Available online at www.sciencedirect.com
ScienceDirect

SoftwareX 3–4 (2015) 37–43

SoftwareX

www.elsevier.com/locate/softx

MaskDensity14: An R package for the density approximant of a univariate based on noise multiplied data

Yan-Xia Lin[a,*], Mark James Fielding[b]

[a] National Institute for Applied Statistics Research Australia, School of Mathematics and Applied Statistics, University of Wollongong, NSW 2500, Australia
[b] DHI Water & Environment Singapore, 1 Cleantech Loop, ♯03-05 CleanTech One, Singapore 637141, Singapore

Received 13 July 2015; received in revised form 28 October 2015; accepted 1 November 2015

Abstract

Lin (2014) developed a framework of the method of the sample-moment-based density approximant, for estimating the probability density function of microdata based on noise multiplied data. Theoretically, it provides a promising method for data users in generating the synthetic data of the original data without accessing the original data; however, technical issues can cause problems implementing the method. In this paper, we describe a software package called *MaskDensity14*, written in the R language, that uses a computational approach to solve the technical issues and makes the method of the sample-moment-based density approximant feasible. *MaskDensity14* has applications in many areas, such as sharing clinical trial data and survey data without releasing the original data.

Keywords: Confidential data; Data masking; Multiplicative noise; Sample-moment-based density approximant

Code metadata

Current code version	version 1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-15-00040
Legal Code License	GPL-2
Code versioning system used	none
Software code languages, tools, and services used	R
Compilation requirements, operating environments & dependencies	Can run on any operating system that is supported by R.
If available Link to developer documentation/manual	
Support email for questions	yanxia@uow.edu.au

1. Motivation and significance

Confidential data are not allowed to be issued to the public without certain levels of protection. A number of methods for protecting data have been recommended and used in practice (see Duncan and Lambert [1], Willenborg and De Waal [2], Oganian [3], Shlomo [4], and references therein).

The multiplicative noise method is one method for providing data protection (Kim and Jeong [5]). The method is briefly described as follows. Let $y_1, y_2, \dots, y_N$ (the original data) be a sample drawn from a sensitive random variable Y. Let C be a positive random variable, independent of Y. When we say the original data $y_1, y_2, \dots, y_N$ were masked by C, it means their masked data have the form $y_i^* = y_i \times c_i, i = 1, 2, \dots, N$, where $\{c_i\}$ is a sample from C. The original data $\{y_i\}_1^N$ are protected by $\{y_i^*\}_1^N$. The system of releasing noise multiplied data is a non-open query based system.

[*] Corresponding author.
E-mail address: yanxia@uow.edu.au (Y.-X. Lin).

http://dx.doi.org/10.1016/j.softx.2015.11.002

A key issue is how to recover the statistical information of the original data based on their noise multiplied data. Recently, many data analysis methods to recover the original information from noise multiplied data have been developed (Kim and Jeong [5]; Sinha, et al. [6]; Hwang [7] and Lin and Wise [8]). These methods are not standard, and some of them are complex to use. As a consequence, Lin [9] introduced a framework on the sample-moment-based density approximant. The framework provides a method for estimating the probability density function of the original data based on noise multiplied data. It gives a solution for generating the synthetic data of the original data without accessing the data themselves. Thus, using standard statistical inference methods to analyze the original data based on noise multiplied data becomes possible. Developing software to implement the method is desirable.

We briefly describe the computational statistical approach developed by Lin [9] as follows. Let $\{y_i\}_1^N$ be a set of original sample data drawn from a random variable Y. The data were masked by a noise C which yielded masked data $\{y_i^*\}_1^N$. Let $\{c_i\}_1^N$, **not the same sample** used to mask $\{y_i\}_1^N$, be another independent sample drawn from C. Assume that Y has moment generating function. Thus, the density function f_Y of Y can be determined by its moments. The sample-moment-based density approximant of the density function of Y is defined as

$$f_{Y,K|\{y_i^*,c_i\}_1^N}(y) = \sum_{k=0}^{K} a_k(y) \frac{\overline{(Y^*)^k}}{\overline{C^k}} \tag{1}$$

where $\overline{(Y^*)^k} = \sum_{i=1}^N (y_i^*)^k/N$ and $\overline{C^k} = \sum_{i=1}^N c_i^k/N$; $a_k(y) = a_k(y; a, b)$ is a polynomial function of y, i.e. a continuous function of y (the details see Lin [9]), where a and b, used in Lin [9], are $\min_{1\le i\le N}\{y_i\}$ and $\max_{1\le i\le N}\{y_i\}$, respectively. Lin [9] showed that $f_{Y,K|\{y_i^*,c_i\}_1^N}$ can well represent the density function of Y given that the sample size N and the upper order of moment K are appropriate (for brevity, we sometimes use "the upper order K" instead of "the upper order of moment K"). Thus, the sample drawn from $f_{Y,K|\{y_i^*,c_i\}_1^N}$ can be considered as synthetic data of the original data. The upper order K was determined with reference to the density function f_Y of the original data in Lin [9].

