

Utility-Based Anonymization for Privacy Preservation with Less Information Loss

Jian Xu¹ Wei Wang¹ Jian Pei² Xiaoyuan Wang¹ Baile Shi¹ Ada Wai-Chee Fu³
¹Fudan University, China ²Simon Fraser University, Canada ³The Chinese University of Hong Kong

¹xujian, weiwang1, xy_wang, bshig@fudan.edu.cn

²jpei@cs.sfu.ca ³adafu@cse.cuhk.edu.hk

ABSTRACT

Privacy becomes a more and more serious concern in applications involving microdata. Recently, efficient anonymizations have attracted much research work. Most of the previous methods use global recoding, which maps the domains of the quasi-identifier attributes to generalized or changed values. However, global recoding does not always achieve effective anonymization in terms of discernability and query answering accuracy using the anonymized data. Moreover, anonymized data is often used for analysis. As we pointed out in many analytical applications, different attributes in a data set may have different utility in the analysis. The utility of attributes has not been considered in the previous methods.

In this paper, we study the problem of utility-based anonymization. First, we propose a simple framework to specify utility of attributes. The framework covers both numeric and categorical data. Second, we develop two simple yet efficient heuristic local recoding methods for utility-based anonymization. Our extensive performance study using both real data sets and synthetic data sets shows that our methods outperform the state-of-the-art multivariate global recoding methods in both discernability and query answering accuracy. Furthermore, our utility-based method can boost the quality of analysis using the anonymized data.

Categories and Subject Descriptors

H.2.8 [Database Applications]: Data Mining

General Terms

Security, Algorithms, Performance

Keywords

Privacy preservation, data mining, k-anonymity, utility-based recoding

1. INTRODUCTION

Recently, privacy becomes a more and more serious concern in applications involving microdata which refers to data published in its raw, non-aggregated form [17]. One important type of privacy attack is re-identifying individuals by joining multiple public data sources. For example, according to [15], more than 85% of the population of the United States can be uniquely identified through their zipcode, gender, and date of birth.

To protect privacy against this type of attacks, k-anonymity is proposed [12; 15]. A data set is k-anonymous (k ≥ 1) if each record in the data set is indistinguishable from at least k other records within the same data set. The larger the value of k, the better the privacy is protected.

Since the concept of k-anonymity has been proposed, efficient methods for anonymization have attracted much research work. A few k-anonymization algorithms have been developed. We summarize the related work briefly in Section 2.2. Generally, to achieve k-anonymity, those methods generalize or suppress the identifier attributes which are the minimal set of attributes in the table that can be joined with external information to re-identify individual records.

Information loss is an unfortunate consequence of anonymization. In order to make the anonymized data as useful as possible, it is required to reduce the information loss as much as possible. A few models have been proposed to measure the usefulness of anonymized data. For example, the discernability model tries to minimize the number of tuples that are indistinguishable as long as they satisfy the k-anonymity requirement.

In this paper, we study the problem of k-anonymization and focus on two interesting issues: anonymization using heuristic local recoding and utility-based anonymization.

1.1 Global and Local Anonymization

Many recent methods (e.g., [4; 8; 9]) use global recoding which maps the domains of the quasi-identifier attributes to generalized or changed values. In other words, the data space is partitioned into a set of (non-overlapping) regions. The anonymization maps tuples in a region to the same generalized or changed tuple. For example, Figures 1(b) demonstrates k-anonymization using global recoding for the table in Figures 1(a), where (age, zipcode) is the quasi-identifier. Tuples R3 and R4 in Figures 1(a) are identical. They are mapped to the same generalized tuple in global recoding. In contrast, local recoding maps (non-distinct) individual tuple to generalized tuples. For example, Figure 1(c) shows k-anonymization using local recoding of the same table in Figures 1(a). The two identical tuples R3 and R4, are mapped to different generalized tuples in local recoding. Clearly, global recoding can be regarded as a specific type of local recoding.

Interestingly, from Figure 1, we can observe that local recoding may achieve a less information loss than global recoding. For example, the two generalized tuples in global recoding have sizes of intervals 8 and 5 in age, and 1 and 0 in zipcode, respectively. In local recoding, the sizes of intervals are 2 and 1 in age, and 1 and 2 in zipcode, respectively. By intuition, smaller the sizes

Row-id	Age	Zipcode
R1	24	53712
R2	25	53711
R3	30	53711
R4	30	53711
R5	32	53712
R6	32	53713

(a) The original table.

