MATHEMATICS, INFORMATICS AND STATISTICS
www.csiro.au

Confidentialising Count and Magnitude Data

Heejae Lee, Jeongin Choi
Christine M O'Keefe

August 2012

Commercial In Confidence

Korea University Department of Statistics
Anam-Dong, Seongbuk-Gu, Seoul 136-701 KOREA

Copyright and disclaimer

Important disclaimer

CSIRO advises that the information contained in this publication comprises general statements based on scientific research. The reader is advised and needs to be aware that such information may be incomplete or unable to be used in any specific situation. No reliance or actions must therefore be made on that information without seeking prior expert professional, scientific and technical advice. To the extent permitted by law, CSIRO (including its employees and consultants) excludes all liability to any person for any consequences, including but not limited to all losses, damages, costs, expenses and any other compensation, arising directly or indirectly from using this publication (in part or in whole) and any information or material contained in it.

Contents

Part I

Introduction

These days, protecting privacy for individual data is getting important in statistical analysis, since there are many possibilities to disclose confidential information about individual units during research. As a way of preventing disclosure risk, it is needed to adjust data through confidentialising measures. In this paper, we deal with several confidentialising methods, especially focusing on *re-distribution*.

To examine how to confidentialise data, we classify the data into two categories – *count data* and *magnitude data*. Count data contain only frequency for each cell, whereas magnitude data have the value of each unit as well as the count, which is the number of units.

In count data, we specify some procedures to detect and modify sensitive cell which is the main reason to generate disclosure risk. For modifying sensitive cells, we design two different cases of re-distribution through example using hexplot. Next, evaluation based on risk and utility loss is added to each method.

Also, this paper presents other discovering rules for sensitive cell and re-distribution for each rule in magnitude data.

Part II

Count data

1 How to find sensitive cells?

1.1 Threshold rule

From now, suppose we are only dealing with the *data cell*, which has non-zero count. A data cell is considered to be sensitive if the cell count is less than a minimum threshold number (say 3, 5 or 10) of records. Because the population size is $n = 500$ (See section 3), the minimum threshold number is selected to 3. (see [1])

Therefore, based on this rule, the cell which has the count less than 3 is identified to be sensitive and needs to be protected by applying confidentialisation methods.

2 How to handle the sensitive cells?

Sensitive cell can be classified as two cases, outlier and non-outlier.

2.1 Outlier

2.1.1 Definition

Based on graph theory, two cells are *adjacent* if they share an edge, one side of hexagon. Also, *path* from $cell_1$ to $cell_n$ is the way through $cell_1, cell_2, ..., cell_n$ where $cell_i$ and $cell_{i+1}$ are adjacent. The *distance* between $cell_1$ and $cell_n$ is defined as the length of shortest path from $cell_1$ to $cell_n$. We can also compute the distance in the same way when $cell_1$ is a *cluster*, which is a set of cells where there is a path between any 2 cells in the cluster. That is, distance from a cell to a cluster of any size is minimum distance of the cell to any cell in the cluster.

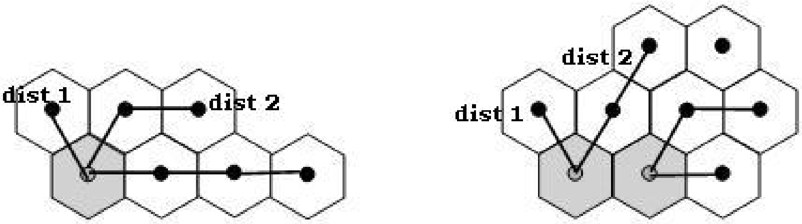


Figure 2.1: Distance

A data cell is regarded to be an outlier if it has the count c less than 3 and there are no data cells within distance c. In the case of a pair of cells, they are also considered to be outliers when each of them has the count 1 and there are no data cells in distance greater than count 2.

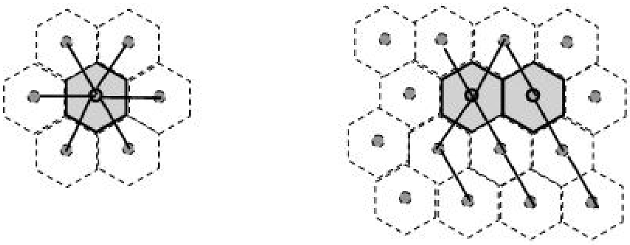


Figure 2.2: Outlier

2.1.2 Method 1 : Cell Suppression

Cell Suppression is traditional measure which protects the sensitive cells by hiding the counts of them. It deletes the cell counts. (see [2]) In tabular data, suppression replaces deleted cell counts with some symbols to distinguish them from zero. However, since our data are represented as hexplot and they have their own locations, an outlier is just eliminated from dataset. In this case, $mass = count * area$ is not preserved. If this method generates new outlier, then iterate until there is no outlier.

