

Private Representative-Based Clustering for Vertically Partitioned Data

Vladimir Estivill-Castro
Institute for Intelligent and Integrated Systems
School of CIT, Griffith University
Nathan Campus 4111 QLD, Brisbane
v.estivill-castro@griffith.edu.au

Abstract

This paper studies how to construct a representative-based clustering algorithms under the scenario that the dataset is partitioned into at least two sections. One section of the data is owned by Alice while the other is owned by Bob. Both want to compute clusters from the union of the data but do not trust each other. Thus, they do not want the other party to learn anything about their share of the data except what can be inferred from the results.

We present a protocol that allows Alice and Bob to carry this task under the k -medoids algorithm. Clustering with medoids (medians or other loss functions) is a more robust alternative that clustering with k -MEANS (the only method for which a privacy preserving protocol is known, but a method that is statistically biased and statistically inconsistent with very low robustness to noise). Our approach highlights the necessary building blocks for extending our protocol to the family of representative-based clustering algorithms.

1. Introduction

Data Mining (DM) technology [5] allows the analysis of large amounts of data. Analyzes of personal data, or analyzes of corporate data (by competitors, for example) create threats to information privacy and may allow for easier data surveillance. With the current emphasis on intelligence for safeguarding modern societies from crime and terrorism, DM technology is to be applied to analyze large amounts of digitally recorded data about the activities and operations of individuals and organization for spotting out the potentially dangerous [28]. This threatens privacy and values of democratic societies.

Companies, governments and research institutions are aware that data is one of the most important corporate assets supporting OLAP (On-Line Analytical Processing) and informed decision-making. DM is part of the arsenal for anal-

ysis that provides crucial insight. Often, personal data is analyzed for purposes beyond the original intent and without consent of individuals involved. The legal framework for privacy seems behind advances in information technology [26, 32]. DM has helped fight tax evasion as well as reduction of possibilities in criminal investigations [5]. DM techniques were used in the aftermath of the Oklahoma city bombing [5]. In the US “Both the Federal Bureau of Investigations and the new Department of Homeland Security have identified data mining as a key component in combating crime and terrorism in the 21st century” [28]. Analyzing data brings collective benefit in many contexts, thus privacy advocates struggle to push legislation restricting the secondary use of personal data. While there are beneficial applications of DM, individuals easily imagine the potential damage of unauthorized analyzes and these perceptions have resulted in a U.S. senate proposal for a “Data Mining Moratorium Act” [19].

Initially, central figures (Fayyad, Piatesky-Shapiro and Smyth [18] and Klösgen [25]) described evident privacy concerns with DM. Privacy evolved into a debate about the possibility or impossibility of performing Data Mining without access to data [32]. Research by Clifton [9, 10] and by Estivill-Castro and Brankovic [6, 15] kept the debate alive. Now, privacy issues in Data Mining are well in the research agenda under the term “Privacy Preserving Data Mining”, indicating that ways for obtaining meaningful answers while ensuring that the data remains private.

Privacy preserving data mining follows now two strong avenues. First, the strand by Estivill-Castro and Brankovic [15] (who analyzed the relationship to methods in statistical databases) favors data perturbation methods. Agrawal and Srikant [2] and Agrawal with Agrawal [1] as well as others [16, 17] have followed this research, referred here as *data perturbation*. Data perturbation methods for clustering are recent [33]. The second line was initiated by Lindell and Pinkas [27] and considers situations where a Data Mining tasks are to be performed by parties that have different sections of the data.

While they will both benefit from obtaining the result, neither wants the other to learn (about each other's data) more than what the result would imply. This type of computations are the focus of "Secure Multi-party Computation" (SMC) [12]. While the work by Lindell and Pinkas [27] concentrated on classification (supervised learning), it assumed that the data split was *horizontal* (the set of attribute vectors for learning is split among the parties; also called *homogeneous* [3, 11, 12]). For vertically partitioned data (where each party holds a subset of the attributes and a record for an individual is thus split across the parties; also called *heterogeneous* [3, 11, 12]), the classification problem was solved [13] for decision trees using an untrusted third party.

The next Data Mining task investigated for vertically partitioned data was finding association rules [37]; while clustering was mentioned as an open problem [12]. A protocol was recently proposed for the k -MEANS algorithm and vertically partitioned data [38]. In this paper we address the problem of finding a protocol for more robust clustering algorithms than k -MEANS. While k -MEANS is popular, it is a poor choice unless strong assumptions that support its induction principle hold for the data at hand [14] (and even for noiseless data from multi-variate normal mixtures with equal covariance, the algorithm is statistically biased).

We will present a protocol that allows two or more parties to carry out a clustering task under the k -medoids algorithm. Clustering with medoids (medians or other loss functions) is a more robust alternative that clustering with k -MEANS. Our approach highlights the necessary building blocks for extending our protocol to the family of representative-based clustering algorithms.

2. Clustering

Clustering finds sub-families among a large collection of data. It is usually described in the DM literature [5] as the tool of preference to understand better a large group and find subgroups. Privacy preserving clustering is more apparent in settings where different data holders have different data about the same individuals (vertically partitioned data). For example, a government agency may wish to group individuals as law obeying citizens, potentially dangerous individuals, and certainly dangerous individuals based on attributes available to the law enforcing agencies. However, a more accurate clustering could be obtained if data about the financial transactions of individuals was available as well. Then, police resources could be more focused for more promising (and perhaps preventive) investigations. But financial transactions or phone records may be the ownership of banks or phone companies that may or may not be obliged to disclose (in some countries these records are not to be made available unless the individual is being charged).

