Approximation algorithms for optimizing privacy and utility

Pranav Khaitan
Dept of Computer Science, Stanford University
pranavkh@cs.stanford.edu

Korra Sathya Babu, Sanjay Kumar Jena, Banshidhar Majhi
Department of Computer Science and Engg, NIT Rourkela
{ ksathyababu, skjena , bmajhi }@nitrkl.ac.in

Abstract - There are two major aspects of Data Privacy –identity disclosure and attribute disclosure. A number of algorithms have been developed to protect identity disclosure. However, many of these algorithms have not been much successful in preventing attribute disclosure. The standard implementation techniques for the commonly used privacy models are NP-hard. They are time-consuming and often unnecessary. This paper investigates the existing algorithms for preserving privacy by preventing identity disclosure. Improved approximation algorithms are proposed to prevent identity disclosure and attribute disclosure. The existing approximation algorithm for k-anonymity is extended by applying simulated annealing to reduce attribute disclosure. The concepts of l-diversity and t-closeness are used for minimizing attribute disclosure in polynomial time. An improved algorithm for calculating the value of t-closeness is also proposed. A new parameter is introduced to anonymize the dataset considering both privacy and utility. An existing algorithm has been modified to consider the utility while determining the best full domain generalization scheme. This is followed by a proposed approximation algorithm which takes into consideration the utility while preserving the privacy. It is an example of partial domain generalization wherein different tuples can be anonymized to different extents.

Keywords- privacy, approximation algorithm, anonymity, diversity, closeness, utility

I. INTRODUCTION

Privacy concerns have increasingly gained a lot of importance over the past few years. As more and more data are being collected and stored, lot of objections are being raised towards the manner in which these data are utilized and published by organizations. There are situations when organizations have to publish microdata even when the individuals do not like their personal information to be made public. However, it is very important to preserve the privacy of individual users while releasing a data set. There are privacy constraints which require organizations to prevent both identity disclosure and attribute disclosure. Several algorithms have been proposed for preventing both of them. The k-anonymity protection model is a very useful and commonly used method for preventing identity disclosure. Two commonly used models for preventing attribute disclosure are l-diversity and t-closeness.

II. RELATED WORK

A. k-Anonymity

L. Sweeney [1] had introduced the k-anonymity protection model, explored related attacks and provided ways in which these attacks can be thwarted.

Definition (k-Anonymity): Let RT(A1,...,An) be a table and QIRT be the quasi-identifier associated with it. RT is said to satisfy k-anonymity if and only if each sequence of values in RT[QIRT] appears with at least k occurrences in RT[QIRT].

B. l-diversity

The l-diversity model is a very useful model for preventing attribute disclosure and it has been introduced in [2].

Definition (l-diversity): A q-block is l-diverse if it contains at least l well-represented values for the sensitive attribute S. A table is l-diverse if every q-block is l-diverse.
Enforcing the *l*-diversity principle ensures *l* well-represented values for the sensitive attribute in every q-block. *l*-diversity has several advantages over Bayes-optimal privacy which are mentioned in [2].

C. t-clossness

The t-closeness model [3] overcomes a major drawback of l-diversity by considering the semantic relationships among the attribute values. It uses the concept of distance between two probability distributions to calculate the closeness of an equivalence class to the entire dataset.

Definition (t-closeness): An equivalence class is said to have t-closeness if the distance between the distribution of a sensitive attribute in one class and the distribution of the attribute in the entire dataset is no more than a threshold t. A table is said to have t-closeness if all equivalence classes have t-closeness.

D. Simulated Annealing

Simulated annealing (SA) is a generic probabilistic metaheuristic for the global optimization problem and can be used for locating a good approximation to the global minimum of a given function in a large search space. The method has been independently described by S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi[4].

E. Approximation Algorithm for k-Anonymity

It has been shown in [5] that the k-Anonymity problem is NP-hard even when the attribute values are ternary and only suppression is allowed. An O(k)-approximation algorithm for the problem is also provided in [5]. The algorithm constructs a forest where each vertex corresponds to a row in the dataset and each tree corresponds to an equivalence class containing a set of rows. The approximation algorithm for k-anonymity as given in [5] is summarized below.

