Using Anonymized Data for Classification

Ali Inan [#1*], Murat Kantarcioglu [#2*], Elisa Bertino [†3*]

[#]*Department of Computer Science, The University of Texas at Dallas*
Richardson, TX 75083, USA
[1]inan@student.utdallas.edu
[2]muratk@utdallas.edu

[†]*Department of Computer Sciences, Purdue University*
West Lafayette, IN 47907, USA
[3]bertino@cs.purdue.edu

***Abstract*— In recent years, anonymization methods have emerged as an important tool to preserve individual privacy when releasing privacy sensitive data sets. This interest in anonymization techniques has resulted in a plethora of methods for anonymizing data under different privacy and utility assumptions. At the same time, there has been little research addressing how to effectively use the anonymized data for data mining in general and for distributed data mining in particular. In this paper, we propose a new approach for building classifiers using anonymized data by modeling anonymized data as uncertain data. In our method, we do not assume any probability distribution over the data. Instead, we propose collecting all necessary statistics during anonymization and releasing these together with the anonymized data. We show that releasing such statistics does not violate anonymity. Experiments spanning various alternatives both in local and distributed data mining settings reveal that our method performs better than heuristic approaches for handling anonymized data.**

I. Introduction

Privacy sensitive information related to individuals, such as medical data, is today collected, stored and processed in a large variety of application domains. Such data is typically used to provide better quality services to individuals and its availability is crucial in many contexts. In the case of healthcare, for example, the availability of such data helps prevent medical errors and enhance patient care. Privacy sensitive data may have many important legitimate uses serving distinct purposes outside the specific domain in which it has initially been collected. For example, drug companies and researchers may be interested in patient records for drug development. Such additional uses of data are important and should certainly be supported. Yet, privacy of the individuals to whom the data is related should be assured as well.

To address the conflicting requirements of assuring privacy while at the same time supporting legitimate use, several different techniques ranging from cryptographic methods [1], [2] to perturbation methods [3] have been proposed. Among those methods, anonymization techniques have emerged as one of the most important privacy preservation tools. Recently, several anonymization methods have been proposed based on different privacy definitions (e.g., k-anonymity [4], l-diversity [5], t-closeness [6]), different algorithms, (e.g., DataFly [4], Top-down specialization [7], Incognito [8], Mondarian multi-dimensional k-anonymity [9]), and different data assumptions (e.g., anonymizing data streams [10]). Anonymization techniques such as k-anonymization [4] usually use privacy metrics that measure the amount of privacy protection and transform the original data in such a way that it will be hard to infer identities of the individuals when released for outside use.

*This work was partially supported by AFOSR grant FA9550-07-1-0041.

R	A_1	A_2
r_1	Masters	35
r_2	Masters	36
r_3	Masters	36
r_4	11th	28
r_5	11th	22
r_6	12th	33

R'	A_1	A_2
r'_1	Masters	[35-37)
r'_2	Masters	[35-37)
r'_3	Masters	[35-37)
r'_4	Senior Sec.	[1-35)
r'_5	Senior Sec.	[1-35)
r'_6	Senior Sec.	[1-35)

TABLE I
DATA SET R AND R'S 3-ANONYMOUS GENERALIZATION R'

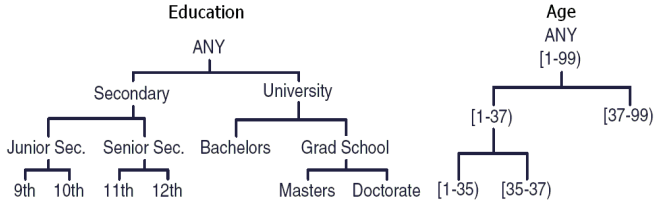


Fig. 1. Value generalization hierarchies for $Education$ and Age attributes

For example, consider data set R in Table I that stores age and education information of certain individuals. Under the k-anonymity approach, quasi-identifier fields Age and $Education$ are generalized according to value generalization hierarchies (e.g., see Figure 1) such that there are at least k individuals that share the same quasi-identifying information. The data set R' in Table I is one of the many 3-anonymous versions of R.

One important purpose for which anonymized data is used is represented by data mining. Data mining harnesses huge amounts of data available in many application domains to extract knowledge crucial to the progress of research and other human activities. A key question is thus whether and how

anonymized data can be effectively used for data mining and which anonymization methods and/or privacy definitions best support different data mining tasks.

For some data mining tasks, there are trivial and effective answers to this question. For example, if we want to build decision trees, we could expand the domains of quasi-identifier attributes by adding generalized values. Consider the anonymized table R' in Table I. Generalized value "Senior Sec." can be considered another category within the domain of $Education$, disregarding the fact that it actually represents either "11th" grade, or "12th" grade.

Mining anonymized data becomes, however, more challenging if we use a data mining method that requires a distance computation between attributes. For example, suppose that we need to compute the Euclidean distance between the attribute values $r'_1.A_2$ and $r'_6.A_2$ from Table I. Then, what is the Euclidean distance between "[35-37)" and "[1-35)"? One trivial solution can be computing the distances based on median values of these ranges. In other words, we can estimate the Euclidean distance between $r'_1.A_2$ and $r'_6.A_2$ as $\sqrt{(36-18)^2}$. The obvious question is whether such trivial approaches are the best way of handling anonymized data.

In this paper we address the problem of classification over anonymized data. We propose a novel approach that models generalized attributes of anonymized data as uncertain information. In our approach, each generalized value of an anonymized record r is accompanied by statistics collected from records in the same equivalence class as r. This extra information, released with the anonymized data, supports the accurate computation of expected values of important functions for data analysis such as dot product and square distance. Consequently, we can seamlessly extend many classification algorithms to handle anonymized data. For example, given the information that $E(r'_1.A_2) = 35.6$ and $E(r'_4.A_2) = 27.6$, we can calculate expected value of the dot product $r_1.A_2^T r_4.A_2$ accurately as $E(r'_1.A_2^T r'_4.A_2) = E(r'_1.A_2) \times E(r'_4.A_2) = 35.6 \times 27.6 = 982.56$. In section III-C, we discuss how such an approach to compute the "expected square distance" and the "expected dot product" can be applied to various classification algorithms.

Also in this paper, based on our distance function approach, we address issues related to different uses of anonymized data sets in the data mining process. We envision the following uses of anonymized data sets:

Classifying Anonymized Data Using Data Mining Models Built on Anonymized Data: This is the typical application of anonymized data sets for data mining. For example, consider a medical researcher who wants to build a classifier on some medical data set. Due to privacy concerns, the researcher is not provided direct access to the data. Instead, the classifier has to be built and tested over an anonymous version of the medical data set.

Classifying Original Data Using Data Mining Models Built on Anonymized Data: In some cases, data mining models built on anonymized data sets may need to be tested on the original data sets (i.e., the data sets with no modification). For example, several hospitals may collaborate to create one large anonymized data set which is then shared among all collaborating hospitals. For a researcher in a participating hospital who is interested in using data mining to classify the patients' of the hospital, there will be at least two options. Either, the researcher can build a classifier using the local data set and use it for new patients or she can build a classifier using the large anonymized data set that involves many more samples and use it for new patients. To see which of these two options is better, we need to be able to classify original data (e.g., the medical records of the new patient) using data mining models built on anonymized data.

Notice that in the above scenario, the researcher could anonymize the set of new patients and convert the problem to classifying anonymized data with models built on anonymized data (which, as we explained above, is a much more simpler problem). Although a perfectly viable alternative, how the set of new patients is anonymized becomes crucially important. Inserting every new patient's data into the hospital's training data set and re-anonymizing the data would take too long. Adjusting anonymization parameters for the small set of new patient data set (to be anonymized independently) is not easy either. On the other hand, if the model built on anonymized data supported classifying original data, new patients could be classified quickly and accurately.

