$	Q_n
Domain	r_1	$\ldots$	r_i	$\ldots$	r_n
Policy	2^{r_1-1}	$\ldots$	2^{r_i-1}	$\ldots$	2^{r_n-1}
$\|F\|$	$O((r_1 - 1)\ln 2)$	$\ldots$	$O((r_i - 1)\ln 2)$	$\ldots$	$O((r_n - 1)\ln 2)$
$\|G_1\|$	$2^{r_1-1}+\ldots+2^{r_n-1}+O((r_1-1)\ln 2)+\ldots+O((r_n-1)\ln 2)$				
$\|G_2\|$	$2^{r_1-1} * \ldots * 2^{r_n-1}$				

5 Parallel Algorithms

This is the second module. Intuitively, we can deal with the problem in two steps. First, the risk and utility cost of policies from the universal policy set S is computed to obtain the policy space $\{S, (R, U)\}$. Second, the skyline over the policy space is calculated to get the policy skyline set, and the partition on the QI attributes as well as risk and utility cost are returned. Obviously, the first module usually generate a huge number of policies as the growth of bit-string length. We know that the skyline calculation has to do a large number of repeated comparisons, thus it is quite time consuming. So it is important to choose some good algorithms for skyline computation.

5.1 SKY-MIN-MR for Exact Scheme

It is worth notice that RDIP problem is a typical $2D$-skyline problem compared to the traditional skyline computation. According to the dimensionality property, Tao et al [38] proposed the minimal MapReduce algorithm, which contains a presort-based skyline algorithm that can achieve balanced space for data storage and bounded net-traffic, constant round for termination and optimal computation cost with appropriate sampling parameter setting $\rho = \frac{t}{|G|} \ln(|G|t)$. To the best of our knowledge, this is one of the best algorithms that can deal with $2D$-skyline computation problem over large datasets. So we propose our baseline parallel algorithm called SKY-MIN-MR by using this MapReduce based framework. Firstly, we enumerate all the policies from S to generate the policy space. Secondly, by adding the risk and utility calculation over policies in the map-shuffle phase, we do $2D$-skyline computation by the minimal MapReduce algorithm. We also provide experimental evaluation and formal analysis for RDIP. In particular, the parallel algorithm SKY-MIN-MR consists of the following four phases, and the pseudo code of SKY-MIN-MR is shown in Algorithm 1.

Algorithm 1 SKY-MIN-MR($\{S,(R,U)\}$,ρ)

Input: $\{S,(R,U)\}$: the policy space, ρ: the sampling parameter.
Output: F: the policy skyline frontier.

sample = Sampling($\{S,(R,U)\}$,ρ);
sky-patition = SKY-PATITION(sample,t);
Broadcast sky-partition;
Local-ST = L-SORT-MR($\{S,(R,U)\}$,sky-partition);
HashMap = HASHMAP-R.setup(Local-ST);
Broadcast HashMap;
F = G-SCREEN-R(Local-ST, HashMap);

Function G-SCREEN-MR(Local-ST, HashMap);
min_u_lowerKey = FindHashMap(key);
for each policy in list **do**
  if policy.U $<$ min_u_Key_lowerR and policy.U $<$ min_u_lowerKey **then**
    policy $\rightarrow F$;
    if policy.U $<$ min_u_Key_lowerR **then**
      min_u_Key_lowerR = policy.U;
return F;

Algorithm 2 SKY-F-MR($\{S,(R,U)\}$, ρ, ε)

Input: $\{S,(R,U)\}$: the policy space, ρ: the sampling parameter, ε: the precision.
Output: F: the policy skyline frontier.

$\{G,(R,U)\}$ = FilterPolicy-M($\{S,(R,U)\}$,ε);
F = SKY-MIN-MR($\{G,(R,U)\}$, ρ)

Function $\{G,(R,U)\}$ = FilterPolicy-M($\{S,(R,U)\}$,ε);
map(key = R,value = string);//string = R + U + policy
risk = key, utility = string.U;
flag = false; // policy is not in approximately dominated area
for each policy in G **do**
  a = risk*ε - policy.R, b = utility*ε - policy.U;
  if (a*b)$\geq$ 0 and (a+b)$>$0 **then**
    // a $\geq$ 0, b$\geq$ 0, a+b$\neq$0
    flag = true;// policy is in the approximately dominated area
    break;
if flag == false **then**
  policy $\rightarrow G$;
  output(key,value);
return $\{G,(R,U)\}$;

- Sky-partition phase: To achieve balanced work load for all participating nodes, a new function Sky-partition which decides the partition strategy is proposed. We build sky-partition with t-1 samples from the set of ordered samples that is received by all nodes for further speed up, and these samples are chosen as the break points.
- Local sort phase: We partition the policies set S based on the regions divided by the sky-partition, and sort them locally in their region independently by using MapReduce, this phase was called L-SORT-MR.
- HashMap building phase: In this phase, the corresponding relation table HashMap of the minimum risk and utility cost about each node is created.
- Global screen phase: In the last phase, we judge if the policy belongs to the skyline set by using HashMap in MapReduce, which was called G-SCREEN-MR.

