RESEARCH ARTICLE OPEN ACCESS

# A Survey on Privacy Preserving Distributed Data Mining Based On Multi-Objective Optimization and Algorithmic Game Theory

R. Prabhavathi [1], S. Sasikala [2]
Research Scholar [1], Head of the Department [2], MCA., M.Phil
Department of Computer Science
Sree Saraswathi Thyagaraja College, Pollachi
Tamil Nadu - India

## ABSTRACT

Use of technology for data collection and analysis has seen an unprecedented growth in the last couple of decades. Individuals and organizations generate huge amount of data through everyday activities. This data is either centralized for pattern identification or mined in a distributed fashion for efficient knowledge discovery and collaborative computation. This, obviously, has raised serious concerns about privacy issues. The data mining community has responded to this challenge by developing a new breed of algorithms that are privacy preserving. Specifically, cryptographic techniques for secure multi-party function evaluation form the class of privacy preserving data mining algorithms for distributed computation environments. However, these algorithms require all participants in the distributed system to follow a monolithic privacy model and also make strong assumptions about the behavior of participating entities. These conditions do not necessarily hold true in practice. Therefore, most of the existing work in privacy preserving distributed data mining fails to serve the purpose when applied to large real-world distributed data mining applications. We develop a novel framework for privacy preserving distributed data mining that allows personalization of privacy requirements for individuals in a large distributed system and removes certain assumptions regarding participant behavior, thereby making the framework efficient and real-world adaptable.

***Keywords:-*** Privacy preserving, K-Means algorithm.

## I. INTRODUCTION

The problem of privacy preserving data mining has become more important in recent years because of the increasing ability to store personal data about users, and the increasing sophistication of data mining algorithms to leverage this information. A number of techniques such as randomization and k-anonymity have been suggested in recent years in order to perform privacy preserving data mining. Furthermore, the problem has been discussed in multiple communities such as the database community, the statistical disclosure control community and the cryptography community. In some cases, the different communities have explored parallel lines of work which are quite similar. This book will try to explore different topics from the perspective of different communities, and will try to give a fused idea of the work in different communities. The key directions in the field of privacy preserving data mining are as follows:

**(i) Privacy Preserving Data Publishing:** These techniques tend to study different transformation methods associated with privacy. These techniques include methods such as randomization, k-anonymity and l-diversity. Another related issue is how the perturbed data can be used in conjunction with classical data mining methods such as association rule mining. Other related problems include that of determining privacy-preserving methods to keep the underlying data useful (utility based methods), or the problem of studying the different definitions of privacy, and how they compare in terms of effectiveness in different scenarios.

**(ii) Changing the results of Data Mining Applications to preserve privacy:** In many cases, the results of data mining applications such as association rule or classification rule mining can compromise the privacy of the data. This has spawned a field of privacy in which the results of data mining algorithms such as association rule mining are modified in order to preserve the privacy of the data. A classic example of such techniques are association rule hiding methods, in which some of the association rules are suppressed in order to preserve privacy.

**(iii) Query Auditing:** Such methods are akin to the previous case of modifying the results of data mining algorithms. Here, we are either modifying or restricting the results of queries.

**(iv)** Cryptographic Methods for Distributed Privacy: In many cases, the data may be distributed across multiple sites, and the owners of the data across these different sites may wish to compute a common function. In such cases, a variety of cryptographic protocols may be used in order to communicate among the different sites, so that secure function computation is possible without revealing sensitive information.

**(v) Theoretical Challenges in High Dimensionality:** Real data sets are usually extremely high dimensional and this makes the process of privacy preservation extremely difficult both from a computational and effectiveness point of view. It has been shown that optimal k-anonymization is NP-hard. Furthermore, the technique is not even effective with increasing dimensionality, since the data can typically be combined with either public or background information to reveal the identity of the underlying record owners.

Use of technology for data collection has seen an unprecedented growth in the last couple of decades. Individuals and organizations generate huge amount of data through everyday activities. Decreasing storage and computation costs have enabled us to collect data on different aspects of people's lives such as their credit card transaction records, phone call and email lists, personal health information and web browsing habits. Security issues, government regulations, and corporate policies require most of this data to be scanned for important information such as terrorist activities, credit card fraud detection, cheaper communications, and even personalized shopping recommendations. Such analysis of private information often raises concerns regarding the privacy rights of individuals and organizations. The data mining community has responded to this challenge by developing a new breed of algorithms that analyze the data while paying attention to privacy issues.

## II. MOTIVATION

Considerable research in privacy preserving data mining is geared towards the census model where the data in a private database is sufficiently `distorted' to prevent leakage of individually identifiable information and then released to entrusted agencies for pattern mining. However, this set of solutions does not encompass all real world problems in data mining. Under many circumstances, data is collected at different locations and the data mining task requires the entire data to be centralized for identifying the global patterns.

