

# Privacy Preserving OPTICS Clustering

**Janardhan Reddy Kondra**



Department of Computer Science and Engineering  
**National Institute of Technology Rourkela**

# Privacy Preserving OPTICS Clustering

*Dissertation submitted in partial fulfillment*

*of the requirements of the degree of*

***Master of Technology***

*in*

***Computer Science and Engineering***

*by*

***Janardhan Reddy Kondra***

(Roll Number: 214CS2397)

*based on research carried out*

*under the supervision of*

***Prof. Sathya Babu Korra***



May, 2016

Department of Computer Science and Engineering  
**National Institute of Technology Rourkela**



Department of Computer Science and Engineering  
**National Institute of Technology Rourkela**

---

**Prof. Sathya Babu Korra**

Assistant Professor

May 20, 2016

## **Supervisor's Certificate**

This is to certify that the work presented in the dissertation entitled *Privacy Preserving OPTICS Clustering* submitted by *Janardhan Reddy Kondra*, Roll Number 214CS2397, is a record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of *Master of Technology in Computer Science and Engineering*. Neither this dissertation nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

---

Sathya Babu Korra

# Dedication

Dedicated to my Parents and inspiring guide...

*Signature*

# Declaration of Originality

I, *Janardhan Reddy Kondra*, Roll Number *214CS2397* hereby declare that this dissertation entitled *Privacy Preserving OPTICS Clustering* presents my original work carried out as a PostGraduate student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections “Reference” or “Bibliography”. I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

May 20, 2016  
NIT Rourkela

*Janardhan Reddy Kondra*

# Acknowledgment

This thesis, though an individual work, has benefited in various ways from several people. Whilst it would be simple to name them all, it would not be easy to thank them enough.

The enthusiastic guidance and support of Asst. Prof. Sathya Babu Korra inspired me to stretch beyond my limits. His profound insight has guided my thinking to improve the final product. My solemnest gratefulness to him. A special thanks to Pramit Mazumdar sir who gave me valuable suggestions throughout my research.

I'd like to thank my parents and my brother who have given me the liberty and means to do anything I want in my life. I'd also like to thank all my friends who have created a healthy environment for competition and camaraderie that enabled me to push further to achieve what I set out to. I'd like to thank NIT Rourkela for all opportunities, comforts, troubles and most importantly the memories that I will cherish for life.

May 15, 2016  
NIT Rourkela

*Janardhan Reddy Kondra*  
Roll Number: 214CS2397

# Abstract

OPTICS is a well-known density-based clustering algorithm which uses DBSCAN [1] theme without producing a clustering of a data set openly, but as a substitute it creates an augmented ordering of that particular database which represents its density-based clustering structure. This resulted cluster-ordering comprises information which is similar to the density based clustering's conforming to a wide range of parameter settings. The same algorithm can be applied in the field of privacy preserving data mining, where extracting the useful information from data which is distributed over a network requires preservation of privacy of individuals information. The problem of getting the clusters of a distributed database is considered as an example of this algorithm, where two parties want to know their cluster numbers on combined database without revealing one party information to other party. This issue can be seen as a particular example of secure multi-party computation and such sort of issues can be solved with the assistance of proposed protocols in our work along with some standard protocols.

***Keywords: Density based clustering; Privacy Preserving; OPTICS; Distributed data; Secure Multi-Party Computation.***

# Contents

|   |            |
|---|------------|
| <b>Supervisor’s Certificate</b>                         | <b>ii</b>  |
| <b>Dedication</b>                                       | <b>iii</b> |
| <b>Declaration of Originality</b>                       | <b>iv</b>  |
| <b>Acknowledgment</b>                                   | <b>v</b>   |
| <b>Abstract</b>   | <b>vi</b>  |
| <b>List of Figures</b>                                  | <b>ix</b>  |
| <b>List of Tables</b>                                   | <b>x</b>   |
| <b>1 Introduction</b>                                   | <b>1</b>   |
| 1.1 Motivation . . . . .                                | 1          |
| 1.2 The Privacy Preserving Data Mining (PPDM) . . . . . | 2          |
| 1.3 Privacy Preserving Data Clustering (PPDC) . . . . . | 3          |
| <b>2 Related Work</b>                                   | <b>5</b>   |
| <b>3 Preliminaries</b>                                  | <b>7</b>   |
| 3.1 DBSCAN Algorithm . . . . .                          | 7          |
| 3.2 OPTICS Algorithm . . . . .                          | 10         |
| 3.3 Distributed Data Mining . . . . .                   | 11         |
| 3.4 Millionaire’s protocol: . . . . .                   | 11         |
| 3.5 Secure scalar product protocol: . . . . .           | 12         |
| <b>4 Secure OPTICS Algorithm</b>                        | <b>14</b>  |
| 4.1 Problem Statement . . . . .                         | 14         |
| 4.1.1 Vertically partitioned database . . . . .         | 14         |
| 4.1.2 Horizontally partitioned database . . . . .       | 14         |
| 4.2 Protocols . . . . .                                 | 14         |
| 4.3 Proposed Methods . . . . .                          | 18         |



|                                 |           |
|---------------------------------|-----------|
| 4.4 Correctness Proof . . . . . | 19        |
| <b>5 Conclusion</b>             | <b>23</b> |
| <b>References</b>               | <b>24</b> |
| <b>Dissemination</b>            | <b>26</b> |

# List of Figures

|     |  |    |
|-----|--|----|
| 3.1 | Directly Density Reachable Points . . . . .            | 8  |
| 3.2 | Density Reachable Objects . . . . .                    | 9  |
| 3.3 | Density-Connected objects . . . . .                    | 9  |
| 3.4 | Clusters w.r.t. Different Density Parameters . . . . . | 10 |
| 3.5 | Horizontally Partitioned Data. . . . .                 | 11 |
| 3.6 | Vertically Partitioned Data. . . . .                   | 12 |
| 3.7 | Arbitrarily Partitioned Data. . . . .                  | 12 |

# List of Tables

# Chapter 1

## Introduction

Ordering the points to identify the clustering structure (OPTICS) [2] is a clustering algorithm based on density which finds the clusters in spatial data. This algorithm's basic theme is similar to that of DBSCAN, but at the same time it discourses one major drawback of DBSCAN Algorithm, here the drawback is that the problem of identifying clusters with some meaning in spatial data with variable density. To overcome that drawback, the objects in database are initially arranged in a way that objects that are close to each other in spatial becomes neighbors in the ordering. Adding to this, another distance metric is also stored, by which the density is represented so that the metric has to be accepted for that particular cluster in a way that the two points will be present in same cluster after applying clustering technique.

### 1.1 Motivation

Consider a scenario, Let two parties A and B have their own private databases  $D_A$  and  $D_B$  respectively . These two parties wants to know the clusters on their combined data, with the help of a clustering technique (OPTICS [2] ) on combined dataset  $D_A$  and  $D_B$  by keeping their individual private information securely i.e. the only information which is known to A about  $D_B$  is that which can be learned from the output of clustering algorithm, and vice versa. This issue can be considered as a secure multi-party computation [3] example. For this issue, here we have given two types of solutions, one is in the presence of trusted third party(TTP) which is used to do the calculations on combined data. Second solution is in the absence of TTP. For example, Consider an example where each hospital is having its own database of their medical records. If there is a chance of sharing data among the hospitals then its easy to mine and generate the meaningful results. But the medical records information is neither freely available nor shared with other hospitals for their privacy concerns as well as confidentiality constraints. Here comes the term privacy preserving data mining, it was Agrawal *et al.* [4] who bought this technique first into mining era.

## 1.2 The Privacy Preserving Data Mining (PPDM)

To provide better services and to make effective decisions in information era, data mining is playing a vital role, thus leads to greater profits in the real time business. To get more perks, several government institutes, small enterprise and other entities in business industry are collecting huge volumes of data about their customers which can be useful for this data mining techniques to make decisions effectively. These data mining techniques results in previous trends of their consumers and habits of particular users which helps the companies to get some idea about how to improve marketing about that particular product. Correspondingly, this extracted information has many more advantages. This may help medical organizations to keep track of history patterns and thus can be useful for better treatment to patients and it even ropes research in medical field. At the same time confidentiality of data is a main concern over networks, since many large repositories are used in data mining techniques are having confidential data that stresses privacy preservation.