Algorithm 3 SKY-FILTER-MR($\{S,(R,U)\}$, ρ, ε, h)

Input: $\{S,(R,U)\}$: the policy space, ρ: the sampling parameter, ε: the precision, h: the depth of filter.
Output: F: the policy skyline frontier.

$\{G,(R,U)\}$ = FilterPolicy-M($\{S,(R,U)\}$, ε, h);
F = SKY-MIN-MR($\{G,(R,U)\}$, ρ)

Function $\{G,(R,U)\}$ = FilterPolicy-M($\{S,(R,U)\}$, ε, h);
map(key = R,value = string);//string = R + U + policy
risk = key, utility = string.U;
flag = false;// policy is not in the approximately dominated area
for each policy in G by inverse sequence **do**
  // reverse traversal
  a = risk*ε - policy.R, b = utility*ε - policy.U;
  depth = 0;// the depth of reverse traversal
  if (a*b)$\geq$ 0 and (a+b)$>$0 **then**
    // a $\geq$ 0, b$\geq$ 0, a+b$\neq$0
    depth = depth + 1;
    flag = true;//policy is in the approximately dominated area
    break;
  if depth $> h$ **then**
    break;
if flag == false **then**
  policy $\rightarrow G$;
  output(key,value);
return $\{G,(R,U)\}$;

Although we can output the accurate results by calculating all the policies from the universal set S, it is clear that the computation cost is exponentially large with the growth of l. We can guarantee the time cost for each machine is $O(l * 2^l/t)$ based on the analysis of [38].

5.2 SKY-F-MR for Approximation Scheme

We get some observations on the experiments of Algorithm 1: 1) Although the policy space $\{S,(R,U)\}$ is large (2^l), the size of its skyline set is small which is about $O(l \ln 2)$ [39]. This indicates that many policies from S are dominated by other policies. 2) The risk and utility cost for many policies are closed to each other by just a narrow margin. To make full use of the characteristics of policy distribution, we can filter data based on approximately domination. Obviously, the filtering power for approximately domination is larger than that for exact domination. Next, we propose an approximate skyline algorithm called SKY-F-MR for RDIP, and it was composed of five phases by adding the following filtering phase on the basis of SKY-MIN-MR. The pseudo code of SKY-F-MR is shown in Algorithm 2.

- Filtering phase: We insert policies into a new policy space $\{G,(R,U)\}$, if their risk and utility cost are neither in the dominated nor the approximately dominated area.

Fig.3 illustrates this statement: the SKY-MIN-MR inserts all the policies of $\{S,(R,U)\}$ into $\{G,(R,U)\}$, if their risk and utility costs do not fall into the dominated area. While, the SKY-F-MR only inserts those policies if their risk and utility costs do neither fall into the dominated region nor into the approximately dominated area. New policies of $\{S,(R,U)\}$ are still compared with all policies that are in

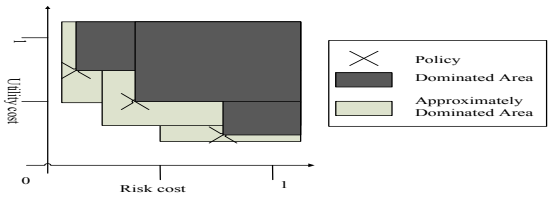


Fig. 3: Dominated versus approximatedly dominated area (with ε =1.5) in RU space.

the common dominating area. However, if no other policy approximately dominates the new one, it will be only inserted into $\{G,(R,U)\}$, which means that the SKY-F-MR tends to insert less policies than the SKY-MIN-MR. So the number of G is smaller and the time for skyline query over $\{G,(R,U)\}$ is less according to the analysis in [38].

5.3 SKY-FILTER-MR for Approximation Scheme

In order to speed up the filtering, two improvements are included in the stage of filtering in SKY-F-MR. We call the new algorithm as SKY-FILTER-MR. Firstly, a reverse traversal output set G is made since the policies are often dominated by the one related with it in non-dominated region. Secondly, the traversal depth h can be tuned to prevent the large overload of filtering. In fact, the bigger is the depth of traversal, the smaller is the average hit ratio of being approximately dominated. Usually, the most effective h is varying with different $\{S,(R,U)\}$ and we can test the effectiveness of h to achieve near-optimal filtering efficiency. The algorithm SKY-FILTER-MR consists of five phases by improving the filtering phase in SKY-F-MR. The pseudo code of SKY-FILTER-MR is shown in Algorithm 3.

- Filtering phase: We insert a policy into the policy space $\{G,(R,U)\}$, if the policy is not in the current subset of policy space $\{G,(R,U)\}$ (the approximately dominated area). Note that, this operation is carried out in reverse order.