Row-id	Age	Zipcode
R1	[24-32]	[53712-53713]
R2	[25-30]	53711
R3	[25-30]	53711
R4	[25-30]	53711
R5	[24-32]	[53712-53713]
R6	[24-32]	[53712-53713]

(b) 3-anonymization by global recoding.

Row-id	Age	Zipcode
R1	[24-30]	[53711-53712]
R2	[24-30]	[53711-53712]
R3	[24-30]	[53711-53712]
R4	[30-32]	[53711-53713]
R5	[30-32]	[53711-53713]
R6	[30-32]	[53711-53713]

(c) 3-anonymization by local recoding.

Figure 1: Global recoding and local recoding. The row-ids are for reference only and are not released with the data. The row-ids are not part of the quasi-identifier.

of intervals in the generalized tuples, less information loss in the anonymization.

Can we use local recoding to achieve less information loss in anonymization effectively? Generally, optimal k -anonymity is NP-hard [10; 2]. In this paper, we propose two simple yet efficient heuristic algorithms using local recoding for k -anonymization. Our extensive empirical study on both real data sets and synthetic data sets show that our method outperforms the state-of-the-art global recoding method in both the discernability and the accuracy of query answering.

1.2 Utility-Based Anonymization

Anonymized data is often for analysis and data mining. As well recognized in many data analysis applications, different attributes may have different utility. For example, consider anonymizing a data set about patients for disease analysis. Suppose we want to achieve k -anonymity, we can generalize from a 4-digit zipcode to a 4-digit prefix (e.g., from 53712 to 5371). Alternatively, we can also generalize attribute age to age groups (from 23 to [20, 30]). In many cases, the age information is critical to disease analysis, while the information loss on the accuracy is often acceptable (a 4-digit prefix in fact still identifies a relatively local region). Thus, the age attribute has more utility than the zipcode attribute, and should be retained as accurately as possible in anonymization.

Can we make the anonymization utility aware? The utility of attributes has not been considered by previous anonymization methods. In this paper, we propose a model for utility-based anonymization. We consider both numeric data and categorical data with an ordered hierarchy. We present a simple method to specify utility for attributes and push them into the heuristic local recoding anonymization methods. Our experimental results show that the utility-based anonymization improves the accuracy in answering targeted queries substantially.

Paper Organization

The rest of the paper is organized as follows. In Section 2, we recall the notions related to anonymization, and review the related work. We present our utility specification framework in Section 3. Our heuristic local recoding methods are developed in Section 4. An extensive performance study on both real data sets and synthetic data sets is reported in Section 5. The paper is concluded in Section 6.

2. K-ANONYMITY AND RELATED WORK

2.1 K-Anonymity

Consider a table $T = (A_1; \dots; A_n)$. A quasi-identifier is a minimal set of attributes $(A_{i_1}; \dots; A_{i_l})$ ($1 \leq i_1 < \dots < i_l \leq n$) in T that can be joined with external information to re-identify individual records. In this paper, we assume that the quasi-identifier is specified by the administrator based on the background knowledge. Thus, we focus on how to anonymize to satisfy the k -anonymity requirement.

Formally, given a parameter k and the quasi-identifier $(A_{i_1}; \dots; A_{i_l})$, a table T is said k -anonymous if for each tuple $t \in T$, there exist at least another $(k - 1)$ tuples $t_1; \dots; t_{k-1}$ such that those k tuples have the same projection on the quasi-identifier, i.e. $t_{(A_{i_1}; \dots; A_{i_l})} = t_{1(A_{i_1}; \dots; A_{i_l})} = \dots = t_{(k-1)(A_{i_1}; \dots; A_{i_l})}$. Tuple t and all other tuples indistinguishable from the quasi-identifier form an equivalence class. We call the class the group that t is generalized.

Given a table T with the quasi-identifier and a parameter k , the problem of k -anonymization is to compute a view T^0 that has the same attributes as T such that T^0 is k -anonymous and T^0 is as close to T as possible according to some quality metric. We shall discuss the quality metrics soon.

