On Variable Constraints in Privacy Preserving Data Mining

Charu C. Aggarwal, Philip S. Yu
IBM T. J. Watson Research Center
{ charu, psyu }@us.ibm.com

Abstract

In recent years, privacy preserving data mining has become an important problem because of the large amount of personal data which is tracked by many business applications. In many cases, users are unwilling to provide personal information unless the privacy of sensitive information is guaranteed. A recent framework performs privacy preserving data mining by using a condensation based approach. In this framework, the privacy of all records is treated homogeneously. It is therefore inefficient to design a system with a uniform privacy requirement over all records. We discuss a new framework for privacy preserving data mining, in which the privacy of all records is not the same, but can vary considerably. This is often the case in many real applications, in which different groups of individuals may have different privacy requirements. We discuss a condensation based approach for privacy preserving data mining in which an efficient method is discussed for constructing the condensation in a heterogeneous way. The heterogeneous condensation is capable of handling both static and dynamic data sets. We present empirical results illustrating the effectiveness of the method.

1 Introduction

Privacy preserving data mining has become an important problem in recent years, because of the large amount of consumer data tracked by automated systems on the internet. The proliferation of electronic commerce on the world wide web has resulted in the storage of large amounts of transactional and personal information about users. In addition, advances in hardware technology have also made it feasible to track information about individuals from transactions in everyday life. In many cases, users are not willing to supply such personal data unless its privacy is guaranteed. Therefore, in order to ensure effective data collection, it is important to design methods which can mine the data with a guarantee of privacy. Some interesting discourses on the nature of privacy in the context of recent trends in information technology may be found in [6, 9, 10]. The recent focus on privacy in data collection has resulted to a considerable amount of research on the subject [1, 2, 3, 4, 5, 7, 11, 12, 15, 16]. A recent approach to privacy preserving data mining has been a *condensation-based* technique [2]. This technique essentially creates condensed groups of records which are then utilized in one of two ways:

- The statistical information in the pseudo-groups can be utilized to generate a new set of pseudo-data which can be utilized with a variety of data mining algorithms.
- The condensed pseudo-groups can be utilized directly with minor modifications of existing data mining algorithms.

The condensation approach of [2] is also referred to as the k-indistinguishability model. A record is said to be *k-indistinguishable*, when there are at least k other records in the data (including itself) from which it cannot be distinguished. Clearly, when a record is 1-indistinguishable, it has no privacy. The k-indistinguishability of a record is achieved by placing it in a group with at least $(k-1)$ other records. This model model shares a number of conceptual characteristics with the k-anonymity model [18], though the algorithms for doing so are quite different. Another important difference between the two schemes is that the former does not rely on domain specific hierarchies (as in the case of the k-anonymity model). The k-indistinguishability model can also work effectively in a dynamic environment such as that created by data streams.

In the model discussed in [2], it was assumed that all records have the same privacy requirement. This is also the case for the k-anonymity model in which the level of privacy is fixed a-priori. In most practical applications, this is not be a reasonable assumption. For example, when a data repository contains records from heterogeneous data sources, it is rarely the case that each repository has the same privacy requirement. Similarly, in an application tracking the data for brokerage customers, the privacy requirements of retail investors are likely to be different from those of institutional investors. Even among a particular class of customers, some customers

(such as high net-worth individuals) may desire a higher level of privacy than others. In general, we would like to associate a different privacy level with each record in the data set.

Let us assume that we have a database $\mathcal{D}$ containing N records. The records are denoted by $\overline{X_1} \ldots \overline{X_N}$. We denote this desired privacy level for record $\overline{X_i}$ by $p(i)$. The process of finding condensed groups with varying level of point specific privacy makes the problem significantly more difficult from a practical standpoint. This is because it is not advisable to pre-segment the data into different privacy levels before performing the condensation separately for each segment. When some of the segments contain very few records, such a condensation may result in an inefficient representation of the data. In some cases, the number of records for a given level of privacy k' may be lower than k'. Clearly, it is not even possible to create a group containing only records with privacy level k', since the privacy level of the entire group would then be less than k'. Therefore, it is not possible to create an efficient (and feasible) system of group condensation without mixing records of different privacy levels. This leads to a number of interesting trade-offs between information loss and privacy preservation. We will discuss these trade-offs and the algorithms to optimize them.

In many cases, the data may be available at one time or it may be available in a more dynamic and incremental fashion. We discuss two cases for our algorithm:

- We discuss an algorithm to perform the condensation when the entire data is available at one time.
- We discuss an algorithm for the case when the data is available incrementally. This is a more difficult case because it is often not possible to design the most effective condensation at the moment the data becomes available.

We will show that in most cases, the algorithm for performing the dynamic group construction is able to achieve results which are comparable to the algorithm for static group construction.

This paper is organized as follows. In the next section, we will discuss some notations and definitions and also introduce the locality sensitive condensation approach. We will first discuss the simple case in which an entire data set is available for application of the privacy preserving approach. This approach will be extended to incrementally updated data sets in section 3. The empirical results are discussed in section 4. Finally, section 5 contains the conclusions and summary.

2 The Condensation Approach

In this section, we will discuss the condensation approach for privacy preserving data mining. Before describing details of the algorithm, we will discuss some notations and definitions. We assume that we have a set of N records, each of which contain d dimensions. We also assume that associated with each data point i, we have a corresponding privacy level $p(i)$. The overall database is denoted by $\mathcal{D}$ whereas the database corresponding to the privacy level p is denoted by $\mathcal{D}_p$. The privacy level for a record is defined as follows:

DEFINITION 2.1. *The privacy level for a given record is defined as the minimum number of other records in the data from which it cannot be distinguished.*

In the condensation based approach, the data is partitioned into groups of records. Records within a given group cannot be distinguished from one another. For each group, we maintain certain summary statistics about the records. This summary statistics provides the ability to apply data mining algorithms directly to the condensed groups of records. This information also suffices to preserve information about the mean and correlations across the different dimensions. The size of the groups may vary, but its size is at least equal to the desired privacy level of each record in that group. Thus, a record with privacy level equal to $p(i)$ may be condensed with records of privacy levels different from $p(i)$. However, the size of that group must at least be equal to the maximum privacy level of any record in that group.