Fig. 4 shows an example of data flow in SKY-FILTER-MR framework with ε = 1. It consists of policy generation and algorithm execution. Consider table D in Fig. 4(a), eight policies are generated as shown in Fig. 4(b). These policies are split into smaller subsets, and each map task handles one split on the node by computing their risk and utility cost. The results are denoted as $\{S,(R,U)\}$ in Fig. 4(c). To avoid too much skyline computation, we add the filter in map phase of the second round, which is denoted as FilterPolicy-M. The detailed description was given from line 3 to line 21 in Algorithm 3. Next, we get $\{G,(R,U)\}$ as shown in Fig. 4(d) with approximate skyline. The sampling parameter ρ equals to 0.75 when the reducer task is set to 2. Suppose one sample is $\{001; 010; 111\}$, then it becomes $\{010; 001; 111\}$ after sorting, and its break point 001 will be recorded in partition file. In Shuffle stage of the third round, the policies are partitioned into two blocks $\{000; 100; 010\}$ and $\{001; 101; 011; 111\}$ based on break point 001. In Reduce stage, the key for each tuple is set to the minimum risk cost of policies in each block. That is, the key for 000, 100 and 010 is 0.021; while the key for 001, 101, 011 and 111 is 0.111. The outputs are shown in Fig. 4(f). In the fourth round, the Map stage is taken in default manner. While in the Reduce stage, a table containing the key and the minimum utility cost for every block is created, that is $\{0.021; 0.1213\}$ for the first block and $\{0.111; 0\}$ for the second block, and we denote this table as HashMap. Finally, all the tuples are checked as shown in Fig. 4(h), we use $min\_u\_lowerKey$ to denote the minimum utility cost of those tuples whose key is lower than the current block, and $min\_u\_Key\_lowerR$ to denote the minimum utility cost of those tuples whose risk is lower than the current tuple, and we take those policies whose utility is smaller than both of them as skyline results, Fig. 4(g) shows the final skyline set.

The following theorem shows that the SKY-FILTER-MR guarantees to generate near-optimal policies.

Lemma 2. The SKY-FILTER-MR algorithm with precision ε generates an ε-approximate skyline frontier.

Proof: Let F^* and F denote the skyline frontier and the ε-approximate skyline frontier respectively.

First, the strictly dominated relationship has transitivity property. That is, $\alpha \prec \beta$ and $\beta \prec \gamma$, $\alpha \prec \gamma$ is obtained. However, the approximately domination relationship has no such property.

Second, the SKY-FILTER-MR algorithm consists of two steps. The first step is FilterPolicy-M and each node filters the policies by approximately dominated relationship, and each policy has only one chance to be filtered. Although the approximately dominated relationship cannot be transmitted, we make an approximately skyline over policies only once and no precision error is accumulated. The second step is SKY-MIN-MR and we make an exact skyline over policies with no transmission error. Thus, when the precision is ε, the SKY-FILTER-MR algorithm generates an ε-approximate skyline frontier.

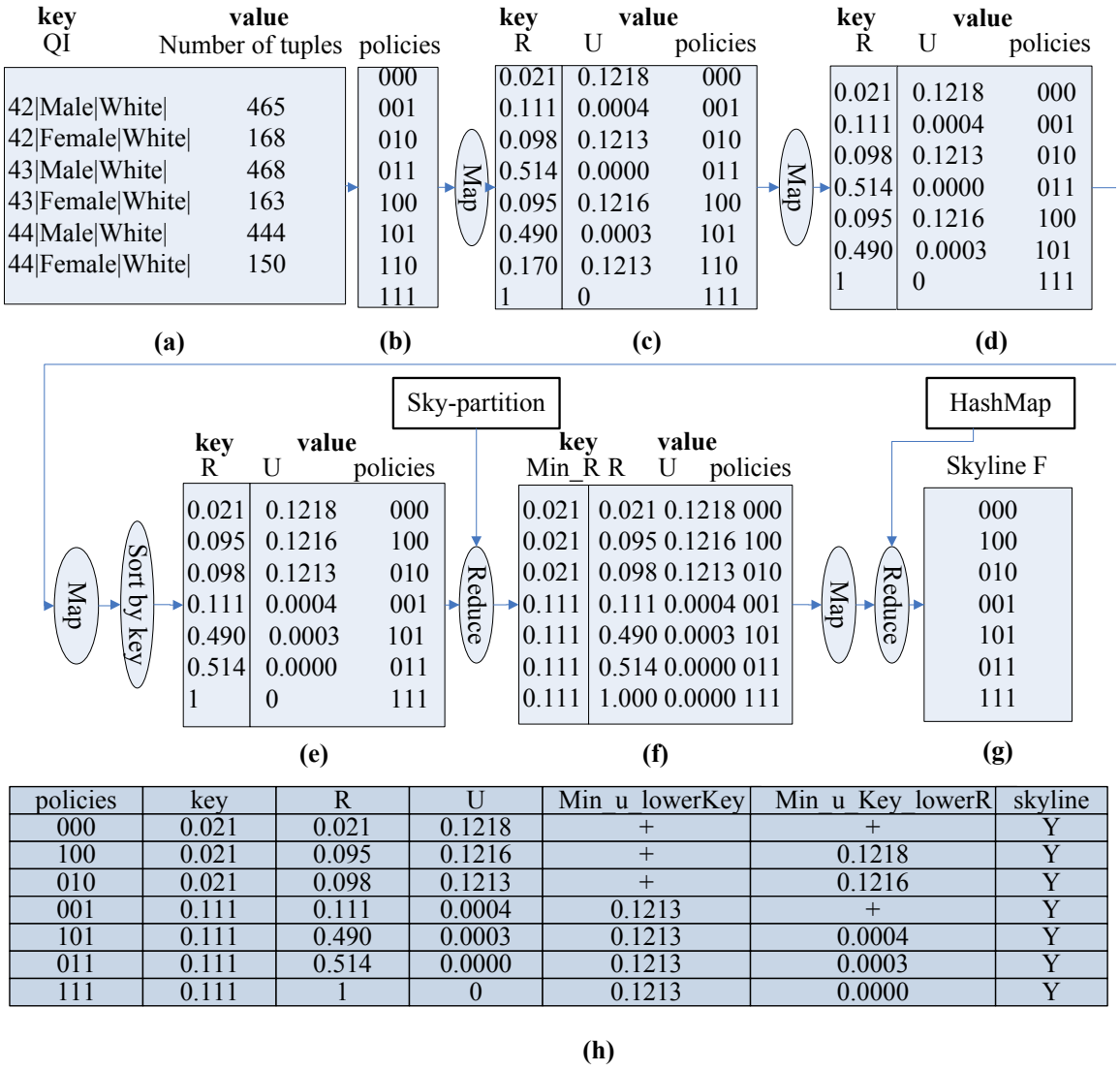
Finally, we give the detailed proof. Each policy p^* in F^* has two states in our algorithm. In the first case, p^* is in F and $p^* \preceq_\varepsilon p^*$, thus Lemma 2 holds. In the second case, p^* is not in F, if p^* is dominated or approximately dominated by p in F, then $p \preceq_\varepsilon p^*$ is obtained, thus Lemma 2 holds. Next, we discuss the condition that p is not in F. 1) If p^* is dominated by p which is not in F, there is a policy p_1 dominates or approximately dominates p. That is $p \preceq p^*$, $p_1 \preceq_\varepsilon p$ or $p_1 \preceq p$, we can get $p_1 \preceq_\varepsilon p^*$, clear Lemma 2 holds. 2) If p^* is approximately dominated by p which is not in F (the first step), there must be one policy p_1 dominates p, that is we have $p \preceq_\varepsilon p^*$, $p_1 \preceq p$, which gets $p_1 \preceq_\varepsilon p^*$, thus Lemma 2 holds.