To implement the technique developed by Lin [9] in practice, there are a number of technical issues to solve. Firstly, the upper order K has to be determined without the reference to f_Y, as f_Y is not available. Secondly, it is desirable that the boundaries a and b are determined without using the information $\max_{1\le i\le N}\{y_i\}$ and $\min_{1\le i\le N}\{y_i\}$ directly, as the information might be confidential.

2. Software description

The *MaskDensity14* software presented in this paper uses the R language, which is widely used system by statisticians (see The R Package for Statistical Computing http://www.r-project.org). *MaskDensity14* follows the standard manner to obtain the smoothed function of the sample-moment-based density approximant $f_{Y,K|\{y_i^*,c_i\}_1^N}$. Based on (1), *MaskDensity14* (Version 1.0) evaluates $f_{Y,K|\{y_i^*,c_i\}_1^N}(y)$ at 512 equal distance points on $[a, b]$. Then, *MaskDensity14* uses standard smoothing and normalizing techniques to obtain the smoothed sample-moment-based density approximant $f_{Y,K|\{y_i^*,c_i\}_1^N}$. The kernel R package adopted by *MaskDensity14* is *ks* (see http://cran.r-project.org/web/packages/ks/ks.pdf).

Determining an appropriate value for the upper order K and boundaries a and b for $f_{Y,K|\{y_i^*,c_i\}_1^N}$ is a critical issue. The upper order K in Lin [9] is determined by comparing the plots of $f_{Y,K|\{y_i^*,c_i\}_1^N}$ and f_Y. Using this way to determine the value for K is impracticable as the original data are not available. The boundaries a and b used in Lin [9] are $\min\{y_i\}$ and $\max\{y_i\}$, respectively. Using those values in $f_{Y,K|\{y_i^*,c_i\}_1^N}$ might cause problems because the values might be confidential, and the data provider might feel uncomfortable to release them to the public.

A method for determining K and boundaries a and b, without directly employing the information of the original data is essential for the software built in this paper.

2.1. Determination of the upper order K in $f_{Y,K|\{y_i^*,c_i\}_1^N}$

Provost [10] pointed out that, if an inappropriate upper order K is used in the density approximant $f_{Y,K}$, it may cause $f_{Y,K}$ taking negative values. Simulation studies by Lin [9] showed that the density approximant will not be more accurate for large values of the upper order K. It is a challenge to determine an appropriate K for $f_{Y,K|\{y_i^*,c_i\}_1^N}$ without reference to f_Y.

Lin [9] used a term "the correlation between two density functions" to evaluate the similarity of two density functions. The implication of the term is related to the concept of "the probability plot correlation coefficient". The Q–Q plot method can be used to compare two probability distributions if they are close to each other. Adopting the idea of the Q–Q plot, we evaluate the value of "the correlation between density functions f_1 and f_2" in the following manner. (1) Independently simulate two samples $\{x_{1,i}\}$ and $\{x_{2,i}\}$ of the same size from f_1 and f_2, respectively. (2) Sort the two samples, and then calculate the sample correlation coefficient of the sorted samples.

Lin [9] demonstrated that the larger the value of "the correlation between $f_{Y,K|\{y_i^*,c_i\}_1^N}$ and f_Y" is, the closer the two functions, $f_{Y,K|\{y_i^*,c_i\}_1^N}$ and f_Y, will be. Motivated by this fact, the following steps are built in *MaskDensity14* for determining the appropriate K in $f_{Y,K|\{y_i^*,c_i\}_1^N}$ without directly using the information of the original data $\{y_i\}_1^N$. Consider the masked dataset $\{y_i^*\}$:

Step 1. Set an initial upper order of moment, $K = 1$ and a maximum upper order of moment to be tested. The maximum upper order of moment set in *MaskDensity14* is 100.

Step 2. Independently simulate a sample $\{c_i\}_1^N$ from C and obtain the smoothed function $f_{Y,K|\{y_i^*,c_i\}_1^N}$ using Eq. (1). In *MaskDensity14*, we assume that the data agency provides the data user with a sample of C. The size of the sample is sufficiently large ($>N$) such that the sample can well represent the probability structure of the noise C. Thus, a sample drawn from the sample population can be considered as a sample drawn from C.[1]

Step 3. Simulate a sample $\{y'_j\}_1^N$ from $f_{Y,K|\{y_i^*,c_i\}_1^N}(y)$.