This example is for illustration, since the situation may actually be a complex setting where data is distributed among different financial institutions — credit card companies, banks, taxation offices, and so on. How can the clustering be performed without the holders of the data being forced to release/share the data to each other (with clear implications to the privacy of the individuals whose data is being disclosed)? Privacy preserving clustering of vertically partitioned data is the technical solutions to this challenge. Figure 1 illustrates that although the parties may have projected data that reflects 4 clusters for each, the cluster assignment for full dimensions is radically different.

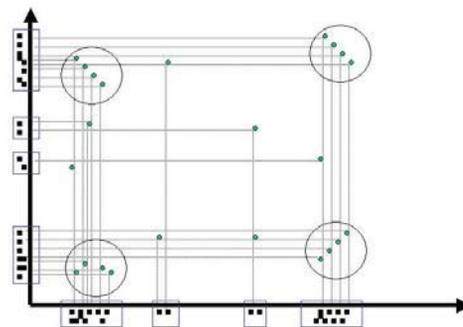


Figure 1. Two parties, each the owner of one projection, obtain radically different clusters than in the full dimensions.

Clustering is a data analysis procedure that identifies commonalities in items. The properties of items are usually encoded as attribute-valued vectors that describe them. Based on this, clustering algorithms find groups. Items placed in the same class are to exhibit similarities. Informally, clustering is described as finding groups that minimize the intra-group dissimilarity while maximizing the inter-group dissimilarity [23]. As descriptions becomes more formal, similarity (or dissimilarity) is replaced by a distance, and even more formally by a metric. Since metrics are usually functions of two inputs, the natural approach is that classes are to be encoded by selecting a prototypical item for each class. Then, distance between classes is the distance between corresponding representatives (this succinct representation of classes gives way to representative-based clustering). How should the representative of a class be chosen? The statistical approach to reducing the square error suggests the mean as vector representation, when the metric is the Euclidean metric. More formally, if

$$Error^2(\vec{u}) = \sum_{\vec{v} \in C} Euclid^2(\vec{u}, \vec{v}), \quad (1)$$

(where C is a class), then $Error^2$ is minimum when $u = \frac{\sum_{\vec{x} \in C} \vec{x}}{|C|}$; that is, u is the arithmetic mean¹.

In this case, this simple solution (defining the “center” of a class as the average of its members) resulted in the algorithm k -MEANS. This algorithm attempts to find a set M of cardinality k minimizing $Error^2(M) = \sum_{i=0}^{n-1} Euclid^2(rep[\vec{x}_i, M], \vec{x}_i)$, where

- \vec{x}_i is the i -th data point,
- $rep[\vec{x}_i, M]$ is the representative of \vec{x}_i in M (usually the closest point in M to \vec{x}_i), and
- the data set $X = \{\vec{x}_0, \dots, \vec{x}_{n-1}\}$ has size n .

Note that the case $\|M\| = 1$ is Equation (1).

k -MEANS is known to obtain only approximate solutions (local optima). However, even if the global optima is found, in many occasions this does not constitute a satisfactory clustering. First, as we mentioned, the resulting representatives are biased estimators of the means of a mixture of normal distributions. However, another displeasing fact is derived from the squared distances that provide high sensitivity to noise and outliers.

The statistical literature defines several formal notions of robustness [39] and indicates medians as overwhelmingly more robust than means as estimators of location. However, the computational complexity of the corresponding optimization problem is dramatically increased. The equivalent of Equation (1) is the evaluation of the absolute error

$$Error(\vec{u}) = \sum_{\vec{v} \in C} Euclid(\vec{u}, \vec{v}). \quad (2)$$

Finding \vec{u} that minimizes $Error(\vec{u})$ is the well-known Fermat-Webber problem regarded by the theory of computation as intractable [4] (unless the dimensions D is one, in which case the answer is the median). Since the case $k = 1$ is so difficult, for continuous domains, median-based clustering is usually avoided. However, when the optimization problem for medians is restricted to MINIMIZE

$$Error(M) = \sum_{i=0}^{n-1} Euclid(rep[\vec{x}_i, M], \vec{x}_i), \quad (3)$$

where $M \subset X$, then, the problem is discrete, in that the representatives are to be vectors in the data. This choice is known in the Knowledge Discovery and DM literature as *medoids* [23, 30]. Although the problem remains NP -complete [22], a very effective heuristic (named TAB after its authors [36]) is usually used for approximate solving Equation (3). Despite being slower than k -MEANS, the clusters found are usually of much better quality.

¹ $Euclid^2(\vec{u}, \vec{v}) = (\vec{u} - \vec{v})^T \cdot (\vec{u} - \vec{v})$ is the Euclidean distance squared, also expressed as $\sum_{d=1}^D (u_d - v_d)^2$, where D is the dimension of the vectors \vec{u} and \vec{v} .

The computational complexity of the clustering problem is important for privacy preserving data mining. For computation with data split across several parties, the Secure Multi-party Computation (SMC) has a general solution for all polynomially bound computations [21]. This generic solution computes $f(\vec{x}, \vec{y})$ for a polynomial-time f using private input \vec{x} from Alice and private input \vec{y} from Bob. Alice learns nothing about \vec{y} except what can be computed from $f(\vec{x}, \vec{y})$ and similarly Bob learns nothing about \vec{x} except what can be inferred from \vec{y} . Why if such solution exists, is there so much interest in protocols for SMC? The first aspect is that the general solution requires f to be explicitly represented as a Boolean circuit of polynomial size. Even if represented as a circuit of polynomial size in its input, the input would represent the entire data sets of all the parties, which for data mining applications are very large. Third, the constants involved are not small, once the circuit is described the parties enter into a protocol holding shares of the inputs to gates and shares of the outputs of gates. Third, the literature shows than much more efficient solutions exist for special cases of f .

