L-Diversity Based Dynamic Update For Large Time-evolving Microdata

Xiaoxun Sun[1] Hua Wang[1] and Jiuyong Li[2]

[1] Department of Mathematics & Computing
University of Southern Queensland, QLD, Australia
Email: {sunx, wang}@usq.edu.au
[2] School of Computer and Information Science
University of South Australia, Adelaide, Australia
Email: jiuyong.li@unisa.edu.au

Abstract. Data anonymization techniques based on enhanced privacy principles have been the focus of intense research in the last few years. All existing methods achieving privacy principles assume implicitly that the data objects to be anonymized are given once and fixed, which makes it unsuitable for time evolving data. However, in many applications, the real world data sources are dynamic. In such dynamic environments, the current techniques may suffer from poor data quality and/or vulnerability to inference. In this paper, we investigate the problem of updating large time-evolving microdata based on the sophisticated l-diversity model, in which it requires that every group of indistinguishable records contains at least l distinct sensitive attribute values; thereby the risk of attribute disclosure is kept under $1/l$. We analyze how to maintain the l-diversity against time evolving updating. The experimental results show that the updating technique is very efficient in terms of effectiveness and data quality.

1 Introduction

Many organizations are increasingly publishing microdata (tables that contain unaggregated information about individuals). These tables can include medical, voter registration, census, and customer data. Some of these microdata need to be released, for various purposes, to other parties in a modified form (without the direct identifying information such as SSN, Name, etc.). But even altered this way, these datasets could still present vulnerabilities that can be exploited by intruders, i.e. persons whose goals are to identify specific individuals and to use the confidential information they discover for malicious purposes. The high volume and availability of released datasets together with ever increasing computational power made the protection against those vulnerabilities an increasingly difficult task. To avoid linking attacks, Samarati and Sweeney [11, 15] proposed a definition of privacy called k-anonymity. A table satisfies k-anonymity if every record in the table is indistinguishable from at least $k-1$ other records with respect to every set of quasi-identifier attributes; such a table is called a k-anonymous table.

Due to its conceptual simplicity, numerous algorithms have been proposed for implementing k-anonymity via generalization and suppression. Samarati [11] presents an algorithm that exploits a binary search on the domain generalization hierarchy to find minimal k-anonymous table. Sun *et al.* [12] recently improve his algorithm by integrating the hash-based technique. Bayardo and Agrawal [3] presents an optimal algorithm that starts from a fully generalized table and specializes the dataset in a minimal k-anonymous table, exploiting ad hoc pruning techniques. LeFevre *et al.* [6] describes an algorithm that uses a bottom-up technique and a priori computation. Fung *et al.* [5] present a top-down heuristic to make a table to be released k-anonymous. As to the theoretical results, Meyerson and Williams [9] and Aggarwal *et al.* [1, 2] proved the optimal k-anonymity is NP-hard (based on the number of cells and number of attributes that are generalized and suppressed) and describe approximation algorithms for optimal k-anonymity. Sun *et al.* [13] proved that k-anonymity problem is also NP-hard even in the restricted cases, which could imply the results in [1, 2, 9] as well.

Recent studies shows that although k-anonymity protects against identity disclosure, it is insufficient to prevent attribute disclosure. To address this limitation of k-anonymity, several models such as p-sensitive k-anonymity [16], (p^+, α)-sensitive k-anonymity [14], l-diversity [8], (α, k)-anonymity [19] and t-closeness [7] were proposed in the literature in order to deal with the problem of k-anonymity. The work presented in this paper is based on l-diversity model, introduce by [8]. The main contribution of [8] is to introduce the l-diversity property, which provides privacy even when the data publisher does not know what kind of knowledge is possessed by the adversary. Most of the existing solutions are limited only to static data release. That is, in such solutions it is assumed that the entire dataset is available at the time of release. Nevertheless, large microdata sets containing private information are time-evolving, meaning that new data are collected and added, and old data are purged.

