International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org, editorijaiem@gmail.com
Volume 2, Issue 7, July 2013 ISSN 2319 - 4847

Distributed Data Mining Privacy by Decomposition (DDMPD) with Naive Bayes Classifier and Genetic Algorithm

Lambodar Jena[1], Narendra Kumar Kamila[2]

Department of Computer science & Engineering,
[1]Gandhi Engineering College, Bhubaneswar.
[2]C.V.Raman college of Engineering, Bhubaneswar.

ABSTRACT

Publishing data about individuals without revealing sensitive information about them is an important problem. Distributed data mining applications, such as those dealing with health care, finance, counter-terrorism and homeland defence, use sensitive data from distributed databases held by different parties. This comes into direct conflict with an individual's need and right to privacy. It is thus of great importance to develop adequate security techniques for protecting privacy of individual values used for data mining. Here, we study how to maintain privacy in distributed mining of frequent itemsets. That is, we study how two (or more) parties can find frequent itemsets in a distributed database without revealing each party's portion of the data to the other. In this paper, we consider privacy-preserving naive Bayes classifier for horizontally partitioned distributed data and propose a two-party protocol and a multi-party protocol to achieve it. By classification accuracy and k-anonymity constraints, the proposed data mining privacy by decomposition (DMPD) method uses a genetic algorithm to search for optimal feature set partitioning. Multiobjective optimization methods are used to examine the tradeoff between privacy and predictive performance.

Keywords- Distributed database, privacy, data mining, classification, k-anonymity

1. INTRODUCTION

Information sharing is a vital building block for today's business world. Data mining techniques have been developed successfully to extract knowledge in order to support a variety of domains—marketing, weather fore- casting, medical diagnosis, and national security. But it is still a challenge to mine certain kinds of data without violating the data owners' privacy. As data mining become more pervasive, privacy concerns are increasing[1].

Distributed data mining is a process to extract globally interesting associations, classifiers, clusters, and other patterns from distributed data [2], where data can be partitioned into many parts either vertically or horizontally [3]. Vertical partition of data means that information about the same set of entities are distributed on different sites. For example, banks collect financial transaction information while IRS collects tax information. Horizontal partition of data means that the same set of information about different entities are distributed on different sites. For example, different hospitals collect the same type of patient data.

Distributed data mining can be classed into two categories [4]. The first is server-to-server where data are distributed across several servers. The second is client- to-server where data reside on each client while a server or a data miner performs mining tasks on the aggregate data from the clients.

A typical example in distributed data mining where privacy can be of great importance is in the field of medical research. Consider the case where a number of different hospitals wish to jointly mine their patient data, for the purpose of medical research. Privacy policy and law do not allow these hospitals from even pooling their data or revealing it to each other due to the confidentiality of patient records. Although hospitals are allowed to release data as long as the identifiers, such as name, address, etc., are removed, it is not safe enough because the re-identification attack can link different public databases to relocate the original subjects [5]. Consequently, providing privacy protection may be critical to the success of data collection, and to the success of the entire task of distributed data mining.

Privacy-preserving data mining was firstly realized by Agrawal and Srikant [6] and Lindell and Pinkas [7] independently in 2000. Since then, a number of privacy- preserving data mining algorithms and protocols have been proposed, such as those for association rule mining [8–12], clustering [13,14], naive Bayes classifiers [15–18], etc. So far, there have been two main approaches for privacy-preserving data mining as follows.

One is the randomization approach. The typical example is Agrawal–Srikant algorithm [6], in which data are randomized through the value class membership (values of an attribute are discretized in to intervals and the interval in which a value lies is returned instead of the original value)and the value distortion(a random value yi is added to each value xi of an attribute) and then the original data distribution is reconstructed by a Bayes approach. Based on the reconstructed distribution, decision trees can be induced [6]. The randomization approach has been used in association rule mining [8,12]. One drawback of the randomization approach is that privacy of data cannot be always fully preserved while achieving precision of data mining results [19].

Another is the cryptographic approach, e.g., secure multi-party computation(SMC),initially suggested by Yao in a 1982 paper [20]. In that publication, the millionaire problem was introduced: Alice and Bob are two millionaires who want to find out which is richer without revealing the precise amount of their wealth. Yao proposed a solution allowing Alice and Bob to satisfy their curiosity while respecting the constraints. This problem and result gave way to a generalization of SMC: a set of n parties with private inputs x1, x2, . . . , xn wish to jointly compute a function f of their inputs, with the property that each party learns the correct output y= f(x1, x2, . . . , xn) and nothing else, even if some of the parties maliciously attempt to obtain more information. This has been shown possible in general [21,22]. Their constructions in the multi-party case are based on representing the computed function as a circuit and evaluating it.

Several researchers have proposed methods for incorporating privacy-preserving requirements in various data mining tasks, such as classification [44]; frequent itemsets [49]; and sequential patterns [36]. In this paper we focus on classification tasks and the k-anonymity model as proposed by Sweeny [46]. A dataset complies with k-anonymity constraints if for each individual, the data stored in the released dataset cannot be distinguished from at least k - 1 individuals whose data also appears in the dataset. Or more generally, each original individual record can only be reconstructed based on released data with a probability that does not exceed 1/k, given knowledge based on information available from external sources.

We propose a new method for achieving k-anonymity, – data mining privacy by decomposition (DMPD). The basic idea is to divide the original dataset into several disjoint projections such that each one of them adheres to k-anonymity. It is easier to make a projection comply with k-anonymity if the projection does not contain all quasi identifier features. Moreover, our procedure ensures that even if the attacker attempts to rejoin the projections, the k-anonymity is still preserved. A classifier is trained on each projection and subsequently, an unlabelled instance is classified by combining the classifications of all classifiers. Because DMPD preserves the original values and only partitions the dataset, we assume that it has a minimal impact on classification accuracy. DMPD does not require domain trees for every attribute nor does it fall into the difficulties encountered by existing algorithms which reduce the validity of the dataset (such as suppression). DMPD supports classification, but can be extended to support other data mining tasks by incorporating various types of decomposition [24].

