Chapter 2
Overview of Patient Data Anonymization

2.1 Anonymizing Demographics

2.1.1 Anonymization Principles

Protecting demographics can be achieved using *perturbative* methods, such as noise addition and data swapping [1], as mentioned in the Introduction. However, these methods fail to preserve data truthfulness (e.g., they may change the age of a patient from 50 to 10), which can severely harm the usefulness of the published patient data. *Non-perturbative* methods preserve data truthfulness, and thus are more suitable for anonymizing patient demographics. We will discuss these methods later in this chapter, but, for now, note that they can be used to enforce anonymization principles, such as k-anonymity [17, 18, 58, 59], which is illustrated below.

Definition 2.1 (k-Anonymity). k-Anonymity is satisfied when each tuple in a table $T(a_1,\ldots,a_d)$, where a_i, $i = 1,\ldots,m$ are quasi-identifiers (QIDs), is indistinguishable from at least $k-1$ other tuples in T w.r.t. the set $\{a_1,\ldots,a_m\}$ of QIDs.

This principle requires each tuple in a table T to contain the same values in the set of quasi-identifier attributes (QIDs) with at least $k-1$ other tuples in T. Recall from Introduction that the set of quasi-identifiers contains, typically innocuous, attributes that can be used to link external data sources with the published table. Satisfying k-anonymity offers protection against identity disclosure, because the probability of linking an individual to their true record, based on QIDs, is no more than $\frac{1}{k}$. The parameter k controls the level of offered privacy and is set by data publishers, usually to five in the context of patient demographics [17]. We also note that not all attributes in T need to be QIDs (i.e., it may be that $m < d$), and that an individual may not be willing to be associated with some of these attributes. The latter attributes are referred to as *sensitive* attributes (SAs), and we will examine them shortly. The process of enforcing k-anonymity is called k-anonymization, and

A. Gkoulalas-Divanis and G. Loukides, *Anonymization of Electronic Medical Records to Support Clinical Analysis*, SpringerBriefs in Electrical and Computer Engineering,
DOI 10.1007/978-1-4614-5668-1_2, © The Author(s) 2013

Table 2.1 (**a**) Original dataset, and (**b**), (**c**) two different four-anonymous versions of it

Id	Postcode	Expense (K)
t_1	NW10	*10*
t_2	NW15	*10*
t_3	NW12	*10*
t_4	NW13	*10*
t_5	NW20	*20*
t_6	NW30	*40*
t_7	NW30	*40*
t_8	NW25	*30*

(a)

Id	Postcode	Expense (K)
t_1	*	*10*
t_2	*	*10*
t_3	*	*10*
t_4	*	*10*
t_5	*	*20*
t_6	*	*40*
t_7	*	*40*
t_8	*	*30*

(b)

Id	Postcode	Expense (K)
t_1	NW[10–15]	*10*
t_2	NW[10–15]	*10*
t_3	NW[10–15]	*10*
t_4	NW[10–15]	*10*
t_5	NW[20–30]	*20*
t_6	NW[20–30]	*40*
t_7	NW[20–30]	*40*
t_8	NW[20–30]	*30*

(c)

Table 2.2 Summary of privacy principles for guarding against sensitive information disclosure

Reference	Type of sensitive information disclosure
[11,47,63,66,67]	Value disclosure
[37]	Semantic disclosure
[32,35,44,45,67]	Range disclosure

it can be performed by partitioning T into groups of at least k tuples, and then transforming the QID values in each group, so that they become indistinguishable from one another. Formally, k-anonymization is explained below.

Definition 2.2 (k-Anonymization). k-Anonymization is the process in which a table $T(a_1,\ldots,a_d)$, where $a_i, i = 1,\ldots,m$ are quasi-identifiers (QIDs), is partitioned into groups $\{g_1,\ldots,g_h\}$ s.t. $|g_j| \geq k$, $j = 1,\ldots,h$, where $|g_j|$ denotes the size of g_j (i.e., number of tuples contained in g_j), and tuples in each g_j are made identical w.r.t. QIDs.

Table 2.1b and c, for example, are both 4-anonymous; *Postcode* is a QID and *Expense* is an SA. These tables were derived by forming two groups of tuples, one containing $\{t_1,\ldots,t_4\}$ and another containing $\{t_5,\ldots,t_8\}$, and then assigning the same value in *Postcode* to all tuples in each group. Specifically, the *Postcode* values in Table 2.1b have been replaced by a value $*$, which is interpreted as "any postcode value", while, in Table 2.1c, by a new value formed by taking the range of all *Postcode* values in a group.

Note that an individual's sensitive information may be disclosed, even when data are anonymized using a "large" k [47]. Specifically, we can distinguish among three types of sensitive information disclosure, which are summarized in Table 2.2. These types of disclosure have not been examined by the medical informatics community, partly because they have not led to reported privacy breaches [16]. However, we report these types of disclosure, for completeness.

Value disclosure involves the inference of an individual's value in a sensitive attribute (SA), such as *Expense* in Table 2.1c. As an example, consider Table 2.1c and an attacker, who knows that an individual lives in an area with $Postcode = NW10$. This allows the attacker to infer that this individual's expense is 10 K.

To prevent value disclosure, an anonymization principle, called l-diversity, was proposed in [47]. This principle requires each anonymized group in T to contain at least l "well represented" SA values [47]. The simplest interpretation of "well represented" is "distinct" and leads to a principle called *distinct* l-diversity [37], which requires each anonymized group to contain at least l distinct SA values. Other principles that guard against value disclosure by limiting the number of distinct SA values in an anonymized group are (a,k)-anonymity [66] and p-sensitive-k-anonymity[63]. However, all these principles still allow an attacker to conclude that an individual is likely to have a certain sensitive value, when that value appears much more frequently than others in the group.

A principle, called recursive (c,l)-diversity [47], addresses this limitation, as explained in Definition 2.3.

Definition 2.3 (Recursive (c, l)-diversity). Assume that a table $T(a_1,\ldots,a_m,sa)$, where $\{a_1,\ldots,a_m\}$ are QIDs and sa is an SA, is partitioned into groups $\{g_1,g_2,\ldots,g_h\}$, such that $|g_j| \geq k$, $j = 1,\ldots,h$, and tuples in g_j will have the same values in each QID after anonymization. Given parameters c,l, which are specified by data publishers, a group g_j is (c,l)-diverse when $r_1 < c \times (r_l + r_{l+1} + \ldots + r_n)$, where $r_i, i \in \{1,\ldots,n\}$ is the number of times the i-th frequent SA value appears in g_j, and n is the domain size of g_j. T is (c,l)-diverse when every g_j, $j = 1,\ldots,h$ is (c,l)-diverse.

Recursive (c,l)-diversity requires each group in T to contain a large number of distinct *SA* values, none of which should appear "too" often. Observe, for example, that the second group of Table 2.1c satisfies recursive $(2,2)$-diversity. This is because it contains three distinct values, whose frequencies in descending order are $r_1 = 2, r_2 = 1$ and $r_3 = 1$, and we have $r_1 < 2 \times (r_2 + r_3)$.

More recently, an anonymization principle, called *privacy skyline*, that can prevent attackers with three different types of background knowledge to infer individuals' sensitive values was proposed in [11]. Privacy skyline considers attackers with knowledge about SA values that an individual I does not have, knowledge about SA values belonging to another individual, and knowledge that a group of individuals, in which I is included, has a certain SA value. We believe that this principle is well-suited to achieve protection of datasets that contain familial relationships. However, we do not consider such datasets in this book.