We are not in a position to represent a solution to Equation (3) as a circuit and also it is impractical to represent the TAB heuristic as a circuit. Therefore, we develop our specialized SMC computation of the TAB heuristic.

3. SMC techniques

We require several building blocks. Some are commonly used in privacy-preserving algorithmic problems.

3.1. SMC value comparison

The origin of SMC is known as Yao’s Millionaire Problem because of the seminal work by Yao [40]. The problem involves two parties, Alice holds a number a while Bob holds b . They want to compute the predicate $a > b$ without neither learning anything else about the others value. While many solutions have been produced improving Yao’s original solution (that required exponential complexity on the number of bits of $(a + b)$), the most common approach [3] is to adopt the recent solution by [7] since the computation complexity is linear on the number of bits of $a + b$.

3.2. Division Protocol and Scalar Product

Alice has two numbers a_1 and a_2 while Bob has another two numbers b_1 and b_2 . They want to compute $(a_1 + a_2)/(b_1 + b_2)$. The solution uses the secure scalar product, where Alice has a vector \vec{v} and Bob has a vector \vec{u} and Alice computes $\vec{v} \cdot \vec{u}$ without revealing each others vector. Several solutions for the secure-multiparty scalar product are available [11, 13] and we will not describe them here for space

reasons. However, the division protocol can easily be described.

- Bob produces two random numbers r_1 and r_2 and sends $\lambda = r_2/r_1$ to Alice.
- Then, both parties use SMC scalar product for Alice only to compute $(a_1, 1)^T \cdot (r_1, r_1 b_1) = r_1(a_1 + b_1)$.
- Using the scalar product again Alice only computes $(a_2, 1)^T \cdot (r_2, r_2 b_2) = r_2(a_2 + b_2)$, without Bob knowing the result.
- Then, Alice computes $\lambda r_1(a_1 + b_1)/r_2(a_2 + b_2)$ which equals the desired result and sends it to Bob.

3.3. Add vectors

The technique was introduced for manipulation of vector operations as the ‘permutation protocol’ [11] and is also known as the ‘permutation algorithm’ [38]. Alice has a vector \vec{x} while Bob has a vector \vec{v} and a permutation π . The goal is for Alice to obtain $\pi(\vec{x} + \vec{v})$; that is Alice obtains the sum \vec{z} of the vectors in some sense. The entries are randomly permuted, so Alice cannot perform $\vec{z} - \vec{x}$ to find \vec{v} . Also, Bob is not to learn \vec{x} . The solution is based on homomorphic encryption for which many implementations are possible [29, 31]. Homomorphic encryption has the property that, for any two values x and y , the encryption of the sum $E(x + y)$ can be computed from the encryptions $E(x)$ and $E(y)$. Thus, the protocol works as follows.

- Alice produces a key pair for a homomorphic public key system and sends the public key to Bob. We denote by $E(\cdot)$ and $D(\cdot)$ the corresponding encryption and decryption system.
- Alice encrypts $\vec{x} = (x_1, \dots, x_k)^T$ and sends $E(\vec{x}) = (E(x_1), \dots, E(x_k))^T$ to Bob.
- Using the public key from Alice, Bob computes $E(\vec{v}) = (E(v_1), \dots, E(v_k))^T$ and uses the homomorphic property to compute $E(\vec{x} + \vec{v})$ as $\vec{z} = E(\vec{x}) * E(\vec{v})$. Then, permutes the entries in \vec{z} according to π , and sends $\pi(\vec{z}) = \pi(E(\vec{x} + \vec{v}))$ to Alice.
- Alice decrypts to obtain $D(\pi(\vec{z})) = D(\pi(E(\vec{x} + \vec{v}))) = \pi(\vec{x} + \vec{v})$.

3.4. Voronoi evaluation

The third technique we will use is the evaluation of a Voronoi-cell query. We present an improved version that does not require to use smc-circuit evaluation anywhere (before the classification step of k -means [38] required circuit evaluation for a comparison predicate of row sums). The problem is formally defined as follows. First, there is a distance $\delta(\vec{x}_i, \vec{x}_j)$ (say the Minkowski distance $\delta_\alpha(\vec{x}_i, \vec{x}_j) =$

$[\sum_{d=1}^D |x_{id} - x_{jd}|^\alpha]^{1/\alpha}$; typically $\alpha = 2$, the Euclidean distance or $\alpha = 1$, the Manhattan distance). Second, we have k -vectors $\{\vec{c}_1, \dots, \vec{c}_j, \dots, \vec{c}_k\}$ among r parties. These vectors represent a set of k -sites in \mathbb{R}^D and each party P_p knows its corresponding projection $\pi_p(\vec{c}_j)$ for each site (and only each party knows such data). An illustration of this is shown in Figure 2 with bidimensional data for two parties ($r = 2$). So, each party can compute $\delta_\alpha(\pi_p(\vec{x}_i), \pi_p(\vec{c}_j))^\alpha$.