Over few years the digital data is getting increased drastically and elevated concerns about individual's private information. These are the concerns which emerged in global. The field data mining is capable of handling these kind of concerns which will be used to find out most useful information at the same time it is hidden from other huge databases. One of the difficult challenges data mining and database is facing is that designing such type of information systems which protects confidentiality of databases without effecting the value of computation. So in the life of human beings data mining algorithms are playing a vital role in refining the quality, as exists in every field, even in this data mining field also there is an nonconstructive part in terms of breaking the privacy.

So this is the mixture of both blessings good and bad, hence it needs future database management systems should be privacy preserving and sensitive to the data that they manage. There are other applications of data mining such as Online Analytical Processors (OLAP), and these should also be sensitive of the target databases. Development of the techniques that include privacy problems will be very productive research for upcoming data processing research. To put in another way, forthcoming days are requiring Privacy Preserving Data Mining (PPDM) which will be carried out by both experts and researchers. The issue of data privacy in KDD (Knowledge Discovery Databases) is defined as Inference Problem [5]. Based on this definition, there is an assumption that the unauthorized data can be anecdotal from genuine replies to queries. This kind of replies in real time lead to the concept of PPDM.

The main theme of PPDM is to extract useful information from the combined datasets without exploding individual's private information to others.

### 1.3 Privacy Preserving Data Clustering (PPDC)

The field of data mining is dynamic in nature, because of this property clustering is done with many algorithms depends on datasets. In last three decades these kind of algorithms are evaluated a lot, so they are tested in time. These algorithms are characterized in many ways such as Model-Based methods, Hierarchical Methods, Constraint Based Methods, Partitioning Methods, Grid-Based Methods, Density-Based Methods, and other methods which are used for High-Dimensional Data that includes methods based on frequent patterns.

We can define PPDC as a process of clustering the objects of a database which minimizes the usage of data by other parties or that reduced the breach of data privacy whilst doing the clustering process on combined datasets. Experts from clustering have to put some efforts to this issue of data privacy in clustering as well. From the definition of clustering it is clear that there is a need to to construct Dissimilarity Matrix of data with the help of some standrd distance metrics (i.e, Euclidean Distance Metric, Manhattan Distance metric, etc.). The main purpose of these metrics is to compare an item with other items in a dataset based on their attributes or properties. Such kind of computations requires to preserve following features of the database.

- **Database should be accessed completely:** The input Database on which clustering needs to be applied has to be accessed completely since it requires a complete scan of whole dataset to compare properties or attributes of all data items with other items. This comparison shows the similarity of the data points a database, same applies to dissimilarity. Hence all data privacy preserving methods have to give the access to whole database.i.e, to all records in dataset. If there is a restriction on dataset or if access is partial then the output clusters can not include all records, at the same time they can't be generalized on complete databases by keeping intended accuracy and reliability.

- **Preserving the originality of attributes:** The distance in data clustering is calculated based their attributes or features, So all privacy preserving algorithms should not alter the original properties/features of a record or transaction. If the algorithm distorts the originality of data records then that leads to reduction of computational value of a record.

By looking at the above precautionary measures, one can say that databases should be accessed completely in data clustering and have to keep originality of data transaction after applying clustering techniques. Hence, while developing a new privacy preserving technique we should ensure that it gives the least distortion of properties of every individual record. And there should be any reduction in access to complete dataset.

Initially, the main focus is on decision trees construction from databases which are distributed over a network. Privacy preserving data mining has becoming the major topic to carry out the research. Especially generating association rules, clustering and classification. But. in this work we only emphasized on clustering which includes privacy. Previously Jha

*et al* [6] has done some research in this type of clustering. He presented k-means algorithm for clustering. This algorithm is applicable to some specific type of data which is partitioned horizontally.

We can apply this clustering technique in several fields. Like, consider an application which uses clustering algorithm, and in that application if privacy is the main concern then we can consider that as one of the example to many privacy preserving algorithms. One such scenario is that suppose there are two Internet Service Providers (ISPs) which collect network traffic over some network, and these two Internet Service Providers (ISPs) wants to get the clusters of their combined information of network traffic without exploding private information of one party traffic data to other ISP. With the help Our OPTICS horizontally partitioned algorithm one can get joint clusters whilst keeping the privacy of their in network traffic at ISPs. Consider another situation, one on-line retail company and an Internet marketing have their private databases with common set of individuals and different attributes (vertically partitioned databases). These two organizations thinks to share their data to get the clusters to get to know optimal customer targets so that they can try to increase their productivity and quality with the help of other company's information without revealing any useful information about attributes. This can be solved with the help of OPTICS for vertically partitioned algorithm.

The remaining work is illustrated as follows: Review some literature work is briefly mentioned in section 2 . Next, section 3 contains all the preliminaries which are used in our algorithm are reviewed briefly. The preliminary algorithms and protocols for computing clusters with OPTICS clustering algorithm are given in section 4 and correctness proofs are also mentioned in the same section. Section 5 concludes the work by some directions which can be useful for future work.

## Chapter 2

### Related Work

Current clustering algorithms generally categorized into two major categories, one is hierarchical and another one is partition based clustering algorithms. In hierarchical clustering algorithms, database D which contains n records is putrefied into a number of levels of nested partitioning we can call them as clusterings. Which can be represented by dendrogram i.e. a tree of smaller objects which consist of only one object after applying several splits over a database D. In that hierarchy, each element of the tree depicts a cluster of that database D. Coming to partitioned algorithms, initially it creates some set of clusters by dividing the databases D with n tuples in a way that the every tuple in a particular cluster is more closure in similarity to other tuple in that particular cluster compared to different cluster's objects. In hierarchical clustering, Single link clustering is one of the commonly used methods [7]. Initially, every object is placed in a different cluster with uniqueness compare to other clusters, From there onwards the two closest similar clusters in present clustering are merged into one cluster until only one final cluster is created for whole database. Based on the same principle , some other algorithms have been suggested in [8] [9].

In [1] by Ester M, *et al*, a clustering algorithm which is based on density based rather that grid-based is proposed. Another approach based on density is WaveCluster [10], whose logic is to apply wavelet transform for the space of features. It is capable of detecting arbitrary shaped clusters with different scales. This algorithm is not density based, but based on grid and hence can only be applied to low-dimensional datasets.

Hinnerburg, *et al* [11] proposed another density-based algorithm namely DenClue. This algorithm is also grid-based only, but in this, information is kept about the grid cells that contains original data objects but not about al grid cells. Hence this algorithm is efficient compared to all other grid based algorithms

In recent times Privacy preserving data mining has becoming a vital area of research. Firstly, it was Rakesh Agrawal and Ramakrishnan Srikanth [4] who introduced this notion. Lindall and Pinkas [12] Stimulated a solution for this kind of problems with the help of



letting parties to collaborate the mining of useful information by keeping the privacy of all participated parties.

There exists many algorithms for privacy preserving K-means clustering [13] [14] [15]. But, in clustering algorithms based on distributed density, the literature is very less. kumar *et al.* [16] has proposed a secure way of clustering with DBSCAN, another density based clustering algorithm. For vertically partitioned data, Vaidya and Clifton [14] presented the problem of privacy preserving clustering. Amirbekyan *et al.* [17] and V. Estivill-Castro [18] also proposed for vertically partitioned data. And clustering can also be done based on perturbation. Oliveira and Zaiane [19] proposed a solution based on this perturbation technique. Not only clustering, Mining of association rules [20] is an area where research is going on to generate the association rules in a privacy preserving way.

## Chapter 3

# Preliminaries

In this section, we explain density based clustering algorithms DBSCAN and OPTICS in briefly and describe the concept of partitioned data which is categorized as horizontally, vertically and arbitrarily partitioned. Some definitions and protocols are also explained which are used in algorithm.

