Reconstruction of Data Distribution from Perturbed Dataset

Temo Bardzimashvili
Department of Mathematics
Michigan State University

East Lansing, MI 48824
Phone: (517) 353-0844
bardzima@msu.edu

Pang-Ning Tan
Department of Computer Science and Engineering
Michigan State University
3115 Engineering Building
East Lansing, MI 48824-1226
Phone: (517) 353-3148
ptan@msu.edu

ABSTRACT

In this project, we are considering the information privacy which now-a-days has become one of the most important issues. We touch upon several techniques of masking the data, namely random distortion, including the uniform and Gaussian noise, applied to the data in order to protect it. Then, after using a certain data recovering techniques we look for the distribution of data obtained. Our task is to determine whether the distributions of the original and recovered data are close enough to each other despite the nature of the noise applied. We are considering an ensemble clustering method to reconstruct the initial data distribution. As the tool for the algorithm implementations we chose the "language of choice in industrial world" – MATLAB.

Keywords

Privacy-preserving data mining, data perturbation, ensemble method, clustering, fuzzy clustering, voting.

1. INTRODUCTION

The dramatic growth of the Internet during the past decade has resulted in the tremendous amount of information. In order to get some idea about the volume of the information available today we mention that databases of two of the largest web resources – National Climatic Data Center and NASA – contain about 600 terabytes of data, which is only about 8% of so-called "deep" web [1]. But along with the availability and the amount of data, the privacy issue has also experienced a big resonance. Despite whether the private data is being retrieved for malicious (i.e. obtaining information about credit card number or bank information) or for official (i.e. information on online activity of individuals gathered by federal government) reasons, people are concerned about keeping the private information undisclosed. Different poll among web users reveal that about 85% of people give their preference to a privacy policy [2, 3].

One of the examples for the data privacy used in real life is the insurance companies. They do not give access to the original data, the private information of their customers. But instead they can provide some sort of statistics of the data changed in some certain way, without providing the original information of individual customers. But even such "vague" data can be used to identify trends and patterns

Basically, there are two approaches of data concealment. The first approach is data randomization (perturbation). Usually it conceals the real data by modifying it randomly, superimposing a random noise on it [3-5]. The second approach uses the cryptography techniques to encode the initial information.

There exist a lot of cases when we need to obtain the information on the initial data. For instance, companies, selling their product in online stores, might be interested in finding out the range of customer age/salary their product should target to. Since this information is not available in its initial state (since customers do not want their personal information to be available for public), a company needs to deal with the perturbed/encrypted data. The main goal of this article is to evaluate the initial distribution of the data using a so called ensemble clustering method [7], and then to compare its efficiency to other methods of data reconstruction.

In this paper we consider the first approach – the data randomization [3]. If we have the initial data set of N independent variables $X=\{x_1, x_2, \ldots, x_N\}$. In order to perturb the data we consider N independent random values $Y=\{y_1, y_2, \ldots, y_N\}$ and the perturbed data set will be given as $X'=X+Y$. In this case it is impossible to reconstruct initial values exactly but it is possible to recover the initial data distribution with some certain precision. There also is some loss of information during the previous distribution reconstruction process. However, the reconstruction algorithms offered in different papers (including this one) are able to recover the original data pattern. Which algorithm one should use, is a matter of a precision and an efficiency of the method.

In Section 2 we state the problem and provide the overview for the related works in this field. Section 3 offers the description of the techniques, such as algorithms and routines used in the paper. Section 4 offers the results obtained. Section 5 contains discussion, future work and summary for the results. In Section 6 the link to all routines is provided. Section 7 lists down the references used during this project.

2. PROBLEM STATEMENT

The basic problem considered in this paper can be abstracted as the following: we have the set of randomly distracted data set. Our task is to obtain the original data distribution based on the present distorted data. Again, as it was mentioned, we reconstruct only distribution, not the actual values of individual records of the dataset.

Before announcing the method to be used in this paper, let us define the concept of clustering, since it is "a mile-stone" of the background theory implemented in algorithms described later.

We consider a set of data points each having a set of attributes. The main goal of clustering is to divide data into groups called clusters, such that data points in one cluster would be more similar to one another and respectively, data points in separate clusters would be less similar to one another. The similarity can be measured based on Euclidean Distance (in case attributes are continuous).

