Understanding Privacy Risk of Publishing Decision Trees

Zutao Zhu, Wenliang Du

Understanding Privacy Risk of Publishing Decision Trees*

Zutao Zhu and Wenliang Du

Department of Electrical Engineering and Computer Science
Syracuse University, Syracuse, NY 13244, USA
{zuzhu,wedu}@syr.edu

Abstract. Publishing decision trees can provide enormous benefits to the society. Meanwhile, it is widely believed that publishing decision trees can pose a potential risk to privacy. However, there is not much investigation on the privacy consequence of publishing decision trees. To understand this problem, we need to quantitatively measure privacy risk. Based on the well-established maximum entropy theory, we have developed a systematic method to quantify privacy risks when decision trees are published. Our method converts the knowledge embedded in decision trees into equations and inequalities (called constraints), and then uses nonlinear programming tool to conduct maximum entropy estimate. The estimate results are then used to quantify privacy. We have conducted experiments to evaluate the effectiveness and performance of our method.

1 Introduction

Decision tree is a powerful data mining tool that has been widely used for classification and prediction in many areas, including financial industry, military affairs, medical research, artificial intelligent, etc. Decision trees can also be used in data publishing, i.e., instead of publishing the raw data, data owners can publish the decision trees built from their raw data. This type of data sharing and dissemination can bring tremendous benefits to the society.

A critical concern faced by data publishing is privacy, because many of the data contain personal information. Decision trees, a form of aggregate information derived from the original dataset, can surely achieve a better privacy preservation than publishing the original data. However, as long as a decision tree is still useful, certain degree of private information is still embedded in it. It is well known that data mining results, such as decision trees and association rules, can lead to potential privacy breach, but it is not well understood how much private information is actually disclosed by a published decision tree. In other words, it is still an open problem to quantitatively measure how much private information is disclosed by decision trees.

* This work has partially supported by Awards No. 0618680 from the United States National Science Foundation.

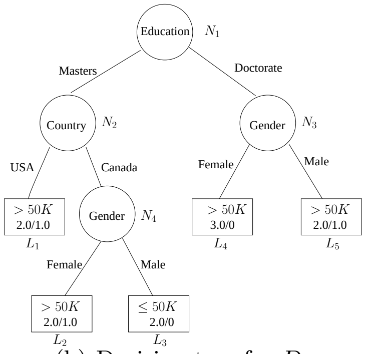
1.1 Motivation

We briefly introduce the decision tree, followed by two examples to demonstrate the potential privacy risk caused by the published decision trees.

Decision Tree. Consider table D_1 in Figure 1(a), which has four attributes *Education*, *Country*, *Gender*, and *Salary*. Attribute *Salary* is treated as sensitive and the data publishers want to ensure that no adversary can infer the salary of any individual with a relatively high confidence. We call this attribute a *Sensitive-Attribute* (SA). The other three attributes are often used to identify an individual. They are called *Quasi-Identifier* (QI) attributes. Usually, they can be acquired by the adversary from other sources [1]. Combined with the external data set, such as the voter registration list, an adversary can use *linking attack* [1, 2] to infer the salary of an individual. QIID refers to a distinct combination of QI attributes, i.e., if two people have identical QI values, their QIIDs will be the same. We use it simply for presentation purposes.

QIID	**Education**	**Country**	**Gender**	**Salary**
q_1	Masters	USA	Female	$\leq 50K$
q_2	Masters	USA	Male	$> 50K$
q_3	Masters	Canada	Male	$\leq 50K$
q_3	Masters	Canada	Male	$\leq 50K$
q_4	Masters	Canada	Female	$\leq 50K$
q_4	Masters	Canada	Female	$> 50K$
q_5	Doctorate	Canada	Female	$> 50K$
q_5	Doctorate	Canada	Female	$> 50K$
q_6	Doctorate	USA	Male	$\leq 50K$
q_6	Doctorate	USA	Male	$> 50K$
q_7	Doctorate	USA	Female	$> 50K$

(a) Microdata D_1



(b) Decision tree for D_1

Fig. 1. Dataset and Decision Tree

Figure 1(b) is a decision tree inducted from the data depicted in Figure 1(a) using ID3 [3] algorithm. Each circle is an internal node, which denotes a test on an attribute. The most informative attribute is selected as the test attribute depending on the attribute selection measure. Branches from a circle denote the outcome of the test. Each rectangle is a leaf node, which holds a class label. The number of tuples a and the misclassified tuples b are listed in the form of "a/b" for each leaf node. The tree predicts whether a person earns less than 50K based on the education, country, and gender information. For any tuple X whose class label is unknown, we can test the attribute values of X against the decision tree. We can trace a path from the root node to a leaf node. The leaf node has the class prediction for X. For instance, the path p, $N_1 \rightarrow N_3 \rightarrow L_4$, states that the probability that the female doctorates earn more than 50K is 100%.

Privacy Issues. As long as a decision tree contains useful aggregate information so that it can be used to predict future data, certain degree of private individual information for the training data is still embedded in it. For the path p: $N_1 \rightarrow$

$N_3 \rightarrow L_4$ that is induced from the training dataset D_1, not only can it predict future tuples, but also disclose the salary information of some tuples in D_1.