Semantic disclosure occurs when an attacker can make inferences, related to SA values in an anonymous group, that they cannot make by observing the SA values in the entire dataset [37]. Consider, for example, the distribution of *Expense* values in the first group in Table 2.1c, and observe that it differs from the distribution of all values in the same attribute in Table 2.1c. This group risks semantic disclosure, because it reveals information that cannot be inferred from the entire dataset. Semantic disclosure can be thwarted by t-closeness, a principle that calls for limiting the distance between the probability distribution of the SA values in an anonymized group and that of SA values in the whole dataset [37]. The smaller the distance, the higher the level of protection achieved.

Range disclosure occurs when sensitive information is inferred in the form of sensitive values [32, 35, 44, 45, 67]. Consider, for example, that a ten-anonymous group contains three distinct values 10, 11, and 12 K in *Expense*, which is an SA. Knowing that an individual's tuple is contained in this group, an attacker can infer the range (10–12 K) for this individual's expense. Clearly, when this range is "small", the disclosure may be considered as sensitive. Anonymization principles to guard against range disclosure by limiting the maximum range of SA values in a group of tuples have been proposed by Loukides et al. [44] and Koudas et al. [32], while LeFevre et al. [35] proposed limiting the variance of sensitive values instead. Xiao et al. [67] assumed that sensitive ranges are determined by individuals themselves and proposed, *personalized privacy*. This principle forestalls range disclosure by limiting the probability of associating an individual with their specified range, and is enforced through generalizing SA values. This may be inappropriate for medical analysis tasks in which SAs should remain intact. A principle, called *Worst Group Protection* (WGP), which prevents range disclosure and can be enforced without generalization of SA values was proposed in [45]. WGP measures the probability of disclosing any range in the least protected group of a table, and captures the way SA values form ranges in a group, based on their frequency and semantic similarity.

2.1.2 Anonymization Algorithms

Most anonymization algorithms to protect patient demographics work in two steps; first, they form groups of tuples in a way that optimizes data utility and/or privacy protection, and then transform QID values to enforce an anonymization principle. In the following, we review existing algorithms in terms of search strategies, optimization objectives, and value recoding models.

Search strategies Achieving k-anonymization with minimum information loss is an NP-hard problem [4, 8, 49, 68], thus many methods employ heuristic search strategies to form k-anonymous groups. Samarati [58] proposed a binary search on the height of DGHs, LeFevre et al. [33] suggested a search similar in principle to the Apriori [5] used in association rule mining, and Iyengar [31] used a genetic algorithm. Partitioning has also been used to form groups in k-anonymization. LeFevre et al. [34, 35] examined several partitioning strategies including techniques originally proposed for kd-tree construction [22], while Iwuchukwu et al. [30] developed a set of heuristics inspired from R-tree construction [25]. Several k-anonymization algorithms are based on clustering [2, 36, 44, 51, 68]. The main objective of these methods is to form groups that optimize a particular objective criterion. In order to do so, they perform greedy search constructing groups in a bottom-up [2, 36, 44, 51] or a top-down fashion [68]. Figure 2.1 provides a classification of heuristic search strategies according to the type of search they adopt. Furthermore, approximation algorithms for the problem of optimal k-anonymity under (simple) information loss measures have been proposed in [4, 49, 53].

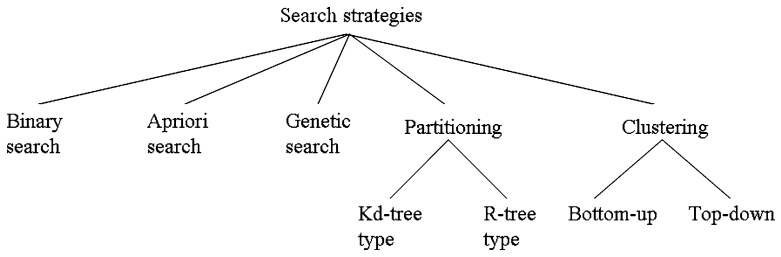

Fig. 2.1 A classification of heuristic search strategies

Table 2.3 Summary of existing grouping strategies w.r.t. their objectives

Reference	Optimization objective	Utility	Sens. inf. protection
[6, 33, 34]	Group-size constrained	Optimal	No guarantee
[8, 35, 68]			
[19]	Utility constrained	Guarantee	No guarantee
[32, 37, 47, 67]	Privacy constrained	Optimal	Guarantee
[44]	Trade-off based	Traded-off with privacy	Traded-off with utility
[46]	Utility-and-privacy constrained	Guarantee	Guarantee

Optimization objectives Anonymization algorithms fall into five categories with respect to their optimization objectives, as can be seen in Table 2.3. *Group-size constrained* algorithms attempt to achieve a maximum level of data utility, subject to a minimum anonymous group size requirement, expressed as k [6, 8, 33–35, 68]. Other algorithms bound the level of information loss incurred during anonymization to ensure that data remain useful for applications, and are referred to as *utility constrained*. However, both group-size and utility constrained algorithms may result in an unacceptably low level of privacy protection from sensitive information disclosure [47]. In response, *privacy constrained* algorithms [32, 37, 47, 67] introduce additional protection constraints (e.g., a minimum level in l-diversity) that released data must satisfy. Another way to deal with utility and privacy is to treat both of them as optimization objectives and attempt to achieve a desired trade-off between them. This *trade-off based* approach, was investigated in [44]. It should be noted, however, that none of the aforementioned approaches can guarantee that data publishers' data utility and privacy protection requirements are satisfied in the anonymized data. In response, a *utility-and-privacy constrained* approach, which allows the specification and enforcement of utility and privacy requirements, was proposed in [46].

Value recoding models After deriving groups of tuples that attempt to optimize their objectives, anonymization algorithms recode QID values using suppression [59], microaggregation [13, 14], or generalization [58, 58]. Suppression suggests eliminating specific QID values, or entire records from the published data [33], while microaggregation involves replacing a group of QID values using the group

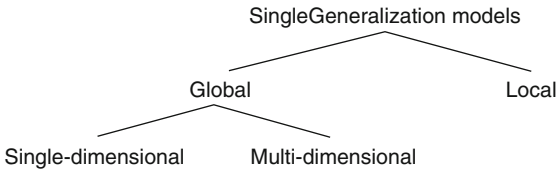

Fig. 2.2 Summary of generalization models

centroid [13] or median value [14] for numerical and categorical QIDs, respectively. Both of these techniques, however, may cause high information loss [33]. Generalization suggests replacing QID values by more general but semantically consistent ones [58, 59]. Thus, suppression can be thought of as the special case of generalization, where all values in a QID are generalized to the most general value (i.e., a value that can be interpreted as any value in the domain of the QID) [4].

Generalization models can be classified into *global* and *local*. Global generalization models involve mapping the domain of QIDs into generalized values [6, 33], and are further grouped into single and multi-dimensional. In the former models, the mapping of a QID value to a generalized value is performed for each QID separately, whereas in the latter ones, the multi-attribute domain of QID values is recoded. On the other hand, in local models, QID values of individual tuples are mapped into generalized values on a group-by-group basis [68]. The different types of generalization models that have been proposed are summarized in Fig. 2.2. For an excellent discussion of these models and formal definitions, we refer the reader to [33, 38].