Each group of records is referred to as a condensed unit. Let $\mathcal{G}$ be a condensed group containing the records $\{\overline{X_1} \ldots \overline{X_k}\}$. Let us also assume that each record $\overline{X_i}$ contains the d dimensions which are denoted by $(x_i^1 \ldots x_i^d)$. The following information is maintained about each group of records $\mathcal{G}$:

- For each attribute j, we maintain the sum of corresponding values. The corresponding value is given by $\sum_{i=1}^{k} x_i^j$. We denote the corresponding first-order sums by $Fs_j(\mathcal{G})$. The vector of first order sums is denoted by $\overline{Fs(\mathcal{G})}$.
- For each pair of attributes i and j, we maintain the sum of the product of corresponding attribute values. The corresponding sum is given by $\sum_{t=1}^{k} x_t^i \cdot x_t^j$. We denote the corresponding second order sums by $Sc_{ij}(\mathcal{G})$. The vector of second order sums is denoted by $\overline{Sc(\mathcal{G})}$.
- We maintain the sum of the privacy levels of the records in the group. This number of denoted by $Ps(\mathcal{G})$.

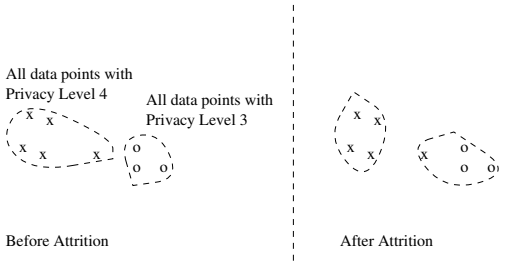

Figure 1: The efficiency of Mixing Different Privacy Levels

- We maintain the total number of records k in that group. This number is denoted by $n(\mathcal{G})$.

The following facts are true about the records in a given group.

OBSERVATION 2.1. *The mean value of attribute j in group $\mathcal{G}$ is given by $Fs_j(\mathcal{G})/n(\mathcal{G})$.*

OBSERVATION 2.2. *The covariance between attributes i and j in group $\mathcal{G}$ is given by $Sc_{ij}(\mathcal{G})/n(\mathcal{G}) - Fs_i(\mathcal{G}) \cdot Fs_j(\mathcal{G})/n(\mathcal{G})^2$.*

We note that the algorithm for group construction must try to put each record in a group which is at least equal to the maximum privacy level of any record in the group. A natural solution is to first classify the records based on their privacy levels and then independently create the groups for varying privacy levels. Unfortunately, this does not lead to the most efficient method for packing the sets of records into different groups. This is because the most effective method for constructing the groups may require us to combine records from different privacy levels. For example, a record with a very low privacy requirement may sometimes naturally be combined with a group of high privacy records in its locality. An attempt to construct a separate group of records with a low privacy requirement may lead to an even higher loss of information. In order to illustrate this point better, we will provide an example.

Consider the set of records illustrated in Figure 1. In this case, there are 3 records with privacy level 3 and 5 records with privacy level 4. One way of grouping the records is to place all the records of privacy level 3 in one group and all records with privacy level 4 in the other. Unfortunately, the group corresponding to privacy level 4 turns out to be ineffective in representing the data. The condensed group utilized from this set of records has poor statistical characteristics, since one of the data points is far removed from the group. Since the condensed statistics of the group does not represent the variations within it, this can lead to an inefficient representation in many cases. In the situation illustrated in Figure 1, it is better to place the outlying record of privacy level 4 into the group with privacy level 3. We also note that it may not be possible to place this outlying record in a group with only two pre-existing members, because of the higher privacy requirement of the record.

First, we need a measure to quantify the effectiveness of a given condensation based approach. In general, this effectiveness is related to the level of compactness with which we can partition the data into different groups. As a goal, this compactness is not very different from the aim of most clustering algorithms. However, the difference here is that there are several constraints on the cardinality of the data points in each group as

Algorithm *ConstructGroups(Level: $MaxPrivacyLevel$, Database: $\mathcal{D}$)*;
begin
  $p = 2$;
  $\mathcal{H}_1$ = Groups from singleton points in $\mathcal{D}_1$;
  while $(p \leq MaxPrivacyLevel)$ **do**
    begin
      $\mathcal{H}_p = Segment(\mathcal{D}_p, p)$;
      $(\mathcal{H}_{p-1}, \mathcal{H}_p) = Cannibalize(\mathcal{H}_{p-1}, \mathcal{H}_p))$;
      $(\mathcal{H}_{p-1}, \mathcal{H}_p) = Attrition(\mathcal{H}_{p-1}, \mathcal{H}_p)$;
      $\mathcal{H}_p = \mathcal{H}_p \cup \mathcal{H}_{p-1}$;
      $p = p + 1$;
    end;
end

Figure 2: The Process of Group Construction for Privacy Preserving Data Mining

Algorithm *Segment(Database: $\mathcal{D}_p$, Privacy level: p)*
begin
  while $\mathcal{D}_p$ contains at least p data points;
    begin
    Sample a data point $\overline{X}$ from $\mathcal{D}_p$;
    Find the $(p-1)$ data points closest to $\overline{X}$ in $\mathcal{D}_p$;
    Create a group $\mathcal{G}$ of p data points comprising $\overline{X}$ and the $p-1$ other closest data points;
    Add $\mathcal{G}$ to the set of groups $\mathcal{H}$;
    end
  Assign remaining data points in $\mathcal{D}_p$ to closest groups;
end

Figure 3: Group Segmentation

Algorithm *Cannibalize(Groups: $\mathcal{H}_{p-1}$, $\mathcal{H}_p$)*;
begin
  for each group $\mathcal{G} \in \mathcal{H}_{p-1}$ **do**
    begin
      for each point in $\mathcal{G}$ perform temporary
      assignment to closest group in $\mathcal{H}_p$;
      if (SSQ of temporary assignment is lower) or
       ($\mathcal{H}_{p-1}$ contains fewer than $(p-1)$ members),
        then make assignment permanent
          else keep old assignment;
    end
end

