Avoiding the Destruction of Data-Mining Utility in Privacy Preserving Data Publishing

Fayola Peters

2011

Keywords: Forensic Evaluation, Prototype Learning, K-Nearest Neighbor

Abstract

Avoiding the Destruction of Data-Mining Utility in Privacy Preserving Data Publishing

Fayola Peters

The field of privacy preserving data publishing (PPDP) has become active in recent years in order to first reduce and/or eliminate the possibility of the re-identification of individuals in a sensitive data-set. Second, to reduce the probability that an attacker is able to link an individual to a sensitive attribute in a published data-set. So far there is agreement on one issue in the PPDP field, that is, the proposed privacy preserving methods are destructive to data mining algorithms. Therefore this work presents a novel suppression algorithm for privacy (SAP) designed to not only preserve privacy at an acceptable level, but also produce a sanitized data-set with a high level of utility. Using 4 data-sets from the UCI repository, and the k-nearest neighbor algorithm to test utility levels before and after the application of SAP, the results of this work demonstrate that SAP generates a sanitized data-set yielding low levels of privacy breaches and statistically comparable utility measures.

Contents

List of Figures

Chapter 1

Introduction

The principal goal of privacy preserving data publishing (PPDP) is to ensure that the published data can preserve individual privacy while remaining useful for workloads such as data mining. However, in recent years there has been a growing concern about preserving the privacy of data-sets to be published. In fact some organizations are wary of publishing their data for fear that individuals in the data-set will be re-identified as is the case of former Massachusetts governor William Weld, and [name] from and AOL data-set. This can happen by joining the published data to voters lists of census data. Beyond, re-identification is sensitive attribute disclosure. This is where an individual in a data set can be associated with a sensitive attribute. Consider the example in Figure 1.1. Here the shaded column represent the sensitive attribute values while the other columns represent the quasi identifiers (QID) and/or non sensitive attribute values. (Please note that the name are only included for ease of explanation. Normally these are removed before publication).

Name	Job	Sex	Age	Disease
Dawn	Lawyer	Female	39	Hepatitis
Jenny	Teacher	Female	23	HIV
John	Nurse	Male	41	HIV
Stan	Mayor	Male	56	Hepatitis

Figure 1.1: Patient Table before sanitization

So let us consider Jenny. In order for an attacker (Jenny's boss Bill) to re-identify her in the data-set, through casual investigation Bill can figure out that Jenny had a recent stay at the hospital. Furthermore, once he figures out Jenny's age, he will learn that Jenny is HIV positive. Clearly, privacy breaches of this nature can lead, in Jenny's case to being discriminated against.

Name	Job	Sex	Age	Disease
Dawn	Lawyer	Female	35-40	Hepatitis
Jenny	Teacher	Female	20-25	HIV
Mary	Teacher	Female	20-25	Hepatitis
Joan	Teacher	Female	20-25	None
John	Nurse	Male	41	HIV
Stan	Mayor	Male	56	Hepatitis

Figure 1.2: Patient Table after sanitization

Other scenarios could involve identity theft, loans and health insurance coverage can be denied and an individual can become a victim to financial fraud.

In this regard, the goal of the work is to introduce a SAP and algorithm designed to add uncertainty to the point where if Bill only has access to the table in Figure 1.2 then he can only be 33% sure that Jenny has HIV rather than 100% sure (Figure 1.1). In attempt to achieve this goal, this research is guided by the following research questions:

- Can using SAP reduce the certainty level for re-identification and sensitive attribute disclosure?
- Can SAP produce usable sanitized data-sets?

This paper is organized as follows: The next section briefly describes the state of the art and offer definitions of privacy. In Section 3, the design and operation of SAP is explained. Section 4 reports on experimental procedure and results. Finally, the paper is concluded and future work discussed in Section 5.

Chapter 2

Related Work and Background

2.1 What is PPDP?

Similarly, researchers agree that PPDP involves the use of techniques to disguise the micro-data records which contain information about specific individuals, while delivering a useful data-set for analysis. This is indicated clearly by Fung [5], who noted that PPDP is a collection of methods and tools for publishing data in a hostile environment so that the published data remains practically useful while individual privacy is preserved. Along the same lines, Domingo et. al. [3] states:

> Statistical disclosure control (also known as privacy-preserving data mining) of microdata is about releasing data-sets containing the answers of individual respondents protected in such a way that: (i) the respondents corresponding to the released records cannot be re-identified; (ii) the released data stay analytically useful.

Also, according to LeFevre [7], protecting individual privacy is an important problem in microdata distribution and publishing. Anonymization algorithms typically aim to satisfy certain privacy definitions with minimal impact on the quality of the resulting data.

2.2 Why is PPDP important?

The importance of PPDP can be answered with a question: *What is data mining without data?*

It is established that data mining is beneficial to many domains such as medical, security, credit fraud and many others. Unfortunately, if the data-sets used in these data mining tasks include a lot of sensitive individual data, the results can lead to privacy breaches. A popular example is the re-identification of William Weld, the former Governor of the state of Massachusetts. Values in quasi identifiers (QIDs) (discussed in Section 5) were linked to his record in a published medical database. More recently, when AOL released the query logs of its users for the purpose of research, Ms. Thelma Arnold was re-identified by the examination of query terms [5].

Such breaches lowers the confidence of data providers in publishing disguised data with the expectation of no breaches. In the previous example AOL had no choice but to quickly remove the data-set. So a drawback is that data holders are less likely to publish data for research and the benefits of data mining are not realized in these domains.

2.3 What is considered a breach of privacy?

In order to realize privacy breaches, one needs to define what constitutes a privacy breach for a particular data-set. There are different levels of privacy. Some levels are determined by individuals in the data-set or by the creators of a privacy policy. An optimal result of a privacy model is defined by Dalenius [2] where he states that access to published data should not enable the attacker to learn anything extra about any target victim, compared to no access to the database, even with the presence of any attacker's background knowledge obtained from other sources [5]. Unfortunately, achieving this ideal is impossible, and so, solutions have focused on trade-offs.

