ISSN No. 0976-5697

Volume 5, No. 6, July-August 2014

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Effectiveness of Multiplicative Data Perturbation for Privacy Preserving Data Mining

Bhupendra Kumar Pandya, Umesh Kumar Singh, Keerti Dixit, Kamal Bunkar
Institute of Computer Science
Vikram University, Ujjain, MP, India

Abstract: Privacy concerns over the ever-increasing gathering of personal information by various institutions led to the development of privacy preserving data. The approach protects the privacy of the data by perturbing the data through a method. The major challenge of data perturbation is to achieve the desired result between the level of data privacy and the level of data utility. Data privacy and data utility are commonly considered as a pair of conflicting requirements in privacy-preserving of data for applications and mining systems. Multiplicative perturbation algorithms aim at improving data privacy while maintaining the desired level of data utility by selectively preserving the mining task and model specific information during the data perturbation process. The multiplicative perturbation algorithm may find multiple data transformations that preserve the required data utility. Thus the next major challenge is to find a good transformation that provides a satisfactory level of privacy data. we are going to handle the problem of transforming a database to be shared into a new one that conceals private information while preserving the general patterns and trends from the original database. I am trying to get advantage of both the dimension like more protected data and relative fast.

Keywords: privacy preservation, multiplicative data perturbation

I. INTRODUCTION

A Statistical database (SDB) is a database system that allows its users to retrieve aggregate statistics (e.g., sample mean and variance) for a subset of the entities represented in the database and prevents the collection of information on specific individuals. In the statistics community, there has been extensive research on the problem of securing SDBs against disclosure of confidential information. This is generally referred to as statistical disclosure control. Statistical disclosure control approaches suggested in the literature are classified into four general groups: conceptual, query restriction, output perturbation and data perturbation [1]. Conceptual approach provides a framework for better understanding and investigating the security problem of statistical database at the conceptual data model level. It does not provide a specific implementation procedure. The Query Restriction approach offers protection by either restricting the size of query set or controlling the overlap among successive queries. The Output Perturbation approach perturbs the answer to user queries while leaving the data in the database unchanged. The Data Perturbation approach introduces noise into the database and transforms it into another version. This research paper primarily focuses on the data perturbation approaches.

Adding random noise to the private database is one common data perturbation approach. In this case, a random noise term is generated from a prescribed distribution, and the perturbed value takes the form: $y_{ij} = x_{ij} + r_{ij}$, where x_{ij} is the i^{th} attribute of the j^{th} private data record, and r_{ij} is the corresponding random noise. In the statistics community, this approach was primarily used to provide summary statistical information (e.g., sum, mean, variance, etc.) without disclosing individual's confidential data. In the privacy preserving data mining area, this approach was considered [2,3] in for building decision tree classifiers from private data. Recently, many researchers have pointed out that additive noise can be easily filtered out in many cases that may lead to compromising the privacy [4,5]. Given the large body of existing signal-processing literature on filtered random additive noise, the utility of random additive noise for privacy-preserving data mining is not quite clear.

The Possible drawback of additive noise makes one wonder about the possibility of using multiplicative noise (i.e., $y_{ij} = x_{ij} * r_{ij}$) for protecting the privacy of the data. Two basic forms of multiplicative noise have been well studied in the statistics community [6]. One multiplies each data element by a random number that has a truncated Gaussian distribution with mean one and small variance. The other takes a logarithmic transformation of the data first, adds multivariate Gaussian noise, then takes the exponential function exp (.) of the noise-added data. As noted in the former perturbation scheme was once used by the Energy Information Administration in the U.S. Department of Energy to mask the heating and cooling degree days, denoted by x_{ij}. A random noise r_{ij} is generated from a Gaussian distribution with mean 1 and variance 0.0225. The random noise is further truncated such that the resulting number r_{ij} satisfies $0.01 \leq |r_{ij}-1| \leq 0.6$. The perturbed data $x_{ij}r_{ij}$ were released.

This research paper gives a brief review and Analysis of perturbation scheme II.

A. Analysis of perturbation schemes with experimental result using matlab:

a. Data to be used:-

In this study we have Students result database of Vikram University, Ujjain. We have randomly selected 7 rows of the data with only 7 attributes(Marks of Foundation, Marks of Mathematics, Marks of Physics, Marks of Computer Science, Marks of Physics Practical, Marks of Computer Science Practical and Marks of Job Oriented Project).

b. Perturbation Scheme II:

Let xij be the value for the i-th attribute of the j-th record in the database as before i=1... n,j=1…m.