2.1.3 Method 2 : Re-distribution

In hexplot, mass needs to be maintained after confidentialising, so we need to apply new method, *re-distribution*. First, find the closest data cell around the outlier. When the outlier has count 1, just transfer 1 to the closest data cell. Otherwise, in case of count 2, distribute count 1 each to the two closest data cells.

2.2 Non-outlier

2.2.1 Method 1

In tabular data, sensitive cell which is not an outlier needs to be adjusted by tabular confidentialising measures, including *suppression, perturbation, aggregation*, and so on.(see [1])

2.2.2 Method 2 : Re-distribution

Same as outliers, mass needs to be preserved in hexplot different from tabular data. Therefore, it is better to apply *re-distribution*, but not the same as above. For the purpose of this discussion, we constructed two algorithms for re-distribution.

Algorithm 1

1. Suppose each of data cell counts $c_1, c_2, ..., c_i, ..., c_n$.
2. Select one sensitive cell.
3. Find the smallest cluster of n cells including the sensitive cell, with total count $\Sigma c_i \geq 3n$. If it is not possible, make size of hex's bigger and start again. However, it is not recommended to always do this because it reduces information on whole map and it is better to change only the areas around the sensitive cells.
4. Starting from the selected sensitive cell, take count 1 or 2 from the nearest and largest cell which has the count larger than 3 to make the sensitive cell have count 3. When the contributing cell becomes 3, stop and repeat with another nearest cell.
5. Suppose the sum of each moving distance of count 1, which is the distance between the origin and the destination cells, $\Sigma d_i = \mu$. In this process, μ should be minimized.

Algorithm 2

1. Similar to Algorithm 1, it needs to be premised that $\Sigma c_i \geq 3N$, when the total number of cells is N.
2. Compute the deficit and excess of all cells in the standard of count 3.
3. Under the principle that each cell having count larger than 3 gives count 1 equally to the sensitive cells, nearest one around sensitive cell gives count 1 first, if there is difference between total deficits and excesses.
4. Same as above, $\Sigma d_i = \mu$ should be minimized.

3 Example

3.1 Data

We used data as constructed by Reiter.(see [3]) Among eight models, we selected three data sets which are suitable for aggregating sensitive cells.

The simulations include three data sets with $n = 500$ observations each. Each data set is comprised of an independent variable, x_1, whose counts are drawn randomly from a normal distribution with mean equal to five and variance equal to one. Values of dependent variable, x_2, differ across the data sets. They are described in Table 3.1. For each of these three scenarios, the model is an ordinary least squares regression of x_2 on x_1. Figures 3.2.(a), 3.3.(a), and

3.4.(a) are hexplots of residuals after regression on x_1.

Description	Generation model for dependent variable
Piecewise	$x_2 = 20 + \epsilon$, for $x_1 < 4.5$
	$x_2 = 10 + 8x_1 + \epsilon$, for $4.5 \leq x_1 < 5.5$
	$x_2 = -4 + 14x_1 + \epsilon$, for $x_1 \geq 5.5$
Outliers	$x_2 = 10x_1 + \epsilon + 30\alpha - 25\beta$, where $\alpha = 1$ for $x_1 = 3.8$, and $\beta = 1$ for $x_1 = 5.8$
Curvilinear	$x_2 = 10(x_1 - 5) - 5(x_1 - 5)^2 + \epsilon$

Table 3.1: Data table

3.2 Simulation

As discussed in Chapter 2, we applied those two algorithms to our data sets expressed as hexplot. Each cell colour represents the count and you can see the legend on the right side of the figure.

<Algorithm 1>

See the Figure 3.1. As the two cells at locations $(1.8, 4.8)$, $(2.6, 3.25)$ are outliers, suppress them. Here, we just suppressed outliers rather than re-distribution.