Classifying Anonymized Data Using Data Mining Models Built on Original Data: In some cases, we may need to classify anonymized data using data mining models built on original data. For example, for homeland security applications, the Department of Homeland Security (DHS) can legally have access to enough original data to build data mining models (e.g., it may have a large enough number of legally retrieved phone records related to suspicious activities.). At the same time, due to privacy concerns, DHS may not be allowed to directly access individually identifiable data (e.g., all phone records stored in the repositories of a phone company) for classification purposes. Instead, DHS may be given access to anonymized data for such purposes. To handle such situations, we need to be able to classify anonymized data using data mining models built on original data.

Distributed Data Mining Using Anonymized Data: If individual privacy is the main concern, data sets that are anonymized locally could be distributed for building data mining models. For example each hospital locally anonymizes the patient discharge data and then shares the anonymized data with other hospitals. The obvious question is whether building classification models from the union of locally anonymized data sets results in good classification results.

In this paper, we address issues related to classification in the last three of the above different uses of anonymized data. We omit the problem of classifying anonymized data using models built on anonymized data because this problem is studied heavily and the problem is inherently the same as the classification problem over original data. Anonymization only maps the quasi-identifier attributes to the domain of generalized values. Since both training and test data are

anonymized, classification can be performed easily over these new domains using existing methods.

A. Our Contributions

To our knowledge, this paper is the first attempt to understand different aspects of building and using classifiers over anonymized data. More specifically,

- We develop a new approach for computing expected dot product and square distance on anonymized data by treating anonymized instances as uncertain information.
- We show how this approach can be used for classification.
- We show how our techniques can be directly used to handle cases where either the training or the test data is anonymized.
- We show the effectiveness of using locally anonymized data sets for distributed data mining.
- We provide extensive experimental analysis to explore possible approaches for building data mining models from anonymized data.
- Our method applies to any anonymization method based on generalization.
- Our method applies to any data analysis task based on dot product and square distance.

The paper is organized as follows. In section II, we discuss the related work. In section III, we formally define the problem and discuss various approaches. We present our experimental results in section IV and conclude in section V.

II. Related Work

In recent years, anonymization has become a widely investigated research direction in an effort to protect individual privacy while allowing microdata release. The initial privacy definition of anonymity by Sweeney, namely k-anonymity [4], has been extended by introducing new constraints such as l-diversity [5] and t-closeness [6].

The very first methods for anonymizing privacy sensitive data sets have been proposed by Sweeney [4]. The single-dimensional global recoding methods proposed by Sweeney, namely $DataFly$ and μ-Argus, are based on generalization and suppression. Later, LeFevre et al. proposed full-domain [8] and multi-dimensional anonymization methods [9]. More recent work suggests partitioning the data into "sensitive" and "quasi-identifier" tables [11], and permuting sensitive values within equivalence classes [12].

Most of these approaches focus on optimizing generic measures without any emphasis on the utility of anonymized data. In [13], Kifer et al. propose publishing marginal tables that contain additional information related to quasi-identifier attributes. Zhang et al. suggest a similar approach to facilitate accurately answering aggregate queries [12]. Xu et al. offer a different approach to increase the utility of anonymized data [14]. In this method, instead of releasing extra information related to anonymized data, the anonymization process itself is made utility-aware by combining cost measures with attribute-utility measures, specified by the party that will utilize the anonymized data. Our method is more in the line of [12] and [13] in that we propose releasing statistics related to quasi-identifier attributes as well. But in contrast to these studies, our work defines utility far more specifically, collects statistics per equivalence class and releases these statistics within the anonymized data as new attributes, so that statistical information corresponding to each record is readily available.

Surprisingly, there has been little research that investigates the performance of data mining algorithms on anonymized data. One exception is the "top-down specialization" (TDS) method [7], where data is anonymized based on the class conditional entropy measure. In essence, TDS is an anonymization method tweaked for building accurate decision trees using anonymized data, while our aim is building a wide range of classifiers using (already) anonymized data. LeFevre et al. [15] propose several algorithms for generating an anonymous data set that can be used effectively over pre-defined workloads. Workload characteristics taken into account by those algorithms include selection, projection, classification and regression. Additionally, LeFevre et al. consider cases in which the anonymized data recipient wants to build models over multiple different attributes.

Most learning algorithms model the input data as points in an m dimensional space, where m is the number of attributes. However, when the input is anonymized over q quasi-identifier attributes, multi-dimensional recoding functions output q-dimensional hyper-rectangles. In order to support learning from regions, LeFevre et al. [15] apply a preprocessing step which maps q-dimensional hyper-rectangles onto points in $2q$-dimensional space. Our approach differs from the work by LeFevre et al. in various aspects. First, we model anonymized data as uncertain data. Second, we consider application scenarios in which anonymized data and original data can be used for testing and training. The scenario considered by LeFevre et al. is restricted to building models over anonymized data; such models are then tested over anonymized data. Third, the transformation they propose corresponds to only one of the several heuristics we propose in section III-B.

Another method of mining anonymized data is modeling it as uncertain information. Ngai et al. [16] propose an algorithm for clustering objects whose location within a bounding hyper-rectangle is not known. Kriegel et al. propose a method for hierarchical density-based clustering of fuzzy objects [17]. Chiu et al. [18] address the problem of mining frequent itemsets, where each item is associated with existential probabilities. Leung et al. [19] consider the same problem and propose a modified version of the F-Growth algorithm that handles uncertain data. Xiao et al. [20] address the problem of distance calculation for uncertain objects. The recent work by Aggarwal [21] aims at unifying anonymization methods and existing tools for uncertain data mining. In this study, every record is transformed into a new, perturbed data record so that the original value is probabilistically distributed around some assigned density function. However, releasing a probability density function for each data record requires changing the anonymity definition. The method we propose relies on the original definition of k-anonymity. Our work is different from

those approaches, and other studies in the area of uncertain data mining and fuzzy systems because we specifically focus on anonymized data for classification purposes and do not assume any probability distribution in our methods.

Nearest neighbor classification with generalization has been investigated by Martin [22]. The main purpose of generalizing exemplars (by merging them into hyper-rectangles) is to improve speed and accuracy as well as inducing classification rules, but not to handle anonymized data. Martin proposes building non-overlapping, non-nested generalized exemplars in order to induce high accuracy. However, we address classification of anonymized data, where overlapping is common for most anonymization methods.

Zhang et al. discuss methods for building naive Bayes and decision tree classifiers over partially specified data [23], [24]. Partially specified records are defined as those that exhibit non-leaf values in the taxonomy trees of one or more attributes. Therefore generalized records of anonymous data can be modeled as partially specified data. However, the application scenario considered by Zhang et al. does not overlap with ours since in their approach classifiers are built on a mixture of partially and fully specified data.

III. Problem Formulation and Solution

In many cases, anonymized data instances contain values that are generalized based on some value generalization hierarchy (e.g., see Figure 1). The main question we address is how to utilize anonymized data sets for data mining purposes. Clearly, we can represent each generalization as a discrete value. If the data mining algorithm at hand works well with discrete values, as in the case of decision tree mining [7], then anonymized data can be used directly. However, for the cases in which we need to compute the distance between instances, the best approach to handle anonymized data is not clear. For example, consider the simple k-nearest neighbor classifier for classifying a new instance $\mathbf{x}_q$ given the training data $(\mathbf{x}_i, y_i)$ for $i = 1 \dots n$ where $\mathbf{x}_i$ denotes a record and $y_i \in \{+1, -1\}$ indicates the class value of $\mathbf{x}_i$. To classify the instance $\mathbf{x}_q$, we need to find the k nearest neighbors of $\mathbf{x}_q$ in the training data set. Such need clearly raises the question of how to compute the distance between two anonymized data instances.

In this paper, we focus on k-nearest neighbor (k-NN) and support vector machine (SVM) classifiers that require computing distance functions and dot products over anonymized data. Although we focus on these two methods, our approach can be applied to other data analysis tasks which require computing distances and/or dot products of anonymized data instances (e.g., k-means clustering). In what follows, we first give a brief overview of the SVM methods and then discuss how to use anonymized data sets for SVM and k-NN classification.