Fung [5], considers two categories of privacy models, 1) considers that a privacy threat occurs when an attacker is able to link a record owner to a record in a published data table to a sensitive attribute in a published data table, to the published data table itself. These are specified as, record linkage, attribute linkage and table linkage respectively. 2) The published data should provide the attacker with little additional information beyond the background knowledge.

Brickell [1], defines a privacy model called sensitive attribute disclosure which occurs when the attacker or adversary learns information about an individual's sensitive attributes. In other words, it captures the gain in the adversaries knowledge due to his observations of the disguised data-set. Also "Microdata privacy can be understood as prevention of membership disclosure" where the attacker should not learn whether a particular individual is included in the database.

In 2007, Wang [13] put forward other definitions of privacy. The article explains:

> There have been two types of privacy concerning data mining. The first type of privacy, called output privacy, is that the data is minimally altered so that the mining result will not disclose certain privacy. The second type of privacy, called input privacy, is that the data is manipulated so that the mining result is not affected or minimally affected.

In this work, the focus is on two prevailing definitions of a privacy breach (these will be refered to as scenario1 (S1) and scenario2 (S2) in the remainder of this paper):

1. S1 - The ability of an attacker to re-identify an individual record in a data-set
2. S2 - The ability of an attacker to associate an individual to a sensitive attribute value.

2.4 What are the methods used to avoid privacy breaches?

The idea of disguising a data-set is known as anonymization. This is performed on the original data-set to "satisfy a specified privacy requirement" [5] resulting in a modified data-set being published. There are five general categories for anonymization, 1) generalization, 2) suppression, 3) anatomization, 4) permutation and 5) perturbation. Most methods and tools created for preserving privacy fall into one or more of these categories.

Before we expand on the above categories, we introduce some information about the structure of the original data-sets:

> *D(Explicit-Identifier, Quasi-Identifier, Sensitive-Attributes, Non-Sensitive-Attributes)*, where Explicit-Identifier is a set of attributes, such as name and social security number, containing information that explicitly identifies record owners; Quasi-Identifier (QID) is a set of attributes that could potentially identify record owners; Sensitive-Attributes consists of sensitive person-specific information such as disease, salary, and disability status; and Non-Sensitive-Attributes contains all attributes that do not fall into the previous three categories [5].

2.4.1 Generalization and suppression

According to [5], generalization or suppression hides some details in QID. If attribute values are numerical, generalization resembles discretization in that exact values are replaced by an interval that covers the exact values. With categorical values, specific values are replaced by general ones, for instance, date of birth, becomes month of birth, or $professional$ replaces $engineer$ and $lawyer$. Suppression replaces some values with a special value. This could be looked upon as *blocking* where special values are replace by a question mark (?) [12].

$k-anonimity$ is one of the methods which makes each record in the table be indistinguishable with k-1 other records by suppression or generalization [9]. There are limitations with k-anonymity including the fact that it does not hide whether a given individual is in the database, or that it does not protect against attacks based on background knowledge [1]. To overcome these problems, researchers have proposed many variations of k-anonymity [8, 9, 15].

2.4.2 Anatomization and permutation

Anatomization and permutation both accomplish a similar task, that is, the de-association of the relationship between the QID and the sensitive attributes. However, anatomization does it by releasing “the data on QID and the data on the sensitive attribute in two separate tables...” with “one common attribute, *GroupID*” [5]. On the other hand, permutation de-associates the relationship between a QID and a numerical sensitive attribute. This is done by partitioning a set of data records into groups and shuffling the sensitive values within each group [14].

2.4.3 Perturbation

A precise definition of perturbation is put forward by [5]:

> The general idea is to replace the original data values with some synthetic data values, so that the statistical information computed from the perturbed data does not differ significantly from the statistical information computed from the original data.

It is important to note that the perturbed data records do not correspond to real world record owners. Also, methods used for perturbation include, additive noise, data swapping and synthetic data generation.

2.5 How are privacy preserving algorithms evaluated?

Three evaluation methods are outlined in [10]. These measures are 1) Information loss measures, 2) Disclosure risk measures and 3) Scores. According to [10], information loss measures are designed to establish in which extent published data is still valid for carrying out the experiments planned on the original data. They take into account the similarity between the original data set and the protected one, as well as the differences between the results that would be obtained with the original data set and the results that would be obtained from the disguised data-set. [10] further explains that disclosure risk measures are used to evaluate the extent in which the protected data ensures privacy and that *scores*, is a summary both information loss and disclosure risk, that is, when these two measures are commensurate, it is possible just to combine them using the average.

Chapter 3

SAP: Suppression Algorithm for Privacy

SAP is based on the concept of rule confusion [cite] where the where instances of a data-set which adheres to a rule are sanitized while the others are kept as is [reasons]. To generate these rules SAP uses ranking algorithm called BORE (best or rest) [6], that determines what values are most critical for classification. Once these are found that values are suppressed (blocked) by replacing them with question marks (?). Intuitively it is expected that by removing these values re-identification can be significantly reduced while producing a useful sanitized data-set.

3.1 The BORE Algorithm

The core of SAP is to find for a particular attribute values more likely to be present for a particular class. To find these values, SAP uses BORE. First we assume that the target class is divided into one class as *best* and the other classes as *rest* [6]. This makes it easy to find the attribute values which have a high probability of belonging to the current *best* class using Bayes theorem. The theorem uses evidence E and a prior probability $P(H)$ for hypothesis $H \in \{best,\ rest\}$, to calculate a posteriori probability $P(H|E) = P(E|H) \times P(H) / P(E)$. When applying the theorem, likelihoods are computed from observed frequencies, then normalized to create probabilities: this normalization cancels out $P(E)$ in Bayes theorem (see Equation 3.1).