Since the attributes not in the quasi-identifier do not need to be changed, to keep our discussion simple but without loss of generality, hereafter we consider only the attributes in the quasi-identifier. That is, for table $T(A_1; \dots; A_n)$ in question, we assume $(A_{i_1}, \dots, A_{i_l})$ is the quasi-identifier.

2.2 Related Work

k -anonymization was proposed by Samarati and Sweeney [11; 15; 14]. Generally, data items are recoded in anonymization, we regard suppression as a specific form of recoding that recodes a data item to null value (i.e., unknown).

Two types of recoding can be used [17]: global recoding and local recoding, as described and demonstrated in Section 1.1. Most previous methods use global recoding. In [11; 13]-domain generalization, a specific type of global recoding, was developed, which maps the whole domain of each quasi-identifier attribute to a general domain in the domain generalization hierarchy. Full-domain generalization guarantees that all values of a particular attribute still belong to the same domain after generalization.

To achieve full-domain generalization, two types of partitioning can be applied. First, single-dimensional partitioning divides an attribute into a set of non-overlapping intervals, and each interval will be replaced by a summary value (e.g., the mean, the median, or the range). On the other hand, (strict) multidimensional partitioning [9] divides the domain into a set of non-overlapping

multidimensional regions, and each region will be generalized into a summary tuple.

Generally, anonymization is accompanied by information loss. Various models have been proposed to measure the information loss. For example, the discernability model [4] assigns to each tuple a penalty based on the size of the group that is generalized, i.e., the number of tuples equivalent to the quasi-identifier. That is,

$$C_{DM} = \sum_{E \in \mathcal{E}} |E|^2$$

Alternatively, the normalized average equivalence class size metric was given in [9]. The intuition of the metric is to measure how well the partitioning approaches the best case where each tuple is generalized in a group of indistinguishable tuples. That is,

$$C_{AVG} = \frac{\text{number of tuples in the table}}{\text{number of group-bys on quasi-identifier}}$$

The quality of anonymization can also be evaluated based on its usefulness in data analysis applications, such as classification [6; 16].

The ideal anonymization should minimize the penalty. However, theoretical analysis [2; 10; 9; 3; 1] indicates that the problem of optimal anonymization under many non-trivial quality metrics is NP-hard. A few approximation methods were developed [3], such as data y [14], annealing [18], and Mondrian multidimensional k-anonymity [9]. Interestingly, some optimal methods [4; 8] have exponential cost in the worst case were proposed. The experimental results in those studies show that they are feasible and achieve good performance in practice.

3. UTILITY-BASED ANONYMIZATION

Without loss of generality, in this paper we assume that generalization is used in anonymization. That is, when a tuple is generalized, the ranges of the group of tuples that are generalized are used to represent the generalization, as illustrated in Figure 2. Other representations such as mean or median are used, the definitions can be revised straightforwardly and our methods still work.

3.1 Utility-Based Anonymization: Motivation

In previous methods, the quality metrics, such as the discernability metric and the normalized average equivalence class size metric discussed in Section 2.2, mainly focus on the size of groups in anonymization. In an anonymized table, when each group of tuples sharing the same projection on the quasi-identifier has the same penalty, the penalty metrics are minimized. However, such metrics may not lead to high quality anonymization.

EXAMPLE 1 (QUALITY METRICS). Suppose we want to achieve 2-anonymity for the six tuples shown in Figure 2. $(X; Y)$ is the quasi-identifier. The six tuples can be anonymized in three groups: $f; a; b; g$; $f; c; d; g$; and $f; e; g$. In this anonymization scheme, both the discernability metric C_{DM} and the normalized average equivalence class size metric C_{AVG} are minimized.

Let us consider the utility of the anonymized data. Suppose each group is generalized using the range of the tuples in the group. That is, a and b are generalized to $(10; 20]$; c and d are generalized to $(50; 60]$; e and f are generalized to $(15; 20]$.

In order to measure how well the generalized tuples approximate the original ones, for each tuple we can use the sum of the interval sizes on all attributes of the generalized tuple to measure

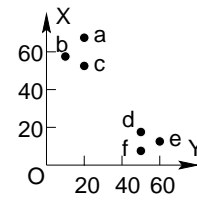


Figure 2: The six tuples in Example 1.

the uncertainty of the generalized tuples. That is, $U(a) = U(b) = 10 + 10 = 20$. Similarly, we get $U(c) = U(d) = 60$ and $U(e) = U(f) = 15$. The total uncertainty of the anonymized table is the sum of the uncertainty of all tuples, i.e., $U(T) = \sum_{t \in T} U(t) = 20 + 20 + 60 + 60 + 15 + 15 = 190$. By intuition, the uncertainty reflects the information loss. The less the uncertainty, the less information is lost.