Step 4. Independently simulate a second sample $\{c'_j\}_1^N$ from C. Mask $\{y'_j\}_1^N$ by using this new sample of noise and yield a new masked dataset $\{y_j'^*\}_1^N$.

Step 5. Sort $\{y_i'^*\}_1^N$ and $\{y_i^*\}_1^N$, respectively. Evaluate the correlation $Cor(K)$ between the two sorted datasets. Keep track of the optimum upper order of moment such that $Cor(K_{\text{opt}}) = \max_{k \le K} Cor(k)$.

Step 6. Update K to $K+1$ and return to Step 2 if $K+1 \le 100$. Stop when $Cor(K)$ drops below a threshold taken as $Cor(K) < 1 - 10\left[1 - Cor(K_{\text{opt}})\right]$ or $K+1 > 100$.

Step 7. Report K_{opt} as the optimum upper order of moment used.

Remarks. (1) Step 5 is the key step in identifying an appropriate upper order of moment for the approximant of f_Y. We explain the logic used to support Step 5 as follows. If the density approximant determined by $\{y_i^*, c_i\}$ is close to the true density function f_Y, $\{y'_i\}$ can be considered as an independent sample from Y. Consequently, $\{y_i'^*\}$ can be considered as an independent sample from YC. Thus, the smoothed density functions determined by $\{y_i^*\}$ and $\{y_i'^*\}$, respectively, will be more likely close to each other.

MaskDensity14 reports the value of "the correlation between the smoothed density functions" given by $\{y_i^*\}$ and $\{y_i'^*\}$. The higher the value is, the relatively better the approximation between $f_{Y,K|\{y_i^*,c_i\}}(y)$ and $f_Y(y)$ will be.

(2) We set the tested maximum upper order of moment to be 100. To save time, we would not like to have the testing procedure going from $K = 1$ to $K = 100$. Our experience (also see examples in Lin [9]) shows that $Cor(K)$ will decrease quite rapidly if K becomes too large. It is due to the result of poor estimates of high-order moments. If the value of $Cor(K)$ decreases too low and

$$1 - Cor(K_{\text{opt}}) < \frac{1}{10}\left[1 - Cor(K)\right],$$

according to our empirical testing, it is not necessary to carry out any further testing. Therefore, we set the threshold $1 - 10[1 - Cor(K_{\text{opt}})]$ in Step 6.

2.2. The boundaries a and b used in MaskDensity14

Example 1 shows the impact of the values of a and b on the plot of $f_{Y,K|\{y_i^*,c_i\}}$.

Example 1. Simulate a sample $\{y_i\}_1^{2000}$ from $N(5, 3^2)$. To purely focus on the impact of the boundaries a and b on the performance of $f_{Y,K|\{y_i^*,c_i\}_1^{2000}}$ without any interference from the noise C, we let $C = 1$.

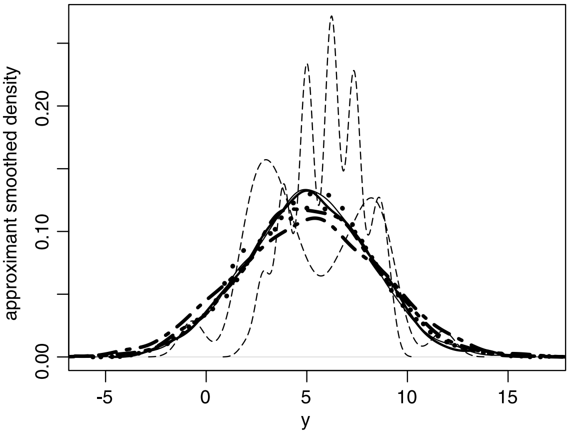


Fig. 1. The smoothed density function for $N(5, 3^2)$ is in (thin) solid line. The approximant smoothed density function $f_{Y,K|\{y_i^*,c_i\}}$ given by PB(1) and PB(2) are in (thin) longdash line; given by PB(3) (i.e. $a = \min\{y_i\}$ and $b = \max\{y_i\}$) in (bold) solid line; given by PB(4) and PB(6) in (bold) dotted line; given by PB(7) and PB(11) in (bold) twodashed line.

To well represent the impact of the boundaries on the plot of $f_{Y,K|\{y_i^*,c_i\}}$, seven pair-boundary (a, b)s: PB(j) = $(\min\{y_i\} + (2 - j + 1)s, \max\{y_i\} - (2 - j + 1)s)$, $j = 1, 2, 3, 4, 6, 7, 11$, are considered, where s is the sample standard error given by $\{y_i\}_1^{2000}$.

The domains determined by the pair-boundaries are subsets of the others in order. The shortest domain is $[\min\{y_i\} + 2s, \max\{y_i\} - 2s]$ and the longest one is $[\min\{y_i\} - 8s, \max\{y_i\} + 8s]$. The pair-boundary PB(3) is determined by $a = \min\{y_i\}$ and $b = \max\{y_i\}$, used as a reference.