In the case of small cluster consisting of five cells, it is not an outlier based on the definition of an outlier above. The cluster is close enough to the other data cells that re-distribution between them needs only a small change in the distance each value is moved.

Start from the cell with count 1 in the small cluster. Because it cannot find its neighbouring cells with count greater than 5, find the smallest group of cells making $\Sigma c_i \geq 3n$. Here, n becomes 25 and Σc_i is equal to 75 .The cluster includes the cells up to the one with count 6, located on $(3.75, -0.6)$. As a result of aggregation, all cells become to have count 3.

Otherwise, if neighbouring cells are enough to make sensitive cell have count 3, the largest among neighbours gives the count.

Finally we can generate Figure 3.2.(b). In this process, μ becomes 279.

<Algorithm 2>

Same as above, we suppress the two outliers.

Compute the total deficit and the number of excessive cells. Then the former becomes 67 and the latter becomes 41. Therefore, take count 1 from all 41 excessive cells equally and select duplicative 26 nearest cells around every sensitive cell.

In this process, μ becomes 403, which is larger than that of algorithm 1. Figure 3.2.(c) is the result of all the procedure.

By the same way, we can make Figure 3.3.(b),(c), and Figure 3.4.(b),(c) on each model and they have the $\mu = 159, 244, 456, 463$ respectively.