A. Overview of SVM Classification

Methods based on SVMs are used extensively for various classification tasks [25]. One of the most popular SVMs, namely C-SVM [25], can be defined as follows: Given training data $(\mathbf{x}_i, y_i)$, $i = 1 \dots n$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{+1, -1\}$ indicates the class value of $\mathbf{x}_i$, solve the following optimization problem

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\dot{\Phi}(\mathbf{x}_i) + b) \geq 1 - \xi_i \text{ for } i = 1 \dots n \\ & \xi_i \geq 0 \end{aligned}$$

where $\Phi : \mathbb{R}^d \mapsto \mathcal{H}$ is a mapping from the feature space to some high dimensional Hilbert space $\mathcal{H}$, $w \in \mathcal{H}$, and $b \in \mathbb{R}$. Here, $C \geq 0$ is a parameter that controls the trade-off between minimization of the margin errors and maximization of the margins. For efficiency reasons, Φ is chosen so that an efficient kernel function K exists, where $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T\dot{\Phi}(\mathbf{x}_j)$. In practice, instead of solving the above optimization problem, the following dual version derived by forming the Lagrangian (α is a Lagrange multiplier) is solved:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{n}\alpha_i \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \text{ for } i = 1 \dots n \\ & \sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned}$$

By solving the dual problem, we can compute w as $w = \sum_{i=1}^{n}\alpha_i y_i \Phi(\mathbf{x}_i)$. A vector $\mathbf{x}_i$ is called a support vector if the corresponding α_i is not equal to zero (for most $\mathbf{x}_i$, we will have $\alpha_i = 0$). We classify a new instance $\mathbf{x}$ by computing the sign of the following function

$$f(x) = \sum_{i=1}^{n_S}\alpha_i y_i K(\mathbf{s}_i, \mathbf{x}) + b$$

where $\mathbf{s}_i$ are the support vectors and n_S is the number of support vectors.

As it can be seen by the above discussion, if we can efficiently and accurately compute $K(\mathbf{x}_i, \mathbf{x}_j)$ for anonymized data instances, we can easily build support vector machines using anonymized data. In section III-C, we discuss the details of how to compute $K(\mathbf{x}_i, \mathbf{x}_j)$ for anonymized data instances for common kernel functions such as:

- **Linear:** $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j$,
- **Polynomial:** $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma\mathbf{x}_i^T\mathbf{x}_j + r)^d$, where $\gamma > 0$,
- **Radial Basis Function (RBF):** $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$ where $\gamma > 0$,
- **Sigmoid:** $K(\mathbf{x}_i, \mathbf{x}_j) = tanh(\gamma\mathbf{x}_i^T\mathbf{x}_j + r)$.

Notice that RBF kernel requires calculating the square distance between two anonymized records. Since k-NN classifiers rely only on distance measures, throughout the rest of this section, we will only focus on computing the (expected) value of SVM kernels. The extension to square distance for k-NN classifiers is straightforward.

B. Heuristic Feature Representation for Anonymized Data

SVM classifiers can easily handle instances with missing values. If the value of an attribute is unknown, not setting

any of the features for that attribute is sufficient (and usually effective) [25]. However, there is no obvious mechanism to handle anonymized data. Therefore, we propose several simple heuristics for feature representation of anonymized, or more specifically, generalized data. We provide three heuristics for numerical attributes and three heuristics for categorical attributes. In combination, there are $3 \times 3 = 9$ different methods to build feature vectors of instances.

1) Numerical Attributes: Generalization of a numerical value yields a range. These ranges could be considered as a form of discretized values. Therefore the first heuristics that we propose is to represent generalized numerical attributes as categorical attributes, whose categories consist of the ranges. One problem with this naive heuristics is that it is not able to capture the relationship between overlapping ranges. Given two generalized values, that is, two ranges, the dot product and square distance of these generalized values is always either 0 or 1, regardless of the length of intersection. Let $gen(.)$ denote the generalization of a quasi-identifier attribute. Consider the generalizations of two numerical values n_1 and n_2, such that $gen(n_1) = r_1 = [35, 37)$ and $gen(n_2) = r_2 = [35, 99)$. Since $r_1 \neq r_2$, r_1 and r_2 are separate components of the feature vector. Assuming Hamming distance over categorical attributes, we have $r_1^T r_2 = 0$ and $\|r_1 - r_2\|^2 = 1$. However, $r_1 \cap r_2 \neq \emptyset$ and $0 \leq n_1^T n_2 < 35 \times 99$ and $0 \leq \|n_1 - n_2\|^2 < 64^2$. Therefore, representing ranges as categories may lead to significant inaccuracies in distance and dot product calculations. The other two heuristics try to address this problem. In the second heuristics, a range is represented by its middle point. In the third heuristics, a range is represented by two features that correspond to the lower and upper bounds of the actual value.

The possible three heuristics are thus as follows:

1) Each range is an independent category: A separate entry in the feature vector is created for all ranges of all generalized numerical attributes. Similar to those of categorical attributes, these features are binary. For example, if a numerical value is generalized to the range [1,37) then we will set the binary feature vector entry that corresponds to the category [1,37) of the corresponding attribute to 1. Any numerical value, that has not been generalized, is left unmodified.
2) Each range is replaced by the mean value: Assuming uniform distribution within the given range, generalized values are represented by mean value of the range. For example, the above range r_1 is replaced with the value 36, and r_2 is replaced with 67. Any numerical value, that has not been generalized, is left unmodified.
3) Each range is replaced with numerical *lower-bound* and *upper-bound* features: Every numerical value, generalized or not, is represented by two values corresponding to the lower bound and upper bound of the actual value, respectively. For example, the above range r_1 is represented as the 2-dimensional feature vector $v = \{35, 37\}$. A numerical value that has not been generalized, say 42, would be represented by $v\prime = \{42, 42\}$.

2) Categorical Attributes: Like for the case of numerical attributes, we have three different heuristics for categorical attributes. Our first heuristics for categorical attributes does not take the generalization hierarchy into account and assumes that each generalization, that is, non-leaf node of the value generalization hierarchy (VGH), is just another category. The other two methods do consider the VGH while building feature vectors.

1) Each generalization is an independent category: This heuristics is quite similar to the first heuristics for numerical attributes. Generalized values are considered a new category. Consider the generalization hierarchy for $Education$ attribute, given in Figure 1. Rather than representing $Education$ as a 7-category attribute (one category for each leaf node), we consider non-leaf nodes as well. In total, if generalized, $Education$ will be a 13-category attribute (one category for each node in the VGH).
2) Any value in the VGH exhibits all features of its parents: The set of features in this heuristics is similar to the previous one. However, instead of setting only one feature per value, all features on the path from the given (leaf or non-leaf) node to the root of the VGH are set. For example, value $9th$ is represented by the set of features {*9th, JuniorSec., Secondary, ANY*}. The generalized value $GradSchool$ is represented by {*GradSchool, University, ANY*}.
3) Generalized value exhibits features of all leaf nodes in the VGH that are its specializations: For this heuristics, the number of features per categorical attribute is equal to the number of leaf nodes in the VGH. Generalizations are captured by setting all features that may be specializations of the given value. For example, value $9th$ is represented by $\{9th\}$ only. The generalized value $SeniorSec$, on the other hand, is represented by $\{11th, 12th\}$.

C. Modeling Anonymized Data as Uncertain Data

The above heuristics can effectively represent anonymized data in many cases and thus support construction of effective classification models. A more general way to use anonymized data for classification purposes is to model it as uncertain data [16]. In other words, when we see an attribute with a range value [35-37), we may consider it as an uncertain value between 35 and 37.

As discussed in section III-A, all we need to do is to calculate kernel functions for anonymized data instances. This can be achieved by observing a few facts about anonymized data. First of all, the quasi-identifier attribute values of anonymized records are *imprecise* due to generalization but are always *accurate* in the sense that generalization values are chosen according to the original data values. Second, for the non-quasi-identifier attributes, we already have the original values, because these values are not generalized. Therefore, we need to compute the "expected" values of the kernel functions with the imprecise attributes.

Next we discuss how to use statistical characteristics of each attribute to compute the expected value of the required kernel function results for SVM and k-NN classifier construction.