As a method of the reconstruction of initial data distribution we use so-called voting technique. In fact, voting is an ensemble clustering method that combines several clustering results to a final partition [7]. Generally speaking, during partition clustering we usually form a single partition of some number of clusters based on some clustering standard.

We implement the voting algorithm in MATLAB. After obtaining results our goal is to the compare effectiveness of the methods suggested. We also try the algorithm for the different types of perturbations such as product and exponential, as well as for various kinds if distributions (normal, uniform). For instance, if X is the initial dataset matrix and Y is matrix consisting if random noise, then in case of product perturbation the perturbed dataset $X' = XY$.

2.1 Related Work

Privacy-preserving is one of the mostly considerable topics in data mining. Respectively, there exist a lot of references and literature on this extensive subject.

Although there exist different categories for the privacy-preserving data mining algorithms (such as ones based on a so called distributed framework and data-swapping approaches [5]), our prime interest is still the random perturbation of data. In [4] such approach is considered: additional random noise modulates the data, such that the individual data values are distorted preserving the original distribution properties if considering the dataset as a whole. After applying random noise, the perturbed data is used to extract the patterns and models. The randomized value distortion technique for learning decision trees [4] and association rule learning [8] are examples of this approach.

There are many different algorithms dealing with the randomly perturbed data sets. One of the mostly used algorithms is so called an expectation-maximization (EM) algorithm considered in [6]. It is also remarked in [6] that the method, based on the Bayesian approach, suggested in [4] "does not take into account the distribution of the original data (which could be used to guess the data value to a higher level of accuracy)". Compared to the method used in [4], EM Algorithm provides more robust evaluation of initial distribution and less information loss even in case of large number of data points. Another method for the privacy-preserving data mining considered in [8] is the association rule analysis.

In this paper we propose new method for the obtaining the original data distribution – the Ensemble Method for Clustering. This method is considered and discussed in [7]. The next section describes the Ensemble Method and its core – the Voting Algorithm in more details.

The main contribution of this article is to develop robust and efficient method for the data distribution reconstruction.

3. METHODOLOGY

As already mentioned, the core technique described in this paper used for the reconstruction is so called the Ensemble Clustering Method. It is used to optimally combine given different clusterings, that is it finds the one clustering that in best way represents the given set of clusterings obtained from different clustering methods.

The basis of the Ensemble Clustering Method is the Voting Algorithm [7,9,10]. The algorithm itself is based on the following idea: let us have a set of m clusterings $S^{(m)}$. Our goal is to obtain one clustering P which will represent whole set $S^{(m)}$ optimally. The algorithm for this is shown on Figure 1.

1. $P_1 := S^{(1)}$
2. for i=2 to m

   For all permutations $\Pi(U^{(i)})$ of columns of $U^{(i)}$ find

   $$\max_{\Pi^i_{max}} trace(P_{i-1}^T \prod(U^{(i)}))$$

   Permutation Π^i_{max} for which the trace of product matrix $P^T\Pi(U^{(i)})$ will be maximum.

   Find P for *i-th* step using recursive formula:

   $$P_i = \frac{i-1}{i} P_{i-1} + \frac{1}{i} \prod^i_{\max}$$

3. P_m is the optimal clustering.

Figure 1: the Voting Algorithm

Generally $U^{(i)}$ is the fuzzy clustering matrix where columns are clusters and rows are the data points. Each matrix element determines the weight of data point belonging to the certain cluster. That is for each row, the sum of all elements will be equal one (except for cases when point does not belong to any cluster – so called noise. In this case all elements in correspondent row will be zeros).

Notice that in step 2 of our algorithm we consider $k!$ permutations of k columns of clustering matrix $U^{(i)}$. For the number of clusters greater than 8-9, our algorithm will become computationally expensive. For such cases there are some other techniques not considered in this paper [7].

4. IMPLEMENTATION AND EXPERIMENT

For our experiment we used MATLAB as the environment for the algorithm implementations.

4.1 Experimental Setup

For the experiment we considered a set of 1000 two dimensional points partitioned into groups. We considered five group partitioning (Figure 2).

