ORIGINAL RESEARCH

New privacy preserving clustering methods for secure multiparty computation

Hirofumi Miyajima[1], Noritaka Shigei[2], Hiromi Miyajima*[2], Yohtaro Miyanishi[3], Shinji Kitagami[4], Norio Shiratori[4]

[1] *Graduate School of Biomedical Science, Nagasaki University, Nagasaki, Japan*
[2] *Graduate School of Science and Engineering, Kagoshima University, Kagoshima, Japan*
[3] *(ISEM) Information Systems Engineering and Management, Inc., Tokyo, Japan*
[4] *Gobal Information and Telecommunication Institute, Waseda University, Tokyo, Japan*

Received: June 20, 2016 **Accepted:** August 17, 2016 **Online Published:** August 29, 2016
DOI: 10.5430/air.v6n1p27 **URL:** http://dx.doi.org/10.5430/air.v6n1p27

ABSTRACT

Many researches on privacy preserving data mining have been done. Privacy preserving data mining can be achieved in various ways by use of randomization techniques, cryptographic algorithms, anonymization methods, *etc.* Further, in order to increase the security of data mining, secure multiparty computation (SMC) has been introduced. Most of works in SMC are developed on applying the model of SMC on different data distributions such as vertically, horizontally and arbitrarily partitioned data. Another type of SMC with sharing data itself to each party attracts attention, and some studies have been done. A simple method to share data was proposed and it was applied to statistical computation. However, for SMC, complicated computation such as data mining has never been proposed. In the previous paper, we proposed a BP learning for SMC and showed the effectiveness of it. In this paper, we propose clustering methods such as k-means and NG for SMC and show the effectiveness in numerical simulation.

Key Words: Cloud computing, Secure multiparty computation, Clustering, K-means, Neural gas

1. INTRODUCTION

Many studies have been done with data mining.[1] Data mining is the process of extraction of information from large dataset. Nowadays, many organizations often utilize data mining, and they also use data obtained from multiple sources (servers) in order to gain more precise or variable knowledge. However, this case causes to anxiety about privacy and secure considerations. One of important interests in the related research community is privacy preserving data mining.[2,3] Privacy preserving data mining arose as a solution to this problem by allowing parties to cooperate in the extraction of knowledge or information without any of the cooperating parties having to reveal their individual data to each other. It is very important to maintain a good trade-off between privacy protection and knowledge discovery.[4,5] Privacy preserving data mining can be achieved in various ways by use of randomization techniques, cryptographic algorithms, anonymization methods, *etc.*[4,6–9] Specifically, data encryption seems to be effective. However, data encryption system requires both encryption and decryption for requests of client or user, so its applications are limited. Therefore, studies with distributed processing for secure data have been attracting attention.[6–9] As one of these studies, secure multiparty computation (SMC) has been introduced.[10–12] The purpose of SMC is to allow parties to carry out distributed computing tasks in secure way. Most of works in SMC have

***Correspondence:** Hiromi Miyajima; Email: miya@eee.kagoshima-u-ac.jp; Address: Graduate School of Science and Engineering, Kagoshima University, 1-21-40 Korimoto, Kagoshima 890-0065, Japan.

been developed on applying the model of SMC on different data distributions such as vertically, horizontally and arbitrarily partitioned data.[13–15] They are the methods that perform their processing for the subset of dataset.[16–18] Therefore, another type of SMC with sharing data itself to each party attracts attention, and some studies have been done.[19,20] A simple method to share data was proposed and they were applied to statistical computation.[12] However, complex calculation processing such as data mining has never proposed. In the previous paper, we proposed BP learning for SMC.[21] On the other hand, there are no studies on clustering using VQ (Vector Quantization). Clustering is the assignment of objective data (objects) into classes so that objects from the same class are more similar than objects of different classes. Clustering is a common technique for statistical data analysis, which is used in many fields such as machine learning, pattern recognition, and image analysis. K-means and Neural Gas (NG) methods are typical techniques of clustering and they are also known as hard- and soft-matching methods, respectively.[22] So for, k-means studies for conventional SMC have been done, but ones for SMC with shared data have never been presented yet. In this paper, we propose clustering methods for SMC and show the effectiveness of them in numerical simulations. In Section 2, we describe past related works on privacy preserving and the idea for sharing data securely. Further, we also introduce k-means and NG methods. K-means, which is a special case of NG, is introduced to explain the basic idea of our proposed method for clustering. In Section 3, we describe our proposed k-means and NG methods for SMC, where detailed algorithms are presented for client and parties. In Section 4, some simulation results are presented to demonstrate the effectiveness of our proposed methods.

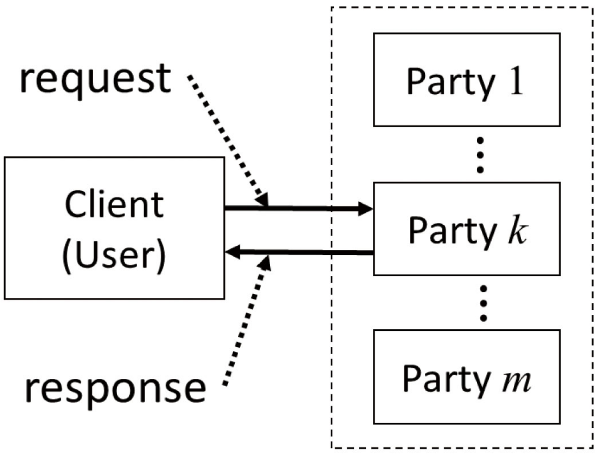


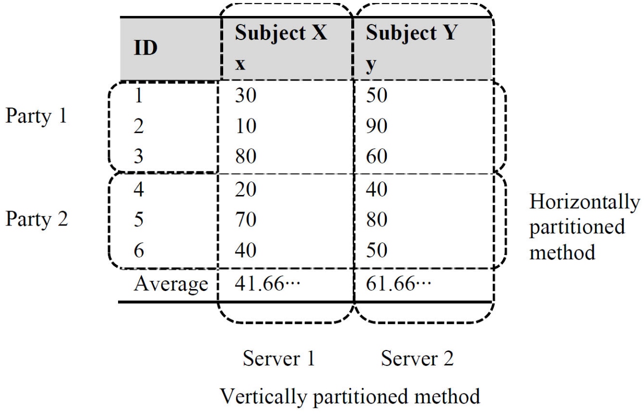
Figure 1. A configuration for SMC with a client and m parties

