Optimizing Privacy-Accuracy Tradeoff for Privacy Preserving Distance-Based Classification

Dongjin Kim, University of Maryland Baltimore County, USA

Zhiyuan Chen, University of Maryland Baltimore County, USA

Aryya Gangopadhyay, University of Maryland Baltimore County, USA

ABSTRACT

Privacy concerns often prevent organizations from sharing data for data mining purposes. There has been a rich literature on privacy preserving data mining techniques that can protect privacy and still allow accurate mining. Many such techniques have some parameters that need to be set correctly to achieve the desired balance between privacy protection and quality of mining results. However, there has been little research on how to tune these parameters effectively. This paper studies the problem of tuning the group size parameter for a popular privacy preserving distance-based mining technique: the condensation method. The contributions include: 1) a class-wise condensation method that selects an appropriate group size based on heuristics and avoids generating groups with mixed classes, 2) a rule-based approach that uses binary search and several rules to further optimize the setting for the group size parameter. The experimental results demonstrate the effectiveness of the authors' approach.

Keywords: *Data Mining, Data Security, Distance-Based Mining, Privacy Preserving Data Mining, Privacy Protection*

INTRODUCTION

With the huge amount of data and its increasingly distributed sources across organizations, accurate, efficient, and fast analysis of the data for finding knowledge has become a major challenge. In many cases, these factors force companies or organizations to outsource their data mining tasks to a third party. In these circumstances, privacy of the outsourced data is a major concern because without proper protection, the data is subject to misuse.

For example, revealing identity information such as social security number, name, address, and date of birth may lead to identity theft. Another type of privacy risk is that revealing sensitive information such as preexisting medical conditions may cause negative impact such as denial of health insurance. Identity theft was the top concern among customers contacting the Federal Trade Commission (Federal Trade Commission, 2007). According to a Gartner

DOI: 10.4018/jisp.2012040102

study (Gartner Inc., 2007), there were 15 million victims of identity theft in 2006. Another study showed that identity theft cost U.S. businesses and customers $56.6 billion in 2005 (MacVittie, 2007). Therefore, legislation such as the Health Insurance Portability and Accountability Act (HIPAA) and the Gramm–Leach–Bliley Act (also known as the Financial Services Modernization Act of 1999) requires that the privacy of medical and financial data being protected.

There has been a rich body of work on privacy preserving data mining (PPDM) techniques. Two excellent surveys can be found at (Aggarwal & Yu, 2008; Vaidya, Zhu, & Clifton, 2005). The goal of privacy preserving data mining is two-fold: to protect privacy of the original data and at the same time still preserve the utility of sanitized data (often measured in quality of data mining). Note that these two goals are conflicting to each other because most PPDM techniques distort the original data (e.g., by adding random noise or making data values less accurate) to provide privacy protection. Obviously, the more distortion introduced, the better the privacy protection, but the lower the utility of data. Most proposed PPDM techniques have some tunable parameters which will lead to different degree of privacy protection and data utility. Thus these parameters need to be set correctly to achieve the optimal privacy and utility tradeoff.

For example, *K*-anonymity is a very commonly used privacy protection model (Sweeney, 2002a) which makes *K* people in the data set indistinguishable such that their identities will not be revealed. A number of techniques have been proposed to implement this model (Bayardo & Agrawal, 2005; LeFevre, DeWitt, & Ramakrishnan, 2005, 2006a, 2006b; Samarati, 2001; Sweeney, 2002b; Xiao & Tao, 2006). However all these techniques must set the correct value of *K*. If *K* is too large, the data may be distorted too much such that the quality of mining may become very poor. If *K* is too small, the degree of privacy protection may not be sufficient. More recently researchers have proposed several privacy models such as *L*-diversity (Machanavajjhala, Kifer, Gehrke, & Venkitasubramaniam, 2007), *t*-closeness (Li, Li, & Venkatasubramanian, 2007), and differential privacy (Dwork, 2006). All these models need to set some parameters, e.g., we need to set proper values for *L* in the L-diversity model, *t* in the t-closeness model, and ε (the degree of differential privacy) in the differential privacy model.

However, there has been little research on how to tune these parameters efficiently and effectively. Most existing research simply leaves the task of setting parameters to users. However, without proper guidelines, users often have trouble to set the correct parameter values. Another alternative is a brute-force approach. This approach tries many possible settings of parameters and examines the utility (often in terms of mining quality) and the degree of privacy protection of each setting. It then selects the setting with the best utility-privacy tradeoff. However, computing the utility and degree of privacy protection often requires two steps: 1) the privacy preserving technique being considered needs to be applied to the original data set to generate a sanitized data set; 2) the data mining algorithm needs to be executed on the sanitized data set to generate mining results. These two steps are both time consuming and the brute-force approach needs to repeat these two steps for every parameter setting. This is clearly inefficient in practice.

This paper studies the problem of optimizing parameters for a popular privacy preserving technique for distance-based classification: the condensation method (Aggarwal & Yu, 2004). The major benefit of the condensation method as compared to other methods is that it generates synthetic data so it is difficult to recover the identity of the original data. It also preserves the statistical properties of the original data so it works well for multiple distance-based classification algorithms. The condensation method works as follows. It divides data into clusters (groups) such that each cluster contains at least *K* points (individuals). Each group is then replaced with synthetic data by preserving statistics of the original group. However, the condensation method needs to set an appropriate value of *K* (the group size) for optimal privacy-utility tradeoff. A very small group size may not provide enough privacy protection and a very large group size may lead to poor mining results.

We have made the following contributions:

- We propose a class-wise condensation method which automatically selects an appropriate group size. This method also ensures that each group (cluster) will only contain records from the same class (i.e., no mixed groups). This often leads to better classification accuracy.
- We propose a rule-based approach to further optimize the group size selection. This approach uses binary search and several rules to quickly narrow down the range of group sizes.
- We conducted extensive experiments using real data sets and the results show that the class-wise algorithm leads to better accuracy without sacrificing privacy protection. The experimental results also show that the rule-based approach often finds optimal or near optimal group sizes and takes much less time than the brute-force approach.

The rest of the paper is organized as follows. We first discuss related work. We then give necessary background about the condensation method and propose the class-wise method. We will then present our rule-based approach to further optimize the group size parameter. Finally we will report experimental results and conclude the paper.

RELATED WORK

Existing studies on privacy preserving data mining consider two scenarios: the centralized scenario when the data is sent to a third party for analysis or is published, and the distributed scenario when several parties want to collaboratively build a data mining model but do not want to directly share their local data. This paper focuses mainly on the first scenario. Next we briefly describe four commonly used privacy models as well as techniques to enforce these models.