2.2 Anonymizing Diagnosis Codes

Electronic medical records contain clinical data, such as patients' diagnoses, laboratory results, active medication, and allergies, as discussed in Introduction. While publishing any patient information could, in principle, breach patient privacy, it is important to recognize that publishing different types of information poses different levels of privacy risk. To estimate the level of risk, the principles of *replication* (i.e., the frequency an attribute value appears in an individual's electronic medical record), *resource availability* (i.e., the number and accessibility of datasets, that are external to an individual's electronic medical record and contain the individual's attribute value), and *distinguishability* (i.e., the extent to which one or more attribute values can be used to re-identify an individual) can be used as a guide. These principles build on those defined by the Federal Committee on Statistical Methodology [20] and are acknowledged by health privacy experts [48]. Based on these principles, it can be seen that diagnosis codes have high replication, because an electronic medical record contains all diagnosis codes a patient has been assigned to during multiple hospital visits, and high resource availability, as they are contained in publicly available hospital discharge summaries. Furthermore, as

a

Name	*Diagnoses*
Anne	a b c d e f
Greg	a b e g
Jack	a e
Tom	b f g
Mary	a b
Jim	c f

b

Diagnoses
(a,b,c,d,e,f,g)
(a,b,c,d,e,f,g)
(a,b,c,d,e,f,g)
(a,b,c,d,e,f,g)
(a,b,c,d,e,f,g)
(a,b,c,d,e,f,g)

c

Diagnoses
(a,b,c) (d,e,f)
(a,b,c) (d,e,f) g
(a,b,c) (d,e,f)
(a,b,c) (d,e,f) g
(a,b,c)
(a,b,c) (d,e,f)

Fig. 2.3 An example of: (**a**) original dataset, and (**b**), (**c**) two anonymized versions of it

we will explain in the next chapter, diagnosis codes are highly distinguishable. Thus, publishing diagnosis codes may lead to the disclosure of patients' identity [40], and anonymization of diagnosis codes can be employed to eliminate this threat.

2.2.1 Anonymization Principles

Anonymizing diagnosis codes is a challenging computational problem, because only a small number out of thousands of possible diagnosis codes are assigned to a patient. In fact, high-dimensional and sparse data are notoriously difficult to anonymize [3], because, intuitively, it is difficult to find values that are sufficiently similar as to be anonymized with "low" information loss. At the same time, the number of diagnosis codes that are associated to a patient may vary significantly. Due to these reasons, it is difficult to anonymize diagnosis codes by employing the anonymization principles and algorithms that have been designed for demographics and were discussed in Sect. 2.1.1. At the same time, somewhat surprisingly, the medical informatics community has focused on anonymizing demographics [17, 52, 59], but not diagnosis codes. In fact, due to their semantics, a patient-level dataset containing diagnosis codes can be modeled as a transaction dataset. That is, data in which a record (also called *transaction*) corresponds to a different patient and contains the set of diagnosis codes that have been assigned to the patient, as shown in Fig. 2.3a.

To describe transaction data, we employ the terminology of the frequent itemset mining framework [5]. Specifically, diagnosis codes are represented as *items* that are derived from a finite set $\mathscr{I} = \{i_1, \ldots, i_M\}$, such as the set of all ICD-9 codes.[1] A subset I of $\mathscr{I}$ is called an *itemset*, and is represented as the concatenation of the items it contains. An itemset that has m items, or equivalently a *size* of m, is called an m-itemset and its size is denoted with $|I|$. For instance, the set of diagnosis codes $\{a,b,c\}$, in the first record of Fig. 2.3a is a three-itemset. A dataset $\mathscr{D} = \{T_1, \ldots, T_N\}$

[1] ICD-9 codes are described in the International Classification of Diseases, Ninth Revision – Clinical Modification, http://www.cdc.gov/nchs/icd/icd9cm.htm

is a set of N records, called *transactions*, and each transaction T_n in $\mathscr{D}$ corresponds to a unique patient. A transaction is a pair $T_n = \langle tid, I \rangle$, where *tid* is a unique identifier,[2] and I is the itemset. A transaction $T_n = \langle tid, J \rangle$ *supports* an itemset I, if $I \subseteq J$. Given an itemset I in $\mathscr{D}$, we use $sup(I, \mathscr{D})$ to represent the number of transactions $T_n \in \mathscr{D}$ that support I. For example, the support of the itemsets $\{a, b\}$ and $\{a, b, c\}$ in the dataset of Fig. 2.3a is 3 and 1, respectively.

Using the above notation, we review anonymization principles for publishing patients' diagnosis codes, starting from the most specific to the more general ones.

Complete *k*-anonymity A k-anonymity-based principle, called *complete k-anonymity*, for anonymizing transaction datasets was proposed by He et al. [28]. This principle assumes that any itemset (i.e., combination of diagnosis codes) in a transaction can lead to identity disclosure and requires each transaction to be indistinguishable from at least $k-1$ other transactions, based on any of these combinations. The following definition explains the concept of complete k-anonymity.

Definition 2.4 (Complete *k*-anonymity). Given a parameter k that is specified by data publishers, a transaction dataset $\mathscr{D}$ satisfies complete k-anonymity when $sup(I_j, \mathscr{D}) \geq k$, for each itemset I_j of a transaction $T_j = \langle tid_j, I_j \rangle$ in $\mathscr{D}$, with $j \in [1, N]$.

Observe that satisfying complete k-anonymity guarantees that an attacker cannot link a patient's identity to fewer than k transactions of the anonymized dataset. For instance, consider the dataset in Fig. 2.3b, in which items a to d have been replaced by a generalized item (a, b, c, d), interpreted as any non-empty subset of $abcd$. This dataset satisfies complete six-anonymity, hence a patient's identity cannot be linked to fewer than six transactions, based on any combination of the diagnosis codes a to d. The authors of complete k-anonymity implicitly assume that attackers may know all the diagnosis codes contained in a patient's transaction. However, this assumption is considered as too strict in most diagnosis code publishing scenarios [41], because, typically, only certain combinations of diagnosis codes of a patient are published [40]. For instance, an attacker who attempts to link the published dataset to hospital discharge records, can only use sets of diagnosis codes that were assigned to a patient during a single hospital visit [40, 41]. Thus, anonymizing a dataset to satisfy complete k-anonymity may result in unnecessary information loss.

k^m-anonymity Terrovitis et al. [60] assume that, due to the semantics of transaction data, it may be difficult for an attacker to learn more than a certain number of a patient's diagnosis codes. Based on this assumption, the authors of [60] proposed k^m-anonymity, which thwarts attackers who know *any* combination of at most m diagnosis codes. This principle is explained in Definition 2.5, and it ensures that no m itemset can be used to associate an individual with fewer than k transactions in the published dataset.

[2] The identifier is used only for reference and may be omitted, if this is clear from the context.