Given generalized instances $gen(\mathbf{x}_i)$ and $gen(\mathbf{x}_j)$, our goal is to calculate $E(K(gen(\mathbf{x}_i), gen(\mathbf{x}_j)))$ for anonymized data instances for common kernel functions. Now let us define $X_d = gen(\mathbf{x}_i)^T gen(\mathbf{x}_j)$ (i.e., X_d is the random variable that represents the dot product of two anonymized instances) and $X_e = \|gen(\mathbf{x}_i) - gen(\mathbf{x}_j)\|^2$ (i.e., X_e is the random variable that represents the square distance between two anonymized instances). Using these notations, we can define the common kernel functions for generalized data as:

- **Linear:** $K(gen(\mathbf{x}_i), gen(\mathbf{x}_j)) = X_d$,
- **Polynomial:** $K(gen(\mathbf{x}_i), gen(\mathbf{x}_j)) = (\gamma X_d + r)^d$, where $\gamma > 0$,
- **Radial Basis Function (RBF):** $K(gen(\mathbf{x}_i), gen(\mathbf{x}_j)) = exp(-\gamma X_e)$ where $\gamma > 0$,
- **Sigmoid:** $K(gen(\mathbf{x}_i), gen(\mathbf{x}_j)) = tanh(\gamma X_d + r)$.

According to the Taylor theorem [26], we know that for any differentiable function $g(X)$, and for random variable X with $E(X) = \mu_X$ and $Var(X) = \sigma_X^2$, we can approximate $g(X)$ around μ_x as:

$$g(X) \sim g(\mu_X) + (X - \mu_X)g'(\mu_X) + \frac{1}{2}(X - \mu_X)^2 g''(\mu_X).$$

Such an approximation provides a first order approximation of the expected value of $g(X)$ as

$$E(g(X)) \sim E(g(\mu_X) + (X - \mu_X)g'(\mu_X)) = g(\mu_X).$$

Also we can obtain a second order approximation [26] of the expected value of $g(X)$ as

$$\begin{aligned} E(g(X)) &\sim E(g(\mu_X) + (X - \mu_X)g'(\mu_X)) \\ &\quad + E\left(\frac{(X - \mu_X)^2 g''(\mu_X)}{2}\right). \end{aligned}$$

Since each of the four above kernel functions is of the form $g(X_d)$ or $g(X_e)$, we can obtain a first order approximation of the expected value of the kernel function results as $g(E(X_d))$ or $g(E(X_e))$ for appropriate functions, g. For example, for a polynomial kernel, we can obtain a first order approximation of $E(K(gen(\mathbf{x}_i), gen(\mathbf{x}_j)))$ as $(\gamma E(X_d) + r)^d$ where $\gamma > 0$. Similarly, for RBF kernel, we can get a first order approximation of $E(K(gen(\mathbf{x}_i), gen(\mathbf{x}_j)))$ as $exp(-\gamma E(X_e))$ where $\gamma > 0$. Therefore, to compute $E(K(gen(\mathbf{x}_i), gen(\mathbf{x}_j)))$, we need to compute $E(X_d) = E(gen(\mathbf{x}_i)^T gen(\mathbf{x}_j))$ and $E(X_e) = E(\|gen(\mathbf{x}_i) - gen(\mathbf{x}_j)\|^2)$.

Let $E(X_d^t)$ and $E(X_e^t)$ denote expected dot product and expected square Euclidean distance on the t^{th} attribute respectively. Then, we can formulate $E(X_d)$ and $E(X_e)$ as a summation of attribute-wise expected values:

$$\begin{aligned} E(X_d) &= \sum_{t=1}^{m} E(X_d^t), \\ E(X_e) &= \sum_{t=1}^{m} E(X_e^t). \end{aligned}$$

We discuss how the components $E(X_d^t)$ and $E(X_e^t)$ can be calculated for numerical and categorical quasi-identifiers[1] next.

1) Numerical Quasi-identifiers: For numerical quasi-identifiers, let us first calculate $E(gen(\mathbf{x}_i)[t]^T gen(\mathbf{x}_j)[t])$ for $i \neq j$ as follows:

$$\begin{aligned} E(X_d^t) &= E(gen(\mathbf{x}_i)[t].gen(\mathbf{x}_j)[t]) \\ &= E(gen(\mathbf{x}_i)[t]).E(gen(\mathbf{x}_j)[t]). \end{aligned}$$

Also, we can easily compute $E(\|gen(\mathbf{x}_i)[t] - gen(\mathbf{x}_j)[t]\|^2)$ (i.e., $E(X_e)$) as follows:

$$\begin{aligned} E(X_e^t) &= E((gen(\mathbf{x}_i)[t])^2) \\ &\quad -2E(gen(\mathbf{x}_i)[t]).E(gen(\mathbf{x}_j)[t]) \\ &\quad +E((gen(\mathbf{x}_j)[t])^2). \end{aligned}$$

The above derivations indicate that we only need to evaluate $E((gen(\mathbf{x}_i)[t])^2)$ and $E((gen(\mathbf{x}_i)[t]))$ for numerical quasi-identifiers to support SVM or k-NN classification. Therefore, as long as we can efficiently estimate the first two moments of the generalized attributes, we do not need to actually know a probability distribution function. For example, for our purposes, all we need to know of some range $r = [1, 35)$ is its expected value and its variance (note that the second moment could be calculated from variance and expected value).

Similarly, we can compute $E(K(gen(\mathbf{x}_i)[t], \mathbf{x}_j[t]))$ for a generalized instance $gen(\mathbf{x}_i)$ and an original instance $\mathbf{x}_j$ by treating $\mathbf{x}_j$ as a constant. In other words, we can set $E((gen(\mathbf{x}_j)[t])^2) = \mathbf{x}_j[t]^2$ and $E((gen(\mathbf{x}_j)[t])) = \mathbf{x}_j[t]$ for the instance $\mathbf{x}_j$ in the above formulas.

2) Categorical Quasi-identifiers: SVM classifiers usually assume Hamming distance for categorical attributes. Therefore, dot product of two categorical values represents the probability that the values are the equal. Therefore, assuming that $gen(\mathbf{x}_i)[t]$ and $gen(\mathbf{x}_j)[t]$ are from the same domain S, $E(gen(\mathbf{x}_i)[t]^T gen(\mathbf{x}_j)[t])$ can be calculated for $i \neq j$ as follows:

Let $f_V(v)$ and $f_W(w)$ denote the probability mass functions of generalized values $V = gen(\mathbf{x}_i)[t]$ and $W = gen(\mathbf{x}_j)[t]$.

$$\begin{aligned} E(X_d^t) &= Pr(V = W) \\ &= \sum_{k \in S} Pr(V = k).Pr(W = k) \\ &= \sum_{k \in S} f_V(k).f_W(k) \end{aligned}$$

The square distance $E(X_e^t)$ can be derived from $E(X_e^t)$ as follows:

$$\begin{aligned} E(X_d^t) &= Pr(V \neq W) \\ &= 1 - E(X_d^t). \end{aligned}$$

Finally, we need to define the probability mass function of an original value $v \in S$, so that $E(X_e)$ and $E(X_d)$ are defined

[1]In the rest of this paper, the term "quasi-identifier set" refers to the quasi-identifier and "quasi-identifier" denotes its quasi-identifying attributes.

on arbitrary pairs of generalized and original values:

$$f_v(k) = \left\{ \begin{array}{ll} 1 & \text{if } v = k \\ 0 & \text{otherwise} \end{array} \right\}.$$

D. Releasing QI-statistics with Anonymized Data

Given no background information about the original data, calculating expected values of kernel functions or square distances can only rely on domain knowledge. Yet, it is highly unlikely that some probability distribution assigned by the domain experts will perfectly model data sets. Therefore in this section, we introduce the novel method of collecting such statistics during the anonymization process, to be released within the anonymized data. We refer to this groups of information as *QI-statistics*.