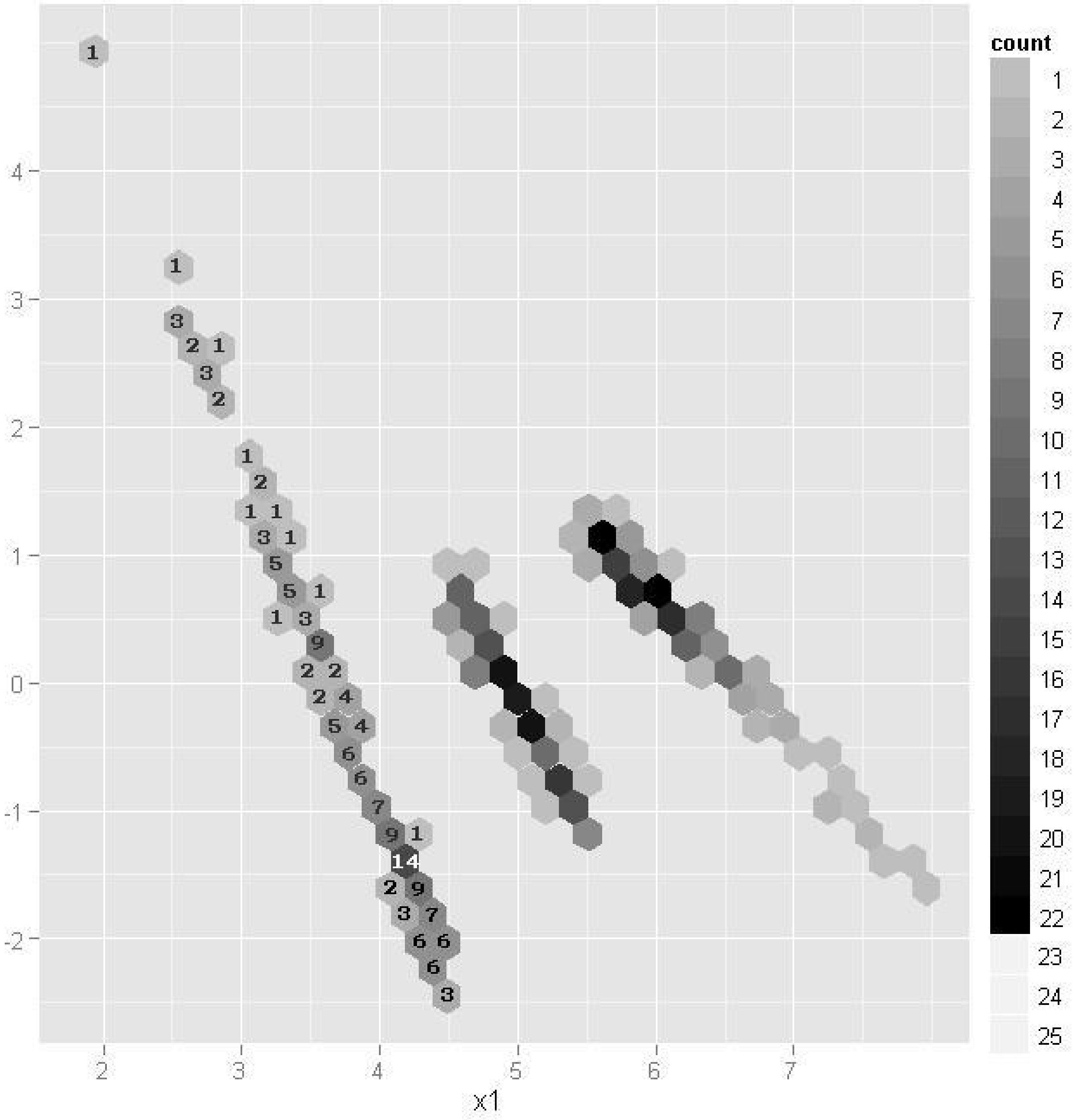
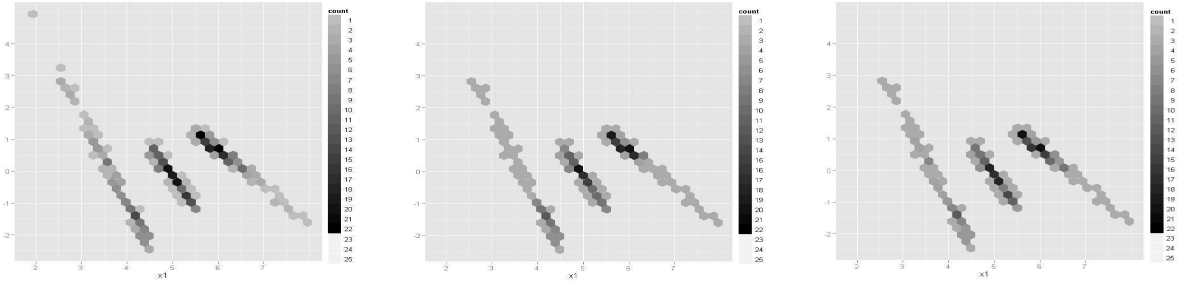
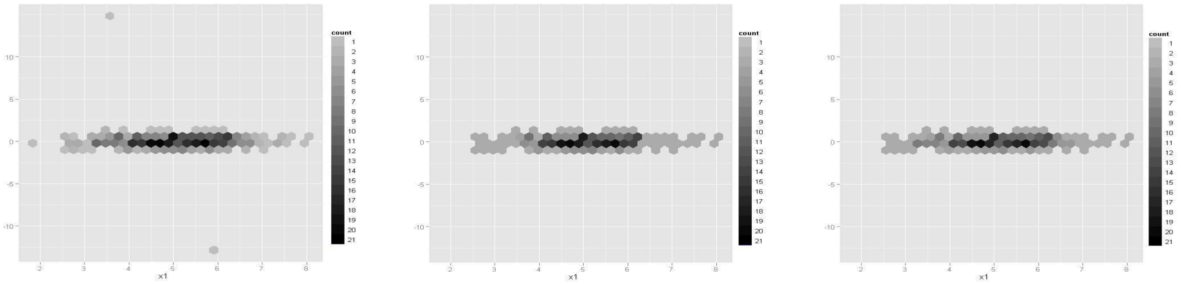


Figure 3.1: Hexplot of piecewise data before re-distribution



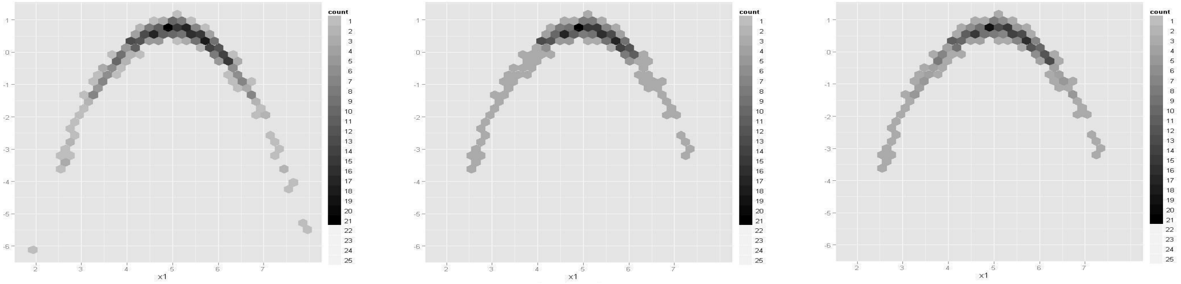
(a)Original (b)Algorithm1$(\mu = 279)$ (c)Algorithm2$(\mu = 403)$