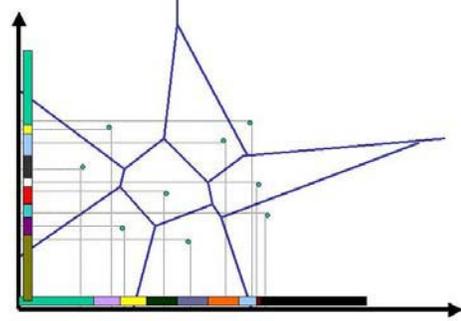


Figure 2. The $D = 2$ Voronoi diagram is different than each $D = 1$ Voronoi diagram.

Note that $\sum_{p=1}^r \delta_\alpha(\pi_p(\vec{x}_i), \pi_p(\vec{c}_j))^\alpha = \delta_\alpha(\vec{x}_i, \vec{c}_j)^\alpha$. Then, given a query point \vec{x}_i , the problem can be considered as finding the row of minimum sum in a matrix with columns $\vec{\delta}_1^{(i)}, \dots, \vec{\delta}_r^{(i)}$ and each column is only known to one party. Each column $\vec{\delta}_p^{(i)}$ is a k -dimensional vector $(\delta_{1p}^{(i)}, \delta_{2p}^{(i)}, \dots, \delta_{kp}^{(i)})^T$, where $\delta_{jp}^{(i)} = \delta_\alpha(\pi_p(\vec{x}_i), \pi_p(\vec{c}_j))^\alpha$. Finding the row with smallest sum is equivalent to finding the Voronoi cell for \vec{x}_i because the sum of row j is the distance from \vec{x}_i to the j -th site to the α power and the function $f(x) = x^\alpha$ is monotone (note that for the Manhattan distance there is a slight ambiguity [24]). For k -means, medians or medoids, $\alpha = 2$ must be used.

The smc-algorithm for Voronoi cell location requires 3 non-colluding parties. These could be among the parties holding the data (but there could be additional parties). These parties involved in the algorithm will only find the number r of parties and the number k of clusters. For convenience, we use P_1, P_2 and P_3 as the 3 parties in our description. We expect parties involved in the computation not to collude with each other because they are performing privacy preserving clustering as a result of not trusting each other. The model used here is a semi-honest model [20]. Mainly, this means that the parties have an interest in the final result. Thus, they would follow the protocol as much as it leads to the correct output. However, anything on top of the protocol that allows them to learn information about data belonging to others, we expect each party to carry out.

The algorithm will use mainly 3 techniques. First, the sums across rows are computed by the parties after adding random numbers per row preventing each other to find out the true partial sum values. Second, a comparison to distinguish if a row s is less than or equal than a row t follows a pattern similar to Yao's Millionaire Problem. Third, random permutation of rows hides away the information developed for further computations, since it masks which index of the rows is the resulting minimum sum.

3.4.1. Step 1: Disguising the sums The first party P_1 generates a k -dimensional vector \vec{v}_j for each party (including itself, i.e. $j = 1, \dots, r$), so that $\sum_{p=1}^r \vec{v}_p = \vec{0}$. Party P_1 also generates a random permutation $\pi \in S_k$ for disguising the order of the rows. Now, P_1 performs the 'permutation protocol' r times. It uses the permutation protocol with each of the parties P_p for $p = 1, \dots, r$ (note that the first permutation protocol is with itself, but the notation is uniform and this is also necessary as we will see shortly). Each time it uses π and \vec{v}_p as its secret part, while the partner party P_p uses $\vec{\delta}_p^{(i)}$ in the protocol as its private vector. Thus, each party obtains $\pi(\vec{v}_p + \vec{\delta}_p^{(i)})$ without revealing its vector of projected distances. Now, all parties except P_2 send this result to P_3 .

3.4.2. Step 2: Finding a row of minimum sum Now, P_3 can compute $\vec{a} = \sum_{p=1, p \neq 2}^r \pi(\vec{v}_p + \vec{\delta}_p^{(i)})$. On the other hand P_2 holds $\vec{b} = \pi(\vec{v}_2 + \vec{\delta}_2^{(i)})$. Now consider the question is the sum in row $\pi(j_1)$ less or equal than the sum in row $\pi(j_2)$. This is equal to the question, does the vector $\sum_{p=1}^r \pi(\vec{v}_p + \vec{\delta}_p^{(i)}) = \sum_{p=1}^r \pi(\vec{v}_p) + \sum_{p=1}^r \pi(\vec{\delta}_p^{(i)}) = \sum_{p=1}^r \pi(\vec{\delta}_p^{(i)})$ have a value in its $\pi(j_1)$ -th entry smaller or equal than in its $\pi(j_2)$ -th entry (recall $\sum_{p=1}^r \pi(\vec{v}_p) = \vec{0}$). We let a_1 be the $\pi(j_1)$ -th entry in \vec{a} in P_3 while we let a_2 be the negative of the $\pi(j_2)$ -th entry in \vec{a} in P_3 . We let b_1 be the $\pi(j_1)$ -th entry in \vec{b} in P_2 while we let b_2 be the negative of the $\pi(j_2)$ -th entry in \vec{b} in P_2 . Then using the division protocol P_2 and P_3 can answer without disclosing their distance vectors if $(a_1 + a_2)/(b_1 + b_2) < 1$ (or equivalently if $(a_1 + a_2) < (b_1 + b_2)$). Thus, P_2 and P_3 have a comparison predicate for the sum of the rows, and would find in k calls to that comparison predicate the index $\pi(i)$ that results in the row with smallest sum. Either P_2 or P_3 sends $\pi(i)$ to P_1 who, as the owner of π can broadcast i to all parties.