Let likelihood = like

$$P(best|E) = \frac{like(best|E)}{like(best|E) + like(rest|E)} \tag{3.1}$$

Unfortunately, one problem was found using the theorem. According to [6], Bayes theorem is a poor ranking heuristic since it is distracted by low frequency evidence. To alleviate this problem the support measure was introduced. Its purpose was to increase as the frequency of a value increases i.e. like(best|E) is a valid support measure hence Equation 3.2.

$$P(best|E) * support(best|E) = \frac{like(best|E)^2}{like(best|E) + like(rest|E)} \tag{3.2}$$

So in this work, each attribute value is ranked according to Equation 3.2. Once ranked, the critical value for each attribute are extracted (those with the highest ranks) and used as a rule for selecting the values of instances to be replaced with a question mark. The whole process is repeated using a different class as *best* and all the others (including the previous *best* class) as *rest*. When a rule for each class has been found the process ends. Figure 3.1 shows the pseudocode for ranking each value of an attribute for an entire training data-set. The parameters include: *a*, the list of attributes from the data-set; *best*, instances with the current best class; and *rest*, instances with classes not including the best class.

```
freq = frequency
like = likelihood
$p_{best}$ = Probability of value in best
$p_{rest}$ = Probability of value in rest
begin initialize a, best, rest, $list = \emptyset$
for all attributes in a do
   for all values in attribute do
      $p_{best}$ = $|best|$ / $|data|$
      $p_{rest}$ = $|rest|$ / $|data|$
      $freq(value|best)$ = $|valueinbest|$ / $|best|$
      $freq(value|rest)$ = $|valueinrest|$ / $|rest|$
      $like(best|value)$ = $freq(value|best)$ x $p_{best}$
      $like(rest|value)$ = $freq(value|rest)$ x $p_{rest}$
      $like_{br}$ = $like(best|value)$ + $like(rest|value)$
      $rank$ = $like(best|value)^2$ / $like_{br}$
      add [value, rank] to list
   end for
end for
return list
```

Figure 3.1: Pseudo code for BORE

3.2 A simple example for SAP

For the data-set in Figure **??**, we want to sanitize it by keeping the SAV and suppressing the QID. By applying the BORE algorithm we generate the following rules which best represent the class.

[rules]

We then replace these values with question marks. For the experiments in Section 4, we vary the number of these that are replaced by 10, 20, and 30%.

The purpose of removing confusing prototypes is to reduce the misclassification rate by an NNC. Figure **??** clearly features the basic algorithm used to accomplish this. Figure **??** shows the general prototype reduction at each level of a data set. A few data sets such as Lymph and Audio show a very small reduction from the original number of prototypes while others such as Breast Cancer, Vote and Tic-Tac-Toe show a radical reduction to less than 10% of the original data set. Please note that since MESO is order dependent, the reduction rate differs once the data set is shuffled.

Chapter 4

SAP Assessment

In Chapter 3, we discussed the design and operation of SAP with the help of a simple example. In this chapter, we look at how SAP can be used to offer an impressive level of privacy while maintaining comparable performance against the original data-sets based on the two scenarios mentioned in Section 2.3. The classifier used in this work to test the utility of the sanitized data-set is the k-NN classifier where $k = 1$ and the performance measures include accuracy (acc), precision (prec), probability of detection (pd) and probability of false alarm (pf). The experiments in this work will focus on how much information the atacker knows and how he can use that knowledge to either re-identify an individual in a data-set or associate that individual with a sensitive attribute value. We model the attacker's knowledge as a query and use a random query generator in order to give a general idea of the level of privacy offered by the sanitized data-set verses an unsanitized data-set. The following sections will detail information about the data used in this study as well as the experimental procedures used for S1 and S2.

4.1 Data Sets

Figure 4.1 lists the four(4) data sets used to assess SAP. The number of instances and attributes per instance are shown for each data set, along with the number of distinct classes of instances. All of these data sets were acquired from the UCI repository [4]. In terms of privacy, we will look at the instances as individual records, the attributes as QIDs and the classes as sensitive attribute values.

Data Set	Instances	Attributes	Class
Breast Cancer	286	9	2
Heart (Cleveland)	303	13	5
Mamography	150	4	3
Vote	435	16	2

Figure 4.1: Data Set Characteristics

4.2 Experimental Method

Before moving forward it is important to note that SAP is used differently for each scenario. For S1, to avoid the re-identification of an individual record, SAP is applied to the QIDs only, while for S2, SAP is only applied on the SAV.

Now, for the experiments, first, we will evaluate SAP as a privacy-preserving algorithm and measure S1 and S2 in a data-set before and after appling SAP. To accomplish this a random query generator is implemented in order to model the knowledge of an attacker. The generator decides what and how many QID values the attacker knows. Then produces a list of these queries making sure that there are no duplicates. The number of queries produced is determined by a stopping criteria which states that if the list of queries does not increase after *n* successive queries, then output the current query list. The value *n* differs for each data-set.

Recall, that each query represents what an attacker knows about an individual, so once the query list is generated, each query is used to determine the probabilty of finding that individual in the data-set. This is done by dividing 1 by the sum of the records that match the query. Additionally, to measure S2 of a data-set, we find the probability of associating an individual to each sensitive attribute value. For example, if *a* and *b* are the SAV of a data-set, then the $P(S2|a) = \frac{\text{number of records matching query with SAV=a}}{\text{sum of the records matching a query}}$.

Next, to measure the utility of SAP the 1-NN classifier is used on the original data-set and the sanitized versions of the data-set. This is done on the UCI data-sets in Figure 4.1 in cross-validation experiments. SAP's performance is measured using probability of detection (pd) and probability of false alarm (pf) completed as follows [11]: By allowing A, B, C and D to represent true negatives, false negatives, false positives and true positives respectfully, it then follows that *pd* also known as recall, is the result of true positives divided by the sum of false negative and true positives $D/(B+D)$. While pf is the result of: $C/(A+C)$. The *pd* and pf values range from 0 to 1. When there are no false alarms $pf = 0$ and at 100% detection, $pd = 1$.

The results were visualized using *quartile charts* as in [11]. To generate these charts the performance measures for the *pd*s and *pf*s are sorted to get the median, lower and upper quartile of numbers. For our quartile charts, the upper and lower quartiles are marked with black lines; the median is marked with a black dot; and the vertical bars are added to mark the 50% percentile value. Figure 4.6 shows an example where the upper and lower quartiles are 39% and 59% respectively, while the median is 49%.

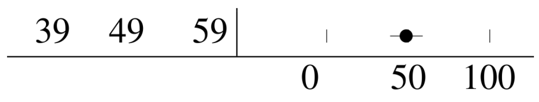


Figure 4.2: Example of quartile chart used in this work.