In section III-C, we have established that some statistical information about quasi-identifiers is sufficient to evaluate the expected value of kernel functions (or square distance for k-NN classification) accurately. Equipped with this knowledge, we propose an extension to existing anonymization methods such that in addition to generalized values of quasi-identifiers, the anonymized data should also contain one new attribute per quasi-identifier. This new attribute consists of expected value and variance for numerical quasi-identifiers and probability mass function for categorical quasi-identifiers.

There are multiple granularities at which statistical information about quasi-identifiers may be collected: per data set in distributed scenarios, per generalized value, per equivalence class and finally per record. In [21], Aggarwal shows that releasing probability distribution functions for each record in the data set is problematic in terms of privacy. Aggregating the statistical information over each generalized value or the entire data set would tackle that problem, but, does not yield the optimum results in terms of precision and privacy. Therefore, we choose the finest granularity that is guaranteed not to violate k-anonymity, that is, per equivalence class.

Below, we first review the k-anonymity definition and then show that calculating *QI-statistics* over each equivalence class does not violate k-anonymity.

Definition: (k-anonymity) A data set that satisfies the following conditions is said to be k-anonymous:

1) The data does not contain any unique identifiers.
2) Every record is indistinguishable from at least $(k-1)$ other records with respect to its quasi-identifier fields.

Theorem: Releasing *QI-statistics* with anonymized data does not violate k-anonymity.

Proof: Among the two conditions of k-anonymity, *QI-statistics* do not violate the first condition because the added information is not unique for any individual represented in the original data (unless of course $k = 1$, in which case privacy obviously is not a concern). In order to determine whether the second condition is violated, let us assume that *QI-statistics* are quasi-identifiers. Under this assumption, k-anonymity will not be violated either, since every member of an equivalence class shares the same values on the attributes that contain statistics of quasi-identifiers. By definition, all equivalence classes are of cardinality at least k. Therefore the second condition is satisfied.

R''	A_1	A_2	S_1	S_2
r''_1	Masters	[35-37)	P(Masters) = 1	$\mu_X = 35.6$, $\sigma^2_X = 0.22$
r''_2	Masters	[35-37)	P(Masters) = 1	$\mu_X = 35.6$, $\sigma^2_X = 0.22$
r''_3	Masters	[35-37)	P(Masters) = 1	$\mu_X = 35.6$, $\sigma^2_X = 0.22$
r''_4	SeniorSec.	[1-35)	P(11th) = 0.66, P(12th) = 0.33	$\mu_X = 27.6$, $\sigma^2_X = 20.22$
r''_5	SeniorSec.	[1-35)	P(11th) = 0.66, P(12th) = 0.33	$\mu_X = 27.6$, $\sigma^2_X = 20.22$
r''_6	SeniorSec.	[1-35)	P(11th) = 0.66, P(12th) = 0.33	$\mu_X = 27.6$, $\sigma^2_X = 20.22$

TABLE II
SAMPLE ANONYMIZED DATA SET WITH STATISTICS OF QIS

The above proof also applies to multiple other anonymity definitions. Among these are l-diversity [5] and t-closeness [6], which only introduce additional conditions regarding the distribution of sensitive attributes within an equivalence class.

Table II illustrates the proposed idea. Instead of releasing R' in Table I, the data holder adds one new attribute S_i per quasi-identifier A_i. If A_i is categorical (e.g., $Education$), then the corresponding S_i stores a probability mass function obtained from the equivalence class of each record. On the other hand, if A_i is numerical (e.g., Age), then S_i contains the mean and variance of A_i within the equivalence class of each record.

Although we have proven above that releasing *QI-statistics* does not violate k-anonymity, it might help the adversary infer the values of some quasi-identifiers. Consider a numerical attribute A_i and a categorical attribute A_j. If a record r has $\sigma^2_X = 0$ on *QI-statistics* S_i, then value of the quasi-identifier A_i is revealed: μ_X. Similarly, $Pr(C) = 1$ on S_j implies that the value of r on A_j is C.

Notice that *QI-statistics* provide extra information only if A_i and A_j are generalized. Therefore, unless the anonymization method of choice performs global recoding, inferring exact values of quasi-identifiers is not possible. Because a local recoding algorithm, under any circumstances, need not generalize a unanimous group of values. Nevertheless, an adversary can still obtain an advantage when σ^2_X is small or $Pr(.)$ is large. Assuming it really is necessary to protect quasi-identifiers beyond the k-anonymity definition, such advantage can be prevented by not releasing *QI-statistics* for some attributes of some records (i.e., if σ^2_X is below a threshold or $Pr(.)$ is above a threshold). Given no additional information, however, we would have to assume that the corresponding attribute follows some probability distribution.

IV. EXPERIMENTAL ANALYSIS

The goals of our experiments are to compare the accuracy of: (1) various classification models built over anonymized data sets and transformed using the different heuristics

proposed in section III-B; (2) the approach of modeling anonymized data as uncertain data using the expected distance functions proposed in section III-C. Our experiments also investigate the relationship between accuracy and various anonymization parameters: the anonymity requirement, k; the number of quasi-identifiers; and, specifically for the distributed data mining scenario, the number of data holders. Additionally, in order to assess how accuracy is affected by the types of available training and test data sets, we have performed experiments related to two different scenarios concerning the training and testing of the classifiers: (1) classifiers trained on original data, and tested on anonymized data; and, (2) classifiers trained on anonymized data and tested on original data. Our primary evaluation metrics is 3-fold average classification accuracy. Cross validation requires too much execution time for more than 3 folds.

We run experiments on Support Vector Machine (SVM) classification methods (section IV-B) and Instance Based learning (IB) methods (section IV-C). For implementation and experimentation purposes, we used the Java source code of the $libSVM$ SVM library, available at [25] and IB1 instance-based classifier from $Weka$ [27].

For our experiments, among the wide variety of available anonymization methods [4], [7], [9], [11], we selected $DataFly$ because it is one of the earliest and most widely known solutions. Another nice feature of this method is the simplicity of the privacy definition that relies only on k. The default anonymity level, k, was set to 32 for all data sets.

A. Data Sets and Value Generalization Hierarchies

We performed our experiments on the real-world $Adult$ and *German Credit* data sets from the UC Irvine Machine Learning Repository [28]. The $Adult$ data set, also known as the *Census Income* data set, contains around 50K records from the 1994 Census database. $Adult$ has been used heavily in previous work on anonymization. The *German Credit* data set contains credit information about 1K individuals. Such smaller data set, when the same level of anonymity as $Adult$ is enforced, shows the effects of over-generalization.

As a preprocessing step, we first merged the training and test data sets of $Adult$ into a single data set. Then we removed all tuples with missing values and shuffled the remaining 45,222 records. Since SVM classification is a time consuming process, in our experiments, we used the first 5K of the shuffled records. The default quasi-identifier set for $Adult$ consists of the attributes {*age, education, week-hours, native-country, capital-gain, workclass*}.

German Credit data set has not been preprocessed (other than random shuffling), since the data set does not contain any missing values. The default quasi-identifier set consists of the attributes {*marital-status (A9), age (A13), purpose (A4), residence (A11), job (A17), credit-amount (A5)*}.

Since we use generalization in our experiments, we adopted the VGHs developed for the $Adult$ data set for all categorical attributes from [7]. For numerical attributes, our VGHs consisted of balanced hierarchies of height equal to 3 and of equi-length intermediary nodes. We also generated VGHs for the *German Credit* data set, since there is no previous work on anonymization that uses this data set. For numerical attributes, we built balanced VGHs similar to those of $Adult$. For categorical attributes, on the other hand, we tried to build balanced VGHs with semantically meaningful generalizations.

Due to space limitations, we report in the paper a subset of our experimental results. For the $Adult$ data set, these results include SVM classification with RBF and linear kernel types and IB1 classification (i.e., poly and sigmoid kernels omitted). For the *German Credit* data set, we provide summary information in terms of average accuracy per classification method.