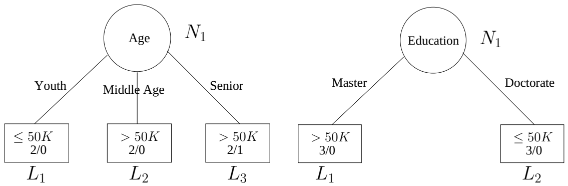
We assume that the class attribute of the tree contains sensitive information and adversaries have the QI part of the data in Figure 1(a). Also, we assume that adversaries know that the domain of $Salary$ is $\{\leq 50K, > 50K\}$. Based on these assumptions, adversaries can learn the private information of others:

For the perfect classified nodes, such as L_3 and L_4, the private information (salary) for q_3, q_5, and q_7 is completely disclosed. For example, we can infer that the salaries of q_5 and q_7 are $> 50K$ because q_5 and q_7 are female doctorates. If q_7 is linked to Alice according to the external data source, such as the voter's registration list, her salary is disclosed. Other leaf nodes are not perfectly classified; they only carry aggregate information for a group of individuals. Do they only describe the aggregate information as it labels in the leaf node? For example, the leaf node L_5 is label with "$> 50K$ (2.0/1.0)". Do we only learn that the probability that the male doctorates earn more than 50K is 50%? The answer is NO. In the following example, we show that the adversaries can derive more information when the internal (i.e. non-leaf) nodes are taken into consideration.

Example 1. Figure 2(b) is a decision tree built from the dataset depicted in Figure 2(a) using ID3 [3] algorithm. Surprisingly, having the above assumptions, we can derive the sensitive value for each tuple with 100% confidence. From the leaf nodes L_1 and L_2 in Figure 2(b), we can derive that the sensitive values for q_1 and q_4 in Figure 2(a) are $\leq 50K$ and $> 50K$, respectively. For the leaf node L_3, we learn that the sensitive value of q_2 and q_3 are different. One is $\leq 50K$ and the other is $> 50K$. We make a guess. If the SA of q_2 were $> 50K$ and the SA of q_3 were $\leq 50K$, *Education* would have been selected as the splitting attribute for the internal node N_1 because the split on *Education* can lead to the most informative result. Masters would have all been classified to $> 50K$ while doctorates $\leq 50K$. The decision tree would have been built as Figure 2(c). However, *Age* is the selected attribute instead. This indicates that our guess is incorrect, and therefore, the SA of q_2 is $\leq 50K$ and the SA of q_3 is $> 50K$.

QIID	Age	Education	Salary
q_1	Youth	Doctorate	$\leq 50K$
q_1	Youth	Doctorate	$\leq 50K$
q_2	Senior	Masters	$\leq 50K$
q_3	Senior	Doctorate	$> 50K$
q_4	MiddleAge	Masters	$> 50K$
q_4	MiddleAge	Masters	$> 50K$

(a) Microdata D_2



(b) Real decision tree (c) Unreal decision tree

Fig. 2. Dataset and decision trees for Example 1

Example 1 shows that an individual of a group does not necessarily follow the aggregate information of the group. We can capture more precise information for a single one than what is labeled in the leaf nodes, when some analysises are performed.

Challenges. For a simple data set and a simple decision tree, we can use manual deduction as above to derive private information. In a realistic scenario, the dataset often have many tuples and decision trees can become quite complicated. It is infeasible to manually derive private information like what we have done in the previous examples. We need a *systematic* method to analyze privacy; the analysis results will help us understand the privacy risk of decision-tree publishing, and thus improve our practice in data publishing. Once the data publishers understand the privacy situation, they can take actions to preserve it rather than directly publishing a raw tree. Some decision trees are published simply because privacy is not placed enough emphasis on. Therefore, we want to study the open problem: *how much private information the adversaries can infer from the published decision tree given the above assumptions*?

We face two challenges to understand the privacy risk of a published decision tree. First, we have to formulate the information in the leaf nodes. There are many forms of a decision tree: some may publish the accurate error rate as well as the class label while some only have the class label. We need to find a generic formulation to accommodate the various types of information. Second, we need to capture the explicit information in the internal nodes. That is, the most informative attribute is selected.

1.2 Overview of Our Approach

We model the privacy quantification as a Non-Linear Programming (NLP) problem, in which $P(SA \mid QI)$ for each QI and SA combination is represented by a variable. We formulate all the knowledge available to adversaries as linear and nonlinear equations (or inequalities) of these variables. We call them the constraints. Estimating $P(SA \mid QI)$ now becomes finding the values for these variables such that all the constraints are satisfied. Very likely, many solutions exist. However, we are not interested in finding just any solution, we are interested in finding a solution that achieves the most unbiased estimate of $P(SA \mid QI)$. This is exactly what can be achieved by using the maximum entropy theory.

Based on this well-established theory, we propose a systematic method to quantify the privacy disclosure risk in decision trees. The focus of this method is how to formulate constraints from all the information available to adversaries. Once the constraints are formulated, finding the maximum entropy solution is given to software tools that are called *solvers*. There are a number of powerful solvers (in particular, non-linear programming solvers) that we can choose. With this systematic method, we are not only able to analyze privacy disclosure risk in a decision tree; more importantly, we are able to help data publishers reduce their privacy risk when publishing their decision trees.

The rest of the paper is organized as follows. The related work is reviewed in Section 2. Section 3 formally defines the problem. Section 4 presents our main method. Section 5 evaluates our method using a real dataset. Section 6 concludes the paper and describes the future work.

