A Condensation Approach to Privacy Preserving Data Mining

Charu C. Aggarwal and Philip S. Yu

IBM T. J. Watson Research Center, 19 Skyline Drive,
Hawthorne, NY 10532
{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. In this paper, we propose a new framework for privacy preserving data mining of multi-dimensional data. Previous work for privacy preserving data mining uses a perturbation approach which reconstructs data distributions in order to perform the mining. Such an approach treats each dimension independently and therefore ignores the correlations between the different dimensions. In addition, it requires the development of a new distribution based algorithm for each data mining problem, since it does not use the multi-dimensional records, but uses aggregate distributions of the data as input. This leads to a fundamental re-design of data mining algorithms. In this paper, we will develop a new and flexible approach for privacy preserving data mining which does not require new problem-specific algorithms, since it maps the original data set into a new anonymized data set. This anonymized data closely matches the characteristics of the original data including the correlations among the different dimensions. 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. For example, a simple transaction such as using the credit card results in automated storage of information about user buying behavior. 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. This has resulted to a considerable amount of focus on privacy preserving data collection and mining methods in recent years [1], [2], [3], [4], [6], [8], [9], [12], [13].

E. Bertino et al. (Eds.): EDBT 2004, LNCS 2992, pp. 183–199, 2004.

A perturbation based approach to privacy preserving data mining was pioneered in [1]. This technique relies on two facts:

- Users are not equally protective of all values in the records. Thus, users may be willing to provide modified values of certain fields by the use of a (publically known) perturbing random distribution. This modified value may be generated using custom code or a browser plug in.
- Data Mining Problems do not necessarily require the individual records, but only distributions. Since the perturbing distribution is known, it can be used to reconstruct *aggregate* distributions. This aggregate information may be used for the purpose of data mining algorithms. An example of a classification algorithm which uses such aggregate information is discussed in [1].

Specifically, let us consider a set of n original data values $x_1 \ldots x_n$. These are modelled in [1] as n independent values drawn from the data distribution X. In order to create the perturbation, we generate n independent values $y_1 \ldots y_n$, each with the same distribution as the random variable Y. Thus, the perturbed values of the data are given by $x_1 + y_1, \ldots x_n + y_n$. Given these values, and the (publically known) density distribution f_Y for Y, techniques have been proposed in [1] in order to estimate the distribution f_X for X. An iterative algorithm has been proposed in the same work in order to estimate the data distribution f_X. A convergence result was proved in [2] for a refinement of this algorithm. In addition, the paper in [2] provides a framework for effective quantification of the effectiveness of a (perturbation-based) privacy preserving data mining approach.

We note that the perturbation approach results in some amount of information loss. The greater the level of perturbation, the less likely it is that we will be able to estimate the data distributions effectively. On the other hand, larger perturbations also lead to a greater amount of privacy. Thus, there is a natural trade-off between greater accuracy and loss of privacy.

Another interesting method for privacy preserving data mining is the k-anonymity model [18]. In the k-anonymity model, domain generalization hierarchies are used in order to transform and replace each record value with a corresponding generalized value. We note that the choice of the best generalization hierarchy and strategy in the k-anonymity model is highly specific to a particular application, and is in fact dependent upon the user or domain expert. In many applications and data sets, it may be difficult to obtain such precise domain specific feedback. On the other hand, the perturbation technique [1] does not require the use of such information. Thus, the perturbation model has a number of advantages over the k-anonymity model because of its independence from domain specific considerations.

The perturbation approach works under the strong requirement that the data set forming server is not allowed to learn or recover precise records. This strong restriction naturally also leads to some weaknesses. Since the former method does not reconstruct the original data values but only distributions, new algorithms need to be developed which use these reconstructed distributions in order to perform mining of the underlying data. This means that for each individual data problem such as classification, clustering, or association rule mining, a new *distribution based* data mining algorithm needs to be developed. For example, the work in [1] develops a new distribution based data mining algorithm for the classification problem, whereas the techniques in [9], and [16] develop methods for privacy preserving association rule mining. While some clever approaches have been developed for distribution based mining of data for particular problems such as association rules and classification, it is clear that using distributions instead of original records greatly restricts the range of algorithmic techniques that can be used on the data. Aside from the additional inaccuracies resulting from the perturbation itself, this restriction can itself lead to a reduction of the level of effectiveness with which different data mining techniques can be applied.

In the perturbation approach, the distribution of each data dimension is reconstructed[1] independently. This means that any distribution based data mining algorithm works under an implicit assumption of treating each dimension independently. In many cases, a lot of relevant information for data mining algorithms such as classification is hidden in the inter-attribute correlations [14]. For example, the classification technique in [1] uses a distribution-based analogue of a single-attribute split algorithm. However, other techniques such as multi-variate decision tree algorithms [14] cannot be accordingly modified to work with the perturbation approach. This is because of the independent treatment of the different attributes by the perturbation approach. This means that distribution based data mining algorithms have an inherent disadvantage of loss of implicit information available in multi-dimensional records. It is not easy to extend the technique in [1] to reconstruct multi-variate distributions, because the amount of data required to estimate multi-dimensional distributions (even without randomization) increases exponentially[2] with data dimensionality [17]. This is often not feasible in many practical problems because of the large number of dimensions in the data.