Finally, the Mann-Whitney U test was used to test for statistical difference before using SAP and after. These results are shown as rank values starting at one(1). The lower the rank value, the better the performance. Please note that the same rank value, indicates no statistical difference.

The following sections describes two(2) experiments, one for each scenario, presents and discusses the results.

4.3 Experiment #1: For S1

The goal for both experiments is to see if data-sets privatized by SAP provide a reasonable level of privacy while maintaining comparable performance measures after classification with the original data-set. In this experiment, the QID's are privatized while the SAV remains the same. This is done in an effort to lower the probability of re-identifying and individual record in a data-set. This part of the experiment follows the pseudo code in Figure 4.3.

Next, to see if SAP maintains utility, 1-NN is used. Here the performance of privatized data-sets in predicting the target class is compare to the performance with the original data-set. To accomplish this, our experiment design follows the pseudo code given in Figure **??** for the standard data sets. From the original data-set, tests were built from 20% of the data, selected at random. The models/prototypes were then learned from the remaining 80% of the data.

This procedure was repeated 5 times, randomizing the order of data in each data-set each time. In the end CLIFF is tested and trained 25 times for each data set.

```
data = {breastcancer, heart-c, mammography, vote}
p = {0.1 0.2 0.3} { each value in p creates a separate private data-set}
queries = Q(QIDs, QID values)
original-data = original data-set
private-data = SAP(original-data, p)
list of P(finding individual record) = list = $\emptyset$
result = list for all private-data and the original-data

for all  data do
    get queries
    for all  p do
        get private-data
    end for
    for all  private-data and original-data do
        for all  queries do
            P(finding individual record) = $\frac{1}{\text{sum of records that match query}}$
            Plist = list + P(finding individual record)
        end for
    end for
    Dlist = Dlist + Plist
end for
All-Dlist = All-Dlist + Dlist
return All-Dlist
```

Figure 4.3: Pseudo code for Experiment #1

4.3.1 Results from Experiment #1

The results for this experiment and Experiment 2 (discussed in the following section) are shown in Figure **??** to Figure **??**. For each data-set, each figure shows the results for a *clean* data-set (without noise) and a *noisy* data-set (with noise). For both the clean and noisy data-sets, the [pd, pf] quartile charts, the percentage of the data-set used for training after using the PLS and the rank values showing the significant difference between the PLS and KNN are presented.

Let us focus on the *clean* results for each data-set from Figure 4.1. First, we compare the CLIFF results with those of the baseline KNN. As shown, despite using only 9% to 15% of the training set, the pds and pfs results of CLIFF compares favorably with those of KNN showing similar or better rank values in most cases. For example, the Mammography(mm) data set (Figure **??**) ranks as number one(1) for both the pd and pf results while KNN ranks at number three(3). In Figure **??**, the Liver(lv) data set exhibits statistically similar results for pd while for pf CLIFF has a much better statistical performance than KNN. Figure **??** to Figure **??** also show encouraging results for CLIFF as compared with KNN, however Figure **??**, the Breast Cancer(bc) data set shows an exception, with the pd and pf statistical results for KNN better than CLIFF.

Next, let us consider the results of the PLS. As compared with the other PLS, CLIFF does as well as or better than the others in all but one(1) case. However, although for the *bc* data set, CNN present statistically better pd and pf results, it does so with 62% of the training data for median pd and pf values of 55% and 39% respectively, while CLIFF only needed 11% of the training data for median pd and pf values of 67% and 20% respectively.

So generally, CLIFF has markedly lower pfs than the other PLS and KNN, this is especially true for the *hh* and *ir* data sets. Also the low pfs does not come at the cost of lower pds, infact, CLIFF's median pd results are competitive (sometimes even the best) of all the other PLS.

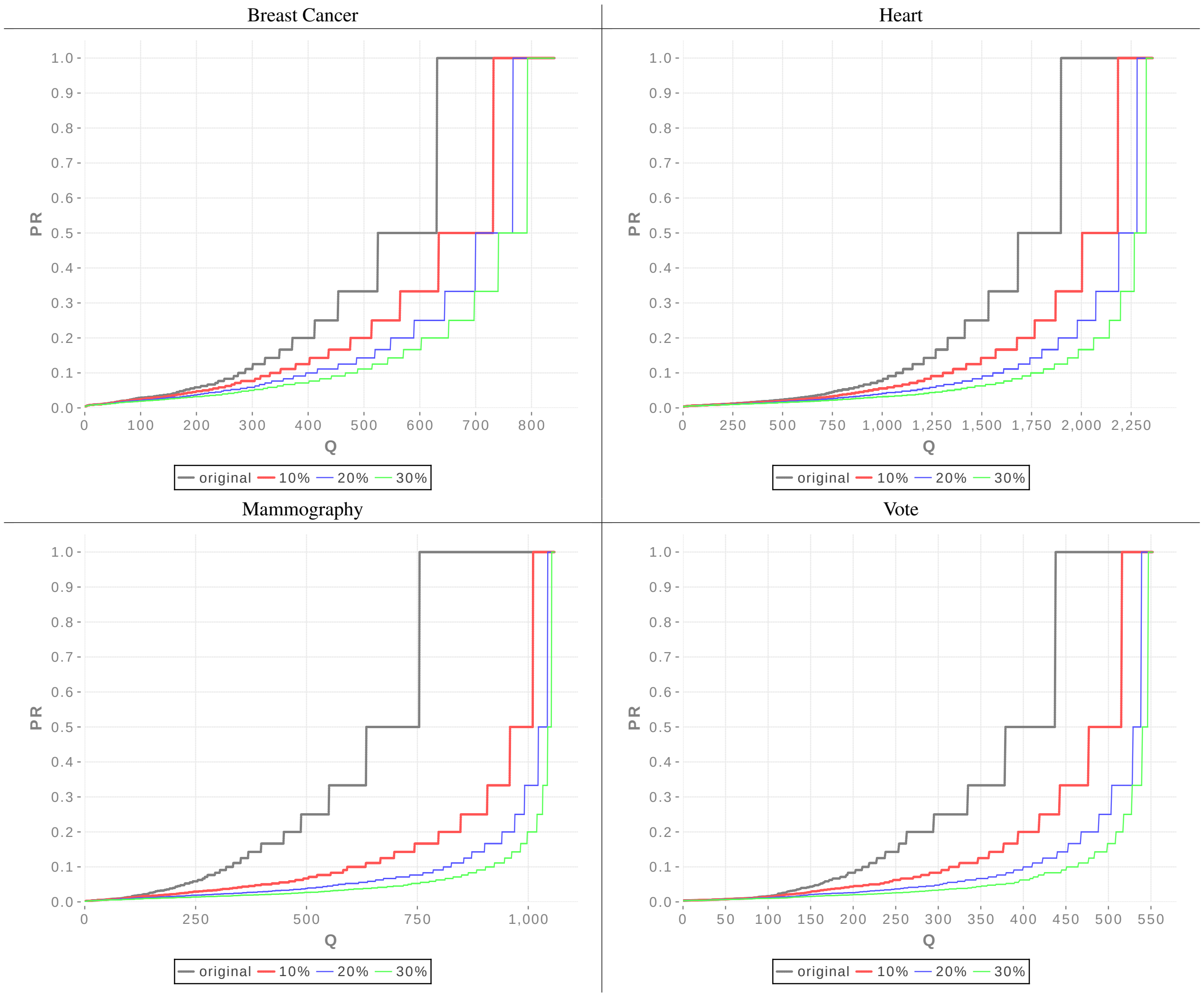
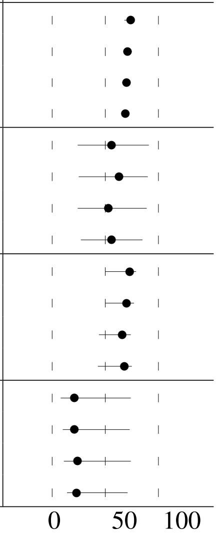