The plots of $f_{Y,K|\{y_i^*,c_i\}}$ based on the seven pair-boundaries are presented in Fig. 1. Fig. 1 shows that the plots of the density approximants given by PB(1) and PB(2) are very different from the plot of the true density function; the plots given by PB(4) and PB(6) are reasonable. The values of $Cor(K_{opt})$ based on PB(4) and PB(6) are all around 0.9997, which are higher than those of $Cor(K_{opt})$ based on PB(1) and PB(2) (0.9873 and 0.9973, respectively). It confirms that, based on the same sample $\{y_i\}$, a density approximant with a relatively higher value of $Cor(K_{opt})$ should give a better approximation of f_Y. As the size of the interval $[a, b]$ increases, the plot of the corresponding density approximant tends to be flat and gradually runs away from the plot of the true density function (see the plots given by PB(7) and PB(11)).

Based on the simulation studies carried out in Example 1 and other examples (not shown here for saving space), the impact of a and b on $f_{Y,K|\{y_i^*,c_i\}}$ can be summarized as follows:

(1) If $[a, b]$ is a subset of $[\min\{y_i\}, \max\{y_i\}]$ with a size much smaller than the size of $[\min\{y_i\}, \max\{y_i\}]$, $f_{Y,K|\{y_i,c_i\}}$ might have fewer chances of being a good approximation of f_Y as $f_{Y,K|\{y_i,c_i\}}$ has to squeeze all the information provided by $\{y_i\}$ into the smaller interval $[a, b]$.

(2) If $[a, b]$ is close to $[\min\{y_i\}, \max\{y_i\}]$ (either a subset or a superset), $f_{Y,K|\{y_i^*,c_i\}}$ is able to give a good approximation of f_Y. Particularly, the difference between the approximants of the density of Y based on domain $[\min_{1\le i\le N}\{y_i\}, \max_{1\le i\le N}\{y_i\}]$ and $[a, b] \supseteq [\min_{1\le i\le N}\{y_i\}, \max_{1\le i\le N}\{y_i\}]$ is not significant, because both approximants are evaluated based on

[1] We can modify *MaskDensity14* and allow the data agency to provide the data user with the probability distribution information of C, instead of a sample from C.

the same sample $\{y_i\}_1^N$ and no $\{y_i\}_1^N$ fall within intervals $[a, \min_{1\le i\le N}\{y_i\})$ and $(\max_{1\le i\le N}\{y_i\}, b]$. The smoothed density function defined on the interval $[a, b]$ will not add too much weight on $[a, \min_{1\le i\le N}\{y_i\})$ and $(\max_{1\le i\le N}\{y_i\}, b]$.

(3) As the size of the interval $[a, b] \supseteq [\min_{1\le i\le N}\{y_i\}, \max_{1\le i\le N}\{y_i\}]$ increases, the normalized smoothed function $f_{Y,K|\{y_i^*,c_i\}}$ based on the pair-boundary (a, b) has to spread more weight to the whole interval $[a, b]$ and the plot of the $f_{Y,K|\{y_i^*,c_i\}}$ will be flattened, compared to the plot of the $f_{Y,K|\{y_i^*,c_i\}}$ based on the pair-boundary $(\min_{1\le i\le N}\{y_i\}, \max_{1\le i\le N}\{y_i\})$.

Let $\{y_i\}_1^N$ be a sample from Y and $\{y_{sub,j}\}$ be a subset of $\{y_i\}_1^N$. Denote Y_{sub} the population of $\{y_{sub,j}\}$. The density approximants of f_Y and $f_{Y_{sub}}$ can be obtained from the masked data $\{y_i^*\}_1^N$ and $\{y^*_{sub,j}\}$, respectively. Since the probability structures of Y and Y_{sub} might not be the same, the appropriate pair-boundaries used in the density approximants of the two populations might not be the same.

With the full knowledge on the original data $\{y_i\}_1^N$, the data agency has no problems in providing the data user with an appropriate pair-boundary (a, b) for $f_{Y,K|\{y_i^*,c_i\}_1^N}$. It is impossible for the data agency to provide an appropriate pair-boundary (a, b) for $f_{Y,K|\{y_i^*,c_i\}_{sub}}$ without knowing in which subset $\{y_{sub,j}\}$ the data user might be interested.

To ensure that the data user has more freedom in exploring the statistical information of the full/subset of the original data, it is of interest how to determine an appropriate pair-boundary for $f_{Y_{sub}}$ based on the information of $\{y_i^*, c_i\}_1^N$ and the appropriate pair-boundary for $f_{Y,K|\{y_i^*,c_i\}_1^N}$ provided by the data agency.