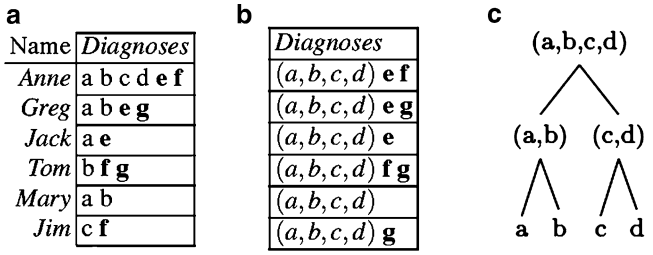

Fig. 2.4 An example of: (**a**) original dataset containing public and sensitive items, (**b**) a $(0.5, 6, 2)$-coherent version of it, and (**c**) a generalization hierarchy

Definition 2.5 (k^m-anonymity). Given parameters k and m, which are specified by data publishers, a dataset $\mathscr{D}$ satisfies k^m-anonymity when $sup(I, \mathscr{D}) \geq k$, for each m-itemset I in $\mathscr{D}$.

The dataset shown in Fig. 2.3a, for example, does not satisfy 2^2-anonymity, because the combination of diagnosis codes ac appears only in one transaction. As an example of a dataset that satisfies 2^2-anonymity, consider the dataset shown in Fig. 2.3c. In the latter dataset, items a to c have been replaced by a generalized item (a, b, c), which is interpreted as any non-empty subset of abc, while items d to f have been replaced by (d, e, f). Thus, the dataset of Fig. 2.3c contains at least two transactions that can be associated with any pair of diagnosis codes a to e.

So far, we have discussed how to prevent identity disclosure, which is essential to comply with data sharing regulations [12, 50]. However, ensuring that patients will not be associated with *sensitive* diagnosis codes (i.e., diagnoses that can socially stigmatize patients) is also important. Examples of sensitive diagnosis codes are sexually transmitted diseases and drug abuse, as they are specified in related policies [62]. To see how sensitive information disclosure can be performed, consider Fig. 2.4a, in which the diagnosis codes e to g are sensitive and are denoted with bold letters. An attacker, who knows that a patient is diagnosed with a and b, can associate the patient with the sensitive diagnosis code e with a probability of $\frac{2}{3}$. Clearly, this may not be acceptable when a healthcare provider's policy requires the maximum probability of inferring a patient's sensitive diagnosis to be $\frac{1}{2}$.

Guarding against sensitive information disclosure has been the focus of two recent works in the data management community [43, 69].

(h, k, p)-coherence Xu et al. [69] introduced (h, k, p)-coherence, which treats diagnosis codes that can lead to identity disclosure (i.e., non-sensitive diagnosis codes) similarly to k^m-anonymity and additionally limits the probability of inferring sensitive diagnosis codes using a parameter h. Specifically, the function of parameter p is the same as m in k^m-anonymity, while h is expressed as a percentage. This anonymization principle is explained below.

Definition 2.6 ((h,k,p)-coherence). Given parameters h, k, and p, which are specified by data publishers, a dataset $\mathscr{D}$ satisfies (h,k,p)-coherence when $sup(I,\mathscr{D}) \geq k$, for each p-itemset I comprised of public items in $\mathscr{D}$, and $\frac{sup(I \cup j,\mathscr{D})}{sup(I,\mathscr{D})} \times 100\,\% \leq h$.

Thus, (h,k,p)-coherence can forestall both identity and sensitive information disclosure. To see this, observe that the dataset in Fig. 2.4b satisfies $(0.5,6,2)$-coherence, and, as such, it prevents an attacker, who knows any pair of diagnosis codes a to d, to infer any of the sensitive codes e to g, with a probability of more than 0.5. This principle assumes that all combinations of p non-sensitive diagnosis codes can lead to identity disclosure and that every diagnosis code needs protection from either identity or sensitive information disclosure. Thus, applying (h,k,p)-coherence in medical data publishing applications, in which only certain diagnosis codes are linkable to external data sources and specific diagnosis codes are sensitive, may unnecessarily incur a large amount of information loss.

ρ-uncertainty Another principle to guard against sensitive information disclosure, called ρ-uncertainty, was introduced by Cao et al. [43]. As can be seen from the definition below, ρ-uncertainty limits the probability of associating a patient with any of their sensitive diagnosis codes, using a threshold ρ.

Definition 2.7 (ρ-uncertainty). Given parameter ρ, which is specified by data publishers, a dataset $\mathscr{D}$ satisfies ρ-uncertainty when $\frac{sup(I \cup j,\mathscr{D})}{sup(I,\mathscr{D})} < \rho$, for each I-itemset in $\mathscr{I}$, where j is a sensitive item in $\mathscr{I}$ such that $j \notin I$.

Different from the aforementioned anonymization principles, ρ-uncertainty can be used to thwart attackers who can use any combination of items (either public or sensitive) to infer an individual's sensitive item. Also, due to the monotonicity of support, we have that $sup(I \cup J,\mathscr{D}) \leq sup(I \cup j,\mathscr{D})$, for every J such that $j \subseteq J$. This implies that ρ-uncertainty ensures that any combination of sensitive items that are not known to an attacker will receive protection as well. For instance, the dataset in Fig. 2.4b does not satisfy 0.5-uncertainty, because an attacker, who knows that an individual is associated with *abcd* and the sensitive item f, can infer that the individual is associated with another sensitive item e with a probability of 0.5. Unfortunately, however, enforcing ρ-uncertainty does not prevent identity disclosure. This implies that this principle is unsuited for being used in scenarios in which preventing identity disclosure is a legal requirement [12, 54, 64], such as those involving the publishing of diagnosis codes.

2.2.2 Generalization and Suppression Models

To enforce the aforementioned anonymization principles, generalization and suppression of items can be applied. The models that have been developed to perform these operations bear some similarity to those that have been proposed for relational

data [33,38], and they can be classified into *global* and *local* models. *Global* models require generalizing or suppressing all instances (i.e., occurrences) of an item in a transaction dataset in the same way, whereas *local* models do not impose this requirement. The dataset shown in Fig. 2.4b, for example, has been anonymized by applying a global generalization model to the dataset of Fig. 2.4a. Note that all instances of the items a to d have been generalized to (a,b,c,d). While local models are known to reduce information loss, they may lead to the construction of datasets that are difficult to be used in practice. This is because data mining algorithms and analysis tools cannot work effectively on these datasets [23].

A hierarchy-based model, which is similar to the full-subtree generalization model introduced by Iyengar [31] for relational data, was proposed by Terrovitis et al. [60]. This model assumes the existence of a generalization hierarchy, such as the one shown in Fig. 2.4c, and requires entire subtrees of original items (i.e., leaf-level nodes in the hierarchy) to be replaced by a unique internal node in the hierarchy. Consider, for example, the hierarchy in Fig. 2.4c. According to the model proposed in [69], a can be generalized to (a,b) or (a,b,c,d), but not to (a,c), as (a,c) is not represented as an internal node in the hierarchy. This model is not suitable for generalizing diagnosis codes, for two reasons. First, it unnecessarily restricts the number of possible generalizations, which may harm data utility [42]. Second, it is based on hierarchies, which, in the case of diagnosis codes, are either not well-designed (e.g., "too" coarse) or non-existent [56]. He et al. [28] applied the hierarchy-based model in a local manner, allowing different occurrences of the same item to be replaced by different generalized items. In a different line of research, Xu et al. [60] proposed applying global suppression to non-sensitive items, and pointed out that the latter operation has the important benefit of preserving the support of original non-suppressed items. Cao et al. [9] proposed a global suppression model that can be applied to both sensitive and not-sensitive items. Overall, generalization typically incurs a lower amount of information loss than suppression, and global generalization models are preferred due to their ability to preserve data utility in data analysis and mining applications.