Construction of Forest : Given an instance of the k-Anonymity problem, an edge weighted complete graph G = (V, E) is constructed. The vertex set V contains a vertex corresponding to each vector in the k-Anonymity problem. For two rows a and b, the unscaled generalization cost for the jth component, $h_{a,b}(j)$, is the lowest level of generalization for attribute j for which the j^{th} components of both a and b are in the same partition, i.e. the lowest level for which both have the same generalized value. The scaled generalization cost is obtained by dividing the unscaled generalization cost with the scaling factor l_j which corresponds to the total number of levels of generalizations for the j^{th} attribute. The weight, w(e), of an edge e = (a, b) is the sum over all components j of the scaled generalization cost, i.e. $w(e) = \sum_j ha, b(j)/lj$. The jth attribute is said to contribute a weight of $h_{a,b}(j)/l_j$ to the edge e. The cost of any k-anonymity solution is the sum of generalization cost of all vertices. OPT is the cost of an optimal k-anonymity solution.

Algorithm:
Step 1: A forest G with cost at most OPT is constructed. Since this forest satisfies the k-anonymity condition, the number of vertices in each tree is at least k.
Step 2: Edges are then deleted to decompose the forest such that each component has between k and max{2k − 1, 3k − 5} vertices. The decomposition is done in a way that does not increase the sum of the costs of the edges.

III. PROPOSED ALGORITHMS

A. Approximation Algorithm for l-diversity

l-diversity is often very difficult to achieve. At times, it is unnecessary to strictly abide by the l-diversity rule. It has been shown in [2] that the problem of finding optimal solution for l-diversity is NP-hard and there exists no polynomial time algorithm for it. We therefore use an approximation algorithm which has a complexity O (KN) where K is the number of iterations performed and N is the size of the dataset. The Simulated Annealing technique is used to perform the approximation. Using this technique one can perform a fixed number of iterations to get a considerable increase in the diversity of the data set. The algorithm is terminated whenever a minimum threshold value is reached. There are different ways of measuring l-diversity [2]. We consider the Entropy l-diversity. The entropy of an equivalence class E is defined to be:

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

in which S is the domain of the sensitive attribute, and p(E, s) is the fraction of records in E that have sensitive value s. A table is said to have entropy l-diversity if for every equivalence class E, Entropy (E) ≥ log l. The flowchart and algorithm of our proposed approximation algorithm for l-diversity is shown below.