Taking into account the discussions above, a standard procedure for determining a and b is adopted in *MaskDensity14*:

(1) If Y is a categorical variable taking values $1, 2, \ldots, M$, let $a = 0$, and $b = M + 1$ (see Section 2.3).

(2) If Y is not a categorical variable, the values of a and b are determined as follows:

Step 2.1 Let a_{basic} and b_{basic} be the boundaries determined by the data agency. With the full knowledge of the original data $\{y_i\}_1^N$, the data agency can find appropriate a_{basic} and b_{basic} such that $[a_{basic}, b_{basic}] \supseteq [\min_{1\le i\le N}\{y_i\}, \max_{1\le i\le N}\{y_i\}]$ and $f_{Y,K|\{y_i^*,c_i\}_1^N}$ is close to the density function of the original data.

Step 2.2 For each $\alpha = 0.01$–0.05 with increment 0.01,[2] let

$$a_\alpha = \max\left\{a_{basic}, \frac{\overline{y^*_{sub}}}{\bar{c}} - \sqrt{1/\alpha\left[\frac{\overline{y^{*2}_{sub}}}{\overline{c^2}} - \left(\frac{\overline{y^*_{sub}}}{\bar{c}}\right)^2\right]}\right\} \quad (2)$$

$$b_\alpha = \min\left\{b_{basic}, \frac{\overline{y^*_{sub}}}{\bar{c}} + \sqrt{1/\alpha\left[\frac{\overline{y^{*2}_{sub}}}{\overline{c^2}} - \left(\frac{\overline{y^*_{sub}}}{\bar{c}}\right)^2\right]}\right\} \quad (3)$$

where $\overline{y^*_{sub}}$ and $\overline{y^{*2}_{sub}}$ are the sample mean and the sample second moment of $\{y^*_{sub,j}\}$, respectively; $\bar{c}$ and $\overline{c^2}$ are the sample mean and the sample second moment of the noise C, respectively;

Step 2.3 For each pair-boundary (a_{basic}, b_{basic}), (a_α, b_α), $\alpha = 0.01, \ldots, 0.05$, determine the optimal upper order K for $f_{Y,K|\{y^*_{sub,j},c_j\}}$ and record $Cor(K_{opt})$, denoted by $Cor(K_{opt,basic})$ and $Cor(K_{opt,\alpha})$, $\alpha = 0.01, \ldots, 0.05$, respectively;

Step 2.4 Let $a = a_{\alpha_0}$ and $b = b_{\alpha_0}$, $\alpha_0 \in \{basic, 0.01, 0.02, \ldots, 0.05\}$ such that

$$Cor(K_{opt,\alpha_0}) = \max\{Cor(K_{opt,basic}), Cor(K_{opt,\alpha}), \alpha = 0.01, \ldots, 0.05\}.$$

Remarks. The logic used to support the standard procedure above is explained as follows.

(i) Given that $[a_{basic}, b_{basic}]$ is a superset of $[\min_{1\le i\le N}\{y_i\}, \max_{1\le i\le N}\{y_i\}]$ and $\{y_{sub,j}\} \subset \{y_i\}$, we have $a_{basic} \le \min\{y_{sub,j}\} \le \max\{y_{sub,j}\} \le b_{basic}$;

(ii) From Tchebichev inequality, we have

$$P(L_\alpha \le Y_{sub} \le U_\alpha) > 1 - \alpha \quad (4)$$

where $L_\alpha = E(Y_{sub}) - \sqrt{Var(Y_{sub})/\alpha}$ and $U_\alpha = E(Y_{sub}) + \sqrt{Var(Y_{sub})/\alpha}$. Ignoring the probability α, $\{y_{sub,j}\}$ will be bounded by

$$[\max\{a_{basic}, L_\alpha\}, \min\{b_{basic}, U_\alpha\}]. \quad (5)$$

By taking into account the information $Cor(K_{opt,basic})$ and $Cor(K_{opt,\alpha})$, and replacing the means by sample means in L_α and U_α, we should expect that $[a_{\alpha_0}, b_{\alpha_0}]$ is a reasonable domain replacing $[\min\{y_{sub,j}\}, \max\{y_{sub,j}\}]$ based on the information of $\{y^*_{sub,j}, c_j\}$. There might be other ways for determining the appropriate pair-boundary (a, b) for $f_{Y,K|\{y^*_{sub,j},c_j\}}$. We leave it as an open question.