Figure 4: Cannibalization Algorithm

Algorithm *Attrition(Groups: $\mathcal{H}_{p-1}$, $\mathcal{H}_p$,
    Privacy Level: p)*;
begin
for each data point $\overline{X}$ in $\mathcal{H}_p$ **do**
  begin
    $Distc(\overline{X}, p)$ = Distance of $\overline{X}$ to
      centroid of its *current* group in $\mathcal{H}_p$;
    $Disto(\overline{X}, p-1)$ = Distance of $\overline{X}$ to
      centroid of its *closest* viable group in $\mathcal{H}_{p-1}$;
    $Improve(\overline{X}) = Distc(\overline{X}, p) - Disto(\overline{X}, p-1)$;
  end;
for each group in $\mathcal{H}_p$ with
      at least $p' > p$ points **do**
  begin
    find (if any) the at most $(p'-p)$ data points
      with largest value of $Improve(\cdot)$ function
        which is larger than 0;
  Assign these at most $(p'-p)$ points to their
      corresponding closest groups in $\mathcal{H}_{p-1}$;
end

Figure 5: Attrition Algorithm

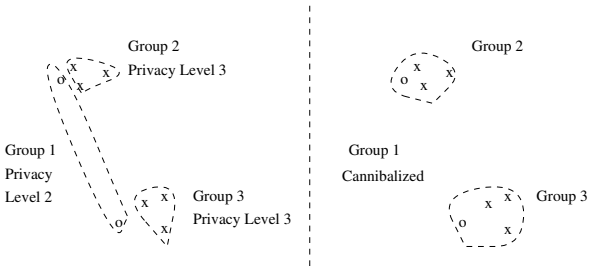

Figure 6: An example of Cannibalization

well as the identity of the data points which can be added to a group with given cardinality. Thus, for the process of quantification of the condensation quality, we simply use the square sum error of the data points in each group. While the privacy level of a group is determined by the number of records in it, the information loss is defined by the average variance of the records about their centroid. We will refer to this quantity as the Sum Squared Error (SSQ).

The method of group construction is different depending upon whether an entire database of records is available or whether the data records arrive in an incremental fashion. We will discuss two approaches for construction of class statistics. The first approach is utilized for the case when the entire database of records is available. The second approach is utilized in an incremental scheme in which the data points arrive one at a time. First, we will discuss the static case in which the entire database of records is available.

The essence of the static approach is to construct the groups using an iterative method in which the groups are processed with increasing privacy level. The overall process of group construction is illustrated in Figure 2. The input to the algorithm is the database $\mathcal{D}$ and the maximum privacy level which is denoted by $MaxPrivacyLevel$. We assume that the segment of the database with privacy level requirement of p is denoted by $\mathcal{D}_p$. We also assume that the set of groups with privacy level of p is denoted by $\mathcal{H}_p$. We note that the database $\mathcal{D}_1$ consists of the set of points which have no privacy constraint at all. Therefore, the group $\mathcal{H}_1$ comprises of the singleton items from the database $\mathcal{D}_1$.

Next, we construct the statistics of the groups in $\mathcal{H}_p$ using an iterative algorithm. In each iteration, we increase the privacy level p by 1, and construct the condensed groups $\mathcal{H}_p$ which have privacy level p. The first step is to construct the group $\mathcal{H}_p$ by using a purely segmentation based process. This process is denoted by *Segment* in Figure 2. This segmentation process is a straightforward iterative approach. In each iteration, a record $\overline{X}$ is sampled from the database $\mathcal{H}_p$. The closest $(p-1)$ records to this individual record $\overline{X}$ are added to this group. Let us denote this group by $\mathcal{G}$. The statistics of the p records in $\mathcal{G}$ are computed. Next, the p records in $\mathcal{G}$ are removed from $\mathcal{D}_p$. The process is repeated iteratively, until the database $\mathcal{D}_p$ is empty. We note that at the end of the process, it is possible that between 1 and $(p-1)$ records may remain. These records can be added to their nearest sub-group in the data. Thus, a small number of groups in the data may contain larger than p data points. The segmentation procedure is illustrated in Figure 3.

Once the segmentation procedure has been performed, we apply the process of *Attrition* and *Cannibalize* in order to further reduce the level of information loss without compromising on the privacy requirements. The purpose of the *Cannibalize* procedure is slightly different. In this procedure, we intend to cannibalize some of the groups in $\mathcal{H}_{p-1}$ and reassign their data points to better fitting groups in $\mathcal{H}_p$. Consider the example illustrated in Figure 6. In this case, we have illustrated three groups. One of the groups (containing two points) has privacy level of two, and another group (containing three points) has privacy level of three. However, the group with privacy level two does not form an natural cluster of data points. In such a case, it may be desirable to break up the group with privacy level 2 and assign one point each to the groups with privacy level 3. Thus, cannibalization is performed when the group $\mathcal{G} \in \mathcal{H}_{p-1}$ does not form a natural cluster. In such cases, it is more effective to cannibalize the group $\mathcal{G}$ and reassign its group members to one or more clusters in $\mathcal{H}_p$. Another example of a situation when cannibalization is desirable is when $\mathcal{H}_{p-1}$ has fewer than $(p-1)$ members. Such a situation arises in situations in which there are very few records for a given privacy level. Consequently, it is not possible to create a group containing only the points at a particular privacy level. We refer to this test for cannibalization as the *numerical test*.

If the group passes the numerical test, we perform an additional *qualitative* test to see if cannibalization should be performed. In order to test whether the cannibalization procedure should be performed, we calculate the SSQ of the regrouping when a temporary assignment of the data points in $\mathcal{G}$ is performed to one or more groups in $\mathcal{H}_p$. If the SSQ of the resulting assignment is lower, then we make this assignment permanent. The pseudo-code for the cannibalization process is illustrated in Figure 4. By performing this operation, the appropriate privacy level of all data points is maintained. This is because the cannibalization process only assigns data points to groups with higher privacy level. Therefore, the assigned data points find themselves in a group with at least their corresponding required privacy.