For example, the US Department of Homeland Security funded PURSUIT project for privacy preserving distributed data integration and analysis aims at analyzing network traffic of different organizations to detect “macroscopic” patter ns for revealing common intrusion detection threats against those organizations. However, network traffic is usually privacy sensitive and no organization is generally willing to share their network traffic information with a third party. Similarly, different collaborative computing environments also require individuals to share their private data for different function computations. For example, peer-to-peer networks are a type of distributed systems that are characterized by huge size in terms of number of participating nodes and a lack of coordination among the nodes. Peer-to-peer systems are emerging as a choice of solution for a new breed of applications such as collaborative ranking, electronic commerce, social community formation, and directed information retrieval. Most of these applications require information integration among the nodes, some of which maybe privacy sensitive. The census model solutions do not work well in many of this emerging distributed privacy sensitive data mining applications. Cryptographic techniques for secure computations have been deployed for such privacy preserving distributed data mining problems.

Broadly speaking, cryptographic protocols compute functions over inputs provided by multiple parties without sharing the inputs with one another. The robustness of cryptographic protocols depends on the mutual trust placed on the parties. The cryptography literature assumes two types of participant behavior. A semi-honest party is curious and attempts to learn about others' private information during the computation, but never deviates from the protocol. Malicious participants deviate from the protocol, collude with others to send spurious messages to reveal others' private data. Protocols that are secure against malicious adversaries are computationally extremely expensive and therefore cannot be used in real-life for large scale data mining applications. Therefore, considerable effort has gone into developing secure protocols in the semi-honest adversary model. However, information integration in such multi-party distributed environments is often an interactive process guided by the dynamics of cooperation and competition among the par-ties. The behavior of these parties usually depends on their own objectives and is guided by whatever maximizes their personal benefits. If getting to know someone's private information is beneficial, then every self-interested party in the computation will try to get that information. Therefore, the assumption of semi-honest behavior falls apart in most real life distributed data mining applications.

Another important shortcoming of existing privacy preserving distributed data mining applications is the definition of a monolithic privacy model for all participants. Privacy is a social concept and, therefore, the privacy concerns of the different participating entities vary, as does their ability to protect their private data due to varying availability of resources. Therefore, in a distributed computing environment it is important that the par-ties be able to tailor their privacy definitions based on their requirements and yet be able to participate in a collaborative computing task. We develop a novel framework for personalized privacy in distributed data mining environments, paying careful attention to performance and real-world adaptability.

## III. PROBLEM ISSUES

We address the following problem. Consider a distributed computing environment consisting of nodes (parties) and connected via an underlying communication infrastructure. Each node has some data which is known only to itself. The nodes can exchange messages with any other node in the network. This research aims at answering the following question: “how can data mining tasks for extracting useful knowledge from the union of all the data be executed in the system such that different nodes participating in the collaborative computation (i) can specify their own privacy requirements without having to adhere to a monolithic privacy definition, (ii) can ensure that the required privacy is actually achieved without having to rely on unrealistic assumptions regarding the behavior of other parties and (iii) can compute the privacy preserving data mining results with an efficient use of resources.

## IV. RELATED WORKS

Advances in technology has enabled collection of a huge amount of data about individuals, groups or organizations from a wide variety of sources. This data collection and sub-sequent data mining often leads to a breach of privacy for the subject under consideration. Privacy preserving data mining is a growing field of research that tries to address the issue of privacy in the context of data mining. The objective of the field of privacy preserving data mining is to modify the data or the data mining protocols in such a way that the `privacy' of the subject is preserved while providing utility in terms of the mining results. When the private data is distributed across multiple data repositories owned by different parties, privacy preservation becomes a different kind of challenge due to personal preferences while doing distributed data mining.

## 4.1 DISTRIBUTED COMPUTING PRIMITIVES

We first define a distributed system and then present different types of algorithms for distributed systems.

### 4.1.1 Distributed Systems

A distributed system is one in which the failure of a computer we didn't even know existed can render our own computer unusable. Distributed computer system has several properties.

(i) Multiple processes: There is generally more than one concurrent process. There can be one or more than one process per node of the distributed system.

(ii) Common goal: Any distributed systems must have a common goal. The processes should collaborate to solve the same problem or task.

(iii) Inter-process communication: In a typical distributed system, each process performs some computation by itself and then communicates with other processes. The communication can be over a network using finite delay messages. The messages are transmitted across the communication channels.

(iv) Disjoint address space: Processes have disjoint address space. Shared-memory architectures are not considered distributed systems.

Mathematically, a distributed system can be represented as a graph G = (V, E), where V is the set of computers or machines or nodes and E is the set of edges or communication links connecting them. The messages are exchanged across the edges. It is generally assumed that the graph is connected i.e. for any two arbitrary nodes vi, vj $\in$ V, there exists a (possibly multi-hop) path from vi to vj. The set of one hop (immediate) neighbors of vi is known as the neighbor set and is denoted as i. Mathematically, it can be written as,

$$i = \{vj \in V \mid (v_i, v_j) \in E\}.$$

### 4.1.2 Types of Distributed Algorithms

Distributed algorithms can be categorized based on the type of communication protocol it uses for inter-process communication.