On the other hand, we may anonymize the tuples in two groups: $f; a; b; g$ are generalized to $(10; 20]$; $[50; 70]$, and $f; c; d; e; f; g$ are generalized to $(50; 60]$; $[10; 20]$. In fact, the data set is 2-anonymous, which is better than 2-anonymous in terms of privacy preservation. Moreover, the total uncertainty in this anonymization is 150, lower than the 2-anonymity scheme.

However, this anonymization scheme has a higher penalty than the 2-anonymity scheme in both the discernability metric C_{DM} and the normalized average equivalence class size metric C_{AVG} . In other words, optimizing the quality metrics on group size metrics always lead to anonymization that minimizes the information loss. ■

Can we have a quality metric that can measure the utility of the anonymized data? Such a utility-based metric should capture the following two aspects.

The information loss caused by the anonymization. When a record is anonymized, it is generalized in its quasi-identifier. The metric should measure the information loss of the generalization with respect to the original data.

The importance of attributes. As well accepted in data analysis such as aggregate queries, different attributes may have different importance in data analysis. In anonymization, we introduce less uncertainty to the important attributes. Such utility-aware anonymization may help to improve the quality of analysis afterwards.

3.2 Weighted Certainty Penalty

We introduce the concept of certainty penalty to capture the uncertainty caused by generalization.

3.2.1 Numeric Attributes

First, let us consider the case of numeric attributes. Let t be a tuple in a table with quasi-identifier $(A_1; \dots; A_n)$, where all attributes are numeric. Suppose a tuple $t = (x_1; \dots; x_n)$ is generalized to tuple $t^0 = ([y_1; z_1]; \dots; [y_n; z_n])$ such that $y_i \leq x_i \leq z_i$ ($1 \leq i \leq n$). On attribute A_i , the normalized certainty penalty is defined as

$$NCP_{A_i}(t) = \frac{z_i - y_i}{|A_i|}$$

where $|A_i| = \max_{t \in T} t.A_i - \min_{t \in T} t.A_i$ is the range of all tuples on attribute A_i .

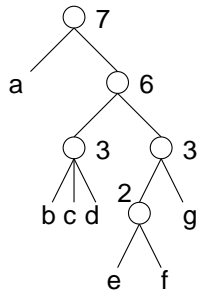


Figure 3: A hierarchy on a categorical attribute.

Let each attribute A_i be associated with a weight w_i to reflect its utility in the analysis on the anonymized data. Then, the weighted certainty penalty of a tuple is given by

$$NCP(t) = \sum_{i=1}^n (w_i \cdot NCP_{A_i}(t)) = \sum_{i=1}^n (w_i \cdot \frac{z_i - y_i}{|A_i|}):$$

Clearly, when all weights are set to 1 and all attributes have ranges $[0; 1]$, the weighted certainty penalty is the norm distance between points $(\max_{t \in G} f_t: A_1 g; \dots; \max_{t \in G} f_t: A_n g)$ and $(\min_{t \in G} f_t: A_1 g; \dots; \min_{t \in G} f_t: A_n g)$, where G is the equivalence group that belongs to.

Our utility-based metric is given by the total weighted certainty penalty on the whole table. That is,

$$NCP(T) = \sum_{t \in T} NCP(t):$$

3.2.2 Categorical Attributes

Distance is often not well defined on categorical attributes which makes measuring utility on categorical attributes difficult. In some previous methods (e.g., [8; 9]), it is assumed that a total order exists on all values in a categorical attribute. In many applications, such an order may not exist. For example, sorting all zipcodes in their numeric order may not reflect the utility properly. Two regions may be adjacent but their zipcodes may not be consecutive.

More often than not, hierarchies exist in categorical attributes. For example, zipcodes can be organized into hierarchy of regions, counties, and states.