One possible method is to publish anonymizations of current microdata, that is, when the new anonymous versions of such a dataset are prepared for release, the current solution is to reprocess the entire dataset, without relying on previous releases of the dataset. However, processing a large dataset in this way to achieve the privacy requirement is timeconsuming. Another approach is to anonymize and publish new records periodically. Then researchers can either study each released dataset independently or merge multiple datasets together for more comprehensive analysis. Although straightforward, this approach may suffer from severely low data quality.

The incremental updates are not well addressed in the previous studies. [17] studies the incremental update issue for k-anonymity model. In this paper, we discuss about the updating technique for large time-evolving microdata on l-diversity model which extend the results in [17]. we propose an updating technique for the maintenance of l-diverse large evolving datasets. Essentially, the proposed technique produces a l-diverse dataset starting from a previous l-diverse release solution for the dataset, which is updated to include the new data in the increment dataset and to delete the obsolete dataset. The anonymous process tries to minimize information loss. As our experimental results show, this updating technique is far more efficient than to re-process the whole updated microdata.

2 Preliminaries

Let T be the initial microdata table and T' be the released microdata table. T' consists of a set of tuples over an attribute set. The attributes characterizing microdata are classified into the following three categories.

- *Identifier attributes* that can be used to identify a record such as Name and Medicare card.
- *Quasi-identifier (QI) attributes* that may be known by an intruder, such as Zip code and Age. QI attributes are presented in the released microdata table T' as well as in the initial microdata table T.
- *Sensitive attributes* that are assumed to be unknown to an intruder and need to be protected, such as Disease or ICD9Code. Sensitive attributes are presented both in T and T'.

In what follows we assume that the identifier attributes have been removed and the quasi-identifier and sensitive attributes are usually kept in the released and initial microdata table. Another assumption is that the value for the sensitive attributes are not available from any external source. This assumption guarantees that an intruder can not use the sensitive attributes to increase the chances of disclosure. Unfortunately, an intruder may use record linkage techniques [18] between quasi-identifier attributes and external available information to glean the identity of individuals from the modified microdata. To avoid this possibility of privacy disclosure, one frequently used solution is to modify the initial microdata,

MCN	Gender	Age	Zip	Diseases
*	Male	25	4350	Hypertension
*	Male	23	4351	Hypertension
*	Male	22	4352	Depression
*	Female	28	4353	Chest Pain
*	Female	34	4352	Obesity
*	Female	31	4350	Flu

Table 1: Microdata

MCN	Gender	Age	Zip	Diseases
*	Male	22-25	435*	Hypertension
*	Male	22-25	435*	Hypertension
*	Male	22-25	435*	Depression
*	Female	28-34	435*	Chest Pain
*	Female	28-34	435*	Obesity
*	Female	28-34	435*	Flu

Table 2: 3-anonymous Microdata

more specifically the quasi-identifier attributes values, in order to enforce the k-anonymity property.

Definition 1 (Quasi-identifier). *A quasi-identifier (QI) is a minimal set Q of attributes in microdata table T that can be joined with external information to re-identify individual records (with sufficiently high probability*

Definition 2 (k-anonymity). *The modified microdata table T' is said to satisfy k-anonymity if and only if each combination of quasi-identifier attributes in T' occurs at least k times.*

A QI-group in the modified microdata T' is the set of all records in the table containing identical values for the QI attributes. There is no consensus in the literature over the term used to denote a QI-group. This term was not defined when k-anonymity was introduced [11, 15]. More recent papers use different terminologies such as equivalence class [19, 8, 7] and QI-cluster [16, 14].

For example, let the set {Gender, Age, Zip Code} be the quasi-identifier of Table 1. Table 2 is one 3-anonymous view of Table 1 since there are two QI-groups and the size of each QI-group is at least 3. So k-anonymity can ensure that even though an intruder knows a particular individual is in the k-anonymous microdata table T, s/he can not infer which record in T corresponds to the individual with a probability greater than $1/k$.