The two most popular privacy models are K-anonymity (Sweeney, 2002a) and L-diversity (Machanavajjhala et al., 2007). K-anonymity prevents revealing of identities of individuals in the data set caused by linking attacks, which link attributes (called quasi-identifier) such as birth date, gender, and ZIP code with publicly available data sets. K-anonymity ensures that there are at least K people with the same quasi-identifier such that the risk of identity disclosure is reduced to 1/K. L-diversity further limits revealing of sensitive attribute values by further requiring that the people with the same quasi-identifier contain at least *L* well-represented sensitive values such that attackers cannot discover the values of sensitive attributes easily. A more advanced model called t-closeness (Li et al., 2007) tries to make sure the distribution of sensitive attributes in each equivalence class is similar to the global distribution. A differential privacy model (Dwork, 2006) is also proposed to perturb the results of aggregations or statistics computed over the data set such that it limits the increased privacy risks for a person to have his or her information present in the data set.

Many privacy protection techniques have been proposed to enforce the above privacy models. The earlier work uses *random perturbation* that adds or multiplies random noise to each of the data elements such that individual data values are distorted while the underlying distributions can be reconstructed (Agrawal & Aggarwal, 2001; Agrawal & Srikant, 2000; Kim & Winkler, 2003). However, the random perturbation approach is subject to several attacking schemes as described (Huang, Du, & Chen, 2005; Kargupta, Datta, Wang, & Sivakumar, 2003).

Generalization and suppression have been used to enforce the K-anonymity, L-diversity, and t-closeness models (Bayardo & Agrawal, 2005; LeFevre et al., 2006a, 2006b; Sweeney, 2002b; Wong, Li, Fu, & Wang, 2006; Xiao & Tao, 2006). Data swapping is another technique that exchanges a subset of attributes between selected pairs of records. Projection-based methods have also been proposed to preserve the distance between data points (which will enable distance-based mining) (Chen & Liu, 2005; Liu, Kargupta, & Ryan, 2006; Mukherjee, Banerjee, Chen, & Gangopadhyay, 2008; Mukherjee, Chen, & Gangopadhyay, 2006). A condensation method (Aggarwal & Yu, 2004) has also been proposed, which generates synthetic data based on the statistics of the original data. This approach enforces K-anonymity model and leads to very good mining quality for many distance-based data mining algorithms. However, all these privacy protection methods need to set some parameters and existing work does not discuss how to efficiently and effectively set these parameters.

There has been a rich body of work in data mining with distributed environment. Since many data mining algorithms generate learning models, existing work in this field applies techniques called *secure multi-party computations* (SMC) (Goldreich, 1998). SMC techniques allow different parties to share information securely and jointly and to calculate some function over datasets of all parties, without revealing local data sets. SMC techniques are typically based on distributed cryptography protocols. A good survey can be found at (Vaidya et al., 2005). In this paper, we focus on the centralized scenario.

CONDENSATION METHOD

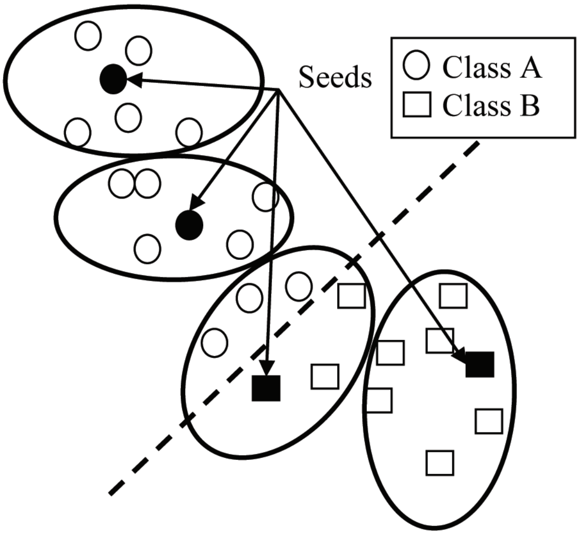
This paper extends the condensation method introduced in Aggarwal and Yu (2004). As mentioned in the related work section, this approach is suitable for distance based classification algorithms such as K-Nearest Neighbor and Support Vector Machines, because it preserves statistical properties of the data. It also does not need to reconstruct mining models. We first give a brief overview of the condensation method and discuss its drawbacks. We will then describe a modified condensation method, which was proposed in Banerjee, Chen, and Gangopadhyay (2010) and addressed some of the problems of condensation. Finally, we propose a class-wise condensation algorithm, which further improves the modified condensation method.

Original Condensation Method: Figure 1 shows an example how the original condensation method works. The circles represent records in class A and rectangles represent records in class B. The condensation method first selects n / k random points from the data as seeds, where *n* is the total number of records and *k* is the degree of K-anonymity. In Figure 1, n=25 and k=6. So 4 points (with dark background) are selected as seeds. For each such seed, the method then assigns *k-1* nearest points to that seed. The seed and these assigned points form an equivalence group. If there are remaining points, they are assigned to closest seed. Each group now contains at least *k* points. For example, in Figure 1, 4 groups are generated, each with at least 6 points.

The method then computes some statistics such as mean and covariance for each group and generates synthetic data using these statistics. The synthetic data points in each group will have the same statistical properties as the original data, but cannot be distinguished from each other because they are generated randomly. Figure 2 shows synthetic data generated by the condensation method. Attackers usually cannot infer the identity of synthetic data records because they cannot distinguish records in the same group.

Modified Condensation Method: A modified condensation method (Banerjee et al., 2010) was proposed to address some shortcomings of the original condensation method. In the original condensation method, seeds are selected randomly so some bad initial choices of seeds may lead to bad grouping. The modified condensation method uses a pre-clustering step (K-means clustering is used in Banerjee et al., 2010) such that initial seeds are centers

Figure 1. Groups generated by condensation method



of clusters generated by K-means. The original method also uses uniform weights on all attributes. In reality, some attributes may be more important than others. For example, the class label attribute in Figure 1 is probably more important than others because it indicates which class the record belongs to. The modified condensation method assigns higher weights to outcome attributes (e.g., the class label).

Figure 3 shows the pseudo code for the modified condensation method. Step 1-1 selects initial seeds as cluster centers using K-means clustering. Step 1-2 sorts the centers by ascending order of cluster size. Some of these clusters may contain fewer than *k* records (thus they do not satisfy K-anonymity) and some may contain more than *k* records. So step 1-3 to 1-5 move records from clusters with more than k records to clusters with fewer than k records such that each cluster will have at least k records.