The perturbation approach also does not provide a clear understanding of the level of indistinguishability of different records. For example, for a given level of perturbation, how do we know the level to which it distinguishes the different records effectively? While the k-anonymity model provides such guarantees, it requires the use of domain generalization hierarchies, which are a constraint on their effective use over arbitrary data sets. As in the k-anonymity model, we use an approach in which a record cannot be distinguished from at least k other records in the data. The approach discussed in this paper requires the comparison of a current set of records with the current set of summary statistics. Thus, it requires a relaxation of the strong assumption of [1] that the data set

[1] Both the local and global reconstruction methods treat each dimension independently.

[2] A limited level of multi-variate randomization and reconstruction is possible in sparse categorical data sets such as the market basket problem [9]. However, this specialized form of randomization cannot be effectively applied to a generic non-sparse data sets because of the theoretical considerations discussed.

forming server is not allowed to learn or recover records. However, only aggregate statistics are *stored* or *used* during the data mining process at the server end.

A record is said to be *k-indistinguishable*, when there are at least k other records in the data from which it cannot be distinguished. The approach in this paper re-generates the anonymized records from the data using the above considerations. The approach can be applied to either static data sets, or more dynamic data sets in which data points are added incrementally. Our method has two advantages over the k-anonymity model:
(1) It does not require the use of domain generalization hierarchies as in the k-anonymity model.
(2) It can be effectively used in situations with dynamic data updates such as the data stream problem. This is not the case for the work in [18], which essentially assumes that the entire data set is available apriori.

This paper is organized as follows. In the next section, we will 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 a condensation approach for data mining. This approach uses a methodology which condenses the data into multiple groups of pre-defined size. For each group, a certain level of statistical information about different records is maintained. This statistical information suffices to preserve statistical information about the mean and correlations across the different dimensions. Within a group, it is not possible to distinguish different records from one another. Each group has a certain minimum size k, which is referred to as the *indistinguishability level* of that privacy preserving approach. The greater the indistinguishability level, the greater the amount of privacy. At the same time, a greater amount of information is lost because of the condensation of a larger number of records into a single statistical group entity.

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 $\mathcal{S}$:

- 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. This sum is equal to $\sum_{t=1}^{k} x_t^i \cdot x_t^j$. 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 total number of records k in that group. This number is denoted by $n(\mathcal{G})$.

We make the following simple observations:
Observation 1: *The mean value of attribute j in group $\mathcal{G}$ is given by $Fs_j(\mathcal{G})/n(\mathcal{G})$.*
Observation 2: *The covariance between attributes i and j in group $\mathcal{G}$ is given by $Sc_{ij}(\mathcal{G})/n(\mathcal{G}) - Fs_i(\mathcal{G}) \cdot Fs_j(\mathcal{G})/n(\mathcal{G})^2$.*

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:

- When the entire data set is available and individual subgroups need to be created from it.
- When the data records need to be added incrementally to the individual subgroups.

The algorithm for creation of subgroups from the entire data set is a straightforward iterative approach. In each iteration, a record $\overline{X}$ is sampled from the database $\mathcal{D}$. The closest $(k-1)$ records to this individual record $\overline{X}$ are added to this group. Let us denote this group by $\mathcal{G}$. The statistics of the k records in $\mathcal{G}$ are computed. Next, the k records in $\mathcal{G}$ are deleted from the database $\mathcal{D}$, and the process is repeated iteratively, until the database $\mathcal{D}$ is empty. We note that at the end of the process, it is possible that between 1 and $(k-1)$ 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 k data points. The overall algorithm for the procedure of condensed group creation is denoted by *CreateCondensedGroups*, and is illustrated in Figure 1. We assume that the final set of group statistics are denoted by $\mathcal{H}$. This set contains the aggregate vector $(\overline{Sc(\mathcal{G})}, \overline{Fs(\mathcal{G})}, n(\mathcal{G}))$ for each condensed group $\mathcal{G}$.

2.1 Anonymized-Data Construction from Condensation Groups

We note that the condensation groups represent statistical information about the data in each group. This statistical information can be used to create *anonymized data* which has similar statistical characteristics to the original data set. This is achieved by using the following method:

- A $d * d$ co-variance matrix $C(\mathcal{G})$ is constructed for each group $\mathcal{G}$. 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. These eigenvectors are determined by decomposing the matrix $C(\mathcal{G})$ in the following form:

$$C(\mathcal{G}) = P(\mathcal{G}) \cdot \Delta(\mathcal{G}) \cdot P(\mathcal{G})^T \qquad (1)$$

```
Algorithm CreateCondensedGroups(Indistinguish. Lvl.: $k$,
          Database: $\mathcal{D}$);
begin
  while $\mathcal{D}$ contains at least $k$ points do
  begin
    Randomly sample a data point $\overline{X}$ from $\mathcal{D}$;
    $\mathcal{G} = \{\overline{X}\}$;
    Find the closest $(k-1)$ records to $\overline{X}$ and add to $\mathcal{G}$;
    for each attribute $j$ compute statistics $Fs_j(\mathcal{G})$;
    for each pair of attributes $i, j$ compute $Sc_{ij}(\mathcal{G})$;
    Set $n(\mathcal{G}) = k$;
    Add the corresponding statistics of group $\mathcal{G}$ to $\mathcal{H}$;
    $\mathcal{D} = \mathcal{D} - \mathcal{G}$;
  end;
  Assign each remaining point in $\mathcal{D}$ to the closest group
  and update the corresponding group statistics;
  end
  return($\mathcal{H}$);
end
```

Fig. 1. Creation of Condensed Groups from the Data

> The columns of $P(\mathcal{G})$ represent the eigenvectors of the covariance matrix $C(\mathcal{G})$. The diagonal entries $\lambda_1(\mathcal{G}) \dots \lambda_d(\mathcal{G})$ of $\Delta(\mathcal{G})$ represent the corresponding eigenvalues. Since the matrix is positive semi-definite, the corresponding eigenvectors form an ortho-normal axis system. This ortho-normal axis-system represents the directions along which the second order correlations are removed. In other words, if the data were represented using this ortho-normal axis system, then the covariance matrix would be the diagonal matrix corresponding to $\Delta(\mathcal{G})$. Thus, the diagonal entries of $\Delta(\mathcal{G})$ represent the variances along the individual dimensions. We can assume without loss of generality that the eigenvalues $\lambda_1(\mathcal{G}) \dots \lambda_d(\mathcal{G})$ are ordered in decreasing magnitude. The corresponding eigenvectors are denoted by $e_1(\mathcal{G}) \dots e_d(\mathcal{G})$.

We note that the eigenvectors together with the eigenvalues provide us with an idea of the distribution and the co-variances of the data. In order to re-construct the anonymized data for each group, we assume that the data within each group is independently and uniformly distributed along each eigenvector with a variance equal to the corresponding eigenvalue. The statistical independence along each eigenvector is an extended approximation of the second-order statistical independence inherent in the eigenvector representation. This is a reasonable approximation when only a small spatial locality is used. Within a small spatial locality, we may assume that the data is uniformly distributed without substantial loss of accuracy. The smaller the size of the locality, the better the accuracy of this approximation. The size of the spatial locality reduces when a larger number of groups is used. Therefore, the use of a large number of groups leads to a better overall approximation in each spatial locality. On the other hand, the use of a larger number of groups also reduced the *number* of points in each group. While the use of a smaller spatial locality improves the accuracy of the approximation, the use of a smaller number of *points* affects the accuracy in the opposite direction. This is an interesting trade-off which will be explored in greater detail in the empirical section.

2.2 Locality Sensitivity of Condensation Process

We note that the error of the simplifying assumption increases when a given group does not truly represent a small spatial locality. Since the group sizes are essentially fixed, the level of the corresponding inaccuracy increases in sparse regions. This is a reasonable expectation, since outlier points are inherently more difficult to mask from the point of view of privacy preservation. It is also important to understand that the locality sensitivity of the condensation approach arises from the use of a fixed group size as opposed to the use of a fixed group radius. This is because fixing the group size fixes the privacy (indistinguishability) level over the entire data set. At the same time, the level of information loss from the simplifying assumptions depends upon the characteristics of the corresponding data locality.

3 Maintenance of Condensed Groups in a Dynamic Setting

In the previous section, we discussed a static setting in which the entire data set was available at one time. In this section, we will discuss a dynamic setting in which the records are added to the groups one at a time. In such a case, it is a more complex problem to effectively maintain the group sizes. Therefore, we make a relaxation of the requirement that each group should contain k data

```
Algorithm DynamicGroupMaintenance(Database: $\mathcal{D}$,
  IncrementalStream: $\mathcal{S}$, DistinguishabilityFactor: k)
begin
  $\mathcal{H} = CreateCondensedGroups(k, \mathcal{D})$;
  for each data point $\overline{X}$ received from incremental stream $\mathcal{S}$ do
  begin
    Find the nearest centroid in $\mathcal{H}$ to $\overline{X}$;
    Add $\overline{X}$ to corresponding group statistics $\mathcal{M}$;
    if $n(\mathcal{M}) = 2 \cdot k$ then $(\mathcal{M}_1, \mathcal{M}_2) = SplitGroupStatistics(\mathcal{M}, k)$;
    Delete $\mathcal{M}$ from $\mathcal{H}$;
    Add $\mathcal{M}_1$ to $\mathcal{H}$;
    Add $\mathcal{M}_2$ to $\mathcal{H}$;
  end
end
```