2. Preliminary

2.1 A system configuration for SMC and related works

The cloud computing system for SMC used in this paper is shown in Figure 1. The system consists of a client and m parties (servers). The client sends data to parties, and each party memorizes the received data. If the client wants to perform a task on the data memorized on the parties, each party performs computation on its own data and sends its computation result to the client. And then, the client performs computation on the received results. If the obtained result is not the final one, the pair of party-side and client-side computations is iterated until the client obtains the final result. The problems to realize SMC in this data processing flow are how to securely share the data among the parties and how to perform the task among the client and the parties.

Table 1. Concept of conventional horizontally and vertically partitioned method



ID	Subject X x	Subject Y y
1	30	50
2	10	90
3	80	60
4	20	40
5	70	80
6	40	50
Average	41.66···	61.66···

Party 1
Party 2
Horizontally partitioned method
Server 1
Server 2
Vertically partitioned method

Three types of methods for partitioning data to be securely shared are known.[16–19] They are horizontal, vertical and arbitrary partitioning methods. In the following, the conventional methods are explained by using a data example of students' marks shown in Table 1. In Table 1, x and y are original data (marks) and ID is the identifier of students. The assumed task is to calculate the average of the data. The first method, the horizontal partitioning method, assigns the horizontally partitioned data to the parties as follows:

Party 1: data for ID=1, 2, 3.

Party 2: data for ID=4, 5, 6.

In the method, party 1 computes two averages for subjects X and Y as $(30 + 10 + 80)/3$ and $(50 + 90 + 60)/3$, respectively. Likewise, party 2 computes two averages for subjects X and Y as $(20 + 70 + 40)/3$ and $(40 + 80 + 50)/3$, respectively. The parties 1 and 2 send the calculated averages to the client and the client obtains the averages of subjects X and Y as $41.66\cdots$ and $61.66\cdots$, respectively. Since each party cannot know half of the dataset, the method preserves privacy. In fact, instead of raw data, encrypted or randomized data are used. The second method, the vertical partitioning method, assigns the vertically partitioned data to the parties as follows:

Party 1: data for subject X.

Party 2: data for subject Y.

In the method, each party calculates the average for either of subject X and Y. Since each party can know data for only one of subject X or Y, the method also preserves privacy. The third method, the arbitrary partitioning method, horizontally and vertically splits the dataset into multiple parts, and the method assigns the split parts to the parties. For example,

Party 1: data of ID=1, 2, 3 for subject X and ID=4, 5, 6 for subject Y.

Party 2: data of ID=4, 5, 6 for subject X and ID=1, 2, 3 for subject Y.

In the method, each party calculates two averages of subjects X and Y for its own data, and sends the calculated averages to the client. The client calculates two averages of subjects X and Y by using the averages calculated by the parties. Since each party knows partial data of subjects X and Y, the method also preserves privacy.

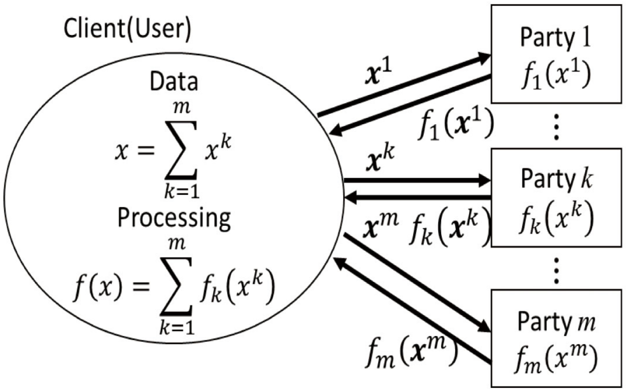


Figure 2. The idea of the proposed method for SMC

For any of the above mentioned methods, if the number of clients is fewer, that is, the size of a partitioned data is larger, a client may more easily guess the feature of all the data from its own subset of data. Therefore, the methods need a large number of parties in order to keep privacy and security. On the other hand, the proposed method to share data itself seems to keep them by a small number of parties.

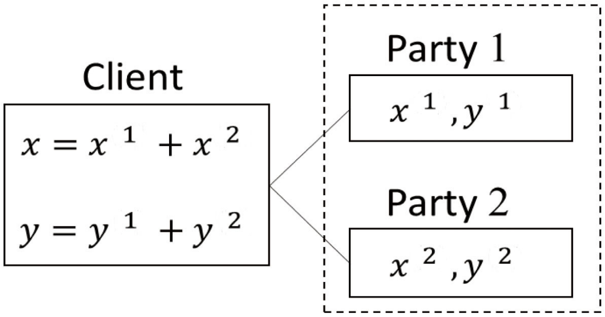


Figure 3. The representation of additional form for SMC

2.2 Data representation for securely sharing data

This subsection explains the data representation for securely sharing data among parties. The proposed methods are based on this representation. Let $\mathbb{R}$ be the set of real numbers. Let $Z_i = \{1, 2, \cdots, i\}$ and $Z_i^* = \{0, 1, \cdots, i\}$ for the positive integer i. Let us consider to share a real number x among m parties and to calculate a function $f(x)$. As shown in Figure 2, a real number x is divided into m pieces of data such as $x = x^1 + \cdots + x^m$. The k-th data x^k for $k{\in}Z_m$ is sent to the k-th party. The k-th party calculates $f_k(x^k)$ and sends its result to the client. The client tries to obtain $f(x)$ by summarizing $f_k(x^k)$. If the final result $f(x)$ is not obtained, similar processing has to be repeated. In order to successfully obtain the final result $f(x)$, $f(x)$ and $f_k(x^k)$ have to have some relation, for example, expressed by the following equation:

$$f(x) = \sum_{k=1}^{m} f_k(x^k) \tag{1}$$

where the equation depends on the data representation for sharing data. By using Figure 3 and Table 2, let us explain the data representation for SMC and the computation in detail.[19,20] Let x and y be positive integers and $m = 2$ be the number of parties. In the data representation, each of x and y is shared by using two real numbers as follows: $x = x^1 + x^2$ and $y = y^1 + y^2$, where x^1, x^2, y^1 and y^2 are real numbers. In the example shown in Table 2, $x^1 = x(r_1/10.0)$, $x^2 = x(1.0 - r_1/10.0)$, $y^1 = y(r_2/10.0)$ and $y^2 = y(1.0 - r_2/10.0)$, where r_1 and r_2 are real random numbers such that $-8.0{\leq}r_1, r_2{\leq}8.0$, $r_1{\neq}0.0$ and $r_2{\neq}0.0$. For example, for ID=1, x^1 and x^2 are computed as $x^1 = 30(2.0/10.0) = 6.0$ and $x^2 = 30(1 - 2.0/10.0) = 24.0$. Note that, since each of x^1, x^2, y^1 and y^2 is randomized data, each party cannot know original data x and y.

Table 2. An Example to explain data representation for the proposed method

ID	Subject X	Subject Y	Additional form					
	x	y	r_1	r_2	x		y	
					x^1	x^2	y^1	y^2
1	30	50	2	-3	6	24	-15	65
2	10	90	-4	-2	-4	14	-18	108
3	80	60	3	4	24	56	24	36
4	20	40	4	2	8	12	8	32
5	70	80	-2	3	-14	84	24	56
6	40	50	-3	-4	-12	52	-20	70
Average	41.66···	61.66···			1.33	40.33···	0.5	61.66···

2.3 Clustering by k-means and NG methods

Vector quantization (VQ) techniques encode a data space, *e.g.*, a subspace $X \subset \mathbb{R}^N$, by utilizing only a finite set $W = \{\boldsymbol{w}^i | i \in Z_r\}$ of reference vectors (also called weight vectors), which N and r are positive integers. That is, the set X is approximated by the set W. In this paper, $X = \{\boldsymbol{x}^i | i \in Z_n\}$ is assumed. In the following, two types of VQ methods are introduced: k-means method and Neural Gas (NG) method.[22] The first method, k-means method, is the most fundamental VQ method. The winner vector $\boldsymbol{w}^{i_{min}(\boldsymbol{x})}$ is defined as follows:

$$i_{min}(\boldsymbol{x}) = \arg\min_{i \in Z_r} ||\boldsymbol{x} - \boldsymbol{w}^i|| \tag{2}$$

The finite set W divides X into r subsets as follows:

$$X = \cup_{i=1}^{r} X_i \tag{3}$$

$$X_i = \{\boldsymbol{x} \in \mathbb{R}^N | ||\boldsymbol{x} - \boldsymbol{w}^i|| \leq ||\boldsymbol{x} - \boldsymbol{w}^j|| \; for \; j \in Z_r\} \tag{4}$$

The evaluation function for the partition is defined as follows:

$$E = \frac{1}{n} \sum_{i=1}^{r} \sum_{\boldsymbol{x} \in X_i} ||\boldsymbol{x} - \boldsymbol{w}^{i_{min}(\boldsymbol{x})}||^2 \tag{5}$$

Each parameter $\boldsymbol{w}^i$ is updated based on the steepest descent method as follows:[23]

$$\triangle \boldsymbol{w}^i = \varepsilon(t) \delta_{ij(\boldsymbol{x}(t))} (\boldsymbol{x}(t) - \boldsymbol{w}^i) \tag{6}$$

where t is the step, $\varepsilon(t)$ is learning coefficient and δ_{ij} is the Kronecker delta. Input vector $\boldsymbol{x}(t)$ is the vector randomly selected from X at step t. Figure 4 shows the algorithm of k-means method.[22,23]

The second method, NG method, is known as a novel VQ method. The feature of NG method is that all the weights are updated based on the rank of distance between input and reference vectors.

Input: Input vector X = $\{x^i | \, i \in \, Z_n\}$

Output: Reference vector W = $\{w^j | \, j \in \, Z_r\}$

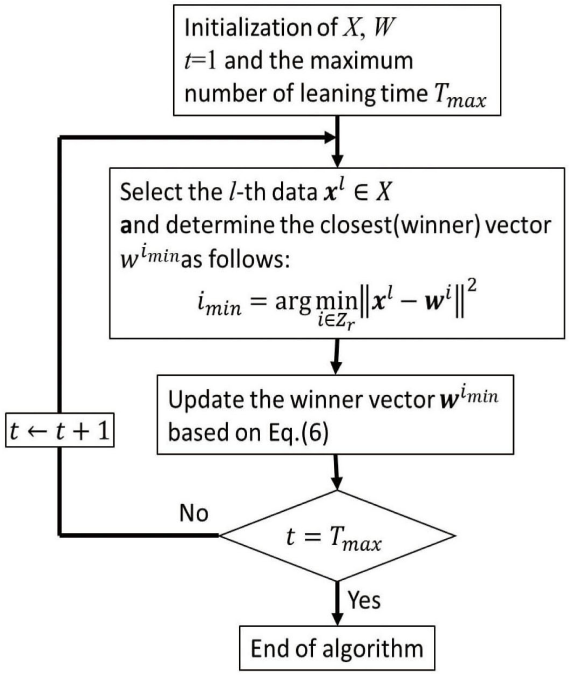

Initialization of X, W
t=1 and the maximum
number of leaning time T_{max}
Select the l-th data $\boldsymbol{x}^l \in X$
and determine the closest(winner) vector
$w^{i_{min}}$as follows:
$i_{min} = \arg\min_{i \in Z_r} \|\boldsymbol{x}^l - \boldsymbol{w}^i\|^2$
Update the winner vector $\boldsymbol{w}^{i_{min}}$
based on Eq.(6)
$t \leftarrow t + 1$
No
$t = T_{max}$
Yes
End of algorithm