DMPD employs a genetic algorithm for searching for optimal feature set partitioning. The search is guided by k-anonymity level constraint and classification accuracy. Both are incorporated into the fitness function. We show that our new approach significantly outperforms existing suppression-based and generalization-based methods that require manually defined generalization trees. In addition, DMPD can assist the data owner in choosing the appropriate anonymity level. Using the Pareto efficiency principle, we offer a better way to understand the tradeoff between classification accuracy and privacy level [42].

2. BACKGROUNDS

2.1 Naive Bayes classification

The naive Bayes classifier, or simple Bayes classifier, works as follows :

(i)Each data sample is represented by an m+1 dimensional feature vector (a1; a2; . . . ; am; c), depicting m+1 measurements made on the sample from m+1 attributes, respectively, A1;A2; . . . ;Am;C, where C is the class attribute and c is the class label.

(ii) Suppose that the domain of C is (C1; C2; . . . ; Cl) where Ci≠Cj for i≠j, and thus there exist ƛ classes. Given an unknown data sample, X =(a1; a2; . . . ; am) (i.e., having no class label),the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the naive Bayes classifier assigns an unknown sample X to class Ci if and only if

$$P(Ci|X) > P(Cj|X) \qquad (1)$$
$$\text{for } 1 \le j \le \lambda, \quad j \ne i.$$

Thus we maximize P(Ci|X). The class Ci for which P(Ci|X) is maximized is called the maximum posteriori hypothesis.

(iii) By Bayes theorem, we have

$$P(Ci|X) = \frac{P(X \mid Ci)P(Ci)}{P(X)} \qquad (2)$$

As P(X) is constant for all classes, only $P(X|Ci)P(Ci)$ need to be maximized. The class prior probabilities may be estimated by

$$P(Ci) = \frac{si}{s} \qquad (3)$$

where si is the number of training samples of class Ci and s is the total number of training samples.

(iv) In order to reduce computation in evaluating $P(X|Ci)$, the naive assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another , given the class label of the sample, that is, there are no dependence relationships among the attributes. Thus

$$P(X|Ci)=\prod_{k=1}^{n} P(ak|Ci) \qquad (4)$$

The probabilities $P(a1|Ci)$, $P(a2|Ci)$,...., $P(am|Ci)$ can be estimated from the training sample, namely,

$$P(ak|Ci) = \frac{sik}{si} \qquad (5)$$

where s_{ik} is the number of training samples of class C_i having the value a_k for A_k.

(v) Inorder to classify an unknown sample X, $P(X|Ci)P(Ci)$ is evaluated for each class Ci . Sample X is then assigned to the class Ci if and only if

$$P(X|Ci)P(Ci) > P(X|Cj)P(Cj) \qquad (6)$$

for $1 \leq j \leq \lambda$, $j \neq i$. In other word, it is assigned to the class Ci for which $P(X|Ci)P(Ci)$ is maximum.

2.2. Genetic algorithm-based search

Genetic algorithms (GA), a type of evolutionary algorithm (EA), are computational abstractions, derived from biological evolution, for solving optimization problems through a series of genetic operations [32]. A GA requires a fitness function that assigns a score (fitness) to each candidate in the current population sample (generation).

The fitness of a candidate depends on how well that candidate solves the problem at hand. Selection of candidates is performed randomly with a bias towards those with the highest fitness value. To avoid locally optimal solutions, crossover and mutation operators are introduced to produce new solutions along the whole search space. Thanks to this capability in developing solutions, the GA is recognized today as a highly reliable global search procedure. Other issues involved in using genetic algorithms are the number of details to define in run settings, such as the size of the population and the probabilities of crossover and mutation, and the stop (convergence) criteria of the algorithm. Specific values often depend greatly on the GA's application.

GAs have found to be useful in many data mining tasks in general and in feature selection in particular [27,31,48]. Empirical comparisons between GAs and other kinds of feature selection methods can be found in [45] as well as in [39]. In general, these empirical comparisons show that GAs, with their associated global search in the solution space, usually obtain better results than local search-based feature selection methods. Inspired by these positive results, Rokach [43] presented a GA based framework for solving feature set partitioning tasks. As in feature selection, GAs demonstrate a clear superiority over all other search methods when searching for accurate feature set partitions.

Mitchell [41] indicates the circumstances in which GAs are particularly useful: ‘‘the space to be searched is large; is known not to be perfectly smooth and unimodal; or it is not well understood or if the fitness function is noisy.” The search space in feature set partitioning is known to be large [43] and in our case, the goal function is not well understood due to the tradeoff between k-anonymity constraints and classification performance.

2.3. Genetic algorithms for multiobjective optimization

The DMPD algorithm presented in this work was extended in a natural way to perform a multiobjective optimization to assist data owners in deciding about an appropriate anonymity level in released datasets.

The multiobjective genetic algorithm (MOGA) was designed to solve multiobjective problems where the objectives are generally conflicting thus preventing simultaneous optimization of each objective. The final choice of the solution depends on the user characterizing a subjective approach. User participation in this process is important for obtaining useful results [25].

Optimizing competing objective functions is different from single function optimization in that it seldom accepts one perfect solution, especially for real-world applications [38]. The most successful approach for multiobjective optimization is to determine an entire Pareto optimal solution set or its representative subset [32,26]. A Pareto optimal set is a set of solutions that are non-dominated with respect to each other. While moving from one Pareto solution to another, there is always a certain amount of sacrifice in one objective(s) vs. a certain amount of gain in the other (s). In [35], the authors reported that 90% of the approaches to multiobjective optimization aimed to approximate the true Pareto front for the underlying problem.

The ultimate goal of a multiobjective optimization algorithm is to identify solutions in the Pareto optimal set. However, identifying the entire Pareto optimal set is practically impossible for many multiobjective problems due to the set's size. In addition, for many problems, especially for combinatorial optimization problems, proof of solution optimality is computationally infeasible. Therefore, a practical approach to multiobjective optimization is to investigate a set of solutions (the best-known Pareto set) that represent the Pareto optimal set as best as possible.