We note that some groups in $\mathcal{H}_p$ may sometimes contain more than p data points. This is due to the effects of the *Segment* and *Cannibalize* procedures discussed earlier. The idea in the *Attrition* procedure is to move these excess points to a better fitting group in $\mathcal{H}_{p-1}$. The movement of these excess points is likely to improve the quality of data representation in terms of reducing the level of information loss. An example of such a case is illustrated in Figure 1. In this case, the group with five data points contains one record which does not fit very well with the rest of the group. In such a case, the reassignment of the data point to a group with privacy level 3 results in a more compact representation. We note that the reassigned data point has privacy level 4. However, the reassignment process results in the group with privacy level 3 containing 4 data points. Therefore, even though the data point with privacy level 4 was assigned to a group with lower privacy level, the resulting group continues to maintain the desired level of privacy for the reassigned data point. For this purpose, during the attrition process we consider only those groups which are *viable* for reassignment. For a group to be considered viable, it must contain at least as many data points as the privacy level (after the assignment). Furthermore, for a group $\mathcal{G}$ containing p' data points and with privacy level p, we can remove at most $(p'-p)$ data points from it without disturbing the privacy level of the remaining group. In order to perform the actual reassignment, we calculate a function called $Improve(\overline{X})$ for each data point $\overline{X} \in \mathcal{G}$. The value of $Improve(\overline{X})$ is defined to be difference between the distance of $\overline{X}$ from its closest viable centroid and the distance from its current centroid. Clearly, the reassignment of the data point $\overline{X}$ to another group is useful only when the value of $Improve(X)$ is larger than 0. We re-assign the at most $(p'-p)$ data points with largest value of $Improve(\cdot)$, provided that the value of $Improve(\cdot)$ for each of these data points is larger than 0. The overall attrition procedure is illustrated in Figure 5.

The processes of segmentation, cannibalization and attrition are applied iteratively to the segment $\mathcal{D}_p$ of the database for each value of the privacy level p. The value of p is incremented by 1 in each iteration up to the maximum privacy level. The set of groups constructed at this point are returned as the final condensation.

Once the condensed statistics have been constructed, anonymized data can be generated as discussed in [2]. The anonymized data is generated using the statistical properties which can be derived from the group. While this new set of points resembles the original data distribution, it maintains the privacy of the data. The process of anonymized group construction is achieved by first constructing a $d * d$ covariance matrix for each group $\mathcal{G}$. This matrix is denoted by $C(\mathcal{G})$. The ijth entry of the co-variance matrix is the co-variance between the attributes i and j of the set of records in $\mathcal{G}$. The eigenvectors of this co-variance matrix are determined by decomposing the matrix $C(\mathcal{G})$ in the following form:

$$C(\mathcal{G}) = P(\mathcal{G}) \cdot \Delta(\mathcal{G}) \cdot P(\mathcal{G})^T \tag{2.1}$$

The columns of $P(\mathcal{G})$ are the eigenvectors of $C(\mathcal{G})$. The diagonal entries $\lambda_1(\mathcal{G}) \ldots \lambda_d(\mathcal{G})$ of $\Delta(\mathcal{G})$ represent

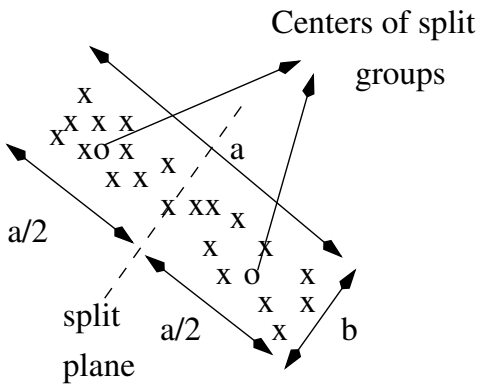

Figure 7: Splitting Group Statistics (Illustration)

the corresponding eigenvalues. It can be shown that the eigenvectors of a covariance matrix form an ortho-normal axis system. This ortho-normal axis-system represents the directions along which the second order correlations are zero. If the data were represented using this ortho-normal axis system, then the covariance matrix would be the diagonal matrix corresponding to $\Delta(\mathcal{G})$. The diagonal entries of $\Delta(\mathcal{G})$ represent the variances along the individual dimensions in this new axis system. We can assume without loss of generality that the eigenvalues $\lambda_1(\mathcal{G}) \ldots \lambda_d(\mathcal{G})$ are ordered in decreasing magnitude. The corresponding eigenvectors are denoted by $e_1(\mathcal{G}) \ldots e_d(\mathcal{G})$. The anonymized data for each group is reconstructed assuming that the data within each group is independently and uniformly distributed along the different eigenvectors. Furthermore, the variance of the distribution along each eigenvector is equal to the corresponding eigenvalue. These approximations are reasonable when only a small spatial locality is used.

3 Dynamic Maintenance of Groups

The process of dynamic maintenance of groups is useful in a variety of settings such as that of data streams. In the process of dynamic maintenance, the points in the data stream are processed incrementally. It is assumed that a set $\mathcal{S}$ of the data points (denoted by $InitNumber$) are available at the beginning of the process. The static process *ConstructGroups* is applied to this set $\mathcal{S}$. Once the initial groups have been constructed, a dynamic process of group maintenance is applied in order to maintain the condensed groups of varying privacy levels.

The incremental algorithm works by using a nearest neighbor approach. When an incoming data point $\overline{X_i}$ is received, we find the closest cluster to it using the distance of the data point $\overline{X_i}$ to the different centroids. While it is desirable to add $\overline{X_i}$ to its closest centroid, we cannot add $\overline{X_i}$ to a given cluster which has fewer than $p(i)-1$ data points in it. Therefore, the data point $\overline{X_i}$ is added to the closest cluster which also happens to have at least $p(i)-1$ data points inside it.

In general, it is not desirable to have groups with high sizes compared to their constituent privacy levels. When such a situation arises, it effectively means that a higher level of representational inaccuracy is created than is really necessary with the privacy requirements of the points within the group. The average privacy level of the group $\mathcal{G}$ can be computed from the condensed statistics. This number is equal to $Ps(\mathcal{G})/n(\mathcal{G})$. This is because $Ps(\mathcal{G})$ is equal to the sum of the privacy levels of the data points in the group.

The split criterion used by our algorithm is that a group is divided when the number of items in the group is more than twice the average privacy level of the items in the group. Therefore, the group is split when the following holds true:

$$n(\mathcal{G}) \geq 2 \cdot Ps(\mathcal{G})/n(\mathcal{G}) \tag{3.2}$$

As in the case of anonymized data construction, we utilize the uniformity assumption in order to split the group statistics. In each case, the group is split along the eigenvector with the largest eigenvalue. This also corresponds to the direction with the greatest level of variance. This is done in order to reduce the overall variance of the resulting clusters and ensure the greatest compactness of representation. An example of this case is illustrated in Figure 7. We assume without loss of generality that the eigenvector $\overline{e_1}$ with the lowest index is the chosen direction the split. The corresponding eigenvalue is denoted by λ_1. Since the variance of the data along $\overline{e_1}$ is λ_1, then the range (a) of the corresponding uniform distribution along $\overline{e_1}$ is given[1] by $a = \sqrt{12 \cdot \lambda_1}$.