2 Related Work

Privacy-preserving data publishing (PPDP) has been extensively studied in the literatures. The goal of PPDP is to publish a disguised version of the original data, such that the private information of the original data is preserved, while the data are still useful. Several methods have been proposed, including generalization [4–6], bucketization [7, 8], and randomization [9–11].

Understanding privacy is one of the essential tasks in PPDP. The goal of this research is to develop metrics to quantify privacy in data publishing. A number of metrics have been proposed, including K-anonymity [4], L-diversity [12], (α, k)-anonymity [13], t-Closeness [14], and m-invariance [15]. Our work fits into this line of studies. The major difference between our work and others is two-fold. First, instead of proposing a new metric, we focus on computing the conditional probability between QI attributes and SA attributes, i.e., $P(SA \mid QI)$. This conditional probability is a building block for most of the existing metrics. Once we can compute this probability, we can adopt the existing metrics to quantify privacy. Second, the existing privacy metrics are intended for data publishing, while the method proposed in this paper targets the publishing of decision trees. Computing $P(SA \mid QI)$ from a dataset (disguised in most cases) is significantly different from computing the same probability from the decision trees.

The privacy consequence of data mining results is studied by Kantarcioglu et al. [16]. This work tries to understand when data mining results violate privacy. The assumption of the work is that the classifier is kept invisible from adversaries, and adversaries can only request an instance be classified by the owner of a classifier, without knowing other information about the classifier. Although this model has its own merit in the client/server model, where mining results are kept at a sever, the scenario it models is quite different from ours. In our work, decision trees are fully accessible to adversaries. Their work performs a black-box analysis while ours is a white-box analysis.

Another area closely related to PPDP addresses how multiple parties can conduct data mining using their joint data, without disclosing to each other their private data. This line of research uses secure multi-party computation (SMC) protocols to protect private information [17, 18]. What is not addressed by SMC studies is how much private information is actually disclosed by the computation results. SMC guarantees that no one in the protocol knows more than what they can derive from the results; however, the results themselves might disclose enough private information. Analyzing how much private information is disclosed by decision trees is exactly the objective of this paper.

Applying the maximum entropy model to estimate privacy is first explored by Du et al. in [19]. They discuss the effect of background knowledge in privacy-preserving data publishing. The work here is dedicated to solve a significantly different problem, that is, to understand the privacy breach when a decision tree is published. Besides, the modeling processes differ far from each other. In [19], all the constraints are explicit according to the disguised dataset. For decision trees, not only do we need to consider the information explicitly in decision trees, we also need to consider the implicit information in decision trees that might cause privacy disclosure.

Another feature of our work is that we do exploit the knowledge about decision-tree building algorithm when deriving private information from a published decision tree. Exploiting the knowledge about algorithms to find private information has also been pursued in several existing studies. Wong et al. [20] explore that adversaries can take advantage of this feature to perform *minimality attack*. Similar attacks are also described by Zhang et al. in [21], and Zhu et al. in [22]. These attacks are based on the information that is not published, but is implied from the published information. Our work follows a similar approach, but the way how we exploit the knowledge of algorithms is quite different from the existing work.

3 Problem Formulation

Assumptions. We make several assumptions in this paper. We assume that the training set consists of two parts: QI attributes and SA attributes. The QI part consists of the information that can also be obtained from other sources. The SA part consists of the information that the data owner wants to protect. This is a general assumption in the field of PPDP. We assume that adversaries have all the data of the QI attributes. This assumption is made because the information in the QI part can be usually obtained via other means [4]. Although in practice, attackers might not know every QI value, this assumption allows us to conduct analysis on the worse-case scenario. For the sake of simplicity in this paper, we assume that there is one SA attribute in the training set, and this attribute is used as the class attribute in a decision tree. We assume that adversaries have the knowledge of the domain of the sensitive attributes, i.e., they know all the possible values of the sensitive attributes. In a decision tree, all the leaf nodes have class labels which are SA values. It is reasonable to make this assumption.

Measuring Privacy. How successful the adversaries can derive an individual's correct SA value depends on the intrinsic conditional probability between QI and SA attributes, i.e., $P(SA \mid QI, \mathcal{O})$, where $\mathcal{O}$ represents all the information available to the adversaries. In most of the existing studies, $\mathcal{O}$ consists of the information from sanitized datasets [4,7,12–14]. In our study, it also comes from the decision trees. For the sake of simplicity, we omit $\mathcal{O}$ from our notation, and only use $P(SA \mid QI)$ in the rest of the paper. Our privacy quantification task can be formally defined as the following:

Problem 1. Let D be the training data set that is used to generate the decision tree(denoted as Ω). Let variable X represent SA attributes, and variable Q represent QI attributes. Given Ω and the QI part of all the tuples in D, derive $P(X \mid Q)$ for all the combinations of Q and X values.

The value of $P(X \mid Q)$ is the primitive behind all the existing privacy measures, i.e., as long as we can compute this conditional probability, we can calculate the existing privacy metrics, such as L-diversity [12], (α, k)-anonymity [13], etc.

Maximum Entropy Modeling. The problem to measure privacy boils down to estimate the distribution of $P(X \mid Q)$, i.e., to assign a probability value to every variable $p(x \mid q)$, where $x \in X$ and $q \in Q$. Such assignment must be consistent with the decision trees that are published. Very likely, there are more than one distributions (we call them solutions) that are consistent with the published decision trees. However, we can only choose one among these distributions; the question is which one should be used to quantify privacy.