Figure 3.2: Piecewise



(a)Original (b)Algorithm1$(\mu = 159)$ (c)Algorithm2$(\mu = 244)$

Figure 3.3: Outliers



(a)Original (b)Algorithm1$(\mu = 456)$ (c)Algorithm2$(\mu = 463)$

Figure 3.4: Curvilinear

4 Discussion

Up to now, we developed two different measures of aggregation to protect privacy of sensitive cells in tabular data. The most important thing is that the confidentialised map is similar to the original map. Therefore, distance of moving has to be minimized and it means that choosing the smallest μ is the best adjustment.

Of the two algorithms, the second one always has larger distance of move since its sensitive cell takes the count from all of the excessive ones, not the nearest ones. Thus, method 1 has the minimum μ and is more efficient than any other methods.

5 Evaluations

5.1 Risk-Utility Analysis

To evaluate our algorithms, we used Risk-Utility Analysis as proposed by Marley and Leaver.(see [5]) There are several ways for measuring risk and utility after confidentialising.

1. Measuring Risk :

   1) Inverse of the variance of the confidentialised counts

   Let u denote the original (unconfidentialised) cell count, c denote the confidentialised cell count, and p denote the amount of change so that $c = u + p$. Then, higher variance of c indicates lower risk of discovering u. Therefore, the measure of risk can be expressed as

   $$R_1(u) = [Var(c|u)]^{-1} = [Var(p|u)]^{-1}.$$

   2) Percentage of cells that are unchanged

   The percentage of cells that are unchanged is equivalent to the percentage of cells that are re-distributed by $p = 0$. This measure of risk is

   $$R_2(u) = P(p = 0|u)$$

2. Measuring Utility loss :

   1) Average percentage change

   At the cell level, we can measure the utility loss by the expected or average percentage change of the unconfidentialised cell count. That is,

   $$U_1(u) = APC = \sum_{p\prime=P_L}^{P_U} \frac{|p\prime|}{u} P(p = p\prime|u) \times 100\%$$

   2) Average absolute difference

   The Absolute Average Difference is the mean of the absolute value of the difference between the original and the confidentialised cell values :

   $$U_2 = AAD = \frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J}|u_{ij} - c_{ij}|$$

   3) Moving distance of count 1, μ

   Our μ also can be considered as an utility loss measure, because it represents the amount of change of the cell count during re-distribution.

   $$U_3(u) = \mu(u) = \frac{1}{2}\sum_{i=P_L}^{P_U}\sum_{j=1}^{|p|}(d_{ij}|u)$$

   Here, d_{ij} is a moving distance of a count 1 between the origin and the destination cells. When count 1 moves into a cell, the cell has positive p, otherwise negative p. Since we need to consider the moving in and out of count 1, we take the absolute value for p. Also, to prevent being computed twice, divide the sum of moving distance into two. Then, the sum of $\mu(u)$ given all the values of u, $\sum_{u=1}^{U}\mu(u)$ becomes μ as discussed before.