In such a case, the original group of size $2 \cdot k$ is split into two groups of equal size. We need to determine the first order and second order statistical data about each of the split groups $\mathcal{M}_1$ and $\mathcal{M}_2$. We assume that the privacy component $Ps(\mathcal{G})$ is also equally divided between the two groups. We first derive the centroid and eigenvector directions for each group. These values are sufficient to reconstruct the values of $Fs_i(\mathcal{G})$ and $Sc_{ij}(\mathcal{G})$ about each group.

Assume that the centroid of the unsplit group $\mathcal{M}$ is denoted by $\overline{Y(\mathcal{M})}$. This centroid can be computed

[1] This calculation was done by using the formula for the standard deviation of a uniform distribution with range a. The corresponding standard deviation is given by $\sqrt{a/12}$.

from the first order values $\overline{Fs(\mathcal{M})}$ as follows:

$$\overline{Y(\mathcal{M})} = (Fs_1(\mathcal{M}), \ldots Fs_d(\mathcal{M}))/n(\mathcal{G}) \quad (3.3)$$

Once the centroid has been computed, those of each of the split groups can be computed as well. From Figure 7, it is easy to see that the centroids of each of the split groups $\mathcal{M}_1$ and $\mathcal{M}_2$ are given by $\overline{Y(\mathcal{M})} - (a/4) \cdot \overline{e_1}$ and $Y(\mathcal{M}) + (a/4) \cdot \overline{e_1}$ respectively. By substituting $a = \sqrt{12 \cdot \lambda_1}$, it is easy to see that the new centroids of the groups $\mathcal{M}_1$ and $\mathcal{M}_2$ are given by $\overline{Y(\mathcal{M})} - (\sqrt{12 \cdot \lambda_1}/4) \cdot \overline{e_1}$ and $\overline{Y(\mathcal{M})} + (\sqrt{12 \cdot \lambda_1}/4) \cdot \overline{e_1}$ respectively.

We will now discuss how to compute the second order statistical values. The first step is the determination of the covariance matrix of the split groups. Let us assume that the ijth entry of the co-variance matrix for the group $\mathcal{M}_1$ is given by $C_{ij}(\mathcal{M}_1)$. We also note that the eigenvectors of $\mathcal{M}_1$ and $\mathcal{M}_2$ are identical to the eigenvectors of $\mathcal{M}$, since the directions of zero correlation remain unchanged by the splitting process. Therefore, we have:

$$\begin{aligned} e_1(\mathcal{M}_1) &= e_1(\mathcal{M}_2) = e_1(\mathcal{M}) \\ e_2(\mathcal{M}_1) &= e_2(\mathcal{M}_2) = e_2(\mathcal{M}) \\ e_3(\mathcal{M}_1) &= e_3(\mathcal{M}_2) = e_3(\mathcal{M}) \\ &\ldots \\ e_d(\mathcal{M}_1) &= e_d(\mathcal{M}_2) = e_d(\mathcal{M}) \end{aligned}$$

The eigenvalue (in the split groups $\mathcal{M}_1$ and $\mathcal{M}_2$) corresponding to $\overline{e_1}(\mathcal{M})$ is equal to $\lambda_1/4$. This is because the splitting process along $\overline{e_1}$ reduces the corresponding variance by a factor of 4. Other eigenvalues remain unchanged. Let $P(\mathcal{M})$ represent the eigenvector matrix of $\mathcal{M}$, and $\Delta(\mathcal{M})$ represent the corresponding diagonal matrix. Then, the new diagonal matrix $\Delta(\mathcal{M}_1) = \Delta(\mathcal{M}_2)$ of $\mathcal{M}_1$ can be derived by dividing the entry $\lambda_1(\mathcal{M})$ by 4. Therefore, we have:

$$\lambda_1(\mathcal{M}_1) = \lambda_1(\mathcal{M}_2) = \lambda_1(\mathcal{M})/4$$

The other eigenvalues of $\mathcal{M}_1$ and $\mathcal{M}_2$ remain the same:

$$\begin{aligned} \lambda_2(\mathcal{M}_1) &= \lambda_2(\mathcal{M}_2) = \lambda_2(\mathcal{M}) \\ \lambda_3(\mathcal{M}_1) &= \lambda_3(\mathcal{M}_2) = \lambda_3(\mathcal{M}) \\ &\ldots \\ \lambda_d(\mathcal{M}_1) &= \lambda_d(\mathcal{M}_2) = \lambda_d(\mathcal{M}) \end{aligned}$$

Thus, the (identical) co-variance matrixes of $\mathcal{M}_1$ and $\mathcal{M}_2$ may be determined as follows:

$$C(\mathcal{M}_1) = P(\mathcal{M}_1) \cdot \Delta(\mathcal{M}_1) \cdot P(\mathcal{M}_1)^T$$

From Observation 2.2, it is clear that the second order statistics of $\mathcal{M}_1$ may be determined as follows:

$$\begin{aligned} &Sc_{ij}(\mathcal{M}_1) = \\ &\quad k \cdot C_{ij}(\mathcal{M}_1) + Fs_i(\mathcal{M}_1) \cdot Fs_j(\mathcal{M}_1)/k \end{aligned}$$