Let We generate the random noise following the multivariate Gaussian Distribution N (0, c ∑ u), where 0 < c < 1 and ∑ u is the covariance matrix of variables u1, u2… un. We denote the noise as eij. Let

zij = uij + eij,
yij = exp(zij)
=exp(In xij+eij)
=xijexp(eij)
= xij hij.

This perturbed data yij is released then. Note scheme assumes that all xij are positive.

c. *Statistical Properties of the Perturbed Data:*

It has been proved [6] that the mean, variance and covariance of the original data attributes can be estimated from the perturbed data.

d. *Mean of xi:*

Let σi2 = c Var(In xi). We have
E(xi) = E(yi) / exp(σi2/2)

e. *Variance of xi:*

Var(xi) = E(xi2) – (E(xi)) 2
=(Var(ui)/exp(2σi2)) – (E(xi) 2/ exp(σi2)) - (E(xi)) 2

f. *Covariance of x_i and x_j:*

$$Cov(x_i x_j) = \left\{ \frac{\sum_{k=1}^{m} y_{ik} y_{jk}}{\exp\left[\left(\sigma_i^2 + 2\rho\sigma_i\sigma_j + \sigma_j^2\right)/2\right]} - \frac{m \frac{\sum_{k=1}^{m} y_{ik}}{m} \frac{\sum_{k=1}^{m} y_{jk}}{m}}{\exp\left[\sigma_i^2 + \sigma_j^2\right]} \right\} \Big/ (m-1),$$

Where ρ is the correlation coefficient of xi and xj, and it can be obtained from the perturbed data. Because the noise was generated to maintain the same correlation structure, the correlation between the perturbed data will be on average the same as that between the original data in log-scale.

Table1: Original dataset before perturbation.

Foundation	Maths	Physics	Com. Sc.	Phy. Prac.	Com. Sc. Prac	Project
56	73	38	42	39	42	42
49	47	22	36	37	42	39
55	57	40	33	39	42	40
60	50	34	53	37	41	38
50	37	11	25	38	41	38
48	61	31	36	40	43	41
61	64	40	40	39	42	39

Table 2: Natural logarithm of original data

4.02	4.29	3.63	3.73	3.66	3.73	3.73
3.89	3.85	3.09	3.58	3.61	3.73	3.66
4.00	4.04	3.68	3.49	3.66	3.73	3.68
4.09	3.91	3.52	3.97	3.61	3.71	3.63
3.91	3.61	2.39	3.21	3.63	3.71	3.63
3.87	4.11	3.43	3.58	3.68	3.76	3.71
4.11	4.15	3.68	3.68	3.66	3.73	3.66

Table 3: Covariance of logarithmic data

0.009	0.008	0.027	0.014	-0.0002	-0.0005	-0.000
0.008	0.050	0.092	0.027	0.0041	0.0023	0.0068
0.027	0.092	0.220	0.077	0.0052	0.0034	0.0090
0.014	0.027	0.077	0.053	-0.0013	2.54	0.0009
-0.000	0.004	0.005	-0.001	0.0008	0.0003	0.0007
-0.000	0.002	0.003	2.54	0.0003	0.0002	0.0004
-0.000	0.006	0.009	0.000	0.0007	0.0004	0.0014

Table 4: Shows the random noise matrix after multivariate Gaussian distribution(c=.01)

1.055	1.062	1.089	1.045	1.053	1.053	1.054
1.202	1.294	1.344	1.243	1.205	1.203	1.213
0.796	0.844	0.863	0.830	0.808	0.808	0.811
1.090	1.060	1.089	1.102	1.065	1.07	1.065
1.032	1.099	1.182	1.083	1.020	1.017	1.024
0.876	0.890	0.930	9.01E-01	0.865	0.865	0.867
0.957	0.956	0.972	0.958	0.934	0.932	0.935

Table 5: Shows the perturbed data(c=.01)

59.108	77.596	41.383	43.929	41.078	44.262	44.279
58.913	60.840	29.568	44.783	44.606	50.545	47.345
43.832	48.110	34.552	27.395	31.542	33.950	32.467
65.429	53.020	37.045	58.457	39.415	4.38	40.488
51.627	40.676	13.003	27.091	38.795	41.718	38.927
42.090	54.302	28.841	3.24	34.604	37.214	35.568
58.399	61.193	38.891	38.320	36.436	39.168	36.494