Fig. 2. Overall Process of Maintenance of Condensed Groups

```
Algorithm SplitGroupStatistics(GroupStatistics: $\mathcal{M}$, GroupSize: $k$);
begin
  Determine covariance matrix $C(\mathcal{M})$;
  { The $j, k$th entry of the covariance matrix is determined using the
   formula $C_{jk}(\mathcal{M}) = Sc_{jk}(\mathcal{M})/n(\mathcal{M}) - Fs_j(\mathcal{M}) \cdot Fs_k(\mathcal{M})/n(\mathcal{M})^2$; }
  Determine eigenvectors $\overline{e_1(\mathcal{M})} \ldots \overline{e_d(\mathcal{M})}$ with eigenvalues $\lambda_1(\mathcal{M}) \ldots \lambda_d(\mathcal{M})$;
  { Relationship is $C(\mathcal{M}) = P(\mathcal{M}) \cdot \Delta(\mathcal{M}) \cdot P(\mathcal{M})^T$
      Here $\Delta(\mathcal{M})$ is a diagonal matrix; }
  { Without loss of generality we assume that $\lambda_1(\mathcal{M}) \geq \ldots \geq \lambda_d(\mathcal{M})$; }
  $n(\mathcal{M}_1) = n(\mathcal{M}_2) = k$;
  $\overline{Fs(\mathcal{M}_1)} = \overline{Fs(\mathcal{M})}/n(\mathcal{M} + \overline{e_1}(\mathcal{M}) \cdot \sqrt{12 \cdot \lambda_1}/4$;
  $\overline{Fs(\mathcal{M}_2)} = \overline{Fs(\mathcal{M})}/n(\mathcal{M}) - \overline{e_1}(\mathcal{M}) \cdot \sqrt{12 \cdot \lambda_1}/4$;
  Construct $\Delta(\mathcal{M}_1)$ and $\Delta(\mathcal{M}_2)$ by dividing diagonal entry $\lambda_1$ of $\Delta(\mathcal{M})$ by 4;
  $P(\mathcal{M}_1) = P(\mathcal{M}_2) = P(\mathcal{M})$;
  $C(\mathcal{M}_1) = C(\mathcal{M}_2) = P(\mathcal{M}_1) \cdot \Delta(\mathcal{M}_1) \cdot P(\mathcal{M}_1)^T$;
  for each pair of attributes $i, j$ do
  begin
    $Sc_{ij}(\mathcal{M}_1) = k \cdot C_{ij}(\mathcal{M}_1) + Fs_i(\mathcal{M}_1) \cdot Fs_j(\mathcal{M}_1)/k$;
    $Sc_{ij}(\mathcal{M}_2) = k \cdot C_{ij}(\mathcal{M}_2) + Fs_i(\mathcal{M}_2) \cdot Fs_j(\mathcal{M}_2)/k$;
  end;
end
```

Fig. 3. Splitting Group Statistics (Algorithm)

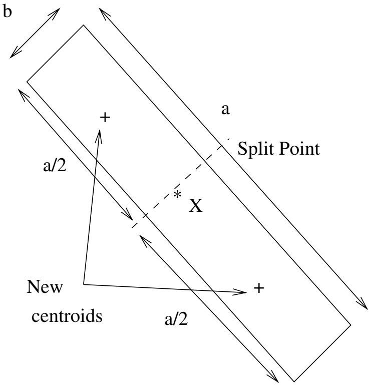


Fig. 4. Splitting Group Statistics (Illustration)

points. Rather, we impose the requirement that each group should maintain between k and $2 \cdot k$ data points.

As each new point in the data is received, it is added to the nearest group, as determined by the distance to each group centroid. As soon as the number of data points in the group equals $2 \cdot k$, the corresponding group needs to be split into two groups of k points each. We note that with each group, we only maintain the group statistics as opposed to the actual group itself. Therefore, the splitting process needs to generate two new sets of *group statistics* as opposed to the data points. Let us assume that the original set of group statistics to be split is given by $\mathcal{M}$, and the two new sets of group statistics to be generated are given by $\mathcal{M}_1$ and $\mathcal{M}_2$. The overall process of group updating is illustrated by the algorithm *DynamicGroupMaintenance* in Figure 2. As in the previous case, it is assumed that we start off with a static database $\mathcal{D}$. In addition, we have a constant stream $\mathcal{S}$ of data which consists of new data points arriving in the database. Whenever a new data point $\overline{X}$ is received, it is added to the group $\mathcal{M}$, whose centroid is closest to $\overline{X}$. As soon as the group size equals $2 \cdot k$, the corresponding group statistics needs to be split into two sets of group statistics. This is achieved by the procedure *SplitGroupStatistics* of Figure 3.

In order to split the group statistics, we make the same simplifying assumptions about (locally) uniform and independent distributions along the eigenvectors for each group. We also assume that the split is performed along the most elongated axis direction in each case. Since the eigenvalues correspond to variances along individual eigenvectors, the eigenvector corresponding to the largest eigenvalue is a candidate for a split. An example of this case is illustrated in Figure 4. The logic of choosing the most elongated direction for a split is to reduce the variance of each individual group as much as possible. This ensures that each group continues to correspond to a small data locality. This is useful in order to minimize the effects of the approximation assumptions of uniformity within a given data locality. We assume that the corresponding eigenvector is denoted by $\overline{e_1}$ and its eigenvalue 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[3] by $a = \sqrt{12 \cdot \lambda_1}$.