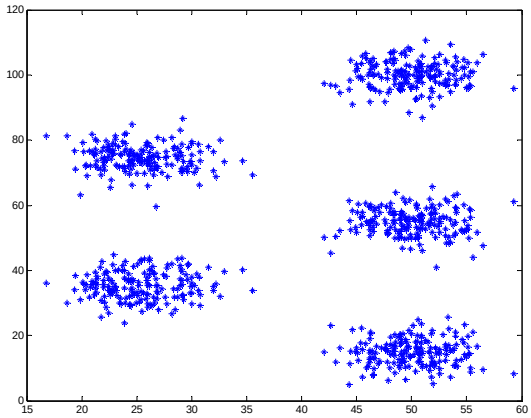

Figure 2: Two-dimensional point distribution.

After that we added Gaussian noise; that is, for each data point (or more exactly for each coordinate of a point) we generated a Gaussian random number with certain deviation and added it to it (Figure 3).

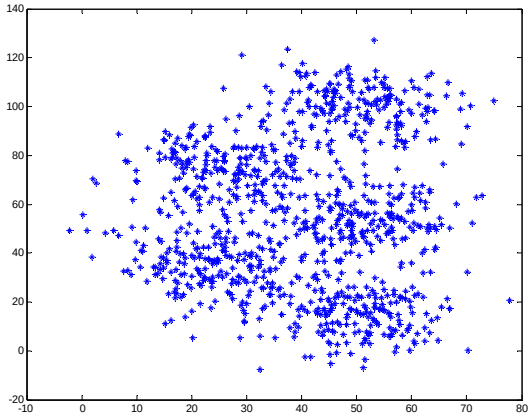

Figure 3: Perturbed data set

For the given dataset we considered two clustering techniques: k-means method and DBSCAN algorithm. For the first method we used MATLAB built-in function kmeans, as for the second one, DBSCAN, we implemented it based on the algorithm described in [11].

Given the perturbed dataset, we applied one of two algorithms mentioned above (for each of them we selected the parameters for which the methods were issuing the best results). As the measure of the quality we considered the correlation between the initial distribution and the one obtained due to the clustering. Namely, we were calculating the correlation between incidence matrices (initial and clustered ones).

Then we considered the Ensemble Method. To obtain the set of clustering to be used in the voting algorithm, we ran k-means and DBSCAN clustering algorithms, 20 times each, with varying parameters for each run.

Parameters for the methods were chosen in the following way: 1) we ran k-means for four consequent numbers of centroids, all around some K number, which was chosen as the one for which the k-means method was issuing clustering with largest correlation to the original data distribution. That is, first 4 runs were performed for K-2 centroids, the next 4 runs for K-1 centroids and so on; 2) since there is no certain technique of finding optimal parameters for DBSCAN algorithm available yet (the one recommended in [7] does not work well for very dense dataset or for the large number of points), we had to choose them manually. However some techniques proposed by mere logics could work in some cases (Table 1).

Table 1. Technique of choosing *EPS* and *MinPts* parameters for DBSCAN algorithm.

1. Find area A enclosing all points in dataset.
2. $EPS \approx \alpha\sqrt{A}$, where α is (roughly) the ratio of the average and maximum densities. For our case $a \approx 0.05$.
3. $MinPts \approx \dfrac{2\pi \cdot EPS^2 \cdot N}{A}$, here N is the total number of data points.

After running these methods we have the set of 40 clustering, which we were using as the input for the Voting Algorithm.

Another challenging issue in our experiment was the varying number of clusters in each clustering produced by methods, while the Voting Algorithm requires equal number of clusters in each clustering. To overcome this problem we were taking the maximal number K_{max} of clusters among all clusterings as the universal one. Then we extended the number of clusters in the clustering to the given number K_{max}.

Since P optimal clustering is a fuzzy one (that is the belonging of point to a cluster is weighted), we choose the cluster where the point has maximal weight. After finding optimal clustering P for the given set, we calculated the incidence matrix and found the correlation between it and the original incidence matrix.

4.2 Experimental Results

We described the experiment we performed in the previous section. As we mentioned the first step after blurring data is applying clustering method to it and the comparison of the obtained and original data distributions. The sample results of clusterings using k-means and DBSCAN methods are shown on Figure 4 and Figure 5 respectively. Here colors represent different clusters.