4. Privacy Preserving Tab

The TAB heuristic works as follows. An initial set $M_0 \subseteq X$ is randomly chosen (and $\|M_t\| = k$, for $t \geq 0$, where t denotes the iteration number). Also, as part of the algorithm's initial setting, the data $X = \{\vec{x}_0, \dots, \vec{x}_{n-1}\}$ is considered as a circular list (with \vec{x}_i followed by $\vec{x}_{(i+1) \bmod n}$).

At iteration t , the algorithm takes the next data point (say \vec{x}_i) in the circular list and attempts to construct a new set of representatives M_t by inserting \vec{x}_i into M_{t-1} and removing some data point from M_{t-1} and placing it at the end of the circular list. Thus, always $\|M_t \cap M_{t-1}\| = k - 1$, and the algorithm is a local-search (hill climber). It evaluates all swaps of \vec{x}_i with the k data points in $M_{t-1} = \{\vec{c}_{0^{t-1}}, \vec{c}_{1^{t-1}}, \dots, \vec{c}_{(k-1)^{t-1}}\}$. Essentially comparing $Error(M_{t-1} \cup \{\vec{x}_i\} \setminus \{\vec{c}_{j^{t-1}}\})$ with $Error(M_{t-1})$, for $j = 0, \dots, k - 1$ (recall $Error(M)$ is defined by Equation (2)). It adopts $M_{t-1} \cup \{\vec{x}_i\} \setminus \{\vec{c}_{j^{t-1}}\}$ as M_t when a reduction in error is achieved. When the algorithm goes around the entire circular list without a successful swap that reduces $Error(M_{t-1})$, the algorithm halts. It has a Tabu search flavor because once a data point is attempted for a swap it is placed at the end of all others.

We now transform this algorithms into an efficient secure multi-party computation where each party holds a projection of the input vectors in X . The first point is that the parties share some input. In particular, the correspondence of the attributes, which is equivalent to knowing the order of the circular list X . The second point is that since $M_t \subseteq X$ (for all t), and the output is a set $M \subseteq X$, the output is represented as set of indexes indicating M among X . Also, as part of the output, each \vec{x}_i must be designated to a cluster (the designated c_j in M that is its representative; that is, each party gets the values of $rep[\vec{x}_i, M]$).

This observations allow us to establish the first two points of the smc-TAB algorithm. First, there is no need for a special termination of the iteration step (as opposed to the scm version of k -MEANS [38]). The parties simply notice that the indexes in M_{t-1} (shared data) do not change on a pass in the entire circular list (alternatively, the cluster assignment of each data point does not vary). So, halting the algorithm is simple. Initializing the algorithm is also simple since this involves choosing k random integers in $[0, n - 1]$ as broadcasting them to all parties as the initial M_0 .

The real challenge is the secure and multi-party computation of the gradient (improvement) by a swap. Namely the calculation of $\Delta(\vec{x}_i, \vec{c}_{j^{t-1}})$

$$= Error(M_{t-1}) - Error(M_{t-1} \cup \{\vec{x}_i\} \setminus \{\vec{c}_{j^{t-1}}\}). \quad (4)$$

This represents two sub-challenges, first, the value of $rep[\vec{x}_i, M]$ corresponds to a Voronoi cell evaluation. That is, finding the corresponding cell for \vec{x}_i in a Voronoi diagram of k points (those in M). We can achieve this by the Voronoi evaluation protocol of the previous section.

The second sub-challenge is the actual evaluation of Equation (4). Although this can be achieved with a sophisticated variation of the Voronoi cell evaluation, we prefer here a presentation we believe is clearer. Let \vec{x}_i be the point being considered for inclusion into the set M_{t-1} of repre-

representatives. We use $\vec{x}_{i'}$ for another data point (in a sense, i' is the index inside the definition of *Error*, while i is the index for the data point TAB considers for promotion to a representative). We consider an $2n$ -dimensional labeling vector \vec{g}_{t-1} that is known by all the parties as a result of the Voronoi cell evaluation (the classification). Simply, the $2i'$ -th entry and the $2i' - 1$ -entry in \vec{g}_{t-1} are both 0 if the data $\vec{x}_{i'}$ does not change representative assignment from M_{t-1} to $M_{t-1} \cup \{\vec{x}_i\} \setminus \{\vec{c}_{j^{t-1}}\}$. In a sense, $\vec{x}_{i'}$ does not contribute to $\Delta(\vec{x}_i, \vec{c}_{j^{t-1}})$. However, if $rep[\vec{x}_{i'}, M_{t-1}]$ is different from $rep[\vec{x}_i, M_{t-1} \cup \{\vec{x}_i\} \setminus \{\vec{c}_{j^{t-1}}\}]$ then the $2i' - 1$ -th entry of \vec{g}_{t-1} is 1, while the $2i'$ entry of \vec{g}_{t-1} is -1. We also now consider the following matrix Δ with $2n$ rows and r columns, where each party P_p knows only the data for one column. The $2'i - 1$ -th entry of the p -column is the distance (to the α power) from $\pi_p(\vec{x}_{i'})$ to the projection (in party p) of its representative in M_{t-1} . That is, $[\delta_\alpha(\pi_p(\vec{x}_{i'}), \pi_p(rep[\vec{x}_i, M_{t-1}]))]^\alpha$. Similarly, The $2'i$ entry of the p -column is the distance (to the α power) from $\pi_p(\vec{x}_{i'})$ to the projection (in party p) of its representative in $M_{t-1} \cup \{\vec{x}_i\} \setminus \{\vec{c}_{j^{t-1}}\}$. With this notation, it is not hard to see that the value of $\Delta(\vec{x}_i, \vec{c}_{j^{t-1}})$ is given by

$$\Delta(\vec{x}_i, \vec{c}_{j^{t-1}}) = \vec{g}_{t-1}^T \pi'^{-1} \left[\pi'(\Delta) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \right]^{1/\alpha}$$