The number of records in each newly formed group is equal to k since 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 $\mathcal{M}_2$. This is done by first deriving the centroid and zero (second-order) correlation directions for each group. The values of $Fs_i(\mathcal{G})$ and $Sc_{ij}(\mathcal{G})$ about each group can also be directly derived from these quantities. We will proceed to describe this derivation process in more detail.

Let us assume that the centroid of the unsplit group $\mathcal{M}$ is denoted by $\overline{Y(\mathcal{M})}$. This centroid can be computed from the first order values $\overline{Fs(\mathcal{M})}$ using the following relationship:

$$\overline{Y(\mathcal{M})} = (Fs_1(\mathcal{M}), \ldots Fs_d(\mathcal{M}))/n(\mathcal{G}) \tag{2}$$

As evident from Figure 4, 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. Therefore, 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. It now remains to compute the second order statistical values. This is slightly more tricky.

[3] 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 $\sqrt{a/12}$.

Once the co-variance matrix for each of the split groups has been computed, the second-order aggregate statistics can be derived by the use of the covariance values in conjunction with the centroids that have already been computed. Let us assume that the ijth entry of the co-variance matrix for the group $\mathcal{M}_1$ is given by $C_{ij}(\mathcal{M}_1)$. Then, from Observation 2, it is clear that the second order statistics of $\mathcal{M}_1$ may be determined as follows:

$$Sc_{ij}(\mathcal{M}_1) = k \cdot C_{ij}(\mathcal{M}_1) + Fs_i(\mathcal{M}_1) \cdot Fs_j(\mathcal{M}_1)/k \tag{3}$$

Since the first-order values have already been computed, the right hand side can be substituted, once the co-variance matrix has been determined. 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:

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

The eigenvalue corresponding to $\overline{e_1}(\mathcal{M})$ is equal to $\lambda_1/4$ because the splitting process along $\overline{e_1}$ reduces the corresponding variance by a factor of 4. All other eigenvectors 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:

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

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

$$C(\mathcal{M}_1) = C(\mathcal{M}_2) = P(\mathcal{M}_1) \cdot \Delta(\mathcal{M}_1) \cdot P(\mathcal{M}_1)^T \tag{4}$$

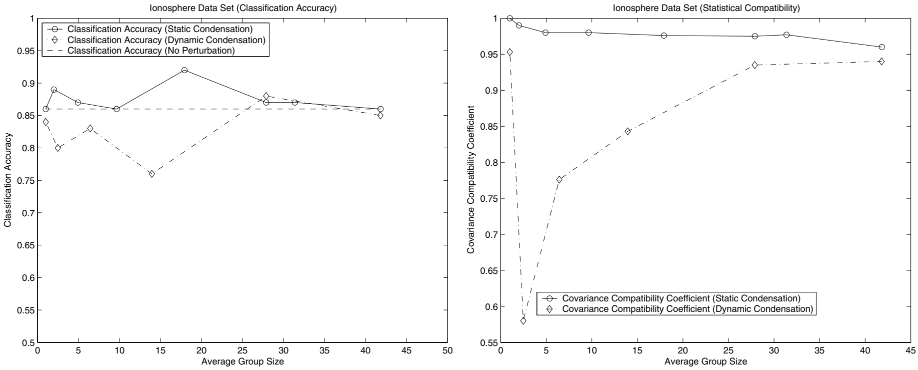
Once the co-variance matrices have been determined, the second order aggregate information about the data is determined using Equation 3. We note that even though the covariance matrices of $\mathcal{M}_1$ and $\mathcal{M}_2$ are identical, the values of $Sc_{ij}(\mathcal{M}_1)$ and $Sc_{ij}(\mathcal{M}_2)$ will be different because of the different first order aggregates substituted in Equation 3. The overall process for splitting the group statistics is illustrated in Figure 3.

3.1 Application of Data Mining Algorithms to Condensed Data Groups

Once the condensed data groups have been generated, data mining algorithms can be applied to the anonymized data which is generated from these groups. After generation of the anonymized data, any *known* data mining algorithm can be directly applied to this new data set. Therefore, specialized data mining algorithms do not need to be developed for the condensation based approach. As an example, we applied the technique to the classification problem. We used a simple nearest neighbor classifier in order to illustrate the effectiveness of the technique. We also note that a nearest neighbor classifier cannot be effectively modified to work with the perturbation-based approach of [1]. This is because the method in [1] reconstructs aggregate distributions of each dimension independently. On the other hand, the modifications required for the case of the condensation approach were relatively straightforward. In this case, separate sets of data were generated from each of the different classes. The separate sets of data for each class were used in conjunction with a nearest neighbor classification procedure. The class label of the closest record from the set of perturbed records is used for the classification process.