### 3.1 DBSCAN Algorithm

DBSCAN (Density Based Spatial Clustering of Applications with Noise) is a clustering algorithm which works based on density. It can identify arbitrary shaped clusters irrespective of input datasets. Here, density is defined as smallest number of points inside some distance of each other. Inputs for this algorithm are MinPts and Eps. MinPts is defined as least number of points in a cluster and at the same time for each and every objects of any cluster the minimum constraint is that there should be some other point and the distance between these two points should be lesser than some minimum value called as Eps. Algorithm works as: Firstly starts at any arbitrary point, then checks if neighborhood of that objects is inside some given predefined radius fulfills the lowest number of points then this point is treated as core point and this process is done recursively with its neighborhood points and at the border objects. Then some other arbitrary tuple is taken and the procedure is continued till all the tuples in database have been located in the clusters. Points which are not part of any cluster and which are not in any of the neighborhood are labeled as noise.

One of key advantages of this algorithm is, it does not requires the number of final clusters in advance. Therefore, in the whole algorithm, the important task is to find the distance and to decide which two points are closer to each other. This can be done by comparing the distance between those points with Eps to know whether the distance is less than or equal to Eps. If those two tuples are belong to one party then it is easy task to find this. Otherwise, we have to implement a private protocol to get the distance between those two tuples which are owned by two different parties.

We illustrate some standard definitions which are used in DBSCAN and OPTICS algorithms.

**Definition 1: Directly Density Reachable.** An object  $p$  is called as **directly density reachable** from another point  $q$  if the point  $q$  lies in the  $\epsilon$ -neighborhood of  $p$  and  $p$  should be a core point. In figure 3.1, point  $q$  is directly density reachable from point  $p$ , but vice versa is not. And density reachability is asymmetric.

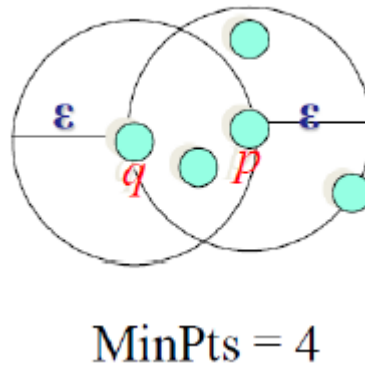


Figure 3.1: Directly Density Reachable Points

**Definition 2: Density Reachable.** A point  $p$  is defined as **density reachable** from another point  $q$  w.r.t two inputs  $\text{Eps}$  and  $\text{MinPts}$  if there exists a chain with some intermediate points  $p_1, \dots, p_n$  where first point  $p_1 = q$  and last point  $p_n = p$  such that  $p_{i+1}$  is directly density reachable from  $p_i$ . As shown in figure 3.2, point  $p$  and  $q$  are density reachable from one to other but not directly.

**Definition 3: Density-Connected.** A point  $p$  in a database  $D$  is defined as **density-connected** to another point  $q$  w.r.t.  $\text{Eps}$  and  $\text{MinPts}$  if there is a middle point  $o$  such that  $o$  and  $p$  are density reachable and  $o$  and  $q$  are also density reachable w.r.t.  $\text{Eps}$  and  $\text{MinPts}$ . as shown in figure 3.3.

**Definition 4: Cluster.** A cluster is defined w.r.t. parameters  $\text{Minpts}$  and  $\text{Eps}$  as a subset of a database  $D$  which is non-empty as follows:

1.  $\forall$  points  $p$  and  $q$ , if  $p$  belongs to a Cluster and if other point  $q$  is density-reachable from this point  $p$  w.r.t. the parameters  $\text{Minots}$  and  $\text{Eps}$ , then  $q$  also belongs to the same cluster. This is called as maximality.
2.  $\forall$  points  $p$  and  $q$  belongs to a cluster  $C$ ,  $p$  is density-connected to  $q$  w.r.t. the parameters  $\text{Mintpts}$  and  $\text{Eps}$ . This is called as connectivity.

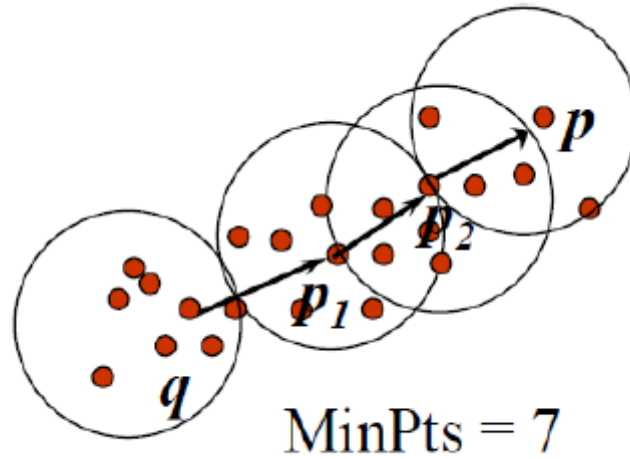


Figure 3.2: Density Reachable Objects

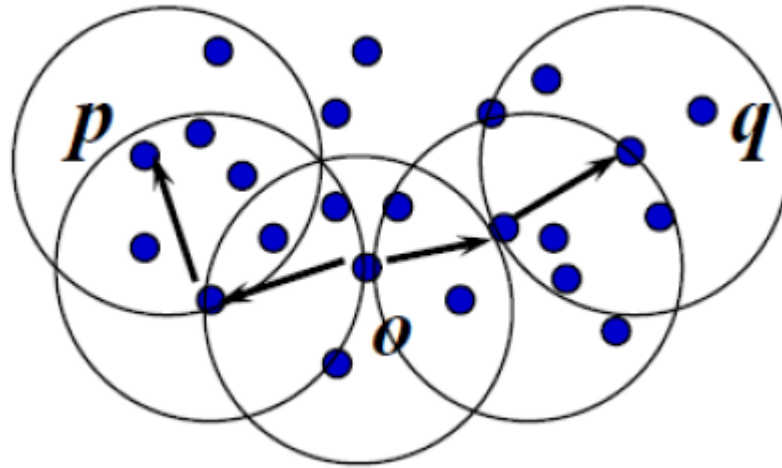


Figure 3.3: Density-Connected objects

**Definition 5: Noise.** Consider  $C_1, \dots, C_m$  be the resulted clusters of input database  $D$  w.r.t input parameters  $MinPts$  and  $Eps$ , noise is defined as set of tuples of database  $D$  not belonging to any of cluster  $C_i$  resulted after applying clustering algorithm. This can be shown as follows:  $noise = \{n \in D \mid \forall j: n \notin C_j\}$ .

**Definition 6: Core-distance of an object  $p$ .**

Let's consider  $p$  as an object of input database  $D$ , and  $\varepsilon$  be the distance metric and  $N_\varepsilon(p)$  be the  $\varepsilon$ -neighborhood of that tuple  $p$ .

By considering  $MinPts$  as a natural number let  $MinPts\text{-distance}(p)$  be the distance from  $p$  to its  $MinPts$ ' neighbor. Then

$core\text{-}distance_{\varepsilon, MinPts}(p)$  is  
 UNDEFINED, if  $Card(N_\varepsilon(p)) < MinPts$ ,

MinPts-distance(p), otherwise.

This core distance can be simply defined as the least distance  $\varepsilon'$  between two points p and q, where one object's  $\varepsilon$ -neighborhood contains another object such away that p will become a core point w.r.t.  $\varepsilon'$  only if  $N_{\varepsilon}(p)$  contains in it's neighbor. If it not present in that neighborhood, it is treated as UNDEFINED value.

**Definition 7: Reachability-Distance of a point p w.r.t. another point q.**

Consider two objects p and q of a dataset D, and  $\varepsilon$ -neighborhood of p is denoted as  $N_{\varepsilon}(p)$  and MinPts as a number from natural set, *reachability – distance* $_{\varepsilon, MinPts}(p,q)$  is defined as UNDEFINED, if  $|N_{\varepsilon}(p)|$  is less than MinPts and it is defined as maximum of core-distance(q) and distance(q, p) otherwise.

## 3.2 OPTICS Algorithm

The main drawback of DBSCAN clustering algorithm is using the global input values. DBSCAN is an optimal clustering for the entire database but that does not reflect the same for the structure of the resulted clusters in-depth. Real-time datasets comprises of regions with various densities, which lead to form some levels of clusters. Hence, one of the main properties of most of the real-data sets is that their internal cluster can't be identified with the help of global parameters like Eps and MinPts in case of DBSCAN. There might be a need of very local densities to identify clusters in different locations of data space. The key application of OPTICS is that it understands density related structure of a dataset intrinsically [2]. Consider a database shown in Figure 3.4. In that database it is literally impossible to generate the clusters as  $C_1, C_2, C_3, A$  and B with the help of global parameters. By using these global inputs the database can be decomposed into either A,B and C clusters or  $C_1, C_2$  and  $C_3$  clusters but not A, B,  $C_1, C_2$  and  $C_3$ . In former situation where  $C_1, C_2, C_3$  are generated A,B and C are treated as noise. This is the problem with the global parameters.