2.3. *MaskDensity14 for categorical data*

Categorical data is the primary type of data considered in confidential microdata. With noise multiplied data, the mass function of the original data can be obtained by the method of moments. Assume that Y is a categorical variable taking values $1, 2, \ldots, M$. The mass function of Y can be obtained by solving simultaneous equations

$$E(Y^{*m})/E(C^m) = \sum_{i=1}^{M} i^m P(Y = m), \quad m = 1, \ldots, M,$$

subject to $E(Y^{*m})$ and $E(C^m)$, $m = 1, \ldots, M$, are available. However, the values of the theoretical means might not be available for the data user in practice. We adopt the method of moments in *MaskDensity14* by replacing $E(Y^{*m})$ and $E(C^m)$ with corresponding sample moments, $m = 1, \ldots, M$. With the high orders of sample moments involved, or sometimes the small size of the sample used, the method of moments might fail by giving negative values to the mass function of Y.

[2] To save time in running the program, we only consider these five different values of α in *MaskDensity14*.

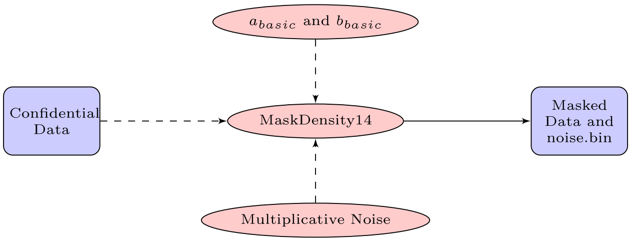
If the method of moments fails, *MaskDensity14* will use the sample-moment-based approximant density method to estimate the mass function of Y.

In theory, the technique of the approximant of a density function is for continuous univariate distributions. However, *MaskDensity14* will use the following way to cope with categorical data. (i) Mask the underlying categorical variable by a continuous noise. Thus, the masked data are no longer categorical data. (ii) Apply the method of sample-moment-based density approximant to the masked data and estimate the smoothed density function of the categorical variable based on the masked data. Obviously, the density approximant will have multiple centers at the levels of the categorical variable. (iii) Finally, use the existing K-means clustering R package to convert the smoothed density approximant to the mass function.

2.4. *Software architecture*

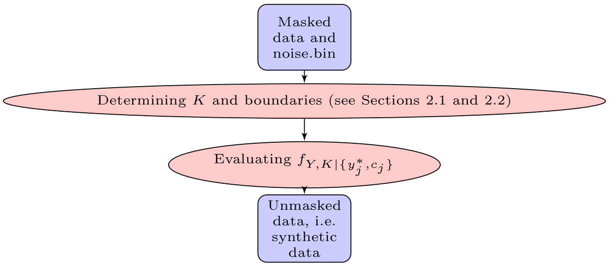
MaskDensity14 is built for two purposes. One purpose is to provide masked data. The other is for the data user generating synthetic data.

The flow chart for producing masked data is presented below:



By default the values of a_{basic} and b_{basic} are $\min\{y_i\}_1^N$ and $\max\{y_i\}_1^N$, respectively. By default, the sample of noise is from a mixture of two normal distributions with means randomly generated. The data agency has the option of providing the values of a_{basic} and b_{basic}, and the noise C used to mask the underlying original data. The binary file "noise.bin" contains a large sample from C, the values of a_{basic} and b_{basic}, and the information about whether the underlying data are numerical or categorical. **The sample noise in "noise.bin" are generated by sampling data from the noise sample used to mask the original data.** This process is run in background during the process of producing masked data. The size of the noise sample in "noise.bin" is ten times larger than that of the original data. In the process of determining the upper order K, the independent noise sample are drawn from this big noise sample. Since there is no link between the entries in the original dataset and the sample noise stored in "noise.bin", the individual entries of the original dataset cannot be identified simply knowing the masked data and the noise sample drawn from "noise.bin". The binary file is recognizable by *MaskDensity14* only.

The masked dataset and the binary file "noise.bin" can be sent out to the data user, and the original data are concealed from the data user. The data user can apply *MaskDensity14* to the files and obtain the synthetic data of the original data. A brief flow chart of the process is presented below:


Masked data and noise.bin
Determining K and boundaries (see Sections 2.1 and 2.2)
Evaluating $f_{Y,K|\{y_j^*,c_j\}}$
Unmasked data, i.e. synthetic data

2.5. *Software functionalities*

There are two main functions, *mask* and *unmask*, in *MaskDensity14*. Function *mask* is used to produce masked data and the binary file *noise.bin*. The outcome of *unmask* is a simulated sample drawn from the sample-moment-based density approximant of the original data.

3. Illustrative examples

Due to the focus of this paper, we only demonstrate the main functions of *MaskDensity14*. Simulation studies and applications will be investigated in another paper.