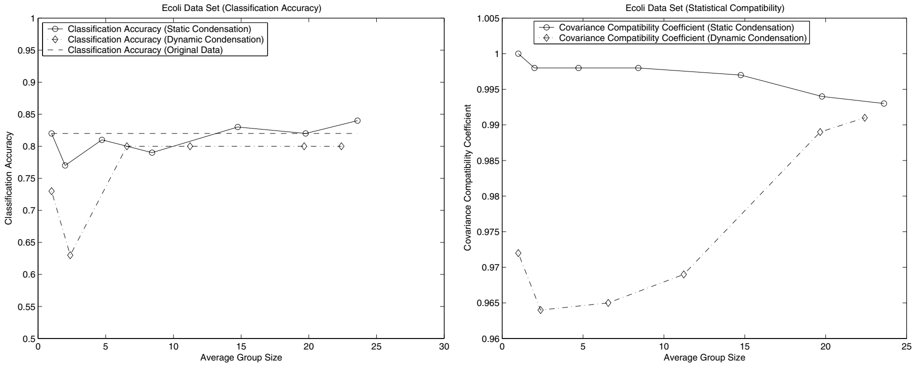
4 Empirical Results

Since the aim of the privacy preserving data mining process was to create a new perturbed data set with similar data characteristics, it is useful to compare the statistical characteristics of the newly created data with the original data set. Since the proposed technique is designed to preserve the covariance structure of the data, it would be interesting to test how the covariance structure of the newly created data set matched with the original. If the newly created data set has very similar data characteristics to the original data set, then the condensed



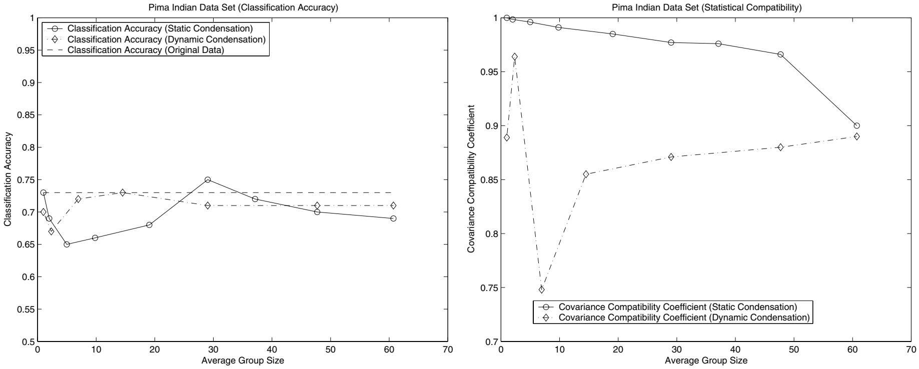
(a) (b)

Fig. 5. (a) Classifier Accuracy and (b) Covariance Compatibility (Ionosphere)



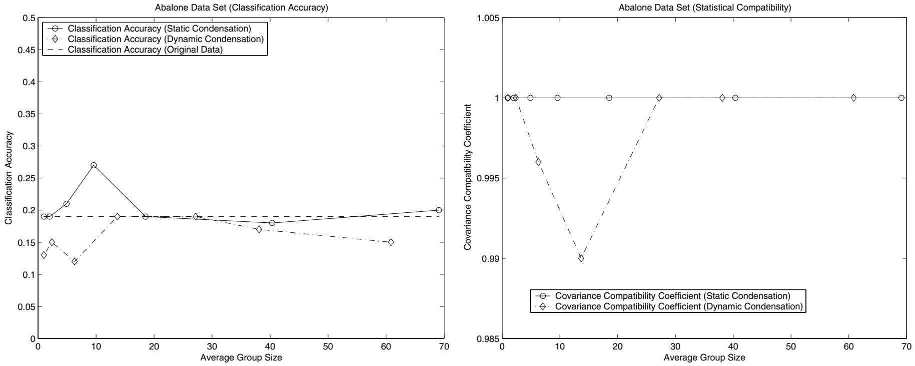
(a) (b)

Fig. 6. (a) Classifier Accuracy and (b) Covariance Compatibility (Ecoli)



(a) (b)

Fig. 7. (a) Classifier Accuracy and (b) Covariance Compatibility (Pima Indian)



(a) (b)

Fig. 8. (a) Classifier Accuracy and (b) Covariance Compatibility (Abalone) data set is a good substitute for privacy preserving 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}. In order to perform this comparison, we computed the statistical coefficient of correlation between the pairwise 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 correlations between the entries, the value of μ is -1.

We tested the data generated from the privacy preserving condensation approach on the classification problem. Specifically, we tested the accuracy of a simple k-nearest neighbor classifier with the use of different levels of privacy. The level of privacy is controlled by varying the sizes of the groups used for the condensation process. The results show that the technique is able to achieve high levels of privacy without noticeably compromising classification accuracy. In fact, in many cases, the classification accuracy improves because of the noise reduction effects of the condensation process. These noise reduction effects result from the use of the aggregate statistics of a small local cluster of points in order to create the anonymized data. The aggregate statistics of each cluster of points often mask the effects of a particular anomaly[4] in it. This results in a more robust classification model. We note that the effect of anomalies in the data are also observed for a number of other data mining problems such as clustering [10]. While this paper studies classification as one example, it would be interesting to study other data mining problems as well.