There are many ways to choose among these solutions. One way is to choose the most informative solution. For example, we can choose a solution that has $p(x \mid q) = 1$ for many x's and q's, as long as it is consistent with the published decision tree. If we use this solution to quantify privacy, the privacy score will not be very good, because for these people with $QI = q$, there is no uncertainty at all for the SA attribute. Therefore, the uncertainty of this solution is low. The question is whether selecting this solution is fair. If we have multiple choices, one having a higher uncertainty and the other a lower uncertainty, to choose a solution with lower uncertainty actually assumes some information we do not possess, and is thus biased. The maximum entropy theory answers the above question quite nicely. It says that based on the given information, the most unbiased estimate of a distribution is the one that maximizes the entropy [23]. Based on this principle, our problem becomes finding a distribution of $P(X \mid Q)$, such that the following conditional entropy $H(X \mid Q)$ is maximized:

$$H(X \mid Q) = -\sum_{Q,X} P(Q)P(X \mid Q) \log P(X \mid Q).$$

Obviously, when there are no constraints, the uniform distribution is the solution that maximizes the entropy. However, the published decision trees do give us a lot of constraints, i.e., the estimated distribution must be consistent with the tree structure, information at the leaf nodes, information at the internal nodes, etc.

To apply the maximum entropy theory to estimate $P(X \mid Q)$, we need to translate all the available knowledge into equations and inequalities using the word of $P(X \mid Q)$. The translation results become our constraints. With these constraints, we can model our privacy quantification problem as the following:

Definition 1. *(Maximum Entropy Modeling) Finding an assignment for $P(X \mid Q)$ for each combination of Q and X, such that the entropy $H(X \mid Q)$ is maximized, while all the constraints k_1, ..., k_n are satisfied, where constraint k_i is obtained via information that we have on decision tree mining process and results.*

Maximum entropy modeling problem is a special case of the NLP problem. There are sophisticated tools that can be used to solve NLP problems, such as KNITRO [24].

4 Deriving Constraints from Decision Tree Classifiers

To apply the Maximum Entropy theory to estimate the information disclosure, we need to understand where we can derive constraints; namely, we need to understand what adversaries know. They obviously know the published decision tree; it is quite likely that they also know the underlying algorithm used to build the decision tree, in particular, the attribute selection measure (e.g. Information Gain or Gini Index). Moreover, we assume that adversaries know the QI part of the training dataset. Therefore, the source of the constraints can be categorized into the following: the leaf nodes of the decision tree, the internal nodes which encode the attribute selection measure, and the QI part of the dataset.

In the following subsections, we describe how to derive constraints from these three sources. We use the example depicted in Figure 1(a) and 1(b) to help us explain our ideas in this section. We frequently use the following two terminologies in our explanation. An *attribute prefix* of a node V in a decision tree is a conjunction of attribute assignments that represents the path from the root to V. We use Λ to denote attribute prefix. A conjunction expression is said to be a *full conjunction expression* if it contains all the QI attributes of the dataset. Without causing any confusion, we simply call it *full expression*. For example, the attribute prefix of the node L_1 in Figure 1(b) is $\Lambda = (Education = Masters) \wedge (Country = USA)$. Λ is not a full expression because it does not contain all the QI attributes.

It should be noted that in our maximum entropy model, we need the entire QI attributes in our constraints, not a subset of it, i.e., each Q in our variable $P(X \mid Q)$ must be a full expression. We show how to represent $P(X \mid \Lambda)$ (where Λ is not a full expression) using $P(X \mid Q)$, where Q's are full expressions. Let Λ represent an attribute prefix of a node V. Let $q_1, \ldots, q_n$ be all the full-expression QIs that satisfy Λ, i.e., they share the same attribute prefix values. For example, if $\Lambda = (Education = Masters) \wedge (Country = USA)$, q_1 and q_2 in Figure 1(a) satisfy Λ because their *Education* and *Country* attributes satisfy Λ. Based on the conditional probability definition, we have the following:

$$P(X \mid \Lambda) = \frac{\sum_{i=1}^{n} P(X \mid q_i) P(q_i)}{P(\Lambda)}. \tag{1}$$

$P(\Lambda)$ and $P(q_i)$ are constants that are known to the adversaries [1]. Armed with Equation (1), we will not pay attention to whether Λ is a full expression or not in the rest of this paper.

4.1 Leaf Nodes

The most obvious source of privacy disclosure in a published decision tree is the leaf nodes, because leaf nodes contain a lot of information, including class labels and sometimes error rates (the error rate indicates the percentage of the misclassified tuples for each leaf node). We show how to derive constraints with or without error rates.

[1] We assume that adversaries know the QI part of the training dataset.