Figure 4. The flowchart of k-means algorithm

The updated step for a reference vector is computed by

$$\triangle \boldsymbol{w}^i = \varepsilon(t) \exp\left(-k_i/\lambda\right) (\boldsymbol{x}(t) - \boldsymbol{w}^i) \tag{7}$$

where k_i is the rank of reference vector $\boldsymbol{w}^i$ for the input vector $\boldsymbol{x}$. The rank of $\boldsymbol{w}^i$ is k_i if the distance between $\boldsymbol{x}$ and $\boldsymbol{w}^i$ for $i \in Z_r$ is the $(k_i - 1)$-th small number. The algorithm based on Equation (7) is shown as follows:[22]

Algorithm

Input vector: $X = \{\boldsymbol{x}^i | i \in Z_n\}$

Number of iterations: T_{max}
Output Reference vector: $W = \{\boldsymbol{w}^j | j \in Z_r\}$

Step 1

(1.1) Initialize W with random numbers.

(1.2) Set $t = 1$.

Step 2

(2.1) Select an input vector $\boldsymbol{x}^l$ for $l \in Z_n$.

(2.2) Calculate the distance between $\boldsymbol{x}^l$ and w^j for $j \in Z_r$ and determine the rank k_j of each reference vector $\boldsymbol{w}^j$.

Step 3

Update all the weights according to Equation (7).

Step 4

If $t = T_{max}$, then the algorithm terminates. Otherwise go to Step 2 with $t \leftarrow t + 1$.

Table 3. Proposed k-means method for SMC

	Client	k-th Party
Initial condition	Each data of $\boldsymbol{w}$ is selected randomly, and send $\left(\boldsymbol{w}^i\right)^k$ for $i \in Z_r$ to each party. Set $t = 1$ Learning coefficient ε_{int} and ε_{fin} are set.	The dataset $\{(\boldsymbol{x}^l)^k \mid l \in Z_n\}$ is memorized.
Step 1	A number q is selected from Z_n randomly and send it to each party	The dataset $\left\{\left(\boldsymbol{w}^i\right)^k \middle| i \in Z_r\right\}$ is memorized.
Step 2		Compute $\Delta^k = (\boldsymbol{x}^q)^k - \left(\boldsymbol{w}^i\right)^k$ for $i \in Z_r$ and send them to Client.
Step 3	Find $i_{min}(\boldsymbol{x}^q) \in Z_r$ as $\arg \min_{i \in Z_r} \left\|\sum_{k=1}^m \Delta^k\right\|^2$ and send it to each party. Compute $\varepsilon(t) = \varepsilon_{int}\left(\frac{\varepsilon_{fin}}{\varepsilon_{int}}\right)^{\frac{t}{T_{max}}}$ and send it to each party.	
Step 4		Update the i_{min}-th reference vector as follows: $\left(\boldsymbol{w}^{i_{min}}\right)^k(t+1) = \left(\boldsymbol{w}^{i_{min}}\right)^k(t) + \Delta\left(\boldsymbol{w}^{i_{min}}\right)^k$ where $\Delta\left(\boldsymbol{w}^{i_{min}}\right)^k = \varepsilon\left[(\boldsymbol{x}^q)^k - \left(\boldsymbol{w}^{i_{min}}\right)^k\right]$
Step 5	If $t = T_{max}$ then the algorithm terminates otherwise go to Step 1 with $t \leftarrow t + 1$.	

3. Privacy preserving k-means and NG

3.1 Proposed k-means method for SMC

A system consisting of a client and m parties is assumed (see Figure 1). When learning, parties share the input and reference vectors by using additional form. Each party updates the shared reference vectors and sends the computation results to the client. The client obtains new reference vectors by summarizing results of m parties. The representation form for sharing input and reference vectors is given as follows:

$$\boldsymbol{x}^l = \left(x_1^l, \cdots, x_i^l, \cdots, x_N^l\right) \tag{8}$$

for $l \in Z_n$,

$$x_i^l = \sum_{k=1}^m (x_i^l)^k \tag{9}$$

for $i \in Z_N$ and

$$\boldsymbol{w}^j = \left(w_1^j, \cdots, x_p^j, \cdots, w_N^j\right) \tag{10}$$

for $j \in Z_r$

$$w_p^j = \sum_{k=1}^m \left(w_p^j\right)^k \tag{11}$$

for $p \in Z_N$, Equation (6) is rewritten using the additional form of Equation (11) as follows:

$$\begin{aligned} \Delta\left(w_p^j\right)^k &= \frac{\partial E}{\partial (w_p^j)^k} = \frac{\partial E}{\partial w_p^j} \times \frac{\partial w_p^j}{\partial (w_p^j)^k} \\ &= \frac{\partial E}{\partial w_p^j} = \Delta w_p^j \end{aligned} \tag{12}$$

Because $\frac{\partial w_p^j}{\partial (w_p^j)^k} = 1$.

Therefore, the following equation holds:

$$\left(w_p^j\right)^k (t+1) = \left(w_p^j\right)^k (t) + K \triangle \left(w_p^j\right)^k. \tag{13}$$

Equation (13) means that each party updates in the same method as the conventional k-means method. Note that it holds for the additional form of Equation (11).