Figure 4.4: Re-identification Levels for sanitized data sets

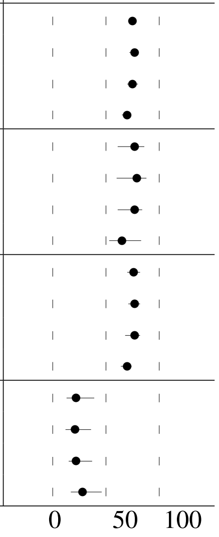
Breast Cancer

	SAP	25%	50%	75%	Q1 median Q3
acc	sap0	68	74	75	
	sap1^{00}	67	71	75	
	sap2^{-}	66	70	72	
	sap3^{-}	67	69	70	
pd	sap0	24	56	91	
	sap1^{00}	25	63	90	
	sap2^{00}	24	53	89	
	sap3^{00}	27	56	85	
prec	sap0	50	73	79	
	sap1^{00}	50	70	77	
	sap2^{00}	44	66	74	
	sap3^{00}	43	68	75	
pf	sap0	8	21	74	
	sap1^{00}	10	21	73	
	sap2^{00}	11	24	74	
	sap3^{00}	14	23	71	
					0 50 100



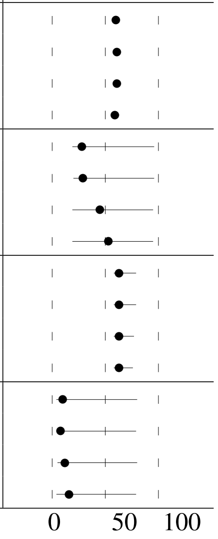
Heart

	SAP	25%	50%	75%	Q1 median Q3
acc	sap0	71	75	78	
	sap1^{00}	72	77	79	
	sap2^{00}	70	75	80	
	sap3^{-}	65	70	72	
pd	sap0	61	77	86	
	sap1^{00}	60	79	88	
	sap2^{00}	61	77	84	
	sap3^{-}	53	65	83	
prec	sap0	70	76	82	
	sap1^{00}	71	77	82	
	sap2^{00}	68	77	82	
	sap3^{-}	64	70	74	
pf	sap0	13	22	39	
	sap1^{00}	12	21	36	
	sap2^{00}	15	22	37	
	sap3^{-}	17	28	46	
					0 50 100



Mammography

	SAP	25%	50%	75%	Q1 median Q3
acc	sap0	60	60	63	
	sap1^{00}	60	61	63	
	sap2^{00}	59	61	63	
	sap3^{00}	57	59	63	
pd	sap0	19	28	96	
	sap1^{00}	20	29	96	
	sap2^{00}	19	45	95	
	sap3^{00}	19	53	95	
prec	sap0	58	63	79	
	sap1^{00}	58	63	79	
	sap2^{00}	58	63	77	
	sap3^{00}	58	63	76	
pf	sap0	4	10	80	
	sap1^{00}	4	8	79	
	sap2^{00}	5	12	80	
	sap3^{00}	4	16	79	
					0 50 100



Vote

	SAP	25%	50%	75%	Q1 median Q3
acc	sap0	91	93	94	
	sap1^{00}	90	92	93	
	sap2^{-}	85	89	91	
	sap3^{-}	75	83	85	
pd	sap0	87	92	96	
	sap1^{00}	85	91	96	
	sap2^{00}	76	86	96	
	sap3^{00}	56	81	98	
prec	sap0	90	93	94	
	sap1^{00}	90	92	94	
	sap2^{-}	86	90	94	
	sap3^{-}	77	85	92	
pf	sap0	4	8	11	
	sap1^{00}	4	9	15	
	sap2^{00}	4	10	22	
	sap3^{00}	2	8	43	
					0 50 100

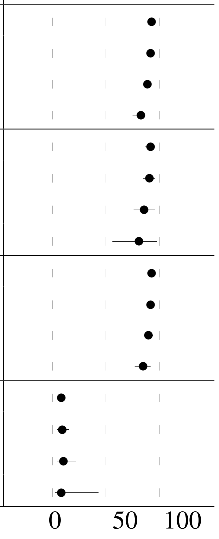


Figure 4.5: 1-NN classification results for scenario 1

4.4 Experiment #2: For S2

The goal here is to see if the CLIFF works well in the presence of noise. This is important because according to [?]:

> In the presence of class noise, ... there are two main problems that can occur. The first is that very few instances will be removed from the training set because many instances

are needed to maintain the noisy (and thus overly complex) decision boundaries. The second problem is that generalization accuracy can suffer, especially if noisy instances are retained while good instances are removed.