Step 2-2 computes mean and covariance for each group. Step 2-3 to 2-5 generate synthetic data with the same mean and covariance. The detail explanations can be found in Aggarwal and Yu (2004).

CLASS-WISE CONDENSATION METHOD

However, both the original condensation method and the modified method may still lead to poor data mining results when the group size (*k*) is not set appropriately.

For example, consider the data set in Figure 1. It contains 25 records, 15 in class A and 10 in class B. The dotted line in Figure 1 shows a clear boundary between two classes (records to the upper left of the line belong to class A and those to the lower right belong to class B). Suppose the user selects *k* as 6, then each group should have at least 6 records and there can be at most 4 groups. However, one of the groups (the second group from the right) will have records from both classes. Since the step of synthetic data generation will make all records in that group indistinguishable from each other, the synthetic data generated from that group may blur the boundary between two classes and make it difficult to use the synthetic data to predict the class label. For example, Figure 2 shows synthetic data generated from Figure 1 and there is no clear boundary between class A and B in the synthetic data.

Figure 2. Synthetic data generated by condensation method

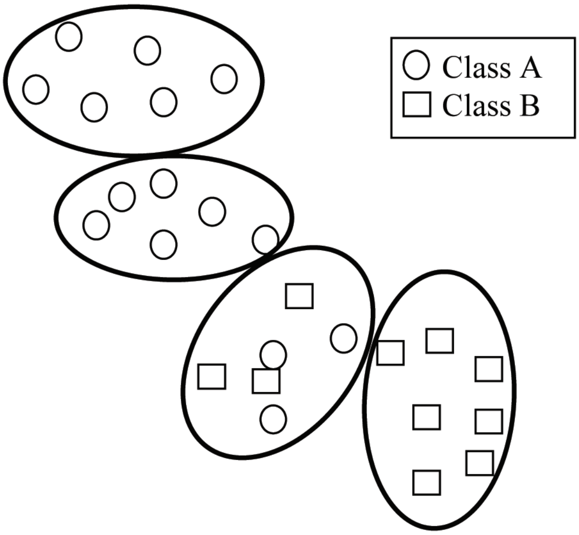


Figure 3. Modified condensation method

```
Input: original data, group size $k$, weight of class label
Output : perturbed data
Step 1. Dividing original data to groups
    1-1. Run K-means clustering to select initial seeds. Here number of
         clusters equals $\lfloor n/k \rfloor$. Weighted Euclidean distance is used.
    1-2. Sorting the clusters based on the cluster size in ascending order. Let
         them be $C_1$, $C_2$, …
    1-3. For each class $C_i$
    1-4.     If $C_i$ has fewer than $k$ records, move $k$-$|C_i|$ records ($|C_i|$ is size of
             $C_i$)from clusters larger than $C_i$ such that these records are closest to the
             centers of $C_i$
    1-5. End for
Step 2. Generate synthetic data
    2-1. For each group $C_i$
    2-2.     Compute mean and covariance matrix for records in $C_i$
    2-3.     Apply Principal Component Analysis over records in $C_i$
    2-4.     Generate $|C_i|$ synthetic data points with mean equals 0 and variance
             equals to Eigen Values after PCA transform
    2-5.     Apply reverse PCA transformation and add mean of each
             dimension to synthetic data (this ensures synthetic data has the same
             dimensions as original data)
    2-6. End For
```

On the other hand, Figure 4 shows a grouping when we set *k*=5. Now each group contains at least 5 records and we create 5 groups. None of these groups contain records from both classes. As a result, the synthetic data generated will still preserve the class boundary.

We have conducted some experiments on real life data sets and we found that this problem is more severe when the number of records in each class is not evenly distributed (i.e., there are classes with many records and classes with very few records). For example, consider a medical data set that contains few patients with a certain type of disease while the majority of patients do not have that disease. Since small classes often have many records that lie close to the boundaries between different classes, records in such classes are often assigned to groups with records from multiple classes. As a result, if users run classification algorithms on synthetic data generated by the condensation or modified condensation methods, the accuracy of classification is often quite low for smaller classes. This often has negative impact on the utility of privacy preserving data mining because it is often quite important to correctly predict the label of the smaller classes (e.g., it is important to predict if someone has higher risk of a certain type of disease).

The challenge is how do we select an appropriate value of *k* and also prevent having groups with mixed classes. In this paper, we propose a class-wise condensation method that solves this challenge. Figure 5 shows the details of the algorithm.

We will use the example in Figure 1 to illustrate how the class-wise method works. If the user does not provide a group size, the method will first compute an appropriate group size *k*. We want to make sure *k* is greater than or equal to a user defined threshold *t*. This prevents the cases when very small groups are generated such that there is not enough privacy protection. In Figure 1, since we have a very small data set with only 25 records, we set *t*=5. For real data sets, larger *t* value should be set. In our experiment, we set *t* according to the data set size.

To make sure each class will have groups containing at least *t* records, Step 1-2 of the algorithm divides the size of each class by *t* and rounds the result to an integer. In Figure 1, class A's size is 15 and class B's size is 10. So we get two integers floor (15/5) = 3 and

Figure 4. Class-wise condensation

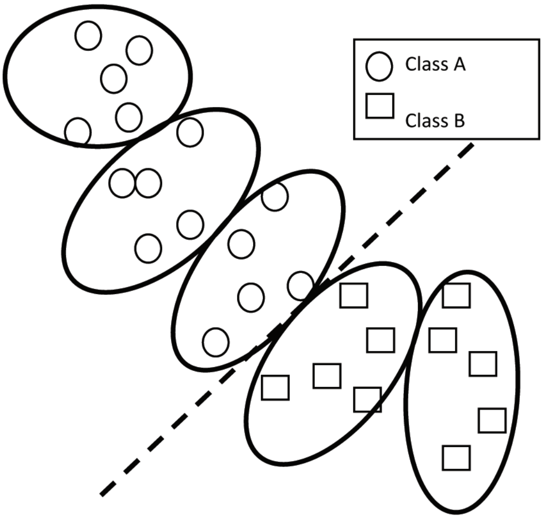


Figure 5. Class-wise condensation method

```
Input: original data, thresholds t, weight of class label, an optional group size
Output : perturbed data
Step 1. If the group size is not provided, calculate an appropriate group size
    1-1.    Find the number of class and the size of each class
    1-2.    Divide the size of each class by t and round it to the largest previous integer, i.e.,
        compute $floor(|C_1|/t), floor(|C_2|/t), ..., floor(|C_m|/t)$
    1-3.    Compute $g=GCD(floor(|C_1|/t), ..., floor(|C_m|/t))$. GCD is the greatest common
        divisor.
Step 2. Run modified condensation method on records in each class (i.e., the method will be run
once for records in each class), with group size $g*t$
```

floor(10/5) = 2. Suppose we consider every *t* records as a unit, these integers indicate how many units each class has.