With these concerns in mind, a multiobjective optimization approach should achieve the following three, often conflicting, goals [38]:

1. The best-known Pareto front should be as close as possible to the true Pareto front. Ideally, the best-known Pareto set should be a subset of the Pareto optimal set.
2. Solutions in the best-known Pareto set should be uniformly distributed and diverse over the Pareto front in order to provide the decision-maker a true picture of the tradeoffs.
3. The best-known Pareto front should capture the whole spectrum of the Pareto front through investigating solutions at the extreme ends of the objective function space.

Being a population-based approach, genetic algorithms are well suited to solve multiobjective optimization problems. A generic single-objective GA can be modified to find a set of multiple, non-dominated solutions in a single run. The ability of the GA to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems.

3. PRIVACY-PRESERVING NAIVE BAYES CLASSIFIER ON DISTRIBUTED DATA

3.1. Two-party privacy-preserving naive Bayes classifier

In this section, we consider as scenario in which two semi-honest users U1 and U2 owning their confidential databases DB1 and DB2, respectively, wish to learn a naive Bayes classifier on the union DB = DB1 U DB2, without revealing privacy of their databases. We assume that the two databases have the same attributes (A1, A2, . . . , Am, C), where C is the class attribute with a domain of {C1, C2, . . ., Cl}. A user is semi-honest in terms that he provides correct inputs to the naive Bayes classifier, but may want to learn something that violates the privacy of another database. The two users jointly classify a new instance X =(a1, a2, . . . , am).

3.2. Multi-party privacy-preserving naive Bayes classifier

In this section, we consider as scenario in which n(n≥2) semi-honest users U1,U2, . . . ,Un owning their confidential databases DB1, DB2, . . . , DBn, respectively, wish to learn a naive Bayes classifier on the union DB = $\bigcup_{i=1}^{n} DBi$ of their databases with the help of two semi-trusted mixers, without revealing privacy of their databases. We assume that all databases have the same attributes (A1,A2, . . . , Am, C), where C is the class attribute with a domain of {C1, C2, . . . , Cl}. The semi-trusted mixer model is used in multi-party protocol, in which each data site sends messages to two semi-trusted mixers.

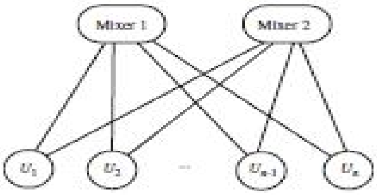


Fig. 1. Semi-trusted mixer model

4. PRIVACY-PRESERVING CLASSIFICATION BY FEATURE SET PARTITIONING

As noted previously, DMPD consists of three main procedures. In the first procedure, a search for an optimal and valid partitioning based on genetic search is carried out. The second involves evaluating a given partitioning while the third N. Matatov et al. / Information Sciences 180 (2010) 2696–2720 2705 procedure combines multiple classifiers to obtain unlabeled instance classifications. Procedure dependency is described as follows: the genetic algorithm uses a wrapper approach to assign a partitioning fitness value while the wrapper uses combined multiple classifiers to obtain test instance prediction. In the final phase, given the optimal partitioning, we anonymize the original dataset.

4.1. Procedure 1: Genetic algorithm-based search

A genetic algorithm for feature set partitioning was presented in [43]. We use this work's partition presentation, crossover and mutation operator.

Given the partitioning Z = {G1, . . . ,Gk, . . . ,Gw}, elements of the matrix B with dimension n x n are assigned as follows:

$$a_{ij} = \begin{cases} 1, & \text{if } i = j \text{ and } \exists k : a_i \in G_k \\ 1, & \text{if } i \neq j \text{ and } \exists k : a_i \in G_k \text{ and } a_j \in G_k \\ -1, & \text{if } \forall k : a_i \notin G_k \text{ and } a_j \notin G_k \\ 0, & \text{otherwise} \end{cases}$$

For example, matrix presentation can be used to present a dataset containing four non-target features in the following partitioning {{a1}{a2,a3}} , Fig. 2.

After the initial population has been produced, the genetic algorithm provides a procedure for choosing the individuals in the population in order to create the offspring that will form the next generation. More particularly, in this stage two parents are chosen and two output offsprings become a combination of these parents using crossover and mutation operator. In consideration of a special encoding scheme for partitioning, there is a crossover operator. The operator, ‘‘group-wise crossover” (GWC), works with probability Pcrossover on two selected partitions. The operator, together with the proposed encoding, does not slow the convergence of the GA [43]. From two of these parents, two ancestor sets are chosen. An ancestor set can be a filtered out, representing a set of features that have not participated in parent partitioning. Fig. 3 presents an example of two offspring from two possible parent partitions from the presented feature set.

As Fig. 3b indicates, the bright gray in the columns and rows denotes ancestor sets from the first parent. The dark grey presents cells copied from another parent. A similar operation is performed on the second parent.

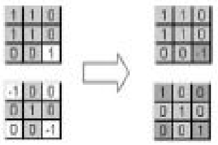
The mutation operator is defined as follows: with probability Pmut, each feature can pass from one subset (source subset) to another (target subset). The feature and subset to which it passes are chosen randomly. If there is only one subset in a partitioning, the source feature will create its own subset. Fig. 4 demonstrates the idea. In this case, the second feature was randomly chosen and passed to a subset containing a third feature. As an inner procedure, the GA uses a wrapper approach to evaluate a partitioning fitness value.

4.2. Procedure 2: Wrapper-based fitness evaluation

We use a wrapper procedure for evaluating the partitioning fitness value. The fitness value was the average accuracy over n runs, where each time n -1 folds were used for training classifiers and one-fold for estimating the generalized accuracy of a validation dataset after combining predictions from different classifiers and obtaining final instance predictions. To obtain appropriate classifications, the wrapper uses Procedure 3 for instance classification.