That is, we multiply the matrix Δ by a r -dimensional vector $\vec{1}$ with all entries set to 1. We take the α -th root in each entry and then we multiply the result from the left with the transpose of \vec{g}_{t-1} . In order to perform this securely without each party P_p revealing its secret column in the matrix Δ we can use smc-scalar product with a third non-colluding party (as for example in secure computation of Decision Trees [13]). However, a random permutation π' is applied to the rows of Δ and its inverse to the entries in \vec{g}_{t-1} for additional security and disguise of which x_i changed representative (applying the permutation in this way does not affect the result, in fact π' could be the identity).

This solution has the problem that the third party may learn distances to some representative (although it does not know which representative or which data points). However, by noticing that TAB can operate with only the sign of $\Delta(\vec{x}_i, \vec{c}_{j^{t-1}})$ (adopting the swap if positive) and as we mentioned before, imitating the Voronoi evaluation we can obtain a more secure result by a commodity server that generates a random number and this random number is placed in all entries of the vector $\vec{1}$ (rather than all entries being 1).

5. Privacy Issues

Each component of our sms-TAB algorithms is secure, but there are some risks that remain as with many of the

privacy preserving algorithms in the literature where parties may learn some information about the data of others as the iterations progress. One iteration of k -MEANS [38], or one step to expand a decision tree [13], or one iteration of our smc-TAB medoid clustering algorithm would not reveal information to any party about data from another. However, the pattern of changes in the representatives in M_t (in our algorithm) or the evolution of representatives $\vec{\mu}^j$ (in the k -MEANS version [38]) over several iterations could potentially allow one party to infer something about the data of another. We make this observation but also remark that it seems also very hard such information could be useful. In the same vein as other privacy-preserving algorithms in the literature, we regard this risk as insignificant.

What may appear as also a risk is that one or more of the parties may not follow the semi-honest model. It is conceivable that some parties may lie about their share of the data in some iterations in order to obtain information about the data of others. For example, a party could secretly make its column in the matrix Δ all zero as well as its column in the Voronoi evaluation for even values of $t < n$ (while using the true values for odd values of t or $t \geq n$). It would be extremely difficult to detect such behavior and the final output would be the same as if the party had behaved under the semi-honest model. However, the same risk applies to the k -MEANS iteration and again the possibility of learning something useful is extremely small. Moreover, if several parties behaved in this way with strange patterns for counterfeiting their values, they will be introducing noise into the overall algorithm that would delay its convergence without any gain and nobody would learn anything useful. So, we find this risk also insignificant.

6. Fuzzy-C-Means and EM

Statistical inference with finite mixtures offers a famous representative-based clustering approach, namely Expectation Maximization (EM) [35]. Typically, each component in the mixture is a multivariate normal distribution

$$N_{\vec{\mu}_j, \Sigma_j}(\vec{x}) = \frac{\exp\{-\frac{1}{2}(\vec{x} - \vec{\mu}_j)^T \Sigma_j^{-1}(\vec{x} - \vec{\mu}_j)\}}{\sqrt{(2\pi)^D |\Sigma_j|}},$$

where $|\Sigma|$ is the determinant of Σ_j . It is also commonly assumed that the covariance matrix Σ_j of each component $N_{\vec{\mu}_j, \Sigma_j}$, although unknown, is diagonal. We let ρ_j denote the rate of participation of the j -th component in the mixture. Also, denoting by $\vec{\mu}_j^t$ the representatives at time t (for $j = 1, \dots, k$), Σ_j^t the covariance matrices (for $j = 1, \dots, k$), and $\vec{\sigma}_j^t$ the k diagonal entries of Σ_j^t , the EM updating rule for estimating the weight (membership) w_{ij}

of data point \vec{x}_i to cluster j at the t -th iteration is:

$$w_{ij}^t = \frac{\left(\frac{1}{n} \sum_{q=0}^{n-1} w_{qj}^t\right) N_{\vec{\mu}_j^{t-1}, \Sigma_j^{t-1}}(\vec{x}_i)}{\sum_{s=1}^k \left(\frac{1}{n} \sum_{q=1}^n w_{qs}^{t-1}\right) N_{\vec{\mu}_s^{t-1}, \Sigma_s^{t-1}}(\vec{x}_i)} \quad (5)$$

for $j = 1, \dots, k$ and $i = 0, \dots, n-1$. This is the expectation phase or *data completion* step (as referred by the statisticians) and correspond to the Voronoi diagram evaluation in k -MEANS (a cluster labeling step). The re-evaluation of the model (the maximization step because of maximum likelihood) computes the new representatives and the covariance of their groups by, for $j, s = 1, \dots, k$:

$$\vec{\mu}_j^t = \frac{\sum_{i=0}^{n-1} w_{ij}^t \vec{x}_i}{\sum_{i=0}^{n-1} w_{ij}^t}, \quad \sigma_{js}^t = \sqrt{\frac{\sum_{i=0}^{n-1} w_{ij}^t (\vec{x}_{is} - \vec{\mu}_{js}^t)^2}{\sum_{i=0}^{n-1} w_{ij}^t}}$$