When error rates are published in a decision tree, the percentage of the correctly classified tuples becomes known. Let e represent the error rate of a leaf node whose attribute prefix is Λ, and let C be the class label of this leaf node. We can derive the following constraint (we call it *rate-constraint*):

$$P(C \mid \Lambda) = (1 - e).$$

Furthermore, the fact that C is selected as the class label indicates that C is the most frequent class among all the classes. Therefore, we can infer that within any leaf node, the percentage of tuples with class label C is larger than those with other class labels. Namely, we have the following constraint (called *label-constraint*):

$$P(C \mid \Lambda) \geq P(W \mid \Lambda), \quad \text{for } \forall\ W \neq C.$$

For example, according to the leaf node L_4 in Figure 1(b), the attribute prefix Λ is {Education = Doctorate and Gender = Female}, and two q_5 tuples and one q_7 tuple in Figure 1(a) are included in L_4. Because the number of mis-classified tuples in L_4 is 0, the error rate e of L_4 is 0. Since the class label is "$> 50K$", we can derive the following rate-constraint:

$$P(> 50K \mid \Lambda) = 1, \quad \text{or } P(\leq 50K \mid \Lambda) = 0,$$

and the following label-constraint:

$$P(\, > \texttt{50K} \mid \Lambda) \geq P(\leq \texttt{50K} \mid \Lambda).$$

Note that in the above example, the label-constraint is redundant. Actually, when there are only two class values, the rate-constraint always implies the label-constraint, because if C is the selected class label for a leaf node, we know $P(C \mid \Lambda)$ is always ≥ 0.5, larger than the other class value that is not selected. However, when there are more than two class values, the error rate might be larger than $P(C \mid \Lambda)$. Therefore, the rate-constraint alone does not always capture the fact that $P(C \mid \Lambda)$ is the largest among all the class values; the label-constraint captures that.

In practice, data publishers might not publish the error rates, i.e., each leaf node is only assigned a class label without a corresponding error rate. In this case, adversaries can only infer the *label-constraint*, not the *rate-constraint*.

4.2 Internal Nodes

In a decision tree, the internal nodes do not seem to contain much information that can lead to privacy disclosure, but actually, they do: the fact that a specific attribute is used as the partition attribute can tell us some information about the training dataset. To use this fact in our maximum entropy model, we need to derive constraints from these internal nodes.

In a decision tree, each internal node represents a subset of tuples that share the same values for certain attributes; these attributes and their values are encoded by the path from the root to this internal node. We use the attribute prefix Λ to represent these attributes and their values (not including the node that is to be splitted). Generally speaking, in decision-tree induction algorithms, at each internal node, an attribute needs to be selected to further partition the records contained in the internal node. The goal of the selection measure is to find the best way to split the tuples such that the expected *impurity* score of the partition is minimized. The following notations are commonly used in decision-tree algorithms.

- $I(\Lambda)$: Impurity score of the node that corresponds to Λ.
- $I(\Lambda, A = A_i)$: impurity score of the node that corresponds to Λ and $A = A_i$. Without causing confusions, we shorten $I(\Lambda, A = A_i)$ as $I(\Lambda, A_i)$.
- $E(\Lambda, A)$: Expected impurity score of using attribute A to partition the node that corresponds to Λ. $E(\Lambda, A)$ is computed using the following formula:

$$E(\Lambda, A) = \sum_{i=1}^{|A|} I(\Lambda, A_i). \tag{2}$$

According to the attribute selection method in the decision tree induction algorithm, the attribute having the best impurity score will be selected as the splitting attribute for the node. Therefore, by seeing that T is the selected attribute at an internal node (say N), we know that the expected impurity score achieved by using T to partition node N is less than that using any other candidate attribute. Let Λ be the attribute prefix of the node N, and let Ψ represent the candidate attributes at node N. We have the following constraint, called *internal-constraint*:

$$E(\Lambda, T) \leq E(\Lambda, W), \quad \text{for } \forall\ W \in \Psi - \{T\}. \tag{3}$$

The actual computation of expected impurity depends on how impurity is measured. Several methods have been used to measure *impurity*, including entropy [3,25] and Gini impurity [26]. In the following, we instantiate Inequality (3) for both entropy-based and Gini impurity measures. At the end, we will get a set of constraints that will be integrated into our Maximum Entropy model.

(1) Gini Impurity Measure. Gini impurity depends on squared probabilities of membership for each target category in the node, which is used by the CART algorithm [26]. Its minimum, zero, is reached when all cases of a node fall into the same category, i.e., the purest case. Gini impurity for a branch that corresponds to Λ and $A = A_i$ is computed in the following formula:

$$I(\Lambda, A_i) = \frac{|D_{\Lambda,A_i}|}{|D_\Lambda|}(1 - \sum_{j=1}^{|C|} P(C_j \mid \Lambda, A_i)^2), \tag{4}$$

where the term $|D_{\Lambda,A_i}|/|D_\Lambda|$ is the weight of the i-th partition.

In our maximum entropy modeling, $P(C_j \mid \Lambda, A_i)$ in the above equation is unknown to adversaries because it is a combination of several variables that are what adversaries want to estimate. Although adversaries cannot estimate these values directly, they can use the information from internal nodes to capture the relationship among these variables. The relationship is captured in Inequality (3) after we combine Equations (4) and (2) together.

We use an example to illustrate the constraints derived from internal nodes. Assume that Gini Index measure is used to generate the tree depicted in Figure 1(b). For the internal node N_2 in Figure 1(b), *Country* and *Gender* are the candidate attributes because *Education* has been used in N_1. Since *Country* is the selected attribute, the Gini Index impurity deduction of *Country* is larger than that of *Gender*. That is, the expected impurity score of *Country* is less than that of *Gender*.