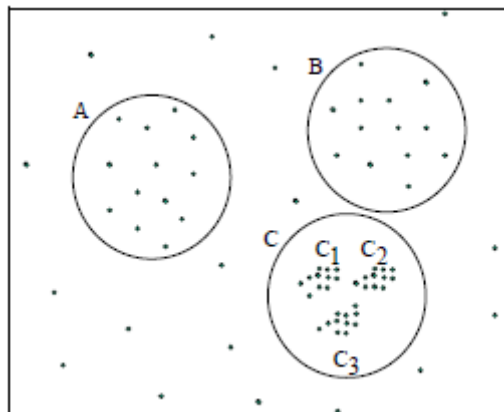


Figure 3.4: Clusters w.r.t. Different Density Parameters

To overcome this kind of problems, it is a better idea to run an algorithm that will produce another special order of the original database w.r.t it's density based clustering structures which contains the information regarding every clustering level of the input dataset. This is up to a distance called "as generating distance"  $\varepsilon$  at the same time the process is easy to analyze.

### 3.3 Distributed Data Mining

This model defines that the current datasets are distributed among many networks or sites. This distribution is further divided into three categories: Vertically, Horizontally and Arbitrarily partitioned data.

As our algorithm is solely concentrated on two-party. Here, two parties (call them Alice and Bob) own data which forms a(virtual) database which consists of their joint data. To be more specific, this virtual database called as  $D = \{ d_1, d_2, \dots, d_n \}$  contains  $n$  records. every record  $d_i$  of  $D$  has some  $m$  values with  $m$  attributes  $(d_{i,1}, d_{i,2}, \dots, d_{i,m})$ . respectively.

Three different formats are there for partitioned data:

**Horizontally Partitioned Data:** Each and every party owns a subset of records of both the datasets of two parties which includes full attributes (see figure 3.2).

**Vertically Partitioned Data:** Each and every party owns all records of both the databases with some attributes (see Figure 3.3).

**Arbitrarily Partitioned Data** [13] Combination of both horizontally and vertically partitioned data (see Figure 3.4).

|           | $attr_1$    | $attr_2$    | $\dots$ | $attr_m$    |
|-----------|-------------|-------------|---------|-------------|
| $d_1$     | $d_{1,1}$   | $d_{1,2}$   | $\dots$ | $d_{1,m}$   |
| $\dots$   | $\dots$     | $\dots$     | $\dots$ | $\dots$     |
| $d_l$     | $d_{l,1}$   | $d_{l,2}$   | $\dots$ | $d_{l,m}$   |
| $d_{l+1}$ | $d_{l+1,1}$ | $d_{l+1,2}$ | $\dots$ | $d_{l+1,m}$ |
| $\dots$   | $\dots$     | $\dots$     | $\dots$ | $\dots$     |
| $d_n$     | $d_{n,1}$   | $d_{n,2}$   | $\dots$ | $d_{n,m}$   |

Data owned by Alice

Data owned by Bob

Figure 3.5: Horizontally Partitioned Data.

### 3.4 Millionaire's protocol:

This protocol is used to compare two numbers owned by two different parties and decides which number is larger in privately. In 1982, Yao [21] introduced this protocol and popularly known as Yao's Millionaire protocol. In general, two multi millionaire parties Alice and Bob

|       | $attr_1$  | ... | $attr_l$  | $attr_{l+1}$ | ... | $attr_m$  |
|-------|-----------|-----|-----------|--------------|-----|-----------|
| $d_1$ | $d_{1,1}$ | ... | $d_{1,l}$ | $d_{1,l+1}$  | ... | $d_{1,m}$ |
| $d_2$ | $d_{2,1}$ | ... | $d_{2,l}$ | $d_{2,l+1}$  | ... | $d_{2,m}$ |
| ...   | ...       | ... | ...       | ...          | ... | ...       |
| $d_n$ | $d_{n,1}$ | ... | $d_{n,l}$ | $d_{n,l+1}$  | ... | $d_{n,m}$ |



 Data owned by Alice  Data owned by Bob

Figure 3.6: Vertically Partitioned Data.

|       | $attr_1$  | $attr_2$  | $attr_3$  | $attr_4$  | = |       | $attr_1$  | $attr_2$  | + |       | $attr_3$  | $attr_4$  |
|-------|-----------|-----------|-----------|-----------|---|-------|-----------|-----------|---|-------|-----------|-----------|
| $d_1$ | $d_{1,1}$ | $d_{1,2}$ | $d_{1,3}$ | $d_{1,4}$ |   | $d_1$ | $d_{1,1}$ | $d_{1,2}$ |   | $d_1$ | $d_{1,3}$ | $d_{1,4}$ |
| $d_2$ | $d_{2,1}$ | $d_{2,2}$ | $d_{2,3}$ | $d_{2,4}$ |   | $d_2$ | $d_{2,1}$ | $d_{2,2}$ |   | $d_2$ | $d_{2,3}$ | $d_{2,4}$ |



 Data owned by Alice  Data owned by Bob

Figure 3.7: Arbitrarily Partitioned Data.

want to get the information about who is richer compared to other, but here the constraint is that they should not exchange their individual information with others.

There are many solutions proposed for this protocol out of which Cachin's [22] method is used in our algorithm which is done on the basis of  $\phi$ -hiding supposition. Communication Complexity for this method is  $O(n)$ . Here,  $n$  represents the number of bits of each number from two parties.

### 3.5 Secure scalar product protocol:

This protocol is used to get the scalar product of two vectors  $A$  and  $B$  securely where Alice private vector is defined as  $A=(A_1,A_2,\dots,A_n)$  and Bob's vector is defined as  $(B_1,B_2,\dots,B_n)$  and their scalar product  $A \cdot B = \sum_{i=1}^n A_i \cdot B_i$ . Goethals *et al.* [2] has proposed very well secured private homomorphic dot product for secure computation. In this algorithm the same protocol is used while finding scalar product of two private vectors. This protocol is given in Algorithm 1.

This protocol is solely based on the method of homomorphic encryption. Here, the

---

**Algorithm 1:** Private homomorphic SSP protocol

---

**Input :** Two private vectors  $A, B \in Z_u^N$  from two parties.

**Output:** Results  $X_A + X_B \equiv A \cdot B \pmod n$  //for large value of  $n$

- 1 Initial step, Party Alice does:
  - 2     Generate a pair of Public and Private key  $(P_k, SK_k)$ .
  - 3     Send Public key  $p_k$  to party Bob.
  - 4 Alice does from  $j=1$  to  $N$
  - 5     Generate a new random string  $s_j$ .
  - 6     Send cipher text  $C_j = Enc_{A_j, s_j}$  to Party Bob
  - 7 Bob does:
  - 8     Set the value  $v \leftarrow \prod_{j=1}^N C_j^{B_j}$
  - 9     Generates a random nonce  $r'$  and a random text message  $X_B$ .
  - 10 Generate  $v' = Enc_{p_k}(-X_B, r')$  and send it to Alice.
  - 11 Alice Does: Computes  $X_A = Dec_{SK_k}(v') = x \cdot y - X_B$ .
- 

cryptosystem used is semantically secure. We can say this protocol is highly secure provided parties are honest while communicating to each other. Since it is secure, Party Bob can see  $N$  random cipher texts which are generated by Alice.

Consider a scenario where two vectors  $x_1$  and  $x_2$  of Bob are shared to Alice, then Alice chooses either  $x_1$  or  $x_2$  as  $x$ , say  $x=x_b$ . Even after having these vectors, Bob can not get significant amount of information about alice's inputs by applying several protocols of polynomial number. He may get some amount of information about  $x_b$  which won't help him in predicting  $b$ . At Alice'e end, she can only see some encryption which is done randomly such as  $x \cdot y - X_B$ . Since party Alice is having keys, she decrypts this kind of messages. Hence no other information is known to Alice.