(i) Broadcast based Algorithms: Broadcasting is a communication protocol in which a message from a node is disseminated to all the nodes in the network. One way of achieving broadcast in networks in which there is no point to point connection among nodes is through flooding. In flooding, whenever a node receives a message, it forwards it to all its neighbors except the one from whom it received. As evident, there is lot of wasted resources and high load on the network since the same message can be transmitted many times along each link. Moreover, each node needs to process an overwhelming number of messages in order to identify and disregard the duplicates. The message complexity is O(|E|), since each edge sends a message once or more. The running time is proportional to the diameter of the network. A slightly more intelligent variant uses directional flooding - it sends messages only in one direction e.g. from lower to higher node identifier.

(ii) Convergecast Algorithms: In convergcast algorithms, the communication takes place on a spanning tree. Such a tree encompassing all the nodes can be easily constructed using a broadcast based spanning tree algorithm. Communication proceeds from the leaf up to the root of the tree. At each step, a node in the tree checks if it has received messages from all its children. If yes, it simply sends a message to its parent up the tree, else it simply waits. The parent does the same computation. The root finally receives a message containing information about the entire network. Similar to broadcast, this technique is also communication expensive: it requires O(|V|) messages since each node sends exactly one message. The running time is proportional to the depth of the tree which can be greater than the diameter of the network. However, once the tree is pre-computed, this technique is extremely simple.

(iii) Local Algorithms: Both the algorithm types discussed earlier suffer from one major drawback - the communication complexity is of the order of the size of the network. This is unacceptable for large networks such as peer-to-peer systems in which the size of the network typically ranges from thousands to millions of nodes. Local algorithms [1] are a different genre of algorithms in which the communication load at each node is either a small constant or sub-linear with respect to the network size, providing excellent scalability for the local algorithm. In a local algorithm, a node typically converges to the correct result by communicating with only a small fraction of nearby neighbors. Primarily for this reason, local algorithms exhibit high scalability.

## 4.2 DISTRIBUTED DATA MINING

Distributed data mining deals with the problem of data analysis in an environment with distributed data, computing nodes and users. This area has seen considerable research during the last decade. Data mining often requires massive amount of resources in storage space and computation time. If the data happens to be distributed at a number of different sites, then centralizing the data to a single storage location requires additional communication resources. Distributed data mining is a field of research that concentrates on developing efficient algorithms for mining of information from distributed data without centralizing it. Depending on how the data is distributed across the sites, distributed data mining algorithms can be divided into two categories:

(i) Algorithms for homogeneous data distribution: For this kind of data distribution, also known as the horizontally partitioned scenario, all attributes or features are observed at every site. However, the set of observations or tuples across the different sites differ.

(ii) Algorithms for heterogeneous data distribution: For this kind of data distribution, also known as the vertically partitioned scenario, each site has all tuples or rows, but only for a subset of the attributes for the overall data set.

### 4.2.1 Data Mining in GRID

Distributed data mining has seen a number of applications on the Grid infrastructure. Informally, a Grid can be defined as - "the ability, using a set of open standards and proto-cols, to gain access to applications and data, processing power, storage capacity and a vast array of other computing resources over the Internet" [2]. Grid computing has gained popularity as a distributed computing infrastructure for many highly computational-intensive tasks which are impossible to execute on a single computer. Grid applications rely on the computing and processing powers of possibly tens to thousands of dedicated or user-donated CPU cycles to perform a task. These users may be entities on the Internet or they may be part of a Grid consortium. The prospect of solving extremely challenging computational problems has found application of Grid computing in many research domains such as weather modeling, earthquake simulation, finance, biology (to study the effect of protein folding), chemistry and high-energy physics. Grid computing was popularized by the seminal work by Foster et al. [3] who are widely recognized as the "father of the modern grids" [4]. A Grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of resources distributed across multiple administrative domains, based on the resources' availability, capacity, performance, cost, and the users' quality-of-service requirements. A Grid infrastructure is not a completely asynchronous network. Since the main goal in Grid is to submit and execute user jobs, there exists centralized authority which monitors and ensures optimal resource allocations. Hoschek et al. [5] discusses the data management issues for Grid data mining. The goal of voluntary Grid computing is to ensure that jobs get executed in the scavenged CPU cycles in an optimal fashion without causing too much inconvenience to the CPU owner. Grid computing is essentially a heterogenous collection of different machines having access to distributed data, and so, researchers have explored the use of distributed data mining algorithms for information extraction from Grids. Talia and Skillicorn [6] argue that the Grid offers unique prospects for mining of large data sets due to its collaborative storage, bandwidth and computational resources. Cannataro et al. [7] address general issues in distributed data mining over the Grid. Several interesting on-going Grid projects involve data mining over the Grid. The Globus Consortium has developed the open source Globus Toolkit [8], to help researchers with Grid computing. Grid computing is closely related to peer-to-peer computing infrastructure in terms of data storage and computing power. However, one basic difference is the absence of any centralized authority in peer-to-peer systems.