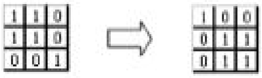
1	0	0	0
0	1	1	0
0	1	1	0
0	0	0	-1

Fig.2. Encoding of example partitioning



(a)Parents (b)Offsprings

Fig.3. Example of GWC operator



(a)Before mutation (b) Mutated offspring

Fig.4. Example of mutation operator

4.3. Procedure 3: Instance classification

In the naive Byes combination, a classification of a new instance is based on the product of the conditional probability of the target feature given the values of the input features in each subset. Mathematically it can be formulated as follows:

$$v_{MAP}(x_q) = \underset{c_j \in dom(y)}{\arg\max} \frac{\prod_{k=1}^{w} \hat{P}_{I(\pi_{G_k \cup y} S)}(y = c_j | a_i = \pi_{G_k} x_q)}{\hat{P}_{I(S)}(y = c_j)^{w-1}}$$

where $\hat{P}_{I(\pi_{G_k \cup y} S)}(y = c_j | a_i = \pi_{G_k} x_q)$ is estimated by using the appropriate frequencies in the relevant decision tree leaves and then adjusting them by performing a Laplace correction.

5. EXPERIMENTAL EVALUATION

The proposed method was evaluated in the presence of k-anonymity constraints for classifications tasks. The comparative experiment was conducted on 10 benchmark datasets containing individual information about various objects. Specifically, the experimental study had the following goals:

1. to examine whether the proposed algorithm succeeded in satisfying the broad range of k-anonymity constraints without sacrificing data mining performance (i.e. the original classification accuracy vs. increasing k-anonymity constraints)
2. to investigate the sensitivity of the proposed method to different classification methods
3. to compare the proposed method to existing k-anonymity-based methods in terms of classification accuracy
4. to examine the sensitivity of runtime cost to different k-anonymity constraints and the scalability of the proposed method.
5. to investigate the sensitivity of the proposed method to different GA settings
6. to investigate the multiobjective DMPD method in providing Pareto frontiers of classification accuracy vs. Partitioning anonymity level.

The following subsections describe the experimental set-up and the results obtained.

5.1. Experimental process

Fig. 5 graphically represents the experimental process that was conducted. The main aim of this process was to estimate the generalized accuracy (i.e. the probability that an instance was classified correctly). First, the dataset (box 1) was divided into a train dataset (box 3) and test dataset (Step 4) using five iterations of a two-fold cross validation (Step 2 – known as the 5x 2 CV procedure). The 5x 2 CV is known to be better than the commonly used 10-fold cross-validation because of the acceptable Type-1 error [23]. At each iteration, the dataset is randomly partitioned into two equal-sized sets, S1 and S2, such that the algorithm is evaluated twice. During the first evaluation S1 is the train dataset and S2 the test dataset, and vice versa during the second evaluation. We apply (Step 5) the k-anonymity method on the train dataset and obtain a new anonymous train dataset (Step 6). Additionally, we obtain a set of anonymity rules (Step 7) that is used to transform the test dataset into
a new anonymous test dataset (Step 8). In the case of DMPD, the rule of partitioning must be applied on an original feature set. For generalization and suppression-based techniques, generalization or suppression rules are applied to different original values in the dataset. An inducer is trained (Step 9) over the anonymous train dataset to generate a classifier (Step 10). Finally the classifier is used to estimate the performance of the algorithm over the anonymous test dataset (Step 11).

The same cross-validation folds were implemented for all the algorithms in the pairwise comparison that we conducted. In each such comparison we used the combined 5 x 2 CV F-test to accept or reject the hypothesis that the two methods (DMPD vs. TDS, DMPD vs. TDR) have the same error rate with a 0.95 confidence level.

It should be noted that the above experimental process is different from the process used by Fung et al. [29]. According to his experimental design, the generalization of the original dataset takes place in the first step. Afterwards, the train and test datasets are split. Thus, the estimated variance of the cross-validation solely measures the inducer’s variance and not the variance due to applying k-anonymity constraints on the underlying dataset.

5.2. Datasets

Privacy-preserving classification algorithms are usually evaluated only on the Adult dataset which has become a commonly used benchmark for k-anonymity [47,30,28]. Recently Fung et al. [29] used a German credit dataset to evaluate the TDR algorithm. In our experimental study we used an additional seven datasets that were also selected from the UCI Machine Learning Repository [40] and which are widely used by the machine-learning community for evaluating learning algorithms. An additional dataset, drawn from a real-world case study performed on commercial banks, is described below.

The datasets vary across such dimensions as the number of target feature classes, instances, input features and their type (nominal, numeric).