□

6 Experiments

In this section, we evaluate the performance of the proposed algorithms on an in-house cluster with one master and 9 slave nodes, each of which has two Inter Xeon(R) E5-2670 2.60GHZ processors with 8 cores, 128GB memory, and installed 64-bit Linux operating system. We implement all algorithms on Hadoop (version 1.2.1), and set the block size of distributed file system to 64MB.

Dataset. In the experiments, we use the Adult dataset from the UCI Machine Learning Repository [3] and the 2009 Census microdata extracted from IPUMS USA [40]. For



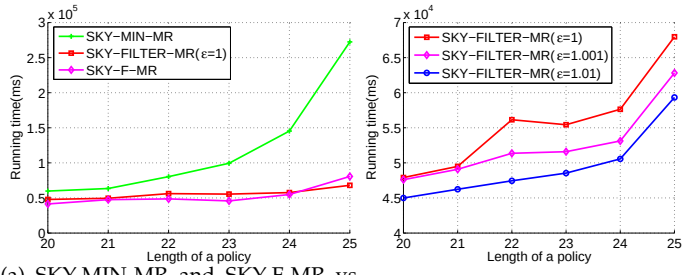
policies	key	R	U	Min_u_lowerKey	Min_u_Key_lowerR	skyline
000	0.021	0.021	0.1218	+	+	Y
100	0.021	0.095	0.1216	+	0.1218	Y
010	0.021	0.098	0.1213	+	0.1216	Y
001	0.111	0.111	0.0004	0.1213	+	Y
101	0.111	0.490	0.0003	0.1213	0.0004	Y
011	0.111	0.514	0.0000	0.1213	0.0003	Y
111	0.111	1	0	0.1213	0.0000	Y

(h)

Fig. 4: An example of data flow in SKY-FILTER-MR framework with $\varepsilon = 1$

Adult, this dataset consists of 32561 tuples without missing values and we retain four fields {Age, Gender, Race, Occupation}. Specifically, we use the demographics {Age, Gender, Race} as the QI attributes and the field {Occupation} as sensitive attribute. In order to have comparison with the ideal area which is the dominated area of k-anonymity, l-diversity and t-closeness, we append one additional attribute, the 5-digit ZIP codes which is generated randomly based on uniform distribution for Tennessee's, to the Adult dataset. In this synthetic dataset, {Age, Gender, Race, Zip} are regarded as the QI attriutes. For Census, this dataset consists of 100,000 tuples and four fields {Age, Gender, Race, Relation} are retained. Thus, the demographics {Age, Gender, Race} are regarded as the QI attributes and the field {Relation} is regarded as the sensitive attribute. Similarly, in order to make comparison with common privacy models, a synthetic dataset based on Census is generated by adding ZIP code and {Age, Gender, Race, Zip} are used as the QI attributes. We set the depth of traversal h = 200 for real datasets and h = 100 in synthetic datasets.