The k-anonymity property ensures protection against identity disclosure, i.e. the identification of an entity (person, institution). However, it does not protect the data against attribute disclosure. To deal with this problem in privacy breach, the l-diversity model was introduced in [8].

Definition 3 (l-diversity principle). *A QI-group is said to have l-diversity if there are at least l "well-represented" values for the sensitive* *attribute. A modified table is said to have l-diversity if every QI-group of the table has l-diversity.*

Machanavajjhala *et al.* [8] gave a number of interpretations of the term "well-represented" in this principle:

1. Distinct l-diversity: The simplest understanding of "well represented" would be to ensure there are at least l distinct values for the sensitive attribute in each QI-group. Distinct l-diversity does not prevent probabilistic inference attacks. A QI-group may have one value appear much more frequently than other values, enabling an adversary to conclude that an entity in the equivalence class is very likely to have that value. This motivated the development of the following two stronger notions of l-diversity.

2. Entropy l-diversity: The entropy of a QI-group G is defined to be:

$$Entropy(G) = -\sum_{s \in S} p(G,s) \mathrm{log} p(G,s)$$

in which S is the set of the sensitive attribute, and $p(G,s)$ is the fraction of records in G that have sensitive value s. A table is said to have entropy l-diversity if for every QI-group G, $Entropy(G) \geq log(l)$. Entropy l- diversity is strong than distinct l-diversity. As pointed out in [8], in order to have entropy l-diversity for each QI-group, the entropy of the entire table must be at least $log(l)$. Sometimes this may too restrictive, as the entropy of the entire table may be low if a few values are very common. This leads to the following less conservative notion of l-diversity.

3. Recursive (c,l)-diversity: Recursive (c,l)-diversity makes sure that the most frequent value does not appear too frequently, and the less frequent values do not appear too rarely. Let m be the number of values in a QI-group, and r_i, $1 \leq i \leq m$ be the number of times that the i^{th} most frequent sensitive value appears in a QI-group G. Then G is said to have recursive (c,l)-diversity if $r_1 < c(r_l + r_{l+1} + ... + r_m)$. A table is said to have recursive (c,l)-diversity if all of its QI-groups have recursive (c,l)-diversity.

In this paper, we adopt the first interpretation of l-diversity, that is, we say a microdata satisfies l-diversity principle, if there are at least l distinct values in each QI-group. We applied the cluster technique reported in [4]. To ensure that l-diversity is correctly enforced, two constraints are required when the clustering process is performed. First, each resulted cluster must have at least l distinct values for the sensitive attribute. If it does, the subsequent generalization of the cluster elements to a common tuple ensures the l-diversity requirement. Second, the clustering method must act towards minimizing the information loss. The clusters should be formed such that the information lost by generalizing each group of tuples to a common value will be as low as possible.

Definition 4. *[4] [Information Loss] Let $cl \in \mathcal{P}$ be a cluster, $gen(cl)$ its generalization information and $A = \{N_1, \cdots N_s, C_1, \cdots C_t\}$ the set of quasi identifier attributes. The information loss caused by generalizing cl tuples to $gen(cl)$ is:*

$$IL(cl) = |cl|(\sum_{j=1}^{s} \frac{|gen(cl)[N_j]|}{|[min_{r \in T} r[N_j], max_{r \in T} r[N_j]]|}) + \sum_{j=1}^{t} \frac{h(\wedge(gen(cl)[C_j]))}{h(H_{C_j})}$$

where $|cl|$ denotes the cardinality of cluster cl; $|[i_1, i_2]|$ is the size of the interval $[i_1, i_2]$ (the value $i_2 - i_1$); $\wedge(w), w \in H_{C_j}$ is the sub-hierarchy of H_{C_j} rooted at w; $h(H_{C_j})$ denotes the height of the tree hierarchy H_{C_j}.