Example 2. Let $\{y_i\}_1^{10000}$ be the original sample data drawn from a random variable Y. The probability distribution of the random variable Y is a mixture of two normal distributions $\text{MixNorm}(m_1 = 30, m_2 = 50, s_1 = 4, s_2 = 2, p = 0.7)$, i.e. $Y = I_{(w=0)}Y_1 + I_{(w=1)}Y_2$, where I is an indicator function, $Y_1 \sim N(30, 4^2)$, $Y_2 \sim N(50, 2^2)$ and w is Bernoulli distributed with $P(w = 0) = 0.3$. Let $C \sim \text{MixNorm}(m_1 = 80, m_2 = 100, s_1 = 5, s_2 = 3, p = 0.4)$ be the multiplicative noise used to mask $\{y_i\}_1^{10000}$.

The R code used to simulate $\{y_i\}$ and $\{c_i\}$ is listed below:

```
set.seed(123)
n=10000
rmulti <- function(n, mean, sd, p)
{
  x <- rnorm(n)
  k<-length(mean)
  u <- sample(1:k, size=n, prob=p, replace=TRUE)
  for(i in 1:k)
    x[u==i]<-mean[i]+sd[i]*x[u==i]
  return(x)
}
y <- rmulti(n=10000, mean=c(30, 50), sd=c(4,2),
p=c(0.3, 0.7))
      # y is a sample drawn from Y.
noise<-rmulti(n=10000, mean=c(80, 100),
sd=c(5,3), p=c(0.6, 0.4))
      # noise is a sample drawn from C.
```

With the original data $\{y_i\}$ and the sample of noise, the data provider can use the following R code to generate a set of masked data of $\{y_i\}$:

Table 1
The summary of statistics given by “$y1\$unmaskedVariable$” and “$y$”.

Data	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
y	16.34	33.63	48.83	43.90	50.74	57.70
$y1\$unmaskedVariable$	15.19	35.12	48.48	44.00	50.99	57.80

```
library(MaskDensity14)
a1<-runif(1, min=min(y)-2,max=min(y))
b1<-runif(1, min=max(y), max=max(y)+2)
ymask<-mask(y, noisefile="noise.bin",
noise, a1=a1, b1=b1)
write(ymask$ystar, "ystar.dat")
```

The values $a1$ and $b1$ are the a_{basic} and b_{basic}, respectively, introduced in Section 2.2. In this example, we let $a1$ and $b1$ be the values randomly selected from $[\min(y) - 2, \min(y)]$ and $[\max(y), \max(y) + 2]$, respectively.

After running the above R code, two files “ystar.dat” and “noise.bin” are generated and ready for the data user. Saving the two files “ystar.dat” and “noise.bin” in the same R working directory, the data user can use the following code to obtain synthetic data of $\{y_i\}_1^{10000}$.

```
library(MaskDensity14)
ystar <- scan("ystar.dat")
y1 <- unmask(ystar, noisefile="noise.bin")
sample<-y1$unmaskedVariable
```

R output $y1\$unmaskedVariable$ gives the synthetic data of the original data $\{y_i\}$. The size of the synthetic data is the same as that of the original data. The data user can apply standard R code to $y1\$unmaskedVariable$ and obtain the output data analysis of the original data. For example, the output “plot(density(y1$unmaskedVariable))” gives the plot of the density approximate of the original data; the output “summary(y1$unmaskedVariable)” gives the estimate of the summary statistics of the original data. The summary statistics for original data and synthetic data are listed in Table 1.

Example 3 gives another example where the original data are categorical.

Example 3. Let Y be a categorical random variable with probability distribution $Bernoulli(0.1) + 1$. The multiplicative noise C is the absolute value of a random variable with distribution $N((a + b)/2, 1 + (a - b)^2/4)$, where $a = 170$ and $b = 80$. The following R code is used to obtain a sample $\{y_i\}$ from Y and a sample from C, respectively. Both of them have size 2000.

```
set.seed(124)
n<-2000
a<-170
b<-80
y<-rbinom(n, 1, 0.1)+1
noise<-(a+b)/2+ sqrt(1+(a-b)^2/4)*rnorm(n, 0,1)
noise[noise<0]<- - noise[noise<0]
```

Since $\{y_i\}$ only takes two values 1 and 2, the boundaries a_{basic} and b_{basic} are 0 and 3, respectively. The R code used to produce $\{y_i^*\}$ is as follows:

```
library(MaskDensity14)
ymask<-mask(factor(y), noisefile="noise.bin",
noise, a1=0,b1=3)
     # using factor(y) because y is
a categorical variable
write(ymask$ystar, "ystar.dat")
```

After running the above code, the files “ystar.dat” and “noise.bin” are ready for the data user.