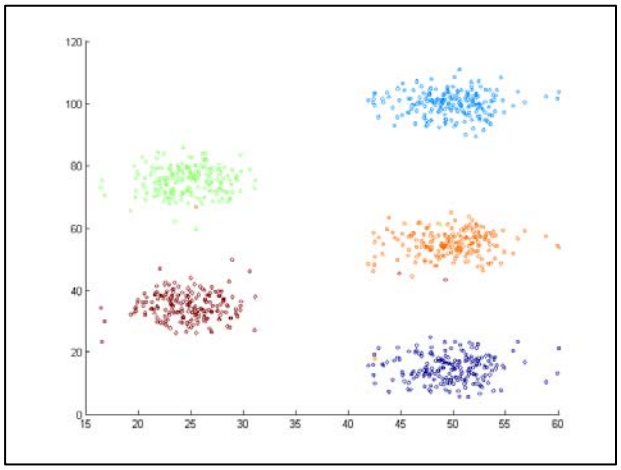

Figure 4. k-means algorithm result. k = 5.

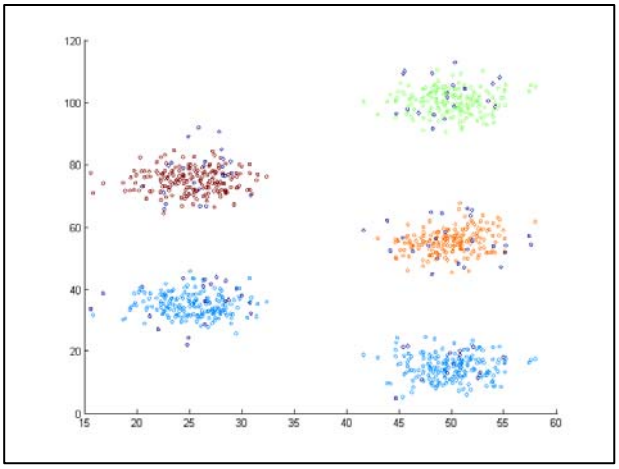

Figure 5. DBSCAN algorithm result.

For k-means the correlation was found to be ~0.9875 and for DBSCAN correlation was ~0.6780. The data was blurred by Gaussian noise with standard deviation of 50% of cluster size.

We tested the Ensemble Method for different strengths of noise. The graph on Figure 6 shows the correlation coefficients obtained from three clustering methods for Gaussian noise with four different standard deviations: 50%, 60%, 70%, 80% (percentage of physical size of cluster).

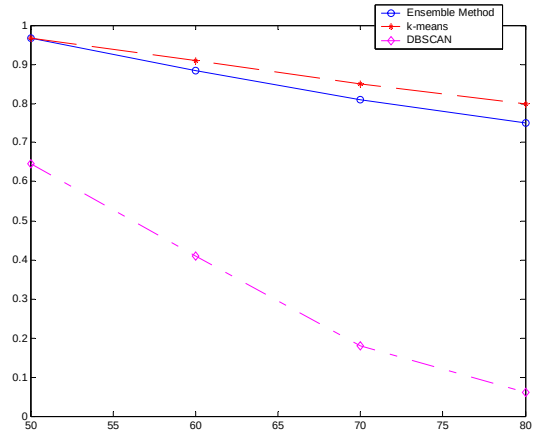

Figure 6. Dependence of correlation on noise strength.

For the noise with 50% standard deviation the clustering obtained using Ensemble Clustering Method looked like one on Figure 7.

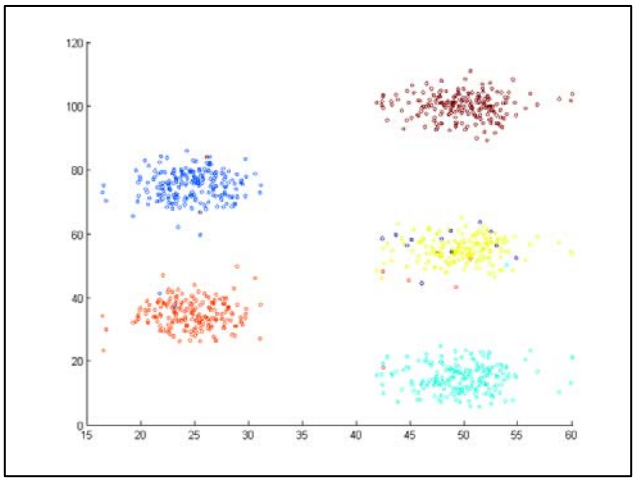

Figure 7. Ensemble Method Clustering.