B. SVM classification

The input format of $libSVM$ requires that instances be expressed in terms of (feature, value) pairs. Following the guidelines in [25], we converted the data sets in tabular form to feature representation by (1) scaling numerical attributes to the range [-1, 1] and (2) representing an m-category categorical attribute using m distinct features. In this setting, the value of feature f that corresponds to the i^{th} attribute of an m-category categorical attribute is set to 1 if an instance exhibits the i^{th} value in the attribute domain[2]. The remaining $(m-1)$ features are set to 0. Please refer to section III-B for details and examples about the various heuristics for feature representation.

Throughout the SVM experiments, default parameters of kernel types provided by $libSVM$ were used. The only other factors that affect classification accuracy are those related to how the data is anonymized, that is: the anonymity requirement k, the number of quasi-identifiers $numQI$ and the number of data holders $numDH$. We discuss our measurements for each anonymization parameter in the next three subsections. For each experiment, default values are as follows: $k = 32$, $numQI = 6$, $numDH = 1$.

When $numDH > 1$, the preprocessed data set is first partitioned into $numDH$ equi-sized data sets, then anonymized according to the fixed anonymity requirement k and finally merged into a single data set. Although this setting assumes that data holders share the anonymity requirement k, data holders could have chosen different levels of anonymity without loss of generality.

As we have discussed in section III-B, there are $3 \times 3 = 9$ different combinations of methods to represent generalized attributes as feature vectors. We denote the classification accuracy measurements of each combination as $BLij$, where $i \in \{1, 2, 3\}$ and $j \in \{1, 2, 3\}$ refer to the categorical and numerical feature representation heuristics introduced in section III-B.1 and section III-B.2, respectively. For example, $BL13$ refers to the "baseline" method of representing ranges as independent categories (heuristics 1 in section III-B.1) and generalizations as a vector of their possible leaf categories (heuristics 3 in section III-B.2).

[2]We assume there exists some arbitrary order among the categories.

Accuracy measurements based on expected values of kernel functions are independent of how attributes are represented as feature vectors. Therefore, there is only one series of experiments for classification over expected value of the kernel function, denoted as Exp. Finally, SVM accuracy over the original data set is denoted as $Orig$.

In each figure of this section, we provide the accuracy of SVM classification on the original data set, $Orig$; our method based on expected values Exp; the best and worst baseline methods based on their average accuracy; and the average accuracy of all baseline methods, denoted as $AvgBL$.

1) Linear kernel: Linear kernel function requires that we compute the expected value of the dot product of two feature vectors. As discussed in section III-C, given the expected values of each feature of a feature vector, the expected value of the linear kernel can be computed accurately.

When the linear-SVM is trained on original data and tested on anonymized data, our proposed method of building SVM classifiers on expected values of the linear kernel, Exp, yields highly accurate models. In comparison to the classification accuracy over the original data, $Orig$, Exp is very resistant to generalization. Even at high levels of anonymity (see Figure 2(a)), classification degrades by at most %2. Figures 2(b) and 2(c) show that this observation also holds for various numbers of quasi-identifiers and data holders. The choice of feature representation heuristics employed plays a significant role under this experiment scenario. While the best baseline method, $BL2$2, achieves around %10 less accurate results than Exp, $BL1$3 can correctly classify only half as many records.

On the other hand, when the linear-SVM is trained on anonymized data and tested on original data, multiple baseline methods perform better than Exp. But the difference in accuracy is only around %5 in Figure 3.

Notice that in Figure 3(c), method $BL3$3 has no value for the test case where the data set is distributed among 6 data holders. $LibSVM$'s solver does not converge in this test case. This situation occurred only on $Adult$ data set when experimenting linear-SVM.

2) RBF kernel: We compute the expected value of RBF kernel function based on the expected value of the square distance of feature vectors. For that purpose, we used a first order approximation of the square distance. As was the case with linear kernels, this method still outperforms the baseline methods when the test data set is anonymized (see Figure 4). Furthermore, as shown in Figure 5, Exp performs better than the baseline methods when the models are built on anonymized data and tested on original data as well.

The advantage of Exp over baseline methods is smaller with RBF kernels (%2, compared to %10-20 of linear kernels). This is because the expected value of linear kernel can be calculated accurately, while we used a first order approximation to calculate RBF kernel's expected value. We believe a second order approximation of Exp would yield even higher accuracy.

3) German Credit data set: German Credit is very similar to $Adult$ in various aspects but the number of records. While the preprocessed $Adult$ dataset has 5K records, *German Credit* contains only 1K records. Yet, these data sets were anonymized to the same level in our experiments.

Method	Tr: Or., Ts: An.		Tr: An., Ts: Or.	
	Linear	RBF	Linear	RBF
Orig	75.575	71.771	75.575	71.771
BL11	72.218	70.096	73.333	70.170
BL12	71.898	70.130	73.873	70.123
BL13	71.391	70.150	73.420	70.150
BL21	73.900	71.718	73.613	71.217
BL22	73.860	71.378	73.967	70.730
BL23	73.206	71.284	73.279	71.111
BL31	73.987	71.765	72.599	70.797
BL32	74.040	71.411	73.360	70.530
BL33	73.306	71.264	73.093	70.850
Exp	74.627	71.978	73.520	71.197

TABLE III
AVERAGE ACCURACY OF SVM CLASSIFIERS OVER *German Credit*

Since *German Credit* is over-generalized, the accuracy does not vary much with different values of anonymization parameters. Therefore, we only discuss average accuracy measurements of the methods. Each value in Table III corresponds to the average of SVM classification accuracy on all experiment parameters (i.e., k, number of quasi-identifiers, number of data-holders).

Our results show that Exp performs better than BL in most test scenarios. In all four exceptions of the case (out of 80, considering sigmoid and poly kernels as well) BL methods perform marginally better than Exp: $BL3$3 with sigmoid kernel and $BL1$2, $BL2$1, $BL2$2 with linear kernel in the third column of Table III.

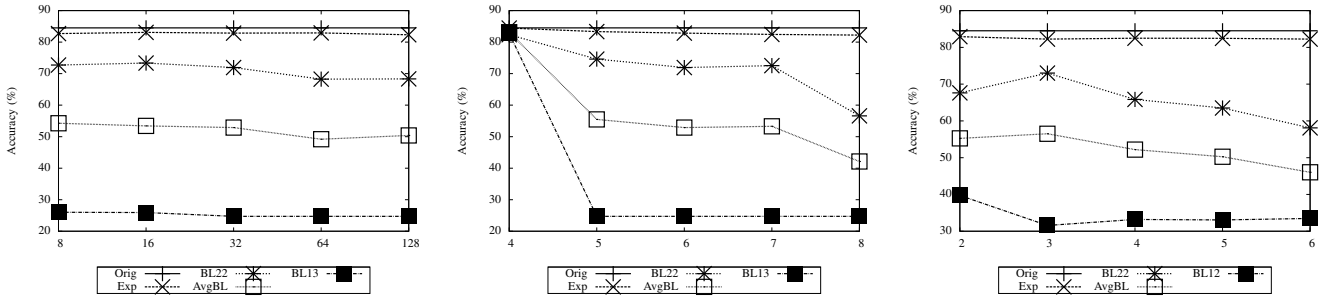
In Table III, the first two columns represent the accuracy averages of models built on original data when tested on anonymized data. The latter two are measurements of models built on anonymized data when tested on original data.

Polynomial and sigmoid kernels cannot separate the data set with any of the methods. The accuracy of both poly-SVM and sigmoid-SVM is almost always $\%70.07$. Since this was not the case with the $Adult$ data set, we attribute the situation to not having optimized the kernel parameters.