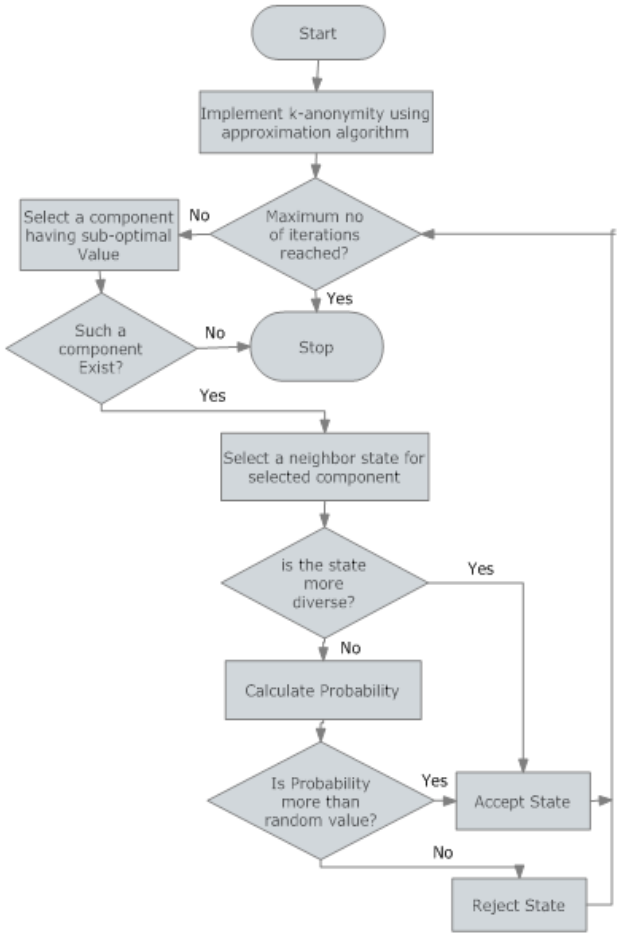

Figure 1. Approximation Algorithm for l-diversity

Algorithm:
Step 1: Execute the approximation algorithm for k-anonymity.
Step 2: For i=1 to maximum iterations, repeat steps 3 to 7.
Step 3: Select a component which has value of l less than the threshold l-value.
Step 4: If no such component exists, we have reached the optimal state and the algorithm is terminated. Else we move to the next step.
Step 5: Select a neighbor of the state and calculate its entropy l-diversity value
Step 6: If the entropy l-diversity value increases then move to the neighbor state
Step 7: If the entropy l-diversity value decreases then decision is based on the following function.

If p is greater than a randomly generated value, accept it or else reject it

$$P = e^{(E_2 - E_1)/T}$$

where E2 – l-Entropy of neighbor state
E1 – l-Entropy of existing state
T – Temperature of the system & varies with time

Algorithm to select a neighbor state:
Step 1: Select any node n from the selected component C.
Step 2: A number k is generated through a number generator.
Step 3: Select m as the kth nearest neighbor for the node n.
Step 4: The neighbor state is obtained by interchanging n and m in the previous graph.

```
ALGORITHM NEIGHBOUR()
{
        n = any node in selected component c
        k = number_gen()
        m = kth nearest neighbor of n
        Return state obtained after interchanging n and m
}
```

Number Generator: For the above algorithm we determine the value of k using a number generator . The number generator returns values from 1 to n-1 where n is the total number of nodes. Ideally, the probability of getting any number should be considerably larger than the probability of getting the next number. The number generator is designed in such a way that there is a much higher probability of getting a small value of k compared to larger values. Consider the following number generator as an example :

```
ALGORITHM NUMBER-GEN
{
        // Retrieves a random value between 0 & 1
        x = random ();
        // Inverse it
        y = 1/x
        // Convert it to integer format
        z = ( integer ) y
        if ( z < n )
                return z
        else
                return 1

}
```

The above algorithm is an example of a number generator wherein the probability of getting the nth nearest number being returned is double that of the $(n+1)^{th}$ nearest neighbor

B. t-closeness Algorithm

There are a number of ways to obtain the distance between two probability distributions. The Kullback-Leibler (KL) distance defined in [6] and the Earth Mover's distance (EMD) defined in [3] are two commonly used methods. EMD is a very popularly used measure and is based on the minimal amount of work needed to transform one distribution to another by moving distribution mass between each other. EMD measures the least amount of work needed to fill the holes with earth. A unit of work corresponds to moving a unit of earth by a unit of ground distance. EMD can be formally defined using the well-studied transportation problem. Let P = (p1, p2, ...pm),Q = (q1, q2, ...qm), and dij be the ground distance between element i of P and element j of Q. Find a flow F = [fij] where fij is the flow of mass from element i of P to element j of Q that minimizes the overall work.
The EMD value can be found using the formula:

$$D[P,Q] = \frac{1}{m-1}\left(|r_1| + |r_1 + r_2| + \cdots + |r_1 + r_2 + \cdots r_{m-1}|\right)$$

$$= \frac{1}{m-1}\sum_{i=1}^{i=m}\left|\sum_{j=1}^{j=i} r_j\right|$$

where $r_i = (p_i - q_i)\log\frac{p_i}{q_i}, for\ (i = 1,2,....,\ m)$

The following approximation algorithm for implementing t-closeness is proposed.
Algorithm:
Step 1: Execute the approximation algorithm for k - anonymity.
Step 2: For i =1 to maximum iterations, repeat steps 3 to 7.
Step 3: Select a component which has value of t-closeness greater than the threshold value.
Step 4: If no such component exists, the optimal state is reached and terminate the algorithm, else move to the next step.
Step 5: Select a neighbor of the state and calculate t-closeness value
Step 6: If the t-closeness value decreases then move to the neighbor state
Step 7: If the t-closeness value increases, then decide based on the following function. If p is greater than a randomly generated value, accept it or else reject it

$$P = e^{(t_1 - t_2)/T}$$

where t2 – EMD value of neighbor state
t1 – EMD value of existing state
T – Temperature of the system & varies with time