Definition 5. *Total information loss for a solution $\mathcal{P} = \{cl_1, \cdots, cl_v\}$ of the l-diversity by clustering problem, denoted by $IL(\mathcal{P})$, is the sum of the information loss measure for all the clusters in $\mathcal{P}$.*

The information loss measure penalizes each tuple with a cost proportional with how "far" the tuple is from the cluster generalization information. Intuitively, the smaller the clusters are in a solution and the more similar the tuples in those groups will be, then less information will be lost. So, the desideratum is to group together the most similar objects (i.e. that cause the least possible generalization) in clusters with respect to the l-diversity requir

3 Dynamic Updating Time-evolving Microdata

Let $\mathcal{P} = \{cl_1, \cdots cl_v\}$ be a solution for the l-diversity problem for the microdata T. There are three problems arisen in the updating process. The first is that when there is a new segment of data that needs to be added to the original microdata, and how to process the update to make it preserve the l-diversity. The second is when parts of the original data needs to be deleted, how to maintain l-diversity. The third one is a hybrid version of adding and deleting. We can solve the third one by independently solve the first and second problem. The first and second problems are described as follows:

Problem 1. The dataset $\triangle^{+}T$ is added to T. How to efficiently update $\mathcal{P}$ to $\mathcal{P}' = \{cl'_1, \cdots cl'_v\}$ that ensures l-diversity for $T \cup \triangle^{+}T$?

Problem 2. The dataset $\triangle^{+}T$ is deleted from T. How to efficiently update $\mathcal{P}$ to $\mathcal{P}' = \{cl'_1, \cdots cl'_v\}$ that ensures l-diversity for $T - \triangle^{-}T$?

The solution to the first problem is as follows. Each tuple r in $\triangle^{+}T$ is added to that cluster in $\mathcal{P}$ that, increased with r, will produce the minimum increase of total information loss. Due to multiple insertions, when a cluster grows bigger than $2k$ elements and it has at least $2l$ distinct sensitive values, we can split that cluster into two sub-cluster in a greedy manner that tries to minimize total information loss.

The solution to the second problem proceeds as follows. Each tuple r in $\triangle^{-}T$ is deleted from the cluster currently containing it. The clusters that remain with less than k elements or less than l distinct sensitive values are dispersed into the other cluster, in order to maintain l-diversity for $T - \triangle^{-}T$. Each element r of cl'_j is relocated to another cluster will produce the minimum increase of the total information loss. If a cluster grows bigger then $2k$ elements and with more than $2l$ distinct sensitive values, that cluster will be split into two, which is the same process as the first problem.

Theorem 1. *Let T be a set of records and l be the specified anonymity requirement. Every cluster that the algorithm finds has at least l distinct sensitive values, but no more than $2l - 1$.*

Proof. As the algorithm finds a cluster with the number of sensitive attribute values of the records is equal to or greater than l, every cluster contains at least l distinct sensitive values. If there is one cluster with less than l distinct sensitive values, each record in this cluster could be relocated to other cluster. That is, in the worst case, the records with $l - 1$ distinct sensitive values are added to another single cluster which already has records with l distinct sensitive values. Therefore, the maximum number of distinct sensitive values in a cluster is $2l - 1$.

4 Experimental Results

In our experiment, we adopted the publicly available data set, Adult Database, at the UC Irvine Machine Learning Repository [10], which has become the benchmark of this field and was adopted by [6, 8, 5]. In this section we compare, in terms of efficiency, scalability, and results

Attribute	Type	Distinct values	Height
Age	Numeric	74	5
Workclass	Categorical	8	3
Education	Categorical	16	4
Country	Categorical	41	3
Marital Status	Categorical	7	3
Race	Categorical	5	3
Gender	Categorical	2	2

Table 3: Features of Quasi-identifier

quality, the static algorithm from [16] with our incremental algorithms. The algorithms have been implemented and executed on P4 machine with 2.4 GHz each and 1 GB of RAM.