## Chapter 4

# Secure OPTICS Algorithm

### 4.1 Problem Statement

Consider a database represented as DB which consists of n tuples.  $DB = \{t_1, t_2, \dots, t_n\}$ . Each and every record is explained by some values with an object of m attributes and it is represented as  $t_i$  by  $(B_{i_1}, B_{i_2}, \dots, B_{i_m})$ .

To simplify, we assumed that the data is distributed between only two parties namely Alice and Bob. Alice's data is denoted as  $D_A$  and Bob's data is denoted as  $D_B$ , in a way that union of  $D_B$  is equal to DB, i.e,  $DB = D_A \cup D_B$ . Our algorithm and other partitioned data clustering techniques are used to perform clustering on  $D_A$  and  $D_B$  such that Alice should not have the knowledge of  $D_B$  and Bob should not have the knowledge of  $D_A$ .

$D_A$  and  $D_B$  on both horizontally and vertically partitioned datasets are illustrated below:

#### 4.1.1 Vertically partitioned database

Alice's data is denoted as  $D_A = \{r_{A_1}, r_{A_2}, \dots, r_{A_n}\}$  and Bob's data is denoted by  $D_B = \{r_{B_1}, r_{B_2}, \dots, r_{B_n}\}$  For each and every record  $r_j$ , Alice is having records with k attributes,  $r_{A_j} = (A_{j_1}, A_{j_2}, \dots, A_{j_k})$  and Bob is having records with m-k attributes represented as  $r_{B_j} = (A_{j_{k+1}}, A_{j_{k+2}}, \dots, A_{j_m})$ . And,  $\{r_j\} = r_{A_j} \cup r_{B_j}$

#### 4.1.2 Horizontally partitioned database

Alice's share is denoted by  $D_A = \{r_{A_1}, r_{A_2}, \dots, r_{A_k}\}$  and Bob's data is denoted by  $D_B = \{r_{B_1}, r_{B_2}, \dots, r_{B_l}\}$  in a way that  $l = n - k$ . All records in two parties databases  $r_{A_j}, r_{B_j}$  are explained by the records of m attributes such as  $(A_{j_1}, A_{j_2}, \dots, A_{j_m})$ .

### 4.2 Protocols

In our actual algorithm, we have used some protocols which are illustrated in this section. These protocol's main purpose is to get the information whether two given input points are neighbor to each other or not in a secured way. Generally there are two such ways to design

privacy preserving algorithms. Coming to the first approach, a Trusted Third Party (TTP) involves in calculation part. With the help of this TTP one can find any calculation by sending the whole dataset to the TTP, and applying the intended clustering algorithm, then finally distributing the results to all the participating parties. Another approach is, designing privacy preserving algorithms with the help of some standard protocols form the literature of multi-party computation. i.e, in the absence of TTP.

### Solutions in the presence Trusted Third Party(TTP).

In general, if Trusted Third Party is present then these proposed protocols are very well secure provided TTP is honest and so we can easily calculate the neighbors of any object in the database with the help of these protocols securely. Computation complexity is so less in these type of protocols where communication complexity is more because we have to send whole datasets to TTP, then TTP has to forward results to participated parties which leads to more communication complexity.

**Protocol 1.** Party Alice's private input is represented as  $\{P_{A_1}, P_{A_2}, \dots, P_{A_n}\}$  and party Bob's private input is represented as  $\{P_{B_1}, P_{B_2}, \dots, P_{B_n}\}$ . Parties Alice and Bob need to find whether  $\{(P_{A_1} + P_{B_1}) \leq Eps^2\}, \{(P_{A_2} + P_{B_2}) \leq Eps^2\}, \dots, \{(P_{A_n} + P_{B_n}) \leq Eps^2\}$

The constraint in this situation is that, at the end of the protocol, party Alice should not have the knowledge of  $\{P_{B_1}, P_{B_2}, \dots, P_{B_n}\}$  and party Bob should not have the knowledge of  $\{P_{A_1}, P_{A_2}, \dots, P_{A_n}\}$ . Additionally, Trusted Third Party is available.

- Alice does:

- Create a shared key with TTP  $SK_{A_1,TTP}, SK_{A_2,TTP}, \dots, SK_{A_n,TTP}$
- Calculate  $\{Q_{A_1}, Q_{A_2}, \dots, Q_{A_n}\} = \{(P_{A_1} + SK_{A_1,TTP}), (P_{A_2} + SK_{A_2,TTP}), \dots, (P_{A_n} + SK_{A_n,TTP})\}$  and sends  $\{Q_{A_1}, Q_{A_2}, \dots, Q_{A_n}\}$  to Bob.

- Bob does:

- Create a shared key with TTP  $\{SK_{B_1,TTP}, SK_{B_2,TTP}, \dots, SK_{B_n,TTP}\}$
- Compute  $Q_{B_1}, Q_{B_2}, \dots, Q_{B_n} = (Q_{A_1} + P_{B_1} + SK_{B_1,TTP}), (Q_{A_2} + P_{B_2} + SK_{B_2,TTP}), \dots, (Q_{A_n} + P_{B_n} + SK_{B_n,TTP})$  and sends  $Q_{B_1}, Q_{B_2}, \dots, Q_{B_n}$  to Trusted Third Party.

- Trusted Third Party does:

- Computes  $\{resu_1, resu_2, \dots, resu_n\} = \{(Q_{B_1} - SK_{A_1,TTP} - SK_{B_1,TTP}), (Q_{B_2} - SK_{A_2,TTP} - SK_{B_2,TTP}), \dots, (Q_{B_n} - SK_{A_n,TTP} - SK_{B_n,TTP})\}$
- For i from 1 to n do  
     If  $(resu_i \leq Eps^2)$  then  $result_i = 1$   
     else  $result_i = 0$
- Send result array to Alice and Bob.

**Lemma 1:** Protocol 1 has complexity as 6 messages in communication form which are transferred to and from TTP, and a very negligible computational complexity which involves some basic operations like addition and subtraction.

**Protocol 2.** Party Alice's private input is represented as  $\{P_{A_1}, P_{A_2}, \dots, P_{A_n}\}$  and party Bob's private input is represented as  $\{Q_{B_1}, Q_{B_2}, \dots, Q_{B_n}\}$ . Alice and Bob need to know whether  $\{(P_{A_1} - Q_{B_1})^2 + (P_{A_2} - Q_{B_2})^2 + \dots + (P_{A_n} - Q_{B_n})^2\} \leq Eps^2$  or not. The constraint in this situation is that, at the end of the protocol, party Alice should not have the knowledge of  $\{Q_{B_1}, Q_{B_2}, \dots, Q_{B_n}\}$  and party Bob should not have the knowledge of  $\{P_{A_1}, P_{A_2}, \dots, P_{A_n}\}$ . Additionally, Trusted Third Party is available.

- Alice does:
  - Create a shared key with TTP  $\{SK_{A_1,TTP}, SK_{A_2,TTP}, \dots, SK_{A_n,TTP}\}$
  - Compute  $\{M_{A_1}, M_{A_2}, \dots, M_{A_n}\} = \{\{P_{A_1} + SK_{A_1,TTP}\}, \{P_{A_2} + SK_{A_2,TTP}\}, \dots, \{P_{A_n} + SK_{A_n,TTP}\}\}$  and sends  $\{M_{A_1}, M_{A_2}, \dots, M_{A_n}\}$  to Bob.
- Bob does:
  - create a shared key with TTP,  $\{SK_{B_1,TTP}, SK_{B_2,TTP}, \dots, SK_{B_n,TTP}\}$
  - Compute  $\{M_{B_1}, M_{B_2}, \dots, M_{B_n}\} = \{(M_{A_1} - Q_{B_1} + SK_{B_1,TTP}), (M_{A_2} - Q_{B_2} + SK_{B_2,TTP}), \dots, (M_{A_n} - Q_{B_n} + SK_{B_n,TTP})\}$  and forward  $\{M_{B_1}, M_{B_2}, \dots, M_{B_n}\}$  to TTP.
- Trusted Third Party does:
  - Calculate result =  $\{(M_{B_1} - SK_{A_1,TTP} - SK_{B_1,TTP})^2, (M_{B_2} - SK_{A_2,TTP} - SK_{B_2,TTP})^2, \dots, (M_{B_n} - SK_{A_n,TTP} - SK_{B_n,TTP})^2\}$
  - If  $\{result \leq Eps^2\}$  Then  
Send Yes to Alice and Bob.  
ELSE Send No to Alice and Bob.