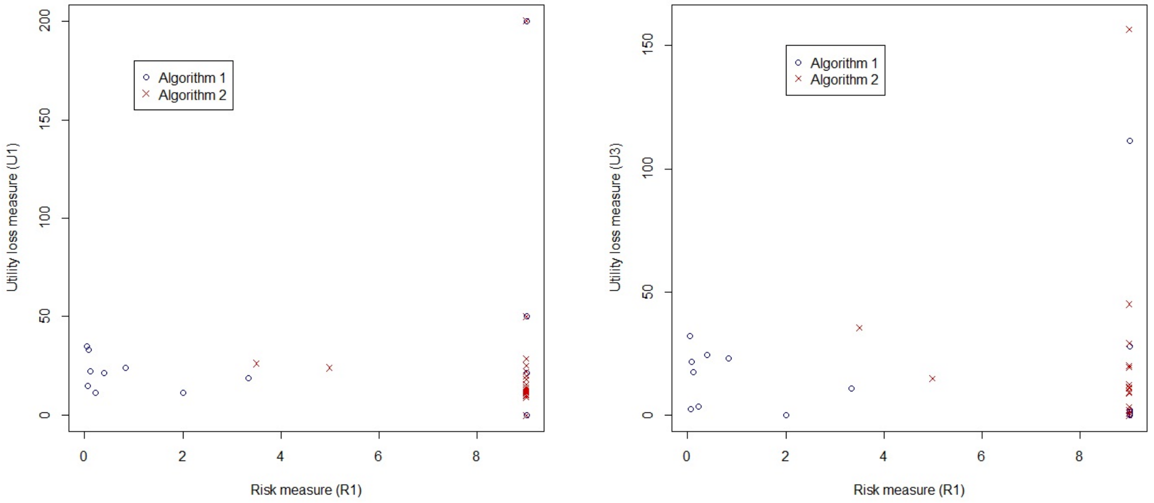
5.2 Example

As noted by Marley and Leaver, the graphs of $R_1(u)$ seem more adequate than $R_2(u)$ because thier empirical values for confidentialising appear to follow the pattern of the theoretical values very closely. It is the reason why we chose $R_1(u)$ to compare the risks between our two algorithms using the same data(Piecewise) in section 3.

Therefore, we made a risk-utility curve of $U_1(u)$ on $R_1(u)$(see Figure 5.1.(a)). The x-axis shows the amount of risk and the y-axis denotes the utility loss. For algorithm 1, the circles show the empirical values of $U_1(u)$ against the empirical values of $R_1(u)$ for all values of $u \in \{1, 2, ..., 22\}$. Also, x symbols represent those of algorithm 2.

Finally, we can conclude that algorithm 1 is better than algorithm 2 because it has less utility loss and risk when using risk measure $R_1(u)$.

Same as Figure 5.1.(a), we made a risk-utility curve of $U_3(u)$ on $R_1(u)$(see Figure 5.1.(b)). Here again, algorithm 1 shows smaller risk than algorithm 2. To examine the utility loss(U_3) in detail, see Figure 5.2 which is the plot of U_3 on unconfidentialised cell count u. The patterns of algorithm 1 and 2 may seem similar, but we can recognize that algorithm 1 preserves more utility than algorithm 2.



(a) (b)

Figure 5.1: Empirical risk-utility curve for Algorithm 1 and 2

Same as $U_1(u)$, algorithm 1 seems better in respect of utility loss measured by U_2. U_2 produces two scalars, 134 and 138 for each algorithm, which indicates algorithm 1 has less utility loss.

In conclusion, judging from discussions so far, it can be suggested that algorithm 1 is more efficient considering both risk and utility.