Now we want to find an appropriate group size that is greater than *t*. There could be many such group sizes. So a simple solution is to only consider group sizes that are multiples of *t*. Clearly, each class should also contain an integer number of groups to make sure there is no mixed group. The next theorem will show how to find such a group size.

Theorem 1: Let $|C_i|$ ($1 \leq i \leq m$) be size of class C_i and *t* be the threshold for minimal group size. Let floor($|C_i|/t$) be the largest integer no greater than $|C_i|$ divided by *t*. Let $GCD(x_1,x_2,..)$ be the greatest common divisor of integers x_1, x_2, … Let $g^*= g^*t$ where $g = GCD(floor(|C_1|/t), floor(|C_2|/t),…, floor(|C_m|/t))$. Then $g^* \geq t$ and each group can be divided by integer number of groups, each with at least g^* rows. We call g^* the *approximate GCD*.

Proof: Clearly *gro* because it is the greatest common divisor. Since $g^*=g^*t$ so g^*o . Since *g* is the greatest common divisor of $floor(|C_1|/t), floor(|C_2|/t)…$, so $floor(|C_i|/t)$ can be divided by *g* without a remainder. Clearly $floor(|C_i|/t)\ t \leq |Ci|$ because $floor(|Ci|/t)$ is the largest integer that is no greater than $|C_i|/t$. Let $floor(|C_i|/t) = g\ m_i$ where m_i is an integer. Then $g\ m_i\ t\ n$ $nC_i|$. This means that each class C_i can be divided into m_i groups such that each group will contain at least g^*t rows.

For example, in Figure 1, *t*=5 and $|C_1|$=15, $|C_2|$=10. Group size = GCD(floor(15/5), floor(10/5)) * 5 = GCD(3,2) * 5 = 5. Thus class A will be divided into 15/5=3 groups and class B will be divided into 10/5 = 2 groups.

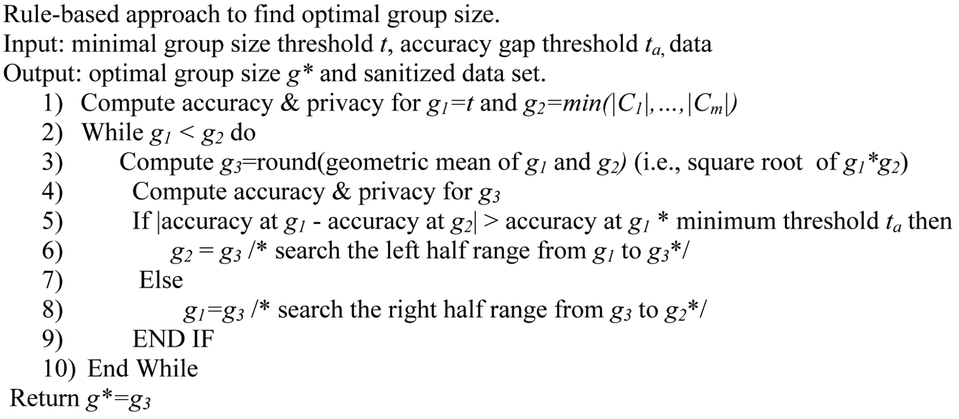
Note that we compute approximate GCD rather than real GCD of class sizes because we want to avoid having too small group sizes. For example, suppose $|C_1|=1001$ *and* $|C_2|=501$. If we directly compute GCD for 1001 and 501, the GCD is 1. Clearly we cannot have a group of size one because there will be no privacy protection. Instead, if we set t=20, we compute GCD(floor(1001/20),floor(501/20))=GCD(50,25)=25. So the group size equals 25 * t = 500. Class C_1 can be divided into 2 groups and C_2 will remain as one group.

Finally, the method just calls the modified condensation method in a class-wise manner. That is, for each class C_i, we extract records in C_i and form a smaller data set. The modified condensation method is then run on this smaller data set. This will divide each class into several groups, each with at least g^* records. The method will generate groups as shown in Figure 4 for the data set in Figure 1. Since data is divided into groups one class at a time, there will be no mixed groups and the accuracy of data mining will be improved.

RULE-BASED APPROACH

In the class-wise algorithm, we set the group size to approximate GCD. However, it is unclear

Figure 6. Rule-based approach to find optimal group size



```
Rule-based approach to find optimal group size.
Input: minimal group size threshold $t$, accuracy gap threshold $t_a$, data
Output: optimal group size $g^*$ and sanitized data set.
    1)  Compute accuracy & privacy for $g_1=t$ and $g_2=min(|C_1|,...,|C_m|)$
    2)  While $g_1 < g_2$ do
    3)       Compute $g_3$=round(geometric mean of $g_1$ and $g_2$) (i.e., square root of $g_1*g_2$)
    4)         Compute accuracy & privacy for $g_3$
    5)         If |accuracy at $g_1$ - accuracy at $g_2$| > accuracy at $g_1$ * minimum threshold $t_a$ then
    6)              $g_2 = g_3$ /* search the left half range from $g_1$ to $g_3$*/
    7)          Else
    8)                $g_1=g_3$ /* search the right half range from $g_3$ to $g_2$*/
    9)         END IF
    10)  End While
 Return $g^*=g_3$
```

whether this group size will lead to the optimal privacy-utility tradeoff. For example, we can reduce the group size to increase the utility of data because smaller group size means less distortion to data. On the other hand, this will also reduce the degree of privacy protection because group size is the value of K in K-anonymity model. Similarly, we may increase the group size such that privacy may increase but utility may decrease.

We can certainly use a brute-force approach to try all possible group sizes and then choose the group size with the best privacy-utility tradeoff. However, as stated in the introduction section, this approach is impractical due to the heavy cost of measuring the utility and privacy for each group size. Thus we also propose a rule-based approach.

This approach is based on binary search to quickly narrow down the range of optimal group size. Figure 6 shows the details of the approach.