Improved method for calculating distance:

While EMD is one of the best measures known so far, it is certainly not perfect [4]. The relationship between the value t and information gain is unclear. For example, the EMD between the two distributions (0.01, 0.99) and (0.11, 0.89) is 0.1, and the EMD between (0.4, 0.6) and (0.5, 0.5) is also 0.1. However, it can be argued that the change between the first pair is much more significant than that between the second pair. In the first pair, the probability of taking the first value increases from 0.01 to 0.11, a 1000% increase. While in the second pair, the probability increase is only 25%. In general, what we need is a measure that combines the distance-estimation properties of the EMD with the probability scaling nature of the KL distance.
This paper proposes another method for calculating the value of t-closeness. This method overcomes the drawbacks faced

in previous methods for t-closeness. The proposed method is based on the following two concepts:
It reflects the distance-estimation properties of the EMD
It possesses the probability scaling nature of the KL distance.

We let $r_i = (p_i - q_i)\log(p_i / q_i)$, *for* $(i = 1,2,....,\ m)$ then the distance between P and Q can be calculated as:

$$D[P,Q] = \frac{1}{m-1}\left(|r_1| + |r_1 + r_2| + \cdots + |r_1 + r_2 + \cdots r_{m-1}|\right)$$

$$= \frac{1}{m-1}\sum_{i=1}^{i=m}\left|\sum_{j=1}^{j=i} r_j\right|$$

The factor $r_i = (p_i - q_i)$ is the same as that used for calculating the EMD value. In the above algorithm, an additional factor $\log(p_i / q_i)$ is introduced to take into consideration the probability scaling factor.

Example: Consider the following distributions:
Distribution pair 1: (0.01, 0.99) and (0.11, 0.89)
Distribution pair 2: (0.4, 0.6) and (0.5, 0.5)
EMD Value for Distribution pair 1 = 0.1
EMD Value for Distribution pair 2 = 0.1
The distance calculated using the improved algorithm for Distribution 1 is |-0.227|+ |-0.227+0.0106| = 0.227 + 0.216 = 0.443
The distance calculated using the improved algorithm for Distribution 2 is |-0.0223|+ |-0.0223+0.0182| = 0.0223 + 0.0041 = 0.0263
It is clearly visible that the difference between the first two pairs is much more than that between the other two pairs. However, according to the EMD algorithm, there is no difference between both the two pairs of distribution. The improved algorithm clearly shows a difference in both the distance values even though it takes into consideration the semantic values like the EMD algorithm.
Also the proposed algorithm is compared with the KL distance metric algorithm [6]. The KL distance is given by:

$$D[P,Q] = \sum_{i=1}^{m} p_i \log\frac{p_i}{q_i} = H(P) - H(P,Q)$$

where $H(P) = \sum_{I=1}^{m} p_i \log p_i$ is the entropy of P and $H(P,Q) = \sum_{I=1}^{m} p_i \log q_i$ is the cross-entropy of P and Q

To do the comparison, we consider an example of a student table where the overall distribution of the grade attribute is Q ={A, B, C, D, P , F}. The first equivalence class has distribution P1 = {A, B, C} and the second equivalence class has distribution P2 = {B, D, F}. Our intuition is that P1 results in more information leakage than P2, because the values in P1 are all in the lower end; thus to have D [P1, Q] > D [P2, Q]. The KL distance measure mentioned would not be able to do so, because from the algorithms point of view values such as B and F are just different points and have no other semantic meaning. These distance measures do not reflect the semantic distance among values. However, the proposed method would be able to judge the difference in both the probability distributions as it takes into account the semantic meanings.