2.2.3 Anonymization Algorithms

Similarly to the problem of k-anonymizing demographic data, applying the aforementioned principles to anonymize transaction data is NP-hard, when one needs to minimize information loss [42, 60]. Thus, a number of heuristic algorithms have been proposed to deal with this problem, and they can be classified based on the privacy principle they adopt, as illustrated in Table 2.4. In the following, we present these algorithms, reviewing the search and data transformation strategies they adopt.

Partition algorithm He et al. [28] proposed *Partition*, a top-down algorithm to enforce complete k-anonymity. As can be seen in the simplified version of Partition, shown in Algorithm 1, the algorithm gets as input an anonymized dataset $\tilde{\mathscr{D}}$,

Table 2.4 Summary of algorithms for preventing identity disclosure in transaction data publishing

Algorithm	Principle	Search strategy	Transformation
Partition [28]	Complete k-anonymity	Top-down partitioning	Local generalization
Apriori [60]	k^m-anonymity	Bottom-up traversal	Global generalization
LRA [61]	k^m-anonymity	Horizontal partitioning	Local generalization
VPA [61]	k^m-anonymity	Vertical partitioning	Global generalization
Greedy [24]	(h,k,p)-coherence	Greedy search	Global suppression (non-sensitive items)
SuppressControl [42]	ρ-uncertainty	Greedy search	Global suppression (any item)

Algorithm 1 Partition($\tilde{\mathscr{D}}, \mathscr{C}, \mathscr{H}, k$) [28]

input: Dataset $\tilde{\mathscr{D}}$, hierarchy cut $\mathscr{C}$, generalization hierarchy $\mathscr{H}$, parameter k
output: Complete k-anonymous dataset $\tilde{\mathscr{D}}'$
1. Start with the most generalized dataset $\tilde{\mathscr{D}}$
2. **if** complete k-anonymity is not satisfied
3. **return** $\tilde{\mathscr{D}}$
4. **else**
5. Find the node u in $\mathscr{H}$ that incurs minimum information loss when replaced by its immediate ascendants in $\mathscr{H}$
6. Update $\mathscr{C}$ by replacing u with its immediate ascendants
7. Update $\tilde{\mathscr{D}}$ based on $\mathscr{C}$
8. Create subpartitions of $\tilde{\mathscr{D}}'$ such that each of them contains all transactions in $\tilde{\mathscr{D}}'$ that have exactly the same generalized items
9. Balance the subpartitions so that each of them has at least k transactions
10. **for each** subpartition $\tilde{\mathscr{D}}''$
11. Execute *Partition*($\tilde{\mathscr{D}}, \mathscr{C}, \mathscr{H}, k$)

a generalization hierarchy $\mathscr{H}$, and a parameter k. $\tilde{\mathscr{D}}$ initially contains a single generalized item that appears in the root of the generalization hierarchy $\mathscr{H}$ and replaces all items. More specifically, $\tilde{\mathscr{D}}$ is constructed based on a hierarchy cut $\mathscr{C}$, i.e., a set of nodes in $\mathscr{H}$, such that every item in the domain $\mathscr{I}$ can be replaced by exactly one node in the set, according to the hierarchy-based generalization model. A hierarchy cut, for example, contains the nodes a, b, and (c,d) in the hierarchy of Fig. 2.4c. The algorithm proposed in [28] works by recursively partitioning $\tilde{\mathscr{D}}$, as long as complete k-anonymity is satisfied. In each execution, Partition is applied to a *subpartition* of at least k transactions in $\tilde{\mathscr{D}}$, which have the same generalized items, and the generalized items in these transactions are replaced by less general ones, in a way that reduces information loss. After Algorithm 1 terminates, all the constructed subpartitions satisfy complete k-anonymity and constitute a partition of the initial anonymized dataset. Thus, these subpartitions are combined into a publishable dataset (this process is straightforward and omitted from Algorithm 1, for clarity).

Partition starts by an anonymized dataset $\tilde{\mathscr{D}}$, in which all items are replaced by the most generalized item (step 1). If $\tilde{\mathscr{D}}$ does not satisfy complete k-anonymity, the

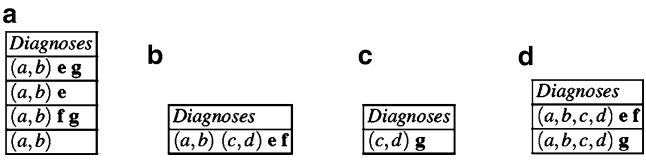

Fig. 2.5 Subpartitions created during the execution of *Partition*

Fig. 2.6 An example of (**a**) complete two-anonymous dataset, created by Partition, and (**b**) 2^2-anonymous dataset, created by Apriori

a

Diagnoses
(a,b,c,d) **e f**
(a,b) **e g**
(a,b) **e**
(a,b) **f g**
(a,b)
(a,b,c,d) **g**

b

Diagnoses
(a,b,c,d) **e f**
(a,b,c,d) **e g**
(a,b,c,d) **e**
(a,b,c,d) **f g**
(a,b,c,d)
(a,b,c,d) **g**

algorithm returns this dataset (steps 2 and 3). Otherwise, it revises the hierarchy cut $\mathscr{C}$ that corresponds to $\tilde{\mathscr{D}}$, by replacing a single node u in $\mathscr{H}$ (the one whose replacement incurs minimum information loss) with its immediate ascendants (steps 5 and 6). After that, Partition updates the transactions in $\tilde{\mathscr{D}}$, so that their generalized items are all contained in the updated hierarchy cut (step 7). This process creates a number of transactions in $\tilde{\mathscr{D}}$ that contain exactly the same generalized items with others. These transactions are identified by the Partition algorithm, which adds them into a subpartition (step 8). Subsequently, the resultant subpartitions are balanced, so that they contain at least k transactions (step 9). This involves redistributing transactions from subpartitions that have more than k transactions to others with fewer than k transactions, and potentially further generalization. Last, Partition is executed using each of these subpartitions as input (steps 10 and 11).

For example, consider applying Partition to anonymize the dataset in Fig. 2.4a, using $k = 2$, and assume that only non-sensitive items are generalized, based on the hierarchy shown in Fig. 2.4c. Initially, Partition is applied to a dataset $\tilde{\mathscr{D}}$ in which all transactions have the most general item (a,b,c,d), and the hierarchy cut contains only (a,b,c,d). The dataset $\tilde{\mathscr{D}}$ satisfies complete three-anonymity, and Parition replaces (a,b,c,d) with (a,b) and (c,d). This results in the three subpartitions, shown in Fig. 2.5a–c. Since the last two subpartitions contain fewer than k transactions, they are merged into the subpartition shown in Fig. 2.5d. Then, the algorithm is executed recursively, first for the subpartition of Fig. 2.5a and then for that of Fig. 2.5d. However, splitting any of these subpartitions further would violate complete two-anonymity, so the algorithm stops. The complete two-anonymous dataset in Fig. 2.6a, which is constructed by combining the subpartitions, can thus be safely released.