Suppose we want to find optimal group size falling in a range from g_1 to g_2. At line 1 we set g_1 to the minimal group size threshold t because it is the smallest group size. We set g_2 to the minimal class size because to ensure that we do not have mixed groups (i.e., groups with records from multiple classes), the group size must be less than or equal to minimal class size. We compute the degree of privacy and utility of sanitized data for g_1 and g_2. Privacy can be measured by calling the class-wise condensation algorithm (Figure 5) using group size g_1 and g_2 to sanitize the data. The degree of privacy protection can then be computed using sanitized data. In this paper we use the accuracy of data mining algorithm to measure utility of sanitized data. This can be computed by running the data mining algorithm on the sanitized data and report mining accuracy.

At line 3 we compute the midpoint of g_1 and g_2. This allows us to narrow down the range of group sizes. In this paper we use geometric mean (i.e., square root of g_1*g_2) rather than arithmetic mean because in practice the gap between g_1 and g_2 is often quite large so using geometric mean allows us to select smaller groups (which lead to higher utility) first. For example, suppose g_1=20, g_2=2000, then geometric mean is 200 while the arithmetic mean is 1010. We use g_3 to represent the geometric mean of g_1 and g_2.

At line 5 to line 9, we try to narrow down the range of group sizes by comparing the accuracy at g_1 and g_2. We use the following rules to decide the new range of group size.

- *Rule 1*: If the difference of the utilities between the smallest (g_1) and the largest (g_2) group sizes is greater than a minimum threshold t_a, then we need to pay more attention to optimize utility (we use accuracy of mining as the utility measure in this paper) because utility changes significantly as we change group size. Otherwise, we need to pay more attention to optimizing privacy because utility does not change significantly for different group sizes. Figure 7 (a) shows the case when utility at g_1 is much higher than the utility at g_2. Figure 7 (b) shows the case when utility at g_1 is about the same as the utility at g_2.

- *Rule 2*: If we want to optimize utility, we will focus on left half of group size range (i.e., from g_1 to g_3) because group sizes in this range lead to higher utility than the group sizes in the right half (i.e., from g_3 to g_2). Figure 7 (a) shows that when utility changes significantly from g_1 to g_2, we should focus on group sizes in the left half.
- *Rule 3*: If we want to optimize privacy, we want to focus on right half (i.e., from g_3 to g_2 because the privacy for group sizes in this range is higher than the privacy for group sizes in the left half. Figure 7 (b) shows that when utility does not change much from g_1 to g_2, we should focus on group sizes in the right half.

Line 5 to line 9 implement these rules. In line 5, if the gap of accuracy between g_1 and g_2 is greater than the accuracy of g_1 multiplied by a minimum threshold t_a, we should optimize accuracy. Hence, at line 6 we will search the left half (i.e., $g_2=g_3$). Otherwise, we should optimize privacy and go to the right half (i.e., $g_1=g_3$).

We repeat the above process until the range of groups becomes empty (i.e., when $g_1 \geq g_2$). In the next section, we will compare the rule-based approach with the brute-force approach. In the brute-force approach, the expected number of group sizes we need to consider is $\mathrm{O}(min(|C1|,\ldots,|Cm|) - \mathrm{t})$ in worst scenario. Whereas, the expected number of group sizes in worst scenario for rule-based approach is $\mathrm{O}\left(\log_2(min(|C1|,\ldots,|Cm|) - \mathrm{t})\right)$.

EXPERIMENTS

This section experimentally evaluates our approach. We first describe set up of the experiment. We then compare our proposed class-wise algorithm with the existing modified condensation method. Finally we compare our rule-based approach with the brute force approach to further optimize the group size for the class-wise algorithm.

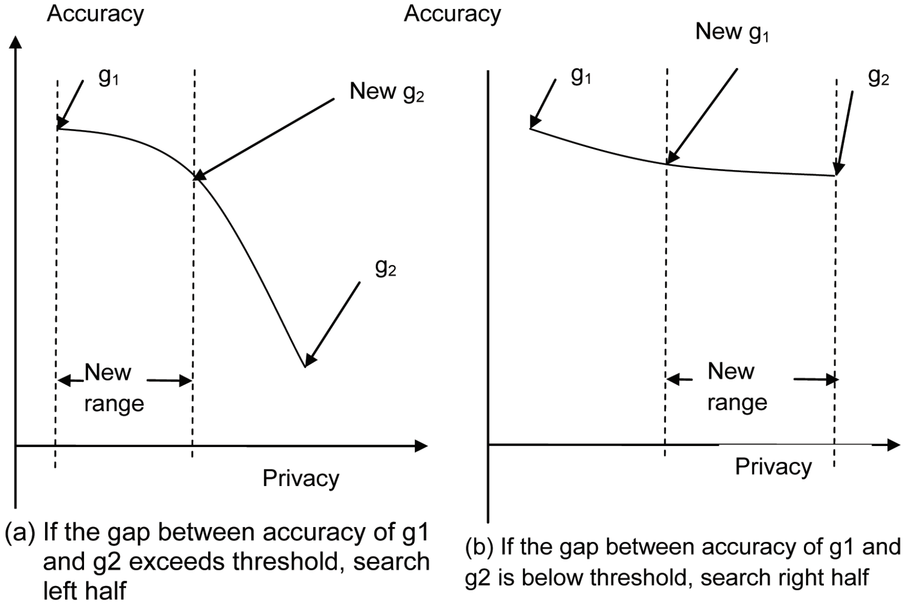
Setup: The experiments were conducted on a machine with Intel(R) Core(TM) Duo CPU E6550 2.33GHz and 2.33GHz, 3.25 of RAM and running Windows XP Professional. All algorithms were implemented using Matlab R2010a. The data mining algorithm used in experiment is K-Nearest Neighborhood classification (KNN) with K=1. Since the condensation approach randomly generates synthetic data, we ran each sanitization method 3 times and report the average results.

We used seven data sets from UCI machine learning repository (Hettich, Blake, & Merz, 1998). 10% of each dataset is randomly chosen as test data and the remaining is used as training data. The details of each data set are listed in Table 1.

Sanitization algorithms: We compared our class-wise sanitization algorithm with the modified condensation method proposed in Banerjee et al. (2010). We do not compare our method to the original condensation method because the modified condensation algorithm is an improved version of the condensation approach and has been shown to provide better utility-privacy tradeoff than the original condensation approach (Banerjee et al., 2010).