Since the result of the clustering is the parametric description of the mixture all parties will obtain the rates ρ_j (for $j = 1, \dots, k$). As in k -MEANS or medoid clustering, each party P_p would have its projection $\pi_p(\mu_j)$ of each of the j centers. A more delicate issue is the covariance matrices Σ_j ($j = 1, \dots, k$). One one hand, it is conceivable that parties may want to compute covariance coefficients of their projections (computing a covariance from two private vectors of separate parties has been considered [11]). On the other hand it is common to consider all Σ_j diagonal matrices. A further common simplification is that all components have the same known covariance Σ (for example, it is even common to assume all diagonal entries are equal to 1 [34]); thus, the only unknown parameter of each component in the mixture is the mean $\vec{\mu}_j$ (the representative). In this special case, the rule for σ_{js}^{t+1} is no longer needed, and in the estimation of w_{ij} the assumed covariance Σ is used in $N_{m_j, \Sigma}^t(\vec{x}_i)$ (Equation (5)).

The advantageous aspect of diagonal matrices Σ_j is that the entire Maximization step can be performed by each party privately and the only aspect that we need to modify for a privacy preserving EM algorithm (for vertically partitioned data) is the evaluation of a multivariate normal distribution. However, EM offers a more elaborate (informative) model of the data as clusters where each data point has a membership rate. At each iteration, all parties would know ρ_j and w_{ij} . The value of w_{ij} is proportional to $\rho_j N_{\vec{\mu}_j, \Sigma_j}(\vec{x})$ and in cases where Σ_j is simple (like the common case of a diagonal matrix) each party would learn $(\vec{x}_i - \vec{\mu}_j)^T (\vec{x}_i - \vec{\mu}_j)$ scaled by a constant that could be inferred from the many \vec{x}_i . Thus, each party could infer the Euclidean distance of \vec{x}_i to each $\vec{\mu}_j$ in the full dimension D and not only its projection. This is simply a consequence that such information can be derived from the output of EM in this case. If we are prepared to accept that each party computes privately $\delta_2(\pi_p(\vec{x}_i), \pi_p(\vec{\mu}_j))^2$ and that each party may learn $\sum_{p=1}^r \delta_2(\pi_p(\vec{x}_i), \pi_p(\vec{\mu}_j))^2 = \delta_2(\vec{x}_i, \vec{\mu}_j)^2$, then

EM can easily have a privacy-preserving implementation without circuit simulation. This may be unsatisfactory in the case of two parties ($r = 2$), since knowing the sum and private distance would allow to know the private distance of the other party; although the actual data point \vec{x}_i would still remain private.

The EM applied with a common known covariance matrix in all components is extremely similar to FUZZY- c -MEANS [8] (w_{ij} corresponds to $\text{MEM}_j^t(\vec{x}_i)$, a fuzzy membership value). Moreover, if the w_{ij} are forced to be crisp, (i.e. $w_{ij} \in \{0, 1\}$), then EM reduces to k -MEANS. The FUZZY- c -MEANS algorithm has a classification step that revises the fuzzy membership and is given by

$$\text{MEM}_j^t(\vec{x}_i) = \frac{(1/(\vec{x}_i - \vec{c}_j^{t-1})^T (\vec{x}_i - \vec{c}_j^{t-1}))^{1/(b-1)}}{\sum_{s=1}^k (1/(\vec{x}_i - \vec{c}_s^{t-1})^T (\vec{x}_i - \vec{c}_s^{t-1}))^{1/(b-1)}}, \quad (6)$$

where b regulates the degree of fuzziness (with crisper classifications as $b \rightarrow 1$ and fuzzier as $b \rightarrow \infty$; while $b = 2$ is common practice). Then, the representatives are revised by

$$\vec{c}_j^t = \frac{\sum_{i=0}^{n-1} [\text{MEM}_j^t(\vec{x}_i)]^b \vec{x}_i}{\sum_{i=0}^{n-1} [\text{MEM}_j^t(\vec{x}_i)]^b}. \quad (7)$$

for $j = 0, \dots, k$. Once again, each party can perform the revision of representatives in their projection and keeping the corresponding part of each representative private. Also, once more, Equation (6) makes possible to learn distances in D dimension as all parties need to no the degree $\text{MEM}_j^t(\vec{x}_i)$ of membership in cluster j for each point \vec{x}_i .

7. Conclusions

We have presented privacy preserving algorithm for robust representative-based clustering. We have achieved this in the case of more demand for clustering, namely the vertical (heterogeneous) partitioning of the data and our methods do not require the use of the theoretical circuit emulation protocols for secure multiparty computation. These are two improvements with respect to the available literature (more robust than k -MEANS and even the privacy preserving k -MEANS available used circuit simulation).

We have highlighted that as the clustering is more robust, the result are models that encode more information about the data. In particular, with EM and a mixture of multivariate normal distributions one is expected to obtain from the result all parameters of the mixture (participation rates for each component, means and covariance matrices for each multi-variate). As this result is shared by the parties, it becomes feasible for parties to infer slightly more about the data. Thus, the participating parties must be made aware of the implications of using these clustering approaches.