#### 4.2.2 Distributed Stream Mining

Computation of complex functions over the union of multiple streams has been studied widely in the stream mining literature. Gibbons et al. [9] present the idea of doing coordinated sampling in order to compute simple functions such as the total number of ones in the union of two binary streams. They have developed a new sampling strategy to sample from the two streams and have shown that their sampling strategy can reduce the space requirement for such a computation from Ω(n) to log(n), where n is the size of the stream. Their technique can easily be extended to the scenario where there are more than two streams. The authors also point out that this method would work even if the stream is non-binary (with no change in space complexity). Much work has been done in the area of query processing on distributed data streams. Chen et al. [10] have developed a system `NiagaraCQ' which allows answering continuous queries in large scale systems such as the Internet. In such systems many of the queries are similar. So a lot of computation, communication and I/O resources can be saved by properly grouping the similar queries. NiagaraCQ achieves this goal by using a grouping scheme that is incremental. They use an adaptive regrouping scheme in order to find the optimal match between a new query and the group to which the query should be placed. If none of these matches, then a new query group is formed with this query.

#### 4.2.3 Data Mining in Ad-hoc Networks

Ad-hoc networks, as the name suggests, consists of a collection of light-weight (possibly mobile) battery-powered sensors capable of communicating via wireless links. Currently such networks are mainly used for data collection from hostile and uninhabited environments such as war fronts, deep seas, volcanos, outer space, and safety critical equipments. The data is usually collected in an offline fashion and shipped to the base station using wired or wireless sensor network. However, with the proliferation of network infrastructure and low maintenance cost, it seems that the next generation of sensor nodes will be able to communicate in an peer-to-peer fashion using the wireless ad-hoc links. It is generally agreed upon that for a sensor, the majority of the power is wasted in communicating with its neighbors. Therefore, these ad-hoc networks form an ideal test bed for communication-efficient distributed data mining algorithms. Optimal node placement in sensor networks is another active area of research. Krause et al. [11] developed a technique in which optimal sensor placement leads to maximization of information and minimization of communication cost. Ghiasi et al. [12] present a technique for logical clustering of the sensors for reducing the cost of data transfer and computation.

#### 4.2.4 Peer-to-Peer Data Mining

Peer-to-peer (P2P) networks are becoming increasingly popular for different applications that go beyond downloading music without paying for it. Social network applications, search and information retrieval, file storage, and certain sensor network applications are examples of popular P2P applications [13]. In many cases, the nodes or peers in such P2P networks are loosely coupled with no shared memory and no synchronization. In general, P2P networks can be viewed as a massive network of autonomous nodes with no central administrator site monitoring their activities. Therefore, data mining in P2P networks requires a different genre of algorithms which are highly scalable and communication efficient. In this section we discuss some techniques for distributed data mining in P2P environments and then discuss some desired properties of P2P data mining algorithms.P2P data mining is a comparatively new field of research. Recently, several data mining algorithms have been proposed in the literature for different mining tasks. These algorithms are either approximate or exact. Datta et al. [14] present an overview of this topic. Probabilistic approximation techniques sometimes rely on sampling either the data or the network nodes. Examples include clustering algorithms described in [15] and [16]. Gossip based algorithms rely on the properties of random walks on graphs to provide estimates of various data statistics. Kempe et al. [17] and Boyd et al. [18] have put forward important theories for development of gossip based algorithms. Deterministic approximation techniques transform the P2P data mining problem into an optimization problem and look for optimal results in the sometimes intractable search space using mathematical approximation. One such approximation is the variational approximation technique proposed by Jordan and Jaakkola [19, 20]. Mukherjee and Kargupta [21] extended the variational approximation techniques for distributed inferencing in sensor networks.

### 4.3 PRIVACY PRESERVING DATA MINING

The area of privacy preserving data mining has been extensively studied by the data mining community. We classify privacy preserving data mining algorithms into three categories:

(i) Data distortion based privacy: These algorithms aim at distorting the original private data, when released, do not divulge any individually identifiable information.

(ii) Cryptography based privacy: Cryptographic protocols are called private when their execution does not reveal any additional information about the involved parties' data, other than what is computed as a result of the protocol execution.

(iii) Output perturbation based privacy: Output perturbation techniques discuss privacy with respect to the information released as a result of querying a statistical database by some external entity.

Privacy preserving data mining as a field has been hugely influenced by the research in statistical disclosure control.

Statistical Disclosure Control: Statistical disclosure control is a field of research that concentrates on how to provide summary statistical information on a statistical database without disclosing individual's confidential data. The privacy issues in such a scenario occur when the summary statistics are computed on the data of very few individuals or when the data of most individuals in the database are identical. Adam and Wortmann [22] provide an extensive review of the security control methods for statistical databases. Statistical disclosure control approaches suggested in the literature are classified into four general groups: conceptual, query restriction, output perturbation and data perturbation. Two models are based on the conceptual approach for disclosure control. The conceptual model [23] provides a framework for investigating the security from the development of the schema to the implementation at the data-model level. The lattice model [24] constitutes a framework for data represented in a tabular form at different levels of aggregation. Disclosure control methods that are based on the query restriction approach provide protection through the following measures [25]: restricting the query set size, controlling the overlap among successive queries and making cells of small size inaccessible to users in the tabular data representation. The data perturbation approach introduces noise into the database and transforms it into a different representation. The methods based on the data perturbation techniques either are probability distribution based or fixed data perturbation based. In the former, a database is considered to be a sample from a population with a given probability distribution and the security control method replaces the original database with another sample from the same population or by the distribution itself. In the latter, the values of the attributes in the database are perturbed and replaced before answering any queries. The output perturbation approach perturbs the answer to user queries while leaving the data in the database unchanged.