An important observation is that even though the covariance matrices of $\mathcal{M}_1$ and $\mathcal{M}_2$ are identical, the values of $Sc_{ij}(\mathcal{M}_1)$ and $Sc_{ij}(\mathcal{M}_2)$ are different because of different first order aggregates substituted in the above formula for $Sc_{ij}(\mathcal{M}_1)$. The overall process for splitting the group statistics is illustrated in Figure 7. Another interesting point to be noted is that the entire purpose of splitting is to keep groups sizes sufficiently compact for data mining algorithms. The process of splitting itself can *never* result in the violation of the privacy condition, since the split group is based on a split of the *statistics*, but not of the data points themselves. In order to understand this point, let us consider the following "example" of a case where the split condition seems to violate privacy. Consider a group having 5 tuples, the privacy constraints of the tuples being 2, 2, 2, 3, 5 respectively. The group does not split because $5 < 2 * 14/5$. Now, if a new tuple having privacy constraint 3 wants to join the group, the splitting condition is satisfied since $6 > 2 * 17/6$. Hence each of the split group corresponds to statistics of 3 data points. Therefore, it would apparently seem that the privacy of the tuple with requirement 5 has been violated. This is *not* the case since we split the *statistics* into two *pseudo-groups* of 3 points each, rather than actually split the *points* themselves. The process of performing the split partitions the statistics based on a *probability distribution assumption* (uniform distribution) rather than using the actual points themselves (which have already been lost in the merged statistics). The tuple with privacy condition 5 may contribute to the statistics of both groups, when the splitting condition is used. Each pseudo-group thus has a privacy level as high as the unsplit group, from the perspective of the *old data points* in it, but at the same time we would need to use the size of the group while considering the addition of *further data points* into the smaller pseudo-groups.

In order to test the quality of our results we applied our approach to a nearest neighbor classifier. In the classification process, the condensation process was performed separately for each class. In the next section, we will discuss the behavior of this nearest neighbor classifier.

4 Empirical Results

We tested the privacy preserving approach over a wide range of data sets and metrics. An important question which arises in the context of a privacy preserving

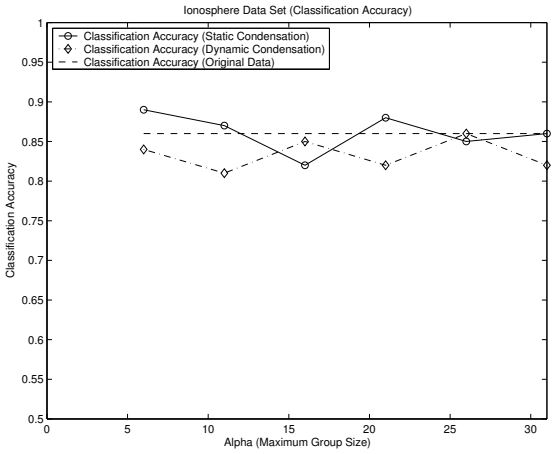

Figure 8: Accuracy of Classifier with Increasing Privacy Level (Ionosphere Data Set)

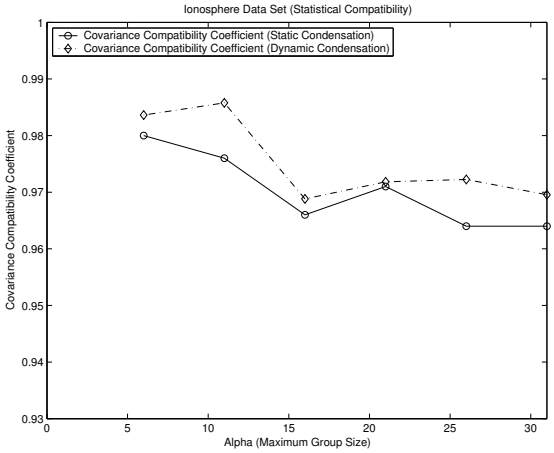

Figure 9: Covariance Compatibility of Condensed Data Set with Increasing Privacy Level (Ionosphere Data Set)

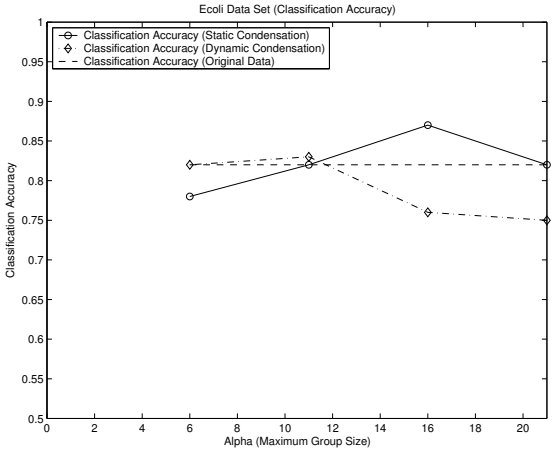

Figure 10: Accuracy of Classifier with Increasing Privacy Level (Ecoli Data Set)

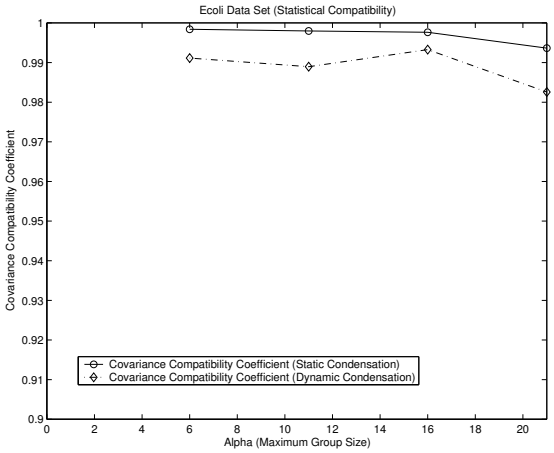

Figure 11: Covariance Compatibility of Condensed Data Set with Increasing Privacy Level (Ecoli Data Set)

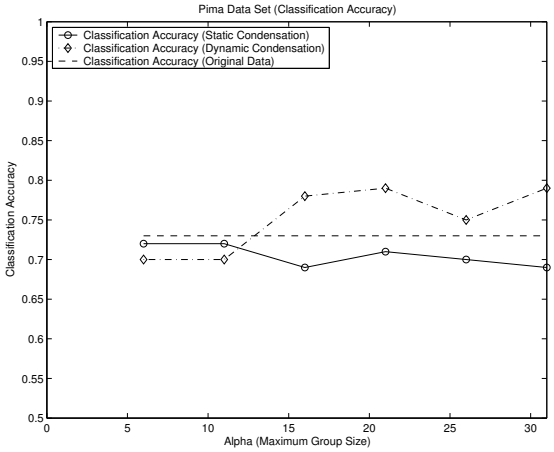

Figure 12: Accuracy of Classifier with Increasing Privacy Level (Pima Indian Data Set)