Efficiency. Here we present the average running time

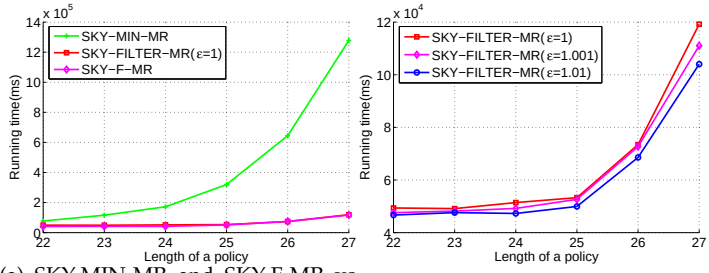


(a) SKY-MIN-MR and SKY-F-MR vs SKY-FILTER-MR

(b) SKY-FILTER-MR vs.ε

Fig. 5: Running time for different l in AD

of the SKY-MIN-MR (the baseline algorithm), SKY-F-MR and SKY-FILTER-MR for different length of policies, and the size ($|G|$) of alternative policy set G of the SKY-MIN-MR and SKY-FILTER-MR algorithms were also reported with varying l. For Adult dataset, we set l from 20 to 25. Different l indicates that the domain of attribute Age is different. The domain for Race and Gender are {White,



(a) SKY-MIN-MR and SKY-F-MR vs SKY-FILTER-MR (b) SKY-FILTER-MR vs.ε

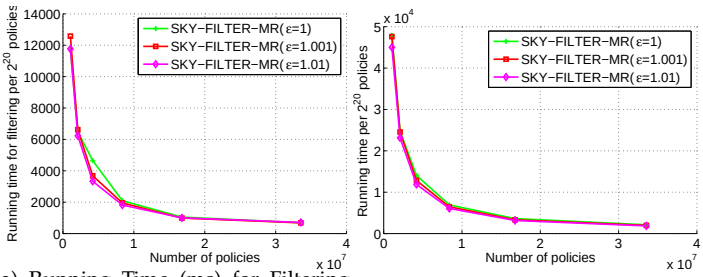
Fig. 6: Running time for different l in CD

TABLE 4: $|G|$ of SKY-MIN-MR (-M-) and SKY-FILTER-MR ($\varepsilon = 1, 1.001, 1.01$) in AD

$\|G\|\backslash\ l$	**20**	**21**	**22**	**23**	**24**	**25**
-M-	2^{20}	2^{21}	2^{22}	2^{23}	2^{24}	2^{25}
ε=1	60739	89673	182280	277057	534294	1216159
1.001	21660	35202	63022	81718	145363	340095
1.01	1432	3525	8199	11361	37530	61771

TABLE 5: $|G|$ of SKY-MIN-MR (-M-) and SKY-FILTER-MR ($\varepsilon = 1, 1.001, 1.01$) in CD

$\|G\|\backslash\ l$	**22**	**23**	**24**	**25**	**26**	**27**
-M-	2^{22}	2^{23}	2^{24}	2^{25}	2^{26}	2^{27}
ε=1	12953	54450	196988	402692	674820	1384353
1.001	6520	20473	86350	223051	393766	873381
1.01	2801	5801	14983	31918	56453	160017



(a) Running Time (ms) for Filtering vs. $|S|$ (b) Total Running Time (ms) vs. $|S|$

Fig. 7: Scalability of the SKY-FILTER-MR algorithm in AD

Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black} and {Female, Male} respectively, and we denote this dataset as AD for convenience. While, l varies from 22 to 27 for Census dataset, the domain for Race is {White, Black, American Indian, Chinese, Japanse, other Asian or Pacific, Other,}, and we denote this dataset as CD. The results for AD are shown in Fig. 5 and Table 4 respectively. It is clear that SKY-F-MR algorithm and SKY-FILTER-MR algorithm are significantly faster than the baseline algorithm, and our improved algorithm is slightly more efficient than the SKY-F-MR algorithm with the increasing length (l) of policies. Meanwhile, we can see from Fig. 5(b) that the improved algorithm with ε = 1.001 is slightly more efficient than that with ε = 1, and the algorithm with ε = 1.01 is slightly more efficient than that with ε = 1.001. For example, when l equals to 25 (Fig. 5), we can see that our proposed algorithms with ε = 1, ε = 1.001 and ε = 1.01 are 4.0076, 4.3365 and 4.5912 times faster than the baseline algorithm on average. Meanwhile, the efficiency gap between SKY-FILTER-MR and the baseline algorithm is getting larger with the growth of l. For example, when l equals to 25 and 20 (Fig. 5), our algorithms with ε = 1 are 4.0076 and 1.2480 times faster than the baseline algorithm. It means that our proposed algorithms are significantly faster than the baseline algorithm for long policies. In addition, the size of G for SKY-FILTER-MR is much smaller than that for the baseline algorithm, and it decreases dramatically with the reducing of ε. Our algorithm outperformed the baseline approach with the number of alternative policies decreased up to 732 times in the best case. As expected, the running time and $|G|$ of all the algorithms were increased with growing l. The experimental results for CD are similar as shown in Fig. 6 and Table 5.