**Lemma 2:** For protocol 2 also computation complexity is very less and negligible where as communication complexity is again six messages which is same as that of protocol 1. Here, complex operations are not there like encryption and decryption functions. So in the presence of trusted Third Party(TTP) the above protocols are simple and secure provided TTP is honest. These two protocols are basic protocols which does not use any intricate functions and does not involve in complex calculations, all they have to do is to forward to TTP and TTP does the rest of the work and finally returns output to both the parties.

**Solutions Without using TTP.** In real time, it's hard to find honest TTP all the time so there's a need to implement some protocols which works in the absence of TTP. To solve this kind of problems we are using some standard protocols like secure scalar product protocol which will be used to find scalar product between two vector securely and the millionaire's

[21] protocols which are described in previous sections. Contrary to above two protocols with TTP, here communication and computational complexities are more because of complex operations like encryption and decryption.

**Protocol 3.** Party Alice's private input is represented as  $\{P_{A_1}, P_{A_2}, \dots, P_{A_n}\}$  and party Bob's private input is represented as  $\{P_{B_1}, P_{B_2}, \dots, P_{B_n}\}$ . Parties Alice and Bob need to find whether  $\{(P_{A_1} + P_{B_1}) \leq Eps^2\}, \{(P_{A_2} + P_{B_2}) \leq Eps^2\}, \dots, \{(P_{A_n} + P_{B_n}) \leq Eps^2\}$ . The major condition in this situation is that, at the end of this protocol, party Alice should not have the knowledge of  $\{P_{B_1}, P_{B_2}, \dots, P_{B_n}\}$  and party Bob should not have the knowledge of  $P_{A_1}, P_{A_2}, \dots, P_{A_n}$ . Additionally, Trusted Third Party is not available.

- Initially a vector with n random numbers is generated by Bob  $RN_{B_1}, RN_{B_1} \dots RN_{B_n}$
- Alice constructs its vector as  $(\frac{P_{A_1}}{Eps^2}, 1), (\frac{P_{A_2}}{Eps^2}, 1), \dots, (\frac{P_{A_n}}{Eps^2}, 1)$   
 $(RN_{B_1}, RN_{B_1}(\frac{P_{B_1}}{Eps^2})), (RN_{B_2}, RN_{B_2}(\frac{P_{B_2}}{Eps^2})), \dots, (RN_{B_n}, RN_{B_n}(\frac{P_{B_n}}{Eps^2}))$ . Here, comes the need of secure scalar product protocol. Now both parties Alice and Bob runs **secure scalar product protocol** [23] and gets result as  $(RN_{B_1}(\frac{P_{A_1}+P_{B_1}}{Eps^2}), (RN_{B_2}(\frac{P_{A_2}+P_{B_2}}{Eps^2}), \dots, (RN_{B_n}(\frac{P_{A_n}+P_{B_n}}{Eps^2}))$  which is known to only Alice.
- Now Alice have output of previous step  $(RN_{B_1}(\frac{P_{A_1}+P_{B_1}}{Eps^2}), (RN_{B_2}(\frac{P_{A_2}+P_{B_2}}{Eps^2}), \dots, (RN_{B_n}(\frac{P_{A_n}+P_{B_n}}{Eps^2}))$  and Bob has the vector of random values  $\{RN_{B_1}, RN_{B_1} \dots RN_{B_n}\}$
- Then both parties Alice and Bon uses the millionaire protocol [22] which will decide whether  $(RN_{B_i}(\frac{P_{A_i}+P_{B_i}}{Eps^2})) > RN_{B_i}$  or not for each value of i from 1 to n.
- If  $(RN_{B_i}(\frac{P_{A_i}+P_{B_i}}{Eps^2})) > RN_{B_i}$ , Then the verdict will be  $(P_{A_i} + P_{B_i}) > Eps^2$  else they decide  $(P_{A_i} + P_{B_i}) < Eps^2$

**Protocol 4.** Party Alice's private input is represented as  $\{P_{A_1}, P_{A_2}, \dots, P_{A_n}\}$  and party Bob's private input is represented as  $\{Q_{B_1}, Q_{B_2}, \dots, Q_{B_n}\}$ . Alice and Bob need to know whether  $\{(P_{A_1} - Q_{B_1})^2 + (P_{A_2} - Q_{B_2})^2 + \dots + (P_{A_n} - Q_{B_n})^2\} \leq Eps^2$  or not. The constraint in this situation is that, at the end of the protocol, party Alice should not have the knowledge of  $\{Q_{B_1}, Q_{B_2}, \dots, Q_{B_n}\}$  and party Bob should not have the knowledge of  $\{P_{A_1}, P_{A_2}, \dots, P_{A_n}\}$ . Additionally, Trusted Third Party is not available.

- A number RN is generated by Bob randomly.
- A vector  $(\frac{P_{A_1}^2}{Eps^2}, \frac{1}{Eps^2}, \frac{-2P_{A_1}}{Eps^2}, \frac{P_{A_2}^2}{Eps^2}, \frac{1}{Eps^2}, \frac{-2P_{A_2}}{Eps^2}, \dots, \frac{P_{A_1}^2}{Eps^2}, \frac{1}{Eps^2}, \frac{-2P_{A_1}}{Eps^2})$  is constructed by Alice and Bob constructs  $RN, RNQ_{B_1}^2, RNQ_{B_1}, \dots, RN, RNQ_{B_n}^2, RNQ_{B_n}$ . Now Alice and Bob invokes the **secure scalar product protocol** [23] and gets  $RN(\frac{(P_{A_1}-Q_{B_1})^2+(P_{A_2}-Q_{B_2})^2 \dots+(P_{A_n}-Q_{B_n})^2}{Eps^2})$

3. Now Alice has above step output  $RN\left(\frac{(P_{A_1}-Q_{B_1})^2+(P_{A_2}-Q_{B_2})^2+\dots+(P_{A_n}-Q_{B_n})^2}{Eps^2}\right)$  and Bob has the random number RN.
4. Two parties Alice and Bob invokes cachin's millionaire protocol [22] to decide whether  $RN\left(\frac{(P_{A_1}-Q_{B_1})^2+(P_{A_2}-Q_{B_2})^2+\dots+(P_{A_n}-Q_{B_n})^2}{Eps^2}\right) > RN$  or not.
5. IF  $RN\left(\frac{(P_{A_1}-Q_{B_1})^2+(P_{A_2}-Q_{B_2})^2+\dots+(P_{A_n}-Q_{B_n})^2}{Eps^2}\right) > RN$  then these two parties decides that  $\{(P_{A_1} - Q_{B_1})^2 + (P_{A_2} - Q_{B_2})^2 + \dots + (P_{A_n} - Q_{B_n})^2\} > Eps^2$  else  $\{(P_{A_1} - Q_{B_1})^2 + (P_{A_2} - Q_{B_2})^2 + \dots + (P_{A_n} - Q_{B_n})^2\} \leq Eps^2$

### 4.3 Proposed Methods

**OPTICS for Vertically Partitioned Data:** OPTICS works in a similar fashion of DBSCAN except that it creates an ordering of the original database, along with storing two metrics called as core-distance and a suitable reachability distance for each and every object. This information is more than sufficient to extract the clusters w.r.t any other distance  $\epsilon'$ , this distance is much smaller than the original generating distance  $\epsilon$  from this order.

Pseudo code for basic OPTICS is illustrated in algorithm:

---

**Algorithm 2:** OPTICS (SetOfObjects,  $\epsilon$ , MinPts, OrderedFile)

---

```

1 InitialOrderedFile.open();// opens in read mode
2 FOR i= 1 TO size of database DO
3   Object := get an item from set of objects;
4   IF Object is not Processed THEN
5     ExpandClusterOrder(database with all points, point p,MinPts,  $\epsilon$  , OrderedFile in
      read mode) //expands cluster
6 OrderedFile.close();
7 END;
```

---

The OPTICS for vertically partitioned data algorithm is done in ExpandClusterOrder method which initiated by selecting an arbitrary object p, then computeDistance function is invoked which takes arguments as the number of attributes in database, selected data point, and records. The main use of this function is to find the distance between the selected arbitrary object and remaining objects at both parties Alice and Bob.