Let Λ be $Education = Masters$; C_1 be $Country = USA$; C_2 be $Country = Canada$; G_1 be $Gender = Female$; G_2 be $Gender = Male$; Let variable p_{i0} represent $P(\leq 50K \mid q_i)$, and let variable p_{i1} represent $P(> 50K \mid q_i)$, where i ranges from 1 to 4 in our example because q_1, q_2, q_3, and q_4 satisfy Λ. For the *Country* attribute, we have the following:

$$I(\Lambda, C_1) = \frac{1}{3}\left[1 - \frac{(p_{10}+p_{20})^2}{2^2} - \frac{(p_{11}+p_{21})^2}{2^2}\right],$$

$$I(\Lambda, C_2) = \frac{2}{3}\left[1 - \frac{(p_{30}+p_{40})^2}{2^2} - \frac{(p_{31}+p_{41})^2}{2^2}\right].$$

For the *Gender* attribute, similarly, we can get $I(\Lambda, G_1)$ and $I(\Lambda, G_2)$.

Using Inequality (3), we have the following *internal-constraint*:

$$I(\Lambda, C_1) + I(\Lambda, C_2) \leq I(\Lambda, G_1) + I(\Lambda, G_2).$$

(2) Entropy-based Impurity Measure. Entropy is used to measure the impurity of a node in some decision tree mining algorithms, such as ID3 and C4.5 [25]. Information gain is based on the concept of entropy used in the information theory. The entropy of the *i*-th branch of a partition using attribute A can be calculated as the following:

$$I(\Lambda, A_i) = \frac{|D_{\Lambda, A_i}|}{|D_\Lambda|}\sum_{j=1}^{|C|} -P(C_j \mid \Lambda, A_i)\log P(C_j \mid \Lambda, A_i).$$

Similar to the Gini Index measure, we combine the above equation with Equation (2) for each candidate attribute, and then we apply the results to Inequality (3), which captures the relationships among several variables corresponding the the internal node.

4.3 Deriving Constraints from Quasi-Identifiers

In our maximum entropy modeling, each variable $P(X \mid Q)$ is a conditional probability, so they must satisfy all the constraints imposed on probabilities. For example, the sum of all conditional probabilities given a specific q_i should be 1. We need to explicitly provide these constraints, so the solutions of our maximum entropy modeling will be meaningful with regard to probabilities.

Similar to [22], we have the following *QI-constraints*:

$$\sum_{i=1}^{m} P(X = x_i \mid Q = q) = 1. \tag{5}$$

If the distribution of SA values are also published along with the decision tree, adversaries will know $P(X = x)$, so we will have the following *SA-constraints*:

$$\sum_{i=1}^{n} P(q_i \mid X = x) = \sum_{i=1}^{n} \frac{P(X = x \mid q_i)P(q_i)}{P(X = x)} = 1,$$

where $P(X = x)$ is the probability of x in the training data set.

5 Experiments

To demonstrate how much sensitive information is disclosed by decision tree classifiers, we evaluate our proposed method using the Adults dataset from the UC Irvine Machine Learning Repository [2]. We use the same setting described in [22]. However, we choose the "Education" attribute as the class attribute. As a result, we have a dataset D, which has 30162 records, with 4480 distinct QI values and 16 distinct SA values. Therefore, we have 4480 *QI-constraints*. We use Gini Index and Information Gain measures to build two decision trees. The number of *rate-constraints*, *label-constraints*, and *internal-constraints* are 518, 7770, 1097, for Gini Index; and 2983, 44745, 3307, for Information Gain, respectively. Our ME method is implemented using C++ and Oracle 9i. All experiments are run on an Intel(R) Pentium(R)-D machine with 3.00 GHz CPU and 4GB physical memory. We use the KNITRO software package [24] to solve our Maximum Entropy Estimation problem.

The output of the program is the estimate of $P(SA \mid QI)$ for all combinations of SA and QI values, based on the information provided by the published decision tree. The closer our estimate is to the original distribution, the more private information is disclosed via the published decision tree. We measure such closeness at two different levels: individual level and overall level, as is described in [22]. They are

$$\mathrm{D}_{individual} = \sum_{x \in SA} P(x|q) \log \frac{P(x|q)}{P^*(x|q)},$$

$$\mathrm{D}_{overall} = \sum_{q \in QI} [P(q) \cdot \sum_{x \in SA} P(x|q) \log \frac{P(x|q)}{P^*(x|q)}],$$

respectively, where $P^*(X \mid Q = q)$ is the *estimated individual distribution*, and $P(X \mid Q = q)$ is the original distribution.

The above two divergence values allow us to understand information disclosure at two different levels. With $D_{individual}$, we can conduct privacy studies for the worst-case scenario, because it allows us to see the result at the individual level; with $D_{overall}$, we can conduct privacy studies for the average-case scenario. As we will show in our experiments, they can tell different things.

The Effect of the Error Rate. Some decision tree mining tools provide accurate error rates and some do not. From the privacy perspective, decision trees with error rates definitely reveal more private information. The overall divergences with and without error rates are plotted in Figure 3. The overall divergence without error rate is much larger than that with error rate; this is true for both Gini Index and Information Gain. However, the impact on information-gain-based decision trees is much more severe than that on gini-index-based decision trees. Generally speaking, with error rate, more private information is disclosed. The reason is that the solution space with error rates is the subset of that of without error rate. More specific information can help the NLP solver to

[2] http://archive.ics.uci.edu/ml/

find solutions that are closer to the original distribution. Therefore, the overall divergence is smaller.