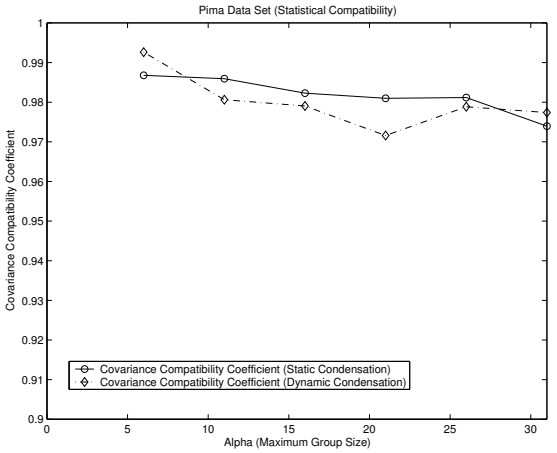

Figure 13: Covariance Compatibility of Condensed Data Set with Increasing Privacy Level (Pima Indian Data Set)

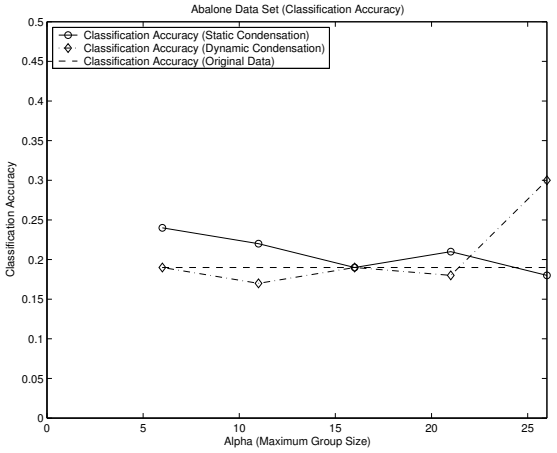

Figure 14: Accuracy of Classifier with Increasing Privacy Level (Abalone Data Set)

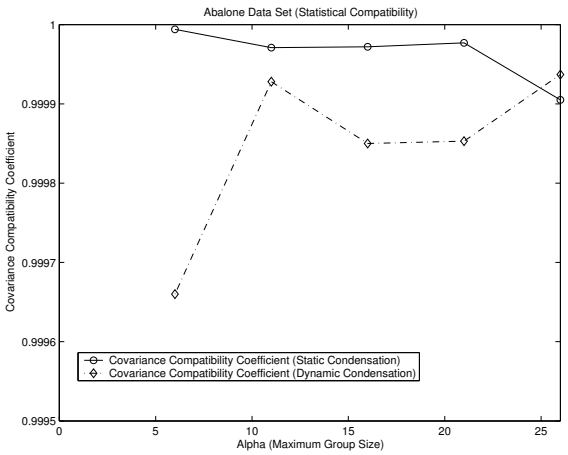

Figure 15: Covariance Compatibility of Condensed Data Set with Increasing Privacy Level (Abalone Data Set)

approach is the nature of the metric to be used in order to test the quality of the approach. The first step is to test the nature of the tradeoff between increased levels of privacy, and the resulting information loss. While the level of privacy is controlled by the average condensed group size, the information loss is measured indirectly in terms of the effect of the perturbation on the quality of data mining algorithms. We tested the accuracy of a simple k-nearest neighbor classifier with the use of different levels of privacy. The minimum privacy level of each data point was generated from a (discrete) uniform distribution in the range $[\alpha - \beta, \alpha]$. By changing the value of α it is possible to vary the level of privacy during the condensation process. The aim of our approach is to show that a high level of privacy can be achieved without significantly compromising accuracy.

Another useful metric for testing the quality of the privacy preserving process arises from the level of matching between the original and perturbed data. This provides insight into the nature of the relationship between the original data set and perturbed data set. The first step is therefore to identify the statistics used for testing the effectiveness of the perturbation process. One simple method is to test how the covariance structure of the perturbed data set matched with the original data set. This is because the covariance structure of the data identifies the essential data properties up to a second order approximation. If the newly created data set has very similar data characteristics to the original data set, then the condensed data set is a good substitute for most data mining algorithms. For each dimension pair (i, j), let the corresponding entries in the covariance matrix for the original and the perturbed data be denoted

by o_{ij} and p_{ij} respectively. We computed the statistical coefficient of correlation between the data entry pairs (o_{ij}, p_{ij}). Let us denote this value by μ. When the two matrices are identical, the value of μ is 1. On the other hand, when there is perfect negative correlation between the entries, the value of μ is -1.

A number of real data sets from the UCI machine learning repository[2] were used for the testing. We used the Ionosphere, Ecoli, Pima Indian and Abalone data sets. The last data set was a regression modeling problem, and therefore the classification measure needed to be redefined. For this problem, the classification accuracy measure used was the percentage of the time that the age was predicted within an accuracy of less than one year by the nearest neighbor classifier. In many cases, the number of data points for a given privacy level for lower than the numerical value of the privacy level itself. In such cases, the mixing of data points for different privacy levels is inevitable. Thus, the condensation process could not have been performed for such cases using the homogeneous k -anonymity model or k-indistinguishability model [2, 18].

The results on classification accuracy for the Ionosphere, Ecoli, Pima Indian, and Abalone data sets are illustrated in Figures 8, 10, 12 and 14 respectively. The value of β was fixed to 4, whereas the value of α was varied over the different data sets. The range of values of α is determined by the number of data points in the particular data set at hand. This value of α is illustrated on the X-axis. On the Y-axis, we have plotted the classification accuracy of the nearest neighbor classifier, when the condensation technique was used. For each graph, we have illustrated the results using both static and dynamic condensation. In addition, a baseline is marked on each graph. This baseline is a horizontal line on the graph which shows the classification accuracy using the original data. It is clear that in most cases, the accuracy of classification reduced with increasing group size. This is a natural tradeoff because a greater amount of privacy is achieved with larger groups sizes. At the same time, it leads to a higher amount of information loss.

In many cases, the quality of the classification improved because of the condensation process. in most cases. While the aim of our approach was to provide a high level of privacy without losing information, it appears that the process of condensation itself actually helped in removing the anomalies in the data for the purpose of classification. This phenomenon is likely to be helpful over a number of different data mining problems in which the aggregate behavior of the data is exposed by the condensation process.