The Partition algorithm is efficient and effective for enforcing complete k-anonymity [28]. However, it is not particularly suited for anonymizing diagnosis codes with "low" information loss. This is because, in this setting, applying

Algorithm 2 Apriori($\tilde{\mathscr{D}}, \mathscr{H}, k, m$) [60]

input: Original dataset $\mathscr{D}$, generalization hierarchy $\mathscr{H}$, parameters k and m
output: k^m-anonymous dataset $\tilde{\mathscr{D}}$

1. $\tilde{\mathscr{D}} \leftarrow \mathscr{D}$
2. **for** $j = 1$ to m
3. **for each** transaction T in $\mathscr{D}$
4. Consider all the j-itemsets of T (generalized or not)
5. S $\leftarrow$ Find every j-itemset I that is supported by fewer than k transactions in $\mathscr{D}$
6. Construct all possible ways to generalize the itemsets in S according to $\mathscr{H}$
8. $\tilde{\mathscr{D}}' \leftarrow$ find the k^j-anonymous dataset that incurs minimum information loss
9. **return** $\tilde{\mathscr{D}}$

complete k-anonymity and hierarchy-based, local generalization may incur excessive information loss, as discussed above.

Apriori algorithm An iterative, bottom-up algorithm for enforcing k^m-anonymity, called *Apriori*, was proposed by Terrovitis et al. [60]. Since any superset of an itemset I has a support that is at most equal to that of I, it is possible for itemsets that need protection to be examined in a progressive fashion; from single items to m itemsets. Thus, Apriori generalizes larger itemsets, based on the way their subsets have been generalized [60]. Generalization is performed by traversing the hierarchy in a bottom-up, breadth-first way, using the hierarchy-based, global generalization model that was discussed above. The replacement of the items in an itemset with more general items (i.e., those in the upper levels of $\mathscr{H}$) can increase its support. This helps the enforcement of k^m-anonymity, but increases the level of information loss. Thus, Apriori starts from leaf-level nodes in the hierarchy and then examines the immediate ascendants of these items, one at a time. This is reminiscent to the strategy followed by the Apriori association rule algorithm [5].

An overview of Apriori is provided in Algorithm 2. The algorithm starts with the original dataset $\mathscr{D}$, which is assigned to $\tilde{\mathscr{D}}$, and performs m iterations (steps 1 and 2). In the j-th iteration, it identifies all possible j-itemsets that are not protected in $\tilde{\mathscr{D}}$ and then constructs a k^j-anonymous version $\tilde{\mathscr{D}}$ of $\mathscr{D}$ that incurs minimum information loss (steps 3–8). This is achieved with the use of a data structure, which stores the non-protected itemsets and their generalized counterparts and allows efficient itemset retrieval and support counting [60]. Subsequently, Apriori proceeds into the next iteration, and, after m iterations, it returns a k^m-anonymous dataset (step 10).

To exemplify, we discuss how Apriori can be applied to the dataset shown in Fig. 2.4a to enforce 2^2-anonymity (assume that only non-sensitive items are generalized). Given the hierarchy of Fig. 2.4c, Apriori considers original items first, but the dataset in Fig. 2.4a violates 2^2-anonymity. Thus, the algorithm attempts the generalization of a and b to (a,b) and that of c and d to (c,d). However, neither of these generalizations suffice to protect privacy, and Apriori eventually generalizes all non-sensitive items to (a,b,c,d). The resultant dataset, shown in Fig. 2.6b, is 2^2-anonymous and can be safely published.

LRA and VPA algorithms Terrovitis et al. [61] also proposed two efficient algorithms that are based on Apriori. The first of these algorithms, called *Local Recoding Anonymization* (LRA), splits $\mathscr{D}$ horizontally, so that the transactions in each subpartition share a large number of items and have a similar number of m-itemsets. Specifically, the transactions in $\mathscr{D}$ are sorted based on Gray ordering [26] and then grouped into subpartitions of approximately equal size. This strategy brings together transactions that will incur "low" information loss when anonymized. After that, a k^m-anonymous dataset is constructed by applying Apriori with the same k and m values, in each subpartition separately. LRA scales better with the size of dataset than Apriori, but still much time is spent to anonymize subpartitions that contain large transactions.

To address this issue and further improve efficiency, the authors of [61] proposed *Vertical Partitioning Algorithm* (VPA), which applies hierarchy-based, global generalization and works in two phases. In the first phase, the domain of items $\mathscr{I}$ is split into subpartitions $\mathscr{I}_1, \ldots, \mathscr{I}_l$ that contain items whose common ancestor lies at a certain level in the hierarchy. For example, partitioning together items whose common ancestor lies at the second level of the hierarchy that is shown in Fig. 2.4c, yields the subpartitions (a,b) and (c,d). This process creates a number of datasets $\mathscr{D}_1, \ldots, \mathscr{D}_l$, each containing one subpartition of $\mathscr{I}$. Then, Apriori is applied with the same k and m values to each of the latter datasets. However, the entire dataset may not be k^m-anonymous, if there are item combinations that span multiple subpartitions of $\mathscr{I}$. Thus, in the second phase, VPA constructs a k^m-anonymous dataset by applying Apriori in the dataset that contains all generalized items created during the previous phase.

LRA and VPA are significantly faster than Apriori and achieve a comparable result in terms of information loss [61]. However, they enforce k^m-anonymity, using hierarchy-based generalization, which makes them unsuited for being applied to anonymize diagnosis codes, as mentioned in Sects. 2.2.1 and 2.2.2, respectively.

We now review two suppression-based algorithms, which provide protection from sensitive information disclosure.

Greedy algorithm Xu et al. [69] proposed Greedy, an algorithm that employs suppression to enforce (h,k,p)-coherence. Central to this algorithm is the notion of *mole*, which is defined below.

Definition 2.8 (Mole). Given an original dataset $\mathscr{D}$, and values for the parameters h, k, and p, a *mole* is defined as an itemset I, comprised of public items in $\mathscr{I}$, such that $sup(I,\mathscr{D}) < k$ and $\frac{sup(I \cup j,\mathscr{D})}{sup(I,\mathscr{D})} > h$, for each sensitive item j in $\mathscr{I}$.

Clearly, no (h,k,p)-coherent dataset contains a mole, and item suppression helps the elimination of moles. At the same time, suppression incurs information loss, which needs to be kept at a minimum to preserve the utility of the published data. Thus, Greedy works by iteratively removing public items from a dataset, until the resultant dataset satisfies (h,k,p)-coherence.

As can be seen in the simplified version of Greedy that is presented in Algorithm 3, this algorithm starts by assigning $\mathscr{D}$ to a dataset $\tilde{\mathscr{D}}$ and then suppresses

Algorithm 3 Greedy$(\mathscr{D}, h, k, p)$ [69]

input: Original dataset $\mathscr{D}$, parameters h, k, and p
output: (h,k,p)-coherent dataset $\tilde{\mathscr{D}}$
1. $\tilde{\mathscr{D}} \leftarrow \mathscr{D}$
2. **for each** 1-itemset I in $\tilde{\mathscr{D}}$ that is a mole
2. $\quad \tilde{\mathscr{D}} \leftarrow \tilde{\mathscr{D}} \setminus I$
3. **while** there exists a mole I in $\tilde{\mathscr{D}}$
4. $\quad$ **for each** public item i in $\tilde{\mathscr{D}}$
5. $\quad\quad MM(i) \leftarrow$ the number of moles in $\tilde{\mathscr{D}}$ that contain item i
6. $\quad\quad IL(i) \leftarrow$ information loss of i
7. $\quad$ find public item i in $\mathscr{I}$ with the maximum $\frac{MM(i)}{IL(i)}$
8. $\quad$ suppress i from all transactions in $\tilde{\mathscr{D}}$
9. **return** $\tilde{\mathscr{D}}$