References

- [1] D. Agrawal and C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. 20th SIGACT-SIGMOD-SIGART PODS 2001. ACM Press.
- [2] R. Agrawal and R. Srikant. Privacy-preserving data mining. SIGMOD, 439–450, Dallas, TX, 2000; ACM Press.
- [3] M.J. Atallah and W. Du. Secure multi-party computational geometry. 7th WADS 165–179. LNCS 2125, 2000.
- [4] C. Bajaj. Proving geometric algorithm non-solvability: An application of factoring polynomials. *J. Symbolic Computation*, 2:99–102, 1986.
- [5] M.J.A. Berry and G. Linoff. *Data Mining Techniques — for Marketing, Sales and Customer Support*. Wiley, NY, 1997.
- [6] L. Brankovic and V. Estivill-Castro. Privacy issues in knowledge discovery and data mining. *AICEC99*, 89–99, Melbourne, 1999.
- [7] C. Cachin. Efficient private bidding and auctions with an oblivious third party. *6th ACM Computer and communications security*, 120–127, 1999. SIGSAC, ACM Press.
- [8] V. Cherkassky and F. Muller. *Learning from Data — Concept, Theory and Methods*. Wiley, NY, 1998.
- [9] C. Clifton. Using sample size to limit exposure to data mining. *J. of Computer Security*, 8(4):281–307, 2000.
- [10] C. Clifton and D. Marks. Security and privacy implications of data mining. *SIGMOD Workshop on Data Mining and Knowledge Discovery*, Montreal, 1996. ACM.
- [11] W. Du and M.J. Atallah. Privacy-preserving cooperative statistical analysis. *17th (ACSAC)*, 102–110, New Orleans, 2001. ACM SIGSAC, IEEE.
- [12] W. Du and M.J. Atallah. Secure multi-party computation problems and their applications: A review and open problems. *New Security Paradigms Workshop*, 13–22, Cloudcroft, 2001. SIGSAC, ACM Press.
- [13] W. Du and Z. Zhan. Building decision tree classifier on private data. *Privacy, Security and Data Mining*, 1–8, 2002. V 14 CRPIT, ACS.
- [14] V. Estivill-Castro. Why so many clustering algorithms – a position paper. *SIGKDD Explorations*, 4(1):65–75, 2002.
- [15] V. Estivill-Castro and L. Brankovic. Data swapping: Balancing privacy against precision in mining for logic rules. *DaWaK-99*, 389–398, Florence, , 1999. LNCS 1676.
- [16] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. *Twenty-Second SIGACT-SIGMOD-SIGART PDOS*, 211–22, San Diego, 2003. ACM Press.
- [17] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. *8th ACM SIGKDD KDD*, 217–218, Edmonton, 2002. ACM Press.
- [18] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. *Advances in Knowledge Discovery and Data Mining*, 1–36, Menlo Park, 1996. AAAI Press / MIT Press.
- [19] R.D. Feingold. S.188 data-mining moratorium act 2003. Bill Sumamry and status of the 108th Congress, Jan. 16th 2003. Cosponsors: Corzine, M. and Wyden, M. and Nelson, M.
- [20] O. Goldreich. Secure multi-party computation. Working draft, 1998.
- [21] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game (extended abstract). *19th ACM STOC*, 218–229, NY, 1987. ACM Press.
- [22] L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12:450–459, 1964.
- [23] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, CA, 2000.
- [24] R. Klein. *Concrete and Bastract Voronoi Diagrams*. Springer, 1998.
- [25] W. Klösigen. Anonymization techniques for knowledge discovery in databases. *KDD*, 186–191, 1995. AAAI Press.
- [26] Laudon. Markets and privacy. *CACM*, 39(9):92–104, 1996.
- [27] Y. Lindell and Pinkas B. Privacy preserving data mining. *CRYPTO-00*, 36–54, Sn Barbara, 2000. LNCS 1880.
- [28] J. Mena. *Investigative Data Mining for Security and Criminal Detection*. Butterworth-Heinemann, US, 2003.
- [29] D. Naccache and J. Stern. A new public key cryptosystem based on higher residues. *5th ACM Computer and Communications Security*, 59–66, 1998. SIGSAC, ACM Press.
- [30] R.T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. *20th VLDB*, 144–155, 1994. Santiago, , Morgan Kaufmann.
- [31] T. Okamoto and S.: Uchiyama. A new public-key cryptosystem as secure as factoring. *EUROCRYPT '98*, 308–318, Espoo, 1998. LNCS 1403.
- [32] D.E. O’Leary. Some privacy issues in knowledge discovery: the OECD personal privacy guidelines. *IEEE Expert*, 10(2):48–52, April 1995.
- [33] S. Oliveira and O.R. Zaiane. Privacy preserving clustering by data transformation. *18th Brazilian SBBD*, 304–318, Maaus, 2003.
- [34] J.J. Oliver, R.A. Baxter, and C.S. Wallace. Unsupervised learning using MML. *13th Machine Learning Conference*, 364–372, San Mateo, 1996. Morgan Kaufmann.
- [35] M.A. Tanner. *Tools for Statistical Inference*. Springer-Verlag, NY, US., 1993.
- [36] M.B. Teitz and P. Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16:955–961, 1968.
- [37] J. Vaidya and C. C. Clifton. Privacy preserving association rule mining in vertically partitioned data. *8th ACM SIGKDD KDD*. ACM Press, 2002.
- [38] J. Vaidya and C. C. Clifton. Privacy-preserving *k*-means clustering over vertically partitioned data. *SIGKDD-ACM KDD*, Washington, 2003. ACM Press.
- [39] R.R. Wilcox. *Introduction to Robust Estimation and Hypothesis Testing*. Academic Press, San Diego, 1997.
- [40] A.C. Yao. Protocols for secure computation. *IEEE FOCS*, 1982.