[2] $http://www.ics.uci.edu/\tilde{}mlearn$

Furthermore, the static condensation approach provided higher quality results than the dynamic technique. This is because the splitting algorithm of the dynamic condensation process introduced an additional level of approximation into the data representation. The splitting procedure assumed a uniform distribution of the data within a condensed group of data points. The accuracy of this approximation reduces when group sizes are small. In such cases, there are simply too few data points to make an accurate estimation of the values of split group statistics. Thus, the use of the uniform distribution approximation reduces the quality of the covariance statistics in the split groups for small group sizes. For this reason, the dynamic condensation process was sometimes less effective than the static condensation approach. However, in all cases, the dynamic condensation approach worked almost as effectively as the classifier on the original data. One notable exception to the general advantage of the static condensation process was the behavior on the Pima Indian data set. In this case, the dynamic condensation process provided results of higher quality for larger group sizes. The reason for this was that the splitting process seemed to improve the quality of the classification. The data set seemed to contain a number of anomalies. These anomalies were removed by the splitting process. This resulted in a higher classification accuracy of the dynamic approach.

We also compared the covariance characteristics of the data sets. The results are illustrated in Figures 9, 11, 13 and 15 respectively. For most data sets, the value of the statistical correlation is almost perfect. This corresponds to the fact that the correlation values was larger than 0.95 in most cases. For some examples such as the Abalone data set (illustrated in Figure 15), the covariance compatibility value was larger than 0.99. These results emphasize the fact that the perturbed data is similar to the original data in terms of its statistical structure. As in the previous case, the results for the case of static condensation were better than those for dynamic condensation. This is again because of the additional inaccuracy introduced by the splitting process. In all cases, the absolute correlation provided by the scheme was very high. In the dynamic case, the correlation coefficient tended to drop for small group sizes. The only exception to this general rule was the ionosphere data set in which the covariance compatibility values were slightly lower for the static case. The covariance compatibility also reduced for extremely large group sizes. This is because in such a case, the pseudo-data no longer represents a particular data locality well. Thus, the covariance compatibility was highest in those cases in which the data contained tight clusters comprising a relatively modest number of

data points. This is because of the following reasons:

- When the number of points in each cluster were large, the accuracy of the uniform distribution assumption during the splitting process is maintained.
- When the clusters are tight, these data points represent a small spatial locality with respect to the rest of the data set. An approximation in a small spatial locality does not significantly affect the overall correlation structure.

We note that the process of representing a small spatial locality in a group and that of representing a larger number of data points in a group are two competing and contradictory goals. It is important to pick a balance between the two, since this tradeoff defines the quality of performance on the underlying data mining algorithm. This balance is externally defined, since the average group size is determined by the privacy requirements of the users. In general, since our approach continued to be as effective as the base classification accuracy over a wide range of group sizes, this illustrates the effectiveness of our methodology in most practical scenarios.

5 Conclusions and Summary

In this paper, we discussed a scheme for privacy preserving data mining in which the data points are allowed to have variable privacy levels. This is useful in a number of applications in which different records have inherently different privacy requirements. We propose a method for privacy protection in a data stream environment using condensed statistics of the data set. These condensed statistics can either be generated statically or they can be generated dynamically in a data stream environment. We tested our results on a number of real data sets from the UCI machine learning repository. The results show that our method produces data sets which are quite similar to the original data in structure, and also exhibit similar accuracy results.

References

[1] C. C. Aggarwal, and S. Parthasarathy, *Mining Massively Incomplete Data Sets by Conceptual Reconstruction*, Proceedings of the ACM KDD Conference, (2001), pp. 227–232,

[2] C. C. Aggarwal, and P. S. Yu, *A Condensation Based Approach to Privacy Preserving Data Mining*, Proceedings of the EDBT Conference, (2004), pp. 183–199.

[3] D. Agrawal, and C. C. Aggarwal, *On the Design and Quantification of Privacy Preserving Data Mining Algorithms*, Proceedings of the ACM PODS Conference, (2002).

[4] R. Agrawal, and R. Srikant, *Privacy Preserving Data Mining*, Proceedings of the ACM SIGMOD Conference, (2000).

[5] P. Benassi, *Truste: An online privacy seal program*, Communications of the ACM, 42(2), (1999), pp. 56-59.

[6] C. Clifton, and D. Marks, *Security and Privacy Implications of Data Mining*, ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, (1996), pp. 15–19.

[7] J. Vaidya, and C. Clifton, *Privacy Preserving Association Rule Mining in Vertically Partitioned Data*, ACM KDD Conference, (2002).

[8] T. M. Cover, J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Inc., New York, (1991).

[9] Cranor L. F. (Ed.) *Special Issue on Internet Privacy*, Communications of the ACM, 42(2), (1999).

[10] The Economist, *The End of Privacy*, (1999).

[11] V. Estivill-Castro, and L. Brankovic, *Data Swapping: Balancing privacy against precision in mining for logic rules*, Data Warehousing and Knowledge Discovery, Springer-Verlag, Lecture Notes in Computer Science 1676, (1999), pp. 389–398.

[12] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, *Privacy Preserving Mining Of Association Rules*, ACM KDD Conference, (2002).

[13] A. Hinneburg, and D. A. Keim, *An Efficient Approach to Clustering in Large Multimedia Databases with Noise*, ACM KDD Conference, (1998).

[14] V. S. Iyengar, *Transforming Data To Satisfy Privacy Constraints*, ACM KDD Conference, (2002).

[15] C. K. Liew, U. J. Choi, and C. J. Liew, *A data distortion by probability distribution*, ACM TODS Journal, (1985), 10(3) pp. 395–411.

[16] T. Lau, O. Etzioni, and D. S. Weld, *Privacy Interfaces for Information Management*, Communications of the ACM, 42(10), (1999), pp. 89–94.

[17] S. Murthy, *Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey*, Data Mining and Knowledge Discovery, 2, (1998), pp. 345–389.

[18] P. Samarati, and L. Sweeney, *Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement Through Generalization and Suppression.* Proceedings of the IEEE Symposium on Research in Security and Privacy, (1998).

[19] S. L. Warner, *Randomized Response: A survey technique for eliminating evasive answer bias*, Journal of the American Statistical Association, 60(309), (1965), pp. 63-69.