The Effect of Attribute Selection Measure. In Section 4.2, we learn that constraints derived from different attribute selection measures are different according to the attribute selection measures. We would like to see whether there is any difference on privacy disclosure between these attribute selection measures. In particular, we would like to study the difference between the Gini Index measure and the Information Gain measure. We assume that error rates are provided. The results are plotted in Figure 4.

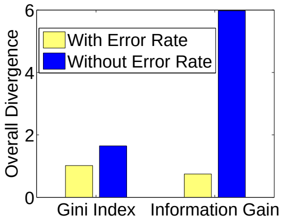


Fig. 3. The effect of error rate

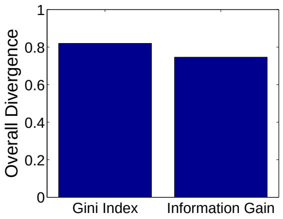


Fig. 4. The effect of selection measure

Case	x_{10}	x_{11}	x_{20}	x_{21}	x_{30}	x_{31}	x_{40}	x_{41}	D_o	D_i^{q2}	D_i^{q3}
S_O	1	0	1	0	0	1	0	1	N/A	N/A	N/A
S_A	1	0	0.5	0.5	0.5	0.5	0	1	0.231	0.693	0.693
S_B	1	0	0.815	0.185	0.815	0.185	0	1	0.068	0.205	0.205

Fig. 5. Impact of the internal nodes for Example 1

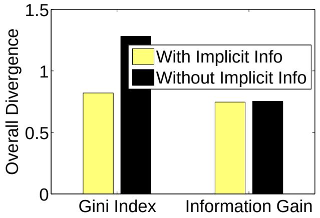


Fig. 6. Effect of implicit information

Top-K	**Difference of** KL_1 **and** KL_2	KL_1	KL_2
1	1.492	0.971	2.463
2	1.371	1.156	2.527
3	1.305	0.942	2.247
4	1.304	0.399	1.703
5	1.262	1.170	2.432
6	1.247	1.187	2.435
7	1.197	1.419	2.616
8	1.195	1.253	2.448
9	1.191	1.441	2.632
10	1.144	1.744	2.888

Fig. 7. Top-10 difference of KL-divergence(KL_1:with Implicit)

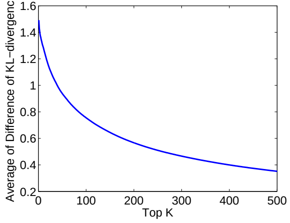


Fig. 8. impact of individual divergence

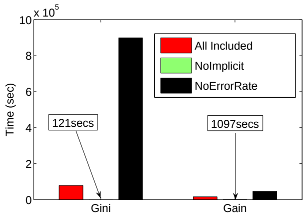


Fig. 9. Running time

From the results, we do see that the overall divergence using gini index is larger from that using information gain. However, there are many factors that cause such a difference, including height of the decision trees, utility of the trees, etc. A comprehensive comparisons of the privacy disclosure between these two measures is beyond the scope of this paper. The goal of this experiment is to show that using our method, data publishers can measure the privacy consequence of their to-be-published decision trees, regardless of what selection method is used.

The Effect of Implicit Information. We have conducted experiments to compare the difference on privacy between with and without implicit information. In the "with" case, we include the internal-constraints while in the "without" case, we exclude the internal-constraints.

First of all, we apply our ME method on Example 1 to illustrate the importance of the implicit information in the decision tree classifier. We have 8 combinations of $P(SA \mid QI)$ since we have 4 distinct QIs and 2 distinct SAs. S_A, S_B are the solutions without and with implicit information, respectively. The original distribution is denoted as S_O. S_A, S_B, and S_O are all listed in Figure 5, where x_{i0} and x_{i1} are the conditional probabilities for q_i whose SA is "$\leq 50K$" and "$> 50K$", respectively. We also list the overall divergence in the D_o column. From the results, we can see that S_B–the results using implicit information–has a smaller overall divergence, and is thus a more accurate estimate.

We also conduct our experiments using the Adult dataset D; we use both the Gini Index measure and the Information Gain measure. In each experiment, we get two estimates, one of which is with the implicit information, the other of which is without the implicit information. In Figure 6, we draw the overall divergences for the two estimations with respect to the real distribution. There is obvious difference for the Gini Index measure. Surprisingly, it shows that there is no major difference for the overall divergences for the Information Gain measure.

To gain a better understanding, we proceed to analyze the *individual divergence* of the result for Information Gain measure. We measure the *individual divergence* between the real distribution and the estimated distribution for each individual QI value. We list the 10 most significant individual divergences in Figure 7, where KL_1 is the individual divergence between the original probabilities and the estimated probabilities when implicit information at internal nodes is used; KL_2 is the corresponding individual divergence when implicit information is not used. From Figure 7, we can clearly tell that KL_1 is significantly smaller than KL_2. For example, in the fourth row in Figure 7, the individual divergence for this QI with the implicit information is 0.399 while that for without the implicit information is 1.703, about 77 percent lower. To fully understand how the implicit information affects the privacy at individual level, we average the top K largest difference between the individual divergences obtained with and without the internal-constraints. The results are plotted in Figure 8; they show that the average impact of the internal-constraint decreases. That is why we do not see much difference if we only measure overall divergence.