### 4.4 DATA DISTORTION BASED PRIVACY

In data distortion techniques, some transformation is usually applied on the data for privacy preservation. Examples of such transformations include adding noise to the data or suppressing certain values and reducing the granularity of the data. It should be noted here that there is a tradeoff between the privacy achieved and the utility of the data mining results. We divide the literature on data distortion based privacy into the following categories: (i) data perturbation, (ii) data micro aggregation, (iii) data swapping, and (iv) data anonymization.

### 4.4.1 Data Perturbation

The data distortion based privacy preservation techniques aim at modifying the private data values by adding additive or multiplicative noise drawn from a probability distribution to the data values. Quantification of privacy is a very import ant aspect in understanding the effectiveness of a technique as a privacy preserving method. There are several quantifications of privacy in the literature of data perturbation based privacy preserving data mining. Agrawal and Srikant [26] said that if the real value can be estimated with c% confidence to be in the range $(\alpha_1, \alpha_2)$, then the interval width $(\alpha_1, \alpha_2)$ is the amount of privacy protection provided by the randomization algorithm. However, this definition does not take into account the initial data distribution. An alternative definition proposed in [27] says that privacy can be quantified by the expression h(A), where h(A) is the differential entropy of a random variable A since it takes into account the inherent uncertainty in the data value. A number of quantification issues in the measurement of privacy breaches has also been discussed by Evfimievski [28]. Kargupta et al. [29] proposed a random matrix based spectral filtering algorithm for reconstructing the private data from additively perturbed data, thereby questioning the privacy guarantees provided by additive perturbation. Later, Guo and Wu [30] provided theoretical bounds on the reconstruction error from spectral filtering and singular value decomposition based reconstruction techniques. With the identification of the fact that the reconstruction gets better with higher correlation among the actual data points, Huang et al. [31] proposed a modified additive perturbation algorithm where the random noise added to the data has similar correlation as the actual data.

### 4.4.2 Data Micro aggregation

To obtain micro aggregates in a data set with n records, these are combined to form g groups each of size at least k. For each attribute, the average value over each group is computed and is used to replace each of the original averaged values. It is a popular approach for protecting the privacy of the confidential attributes in statistical databases. For univariate confidentiality in attributes, the confidential attribute is sorted for creating the groups [32]. For multivariate micro aggregation, confidential attributes are grouped using a clustering technique [33]. The optimal k-partition, from the information loss point of view, is defined to be the one that maximizes homogeneity wi thin a group: the higher that homogeneity, the lower the information loss, since micro aggregation replaces values in a group by the group centroid. Obviously, in the extreme case of all identical values, this can lead to a privacy breach.

### 4.4.3 Data Swapping

Other than adding or multiplying noise to the data, another approach to preserve privacy is to swap data values across records in a database, also known as data swapping [34]. This method preserves the marginal’s of individual attributes of the data and is therefore, very useful for privacy preserving aggregate computations. This technique does not follow the general principle of randomization which allows the value of a record to be perturbed independently of the other records. Therefore, this technique can be used in combination with other frameworks, as long as the swapping process is designed to preserve the definitions of privacy for that model.

### 4.4.4 Data Anonymization

Data anonymization is a privacy preserving technique addressing some of the limitations of randomization. In anonymization algorithms, the granularity of representation is lowered by generalization and suppression so that individually identifiable information is absent in the released database. In generalization, the attribute values are generalized to a range of acceptable values while in suppression the attribute value is deleted from the database to avoid identification of individuals. The most popular anonymization based privacy model called the k-anonymity was proposed by Sweeney [35]. k-anonymity states that each release of data must be such that every combination of values of released attributes that are externally available and, therefore, available for linking attacks on privacy, can be indistinctly matched to at least k respondents. The basic approach proposed in [35] is a greedy solution using domain generalization hierarchies of quasi-identifiers to build k-anonymous tables. Subsequently, there has been extensive research on the k-anonymity model of privacy. Meyerson and Williams [36] does a complexity analysis of the k-anonymization problem and states that optimal k-anonymity is an NP hard problem. The optimality is based on a cost metric defined on the quality of t he privacy achieved versus the utility of the released data. A number of heuristic methods have been proposed for optimally k-anonymizing a data set. One such method proposed by Bayardo and Agrawal [37] attempts to bound the running time of the search algorithm by presetting a desired quality of the output, which might not be the optimal quality.