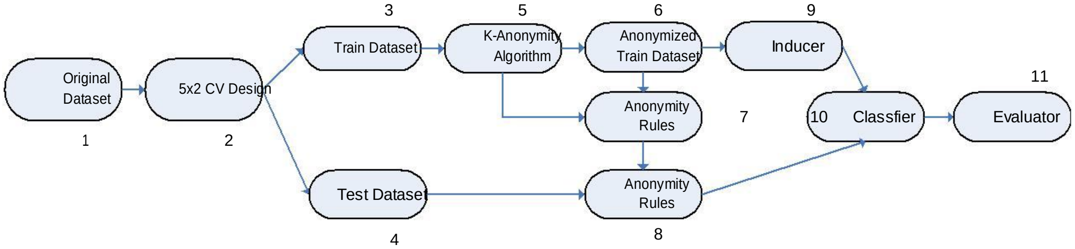


Fig 5. The Experimental Process

Age	Work class	fnlwgt	Education	Educa-tion num	marital-status	Occupation	Relation-ship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
27	Local-gov	54826	Assoc-voc	10	Widowed	Prof-specialty	Not-in-family	White	Female	0	0	20	US	<=50K.
47	Local-gov	181344	Some-college	10	Married	Prof-specialty	Husband	Black	Male	0	0	55	US	>50K.
41	Local-gov	523910	Bachelors	13	Married	Prof-specialty	Husband	Black	Male	0	0	40	US	<=50K.
42	Local-gov	254817	Some-college	10	Married	Prof-specialty	Not-in-family	White	Female	0	1340	40	US	<=50K.
27	Private	213921	HS-grad	9	Divorced	Other-service	Not-in-family	White	Male	0	0	40	US	<=50K.
46	Private	51618	HS-grad	9	Married	Other-service	Not-in-family	White	Female	0	0	40	US	<=50K.
59	Private	159937	HS-grad	9	Married	Sales	Husband	White	Male	0	0	48	US	>50K.
49	Private	343591	HS-grad	9	Divorced	Prof-specialty	Not-in-family	Black	Female	14344	0	40	US	>50K.
53	Private	346253	HS-grad	9	Married	Sales	Husband	White	Male	0	0	35	US	<=50K.
55	Private	198282	Bachelors	13	Married	Sales	Husband	White	Male	15024	0	60	Germany	>50K.
40	Private	118853	Bachelors	13	Widowed	Sales	Unmarried	White	Male	0	0	60	Germany	<=50K.
49	Private	77143	Bachelors	13	Married	Sales	Husband	White	Male	0	0	40	Germany	>50K.
47	Private	253814	HS-grad	9	Divorced	Prof-specialty	Unmarried	White	Female	0	0	25	US	<=50K.
21	Private	312956	HS-grad	9	Married	Prof-specialty	Not-in-family	Black	Male	0	0	40	US	>50K.
43	Private	114580	Some-college	10	Married	Prof-specialty	Not-in-family	White	Female	0	0	40	US	<=50K.
61	State-gov	267989	Bachelors	13	Married	Prof-specialty	Husband	White	Male	0	0	50	US	>50K.
46	State-gov	102628	Masters	9	Widowed	Prof-specialty	Unmarried	White	Male	0	0	40	US	<=50K.
28	State-gov	175325	HS-grad	9	Widowed	Prof-specialty	Unmarried	White	Male	0	0	40	US	<=50K.
28	State-gov	149624	Bachelors	13	Divorced	Prof-specialty	Unmarried	White	Male	0	0	70	US	>50K.
56	State-gov	149624	Bachelors	13	Divorced	Other-service	Unmarried	White	Male	0	0	25	US	>50K.

Fig.6. The adult data sample

5.3. Effect of k-anonymity constraints on classification accuracy

In this section we analyze the effect of the value of k (anonymity level) on classification accuracy. Table 1 shows the accuracy results obtained by the proposed algorithm for four different values of k for various datasets using different inducers. In this section we take the top nine features as quasi-identifiers (top eight for nursery and pima datasets). Note that the column with k = 1 represents the DMPD algorithm result (i.e. when no k-anonymity constraints were implied) enabling us to examine the effect of anonymity on the accuracy of the results. The superscript ‘‘*” indicates that the degree of accuracy of the original dataset was significantly higher than the corresponding result with a confidence level of 95%.
As expected, the results indicate that there is a tradeoff between accuracy performance and the anonymity level for most datasets. Usually, increasing the anonymity level decreases accuracy. For some datasets, the feature set partitioning approach improves baseline accuracy, even despite applying k-anonymity constraints such as for vote, cmc, pima, wisconsine or nursery. Supervised discretization, as a part of DMPD, also contributes to classification accuracy, for example, in the heart dataset. These above results are marked out for both inducer types.

Table 1 : Accuracy vs.
anonymity for DMPD algorithm.

Dataset	Inducer	k-Anonymity level					
Adult	k	Baseline	1	50	100	200	500
	C4.5	85.58 ± 0.51	86.59 ± 0.22	83.03 ± 0.10*	83.00 ± 0.10*	82.63 ± 1.15*	82.99 ± 0.08*
	Naïve Bayes	82.68 ± 0.27	84.87 ± 0.20	81.81 ± 0.13*	81.41 ± 0.79*	81.47 ± 0.77*	81.19 ± 1.11*
	k	Baseline	1	10	20	30	50
Credit	C4.5	85.53 ± 2.45	85.82 ± 1.54	86.71 ± 1.15	85.08 ± 1.75	85.29 ± 1.99	85.66 ± 2.19
	Naïve Bayes	76.82 ± 1.87	85.66 ± 1.89	85.94 ± 1.42	86.46 ± 1.46	85.66 ± 2.45	85.45 ± 2.98
Vote	C4.5	96.23 ± 2.41	96.35 ± 1.95	96.26 ± 2.09	96.35 ± 2.38	95.04 ± 4.06	96.70 ± 1.91
	Naïve Bayes	90.78 ± 1.98	96.26 ± 2.05	96.43 ± 1.85	96.52 ± 2.09	96.61 ± 1.85	96.52 ± 1.97
Wisconsine	C4.5	93.28 ± 1.83	97.00 ± 0.90	95.97 ± 1.40*	94.88 ± 1.00*	95.09 ± 1.71	95.23 ± 1.77
	Naïve Bayes	93.28 ± 1.78	96.11 ± 1.01	95.27 ± 1.57	94.77 ± 2.09	93.89 ± 2.03	93.04 ± 1.09*
German	C4.5	69.88 ± 2.12	73.23 ± 7.44	71.86 ± 2.35	70.78 ± 2.22	71.12 ± 2.51	63.01 ± 16.49
	Naïve Bayes	73.81 ± 1.13	75.03 ± 2.03	73.79 ± 2.37	73.71 ± 2.30	71.26 ± 2.59	69.84 ± 1.87*
Heart	C4.5	74.89 ± 2.97	84.03 ± 6.71	79.18 ± 4.14	78.96 ± 3.53	77.31 ± 3.76	74.78 ± 3.83
	Naïve Bayes	84.89 ± 3.44	82.61 ± 4.37	79.93 ± 4.43	77.69 ± 4.84	78.06 ± 4.75	74.93 ± 3.38*
Portfolio	C4.5	74.82 ± 0.98	76.41 ± 0.56	72.04 ± 1.34*	71.61 ± 2.37*	72.07 ± 1.13*	71.76 ± 1.57*
	Naïve Bayes	58.96 ± 1.81	75.46 ± 0.71	68.74 ± 2.67	69.11 ± 1.84*	68.58 ± 2.72*	68.46 ± 2.47*
Cmc	C4.5	50.90 ± 1.61	56.59 ± 2.73	52.53 ± 4.64	51.51 ± 4.53	51.02 ± 4.19	48.30 ± 2.30
	Naïve Bayes	48.78 ± 1.37	54.04 ± 2.57	54.04 ± 3.33	52.73 ± 4.69	52.12 ± 3.49	48.07 ± 2.00*
Pima diabetes	C4.5	72.09 ± 2.20	78.20 ± 2.05	77.75 ± 1.53	77.62 ± 2.66	76.79 ± 3.01	75.85 ± 1.87
	Naïve Bayes	75.20 ± 2.67	77.75 ± 1.93	77.75 ± 2.27	78.02 ± 2.34	78.41 ± 2.45	78.30 ± 2.26
	k	Baseline	1	50	100	150	200
Nursery	C4.5	96.16 ± 0.31	97.13 ± 0.52	92.64 ± 1.15*	91.15 ± 0.81*	90.86 ± 1.27*	88.91 ± 4.79
	Naïve Bayes	90.08 ± 0.40	90.25 ± 0.51	90.24 ± 0.57	90.27 ± 0.67	90.00 ± 0.77	89.46 ± 0.94