Metrics: Since distance-based mining is typically applied to numerical data, we used the confidence interval metric proposed in Agrawal and Srikant (2000) to measure privacy because it works for numerical

Figure 7. Illustration of rules



(a) If the gap between accuracy of g1 and g2 exceeds threshold, search left half

(b) If the gap between accuracy of g1 and g2 is below threshold, search right half

data. When transformed attribute *x* can be estimated with *c%* confidence in the interval $[x_1, x_2]$, then privacy equals

$$\frac{(x_2 - x_1)}{\max(x) - \min(x)}$$

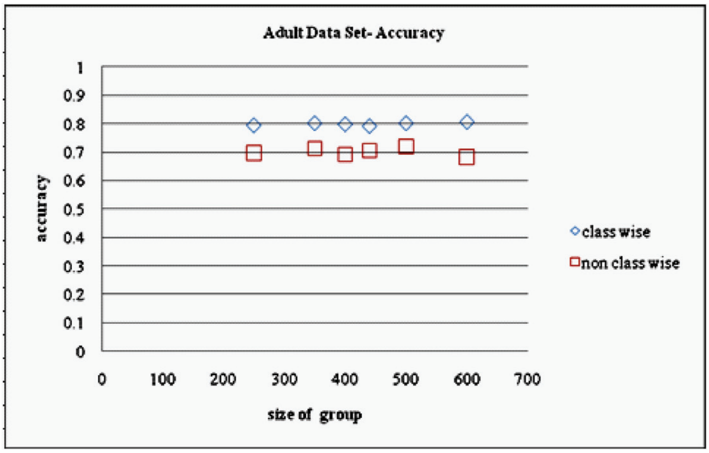
Where max(x) is the maximal value of x and min(x) is the minimal value. In the experiments, 95% confidence is used.

The quality of classification is measured by accuracy. Accuracy is measured by calculating the percentage of correct classification in whole test dataset. In addition, we also show the accuracy for each class.

Table 1. Properties of data sets used in experiments

Dataset	Rows	columns	number of class
Adult Data Set	32561	14	2
Brest Cancer Wisconsin(Diagnostic) Data Set	569	32	2
Johns Hopkins University Ionosphere Data Set	351	34	2
Waveform Database Generator	5000	40	3
German Credit Data Set	1000	24	2
Iris Plants Database	150	4	3
Magic Gamma Telescope Data 2004	19020	11	2

Figure 8. Accuracy of KNN for adult data set



Results of comparing sanitization algorithms: We compared our class-wised method with the modified condensation method (referred to as non class-wised). We also varied the group sizes.

Figure 8 reports the accuracy of K-Nearest Neighbor Classification for Adult data set. We also varied the group sizes from 200 to 600. The results show that for all group sizes tested, our approach (class-wise approach) leads to higher accuracy than the non class-wise approach. This is expected because our approach will not generate mixed groups, thus the utility of data is better preserved.

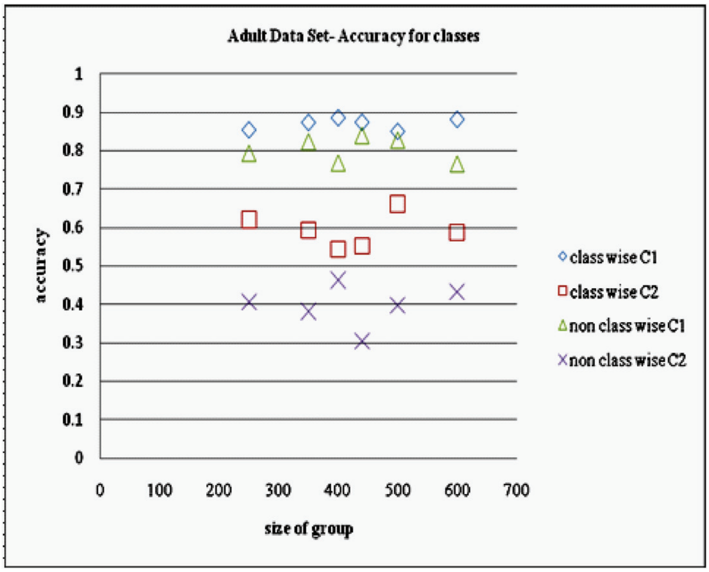
Figure 9 shows the accuracy for each class for the Adult data set. Adult data set contains two classes: C_1 and C_2. The size of class C_1 is significantly larger than the size of C_2. The results show that the classification accuracy of C_2 is quite low (less than 50%) for the non class-wise approach. This is expected because the non class-wise method generates some mixed groups. Since class C_2 is much smaller than C_1, its records are more likely to be mixed with records from C_1 in the same group. This leads to lower accuracy for C_2. On the other hand, our approach has significantly higher accuracy for C_2 because our approach does not generate mixed groups.

Figure 10 reports the privacy for both methods. The results show that for all the group sizes, our approach (the class-wise approach) has slightly higher privacy than the non class-wise approach. Since the class-wise approach also has much higher utility than the non class-wise approach, the class-wise approach is clearly superior to the non class-wise approach.

The results for other data sets are similar and omitted due to space constraint. The results show that the class wise algorithm leads to higher accuracy than non class-wise algorithm. The gap is often more significant for smaller classes. Note that smaller classes are often quite important, e.g., in the Wisconsin Breast Cancer data set the smaller class has malignant tumor and thus it is important to correctly predict the class label for anyone in that class. The privacy of both approaches does not differ much, especially for larger group sizes. Overall, the class-wise algorithm has better utility-privacy tradeoff than the non class-wise algorithm.

Optimizing Group Size: Next we evaluate our rule-based approach to select the optimal group size. We set the threshold *t* for minimal group size to about 5-20% of the size of the minimal class size. The rationale is that *t* needs to be fraction of the minimal class size to make sure we do

Figure 9. Accuracy per class for adult data set



not have mixed groups. We have also tried different *t* values in the experiment and the results do not change much.

We set the threshold t_a for accuracy gap to 5%. For the brute-force approach, since it is impractical to try all possible group sizes, we divided the range of possible group sizes into a number of equal width buckets and plotted accuracy and privacy for each bucket boundary points. We then manually pick the group size with the best utility-privacy tradeoff.

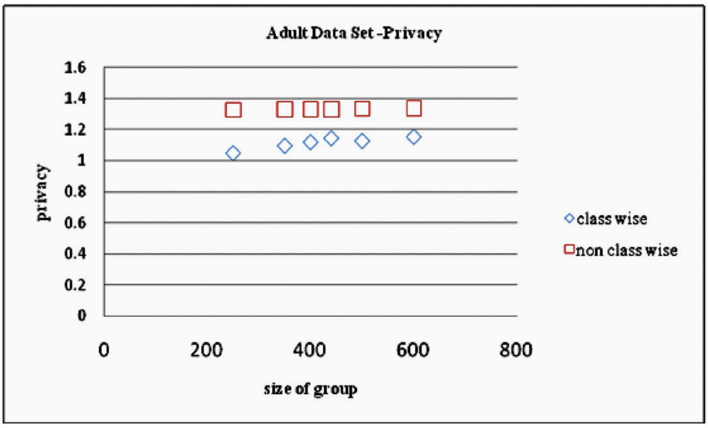
Table 2 reports the group size returned by rule-based and brute-force approaches, as well as privacy and accuracy at those group sizes. It also reported the value of *t* and the approximate GCD (i.e., *g*t*) returned by the class-wise algorithm, where *g* is the greatest common divisor of class sizes divided by *t*.