a

Diagnoses
a b **e f**
a b **e g**
a **e**
b **f g**
a b
g

b

Sensitive Association Rules		
$a \rightarrow$ **e**	$a \rightarrow$ **f**	$a \rightarrow$ **g**
$b \rightarrow$ **e**	$b \rightarrow$ **f**	$b \rightarrow$ **g**
$c \rightarrow$ **e**	$c \rightarrow$ **f**	$d \rightarrow$ **e**
$d \rightarrow$ **f**	$e \rightarrow$ **f**	**e** $\rightarrow$ **g**
f $\rightarrow$ **g**	**f** $\rightarrow$ **e**	**g** $\rightarrow$ **e**
g $\rightarrow$ **f**		

c

Diagnoses
a b c d
a b **g**
a
b **g**
a b
c

d

Diagnoses
a c d
a **g**
a
g
a
c

Fig. 2.7 An example of (**a**) $(0.5,2,2)$-coherent dataset produced by Greedy [69], (**b**) SARs used by SuppressControl [9], (**c**) intermediate dataset produced by SuppressControl [9], and (**d**) 0.5-uncertain dataset produced by SuppressControl

all moles of size 1 from $\tilde{\mathscr{D}}$ (steps 1 and 2). Next, Greedy iterates over all public items in $\tilde{\mathscr{D}}$, and suppresses the item i with the largest ratio between $MM(i)$, the number of items that contain i, and $IL(i)$, the amount of information loss that suppressing i incurs (steps 3–8). Finding i is performed efficiently using a data structure that organizes moles similarly to the way frequent itemsets are stored in an FP-tree [27]. As for the score IL for an item, it is either determined by data publishers, or set to $sup(i,\mathscr{D})$. The process of suppressing items ends when $\tilde{\mathscr{D}}$ satisfies (h,k,p)-coherence, and, after that, Greedy returns $\tilde{\mathscr{D}}$ (step 9).

To see how Greedy works, consider applying it to the dataset of Fig. 2.4a using $h = 0.5$, $k = 2$, and $p = 2$, when $IL = sup(i,\mathscr{D})$, for each of the public items a to d. The algorithm starts by suppressing d, as it is supported by a single transaction. Then, it suppresses c, because $\frac{MM(c)}{IL(c)} = \frac{3}{2}$ is larger than the corresponding fractions of as all other public items. This suffices to satisfy $(0.5,2,2)$-coherence, hence Greedy returns the dataset shown in Fig. 2.7a.

However, Greedy may incur significant information loss if applied to protect diagnosis codes for two reasons. First, it employs (h,k,p)-coherence, which does not take into account detailed privacy requirements that are common in medical data publishing (see Sect. 2.2.1). Second, Greedy uses suppression, which is a rather drastic operation compared to generalization. For instance, enforcing $(0.5,6,2)$-coherence using Greedy requires suppressing all public items. On the other hand,

there are generalized datasets, such as the one in Fig. 2.4b, that satisfy $(0.5, 6, 2)$-coherence, while incurring much lower information loss.

SuppressControl algorithm Cao et al. [9] have proposed *SuppressControl*, a greedy, suppression-based algorithm to enforce ρ-uncertainty. Central to this algorithm is the notion of *Sensitive Association Rule* (SAR) that is defined below.

Definition 2.9 (Sensitive Association Rule (SAR)). Given an original dataset $\mathscr{D}$, and a value for the parameters ρ, a *sensitive association rule* is defined as an implication $I \rightarrow j$, where I is an itemset in $\mathscr{I}$, called the *antecedent* of $I \rightarrow j$, and j is a sensitive item in $\mathscr{I}$ such that $j \notin I$, called the *consequent* of $I \rightarrow j$.

Given a dataset $\mathscr{D}$ and a set of SARs, the dataset satisfies ρ-uncertainty when, for every SAR $I \rightarrow j$, we have $\frac{sup(I \cup j, \mathscr{D})}{sup(I, \mathscr{D})} \leq \rho$, as can be seen from Definition 2.7. Thus, SuppressControl considers each SAR that can be constructed from the items in $\mathscr{D}$ and suppresses one or more items in the SAR, from all transactions in the latter dataset, until $\mathscr{D}$ satisfies ρ-uncertainty.

Specifically, the algorithm works iteratively, as follows. In the i-th iteration, it finds a set of SARs S whose antecedents contain exactly i items. If such a set cannot be constructed, SuppressControl returns $\tilde{\mathscr{D}}$ (steps 1–5). Otherwise, it updates S by discarding every SAR that does not violate ρ-uncertainty (steps 6 and 7). Next, SuppressControl iterates over all SARs in S, and suppresses items in them, starting with the item l that has the maximum ratio between the number of SARs that contain l and $sup(l, \tilde{\mathscr{D}})$ (steps 9–12). After suppressing l, SuppressControl updates $\mathscr{S}$ by removing all SARs that contain this item (steps 13 and 14), and proceeds into considering the next SAR in $\mathscr{S}$, if there is one. Otherwise, the algorithm proceeds to the next iteration, in which SARs with antecedents larger by one item than those of the SARs considered before, are examined. Last, when all SARs that need protection have been considered, SuppressControl returns $\tilde{\mathscr{D}}$, which satisfies ρ-uncertainty (step 15).

As an example, consider applying SuppressControl to the dataset of Fig. 2.4a, using $\rho = 0.5$. The algorithm starts by constructing all the antecedents of SARs that are comprised of 1 item in this dataset (i.e, a to $\mathbf{g}$), and then discards the SARs that do not need protection, which are highlighted in Fig. 2.7c (steps 1–7). Then, SuppressControl computes the ratios between the NI and support scores for all items, and suppresses the sensitive item $\mathbf{f}$, which has the maximum ratio $\frac{NI(\mathbf{f})}{sup(\mathbf{f}, \tilde{\mathscr{D}})} = \frac{6}{3} = 2$. In this case, the corresponding ratio for $\mathbf{f}$ is also 2, and SuppressControl breaks the tie arbitrarily. Next, the algorithm updates the set of SARs $\mathscr{S}$ by discarding the SARs that contain $\mathbf{f}$ in Fig. 2.7b. After that, the algorithm suppresses the item $\mathbf{e}$ and discards the SARs that contain this item in Fig. 2.7b. At this point, $\tilde{\mathscr{D}}$ is as shown in Fig. 2.7c, and S is empty. Thus, SuppressControl proceeds into the next iteration, in which SAR considers $ab \rightarrow \mathbf{g}$, the only SAR that contains two items in its antecedent and can be formed, based on $\tilde{\mathscr{D}}$. To protect this SAR, the algorithm suppresses b and returns the dataset in Fig. 2.7d, which satisfies 0.5-uncertainty.