Table 6: Shows the random noise matrix after multivariate Gaussian distribution(c=.10)

1.186	1.213	1.309	1.152	1.178	1.180	1.181
1.790	2.261	2.547	1.994	1.806	1.796	1.846
0.487	0.584	0.629	0.555	0.511	0.510	0.516
1.315	1.203	1.311	1.363	1.221	1.23	1.222
1.106	1.349	1.697	1.289	1.067	1.056	1.079
0.660	0.692	0.795	7.20	0.632	0.633	0.637
0.871	0.867	0.914	0.873	0.806	0.801	0.810

Table 7: Shows the perturbed data(c=.10)

66.43	88.55	49.76	48.40	45.95	49.58	49.64
87.74	106.30	56.03	71.80	66.82	75.43	72.00
26.83	33.34	25.17	18.31	19.93	21.43	20.67
78.90	60.19	44.59	72.25	45.19	5.06	46.43
55.32	49.92	18.67	32.23	40.57	43.31	41.01
31.68	42.22	24.67	2.59	25.29	27.22	26.15
53.14	55.53	36.59	34.92	31.45	33.68	31.61

Table: 8 Mean of original and perturbed data

original data	54.14	55.57	30.85	37.85	38.42	41.85	39.57
Perturbed data(c=.01)	53.20	55.53	30.89	37.91	37.06	40.52	38.36
Perturbed data(c=.10)	56.15	61.29	35.50	42.40	38.31	42.04	40.07

As seen in the table, the estimates of the means from the perturbed data are all close to those from original data.

Euclidean distance of original data

32.09	18.60	26.51	48.66	17.20	11.09	21.77
23.76	18.60	17.08	27.87	22.78	36.55	12.56
11.61	39.78	24.18	20.19	33.43	43.80	16.76

Euclidean distance of perturbed data(c=.01)

21.96	41.94	29.85	50.48	35.90	20.38	37.20
21.57	35.06	29.91	21.04	41.05	27.11	11.36
24.26	43.72	37.16	23.90	24.60	35.79	21.37

Euclidean distance of perturbed data(c=.10)

54.41	91.83	39.58	54.64	77.24	49.08	143.51
63.82	99.36	128.41	100.69	94.19	51.39	15.89
44.65	55.13	80.32	53.27	37.51	25.08	31.16