C. Utility based Privacy Preservation

In the section that follows, we aim to maximize our utility while satisfying the privacy constraints. We introduce a new parameter E for any anonymized dataset v. E is defined as:

$$E(v) = \frac{AnonymizationCost}{PurityGain}$$

The parameter E takes into consideration both the anonymization cost and the information utility. A lower anonymization cost implies that lesser values are being suppressed which implies that more information can be obtained from the given dataset. Anonymization cost for the above parameter is measured using the concept of generalization cost mentioned in 2.5.1. The anonymization cost should be minimized to maximize the utility. An increase in purity results in an improvement in the utility of the data. Therefire, the purity gain should be maximized to get maximum information from the dataset. **T**he purity of a dataset is measured using the following formula:

$$Purity = \sum_{s\in S} p(E,s)^2$$

where S is the domain of the sensitive attribute, and p(E, s) is the fraction of records in E that have sensitive value s.

1) Proposed Algorithm for full domain generalization

The incognito algorithm proposed in [7] provides a practical framework for implementing one model of k--anonymization, called full-domain generalization. It introduces a set of algorithms for producing minimal full-domain generalizations, and shows that these algorithms perform up to an order of magnitude faster than previous algorithms on two real-life databases. The Incognito algorithm generates the set of all possible k-anonymous full-domain generalizations of T, with an optional tuple suppression threshold. However, we need to select the best generalization scheme from all possible schemes. We make this selection on the basis of the proposed utility factor. The following algorithm is a modified form of the incognito algorithm which utilizes our utility based parameter E(v) to obtain the optimum generalization scheme. In this algorithm, each node represents a generalization scheme for the entire dataset.

Enhanced Algorithm

Step 1: Set MinimumE = 0.
C1=Nodes in the domain generalization hierarchies of attributes
E1=Edges in the domain generalization hierarchies of attributes
Step 2: Repeat steps 3 to 10 for i=1 to total number of quasi-identifier attributes
Step 3: A modified breadth-first search is carried out over the graph (C_i,E_i) to get the set of multi-attribute

generalizations of size i with respect to which T is k-anonymous. This set is denoted as S_i .

Step 4: All the nodes in C_i with no nodes directed to them are added to the queue and steps 5 to 9 are repeated till the queue is empty

Step 5: Calculate the parameter E(v) for all nodes in the queue.

Step 6: The value of MinimumE is set to the minimum of previous MinimumE and E (v) value of all newly constructed nodes.

Step 7: Nodes satisfying k-anonymity are added to the set Final.

Step 8: Those nodes which do not satisfy k-anonymity in the present state are added to queue.

Step 9: Those nodes which have value greater than MinimumE are removed from the queue.

Step 10: After obtaining Si, the algorithm constructs the next level of candidate nodes C_{i+1} and edges E_{i+1}

Step 11: After all iterations are carried out, the node having minimum value for E (v) from among all nodes in Final is selected and returned. This node is the optimal node satisfying k-anonymity