Let $v_1; \dots; v_l$ be a set of leaf nodes in a hierarchy tree. Let u be the node in the hierarchy on the attribute such that an ancestor of $v_1; \dots; v_l$, and u does not have any descendant that is still an ancestor of $v_1; \dots; v_l$. u is called the closest common ancestor of $v_1; \dots; v_l$, denoted by $\text{ancestor}(v_1; \dots; v_l)$. The number of leaf nodes that are descendants of u is called the size of u , denoted by $\text{size}(u)$.

Can we use the hierarchy information to measure the utility of categorical attributes?

EXAMPLE 2 (UTILITY ON CATEGORICAL ATTRIBUTES).

Consider a categorical attribute of domain $\{a; b; c; d; e; f; g\}$. Suppose a hierarchy exists on the attribute as shown in Figure 3. The values appear in the leaf nodes in the hierarchy tree.

Intuitively, if we generalize tuples having values b and c , the anonymized tuples have good utility on this categorical attribute, since b and c share the same parent in the hierarchy. On the other hand, putting a and f into the same generalized group may have poor

utility on the attribute since the common ancestor of a and f is far away from f .

One may wonder whether the shortest distance between a and v in the hierarchy tree can be used as the certainty penalty. Unfortunately, it does not work well. Consider Figure 3 again. Intuitively, generalizing d and e together is better than generalizing a and d together, since the closest common ancestor of d and e is in a hierarchical level lower than the closest common ancestor of a and d . However, the shortest distance between d and e is 5, while the shortest distance between a and d is only 4. If we use the shortest distance as the guide, then merging a and d is better than merging d and e . In other words, the shortest distance may be misleading. To measure the utility of merging two values a and y into the same generalized group, we can observe that the critical factor is the closest common ancestor of x and y , how many other values are also the descendants of u . The smaller the number, the smaller the uncertainty introduced by the generalization. ■

Based on the observation in Example 2, we define the certainty penalty on categorical attributes as follows.

Suppose a tuple t has value v on a categorical attribute A . When it is generalized in anonymization, the value will be replaced by a set of values $v_1; \dots; v_l g$, where $v_1; \dots; v_l$ are the values of tuples on the attribute in the same generalized group. We define the normalized certainty penalty of t as follows.

$$NCP_A(t) = \frac{\text{size}(u)}{|A|};$$

where $|A|$ is the number of distinct values on attribute A . Here, we assume that each leaf node is of the same importance. The definition can be straightforwardly extended by assigning weights to internal nodes to capture the more important leaf nodes in each hierarchical structures. Limited by space, we omit the details here.

EXAMPLE 3. Let us consider the cases discussed in Example 2 again. Putting a and d together in a group has penalty 4, and putting d and e together in a group has penalty 5, which is smaller than the case of a and d . ■

Putting things together, for a table consisting of both numeric and categorical attributes, the total weighted normalized certainty penalty is the sum of the weighted normalized certainty penalty for all tuples. That is,

$$NCP(T) = \sum_{t \in T} \sum_{i=1}^n (w_i \cdot NCP_{A_i}(t));$$

where $NCP_{A_i}(t)$ should be computed according to whether A_i is a numeric or categorical attribute.

Given a table T , a parameter k , the weights of attributes and the hierarchies on categorical attributes, the problem of optimal utility-based anonymization is to compute a k -anonymous table such that the weighted normalized certainty penalty is minimized.

3.3 Complexity

The previous studies show that the problem of optimal k -anonymity is NP-hard under various quality models. The utility-based model we propose here is a generalization of the suppression model. We have the following results on the complexity.

LEMMA 1 (CATEGORICAL ATTRIBUTES). Suppose the problem of optimal utility-based k -anonymization is NP-hard for $k=2$.

Input: a table T , parameters k , weights of attributes, and hierarchies on categorical attributes;
 Output: a k -anonymous table T^0 ;
 Method:
 1: Initialization: create a group for each tuple;
 2: WHILE there exists some group G such that $|G| < k$ DO
 3: FOR each group G such that $|G| < k$ DO
 4: scan all other groups once to find group G^0 such that $NCP(G \cup G^0)$ is minimized;
 5: merge group G and G^0 ;
 6: FOR each group G such that $|G| \geq 2k$ DO
 7: split the group into $\lfloor \frac{|G|}{k} \rfloor$ groups such that each group has at least k tuples;
 8: generalize and output the surviving groups;

Figure 4: The bottom-up algorithm.