C. IB1 classification

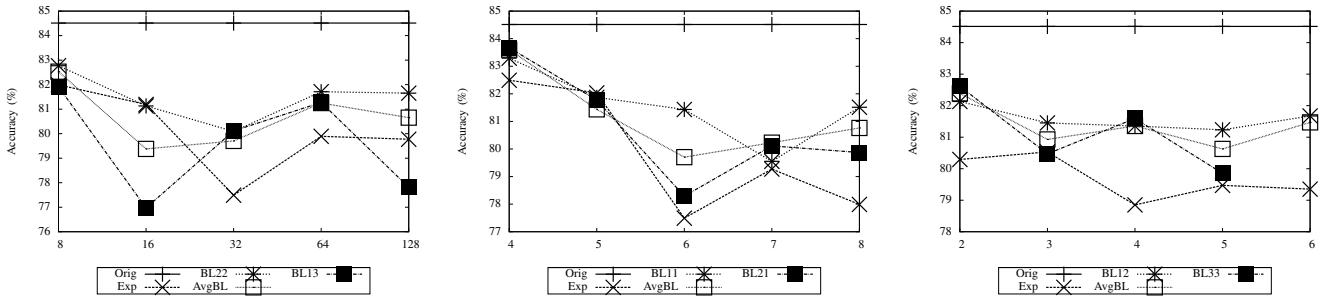
In addition to SVM classification, we were also interested in the performance of our proposed method on the much simpler IB1 classification method. Here, IB stands for "Instance-Based" classification, which contain k-NN classification. In $Weka$ library, IB1 represents 1-nearest neighbor (i.e., $k = 1$ for k-NN) classification that classifies instances according to the class value of their nearest neighbor [27]. Here, the nearest neighbor is chosen based on distance metrics that, in our case, consisted of Euclidean distance for numerical attributes and Hamming distance for categorical attributes.

The experiments conducted with IB1 classification serve multiple purposes. These experiments show that: (1) Our method applies to other data mining models based on dot product and distance functions (e.g., k-Medoids clustering). (2)



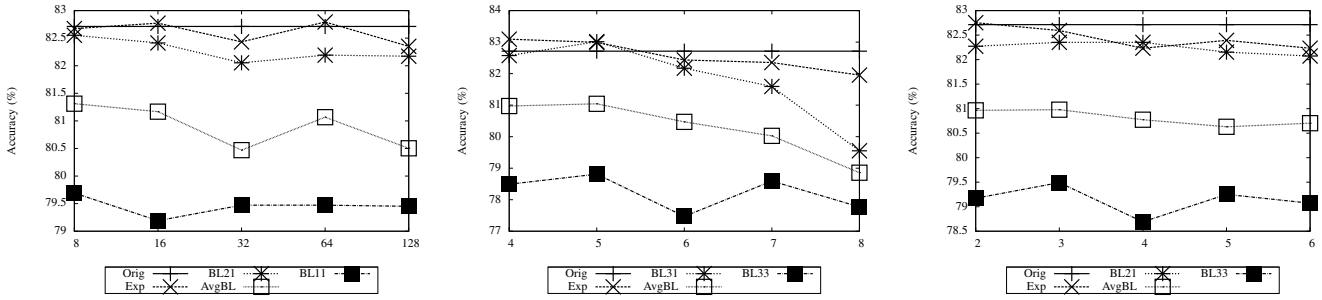
(a) Anonymity requirement, k (b) Number of QIs (c) Number of DHs

Fig. 2. *Adult* - Linear-SVM results when trained on original, tested on anonymized data



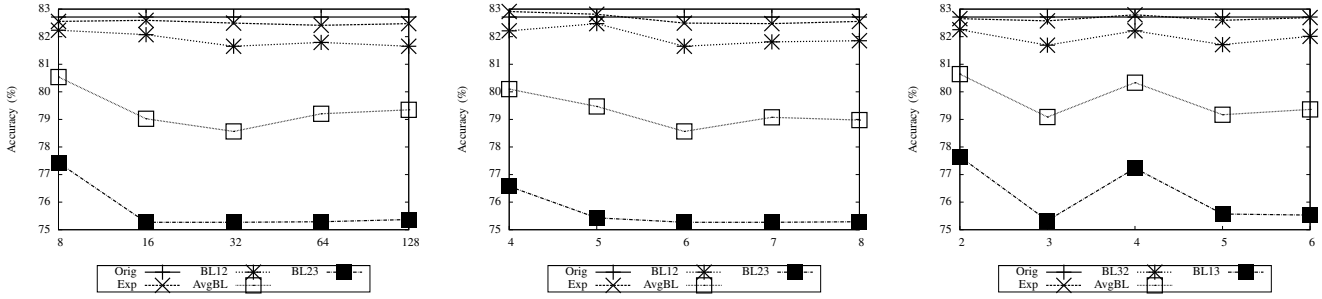
(a) Anonymity requirement, k (b) Number of QIs (c) Number of DHs

Fig. 3. *Adult* - Linear-SVM results when trained on anonymized, tested on original data



(a) Anonymity requirement, k (b) Number of QIs (c) Number of DHs

Fig. 4. *Adult* - RBF-SVM results when trained on original, tested on anonymized data



(a) Anonymity requirement, k (b) Number of QIs (c) Number of DHs

Fig. 5. *Adult* - RBF-SVM results when trained on anonymized, tested on original data

Method	Tr: Or., Ts: An.	Tr: An., Ts: Or.
Orig	70.270	70.270
BL	68.381	68.361
Exp	68.708	69.009

TABLE IV
AVERAGE ACCURACY OF IB1 CLASSIFIER OVER *German Credit*

Our method easily integrates with different data representation methods (i.e., feature vectors of $libSVM$, relations of $Weka$). (3) The advantage of our method is not restricted to powerful SVM methods. Our approach attains high accuracy with much simpler methods like k-NN as well.

We performed experiments on only one "baseline" method for IB1 experiments, that corresponds to heuristics 2 for numerical quasi-identifiers (see section III-B.1) combined with heuristics 1 for categorical quasi-identifiers (see section III-B.2). That is, any range of numerical values is replaced with the mid-point of the range and generalized categories are treated just like leaf categories. Other heuristics required multi-valued attribute support, which was not yet available in our version of $Weka$ library.

Experiment scenarios and anonymization parameters for IB1 classification are similar to SVM experiments. We denote the accuracy measurements of the baseline method as BL and the expected distance method as Exp. Accuracy of IB1 classification over the original data set is denoted as $Orig$.

1) Adult data set: Our results indicate that Exp outperforms BL on every test scenario. These results are similar to our results with RBF-SVM because both methods rely on square distance. Please refer to section IV-B.2 for a discussion of our results with RBF-SVM.

2) German Credit data set: A summary of the results is provided in Table IV. In the table, the first column provides the average accuracy of classifiers built on original data when tested with anonymized data. The second column provides the average classification accuracy of classifiers built on anonymized data when tested with original data. According to the results, our method performs marginally better than the baseline method. The results in Table IV are consistent with the results obtained with SVM classifiers. Our discussion of experimental results with RBF-SVM classification in section IV-B.3 applies here as well, since both methods are based on square distance function.

D. Discussion of the experimental results

Overall, the results indicate that our method performs better than heuristic methods. We found out that our method performs better with SVM kernels, whose expected value can be computed accurately (e.g., linear kernel, poly kernel and sigmoid kernel). Expected value calculation of RBF kernel function is less accurate because we use a first order approximation of square distance on numeric quasi-identifiers. Consequently, our method has less advantage over baseline methods with RBF kernels. This situation is also evident from IB1 experiment results, where classification is based only on square distance between pairs of records.

Another important result of the experiments is that the method used for representing generalized values has less impact on over-generalized data. This is evident from the experiments conducted with *German Credit* data set. Intuitively, as the data set is anonymized to higher levels, quasi-identifiers contain less information and classifiers rely more on non-quasi-identifying attributes. Since non-quasi-identifying attributes are not affected from anonymization, the performance of baseline methods are closer to our method.