A number of real data sets from the UCI machine learning repository[5] were used for the testing. The specific data sets used were the Ionosphere, Ecoli, Pima Indian, and the Abalone Data Sets. Except for the Abalone data set, each of these data sets correspond to a classification problem. In the abalone data set, the aim of the problem is to predict the age of abalone, which is a regression modeling problem. 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.

The results on classification accuracy for the Ionosphere, Ecoli, Pima Indian, and Abalone data sets are illustrated in Figures 5(a), 6(a), 7(a) and 8(a) respectively. In each of the charts, the average group size of the condensation groups is indicated 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. Three sets of results have been illustrated on each graph:

- The accuracy of the nearest neighbor classifier when static condensation was used. In this case, the static version of the algorithm was used in which the entire data set was used for condensation.
- The accuracy of the nearest neighbor classifier when dynamic condensation was used. In this case, the data points were added incrementally to the condensed groups.

[4] We note that a k-nearest neighbor model is often more robust than a 1-nearest neighbor model for the same reason.

[5] *http : //www.ics.uci.edu/˜mlearn*

- We note that when the group size was chosen to be one for the case of static condensation, the result was the same as that of using the classifier on the original data. Therefore, a horizontal line (parallel to the X-axis) is drawn in the graph which shows the baseline accuracy of using the original classifier. This horizontal line intersects the static condensation plot for a groups size of 1.

An interesting point to note is that when dynamic condensation is used, the result of using a group size of 1 does not correspond to the original data. This is because of the approximation assumptions implicit in splitting algorithm of the dynamic condensation process. Specifically, the splitting procedure assumed a uniform distribution of the data within a given condensed group of data points. Such an approximation tends to lose its accuracy for very small group sizes. However, it should also be remembered that the use of small group sizes is not very useful anyway from the point of view of privacy preservation. Therefore, the behavior of the dynamic condensation technique for very small group sizes is not necessarily an impediment to the effective use of the algorithm.

One of the interesting conclusions from the results of Figures 5(a), 6(a), 7(a) and 8(a) is that the static condensation technique often provided *better* accuracy than the accuracy of a classifier on the original data set. The effects were particularly pronounced in the case of the ionosphere data set. As evident from Figure 5(a), the accuracy of the classifier on the statically condensed data was higher than the baseline nearest neighbor accuracy for almost all group sizes. The reason for this was that the process of condensation affected the data in two potentially contradicting ways. One effect was to add noise to the data because of the random generation of new data points with similar statistical characteristics. This resulted in a reduction of the classification accuracy. On the other hand, the condensation process itself removed many of the anomalies from the data. This had the opposite effect of improving the classification accuracy. In many cases, this trade-off worked in favor of improving the classification accuracy as opposed to worsening it.

The use of dynamic classification also demonstrated some interesting results. While the absolute classification accuracy was not quite as high with the use of dynamic condensation, the overall accuracy continued to be almost comparable to that of the original data for modestly sized groups. The comparative behavior of the static and dynamic condensation methods is because of the additional assumptions used in the splitting process of the latter. We note that the splitting process uses a uniformly distributed assumption of the data distribution within a particular locality (group). While this is a reasonable assumption for reasonably large group sizes within even larger data sets, the assumption does not work quite as effectively when either of the following is true:

- When the group size is too small, then the splitting process does not estimate the statistical parameters of the two split groups quite as robustly.
- When the group size is too large (or a significant fraction of the overall data size), then a set of points can no longer be said to represent a locality of the data. Therefore, the use of the uniformly distributed assumption for splitting

and regeneration of the data points within a group is not as robust in this case.

These results are reflected in the behavior of the classifier on the dynamically condensed data. In many of the data sets, the classification accuracy was sensitive to the size of the group. While the classification accuracy reduced upto the use of a group size of 10, it gradually improved with increasing groups size. In most cases, the classification accuracy of the dynamic condensation process was comparable to that on the original data. In some cases such as the Pima Indian data set, the accuracy of the dynamic condensation method was even higher than that of the original data set. Furthermore, the accuracy of the classifier on the static and dynamically condensed data was somewhat similar for modest group sizes between 25 to 50. One interesting result which we noticed was for the case of the Pima Indian data set. In this case, the classifier worked more effectively with the dynamic condensation technique as compared to that of static condensation. The reason for this was that the data set seemed to contain a number of classification anomalies which were removed by the splitting process in the dynamic condensation method. Thus, in this particular case, the splitting process seemed to improve the overall classification accuracy. While it is clear that the effects of the condensation process on classification tends to be data specific, it is important to note that the accuracy of the condensed data is quite comparable to that of the original classifier.