Scalability. Here we test the scalability of SKY-FILTER-MR algorithm. We vary the total number of polices $|S|$ from 2^{20} to 2^{25} in AD and from 2^{22} to 2^{27} in CD, and test the time for filtering and total running time of our algorithm respectively. The precision parameter ε was set to 1, 1.001 and 1.01. The results are shown in Fig. 7 and Fig. 8. From Fig. 7, we can see that our proposed algorithms scale quite well. When the size of S is larger than 1048576, the increasing speed of S is higher than the increasing speed of the running time. The results further confirmed that our algorithms can be applied to handle very large datasets.

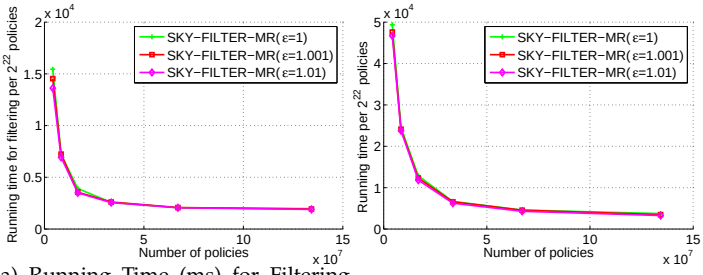
TABLE 6: Running time and $|G|$ vs.ε (for the SKY-FILTER-MR algorithm in synthetic dataset based on Adult)

ε	**1**	**1.001**	**1.005**	**1.01**	**1.05**
time(ms)	46413	45021.1	44094	10984	9637
$\|G\|$	6739	4090	2639	1394	338

TABLE 7: Running time and $|G|$ vs.ε (for the SKY-FILTER-MR algorithm in synthetic dataset based on Census)

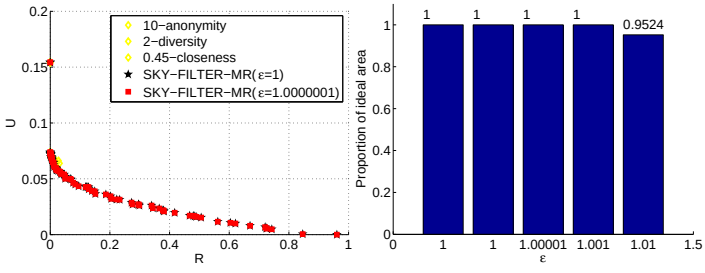
ε	**1**	**1.0000001**	**1.00001**	**1.001**	**1.01**
time(ms)	44612	42496	42515	13570	9553
$\|G\|$	7060	1706	1630	880	352

Effects of parameter ε. Here we study how the parameter ε affects the efficiency and effectiveness of the SKY-FILTER-MR algorithm. For this purpose, we set ε = 1, 1.001, 1.005, 1.01, 1.05 for synthetic dataset based on Adult, and ε = 1, 1.0000001, 1.00001, 1.001, 1.01 for synthetic dataset based on Census, and use the running time, the size $|G|$ of alternative policy set G and the proportion of ideal area to evaluate the efficiency and effectiveness of our algorithm, respectively. The dominating area was composed of those policies who dominate 10-anonymity, 2-diversity, and 0.45-closeness which are commonly adopted privacy protection levels [11], and we denote this dominating area as ideal area. For three privacy models, we utilize Incognito algorithm [41] to find the skyline set. Also, we set l = 20, and similar results can be observed for other values. We test the SKY-FILTER-MR algorithm in the synthetic dataset. The results of running time with varying ε are shown in Table 6 and Table 7. As we can see, the running time decreases with increasing ε. In Table 6, when ε =1.01 or 1.05, the running time of the algorithm is much more smaller than that with ε = 1.005. However, when $\varepsilon \leq 1.005$, the running time smoothly decreasing with increasing ε. This is because the filter stage



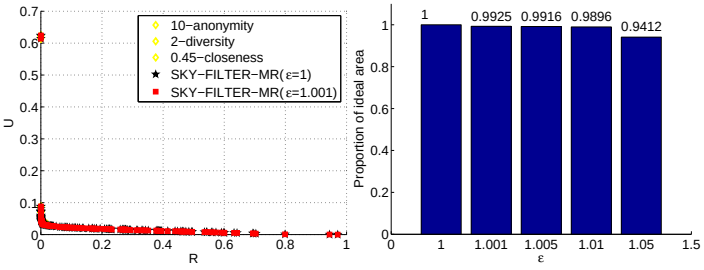
(a) Running Time (ms) for Filtering vs. $|S|$ (b) Total Running Time (ms) vs. $|S|$

Fig. 8: Scalability of the SKY-FILTER-MR algorithm in CD



(a) R-U Space (b) Proportion of Ideal Area vs. ε

Fig. 10: Effectiveness of the SKY-FILTER-MR algorithm in synthetic dataset based on Census



(a) R-U Space (b) Proportion of Ideal Area vs. ε

Fig. 9: Effectiveness of the SKY-FILTER-MR algorithm in synthetic dataset based on Adult

takes relatively long time to get space $\{G, (R, U)\}$ when ε is very small. However, when ε is large and $|G|$ is small, the computation can be done with centralized algorithm ($G < 1500$) to decrease the time cost of skyline calculation. Similar conclusions can be obtained in Table 7.