The correlation in this case was 0.9676.

4.3 Discussion of Results

From results described above we see that for the particular data distribution we considered both k-means (with k=5) and Ensemble Method perform well. The explanation why the k-means perform so well, despite it is sensitive to noise is the following: the data we considered is divided into groups having elliptical shapes, and this kind of clustering is easily dealt by k-means method. Taking the proper number of centroids, k-means method finds elliptical disjoint clusters very efficiently. At the same time, despite that DBSCAN is robust to noise, in such uniform point distribution that we have after applying a strong noise this algorithm becomes inefficient. And since the Voting Algorithm used in Ensemble Clustering Method chooses the optimal clustering way among those provided, it performes almost

as good as the best clustering method (k=5 k-means in our case). One can see that the measure of efficiency, correlation is very close to its maximum. As the noise increases, the correlation between restored data distribution decreases slowly remaining high (about 0.85) for a rather big range of noise.

5. CONCLUSIONS AND FUTURE WORK

In this paper we cover the Privacy-Preserving Data Mining, one of the most interesting and popular topics of researches today. As the important issue in this area, we consider the possibility of original data distribution restoration from the available perturbed dataset. In addition to the several other techniques available (such as Bayes Rule and Expectation Maximization based techniques) we propose the brand new one, which is based on the recently invented approach concerning the merging several different clusterings into optimal one – the Ensemble Method. To examine our proposition, we consider the two-dimensional dataset, where the data points are grouped into five elliptic-shaped partitions. To perturb data, we apply Gaussian noise, therefore masking real values of data points. Now, given the perturbed dataset we use two clustering algorithms, k-means and DBSCAN, to cluster the perturbed dataset, that is to find the original partitions. After this we provide the Voting Algorithm used by Ensemble Method with the set of forty clusterings obtained from running k-means and DBSCAN with varying parameters and obtain one optimal clustering. As the measure of the efficiency of the original data distribution restoration we consider the correlation between the original and restored incidence matrices. We calculate the correlation coefficients for all clustering methods.

As we see from the description of the experiment results, the Ensemble Method provides very efficient way of data distribution restoration. The correlation between the original data distribution and the restored one is very close to one. At the same time the Ensemble Method performs almost as efficiently as the best clustering algorithm. In our example the k-means algorithm with 5 centroids performs the best, and the data quality obtained using the Ensemble Method is only slightly worth. Since the Ensemble Method only chooses the best clustering among provided ones, that means that it is rather independent on clustering methods inappropriate to the given task.

In this project we consider two-dimensional, partitioned data distributed normally into several elliptic partitions. As the future work we will consider other than Gaussian distribution of data. Also, different kinds of noise (uniformly distributed, exponential) will be considered. Besides two clustering methods considered in this project, we will add some other one, such as Nearest Neighbor, fuzzy k-means etc. We also will work on estimation of efficiency of the Ensemble Method compared to the other techniques of data distribution reconstruction [5].

6. WWW AVAILABILITY

All MATLAB routines created for this project are available at the following address:

http://www.msu.edu/~bardzima/files/votem.zip

7. REFERENCES

[1] University of California, Berkley (2000). How much information? http://www.sims.berkeley.edu/research/projects/how-much-info/internet.html

[2] Joseph Turow, 2003, Americans and Online Privacy, The System Is Broken. http://www.asc.upenn.edu/usr/jturow/internet-privacy-report/36-page-turow-version-9.pdf

[3] Alexandre Evfimievski. Randomization in Privacy Preserving Data Mining

[4] R. Agrawal and R. Srikant. Privacy-preserving data mining. 2000. ACM Press.

[5] H. Kargupta, S. Datta, Qi Wang and Krishnamoorthy Sivakumar. Random Data Perturbation Techniques and Privacy Preserving Data Mining.

[6] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms.

[7] A. Weingessel, E. Dimitriadou, K. Hornik. An Ensemble Method for Clustering. DSC 2003 Working Papers.

[8] S. Evfimievski. Randomization techniques for privacy preserving association rule mining.

[9] E. Dimitriadou, A. Weingessel, K. Hornik. Voting in Clustering and Finding the Number of Clusters.

[10] E. Dimitriadou, A. Weingessel, K. Hornik. A Voting-Merging Clustering Algorithm.

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.