The detailed algorithm of k-means method is shown in Table 3. The general flow of Table 3 is as follows:

In Step 1, client selects a number $q{\in}Z_n$ randomly and it is sent to each party. For any $k{\in}Z_m$, the k-th component for $\boldsymbol{w}^i{\in}W$ is stored in the k-th party. In Step 2, the k-th party computes the k-th element of the difference between $\boldsymbol{x}^q$ and $\boldsymbol{w}^i$ for $i{\in}Z_n$ and their results are sent to the client. In Step 3, the client computes the distance $||\boldsymbol{x}^q - \boldsymbol{w}^i||$ between q-th data $\boldsymbol{x}^q$ and i-th reference vector $\boldsymbol{w}^i$ and determines the index i_{min} of reference vector withthe minimum distance. This computation is verified as follows:

$$\sum_{k=1}^{m} \triangle^k = \sum_{k=1}^{m}((\boldsymbol{x}^q)^k - (\boldsymbol{w}^i)^k) = \sum_{k=1}^{m}(\boldsymbol{x}^q)^k - \sum_{k=1}^{m}(\boldsymbol{w}^i)^k$$
$$= \boldsymbol{x}^q - \boldsymbol{w}^i \tag{14}$$

The index i_{min} is sent to each party. At the same step, learning coefficient is updated and sent to each party. In Step 4, the elements of reference vectors for each party are updated based on Equation (13). As a result, each party shares each reference vector $\boldsymbol{w}^i$ for $i{\in}Z_r$ at learning step t.

3.2 Proposed NG method for SMC

Likewise, we can get the result of NG for SMC.

Given an input vector $\boldsymbol{x}$, the rank $e_i(\boldsymbol{x}, \boldsymbol{w}^i)$ of the reference vector $\boldsymbol{w}^i$ for $i{\in}Z_r$ is determined, being the reference vector for which there are $e_i(\boldsymbol{x}, \boldsymbol{w}^i)$ pieces of vectors $\boldsymbol{w}^j$ such that

$$||\boldsymbol{x} - \boldsymbol{w}^j|| < ||\boldsymbol{x} - \boldsymbol{w}^i|| \tag{15}$$

where $j{\in}Z^*_{r-1}$.

Then updating formula is given by

$$\triangle(w^i_p)^k = \varepsilon{\cdot}h_\lambda\left(e_i(\boldsymbol{x}, \boldsymbol{w}^i)\right){\cdot}(x^k_p - (w^i_p)^k) \tag{16}$$

$$h_\lambda(e_i(\boldsymbol{x}, \boldsymbol{w}^i)) = \exp\left(-e_i(\boldsymbol{x}, \boldsymbol{w}^i)/\lambda\right) \tag{17}$$

where $\varepsilon{\in}[0,1]$ and $\lambda > 0$.

Table 4. Proposed NG method for SMC

	Client	***k*-th Party**
Initial condition	Each data of $\boldsymbol{w}$ is selected randomly, and send $\left(\boldsymbol{w}^i\right)^k$ for $i \in Z_r$ to each party. Set $t = 1$. Learning coefficient ε_{int} and ε_{fin} are set.	The dataset $\{(\boldsymbol{x}^l)^k \mid l \in Z_n\}$ is memorized.
Step 1	A number q is selected from Z_n randomly and send it to each party	The dataset $\left\{\left(\boldsymbol{w}^i\right)^k \middle| i \in Z_r\right\}$ is memorized.
Step 2		Compute $\Delta^k = (\boldsymbol{x}^q)^k - \left(\boldsymbol{w}^i\right)^k$ for $i \in Z_r$ and send them to Client.
Step 3	Compute $\mathrm{e}_i\left(\boldsymbol{x}^q, \boldsymbol{w}^i\right)$ for $i \in Z_r$ using Equation (15) as $\left\|\boldsymbol{x}^q - \boldsymbol{\omega}^i\right\| = \left\|\sum_{k=1}^m \Delta^k\right\|$ and $\varepsilon(t) = \varepsilon_{int}\left(\frac{\varepsilon_{fin}}{\varepsilon_{int}}\right)^{\frac{t}{T_{max}}}$. Send them to each party.	
Step 4		Update all the reference vectors as follows: $\left(\boldsymbol{w}^{i_{min}}\right)^k(t+1) = \left(\boldsymbol{w}^{i_{min}}\right)^k(t) + \Delta\left(\boldsymbol{w}^{i_{min}}\right)^k$ where $\Delta\left(\boldsymbol{w}^{i_{min}}\right)^k = \varepsilon h_\lambda\left[e_i\left(\boldsymbol{x}^q, \boldsymbol{w}^i\right)\right]\left[(\boldsymbol{x}^q)^k - \left(\boldsymbol{w}^i\right)^k\right]$ and $h_\lambda[e_i(\boldsymbol{x}, \boldsymbol{w})] = exp[-e_i(\boldsymbol{x}, \boldsymbol{w}) \;/\; \lambda]$
Step 5	If $t = T_{max}$ then the algorithm terminates otherwise go to Step 1 with $t \leftarrow t + 1$.	

If $\lambda{\to}0$, Equation (17) becomes equivalent to the k-means method. The proposed NG algorithm for SMC is shown in Table 4. In k-means method, (winner) reference vector $\boldsymbol{w}^{i_{min}}$ is only updated. In NG method, all the reference vectors are updated using the rank based on the distance between input vector and each reference vector. That is, in Step 3 of Table 4, the rank $e^i(\boldsymbol{x}^q, \boldsymbol{w}^i)$ of $\boldsymbol{w}^i$ for $i{\in}Z_r$ based on the distance $||\boldsymbol{x}^q - \boldsymbol{w}^i||$ between input vector $\boldsymbol{x}^q$ and reference vector $\boldsymbol{w}^i$ is computed and all the ranks are sent to each party. In Step 4, each element of reference vectors is updated based on them.