### 4.4.5 Vulnerabilities of Data Distortion Techniques

There has been considerable research in analyzing the vulnerabilities of existing privacy preserving data mining techniques. Some of these efforts have assumed the role of an attacker and developed techniques for breaching privacy by estimating the original data from the perturbed data and any additional available prior knowledge. Additive data perturbation attacks use eigen analysis for filtering the protected data. The idea for techniques such as PCA [31] is that even after addition of random noise, the correlation structure in the original data can be estimated with considerable accuracy. This then leads to removal of the noise in such a way that it fits the aggregate correlation structure of the data. It has been shown that such noise removal results in prediction of values which are fairly close to their original values. Kargupta et al. [29] use results from matrix perturbation theory and spectral analysis of large random matrices to propose a filtering technique for random additive noise. They show that when the variance of noise is low and the original data has correlated components, then spectral filtering of the co variance matrix can recover the original data with considerable accuracy. A second kind of adversarial attack uses publicly available information. Assuming that the distribution of the perturbation is known, a maximum likelihood fit of the potential perturbation to a publicly available data creates a privacy breach. The higher the log likelihood fit, the greater the probability that the public record corresponds to a private data record. For multiplicative perturbation, privacy breach is in general more difficult if the attacker does not have prior knowledge of the data. However, with some prior knowledge, two kinds of attacks are possible [38]. In the known input-output attack, the adversary knows some linearly independent collection of records, and their mapping to the corresponding perturbed version and linear algebra techniques can be used to reverse engineer the nature of the privacy preserving transformation. For the known sample attack, the adversary has a collection of independent samples from the original data distribution and assumes that the perturbation matrix is orthogonal. Using this, he can replicate the behavior of the original data using eigen analysis techniques.

## V. MULTI-OBJECTIVE OPTIMIZATION BASED PERSONALIZED PRIVACY

Privacy preserving data mining is a relatively new field of research and the pioneering works in this area has shown that in most cases, privacy comes at a cost. Sometimes this cost is in terms of the amount of excess computation that needs to be performed to ensure privacy and sometimes it is additional communication for secure multi-party computation techniques. Other than requirement of additional resources, privacy also comes at the cost of utility in many situations. The quality of the data mining results is compromised due to different kinds of perturbation or anonymization techniques. Therefore, privacy preservation for data mining can be thought of as an optimization problem. The problem of utility based privacy preserving data mining was first studied formally by Kifer where the problem of dimensionality in the process of anonymizing data for privacy preservation was addressed by separately publishing marginal tables containing attributes which have utility, but were not as good in terms of privacy preservation. The approach is based on the idea that the generalization performed on the marginal tables and the actual tables do not need to be the same. The optimality is based on a cost metric defined on the quality of the privacy achieved versus the utility of the released data. A number of heuristic methods have been proposed to find the optimal anonymization of the given data. One such method proposed by Bayardo and Agrawal attempts to bind the running time of the search algorithm by presetting a desired quality of the output, which might not be the optimal quality. The algorithm assigns a penalty to each data record based on how many records in the transformed data set are indistinguishable from it. If an unsuppressed record falls into an induced equivalence class of size j, that record is assigned a penalty of j. If a record is suppressed, it is assigned a penalty of |D|, where |D| denotes the size of the data set D. If g denotes the anonymization function for a given k, then mathematically, the algorithm optimizes the objective function:

$$\mathrm{Cost}(g,k,D) = \sum_{\forall k s.t. |k| \geq k} |k|^2 + \sum_{\forall k s.t. |k| < k} |D||k|$$

where k is the set of equivalence classes of records in D. The first sum computes penalties for all non-suppressed records, the second for suppressed records. The utility measure in this approach is called the generalization height. Other measures of utility for optimal anonymization include size of the anonymized group for the ℓ-diversity approach and privacy information loss ratio. For randomization based privacy preservation, Zhu and Liu propose a metric based on the mutual information between the randomized and original data. They propose optimization of this metric for an optimal privacy utility combination for density estimation tasks on the data.

A different connotation of optimal privacy involves paying attention to the privacy requirements of individual data owners participating in the data mining task. A condensation based approach has been proposed for addressing variable constraints on the privacy of data tuples depending on the data owner’s preferences. This technique constructs groups of non-homogeneous size from the data, such that it is guaranteed that each record lies in a group whose size is at least equal to its anonymity level. Subsequently, pseudo-data are generated from each group so as to create a synthetic data set with the same aggregate distribution as the original data. A comparatively recent work on personalized privacy based on k-anonymization has been proposed by Xiao and Tao. In this approach the entire data set is divided into domains in the form of an ontological graph structure and the individuals can specify the level of privacy required for the sensitive attributes by specifying the node level in the generalization hierarchy. The authors propose a greedy algorithm to obtain the optimal privacy for different sensitive attributes depending on the individual's preference. Although there has been some research in the area of optimization and privacy, it has never been studied in the light of distributed data mining.