Table 3 provides a brief description of the data including the attributes we used, the type of each attribute data, the number of distinct values for each attribute, and the height of the generalization hierarchy for each attribute. In all the experiments, we considered Age as the set of numerical quasi-identifier attributes, and Work-class, Marital-status, Occupation, Race, Sex, and Native-country as the set of categorical quasi-identifier attributes. l-diversity property was enforced in respect to the quasi-identifier consisting of all these seven attributes. We removed all tuples that contained the unknown value for one or more of the quasi-identifier attributes from the data.

The experiment contains three steps. First, the static algorithm from [16] was applied on a dataset, which is a subset extracted from the entire adult dataset. Second, we applied the dynamic algorithm to update the clusters produced by the static algorithm and considering several different choices of inserting/deleting dataset. Third, the static algorithm was applied on the entire new updated dataset datasets. When doing inserting, T has 10000 objects, and the inserting dataset had different sizes, varying between 0.5% and 50% of the entire adult dataset. When doing deleting, the deleting parts had different sizes, varying between 50 and 5000 tuples. The values considered for l were 2, 4.

In Fig. 1(a) and 2(a), we compare: a) the information loss for each set of clusters obtained by applying the static l-diversity algorithm, followed by updating algorithms on the corresponding updated dataset; b) the information loss for the set of clusters obtained by applying the static algorithm on the updated dataset. We can expect that the information loss obtained by the updating algorithm deteriorates when the increment/ decrement dataset grows in size w.r.t. the initial dataset size. Neverthe-

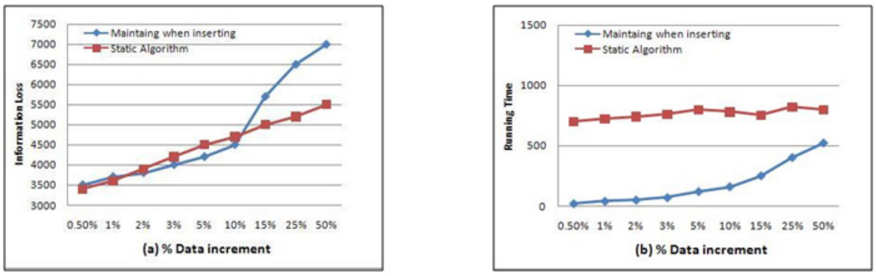

Fig. 1: Information Loss vs Running Time (I)

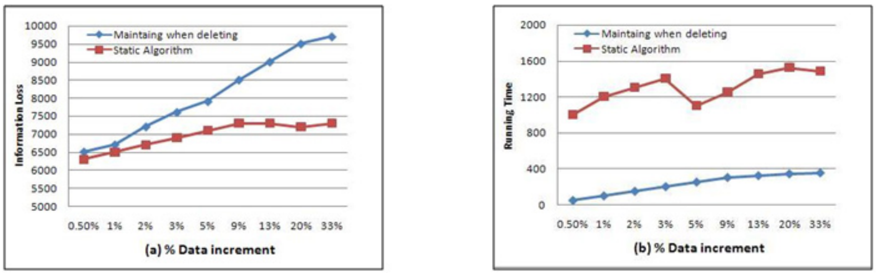

Fig. 2: Information Loss vs Running Time (II)

less, as is usually the case in the real world databases evolution, for small modification amounts, the information loss remains at about the same level as if we would use the static algorithm. From these experiments, we draw the conclusion that the updating algorithm can be used for maintaining l-diverse microdata when the changing portions of the datasets are small.

Fig. 1(b) and Fig. 2(b) illustrate the running time for the updating algorithms compared with the static algorithm. The time for incrementally processing the datasets grows with the size of the datasets, however, it is still significantly lower than the time required to process the datasets with the static algorithm. Whether to use updating algorithm or a static one is to be decided by the requirement of data quality and execution time. The advantages of dynamic updating algorithms can maintain acceptable data quality while the running time is tolerated.