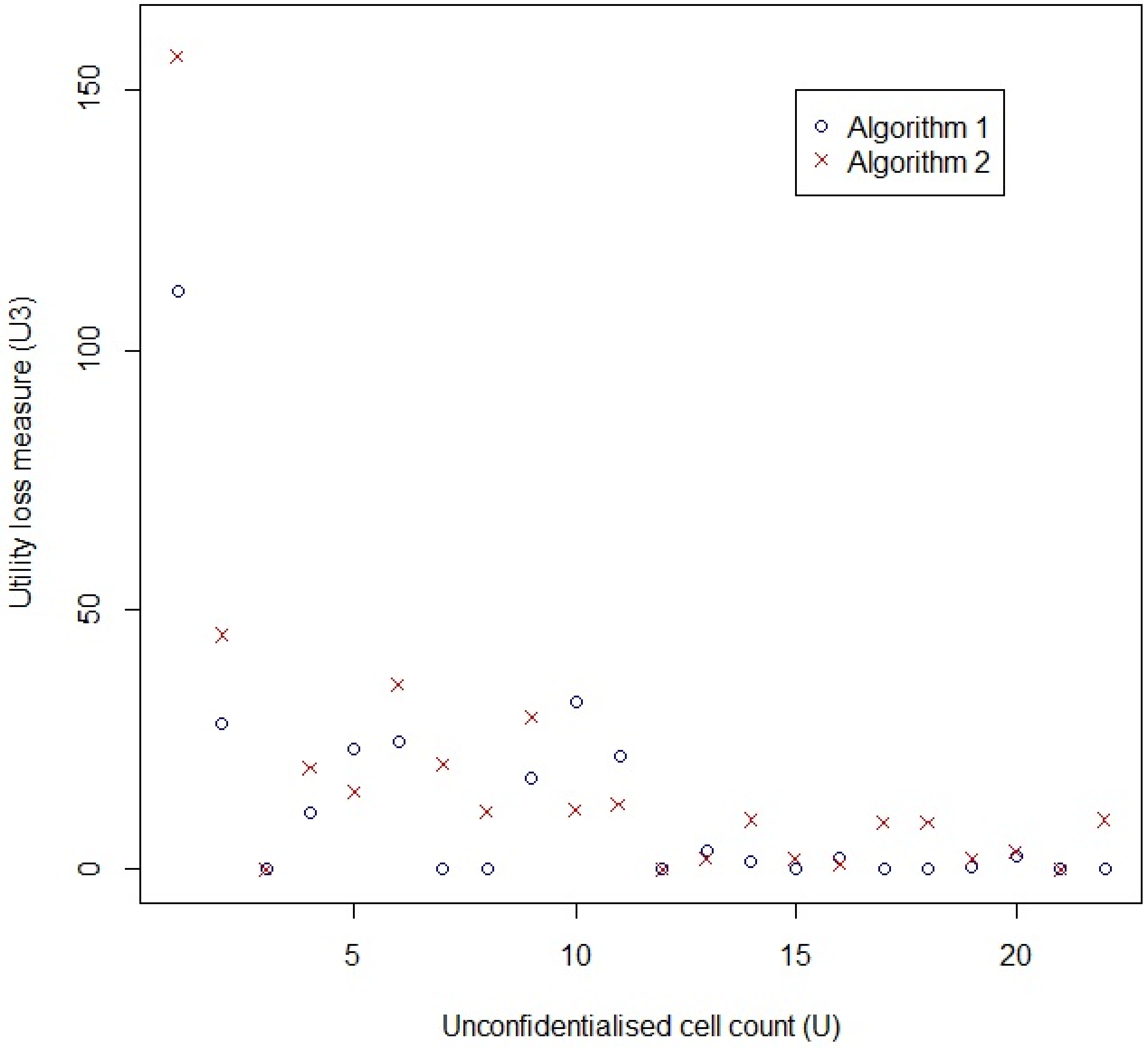


Figure 5.2: Empirical values of $U_3(u)$ for Algorithm 1 and 2

Part III

Magnitude data

In magnitude data, a data cell has not only count, which is the number of contributors, but also the value of each contributor.(see [4])

6 How to find sensitive cells?

6.1 Threshold rule

Same as count data, we define *data cell* as the cell with non-zero value. Also suppose all the cells are data cells. When the number of cell contributors is less than a minimum threshold number (say 3, 5 or 10), the data cell is regarded as sensitive. Here, the minimum threshold number is selected to 3. (see [1])

6.2 Dominance rule

There are three different cases for dominance rule. (see [1])

6.2.1 Maximum threshold rule

The *maximum threshold rule* identifies the sensitive cell when its largest contributor exceeds a maximum threshold percentage (say 90%) of the cell total. For example, if there are five contributing values of 48, 7, 4, 9, and 6, then the largest contributor 30 exceeds 90% of the total value. Therefore, the cell is determined to be sensitive.

6.2.2 (n, k) rule

The second rule is *(n, k) rule*. (see [2]) This rule has two parameters, a positive integer n and a percentage k. The cell contributors are sorted by decreasing order of their values and the cell is sensitive if fewer than largest n units contribute to at least k of the cell total value. In application, n is set to be less than 3 and k greater than or equal to 90. For instance, a cell with contributors 67, 105, 1, 8, 3, and 2 is sensitive when n is set to 2 and k to 90.

6.2.3 p-percent rule

Under the *p-percent rule*, a cell is said to be sensitive if the total contribution of all but the largest and second largest contributors is less than p-percent of the largest contribution. Practically, p is set to be less than 10. For example, when seven contributors have the value 178, 99, 2, 1, 4, 3, and 2 respectively, the cell is considered to be sensitive.

7 How to handle the sensitive cells?

7.1 Threshold rule

To adjust sensitive cells based on threshold rule, we can use the algorithm similar to that of count data.