In five out of seven data sets the rule-based approach and brute-force approach return the same group sizes. The exceptions are the Breast Cancer dataset and the Iris dataset, where the brute-force approach returns larger group sizes. However, the privacy and accuracy for those group sizes in the two data sets are quite similar to the privacy and accuracy of the group sizes returned by the rule-based approach.

Interestingly, the results also show that in most case, group sizes returned by rule-base approach and the brute-force approach are the same or very close to the approximate GCD selected by the class-wise algorithm. When group size equals the approximate GCD, there is no group with mixes classes, so accuracy is quite reasonable. Thus the gap of accuracy between minimum thresholds of group size (i.e., t) and the approximate GCD is not very significant and large group sizes lead to more privacy protection. Hence approximate GCD is often a very good choice. The only exception for the class-wise algorithm is the Ionosphere data set where the rule-based approach returned group size 45 because it leads to higher accuracy but similar privacy.

Execution time: Figure 11 shows the number of group sizes checked by the rule-based approach and the brute-force approach. For each group size, we need to call the class-wise algorithm to compute the ac-

Figure 10. Privacy for adult data set



curacy and privacy. Figure 12 reports the total execution time.

The results show that for very small data sets (Iris, German Credit, Ionosphere, and Breast Cancer), there is not much difference in terms of execution time for both approaches because there are not many group sizes to be checked with and each check is not that expensive due to small data size. However, the rule-based approach leads to significant saving in execution time for the 3 larger data sets: Magic, Waveform, and Adult because it needs to check significantly smaller number of group sizes and the cost of checking privacy and accuracy for a large data set is significant.

Findings: Based on the experimental results, we find that the class-wise condensation algorithm leads to significantly better accuracy than the non class-wise algorithm, without sacrificing privacy protection. The improvement is more significant for smaller classes because they tend to be put into mixed groups in the non-class wise algorithm.

Table 2. Results of rule-based approach and brute force approach

Dataset	*t*	Approx. GCD	Group size rule base	Privacy rule base	Accuracy rule base	Group size brute force	Privacy brute force	Accuracy brute force
Breast cancer	20	60	60	0.6091	0.9942	180	0.6633	0.9825
Adult	1120	6720	6720	1.3326	0.8034	6720	1.3326	0.8034
Wave form	200	1400	1400	0.7479	0.8013	1400	0.7479	0.8013
Iono-sphere	30	90	45	1.4375	0.8981	51	1.4731	0.8967
Magic	480	4800	4800	0.726	0.7673	4800	0.726	0.7673
Iris	10	40	40	0.6564	0.9556	40	0.6564	0.9556
German	10	190	190	1.7301	0.7133	190	1.7238	0.7133

Figure 11. The number of group sizes checked

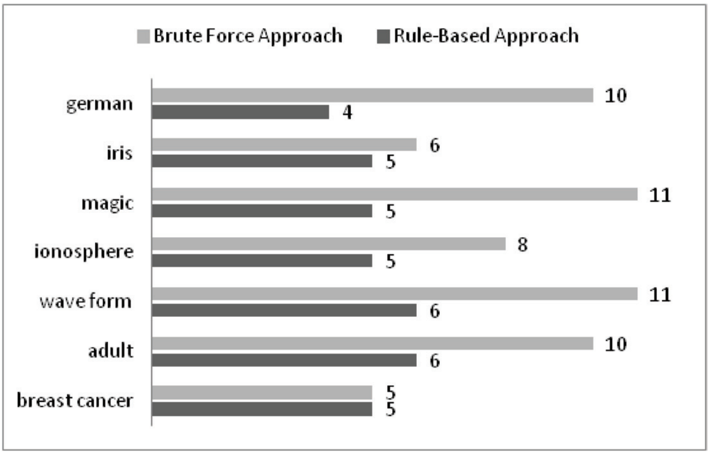
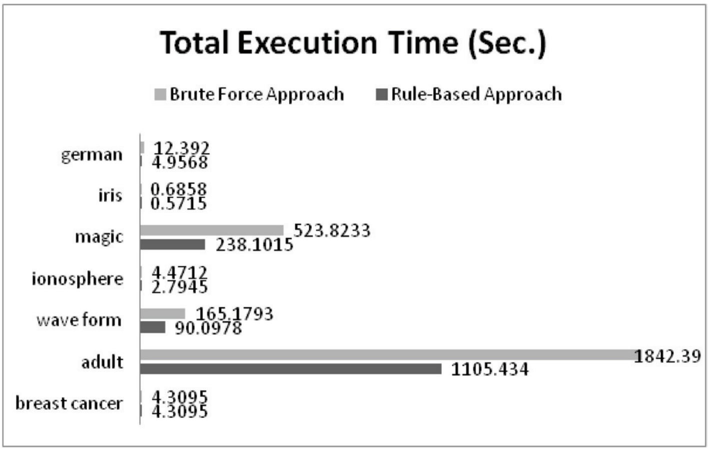


Figure 12. Total execution time of rule-based approach and brute force approach



We also find that using approximate GCD often leads to almost optimal group size. This is because using approximate GCD will eliminate mixed groups and thus leads to reasonable accuracy in most cases. In case approximate GCD is not sufficient, we can use the rule-based approach to find an appropriate group size. The results show that in most cases the rule-based approach will find the same group sizes as the brute-force approach, and the rule-based approach is more efficient for large data sets. Since it is quite expensive to compute accuracy and privacy for large data sets, rule-based approach brings significant savings in execution time for large data sets.

CONCLUSION

In this paper, we propose a novel class-wise condensation algorithm which improves the utility of sanitized data and at the same time still protects privacy. We also study the problem of optimizing the group size parameter for the condensation approach, which is a common privacy preserving data mining method for distance-based mining. We proposed an efficient rule-based approach that uses binary search and several rules to quickly find the appropriate group sizes. Experiments verified the benefits of the class-wised and the rule-based approach.

In future research, we want to extend the rule-based approach to other parameter optimizing problems for privacy preserving data mining techniques.