4. Numerical simulations

In this chapter, we present two simulations involving artificial and real-world data sets.

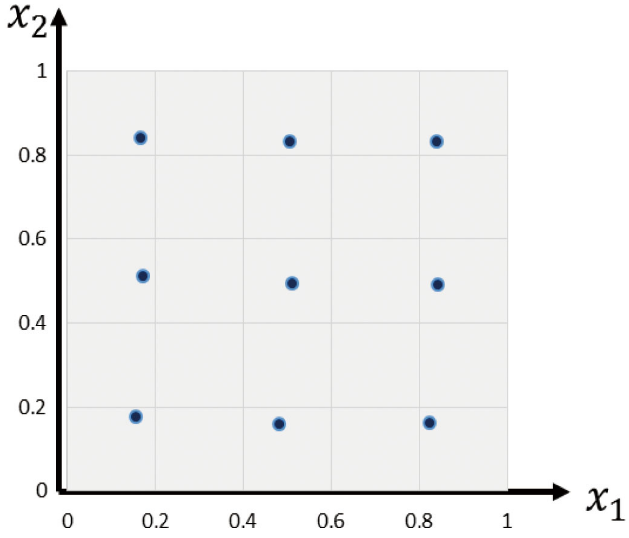


Figure 5. A configuration of nine reference vectors for proposed NG with $m = 10$ after learning

4.1 Clustering problems for artificial data

In the first simulation of artificial data set, we approximate the space $[0, 1]^2$ as shown in Figure 5 by nine and thirty-six reference vectors, *i.e.*, $r = 9$ and $r = 36$. That is, the problem is how the space $[0, 1]^2$ is approximated by nine and thirty-six points. The initial values of W are set randomly and $T_{max} = 50,000$, $\varepsilon_{int} = 0.1$ and $\varepsilon_{fin} = 0.01$. In the first case, we perform the comparison of the learning speed between conventional and proposed methods. Figure 6 shows the graph of learning speed of proposed and conventional methods, where the vertical and horizontal axes are evaluation value of Equation (5) and learning time, respectively. It means that learning speed for them is almost the same. Further, Figure 5 shows a configuration for nine reference vectors after learning for the proposed method of NG with $m = 10$. We can see how the space $[0, 1]^2$ is approximated by nine points. In the second case, we show the comparison of approximation accuracy between conventional and proposed methods. Table 5 shows the result of approximation accuracy using MSE of Equation (5). The result is the average for twenty trials and shows that approximation accuracy of conventional and proposed methods is almost the same.

Table 5. The result for clustering problem to approximate the space $[0, 1]^2$

		r = 9	**r = 36**
Conventional	*k*-means	0.018	0.004
	NG	0.018	0.004
Proposed	(*m* = 3)	0.018	0.004
k-means	(*m* = 10)	0.018	0.005
Proposed NG	(*m* = 3)	0.019	0.004
	(*m* = 10)	0.019	0.004

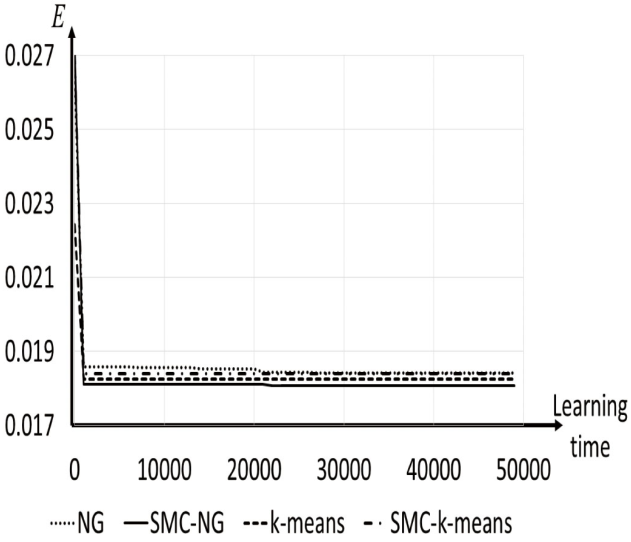
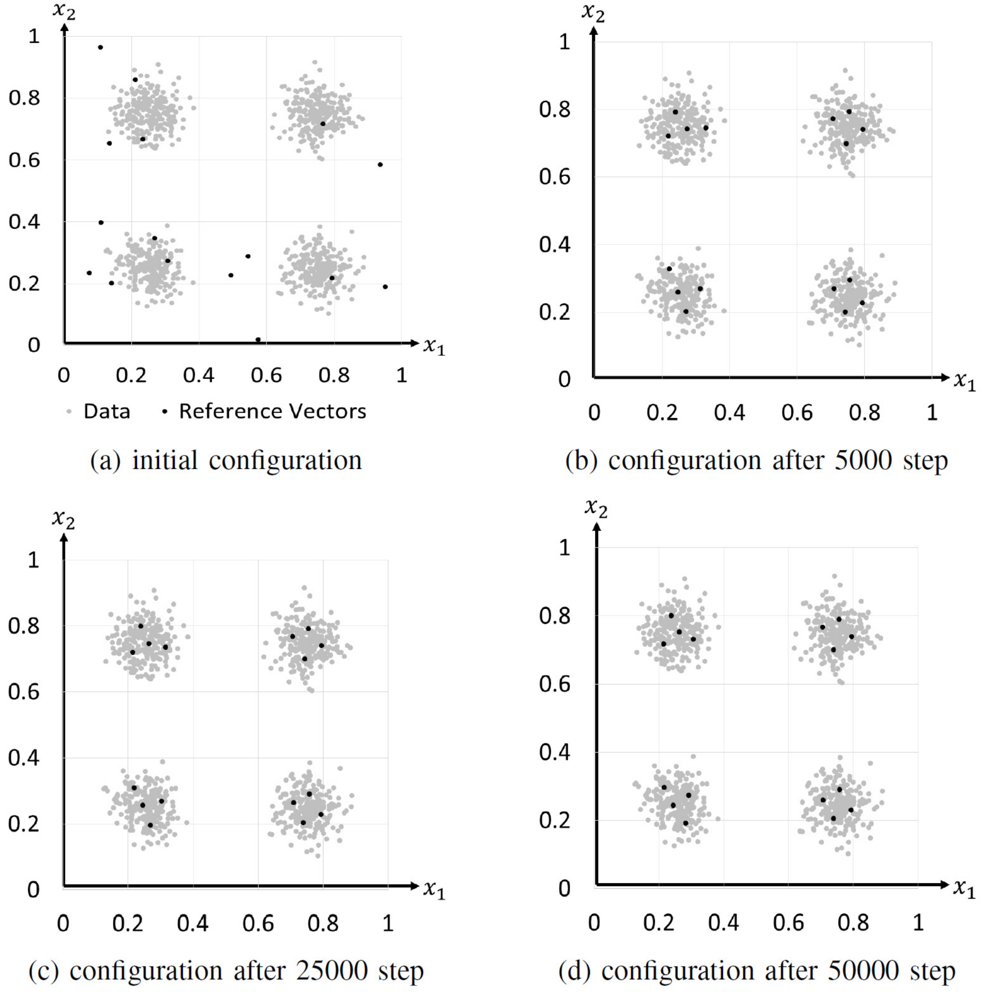