1. Suppose each of the number of cell contributors $t_1, t_2, ..., t_i, ..., t_n$.

2. Select one sensitive cell.
3. Find the smallest group of n neighbouring cells including the sensitive cell, with total number of contributors $\Sigma t_i \geq 3n$. If it is not possible, it means there are too many sensitive cells, so this case is not suitable to apply this algorithm.
4. Start from the selected sensitive cell c_1. Take 1 or 2 contributors with smallest value from the nearest cell c_2 which is not sensitive, to make the sensitive cell have 3 contributors. When selecting c_2, among every neighbouring cell of c_1, find a contributor having the smallest value. The cell having this contributor becomes c_2. When t_2 becomes 3, stop and repeat with another nearest cell.
5. Suppose the sum of each moving distance of 1 contributor between the origin cell and the destination cell, $\Sigma d_i = \mu$. In this process, μ should be minimized.

7.2 Dominance rule

7.2.1 Maximum threshold rule

1. Suppose each of the values in a sensitive cell c_1, to be $v_1, v_2, ..., v_i, ..., v_n$ by decreasing order.
2. Replace v_1 with $v_1\prime$ which is the maximum threshold percentage of the total, mainly 50%.
3. To maintain the sum of all values, distribute the difference $v_1 - v_1\prime$ to the neighbouring cell c_2. When selecting c_2, among every neighbouring cell of c_1, find a contributor having the smallest value. The cell having this contributor becomes c_2.
4. Then distribute the difference according to the ratio among values in c_2.
5. Iterate the procedure until the cell is not sensitive by any other rules.

7.2.2 (n, k) rule

1. Suppose each of the largest n values in a sensitive cell c_1, to be $v_1, ..., v_n$ by decreasing order.
2. Replace Σv_i with $\Sigma v_i\prime$ which takes k percentage of the total. In this process, $v_1\prime, ..., v_n\prime$ should keep their initial ratio.
3. To maintain the sum of all values, distribute the difference $\Sigma v_i - \Sigma v_i\prime$ to the neighbouring cell c_2. When selecting c_2, among every neighbouring cell of c_1, find a contributor having the smallest value. The cell having this contributor becomes c_2.
4. Then distribute the difference according to the ratio among values in c_2.
5. Iterate the procedure until the cell is not sensitive by any other rules.

7.2.3 p-percent rule

1. Suppose each of the values in a sensitive cell c_1, to be $v_1, v_2, ..., v_i, ..., v_n$ by decreasing order.
2. Replace v_1 with $v_1\prime$ which is $p\%$ of the total value except for v_1 and v_2.
3. To maintain the sum of all values, distribute the difference $v_1 - v_1\prime$ to the neighbouring cell c_2. When selecting c_2, among every neighbouring cell of c_1, find a contributor having the smallest value. The cell having this contributor becomes c_2.

4. Then distribute the difference according to the ratio among values in c_2.
5. Iterate the procedure until the cell is not sensitive by any other rules.

Part IV

Conclusion

So far, we discussed about preventing disclosure by confidentialising – re-distribution – in both count and magnitude data. Consequently, it is suggested to modify the information by choosing the method that makes μ and risk to the smallest. We think this technique would be useful to analyse both tabular and spatial data.

Acknowledgements

We'd like to thank Christine O'Keefe sincerely for her very helpful comments during the revision of this paper.

References

[1] C. O'Keefe, "Confidentialising maps of mixed point and diffuse spatial data"

[2] G.T. Duncan and M. Elliot, *statistical Confidentiality*, Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4419-7802-8$_4$,2011.

[3] J. Reiter, "Model diagnostics for remote access regression servers," 2003

[4] N. Shlomo and C. Young, "Statistical disclosure control methods through a risk-utility framework", 2006

[5] J.K. Marley and V.L. Leaver, "A method for confidentialising user-defined tables: Statistical properties and a risk-utility analysis", 2011

CONTACT US

t 1300 363 400
+61 3 9545 2176
e enquiries@csiro.au
w www.csiro.au

YOUR CSIRO

Australia is founding its future on science and innovation. Its national science agency, CSIRO, is a powerhouse of ideas, technologies and skills for building prosperity, growth, health and sustainability. It serves governments, industries, business and communities across the nation.

FOR FURTHER INFORMATION

CSIRO Mathematics, Informatics and Statistics
Christine M OKeefe
t +61 3 9545 8043
e Christine.OKeefe@csiro.au
w

Korea University Department of Statistics
Heejae Lee
t +82 10 7190 2448
e statlee712@korea.ac.kr
w

Korea University Department of Statistics
Jeongin Choi
t +82 10 3311 3455
e jeongin0810@korea.ac.kr
w