### Distributed Averaging Algorithm (DAvg)

```
Input of node $v_k$:
  Convergence rate ρ, local data $x^{(l)}_{i,k}$ round
Initialization:
  Set $z^{(0)}_{i,k} \leftarrow x^{(l)}_{i,k}$;
  Set round ← 1;
On receiving a message $z^{(t)}_{i,k'}$ from $v_{k'}$ :
  $z^{(t+1)}_{i,k} = z^{(t)}_{i,k} + \rho \sum_{a\in\Gamma_k}^{n} (z^{(t)}_{i,a} - z^{(t)}_{i,k})$
Send $z^{(t+1)}_{i,k}$ to all neighbors in $\Gamma_k$;
```

In distributed averaging, the objective is to compute the global average $\Delta^{(l)}_i = \frac{1}{n}\sum_{i=1}^{n} x^{(l)}_{i,k}$ of the lower bound (and similarly upper bound) of every constraint xi of x. $x^{(l)}_{i,k}$ is the lower bound of constraint xi for node vk and n is the size of the network. For convenience, we are going to refer to $\Delta^{(l)}_i$ as $\Delta_i$ through the rest of this section. In the naive solution, all nodes can exchange messages with every other node in the system to compute the correct average. However, this solution is highly synchronous and does not scale well for large distributed environments such as P2P networks. Distributed approaches include the iterative Laplacian based approach proposed by Mehyar et al., the LTI approach proposed by Scherber and Papadopoulos. The basic idea of all these approaches is to maintain the current estimate of $\Delta_i$ denoted by $z^{(t)}_i$ and exchange messages with its immediate neighbors to update $z^{(t)}_i$. As iteration $t \to \infty, z^{(t)}_i \to \infty$, i.e. the system asymptotically converges to the correct average. we adopt the distributed averaging algorithm (DAvg as shown in Algorithm-1) proposed by Scherber and Papadopoulos. The authors exploit the properties of the symmetric negative semi-definite connectivity matrix Ω to derive the update rule for asymptotic convergence which is $z^{(t)}_i = Wz^{(t-1)}_i$, where $z^{(t)}_i$ denotes a column vector of the estimates of all the nodes at time t, i.e. $z^{(t)}_i = [z^{(t)}_{i,1} z^{(t)}_{i,2} ... z^{(t)}_{i,n}]^T$ and W is a matrix used in first order linear transformation rules. At initialization, $z^{(0)}_i = x_i = [x^{(l)}_{i,1} x^{(l)}_{i,2} ... x^{(l)}_{i,n}]$.

In order for $z^{(t)}_i$ to converge $\Delta_i$, W must satisfy the following properties: (i)W.1 =WT .1 =  1, where 1 denotes a (n x 1) vector of all ones and (ii) the eigen values of W, λi when arranged in descending order are such that λ1 = 1 and |λi| < 1 for i > 1. If  is a symmetric matrix, then W can be constructed from as follows: $W = I + \rho\Omega$. Here I is the (n x n) identity matrix and ρ is a small number which determines the stability of the solution and the convergence rate. Typically, ρ can be set to $\frac{1}{\max_i |\Omega_{ii}|}$. For updating from time t to t + 1, the update rule for any node dk can be written as

$$z^{(t+1)}_{i,k} = z^{(t)}_{i,k} + \rho \sum_{a\in\Gamma_k}^{n} (z^{(t)}_{i,a} - z^{(t)}_{i,k}) .$$

## VI. CONCLUSION

We have presented a multi-objective optimization framework for privacy protection in a multi-party environment. Since privacy is intricately related to one's preferences such as data, computing power, etc., we feel a party should be given the freedom to specify its own privacy requirement. Therefore, a uniform model and privacy constraint for each node in the network is not desirable; we need a personalized solution for each node. To achieve this, we have proposed a multi-objective optimization based frame-work where each node may have a different set of constraints signifying its desired privacy and cost. The Pareto optimal solution set provides the privacy/cost tradeoff for each node. To ensure that each node generates a solution in the same Pareto optimal set, which is important for the distributed data mining algorithm to work correctly, we take an average over the constraints of all the nodes. For this purpose, we use an existing asynchronous distributed averaging protocol which, without centralizing all the constraints, can generate a "global" constraint for the multi-objective optimization problem.

## REFERENCES

[1]. D. Parkes. ibundle: An efficient ascending price bundle auction. In Proceedings of EC'99, pages 148–157, 1999.

[2]. What is Grid ? http://www.eu-degree.eu/DEGREE/General%\20questions/copy_of_what-is-grid.

[3]. I. Foster and C. Kesselman. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 2004.

[4]. D. Talia and D. Skillicorn. Mining Large Data Sets on Grids: Issues and Prospects. Computing and Informatics, 21(4):347–362, 2002.

[5]. W. Hoschek, F. Ja´en-Martınez, A. Samar, H. Stockinger, and K. Stockinger. Data Management in an International Data Grid Project. In Proceedings of the First International Workshop on Grid Computing, pages 77–90, London, UK, 2000.

[6]. D. Talia and D. Skillicorn. Mining Large Data Sets on Grids: Issues and Prospects. Computing and Informatics, 21(4):347–362, 2002.