The following code is used to obtain a set of synthetic data of $\{y_i\}$ with the same size and the estimated mass function of Y.

```
library(MaskDensity14)
ystar<-scan("ystar.dat")
y1 <- unmask(ystar, noisefile="noise.bin")
unmaskY<-y1$unmaskedVariable  # synthetic data
mass_function<-y1$prob  # estimated
mass function
```

The true mass function is $P(Y = 1) = 0.9$ and $P(Y = 2) = 0.1$. The proportions of $Y = 1$ and $Y = 2$ given by the sample of the original data are 0.9055 and 0.0945, respectively. The estimated mass function based on noise multiplied data is $P(Y = 1) = 0.90100844$ and $P(Y = 2) = 0.109802758$, which is very close to the true mass function.

4. Impact

MaskDensity14 is a promising software package for estimating the density function of univariate random variables based on noise multiplied data. The advantages of *MaskDensity14* can be summarized as follows:

No restriction on the type of distribution of the multiplicative noise

The data agency has a broad range of choices on the multiplicative noise for protecting the original data. Indeed, the noise sample used to mask the original data has a certain level impact on the accuracy of density approximant. This issue is beyond the focus of this paper.

More possibilities for data agencies to share data with the public

With *MaskDensity14*, the process of sharing data information for the public can be simplified. The data agency might have less responsibility for data analytics. It is particularly important for data agencies that have no necessary resource for doing advanced data analysis.

Standard statistical methods for data analysis

The complexity of the statistical inference based on noise multiplied data is reduced.

Lin and Wise [8] showed that the level of protection of the underlying original data can be maintained through an appropriate multiplicative noise even if the probability distribution of the noise is available to the public. To provide extra protection for the underlying original data, *MaskDensity14* encrypts the information of the multiplicative noise into a binary file. There may be other better methods to replace this manner.

A critical issue in *MaskDensity14* is about the decision on the upper order of moment K. It is possible to improve the accuracy of the density approximant further if there is a better way to determine the upper order K.

MaskDensity14 provides the data user with opportunities to explore the statistical information of the subset of the original dataset. The accuracy of the density approximant of a subset of data can be further improved if there is a better manner for determining the boundaries for the subset data based on the information provided by the data provider.

The method of the sample-moment-based density approximant is for univariate distributions. In real life, developing a computational statistical method for multivariate distributions is desirable. With *MaskDensity14* built, it will provide help in developing software for estimating joint density functions based on noise multiplied data.

The software developed in this paper provides data agencies and data users with a totally different process in data protection and confidential data analysis. Data agencies can mainly focus on the issue of data protection, and data users can generate synthetic data by themselves rather than receiving synthetic data from data providers.

5. Conclusions

MaskDensity14 is applied to univariate distributions. With *MaskDensity14* built, the method of sample-moment-based density approximant becomes feasible. The data user can produce (asymptotically) synthetic data of the original data by himself and carry out data analysis on the original data by using standard statistical methods without accessing the original data. Compared to existing methods, *MaskDensity14* provides a different manner in data protection and data statistical information recovery.

Acknowledgments

We thank our colleague Dr. Carole Birrell who provided proofreading on this paper. We also thank Mr. Luke Mazur, who offered help in developing *MaskDensity14*. Finally, we appreciate three anonymous reviewers' helpful comments and valuable suggestions that greatly improved the manuscript.

References

[1] Duncan GT, Lambert D. Disclosure limited data dissemination (with comment). J Amer Statist Ass 1986;81:1–28.

[2] Willenborg L, de Waal T. Elements of statistical disclosure control. Lecture notes in statistics, vol. 155. New York: Springer-Verlag; 2001.

[3] Oganian A. Multiplicative noise protocals. In: Domingo-Ferrer J, Magkos E, editors. Privacy in statistical databases 2010. LNCS, vol. 6344. Heidelerg: Springer; 2010. p. 107–17.

[4] Shlomo N. Releasing microdata: Disclosure risk estimation, data masking and assessing utility. J Priv Confident 2010;2:73–91.

[5] Kim JJ, Jeong DM. Truncated triangular distribution for multiplicative noise and domain estimation. JSM 2008;1023–30.

[6] Sinha B, Nayak TK, Zayatz L. Privacy protection and quantile estimation from noise multiplied data. Sankhya B 2011;73:297–315.

[7] Hwang JT. Multiplicative errors-in-variables models with applications to recent data released by the U.S. Department of Energy. J Amer Statist Assoc 1986;81:680–8.

[8] Lin Y-X, Wise P. Estimation of regression parameters from noise multiplied data. J Priv Confident 2012;4:55–88.

[9] Lin Y-X. Density approximant based on noise multiplied data. In: Domingo-Ferrer J, editor. Privacy in statistical databases 2014. LNCS, vol. 8744. Springer International Publishing Switzerland; 2014. p. 89–104.

[10] Provost SB. Moment-based density approximants. Math J 2005;9:728–56.