5.4. Scalability analysis

The aim of this section is to examine the DMPD's ability to handle expanding datasets in an elegant manner. We tested for scalability using the procedure that Fung et al. [29] proposed to measure the runtime costs of algorithms on large datasets. We based our scalability test on a German dataset. For this purpose, the original dataset containing 1000 records was expanded as follows: for every original instance q, we added r -1 variations where r is a scale factor. Together with all original instances, the enlarged dataset has r x 1000 instances. Each instance variation was generated by randomly drawing appropriate feature values (xqi, yq) from the feature domain (dom(ai), dom(y)).

We conducted all experiments on a hardware configuration that included a desktop computer implementing a Windows XP operating system with Intel Pentium 4, 2.8 GHz, and 1 GB of physical memory. Table 2 presents the average time (in minutes) measured on 10 runs for various values of r, a quasi-identifier that includes top 5 features and a k-anonymity level = 10. The time, including model generation time, reflects the runtime cost of a J4.8 inducer in a WEKA package and is beyond our control. Fig. 7 shows that the execution time is almost linear in the number of records. This confirms the DMPD's ability to handle a growing search space since an increase in train dataset size leads to more partitions that the algorithm must consider.

5.5. Genetic Algorithm(GA) settings for DMPD

A major problem with genetic algorithms is their sensitivity to the selection of various parameters such as population size, maximum number of generations, crossover and mutation probabilities. This sensitivity is due to a degree of randomness in searching for an optimal solution. We set crossover and mutation values to probabilities that are similar to those found in the GA literature [32,34]. In this section we try to estimate if better results can be obtained from a larger population size or a larger number of generations. Obviously, we have here a probable tradeoff between, better partitioning structure and computational costs of the algorithm due to our increasing the GA settings.

Our default settings for DMPD were 100 generations and 50 individuals in a population. Here we present some experiments from the German and cmc datasets. We present two of 12 experiments that were performed on the dataset with a C4.5 inducer with a minimum of k-anonymity constraints (top5 and k = 10). The number of generations considered is 50, 100, 150 and 200. Test runs were carried out on two populations numbering 50 and 100 individuals, respectively.

As Fig. 8 shows, increasing GA settings beyond 100 generations and 50 partitionings does not significantly improve classification accuracy. Similar behavior is true for other datasets used in the experimental study. The evidence points to the DMPD's ability to achieve good feature set partitioning with a relatively low number of generations and population size.

Table 2 : Scalability analysis for DMPD.

Scalability factor	Algorithm runtime (min)	Dataset size
1	1.41	1000
5	6.55	5000
10	13.90	10,000
15	24.48	15,000
20	32.76	20,000

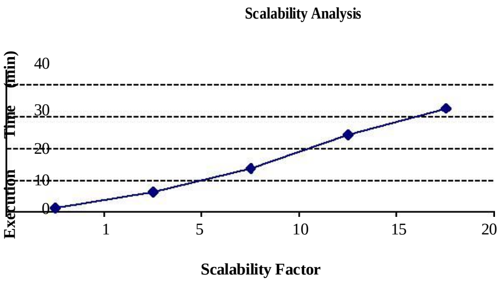


Fig. 7. Scalability trend in the extended German dataset.

5.6. Multiobjective DMPD evaluation

In this section we introduce experimental results of the DMPD for multiobjective decision-making in privacy-preserving data mining. We follow a method presented by Grunert da Fonseca et al. [33] for experimentally evaluating multiobjective GA. Such an evaluation is performed on the basis of multiple, independent optimization runs and by building attainment functions. The concept of empirical attainment function is used in Pareto frontier estimation by building a specific type of summary attainment surface [37] on objective space. In our case, there are two objectives: partitioning anonymity level and classification accuracy. The worst attainment surface describes the region that was attained by all optimization runs and provides a pessimistic estimation for the true Pareto frontier. The region, attained by only one optimization run, presents us with the best attainment surface and this is an optimistic estimation for a true Pareto frontier. Median attainment surface, attained by 50% of optimization runs, provides a reasonable estimation for the true Pareto frontier [26,37]. In Fig. 9 we present three types of empirical surfaces for true Pareto frontier estimation created by an algorithm for an Adult dataset with top5 and top9 quasi-identifier constraints and a C4.5 inducer.

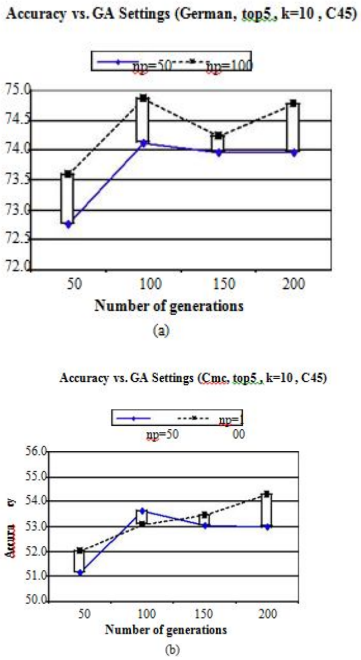


Fig. 8. GA settings vs. accuracy for german and cmc datasets.