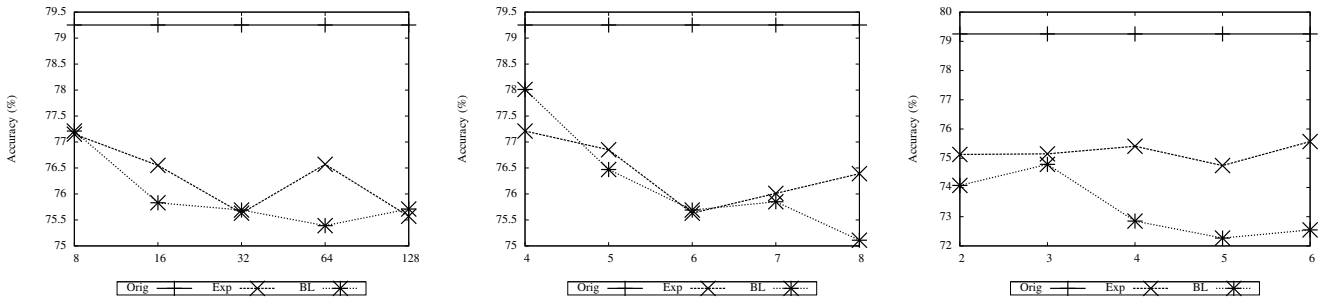
Among different feature representation heuristics, we averaged the accuracy measurements over all experiments in order to get an idea of best and worst heuristics. Our results indicate that replacing ranges with their mean value performs the best for numerical attributes (i.e., heuristics 2 in section III-B.1). For categorical attributes, representing any value by all features of its parents performs better than the other heuristics (i.e., heuristics 2 in section III-B.2). The heuristics with the least average accuracy was heuristics 3 for categorical attributes. In this heuristics, every generalized value is represented by all leaf categories that are its specializations. For numerical attributes, representing generalized values as independent categories, that is heuristics 1, performs significantly worse (%5 on the average) than the other heuristics. This is not very surprising because when encoded as a categorical attribute, the generalized numerical values can not be compared efficiently with the original values.

Based on our experimental results, for all different data mining tasks mentioned in section I (e.g., distributed data mining, building classifiers on anonymized data that will be used for classifying original data), anonymization does not significantly degrade classification. In all experimental results, both baseline methods and our method return models quite as accurate as models trained and tested over original data.

Our experimental analysis also indicates that, if the main concern is individual privacy, then building classification models over the union of locally anonymized data sets is an inexpensive but accurate option in comparison to cryptographic solutions for distributed data mining [1], [2]. Our experiments over multiple data-holder scenarios indicate that achieving high accuracy is possible under such scenarios.

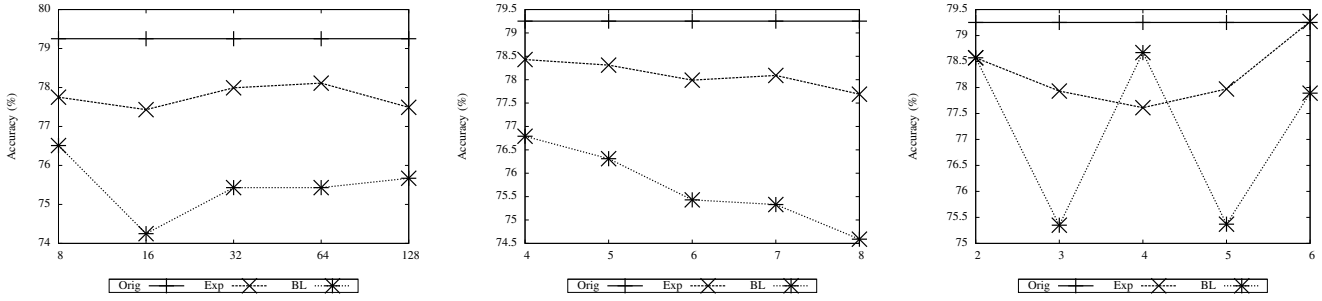
V. CONCLUSIONS

In this paper we extensively studied various alternatives for building classification models using anonymized data under different scenarios. We proposed a method for calculating the expected values of important functions such as dot product and square distance by modeling anonymized data as uncertain data. Our experiments indicate that by carefully releasing statistics of quasi-identifiers, it is possible to build models that can accurately handle anonymized data. We have also shown that locally anonymized data could directly be utilized in privacy preserving distributed data mining, without significant degradation of classification accuracy.



(a) Anonymity requirement, k (b) Number of QIs (c) Number of DHs

Fig. 6. *Adult* - IB1 results when trained on original, tested on anonymized data



(a) Anonymity requirement, k (b) Number of QIs (c) Number of DHs

Fig. 7. *Adult* - IB1 results when trained on anonymized, tested on original data

REFERENCES

[1] Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Journal of Cryptology*, vol. 15, no. 3, pp. 177–206, 2002.

[2] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1026–1037, 2004.

[3] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *ICDM '03*, Melbourne, FL, USA, 2003, pp. 96–106.

[4] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571–588, 2002.

[5] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in *ICDE '06*, Atlanta, GA, USA, 2006, p. 24.

[6] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *ICDE '07*, Istanbul, Turkey, 2007, pp. 106–115.

[7] B. Fung, K. Wang, and P. Yu, "Top-down specialization for information and privacy preservation," in *ICDE '05*, Tokyo, Japan, 2005, pp. 205–216.

[8] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: efficient full-domain k-anonymity," in *SIGMOD '05*, Baltimore, MD, USA, 2005, pp. 49–60.

[9] K. Lefevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *ICDE '06*, Atlanta, GA, USA, 2006, pp. 25–36.

[10] K.-L. Tan, B. Carminati, E. Ferrari, and C. Jianneng, "Castle: A delta-constrained scheme for k-anonymizing data streams," in *ICDE '08*, Cancún, México, 2008, pp. 1376–1378.

[11] X. Xiao and Y. Tao, "Anatomy: simple and effective privacy preservation," in *VLDB '06*, Seoul, Korea, 2006, pp. 139–150.

[12] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu, "Aggregate query answering on anonymized tables," in *ICDE '07*, Istanbul, Turkey, 2007, pp. 116–125.

[13] D. Kifer and J. Gehrke, "Injecting utility into anonymized datasets," in *SIGMOD '06*, Chicago, IL, USA, 2006, pp. 217–228.

[14] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu, "Utility-based anonymization using local recoding," in *KDD '06*, Philadelphia, PA, USA, 2006, pp. 785–790.

[15] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Workload-aware anonymization," in *KDD '06*, Philadelphia, PA, USA, 2006, pp. 277–286.

[16] W. K. Ngai, B. Kao, C. K. Chui, R. Cheng, M. Chau, and K. Y. Yip, "Efficient clustering of uncertain data," in *ICDM '06*, Hong Kong, China, 2006, pp. 436–445.

[17] H.-P. Kriegel and M. Pfeifle, "Hierarchical density-based clustering of uncertain data," in *ICDM '05*, Houston, TX, USA, 2005, pp. 689–692.

[18] C. K. Chui, B. Kao, and E. Hung, "Mining frequent itemsets from uncertain data," in *PAKDD '07*, Nanjing, China, 2007, pp. 47–58.

[19] C. K.-S. Leung, C. L. Carmichael, and B. Hao, "Efficient mining of frequent patterns from uncertain data," in *ICDMW '07*, Istanbul, Turkey, 2007, pp. 489–494.

[20] L. Xiao and E. Hung, "An efficient distance calculation method for uncertain objects," in *CIDM '07*, Honolulu, HI, USA, 2007, pp. 10–17.

[21] C. C. Aggarwal, "On unifying privacy and uncertain data models," in *ICDE '08*, Cancún, México, 2008, pp. 386–395.

[22] B. Martin, "Instance-based learning: Nearest neighbour with generalisation," Master's thesis, University of Waikato, Computer Science Department, Hamilton, New Zealand, 1995.

[23] J. Zhang, D.-K. Kang, A. Silvescu, and V. Honavar, "Learning accurate and concise naive bayes classifiers from attribute value taxonomies and data," *Knowl. Inf. Syst.*, vol. 9, no. 2, pp. 157–179, 2006.

[24] J. Zhang and V. Honavar, "Learning from attribute value taxonomies and partially specified instances," in *ICML '03*, Washington, DC, USA, 2003, pp. 880–887.

[25] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[26] P. Bickel and K. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics*, 2nd ed. Prentice Hall, 2000.

[27] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. Morgan Kaufmann, 2005.

[28] D. Newman, S. Hettich, C. Blake, and C. Merz, "UCI repository of machine learning databases," 1998.