Performance. To understand the performance of our proposed method, we conduct two sets of experiments to learn the running time and the memory usage of our ME method. One is for the Gini Index measure while the other is for the

Information Gain measure. In each set, “All Included” means all the constraints are included, with the error rate and the implicit information; “NoImplicit” means no implicit information is included; “NoErrorRate” means no error rate is published. The running time is shown in Figure 9. We find that it is more time-consuming for the Information Gain measure to get the solution than for the Gini Index measure. Intuitively, in Information Gain measure, we need to perform the logarithm computation while in Gini Index, multiplication is performed. Logarithm computation is much more costly. Moreover, we also find that the memory usage of the Information Gain measure (2.5G) is much larger than that of the Gini Index measure (1.2G). This difference is caused by logarithm computation and the different number of constraints. We have observed that the total running time for “All Included” is far less than that of “NoErrorRate”. This is because the search space for “All Included” is much smaller than that of “NoErrorRate” due to the fact that the former search space is a subset of the latter. On the other hand, we find out that the total running time for “All Included” is more than that of “NoImplicit”. Without the internal-constraints, our solver only has linear constraints to evaluate. Solvers usually run much slower if there are non-linear constraints, such as those derived from implicit information.

6 Conclusion and Future Work

We propose a systematic method to quantitatively measure the private information disclosed by decision tree classifiers. Our method is based on a well-established principle, the Maximum Entropy Principle. We model both leaf nodes and internal nodes as constraints. We then feed these constraints to a Non-Linear Programming software to find the maximum entropy estimate. Our experiments have shown that the proposed method is quite effective.

We also realize that in building decision trees, the training dataset is only a subset (e.g. two third) of the original dataset; as long as we do not publish the information about this subset, adversaries do not know which tuples from the dataset are selected as training data. Although adversaries can still use Maximum Entropy to conduct estimate, the accuracy of the estimate will be affected. We plan to study how the training data selection process affect the privacy of decision trees.

Several other directions can also be followed in our future work. One direction is to extend this method to deal with other data mining results. Another interesting direction is to develop methods to disguise the decision tree mining results, such that the privacy requirements are satisfied, while at the same time, the utility of the published results is not sacrificed too much.

References

1. Samarati, P.: Protecting respondents’ identities in microdata release. IEEE transactions on Knowledge and Data Engineering **13**(6) (2001)

2. Sweeney, L.: k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness, and Knowlege-Based Systems **10**(5) (2002)
3. Quinlan, J.R.: Induction of decision trees. Machine Learning 1 (1986)
4. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report (1998)
5. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito:efficient full-domain k-anonymity. In: Proceedings of the 2005 ACM SIGMOD. (June 12 - June 16 2005)
6. Bayardo, R.J., Agrawal, R.: Data privacy through optimal k-anonymization. In: ICDE'05
7. Xiao, X., Tao, Y.: Anatomy: Simple and effective privacy preservation. In: VLDB'06
8. Martin, D.J., Kifer, D., Machanavajjhala, A., Gehrke, J.E., Halpern, J.: Worst case background knowledge. In: ICDE'07
9. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: SIGMOD'00, Dallas, TX USA (May 15 - 18 2000) 439–450
10. Rizvi, S., Haritsa, J.R.: Maintaining data privacy in association rule mining. In: Proceedings of the 28th VLDB Conference, Hong Kong, China (2002)
11. Evfimievski, A., Srikant, R., Agrawal, R., Gehrke, J.: Privacy preserving mining of association rules. In: KDD'02
12. Machanavajjhala, A., Gehrke, J.E., Kifer, D., Venkitasubramaniam, M.: L-diversity: Privacy beyond k-anonymity. In: ICDE'06
13. Wong, R.C.W., Li, J., Fu, A.W.C., Wang, K.: (α, k)-anonymity: An enhanced k-anonymity model for privacy-preserving data publishing. In: KDD'06
14. Li, N., Li, T.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: ICDE'07
15. Xiao, X., Tao, Y.: m-invariance: Towards privacy preserving re-publication of dynamic datasets. In: SIGMOD'07
16. Kantarcioglu, M., Jin, J., Clifton, C.: When do data mining results violate privacy? In: KDD'04
17. Vaidya, J., Clifton, C.: Privacy-preserving decision trees over vertically partitioned data
18. Wright, R., Yang, Z.: Privacy-preserving bayesian network structure computation on distributed heterogeneous data. In: KDD'04
19. Du, W., Teng, Z., Zhu, Z.: Privacy-MaxEnt: Integrating background knowledge in privacy quantification. In: SIGMOD'08
20. Wong, R., Fu, A., Wang, K., Pei, J.: Minimality attack in privacy preserving data publishing. In: VLDB'07
21. Zhang, L., Jajodia, S., Brodsky, A.: Information disclosure under realistic assumptions: Privacy versus optimality. In: CCS'07
22. Zhu, Z., Wang, G., Du, W.: Deriving private information from association rule mining results. In: ICDE '09
23. Jaynes, E.T.: Information theory and statistical mechanics. Physical Review **106**(4) (1957) 620–630
24. Byrd, R., Nocedal, J., Waltz, R.: Knitro: An integrated package for nonlinear optimization. In di Pillo, G., Roma, M., eds.: Large-Scale Nonlinear Optimization. Springer-Verlag (2006) 35–59
25. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers (1993)
26. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Chapman Hall/CRC (1984)