5 Conclusion and Future Work

In this paper, we identified and investigate the problem of maintaining l-diversity in time evolving microdata, and proposed a simple yet effective solution. Maintaining l-diversity against various types of dynamic updates is an important and practical problem. As we show in experiments, the running time of the dynamic updating algorithms is significantly lower than that of the static algorithm. From the data quality perspective, the information loss is also comparable with the information loss obtained by applying the non-incremental algorithm to the final dataset. As future work, we will make more comprehensive experimental studies to compare the dynamic method with others and extend to other privacy paradigms.

Acknowledgement

Thanks for Professor Traian Marius Truta for his valued suggestion. This research was funded by Australian Research Council (ARC) grant DP0774450 and DP0663414.

References

1. G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas and A. Zhu. Anonymizing tables. *In Proc. of the 10th International Conference on Database Theory (ICDT05)*, pp. 246-258, Edinburgh, Scotland.
2. G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, A. Zhu. Approximation algorithms for k-anonymity. *Journal of Privacy Technology*, paper number 20051120001.
3. R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymity. *In Proceedings of the 21st International Conference on Data Engineering (ICDE)*, 2005.
4. J. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-Anonymization using Clustering Techniques In Internal Conference on Database Systems for Advanced Applications (DASFAA), April 2007
5. B. Fung, K. Wang, P. Yu. Top-down specialization for information and privacy preservation. *In Proc. of the 21st International Conference on Data Engineering (ICDE05)*, Tokyo, Japan.
6. K. LeFevre, D. DeWitt and R. Ramakrishnan. Incognito: Efficient Full-Domain k-Anonymity. *In ACM SIGMOD International Conference on Management of Data*, June 2005.
7. N. Li, T. Li, S. Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. *ICDE 2007*: 106-115
8. A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-Diversity: Privacy beyond k-anonymity. *ICDE 2006.*
9. A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. *In Proc. of the 23rd ACM-SIGMOD-SIGACT-SIGART Symposium on the Principles of Database Systems*, pp. 223-228, Paris, France, 2004.
10. D. J. Newman, S. Hettich, C. L. Blake and C. J. Merz. UCI Repository of Machine Learning Databases, available at `www.ics.uci.edu/-mlearn/MLRepository.html`, University of Califonia, Irvine, 1998.
11. P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010-1027. 2001

12. X. Sun, M. Li, H. Wang and A. Plank. An efficient hash-based algorithm for minimal k-anonymity problem. *31st Australasian Computer Science Conference (ACSC 2008)*, Wollongong, NSW, Australia. CRPIT 74, pp: 101-107.
13. X. Sun, H. Wang and J. Li. On the complexity of restricted k-anonymity problem. *10th Asia Pacific Web Conference (APWeb 2008)*, LNCS 4976, pp: 287-296, Shenyang, China.
14. X. Sun, H. Wang, J. Li, T. M. Traian and P. Li. (p^+, α)-sensitive k-anonymity: a new enhanced privacy protection model. *In 8th IEEE International Conference on Computer and Information Technology (IEEE-CIT 2008)*, 8-11 July 2008, Sydney, Australia. pp:59-64.
15. L. Sweeney. k-anonymity: A Model for Protecting Privacy. *International Journal on Uncertainty Fuzziness Knowledge-based Systems*, 10(5), pp 557-570, 2002
16. T. M. Traian and V. Bindu, Privacy Protection: l-diversity Property *International Workshop of Privacy Data Management (PDM2006), In Conjunction with 22th International Conference of Data Engineering (ICDE)*, Atlanta, 2006.
17. T. M. Truta, Alina Campan, k-Anonymization Incremental Maintenance and Optimization Techniques, ACM Symposium on Applied Computing (SAC2007), special track on Data Mining, Seoul, Korea, 2007
18. W. E. Winkler. Advanced Methods for Record Linkage, *Proceedings of the Section on Survey Research Methods, American Statistical Society*, 467-472, 1994
19. R. Wong, J. Li, A. Fu, K. Wang. (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. *KDD 2006*: 754-759.