We report the proportion of ideal area of the SKY-FILTER-MR algorithm with different ε in Fig. 9 and Fig. 10. First, it is intuitively plausible (Fig. 9(a) and Fig. 10(a)) that policies generated from SKY-FILTER-MR dominate the policies from k-anonymity, l-diversity and t-closeness, that is, SKY-FILTER-MR dominates the ideal area. Second, we make a quantitative analysis of dominant condition about our algorithm. From Fig. 9(b), we can find that the proportion of ideal area decreases with increasing ε. The reason is that ε controls the quality of skyline frontier policies and a large ε results in a low quality. However, we get relatively high quality with different ε. For instance, when ε = 1.001 (Fig. 9(a)), the skyline frontier of SKY-FILTER-MR algorithm is close to the exact skyline frontier F.

7 Conclusions

We study the recommendation on a great number of de-identification policies using MapReduce. Firstly, we put forward an effective way of policy generation on the basis of newly proposed definition, which can decreases the time of generating policies and the size of alternative policy set dramatically. Secondly, we propose SKY-FILTER-MR, which is a three-round MapReduce-based parallel algorithm, to answer skyline de-identification policies efficiently. We use bit-strings to represent one policy in the framework. In order to further improve the performance, a vigorous approximate skyline scheme is proposed to decrease the number of alternative policy set. By integrating the approximate skyline with the minimal MapReduce algorithm, the filtering power in the Map phase of first round was optimized without increasing the transmission cost. We perform comprehensive experimental evaluation on both real-world and synthetic datasets, and the results indicate good performance and scalability of our proposed SKY-FILTER-MR.

There are several aspects could be improved in the future. For example, our framework requires a complete ordering in the domain of all QI attributes. In order to compare with original algorithms, the domain of attribute which is unordered in original state must be ordered randomly. We plan to make more efforts to design an effective approach to deal with the unordered attributes.

Acknowledgments

This work is supported by the NSFC under grant 61472148. The authors would like to thank the editor and reviewers for their insightful comments.

References

[1] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Comput. Surv.*, vol. 42, no. 4, pp. 14:1–14:53, 2010.

[2] X. MA, H. Li, J. Ma, Q. Jiang, S. Gao, N. Xi, and D. Lu, "Applet: A privacy-preserving framework for location-aware recommender system," *Sci China Inf Sci*, vol. 59, no. 2, pp. 1–15, 2016.

[3] W. Xia, R. Heatherly, X. Ding, J. Li, and B. Malin, "Efficient discovery of de-identification policies through a risk-utility frontier," in *CODASPY*, 2013, pp. 59–70.

[4] K. Benitez, G. Loukides, and B. Malin, "Beyond safe harbor: Automatic discovery of health information de-identification policy alternatives," in *IHI*, 2010, pp. 163–172.

[5] K. E. Emam, "Heuristics for de-identifying health data," *IEEE Security and Privacy*, vol. 6, no. 4, pp. 58–61, 2008.

[6] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 555–570, 2002.

[7] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "ℓ-diversity: Privacy beyond k-anonymity," in *TKDD*, 2007, pp. 1–52.

[8] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and ℓ-diversity," in *ICDE*, 2007, pp. 106–115.

[9] J. Brickell and V. Shmatikov, "The cost of privacy: Destruction of data-mining utility in anonymized data publishing," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 70–78.

[10] J. Cao and P. Karras, "Publishing microdata with a robust privacy guarantee," *Proc. VLDB Endow.*, vol. 5, no. 11, pp. 1388–1399, 2012.

[11] W. Xia, R. Heatherly, X. Ding, J. Li, and B. A. Malin, "Ru policy frontiers for health data de-identification," *Journal of the American Medical Informatics Association*, vol. 22, no. 5, pp. 1029–1041, 2015.