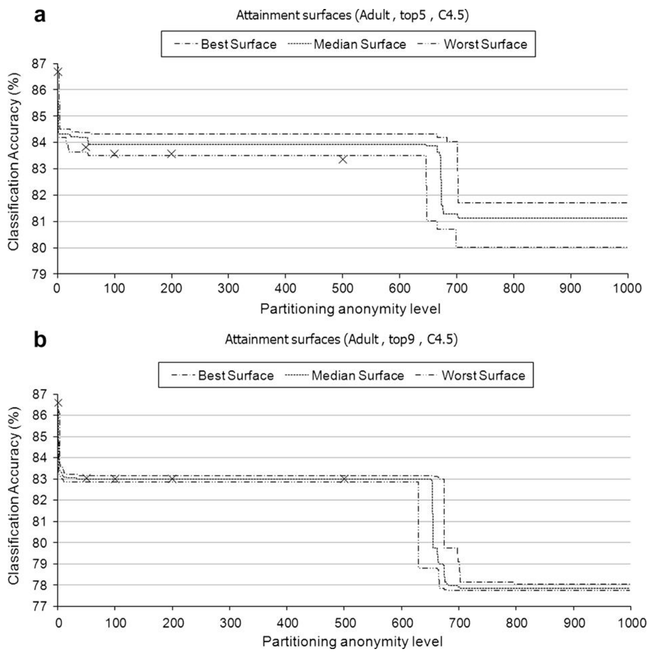


Fig. 9. Accuracy vs. k-anonymity level tradeoff.

From the graphs we conclude that there is a clear tradeoff between the two objectives (Fig. 9a). The graphs provide an important tool for a data owner. If he can agree to a k-level between 1 and 3, the classification accuracy that can be achieved is nearly 86%. A further key decision point arises near a k-level of 660. Classification performance from k = 3 to k = 655 is essentially the same excluding some negligible decreases in accuracy in the range between k = 4 and k = 35. If these k-anonymity levels are insufficient for the data owner, he can advance along the frontier to an additional k level. Increasing k up to 1000, results in a decrease of 5% in the classification accuracy.

6. CONCLUSIONS

In this paper we presented a new method for preserving privacy in classification tasks using a naive Bayes classifier and k-anonymity framework with genetic algorithm. The proposed method was designed to work with no prior knowledge and with any inducer. Compared to existing state-of-the-art methods, the new method also shows a higher predictive performance on a wide range of datasets from different domains. Additional issues to be further studied include: Examining DMPD with other classification algorithms, Examining the possibility of using DMPD along with known generalization/suppression-based methods that could result in improved data mining results in terms of classification accuracy and discovered patterns, Extending k-anonymity to l-diversity framework. Extending DMPD to handle multiobjective optimization with different quasi-identifier sets. The main idea is to eliminate the k-anonymity model assumption; the quasi-identifier set is determined prior to performing data anonymization.

References

[1.] D. Struck, Don’t store my data, Japanese tell government, in: International Herald Tribune, 25 August 2002.

[2.] H. Kargupta, P. Chan, Advances in Distributed and Parallel Knowledge Discovery, MIT, AAAI Press, Cambridge, New York, 2000.

[3.] J. Vaidya, C. Clifton, Privacy-preserving data mining: Why, how and when, IEEE Security and Privacy, November/December 2004, pp. 19–27.

[4.] N. Zhang, S. Wang, W. Zhao, A new scheme on privacy-preserving data classification, in: Proceedings of KDD’05, 2005, pp. 374–383.

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

[6.] R. Agrawal, R. Srikant, Privacy-preserving data mining, in: Proceed- ings of ACM SIGMOD International Conference on Management of Data, 2000, pp. 439–450.

[7.] Y. Lindell, B. Pinkas, Privacy preserving data mining, Journal of Cryptology 15 (3) (2002) 177–206.

[8.] A. Evfimievski, S. Ramakrishnan, R. Agrawal, J. Gehrke, Privacy- preserving mining of association rules, in: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, July 2002.

[9.] M. Kantarcioglu, J. Vaidya, Privacy preserving naive Bayes classifier for horizontally partitioned data, in: Proceedings of IEEE Workshop on Privacy Preserving Data Mining, 2003.

[10.]J. Vaidya, C. Clifton, Privacy-preserving association rule mining in vertically partitioned data, in: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, 2002, pp. 639–644.

[11.]V.S. Verykios, A.K. Elmagarmid, E. Bertino, Y. Saygin, E. Dasseni, Association rule hiding, IEEE Transactions on Knowledge and Data Engineering 16 (4) (2004) 434–447.

[12.]S.J. Rizvi, J.R. Haritsa, Maintaining data privacy in association rule mining, in: Proceedings of the 28th International Conference on Very Large Data Bases, 2002, pp. 682–693.

[13.]C. Clifton, M. Kantarcioglou, X. Lin, M.Y. Zhu, Tools for privacy preserving distributed data mining, SIGKDD Exploration 4 (2) (2002) 1–7.

[14.]J. Vaidya, C. Clifton, Privacy-preserving k-means clustering over vertically partitioned data, in: Proceedings of 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, 2003, pp. 206–215.

[15.]M. Kantarcioglu, J. Vaidya, Privacy-preserving naive Bayes classifier for horizontally partitioned data, in: IEEE Workshop on Privacy Preserving Data Mining, 2003.

[16.]J. Vaidya, C. Clifton, Privacy preserving naive Bayes classifier on vertically partitioned data, in: 2004 SIAM International Conference on Data Mining, 2004.

[17.]R. Wright, Z. Yang, Privacy-preserving Bayesian network structure computation on distributed heterogeneous data, in: KDD'04, Seattle, Washington, USA, August 2004

[18.]Z. Yang, S. Zhong, R. Wright, Privacy-preserving classification of customer data without loss of accuracy, in: Proceedings of the 5th SIAM International Conference on Data Mining, Newport Beach, CA, April2005.

[19.]H.Kargupta,S.Datta,Q.Wang,K.Sivakumar,On the privacy preserving properties of random data perturbation techniques, in: Proceedings of the 3rd International Conference on DataMining, 2003,pp.99–106.

[20.]A.C.Yao,Protocols for secure computations(extended abstract),in: Proceedings of the 23th IEEE Symposiumon Foundations of Computer Science,1982,pp.160–164.