Now Parties Alice and Bob has distances  $dist_A$  and  $dist_B$  respectively. Our aim is to find  $dist_A + dist_B \leq Eps^2$ . Here, by applying either protocol 1(with TTP) or protocol 3(without TTP) we will get the solution. Then we can say whether the selected arbitrary point p can be marked as a core point or not. If that point p belongs to core point then a cluster is created

from 6 to 30 steps which is normal OPTICS algorithm. If that selected point is not defined as a core point then that point is treated as noise. This process is called until all the points in database are classified.

### OPTICS for Horizontally Partitioned Data:

Initially an arbitrary point  $p$  is selected at party Alice, then this party Alice starts finding all the nearest neighbors of chosen arbitrary point. In algorithm 5, initial two steps are used to select a point then from 3 to 7 steps, party Bob gets the neighbors of chosen point. To check whether the point is core point or not step 10 condition is used. Once the point is identified as core then a cluster is created by including all the points at both Alice and Bob which are density reachable from that point, this process carried out from steps 13 to 30.

Unlike in vertically partitioned data algorithm, here some points may be unclassified at party Bob, then last few steps are used to apply normal OPTICS algorithm.

## 4.4 Correctness Proof

This section is to show that secure OPTICS algorithm gives exact results as that of normal OPTICS algorithm. It returns same set of ordering of points and generates same clusters as that of normal non secure OPTICS algorithm on vertical or horizontal partitioned databases.

**Theorem 1.** The same set of ordering points and clusters are generated by Privacy preserving Secure OPTICS algorithm for vertically partitioned data as its OPTICS algorithm does with input database  $DB = DB_A \cup DB_B$

**Proof.** To start with, an arbitrary point  $p$  is chosen by normal OPTICS algorithm. Even in this OPTICS algorithm for vertically partitioned data(Algorithm 3 and Algorithm 4) algorithm also same process is followed. Once the point is chosen, to find the neighbors of this point in this algorithm protocol 1 and protocol 2 are used. In normal OPTICS algorithm to find the distance between two points  $x, y$  and to check whether they are neighbors or not we use the equation  $(y_{A_1} - x_{A_1})^2 + (y_{A_2} - x_{A_2})^2 + \dots + (y_{A_k} - x_{A_k})^2 + (y_{A_{k+1}} - x_{A_{k+1}})^2 + (y_{A_{k+2}} - x_{A_{k+2}})^2 + \dots + (y_{A_m} - x_{A_m})^2$ , where first  $k$  attributes are at Party Alice then next  $n-k$  attributes at Party Bob. But here, party Alice calculates  $distance_A = (y_{A_1} - x_{A_1})^2 + (y_{A_2} - x_{A_2})^2 + \dots + (y_{A_k} - x_{A_k})^2$  and then party Bob calculates  $distance_B = (y_{A_{k+1}} - x_{A_{k+1}})^2 + (y_{A_{k+2}} - x_{A_{k+2}})^2 + \dots + (y_{A_m} - x_{A_m})^2$ . All we need to find is whether  $distance_A + distance_B \leq Eps^2$ .

By using protocol 1, this can be done easily. We simply pass these two distance metrics to TTP and then TTP gives the intended output to both parties Alice and Bob without revealing their individual information to other parties. In protocol 3, this is done by using two standard protocols namely, millionaire's and secure scalar product. These two protocols are very well secured and already proved to be correct in [23] [22].

**Algorithm 3:** OPTICS algorithm for Vertically Partitioned Data

---

**Input** : point  $p$ , neighbor, clsID, Database DB, OrderedFile  
**Output:** DataBase DB with some classified data points

- 1 /\* First  $k$  attributes are owned by Alice, next  $m-k$  attributes are owned by Bob B of the database DB with  $n$  tuple. \*/
- 2 Select one objects from combined dataset  $p = p_A \cup p_B$ , this point should not be a classified point
- 3 Party Alice calculates:  $distance_A = \text{ComputeDistance}(p_A, k, n)$
- 4 Party Bob calculates:  $distance_B \leftarrow \text{ComputeDistance}(p_B, m-k, n)$
- 5 Invokes either protocol 1 or protocol 3 with Alice contains  $distance_A$ , and Bob contains  $distance_B$  by both parties Alice and Bob. Based on the output decides whether the object  $p$  is treated as a core point or noise object.
- 6 if  $p$  is core point then
  - 7 Assign seeds value as all the neighbors of point  $p$  which are retrieved by using protocols
  - 8 Assign clusterID to  $p$ .
  - 9 Assign  $\text{Object.Reachability}_{distance} := \text{UNDEFINED}$ ;
  - 10  $\text{point.setCoreDistance}(e, \text{neighbors}, \text{MinPts})$ ;
  - 11  $\text{presentObject.Processed} := \text{TRUE}$ ;  $\text{OrderedFile.write}(\text{Object})$ ; while seeds are not empty do
    - 12 assign presentp to first point in seeds
    - 13 Alice does :  $res_A \leftarrow \text{computeDistance}(\text{present}_{p_A}, k, n)$
    - 14 Bob does :  $res_B \leftarrow \text{computeDistance}(\text{present}_{p_B}, m-k, n)$
    - 15 Now run either protocol 1 or protocol 3 where party Alice owns  $res_A$  and party B owns  $res_B$  and save the returned result of either protocol 1 or protocol 3 into result
    - 16 if  $|result|$  is greater than  $\text{MinPts}$
    - 17 for  $i=1$  to  $|result|$  do
    - 18  $\text{resp} \leftarrow \text{res.get}(i)$
    - 19 if  $\text{resp.clusterID}$  is not not classified then
    - 20 append this resp to seeds
    - 21  $\text{DB.changeclusterId}(\text{resp}, \text{clusterId})$
    - 22 mark  $p$  as classified end
    - 23 else if  $\text{resp.clusterID}$  is Noise then
    - 24  $\text{DB.changeclId}(\text{resp}, \text{clusterId})$
    - 25 IF  $\text{presentObject.core}_{distance}$  is= UNDEFINED THEN
    - OrderSeeds.update(presentObejct,neighbors);
    - 26 end
    - 27 end
  - 28 end
  - 29 end
  - 30 end
  - 31 else Mark  $p$  as noise.

---

---

**Algorithm 4:** ComputeDistance function

---

**Input** : Object p, attributeCount, recordsCount, Dataset

**Output:** Distance

```
1 for j=1 to recordCount do
2   if(j ≠ p) Then
3     distancej = disance(j, p);
4     /* {distance(j, p) ← (jA1 - pA1)2 + .....(jAattributeCount - pAattributeCount)2} */
5 distance ← {distance1, distance2, ..., distanceattributeCount}
```

---

So the clusters formed and ordering points generated by the OPTICS algorithm are same as that of the OPTICS algorithm for the data which is partitioned vertically.

**Theorem 2.** The same set of ordering points and clusters are resulted by using Secure OPTICS algorithm for vertically partitioned data as its OPTICS algorithm does with input database DB which is combination of  $DB_A$  and  $DB_B$

**Proof.** To start with, an arbitrary point x is chosen by normal OPTICS algorithm. Even in this OPTICS algorithm for horizontally partitioned data(Algorithm 4) algorithm also same process is followed. In horizontally partitioned data, there are two scenarios. To find the neighbors of an arbitrary point x first we have to check whether the other point belongs to same party or not. If the other point is from same party then normal OPTICS algorithm is used to find the distance and then compared with MinPts finally decision will be made whether these two points are neighbors or not. But other scenario is that the point is from other party say Bob then protocol 2 and protocol 4 ar used to make the decision whether these two points are neighbors or not. In normal OPTICS algorithm to find the distance between two points x,y and to check whether they are neighbors or not we use the equation  $(y_{A_1} - x_{A_1})^2 + (y_{A_2} - x_{A_2})^2 + \dots + (y_{A_k} - x_{A_k})^2 + (y_{A_{k+1}} - x_{A_{k+1}})^2 + (y_{A_{k+2}} - x_{A_{k+2}})^2 + \dots + (y_{A_m} - x_{A_m})^2$ . Here if x and y are from same party then its easy to say whether they are neighbors are not with normal OPTICS algorithm. Suppose if x is from Alice's dataset and y is from Bob's dataset then protocol 2 or protocol 4 are used.