[12] W. Qardaji, W. Yang, and N. Li, "Priview: Practical differentially private release of marginal contingency tables," in *SIGMOD*, 2014, pp. 1435–1446.
[13] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Privbayes: Private data release via bayesian networks," in *SIGMOD*, 2014, pp. 1423–1434.
[14] T. Rekatsinas, A. Deshpande, and A. Machanavajjhala, "Sparsi: Partitioning sensitive data amongst multiple adversaries," *Proc. VLDB Endow.*, vol. 6, no. 13, pp. 1594–1605, 2013.
[15] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern, "Worst-case background knowledge for privacy-preserving data publishing." in *ICDE*, 2007, pp. 126–135.
[16] W. K. Wong, N. Mamoulis, and D. W. L. Cheung, "Non-homogeneous generalization in privacy preserving data publishing," in *SIGMOD*, 2010, pp. 747–758.
[17] B. Fung, K. Wang, and P. S. Yu, "Top-down specialization for information and privacy preservation," in *ICDE*, 2005, pp. 205–216.
[18] R. J. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *ICDE*, 2005, pp. 217–228.
[19] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *International Conference on Data Engineering*, 2006, pp. 25–25.
[20] M. E. Nergiz, C. Clifton, and A. E. Nergiz, "Multirelational k-anonymity." in *ICDE*, 2007, pp. 1417–1421.
[21] H. Park and K. Shim, "Approximate algorithms for k-anonymity," in *SIGMOD*, 2007, pp. 67–78.
[22] X. Xiao and Y. Tao, "Anatomy: Simple and effective privacy preservation," in *Proceedings of the 32Nd International Conference on Very Large Data Bases*, 2006, pp. 139–150.
[23] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vision*, vol. 40, no. 2, pp. 99–121, 2000.
[24] M. E. Nergiz, M. Atzori, and C. Clifton, "Hiding the presence of individuals from shared databases," in *SIGMOD*, 2007, pp. 665–676.
[25] C. Li, M. Hay, G. Miklau, and Y. Wang, "A data- and workload-aware algorithm for range queries under differential privacy," *Proc. VLDB Endow.*, vol. 7, no. 5, pp. 341–352, 2014.
[26] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.
[27] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.
[28] G. Cormode, C. M. Procopiuc, E. Shen, D. Srivastava, and T. Yu, "Empirical privacy and empirical utility of anonymized data," in *Data Engineering Workshops (ICDEW)*, 2013, pp. 77–82.
[29] P. Samarati, "Protecting respondents identities in microdata release," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 13, no. 6, pp. 1010–1027, 2001.
[30] X. Xiao and Y. Tao, "Personalized privacy preservation," in *SIGMOD*, 2006, pp. 229–240.
[31] S. Börzsönyi, D. Kossmann, and K. Stocker, "The skyline operator," in *Proceedings of the 17th International Conference on Data Engineering*, 2001, pp. 421–430.
[32] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, "Skyline with presorting." in *ICDE*, 2003, pp. 717–719.
[33] K.-L. Tan, P.-K. Eng, and B. C. Ooi, "Efficient progressive skyline computation," in *Proceedings of the 27th International Conference on Very Large Data Bases*, 2001, pp. 301–310.
[34] I. Bartolini, P. Ciaccia, and M. Patella, "Efficient sort-based skyline evaluation," *ACM Trans. Database Syst.*, vol. 33, no. 4, pp. 31:1–31:49, 2008.
[35] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "An optimal and progressive algorithm for skyline queries," in *SIGMOD*, 2003, pp. 467–478.
[36] J. D. West, I. Wesley-Smith, and C. T. Bergstrom, "A recommendation system based on hierarchical clustering of an article-level citation network," *IEEE Transactions on Big Data*, vol. 2, no. 2, pp. 113–123, 2016.
[37] Y. Park, J.-K. Min, and K. Shim, "Parallel computation of skyline and reverse skyline queries using mapreduce," *Proc. VLDB Endow.*, vol. 6, no. 14, pp. 2002–2013, 2013.
[38] Y. Tao, W. Lin, and X. Xiao, "Minimal mapreduce algorithms," in *SIGMOD*, 2013, pp. 529–540.
[39] J. L. Bentley, H. T. Kung, M. Schkolnick, and C. D. Thompson, "On the average number of maxima in a set of vectors and applications," *J. ACM*, vol. 25, no. 4, pp. 536–543, 1978.
[40] S. Ruggles, J. T. Alexander, K. Genadek, R. Goeken, M. B. Schroeder, and M. Sobek, "Integrated public use microdata series: Version 5.0," *University of Minnesota,Minneapolis*, 2010.
[41] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient full-domain k-anonymity," in *SIGMOD*, 2005, pp. 49–60.

Xiaofeng Ding is currently working as an associate professor in the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. He received his PhD degree in Computer Science from Huazhong University of Science and Technology, Wuhan, China in 2009. His research interests mainly include data privacy and query processing, spatial databases and graph databases.

Li Wang is currently a Master student at the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. She is a member of the the Services Computing Technology and System Lab, Cluster and Grid Computing Lab. Her research interests include privacy and query processing in big data.

Zhiyuan shao is an associate professor of school of computer science and technology, Huazhong University of Science and Technology (HUST), Wuhan, China. He received Master and Ph.D. degrees at HUST in 2000 and 2005, respectively. His research interests cover cluster computing, high performance computing, and graph computing.

Hai Jin received the PhD degree in computer engineering in 1994 from Huazhong University of Science and Technology (HUST), China, where he is currently the Cheung Kong professor of the School of Computer Science and Technology. In 1996, he was awarded a German Academic Exchange Service Fellowship to visit the Technical University of Chemnitz in Germany. He worked at the University of Hong Kong between 1998 and 2000, and as a visiting scholar at the University of Southern California between 1999 and 2000. He was awarded the Distinguished Young Scholar Award from the National Science Foundation of China in 2001. He is the chief scientist of ChinaGrid, the largest grid computing project in China. He is the member of Grid Forum Steering Group (GFSG). He has coauthored 15 books and published more than 500 research papers. His research interests include distributed computing, computer architecture, virtualization technology, cluster computing and grid computing, big data privacy and security, crowdsourcing, network storage, and network security. He is a senior member of the IEEE and a member of the ACM.