Figure 6. Comparison of the learning speed between conventional and proposed methods in the case of $m = 10$

Table 6. The result for clustering problem to approximate four separated clusters

		r = 4	**r = 16**
Conventional	*k*-means	0.014	0.002
	NG	0.013	0.002
Proposed	(*m* = 3)	0.014	0.002
k-means	(*m* = 10)	0.014	0.002
Proposed NG	(*m* = 3)	0.011	0.002
	(*m* = 10)	0.010	0.002

In the second simulation of artificial data set, we approximate the data distribution consisting of four separated clusters by four and sixteen points. On each cluster, the density of data points is normal N(0.25, 0.05) and N(0.75, 0.05) for x_1 and x_2 distribution (see Figure 7). Each value of W is set randomly and let $T_{max} = 50,000$, $\varepsilon_{int} = 0.1$ and ε_{fin}, respectively. Table 6 shows the result of approximation accuracy using MSE for Equation (5). The result is the average value from twenty trials and shows that the ability for conventional and proposed methods is almost the same. Further, Figure 7 shows a result for $m = 10$ that sixteen reference vectors are depicted as points. Initial values of W are chosen randomly, which is shown in Figure 7(a). We also show configurations after 5,000, 25,000 and 50,000 steps. At the end of the procedure, reference vectors are separated into four clusters, *i.e.*, each cluster is approximated by four reference vectors.



(a) initial configuration

(b) configuration after 5000 step

(c) configuration after 25000 step

(d) configuration after 50000 step

Figure 7. Configurations in learning steps for proposed NG with $m = 10$

Table 7. The dataset for real-world data

	Iris	Wine	Sonar	BCW	Spam	Skin
# data	150	178	208	683	4,601	245,057
# input	4	13	60	9	57	3
# class	3	3	2	2	2	2

4.2 Clustering problems for real-world data

Six real-world data sets including Iris, Wine, BCW, Spam (Spambase) and Skin (Skin Segmentation) coming from UCI machine learning repository have been considered in this simulation as shown in Table 7,[24] where #data, #input and #class mean the numbers of data, input variables and classes, respectively. As the initial condition, initial values of W are selected randomly from $[0, 1]$ and the maximum number of learning times are $50,000\times$#data for Iris, Wine, Sonar and BCW, $20,000\times$#data for Spam and $2,000\times$#data for Skin, respectively. Let $\varepsilon_{int} = 0.1$ and $\varepsilon_{fin} = 0.01$. The problem is how each dataset is approximately by reference vectors. In this simulation, we perform two cases: the first case is $r =$ #class that each class is approximated by one reference vector and the second case is $r = 4\times\#$class that each class is approximated by four reference vectors. Table 8 and Table 9 show the results of the misclassification rates for two cases, where the misclassification rate means the ratio of the misclassification data to all data, and each result is the average value from twenty trials for Iris, Wine, Sonar and BCW and ten trials for Spam and Skin. Both results show that approximation accuracy for conventional and the proposed methods is almost the same.

Table 8. The result of real-world data for proposed methods with $r = \#class$

		Iris	**Wine**	**Sonar**	**BCW**	**Spam**	**Skin**
Conventional	*k*-means	11.3	19.7	45.6	3.9	35.5	44.9
	NG	6.7	10.6	44.7	4.0	26.8	45.7
Proposed	(*m* = 3)	12.8	15.1	46.6	3.9	33.6	44.9
k-means	(*m* = 10)	15.4	12.1	45.4	4.0	33.6	44.9
Proposed NG	(*m* = 3)	6.8	10.8	44.7	10.1	26.5	45.7
	(*m* = 10)	5.6	11.2	44.7	10.1	26.7	45.7

Table 9. The result of real-world data for proposed methods with $r = \#class\times 4$

		Iris	**Wine**	**Sonar**	**BCW**	**Spam**	**Skin**
Conventional	*k*-means	8.5	9.6	41.5	3.1	30.2	6.5
	NG	7.1	10.9	44.1	3.0	23.3	9.4
Proposed	(*m* = 3)	9.1	8.5	40.8	3.0	31.6	5.8
k-means	(*m* = 10)	9.3	9.8	41.6	2.9	23.9	5.6
Proposed NG	(*m* = 3)	6.6	9.1	44.4	2.9	22.6	9.9
	(*m* = 10)	6.8	9.2	43.6	3.1	22.6	10.9

5. Conclusions

The SMC is one of secure data sharing and computing methods and it can perform privacy preserving data mining. So far, most of works in SMC are developed on applying the model of SMC on different data distributions. On the other hand, another type of SMC with sharing data itself to each party attracts attention, and some studies have been done. In the previous paper, we proposed BP learning for SMC. The idea of our study is to perform privacy preserving data mining as "shared data + parallel algorithm". That is, it is to find the representation of shared data and to construct parallel algorithm. In this paper, we proposed clustering methods for SMC with the above mentioned data form and proved the validity of them according to the idea.