ACKNOWLEDGMENT

This material is based in part upon work supported by the National Science Foundation under Grant Number IIS 0713345. The work was done when Dongjin Kim was at UMBC.

REFERENCES

Aggarwal, C. C., & Yu, P. S. (2004). A condensation approach to privacy preserving data mining. In *Proceedings of the 9th International Conference on Extending Database Technology*, Heraklion, Crete, Greece.

Aggarwal, C. C., & Yu, P. S. (2008). *Privacy-preserving data mining: Models and algorithms*. New York, NY: Springer.

Agrawal, D., & Aggarwal, C. C. (2001). On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the 20th ACM SIGMOD SIGACT-SIGART Symposium on Principles of Database Systems*, Santa Barbara, CA (pp. 247-255).

Agrawal, R., & Srikant, R. (2000). Privacy preserving data mining. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, Dallas, TX (pp. 439-450).

Banerjee, M., Chen, Z., & Gangopadhyay, A. (2010). A utility-aware and holistic approach for privacy preserving distributed mining with worst case privacy guarantee. In *Proceedings of the Secure Knowledge Management Workshop*, New Brunswick, NJ.

Bayardo, R. J., & Agrawal, R. (2005). Data privacy through optimal k-anonymization. In *Proceedings of the 21st International Conference on Data Engineering*, Tokyo, Japan (pp. 217-228).

Chen, K., & Liu, L. (2005). A random rotation perturbation approach to privacy-preserving data classification. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, Houston, TX (pp. 589-592).

Dwork, C. (2006). Differential privacy. In *Proceedings of 33rd International Colloquium on Automata, Languages and Programming, Part II*, Venice Italy (pp. 1-12).

Federal Trade Commission. (2007). *Identity theft resource center: Facts and statistics: Find out more about the nation's fastest growing crime*. Retrieved from http://www.idtheftcenter.org/artman2/publish/m_facts/Facts_and_Statistics.shtml

Gartner Inc. (2007). *Gartner says number of identity theft victims has increased more than 50 percent since 2003*. Retrieved from http://www.gartner.com/it/page.jsp?id=501912

Goldreich, O. (1998). *Secure multi-party computation*. Rehovot, Israel: Weizmann Institute of Science.

Hettich, S., Blake, C. L., & Merz, C. J. (1998). *UCI Repository of machine learning databases*. Retrieved from http://www.ics.uci.edu/simmlearn/MLRepository.html

Huang, Z., Du, W., & Chen, B. (2005). Deriving private information from randomized data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Baltimore, MD (pp. 37-48).

Kargupta, H., Datta, S., Wang, Q., & Sivakumar, K. (2003). On the privacy preserving properties of random data perturbation techniques. In *Proceedings of the Third IEEE International Conference on Data Mining*, Melbourne, FL (pp. 99-106).

Kim, J. J., & Winkler, W. E. (2003). *Multiplicative noise for masking continuous data* (Tech. Rep. No. 2003-01). Washington, DC: Statistical Research Division, U.S. Bureau of the Census.

LeFevre, K., DeWitt, D. J., & Ramakrishnan, R. (2005). Incognito: Efficient full-domain k-anonymity. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Baltimore, MD (pp. 49-60).

LeFevre, K., DeWitt, D. J., & Ramakrishnan, R. (2006a). Mondrian multidimensional k-anonymity. In *Proceedings of the 22nd International Conference on Data Engineering*, Atlanta, GA (p. 25).

LeFevre, K., DeWitt, D. J., & Ramakrishnan, R. (2006b). Workload-aware anonymization. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA (pp. 277-286).

Li, N., Li, T., & Venkatasubramanian, S. (2007). t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *Proceedings of the 23rd International Conference on Data Engineering*, Istanbul, Turkey (pp. 106-115).

Liu, K., Kargupta, H., & Ryan, J. (2006). Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering*, *18*(1), 92–106. doi:10.1109/TKDE.2006.14

Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, *1*(1), 1–52.

MacVittie, D. (2007, August 31). Javelin 2006 identity fraud report. *Network Computing*.

Mukherjee, S., Banerjee, M., Chen, Z., & Gangopadhyay, A. (2008). A privacy preserving technique for distance-based classification with worst case privacy guarantees. *Data & Knowledge Engineering*, *66*(2), 264–288. doi:10.1016/j.datak.2008.03.004

Mukherjee, S., Chen, Z., & Gangopadhyay, A. (2006). A privacy preserving technique for euclidean distance-based mining algorithms using fourier-related transforms. *Very Large Data Base Journal*, *15*(4), 293–315. doi:10.1007/s00778-006-0010-5

Samarati, P. (2001). Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, *13*(6), 1010–1027. doi:10.1109/69.971193

Sweeney, L. (2002a). K-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, *10*(5), 557–570. doi:10.1142/S0218488502001648

Sweeney, L. (2002b). Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, *10*(5), 571–588. doi:10.1142/S021848850200165X

Vaidya, J., Zhu, Y. M., & Clifton, C. W. (2005). *Privacy preserving data mining (Advances in Information Security)*. New York, NY: Springer.

Wong, R. C. W., Li, J., Fu, A., & Wang, K. (2006). (alpha, k)-anonymity: An enhanced k-anonymity model for privacy preserving data publishing. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA (pp. 754-759).

Xiao, X., & Tao, Y. (2006). Anatomy: Simple and effective privacy preservation. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, Seoul, Korea (pp. 139-150).

Dongjin Kim works for Korea Expert as a Business Rule Management System Engineer. He received his Master of Science degree from the Department of Information Systems at the University of Maryland Baltimore County in December 2010. His research specialization is in the areas of privacy preserving data mining in the medical field and Enterprise Decision Support System.

Zhiyuan Chen is an associate professor at the Department of Information Systems, University of Maryland Baltimore County. He received a PhD degree in Computer Science from Cornell University in August 2002. His research interests include privacy preserving data mining, data navigation and visualization, XML, automatic database tuning, and database compression.

Aryya Gangopadhyay is a Professor and the Chair of the Department of Information Systems at UMBC. His current research interests include privacy preserving data mining and healthcare IT. Dr. Gangopadhyay has authored/edited five books, and published numerous peer-reviewed scholarly articles in journals, conference proceedings, and book chapters. Dr. Gangopadhyay has been the Editor-in-Chief of the International Journal of Computational Models and Algorithms in Medicine *since 2009. Dr. Gangopadhyay 's research has been funded by federal agencies such as NSF, NIST, U.S. Department of Education, and state agencies such as Maryland State Highway Administration and Maryland state Board of Elections.*