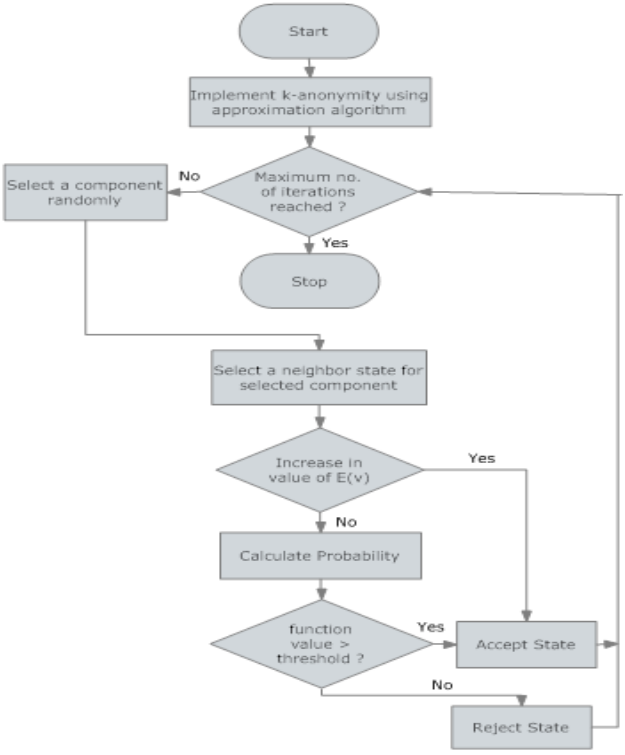

Figure 2. Utility based Privacy Preservation algorithm for partial domain generalization

Drawbacks of Full Domain Generalization Algorithm

In the full domain generalization algorithm, the output is a generalization scheme which must enforced on the entire data set. However in most occasions, it is unnecessary to enforce the same generalization scheme on tuples. It would lead to a great increase in the anonymization cost. There may some groups of tuples which could have been k-anonymized at a lower cost.

Another drawback of the full-domain generalization algorithm is the cost factor. In this algorithm, the parameter E(v) has to be calculated for the entire dataset at each generalization level. This is not required in a partial domain generalization algorithm.

2) Partial domain generalization

We propose a partial domain generalization algorithm which utilizes the parameter E for balancing utility and privacy. This algorithm overcomes some of the major drawbacks of the full domain algorithm. It does not require all tuples to be anonymized to the same extent. This is an approximation algorithm, and runs in polynomial time which would be much lesser than the previous algorithms giving optimal solutions. However, unlike the previous algorithm, it is not bound to give optimal results. Figure 2 shows the flowchart for this algorithm

IV. RESULTS AND OBSERVATIONS

A large student database was used for conducting all the experiments and implementing the proposed algorithms. The coding was carried out in Java on a P4 Dual Core machine

A. Approximation Algorithm for l-diversity

Firstly, we apply our proposed approximation algorithm for l-diversity on the sample data set. In the graph below, it can be seen that the value of l diversity is improved by repeated applications of the approximation technique. The increase in l-diversity value of entropy value is compared for different number of iterations. From the graph, it can be seen that the entropy increases as the number of iterations increases. Another important observation is that, as we increase the number of iterations, the rate of change of entropy decreases. Figure 3 shows the results of the approximation algorithm for l-diversity.

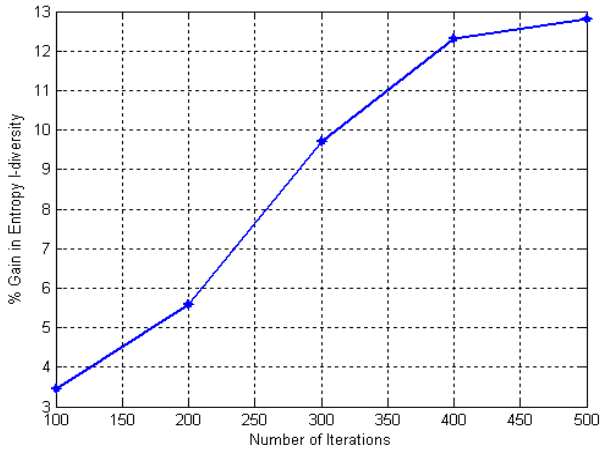

Figure 3. Percentage Gain in Entropy l-diversity.