With that said, in this experiment we repeat Experiment 1, only this time noise is introduced to training data by randomly changing the target class of 10% of the instances in the training data to any other target class value. The the results here will indicate how well the 1NN classifier along with the different PLS are able to predict the correct target class even if some of its training data is faulty [**?**].

4.4.1 Results from Experiment 2

The *noisy* tables in Figure **??** to Figure **??** displays the results of Experiment 2. The first thing to recognize is that compared to the $clean$ tables, there is a general degradation of the pd and pf results for each data set. However in some cases such as the pd results for *bc* the pds can degrade by as little as 1%. The second thing to recognize is that while the *sizes* of the training data basically remains the same for CLIFF (differences of no more that 2%) for all data sets, the other PLS display a general increase in the training set sizes except for PSC whose training size decrease from 42% to 40%.

4.5 Threats to Validity

4.5.1 Construct Validity

4.5.2 Internal Validity

4.5.3 Conclusion Validity

4.5.4 External Validity

Collectively, the results of the above experiments indicate that CLIFF may be effective in the field of forensic interpretation where a very low false alarm rate pf is desired. Here, CLIFF can be used to help lower the pfs of a forensic interpretation model. The results of Experiment 1 indicate this possibility in the median pf results where CLIFF's values ranges from 0 to 47 and are lower or the same as the baseline (KNN) pf results.

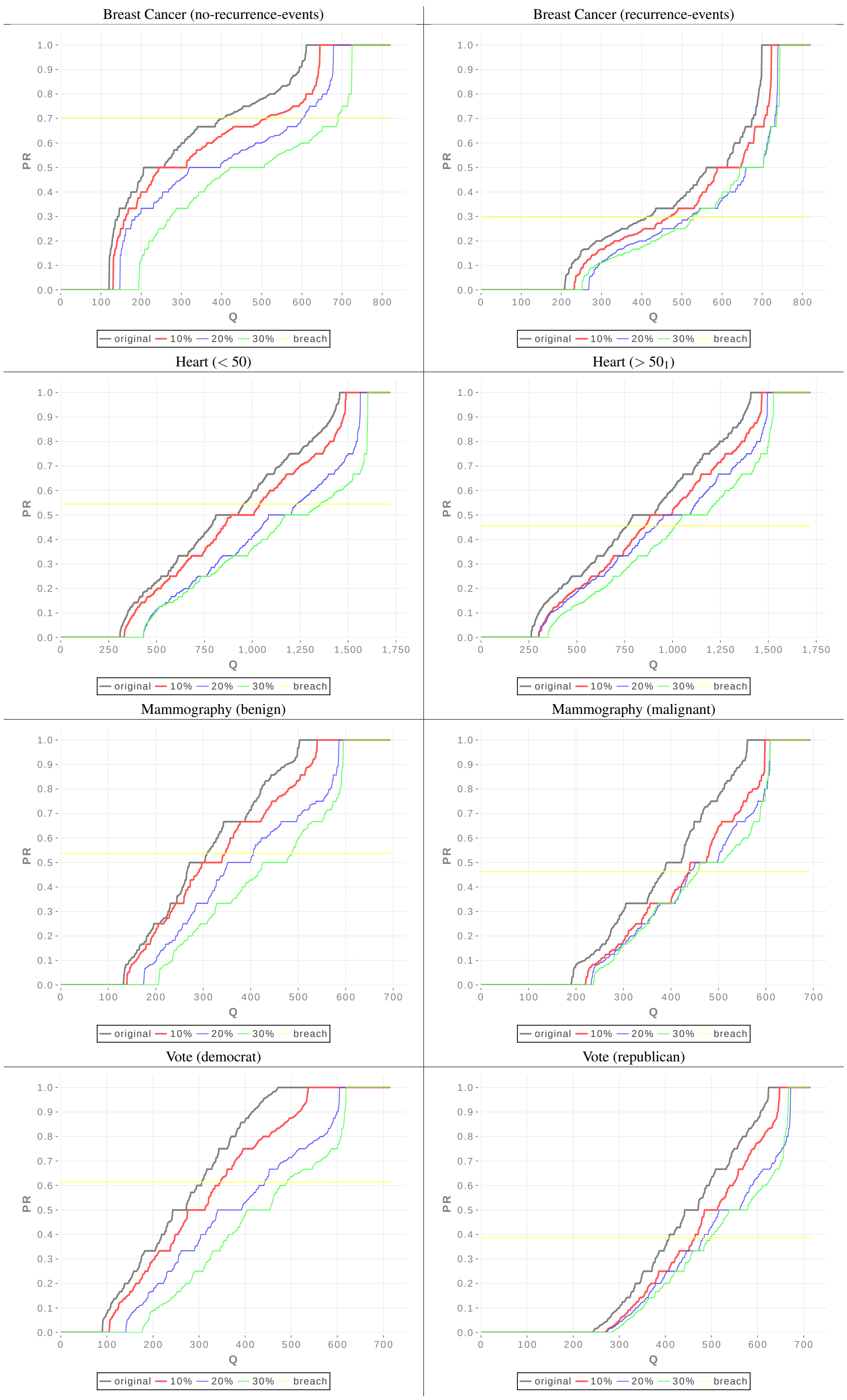
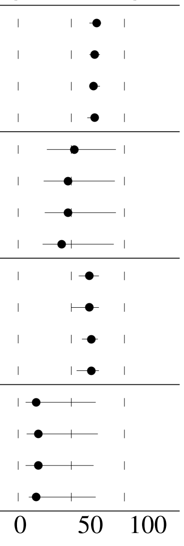