[21.]O.Goldreich,S.Micali,A.Wigderson,How to play any mental game—a completeness theorem for protocols with honest majority, in: Proceedings of the 19th ACM Symposium on the Theory of Computing, 1987,pp.218–229.

[22.]D.Chaum, C.Crepeau, I.Damgard, Multi party unconditionally secure protocols,in:Proceedings of the 20th ACMS ymposiumon the TheoryofComputing,1988,pp.11–19.

[23.]E. Alpaydin, Combined 5 _ 2 CV F-test for comparing supervised classification learning classifiers, Neural Computation 11 (1999) 1975–1982.

[24.]S. Cohen, L. Rokach, O. Maimon, Decision-tree instance-space decomposition with grouped gain-ratio, Information Sciences 177 (17) (2007) 3592– 3612

[25.]U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery: an overview, in: Advances in Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA, 1996, pp. 1–31.

[26.]C.M. Fonseca, P.J. Fleming, Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, in: S. Forrest (Ed.), Proc. of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1993, pp. 416–423

[27.]A. Freitas, Evolutionary algorithms for data mining, in: O. Maimon, L. Rokach (Eds.), The Data Mining and Knowledge Discovery Handbook, Springer, 2005, pp. 435–467

[28.]A. Friedman, A. Schuster, R. Wolff, Providing k-anonymity in data mining, VLDB 17 (4) (2008) 789–804.

[29.]B.C.M. Fung, K. Wang, P.S. Yu, Anonymizing classification data for privacy preservation, IEEE Transactions on Knowledge and Data Engineering 19 (5) (2007) 711–725.

[30.]B.C.M. Fung, K. Wang, P.S. Yu, Top-down specialization for information and privacy preservation, in: Proc. of the 21st IEEE International Conference on Data Engineering, ICDE05, IEEE Computer Society, Washington, DC, 2005, pp. 205–216

[31.]M.S. Gibbs, G.C. Dandy, H.R. Maier, A genetic algorithm calibration method based on convergence due to genetic drift, Information Sciences 178 (14) (2008) 2857–2869

[32.]D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Boston, Maryland, 1989

[33.]V. Grunert da Fonseca, C.M. Fonseca, A.O. Hall, Inferential performance assessment of stochastic optimizers and the attainment function, in: E. Zitzler et al. (Eds.), Proc. of Conference on Evolutionary Multi-Criterion Optimization (EMO 2003), Lecture Notes in Computer Science, vol. 1993, Springer-Verlag, 2001, pp. 213–225.

[34.]R.L. Haupt, S.E. Haupt, Practical Genetic Algorithms, second ed., John Wiley, 2004.

[35.] D.F. Jones, S.K. Mirrazavi, M. Tamiz, Multiobjective meta-heuristics: an overview of the current state-of-the-art, European Journal of Operational Research 137 (1) (2002) 1–9.

[36.] S.W. Kim, S. Park, J.I. Won, A.W. Kim, Privacy preserving data mining of sequential patterns for network traffic data, Information Sciences 178 (3) (2008) 694–713

[37.] J. Knowles, A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers, in: Proc. of the Fifth International Conference on Intelligent Systems Design and Applications (ISDA V), IEEE Computer Society, Washington, DC, 2005, pp. 552–557.

[38.] A. Konaka, D.W. Coitb, A.E. Smithc, Multi-objective optimization using genetic algorithms: a tutorial, Reliability Engineering and System Safety 91 (2006) 992–1007

[39.] C.J. Merz, P.M. Murphy, UCI Repository of machine learning databases, Department of Information and Computer Science, University of California, Irvine, CA, 1998

[40.] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1996.

[41.] S.R.M. Oliveira, O.R. Zaïane, Toward standardization in privacy-preserving data mining, in: Proc. of the Third Workshop on Data Mining Standards, ACM, New York, NY, 2004, pp. 7–17

[42.] M. Meints, J. Moller, Privacy preserving data mining – a process centric view from a European perspective, Available online at http://www.fidis.net, 2004.

[43.] D. Shah, S. Zhong, Two methods for privacy preserving data mining with malicious participants, Information Sciences 177 (23) (2007) 5468–5483.

[44.] P.K. Sharpe, R.P. Glover, Efficient GA based techniques for classification, Applied Intelligence 11 (1999) 277–284.

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

[46.] K. Wang, P.S. Yu, S. Chakraborty, Bottom-up generalization: a data mining solution to privacy protection, in: Proc. of the Fourth IEEE International Conference on Data Mining, IEEE Computer Society, Washington, DC, 2004, pp. 249–256.

[47.] J. Zhang, J. Zhuang, H. Du, S. Wang, Self-organizing genetic algorithm based tuning of PID controllers, Information Sciences 179 (7) (2009) 1007–1018

[48.] S. Zhong, Privacy-preserving algorithms for distributed mining of frequent itemsets, Information Sciences 177 (2) (2007) 490–503

[49.] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evolutionary Computation 8 (2) (2000) 173–195.

[50.] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength Pareto evolutionary algorithm, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Tech. Rep. 103, 2001.

[51.] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Transactions on Evolutionary Computation 3 (4) (1999) 257–271.

AUTHOR

Lambodar Jena received his M.Tech degree in Computer Science and Application from Indian School of Mines(ISM), Dhanbad in the year 2000 and currently pursuing doctorate(Ph.D) degree from Utkal University, India . His research interest includes data mining ,data privacy, image mining, web mining and wireless sensor networking. He has several publications in national/international journals and conference proceedings.

Narendra Kumar Kamila received his M.Tech degree in Computer Science and Engineering from Indian Institute of Technology(IIT), Kharagpur and doctorate(Ph.D) degree from Utkal University, India in the year 2000. Later he had visited USA for his post doctoral work at University of Arkansas in 2005. His research interest includes artificial intelligence, data privacy, image processing and wireless sensor networking. He has several publications in journal and conference proceedings. His professional activities include teaching computer science, besides he organizes many conferences, workshops and faculty development programs funded by All India Council for Technical Education, Govt. of India. Dr. Kamila has been rendering his best services as editorial board member and editor-in-chief of many international journals and conference proceedings.