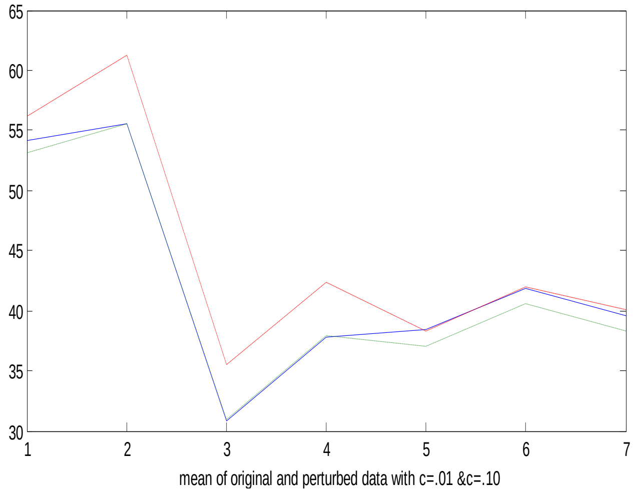

Figure 3.3

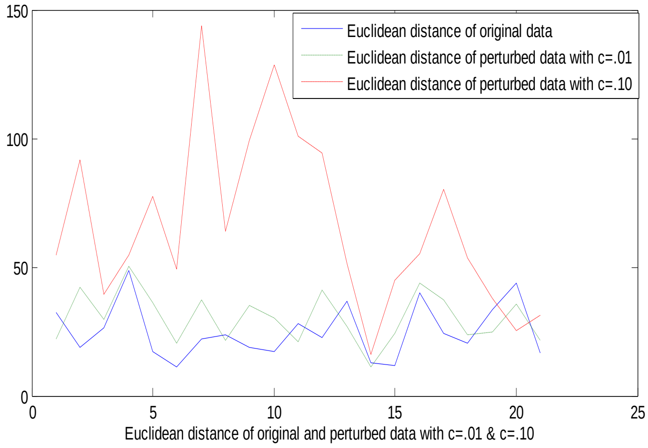

Figure 3.4

We have taken natural logarithm of original data and then the covariance is computed of this logarithm data. We have used the mvnrnd() function of matlab on the above obtained data to get new datasets through multivariate Gaussian distribution with c=.01 and c=.10. Exponential of the resultant matrix is the noise datasets. These resultant noise data sets are multiplied with the original data set to form the perturb datasets. we have evaluated mean of original and perturbed datasets with mean() fuction of Matlab. In graph 3.3 the blue line shows the mean of original data ,green line shows the mean of perturbed data with c=.01 and red line shows the perturbed data with c=.10. As seen in the graph, the estimates of the means from the perturbed datasets are all close to those from original data. We use pdist() function of Matlab to compute the Euclidian distance of original data set and the perturbed datasets.

We have plotted the graph 3.4 which shows the comparison between Euclidean Distances of original data and perturbed data after applying Perturbation Scheme II. In graph 3.4 the blue line shows the Euclidean distance of original data ,green line shows the mean of perturbed data with c=.01 and red line shows the perturbed data with c=.10.

The above graph shows that although the original attribute mean can be estimated from the perturbed data, but the Euclidean Distance among the data records are not necessarily preserved after perturbation.

II. CONCLUSION

This research paper reviews Second traditional multiplicative data perturbation technique that have been studied in statistics community. The effectiveness of multiplicative data perturbation techniques for privacy preserving data mining have been analyzed and also the security of multiplicative data perturbation scheme after applying logarithmic transformation have been examined. These perturbations are primarily used to mask the private data while allowing summary statistics (e.g., sum, mean, variance and covariance) of the original data to be estimated.

This approach is to take a logarithmic transformation, compute a covariance matrix of the transformed data, generate random number which follows mean 0 and variance/covariance c times the variance/covariance computed in the previous step, add the noise to the transformed data and take antilog of the noise added data. Both schemes were tried on students result data.

On the surface, multiplicative perturbation seems to change the data more than additive perturbation. However, by taking logarithms on the perturbed data, the multiplicative perturbation turns into an additive perturbation.

For perturbation scheme II, after logarithmic transformation, we have ln xij+eij . The noise term is chosen from N(0, c_lnX), where _lnX is the covariance of the original data in log scale.

The objective of these perturbation schemes is to mask the private data while allowing summary statistics to be estimated. However, problems in data mining are somewhat different. Data mining techniques, such as clustering, classification, prediction and association rule mining, are essentially relying on more sophisticated relationships among data records or data attributes, but not simple summary statistics. The traditional multiplicative perturbations distort each data element independently, therefore Euclidean distance and inner product among data records are usually not preserved, and the perturbed data cannot be used for many data mining applications.

These perturbation schemes are equivalent to additive perturbation after the logarithmic transformation. Due to the large volume of research in deriving private information from the additive noise perturbed data, the security of these perturbation schemes is questionable.

III. REFERENCES

[1]. N. R. Adam and J. C. Worthmann, "Security-control methods for statistical databases: a comparative study," ACM Computing Surveys (CSUR), vol. 21, 145-146 no. 4, pp. 515–556, 1989.

[2]. R. Agrawal and R. Srikant, "Privacy preserving data mining," in Proceedings of the ACM SIGMOD Conference on Management of Data, Dallas, TX, May 2000, pp. 439–450.

[3]. D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in Proceedings of the twentieth ACM SIGMOD- IGACT-SIGART symposium on Principles of Database Systems, Santa Barbara, CA, 2001, pp. 247–255.

[4]. H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in Proceedings of the IEEE International Conference on Data Mining, Melbourne, FL, November 2003.

[5]. S. Guo and X. Wu, "On the use of spectral filtering for privacy preserving data mining," in Proceedings of the 21st ACM Symposium on Applied Computing, Dijon, France, April 2006, pp. 622–626.

[6]. J. J. Kim and W. E. Winkler, "Multiplicative noise for masking continuous data," Statistical Research Division, U.S. Bureau of the Census, Washington D.C., Tech. Rep. Statistics #2003-01, April 2003.

[7]. Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in Proceedings of the 2005 ACM SIGMOD Conference, Baltimroe, MD, June 2005, pp. 37–48.

[8]. K. Muralidhar, R. Parsa, and R. Sarathy, "A general additive data perturbation method for database security," Management Science, vol. 45, no. 10, pp. 1399–1415, 1999.