Figure 4.6: Sensitive attribute disclosure

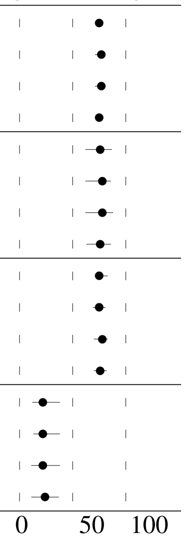
Breast Cancer

	SAP	25%	50%	75%	Q1 median Q3
acc	sap0	67	74	78	
	sap1^{00}	67	72	77	
	sap2^{00}	67	71	77	
	sap3^{00}	65	72	75	
pd	sap0	27	53	92	
	sap1^{00}	24	47	91	
	sap2^{00}	25	47	92	
	sap3^{00}	23	41	90	
prec	sap0	57	67	76	
	sap1^{00}	50	67	76	
	sap2^{00}	60	69	75	
	sap3^{00}	55	69	76	
pf	sap0	7	17	73	
	sap1^{00}	8	19	75	
	sap2^{00}	7	19	71	
	sap3^{00}	10	17	73	



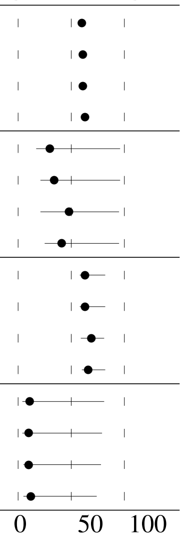
Heart

	SAP	25%	50%	75%	Q1 median Q3
acc	sap0	72	75	78	
	sap1^{00}	71	77	78	
	sap2^{00}	71	77	79	
	sap3^{00}	72	75	79	
pd	sap0	62	76	87	
	sap1^{00}	62	78	86	
	sap2^{00}	62	78	88	
	sap3^{00}	63	76	86	
prec	sap0	71	75	83	
	sap1^{00}	69	75	81	
	sap2^{00}	70	78	83	
	sap3^{00}	70	76	82	
pf	sap0	12	22	38	
	sap1^{00}	13	22	38	
	sap2^{00}	11	22	38	
	sap3^{00}	11	24	37	



Mammography

	SAP	25%	50%	75%	Q1 median Q3
acc	sap0	58	60	62	
	sap1^{00}	59	61	64	
	sap2^{00}	59	61	64	
	sap3^{++}	60	63	66	
pd	sap0	17	30	96	
	sap1^{00}	21	34	96	
	sap2^{00}	21	48	95	
	sap3^{00}	25	41	95	
prec	sap0	58	63	82	
	sap1^{00}	58	63	82	
	sap2^{00}	59	69	81	
	sap3^{00}	60	66	82	
pf	sap0	4	11	81	
	sap1^{00}	4	10	79	
	sap2^{00}	5	10	78	
	sap3^{00}	5	12	74	



Vote

	SAP	25%	50%	75%	Q1 median Q3
acc	sap0	90	92	95	
	sap1^{00}	90	92	95	
	sap2^{00}	90	92	94	
	sap3^{00}	91	92	95	
pd	sap0	88	92	96	
	sap1^{00}	88	92	95	
	sap2^{00}	88	93	96	
	sap3^{00}	88	92	96	
prec	sap0	89	93	96	
	sap1^{00}	89	93	96	
	sap2^{00}	88	93	95	
	sap3^{00}	89	94	96	
pf	sap0	4	8	11	
	sap1^{00}	4	8	11	
	sap2^{00}	4	7	12	
	sap3^{00}	3	8	11	

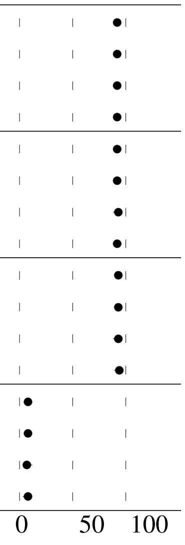


Figure 4.7: 1-NN classification results for S2

Chapter 5

Conclusions and Future Work

We have presented an overview of PPDP with regards to a single data-set. The importance of reliable methods in PPDP is made clear, in that, if data providers have no confidence in the privacy methods proposed, they will not publish data for the purpose of research. In that vein, these data providers should be provided with not only, methods and tools to disguise their data before being published, but also evaluation measures to help them decided how effective the chosen tool is and whether or not they accept the level of privacy protection offered by the technique.

Appendix A

Implementation of SAP and Support Code

A.1 SAP Functions

```
(defn get-queries [data-k num1]
  (loop [pre nil terminator 0 result []]
    (if (= terminator num1)
      result
      (recur result
             (if (= pre result) (inc terminator) 0)
             (let [myone (let [que (get-query data-k)]
                            (if (and (not (member? que result))
                                     (= true
                                        (query-worth
                                         que
                                         data-k)))
                              que
                              nil))
                   ans (if (= nil myone)
                         result
                         (conj result myone))]
                 ans)))))
```

Figure A.1: CODE: get-queries.clj

```
(defn get-rules [D]
  (let [br (my-best-rest D)]
    (loop [br1 br result []]
      (if (empty? br1)
        result
        (recur (rest br1)
               (conj
                result
                (rank-vals D
                           (first (first br1)) (second (first br1)))))))))
```

Figure A.2: CODE: `get-rules.clj`

```
(defn sap-S1 [D p]
"For S1, where only the QIDs are suppressed"
  (let [crits (get-rules D)
        insts (mygroup D)
        result (map #(suppress-instances %1 %2 p) crits insts)]
    (matrix (apply concat result))))

(defn suppress-instances [crit inst p]
  (let [i (last (trans inst))]
  (loop [c (sort-by last crit) result []] ;change made for class only
    (if (empty? c)
      (trans (bind-rows (matrix result) (last (trans inst))))
      (recur
       (rest c)
       (conj result (si-percent1 p (first (first c)) i)))))))

(defn si-percent1 [p c ilst]
  (let [num1 (Math/ceil (* p (count ilst)))
        n1 (sample (range (count ilst)) :size num1 :replacement false)
        n (if (not= nil (nrow n1)) n1 (vector n1))]
    (loop [x 0 i ilst result []]
      (if (empty? i)
        result
        (recur
         (inc x)
         (rest i)
         (conj result (if (member? x n) -10000 (first i))))))))
```