We also compared the covariance characteristics of the data sets. The results are illustrated in Figures 5(b), 6(b), 7(b) and 8(b) respectively. It is clear that in each data set, the value of the statistical correlation μ was almost 1 for each and every data set for the static condensation method. In most cases, the value of μ was larger than 0.98 over all ranges of groups sizes and data sets. While the value of the statistical correlation reduced slightly with increasing group size, its relatively high value indicated that the covariance matrices of the original and perturbed data were virtually identical. This is a very encouraging result since it indicates that the approach is able to preserve the inter-attribute correlations in the data effectively. The results for the dynamic condensation method were also quite impressive, though not as accurate as the static condensation method. In this case, the value of μ continued to be very high (> 0.95) for two of the data sets. For the other two data sets, the value of μ reduced to the range of 0.65 to 0.75 for very small group sizes. As the average group sizes increased to about 20, this value increased to a value larger than 0.95. We note that in order for the indistinguishability level to be sufficiently effective, the group sizes also needed to be of sizes at least 15 or 20. This means that the accuracy of the classification process is not compromised in the range of group sizes which are most useful from the point of view of condensation. The behavior of the correlation statistic for dynamic condensation of small group sizes is because of the splitting process. It is a considerable approximation to split a small discrete number of discrete points using a uniform distribution assumption. As the group sizes increase, the value of μ increases because of the robustness of using a larger number of points in each group. However, increasing group sizes beyond a certain limit has the opposite effect of reducing μ (slightly). This effect is visible in both the static and dynamic condensation methods. The second effect is because of the greater levels of approximation inherent in using a uniform distribution assumption over a larger *spatial locality*. We note that when the overall data set size is large, it is more effectively possible to simultaneously achieve the seemingly contradictory goals of using the robustness of larger group sizes as well as the effectiveness of using a small locality of the data. This is because a modest group size of 30 truly represents a small data locality in a large data set of 10000 points, whereas this cannot be achieved in a data set containing only 100 points. We note that many of the data sets tested in this paper contained less than 1000 data points. These constitute difficult cases for our approach. Yet, the condensation approach continued to perform effectively both for small data sets such as the Ionosphere data set, and for larger data sets such as the Pima Indian data set. In addition, the condensed data often provided more accurate results than the original data because of removal of anomalies from the data.

5 Conclusions and Summary

In this paper, we presented a new way for privacy preserving data mining of data sets. Since the method re-generates multi-dimensional data records, existing data mining algorithms do not need to be modified to be used with the condensation technique. This is a clear advantage over techniques such as the perturbation method discussed in [1] in which a new data mining algorithm needs to be developed for each problem. Unlike other methods which perturb each dimension separately, this technique is designed to preserve the inter-attribute correlations of the data. As substantiated by the empirical tests, the condensation technique is able to preserve the inter-attribute correlations of the data quite effectively. At the same time, we illustrated the effectiveness of the system on the classification problem. In many cases, the condensed data provided a higher classification accuracy than the original data because of the removal of anomalies from the database.

References

1. Agrawal R., Srikant R.: Privacy Preserving Data Mining. Proceedings of the ACM SIGMOD Conference, (2000).
2. Agrawal D. Aggarwal C. C.: On the Design and Quantification of Privacy Preserving Data Mining Algorithms. ACM PODS Conference, (2002).
3. Benassi P. Truste: An online privacy seal program. Communications of the ACM, 42(2), (1999) 56–59.
4. Clifton C., Marks D.: Security and Privacy Implications of Data Mining. ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discove ry, (1996) 15–19.
5. Clifton C., Kantarcioglu M., Vaidya J.: Defining Privacy for Data Mining. National Science Foundation Workshop on Next Generation Data Mining, (2002) 126–133.

6. Vaidya J., Clifton C.: Privacy Preserving Association Rule Mining in Vertically Partitioned Data. ACM KDD Conference, (2002).
7. Cover T., Thomas J.: Elements of Information Theory, John Wiley & Sons, Inc., New York, (1991).
8. Estivill-Castro V., Brankovic L.: Data Swapping: Balancing privacy against precision in mining for logic rules. Lecture Notes in Computer Science Vol. 1676, Springer Verlag (1999) 389–398.
9. Evfimievski A., Srikant R., Agrawal R., Gehrke J.: Privacy Preserving Mining Of Association Rules. ACM KDD Conference, (2002).
10. Hinneburg D. A., Keim D. A.: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. ACM KDD Conference, (1998).
11. Iyengar V. S.: Transforming Data To Satisfy Privacy Constraints. ACM KDD Conference, (2002).
12. Liew C. K., Choi U. J., Liew C. J.: A data distortion by probability distribution. ACM TODS Journal, 10(3) (1985) 395-411.
13. Lau T., Etzioni O., Weld D. S.: Privacy Interfaces for Information Management. Communications of the ACM, 42(10) (1999), 89–94.
14. Murthy S.: Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. Data Mining and Knowledge Discovery, Vol. 2, (1998), 345–389.
15. Moore Jr. R. A.: Controlled Data-Swapping Techniques for Masking Public Use Microdata Sets. Statistical Research Division Report Series, RR 96-04, US Bureau of the Census, Washington D. C., (1996).
16. Rizvi S., Haritsa J.: Maintaining Data Privacy in Association Rule Mining. VLDB Conference, (2002.)
17. Silverman B. W.: *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, (1986).
18. Samarati P., Sweeney L.: 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).