[7]. M. Cannataro, A. Congiusta, A. Pugliese, D. Talia, and P. Trunfio. Distributed Data Mining on Grids: Services, Tools, and Applications. IEEE Transactions On Systems, Man, And Cybernetics Part B: Cybernetics, 34(6):2465–2451, 2004.

[8]. Globus Consortium. http://www.globus.org/.

[9]. P. Gibbons and S. Tirthapura. Estimating Simple Functions on the Union of Data Streams. In Proceedings of SPAA'01, pages 281–291, Crete, Greece, 2001.

[10]. J. Chen, D. DeWitt, F. Tian, and Y. Wang. Niagara CQ: a Scalable Continuous Query System for Internet Databases. In Proceedings of SIGMOD'00, pages 379–390, Dallas, Texas, 2000.

[11]. A. Krause, A. Singh, and C. Guestrin. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. J. Mach. Learn. Res., 9:235–284, 2008.

[12]. S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh. Optimal Energy Aware Clustering in Sensor Networks. Sensors, 2:258–269, 2002.

[13]. P2P Wikipedia. http://en.wikipedia.org/wiki/Peer-to-peer.

[14]. S. Datta, K. Bhaduri, C. Giannella, R. Wolff, and H. Kargupta. Distributed Data Mining in Peer-to-Peer Networks. IEEE Internet Computing, 10(4):18–26, 2006.

[15]. S. Bandyopadhyay, C. Giannella, U. Maulik, H. Kargupta, K. Liu, and S. Datta. Clustering Distributed Data Streams in Peer-to-Peer Environments. Information Science, 176(14):1952–1985, 2006.

[16]. S. Datta, C. Giannella, and H. Kargupta. K-Means Clustering over Large, Dynamic Networks. In Proceedings of SDM'06, pages 153–164, Maryland, 2006.

[17]. D. Kempe, A. Dobra, and J. Gehrke. Gossip based computation of aggregate information. In

Proc. of FOCS'03, Cambridge, MA, October 2003.

[18]. S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip Algorithms: Design, Analysis and Applications. In Proceedings Infocom'05, pages 1653–1664, Miami, March 2005.

[19]. T. Jakkola. Tutorial on Variational Approximation Methods. In Advanced Mean Field Methods: Theory and Practice, 2000.

[20]. M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An Introduction to Variational Methods for Graphical Models. Machine Learning, 37(2):183–233, November 1999.

[21]. S. Mukherjee and H. Kargupta. Distributed Probabilistic Inferencing in Sensor Networks using Variational Approximation. J. Parallel Distrib. Comput., 68(1):78–92, 2008.

[22]. N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: a comparative study. ACM Computing Surveys, 21(4):515–556, 1989.

[23]. F. Y. Chin and G. Ozsoyoglu. Auditing and inference control in statistical databases. IEEE Transactions on Software Engineering, 8(6):574–582, 1982.

[24]. D. E. Denning and J. Schlorer. Inference controls for statistical databases. IEEE Computer, 16(7):69–82, 1983.

[25]. D. E. Denning. Cryptography and data security. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1982.

[26]. R. Agrawal and R. Srikant. Privacy preserving data mining. In Proceedings of the SIGMOD'00, pages 439–450, Dallas, TX, May 2000.

[27]. D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In Proceedings of PODS'01, pages 247–255, Santa Barbara, CA, 2001.

[28]. A. Evfimevski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In Proceedings of SIGMOD/PODS'03, San Diego, CA, June 2003.

[29]. H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In Proceedings of ICDM'03, pages 99–106, Melbourne, FL, November 2003.

[30]. S. Guo and X. Wu. On the use of spectral filtering for privacy preserving data mining. In Proceedings of SAC'06, pages 622–626, Dijon, France, April 2006.

[31]. Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In Proceedings of SIGMOD'05, pages 37–48, Baltimroe, MD, June 2005.

[32]. S. L. Hansen and S. Mukherjee. A polynomial algorithm for optimal univariate micro aggregation. IEEE Transactions on Knowledge and Data Engineering, 15(4):1043–1044, 2003.

[33]. J. Domingo-Ferrer and J. M. Mateo Sanz. Practical data-oriented micro aggregation for statistical disclosure control. IEEE Transactions on Knowledge and Data Engineering, 14(1):189–201, 2002.

[34]. S. E. Fienberg and J. McIntyre. Data swapping: Variations on a theme by dale. Technical report, National Institute of Statistical Sciences, Research Triangle Park, NC, 2003.

[35]. L. Sweeney. k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10(5):557–570, 2002.

[36]. A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In Proceedings of PODS'04, pages 223–228, New York, NY, USA, 2004. ACM.

[37]. R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In Proceedings of ICDE'05, pages 217–228, Washington, DC, USA, 2005. IEEE Computer Society.

[38]. K. Liu, C. Giannella, and H. Kargupta. An attacker's view of distance preserving maps for privacy preserving data mining. In Proceedings of PKDD'06, Berlin, Ger-many, September 2006.

[39]. A. C. Yao. How to Generate and Exchange Secrets (Extended Abstract). In FOCS, pages 162–167, 1986.