Figure A.3: CODE: `sap-S1.clj`

```
(defn sap-S2 [D p]
"For S2, where only the SAV is suppressed"
  (let [insts (mygroup D)
        result (map #(si-percent p %) insts)]
    (matrix (apply concat result))))

(defn si-percent [p inst]
  (let [data (trans (butlast (trans inst)))
        ilst (last (trans inst))
        num1 (Math/ceil (* p (count ilst)))
        n1 (sample (range (count ilst)) :size num1 :replacement false)
        n (if (not= nil (nrow n1)) n1 (vector n1))]
    (loop [x 0 i ilst result []]
      (if (empty? i)
        (bind-columns data (matrix result))
        (recur
         (inc x)
         (rest i)
         (conj result (if (member? x n) -10000 (first i))))))))
```

Figure A.4: CODE: `sap-S2.clj`

A.2 k-Nearest Neighbor

```
(defn knn-classifier [one kn D distance]
  (if (= (nrow D) 1) (last D)
      (let [k-nearest (map
                       second
                       (take kn
                             (sort-by
                              first
                              (map #(vector (distance one %) %)
                                   D))))
            klass (if (= kn 1)
                    [(last (first k-nearest)) (last (first k-nearest))]
                    (last (Transpose k-nearest)))
            classification (k-majority klass)]
        classification)))

(defn knn [n k data distance n1]
  (let [trainer (fn [data]
                  (matrix (filter #(not= -10000 (last %)) (sap data n1))))
        tester (fn [one model]
                 (vector (last one)
                         (knn-classifier
                          (butlast one)
                          k
                          model
                          distance)))
        klasses (fn []
                  (uc (to-vect (sel data :cols (- (ncol data) 1)))))]
    (let [answer (nway-new n k data trainer tester distance)
          ppp (to-vect
               (trans
                (apply
                 concat
                 (map #(abcd-stats
                        (first %)
                        (second %)
                        (klasses))
                      answer))))]
      ppp)))
```

Figure A.5: CODE: `knn.clj`

A.3 Experiment

```
(defn demo001worker [n k D distance]
  (let [sap0 (knn10 n k D distance 0.0)
        sap1 (knn10 n k D distance 0.1)
        sap2 (knn10 n k D distance 0.2)
        sap3 (knn10 n k D distance 0.3)]
     (concat
      ["#data learner acc pd prec pf"]
      (Transpose (vector (repeat (count (nth sap0 1)) 10)
                         (repeat (count (nth sap0 1)) 'sap0)
                         (to-vect (nth sap0 0))
                         (to-vect (nth sap0 1))
                         (to-vect (nth sap0 2))
                         (to-vect (nth sap0 3))))
      (Transpose (vector (repeat (count (nth sap1 1)) 10)
                         (repeat (count (nth sap1 1)) 'sap1)
                         (to-vect (nth sap1 0))
                         (to-vect (nth sap1 1))
                         (to-vect (nth sap1 2))
                         (to-vect (nth sap1 3))))
      (Transpose (vector (repeat (count (nth sap2 1)) 10)
                         (repeat (count (nth sap2 1)) 'sap2)
                         (to-vect (nth sap2 0))
                         (to-vect (nth sap2 1))
                         (to-vect (nth sap2 2))
                         (to-vect (nth sap2 3))))
      (Transpose (vector (repeat (count (nth sap3 1)) 10)
                         (repeat (count (nth sap3 1)) 'sap3)
                         (to-vect (nth sap3 0))
                         (to-vect (nth sap3 1))
                         (to-vect (nth sap3 2))
                         (to-vect (nth sap3 3))))) ))
```

Figure A.6: CODE: `experiment.clj`

Bibliography

[1] Justin Brickell and Vitaly Shmatikov. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 70–78, New York, NY, USA, 2008. ACM.

[2] T Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 1977.

[3] Josep Domingo-Ferrer and Ursula Gonzalez-Nicolas. Hybrid microdata using microaggregation. *INFORMATION SCIENCES*, 180(15):2834–2844, AUG 1 2010.

[4] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[5] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. Privacy-Preserving Data Publishing: A Survey of Recent Developments. *ACM COMPUTING SURVEYS*, 42(4), JUN 2010.

[6] O. Jalali, T. Menzies, and M. Feather. Optimizing requirements decisions with keys. In *Proceedings of the PROMISE 2008 Workshop (ICSE)*, 2008. Available from `http://menzies.us/pdf/08keys.pdf`.

[7] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. Workload-aware anonymization techniques for large-scale datasets. *ACM TRANSACTIONS ON DATABASE SYSTEMS*, 33(3), AUG 2008.

[8] Nissim Matatov, Lior Rokach, and Oded Maimon. Privacy-preserving data mining: A feature set partitioning approach. *INFORMATION SCIENCES*, 180(14):2696–2720, JUL 15 2010.

[9] Hyoungmin Park and Kyuseok Shim. Approximate algorithms with generalizing attribute values for k-anonymity. *INFORMATION SYSTEMS*, 35(8):933–955, DEC 2010.

[10] Vicenc Torra, Yasunori Endo, and Sadaaki Miyamoto. ON THE COMPARISON OF SOME FUZZY CLUSTERING METHODS FOR PRIVACY PRESERVING DATA MINING: TOWARDS THE DEVELOPMENT OF SPECIFIC INFORMATION LOSS MEASURES. *KYBERNETIKA*, 45(3):548–560, 2009.

[11] Burak Turhan, Tim Menzies, Ayse B. Bener, and Justin Di Stefano. On the relative value of cross-company and within-company data for defect prediction. *EMPIRICAL SOFTWARE ENGINEERING*, 14(5):540–578, OCT 2009.

[12] VS Verykios, E Bertino, IN Fovin, LP Provenza, Y Saygin, and Y Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD RECORD*, 33(1):50–57, MAR 2004.

[13] Shyue-Liang Wang, Bhavesh Parikh, and Ayat Jafarl. Hiding informative association rule sets. *EXPERT SYSTEMS WITH APPLICATIONS*, 33(2):316–323, AUG 2007.

[14] Nan Zhang and Wei Zhao. Privacy-preserving data mining systems. *COMPUTER*, 40(4):52+, APR 2007.

[15] Dan Zhu, Xiao-Bai Li, and Shuning Wu. Identity disclosure protection: A data reconstruction approach for privacy-preserving data mining. *DECISION SUPPORT SYSTEMS*, 48(1, Sp. Iss. SI):133–140, DEC 2009.