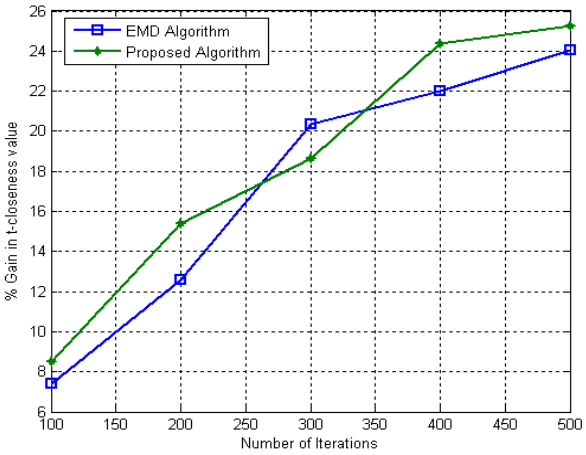

Figure 4. Percentage Gain in t-closeness value

B. *Approximation Algorithm for balancing Utility and Privacy*

The implementation was done for the proposed approximation algorithm on our sample database. The comparison was done by varying the number of iterations used in the simulated annealing method. Like the previous algorithms, E (v) shows a clear improvement in its value as repeated iterations are done. The primary purpose of using E (v) is to maximize the utility while minimizing the required Anonymization. Figure 5 shows the result.

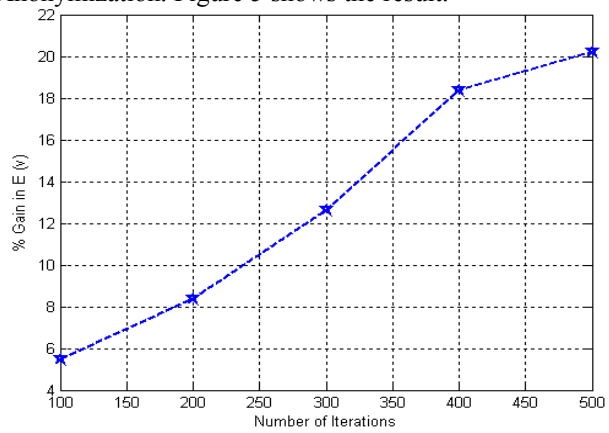

Figure 5. Performance graph of Utility based Privacy Preservation.

V. CONCLUSION

The proposed approximation algorithm using simulated annealing results in a significant gain in privacy values. The privacy values improve as the number of iterations increase. Moreover, the proposed algorithms are much faster than the optimal algorithm. A new parameter for calculating the utility E (v) has also been demonstrated successfully. This parameter has been applied for both full domain and partial domain generalization. It has been shown that the proposed t-closeness method gives a more clear picture about the given data than that obtained using EMD algorithm.

VI. FUTURE WORK

The approximation algorithms used by us do not take into consideration the anonymization cost while improving the values of l and t. Keeping the cost at its minimum may not be possible in several situations. Therefore, one also has to consider the anonymization cost before accepting or rejecting the new state. The proposed model requires an analyzing function to generate the minimum values of k and t. This analyzing function is very time consuming computationally. Research needs to be done in this direction so as to get an optimum value of k and t in the least possible time.

VII. REFERENCES

[1] L. Sweeney. k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, Vol. 10, October, 2002; PP: 557-570.

[2] Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam,M. l-Diversity: Privacy beyond k-anonymity. ACM Trans. On Knowledge Discovery from Data, Vol. 1, March, 2007, PP: 52 pages.

[3] N. Li and T. Li. t-closeness: Privacy beyond k-anonymity and l-diversity. Proceedings of ICDE, April, 2007, PP: 106-115.

[4] S. Kirkpatrick; C. D. Gelatt; M. P. Vecchi, Optimization by Simulated Annealing, Science, New Series, Vol. 220, No. 4598. (May 13, 1983), pp. 671-680.

[5] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation algorithms for k-anonymity. Transaction of Journal of Privacy Technology (JOPT), Vol 1. November, 2005.

[6] S. L. Kullback and R. A. Leibler. On information and sufficiency. Ann. Math. Stat., 22:79–86, 1951

[7] Kristen LeFevre , David J. DeWitt , Raghu Ramakrishnan, Incognito: efficient full-domain K-anonymity, Proceedings of the 2005 ACM SIGMOD international conference on Management of data, June 14-16, 2005.

[8] Latanya Sweeney, Achieving k-anonymity privacy protection using generalization and suppression, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol 10 PP: 571-588, October 2002.

[9] Dakshi Agrawal , Charu C. Aggarwal, On the design and quantification of privacy preserving data mining algorithms, Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, p.247-255, May 2001, Santa Barbara, California.