By using protocol 2, this can be done easily. We simply pass these two data points to TTP and then TTP gives the intended output to both parties Alice and Bob without revealing their individual information to other parties. In protocol 4, this is done by using two standard protocols namely,millionaire's and secure scalar product. These two protocols are very well secured and already proved to be correct in [23] [22]. So the clusters formed and ordering points generated by the OPTICS algorithm are same as that of the OPTICS algorithm for horizontally partitioned data.



**Algorithm 5:** OPTICS algorithm for Horizontally Partitioned Data

---

**Input** : point  $p$ , neighbor, clsID, Database DB, OrderedFile  
**Output:** DataBase DB with some classified data points

- 1 /\* First  $k$  records are owned by Alice, then next  $n-k$  records are owned by Bob\*/
- 2 Select an arbitrary point  $p$  at Party A which is unclassified.
- 3  $allneighbors_A \leftarrow$  neighbors of point  $p$  at party Alice.
- 4 for  $i = k+1$  to  $n$  do //Bob's data
- 5     Run either protocol 2(with TTP) or protocol 4(without TTP), which gives whether the point  $i$  is in the neighborhood region or not. Here, point  $p$  is at party Alice where as  $i$  point is at party Bob. Write into  $neighbor_B$  if the point  $i$  present in its neighborhood.
- 6 end of for
- 7 if total number of points in the neighborhood Alice and Bob are less than or equal to  $MinPts$  then
- 8     Alice does:  $seed_{s_A} = neighborhood_{s_A}$ , and update CLUSTERID to point  $p$
- 9     Bob does:  $seed_{s_B} = neighborhood_{s_B}$
- 10    Alice does:
- 11    while  $seed_{s_A}$  is not empty do
- 12       $presentp \leftarrow seed_{s_A}.getfirst()$
- 13      assign  $result_A$  to neighbors of  $presentp$  at party Alice
- 14      for  $i = (k+1)$  to  $n$  do
- 15        execute either protocol 2(with TTP) or protocol 4(without TTP) which tells whether the point  $i$  is present in the neighborhood of  $presentp$  or not where point  $p$  is at party Alice and point  $i$  is at Party Bob, and write into  $neighborhood_B$  if the point  $i$  is present in the neighborhood of point  $p$ .
- 16      end
- 17    if total number of points in the neighborhood Alice and Bob are greater than  $MinPts$  then do
- 18      for  $i = 1$  to number of points in  $result_A$  do
- 19         $result_{P_A} = result_A.get(i)$
- 20        if  $result_{P_A}.clusterID$  is not classified or  $result_{P_A}.clsID$  is Noise
- 21        then
- 22           $seed_{s_A}.append(result_A)$
- 23          Assign CLUSTERID(CLSID) to  $result_{P_A}$
- 24        end
- 25      end
- 26    end
- 27    end
- 28    Party Bob does: If  $seed_{s_B}$  is not empty then Repeat the steps from 10 to 26 by replacing Alice with Bob and Bob with Alice.
- 29 end
- 30 else mark the point  $p$  as noise.
- 31 Above process is repeated until all points at party Alice are clustered, if still Bob left with any unclassified points then normal OPTICS is used to get the clusters.

---

## **Chapter 5**

### **Conclusion**

As DBSCAN works for all types of data compared to other privacy preserving algorithms, OPTICS also works for all types of data by preserving the privacy of individuals, since OPTICS is inherited from DBSCAN and overcomes the disadvantage of DBSCAN as well. The resultant clusters in OPTICS are same as natural clusters. Distances are calculated by using the protocols proposed in this work, and they work for both vertically partitioned data and horizontally partitioned data. With the help of these protocols we present a OPTICS clustering algorithm for partitioned datasets. In future, there is a scope for extending this work to arbitrary partitioned datasets. And can also look at other privacy preserving algorithm which can be used for mixed types of datasets.

# References

- [1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise.” in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [2] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “Optics: ordering points to identify the clustering structure,” in *ACM Sigmod Record*, vol. 28, no. 2. ACM, 1999, pp. 49–60.
- [3] O. Goldreich, “Secure multi-party computation,” *Manuscript. Preliminary version*, pp. 86–97, 1998.
- [4] R. Agrawal and R. Srikant, “Privacy-preserving data mining,” in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 439–450.
- [5] L. Chang and I. Moskowitz, *A study of inference problems in distributed databases*. Springer, 2003.
- [6] S. Jha, L. Kruger, and P. McDaniel, “Privacy preserving clustering,” in *Computer Security—ESORICS 2005*. Springer, 2005, pp. 397–417.
- [7] R. Sibson, “Slink: an optimally efficient algorithm for the single-link cluster method,” *The Computer Journal*, vol. 16, no. 1, pp. 30–34, 1973.
- [8] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.
- [9] K. Hattori and Y. Torii, “Effective algorithms for the nearest neighbor method in the clustering problem,” *Pattern Recognition*, vol. 26, no. 5, pp. 741–746, 1993.
- [10] G. Sheikholeslami, S. Chatterjee, and A. Zhang, “Wavecluster: A multi-resolution clustering approach for very large spatial databases,” in *VLDB*, vol. 98, 1998, pp. 428–439.
- [11] A. Hinneburg and D. A. Keim, “An efficient approach to clustering in large multimedia databases with noise,” in *KDD*, vol. 98, 1998, pp. 58–65.
- [12] Y. Lindell and B. Pinkas, “Privacy preserving data mining,” in *Advances in Cryptology—CRYPTO 2000*. Springer, 2000, pp. 36–54.
- [13] G. Jagannathan and R. N. Wright, “Privacy-preserving distributed k-means clustering over arbitrarily partitioned data,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 593–599.
- [14] J. Vaidya and C. Clifton, “Privacy-preserving k-means clustering over vertically partitioned data,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 206–215.
- [15] P. Bunn and R. Ostrovsky, “Secure two-party k-means clustering,” in *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 486–497.

- 
- [16] K. A. Kumar and C. P. Rangan, "Privacy preserving dbscan algorithm for clustering," in *Advanced Data Mining and Applications*. Springer, 2007, pp. 57–68.
- [17] A. Amirbekyan and V. Estivill-Castro, "Privacy preserving dbscan for vertically partitioned data," in *Intelligence and Security Informatics*. Springer, 2006, pp. 141–153.
- [18] V. Estivill-Castro, "Private representative-based clustering for vertically partitioned data," in *Computer Science, 2004. ENC 2004. Proceedings of the Fifth Mexican International Conference in*. IEEE, 2004, pp. 160–167.
- [19] S. R. Oliveira and O. R. Zaiane, "Privacy preserving clustering by data transformation," *Journal of Information and Data Management*, vol. 1, no. 1, p. 37, 2010.
- [20] S. J. Rizvi and J. R. Haritsa, "Maintaining data privacy in association rule mining," in *Proceedings of the 28th international conference on Very Large Data Bases*. VLDB Endowment, 2002, pp. 682–693.
- [21] A. C. Yao, "Protocols for secure computations," in *Foundations of Computer Science, 1982. SFCS'82. 23rd Annual Symposium on*. IEEE, 1982, pp. 160–164.
- [22] C. Cachin, "Efficient private bidding and auctions with an oblivious third party," in *Proceedings of the 6th ACM conference on Computer and communications security*. ACM, 1999, pp. 120–127.
- [23] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikäinen, "On private scalar product computation for privacy-preserving data mining," in *Information Security and Cryptology–ICISC 2004*. Springer, 2004, pp. 104–120.

# Dissemination

## Conference Presentations

1. Janardhan Reddy kondra, Sambit Kumar mishra, Santhosh Kumar Bharti, "Honeypot-based Intrusion Detection System: A Performance Analysis." INDIACom-2016: 3rd International Conference on "Computing for Sustainable Global Development", 16th – 18th March, 2016.