Proof sketch. We can show that the suppression model used in [2] is a special case of the weighted normalized certainty penalty defined here, where all weights are set to 1 and all hierarchies have only two levels: the detailed values and suppression. This follows from the result in [2].

Following from the lemma, we have the following result.

THEOREM 1 (COMPLEXITY). The problem of optimal utility-based anonymization is NP-hard.

In fact, for a table consisting of only numeric attributes, the problem is still NP-hard. Limited by space, we omit the details.

4. GREEDY METHODS

In this section, we develop heuristic methods for utility-based anonymization. We propose two greedy algorithms. The first method conducts a bottom-up search, while the second one works top-down.

4.1 The Bottom-Up Method

To maximize the utility of the anonymization of a tuple, we may “cluster” the tuples locally according to the weighted certainty penalty. Those compact clusters having at least k tuples can be generalized. This idea leads to our bottom-up method. At the beginning, we treat each tuple as an individual group. In each iteration, for each group whose population is less than k , we merge the group with the other group such that the combined group has the smallest weighted certainty penalty. The iterations continue until every group has at least k tuples. The algorithm is shown in Figure 4.

The bottom-up algorithm is a greedy method. In each round, it merges groups such that the resulted weighted certainty penalty is locally minimized. In one iteration, if one group is merged with multiple groups, it is possible that the group becomes larger than k . In order to avoid over-generalization, if a group has more than $2k$ tuples, then the group should be split. It is guaranteed that the resulted table, each group has up to $(2k - 1)$ tuples. Please note that, unlike many previous methods that try to minimize the average number of tuples per group, our algorithm tries to

Input: a table T , parameters k , weights of attributes, and hierarchies on categorical attributes;
 Output: a k -anonymous table T^0 ;
 Method:
 1: IF $|T| \leq k$ THEN RETURN T
 2: ELSE
 3: partition T into two exclusive subsets T_1 and T_2 such that T_1 and T_2 are more local than T , and either T_1 or T_2 have at least k tuples;
 4: IF $|T_1| > k$ THEN recursively partition T_1 ;
 5: IF $|T_2| > k$ THEN recursively partition T_2 ;
 6: adjust the groups so that each group has at least k tuples;

Figure 5: The framework of the top-down greedy search method

reduce the weighted certainty penalty, which reflects the utility of the anonymized data. At the same time, they also keep the number of tuples per group small.

EXAMPLE 4 (ADVANTAGES OF THE BOTTOM-UP METHOD). To understand the difference between our method and the previous methods, let us check the case in Figure 2. The bottom-up method generates two groups: $\{a, b, c\}$ and $\{d, e, f, g\}$, as expected in Example 1. Although it does not minimize the average group size, it optimizes the utility of the anonymized data – the information loss is better than an l_2 -anonymous scheme in this example. Moreover, as a byproduct, the result is anonymous, which means a stronger protection of privacy.

After the k -th round, the number of tuples in a group is at least k . Therefore, by at most $\log_2 \frac{n}{k}$ iterations, each group has at least k tuples, and thus the generalized groups satisfy the k -anonymity requirement. The complexity of the algorithm is $O(n \log_2 \frac{n}{k} |T|^2)$ on table T .

The bottom-up method is a local recoding method. It does not split the domain. Instead, it only searches the tuples. Different groups may have overlapping ranges. Moreover, in the step of splitting several tuples with the identical quasi-identifier may belong to different groups.

4.2 A Top-Down Approach

The major cost in the bottom-up method is to search for the best groups (Step 4 in Figure 4). In the bottom-up method, we have to use a two-level loop to conduct the search. We observe that if we can partition the data properly so that the tuples in each partition are local, then the search of the nearest neighbors can be improved. Motivated by this observation, we develop the top-down approach. The general idea is as follows. We partition the table into several groups. A set of tuples is partitioned into subsets if each subset is more local. That is, likely they can be further partitioned into smaller groups that reduce the weighted certainty penalty. After the partitioning, we merge the groups that are smaller than k to honor the k -anonymity requirement.

To keep the algorithm simple, we consider binary partitioning. That is, in each round, we partition a set of tuples into two subsets. The algorithm framework is shown in Figure 5.

Now, the problem becomes how we can partition a set of tuples into two subsets so that they are compact and likely lead to a smaller weighted certainty penalty. We adopt the following heuristic. We