Further, we demonstrated the performance of proposed methods by numerical simulations. In the future works, we will apply the proposed method to other problems on data mining such as kernel k-means and deep learning.

References

[1] Frawley WJ, Piatetsky-Shapiro G, Matheus CJ. Knowledge Discovery in Databases: An Overview. AI Magazine. 1992; 13(3): 57-70.

[2] Aggarwal CC, Yu PS. Privacy-Preserving Data Mining: Models and Algorithms. ISBN 978-0-387-70991-8, Springer-Verlag; 2009.

[3] Shamir A. How to share a secret, Comm. ACM. 1979; 22(11): 612-3. http://dx.doi.org/10.1145/359168.359176

[4] Beimel A. Secret-sharing schemes: a survey, in Proc. of the Third international conference on Coding and cryptology (IWCC 11); 2011. http://dx.doi.org/10.1007/978-3-642-20901-7_2

[5] Subashini S, Kavitha V. A survey on security issues in service delivery models of cloud computing. J. Network and Computer Applications. 2011; 34(11): 1-11. http://dx.doi.org/10.1016/j.jnca.2010.07.006

[6] Gentry C. Fully Homomorphic Encryption Using Ideal Lattices. STOC2009. 2009: 169-78.

[7] HElib. An Implementation of homomorphic encryption. https://github.com/shaih/HElib

[8] Rajesh N, Sujatha K, Arul Lawrence A. Survey on Privacy Preserving Data Mining Techniques using Recent Algorithms. International Journal of Computer Applications. 2016; 133(7): 30-3. http://dx.doi.org/10.5120/ijca2016907917

[9] Rathna SS, Karthikeyan T. Survey on Recent Algorithms for Privacy Preserving Data mining. International Journal of Computer Science and Information Technologies. 2015; 6 (2): 1835-40.

[10] Canetti R, Feige U, Goldreich O, et al. Adaptively secure multi-party computation. Twenty-eighth Acm Symposium on Theory of Computing. 2001: 639-48.

[11] Cramer R, Damgård I, Maurer U. General secure multi-party computation from any linear secret-sharing scheme. EUROCRYPT. 2000: 331-9. http://dx.doi.org/10.1007/3-540-45539-6_22

[12] Ben-David A. Fair play MP: a system for secure multi-party computation. ACM CCS '08; 2008.

[13] Upmanyu M, Namboodiri AM, Srinathan K, et al. Efficient Privacy Preserving K-Means Clustering. LNCS. 2010; 6122: 154-66. http://dx.doi.org/10.1007/978-3-642-13601-6_17

[14] Doganay MC, Pedersen TB, Saygın Y, et al. Distributed Privacy Preserving k-Means Clustering with Additive Secret Sharing. Proc. of the 2008 Int. Workshop on privacy and anonymity in information society. ACM. 2008: 3-11. http://dx.doi.org/10.1145/1379287.1379291

[15] Samet S, Miri A. Privacy Preserving k-means Clustering in Multi-Party Environment. SECRYPT 2007 - International Conference on Security and Cryptography. 2007: 381-5.

[16] Yuan J, Yu S. Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing. IEEE Trans. On Parallel and Distributed Systems. 2013; 25(1): 212-21. http://dx.doi.org/10.1109/TPDS.2013.18

[17] Schlitter N. A Protocol for Privacy Preserving Neural Network Learning on Horizontal Partitioned Data. Privacy Statistics in Databases (PSD); 2008.

[18] Chen T, Zhong S. Privacy-Preserving Back Propagation Neural Network Learning. IEEE Trans. on NN. 2009; 20(10): 1554-64. PMid:19709975. http://dx.doi.org/10.1109/TNN.2009.2026902

[19] Chida K. A Lightweight Three-Party Secure Function Evaluation with Error Detection and Its Experimental Result. IPSJ Journal. 2011; 52 (9): 2674-85 (in Japanese).

[20] Miyanishi Y, Kanaoka A, Sato F, et al. New Methods to Ensure Security to Increase User's Sense of Safety in Cloud Services. Proc. of The 14th IEEE Int. Conference on Scalable Computing and Communications. 2014: 859-65. http://dx.doi.org/10.1109/uic-atc-scalcom.2014.37

[21] Miyajima H, Shigei N, Miyajimay H, et al. A Proposal of Back Propagation Learning for Secure Multi-Party Computation Methods, Proc. of the Int. Multi Conference of Engineers and Computer Scientists. 2016: 381-6.

[22] Martinetz TM, Berkovich SG, Schulten KJ. Neural Gas Network for Vector Quantization and its Application to Time-series Prediction. IEEE Trans. Neural Network. 1993; 4(4): 558-69. PMid:18267757. http://dx.doi.org/10.1109/72.238311

[23] Miyajima H, Shigei N, Miyajima H. Performance Comparison of Hybrid Electromagnetism-like Mechanism Algorithms with Descent Method. JAISCR. 2015; 5(4).

[24] UCI Repository of Machine Learning Databases and Domain Theories. ftp://ftp.ics.uci.edu/pub/machinelearning-Databases