Algorithm 4 SuppressControl($\mathscr{D}, \rho$) [9]

input: Original dataset $\mathscr{D}$, parameter ρ
output: Dataset $\tilde{\mathscr{D}}$ that satisfies ρ uncertainty

$\tilde{\mathscr{D}} \leftarrow \mathscr{D}$
for each i from 1 to $|\mathscr{I}|$
  $\mathsf{S} \leftarrow$ the antecedents of all SARs that contain i items
  if $\mathsf{S} = \varnothing$
    return $\tilde{\mathscr{D}}$
  for each SAR $I \rightarrow j$ such that $\frac{sup(I \cup j, \tilde{\mathscr{D}})}{sup(I, \tilde{\mathscr{D}})} \leq \rho$
    $\mathsf{S} \leftarrow \mathsf{S} \setminus \{I \rightarrow j\}$
  while $\mathsf{S} \neq \varnothing$
    for each item l contained in an SAR in S
      $NI(l) \leftarrow$ the number of SARs in S that contain item i
    find the item l with the maximum $\frac{NI(l)}{sup(l, \tilde{\mathscr{D}})}$
    suppress l from all transactions in $\tilde{\mathscr{D}}$
    $\mathsf{S}_l \leftarrow$ find all SARs in S that contain the item l
    $\mathsf{S} \leftarrow \mathsf{S} \setminus \mathsf{S}_l$
return $\tilde{\mathscr{D}}$

2.3 Anonymizing Genomic Data

While de-identification and anonymization of demographics and diagnosis codes guard against linkage attacks, an individual's record may be distinguishable with respect to genomic data [48]. Lin et al. [39], for example, estimated that an individual is unique with respect to approximately 100 single nucleotide polymorphisms (SNPs), i.e., DNA sequence variations occurring when a single nucleotide in the genome differs between paired chromosomes in an individual. Meanwhile, genomic sequences contain potentially sensitive information, including the ancestral origin of an individual [55] or genetic information about their family members [10], which are likely to be abused, if linked to an individual's identity [57]. To prevent such inferences, only aggregate statistics related to individuals' genetic information were deposited into the public section of dbGaP repository.

However, Homer et al. [29] have shown that such aggregate statistics may still allow an attacker to infer whether an identified individual belongs to the case or control group of a Genome-Wide Association Study (GWAS) (i.e., if the individual is diagnosed with a GWAS-related disease or not). To achieve this, an attacker needs access to an individual's DNA and to a reference pool of DNA from individuals of the same genetic population as the identified individual (e.g., the publicly available data from the HapMap project.[3]) This allows the attacker to compare the identified individual's SNP profile against the Minor Allele Frequencies (MAFs) [4] of the DNA

[3] http://hapmap.ncbi.nlm.nih.gov/

[4] Minor Allele Frequencies (MAFs) are the frequencies at which the less common allele occurs in a given population.

mixture (e.g., the case group in a GWAS) and the reference population, and then to statistically assess the presence of the individual in the "mixture".

The NIH and Wellcome Trust responded to the findings of Homer et al. quickly, by removing genomic summaries of case and control cohorts from the public section of databanks, such as dbGaP [70], while further research investigated the feasibility of Homer's attack [7, 65, 71]. Wang et al. [65] noted that attackers may not have access to MAFs (e.g., when other test statistics are published instead) or to large numbers of independent SNPs from the identified individual and their corresponding allele frequencies from the mixture, which are required for Homer's attack to succeed. Furthermore, Brown et al. [7] showed that many individuals can be wrongly identified as belonging to the case group, because the assumptions about adversarial knowledge made in [29] may not hold in practice. Wang et al. [65] introduced two other attacks that are applicable to aggregate statistics [65]; one that can statistically determine the presence of an individual in the case group, based upon the r^2 measure of the correlation between alleles, and another that allows the inference of the SNP sequences of many individuals that are present in the GWAS data, based on correlations between SNPs.

Recently, Fienberg et al. [21] examined how aggregated genomic data may be published without compromising individuals' privacy, based on *differential privacy* [15]. The latter principle requires computations to be insensitive to changes in any particular individual's data and can be used to provide privacy, as mentioned in Introduction. This is because, differentially private data do not allow an attacker to make inferences about an identified individual that they could not make if the individual's record was absent from the original dataset. In [21], two methods for releasing aggregate statistics for GWAS in a differentially private way were proposed. The first method focuses on the publication of the χ^2 statistic and p-values and works by adding Laplace noise to the original statistics, while the second method allows releasing noisy versions of these statistics for the most relevant SNPs.

References

1. Adam, N., Worthmann, J.: Security-control methods for statistical databases: a comparative study. ACM Comput. Surv. **21**(4), 515–556 (1989)
2. Aggarwal, C., Yu, P.: A condensation approach to privacy preserving data mining. In: EDBT, pp. 183–199 (2004)
3. Aggarwal, C.C.: On k-anonymity and the curse of dimensionality. In: VLDB, pp. 901–909 (2005)
4. Aggarwal, G., Kenthapadi, F., Motwani, K., Panigrahy, R., Zhu, D.T.A.: Approximation algorithms for k-anonymity. Journal of Privacy Technology (2005)
5. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB, pp. 487–499 (1994)
6. Bayardo, R., Agrawal, R.: Data privacy through optimal k-anonymization. In: 21st ICDE, pp. 217–228 (2005)
7. Braun, R., Rowe, W., Schaefer, C., Zhang, J., Buetow, K.: Needles in the haystack: identifying individuals present in pooled genomic data. PLoS Genetocs **5**(10), e1000,668 (2009)

8. Byun, J., Kamra, A., Bertino, E., Li, N.: Efficient k-anonymity using clustering technique. In: DASFAA, pp. 188–200 (2007)
9. Cao, J., Karras, P., Kalnis, P., Tan, K.L.: Sabre: a sensitive attribute bucketization and redistribution framework for t-closeness. VLDBJ **20**, 59–81 (2011)
10. Cassa, C., Schmidt, B., Kohane, I., Mandl, K.D.: My sister's keeper? genomic research and the identifiability of siblings. BMC Medical Genomics **1**, 32 (2008)
11. Chen, B., Ramakrishnan, R., LeFevre, K.: Privacy skyline: Privacy with multidimensional adversarial knowledge. In: VLDB, pp. 770–781 (2007)
12. Medical Research Council: MRC data sharing and preservation initiative policy. http://www.mrc.ac.uk/ourresearch/ethicsresearchguidance/datasharinginitiative (2006)
13. Domingo-Ferrer, J., Mateo-Sanz, J.M.: Practical data-oriented microaggregation for statistical disclosure control. IEEE Trans. on Knowledge and Data Engineering **14**(1), 189–201 (2002)
14. Domingo-Ferrer, J., Torra, V.: Ordinal, continuous and heterogeneous k-anonymity through microaggregation. DMKD **11**(2), 195–212 (2005)
15. Dwork, C.: Differential privacy. In: ICALP, pp. 1–12 (2006)
16. Emam, K.E.: Methods for the de-identification of electronic health records for genomic research. Genome Medicine **3**(4), 25 (2011)
17. Emam, K.E., Dankar, F.K.: Protecting privacy using k-anonymity. Journal of the American Medical Informatics Association **15**(5), 627–637 (2008)
18. Emam, K.E., Dankar, F.K., et al.: A globally optimal k-anonymity method for the de-identification of health data. Journal of the American Medical Informatics Association **16**(5), 670–682 (2009)
19. Farkas, C., Jajodia, S.: The inference problem: a survey. SIGKDD Explorations **4**(2), 6–11 (2002)
20. Federal Committee on Statistical Methodology: Report on statistical disclosure limitation methodology. http://www.fcsm.gov/working-papers/totalreport.pdf (2005)
21. Fienberg, S.E., Slavkovic, A., Uhler, C.: Privacy preserving gwas data sharing. In: IEEE ICDM Worksops, pp. 628–635 (2011)
22. Friedman, J., Bentley, J., Finkel, R.: An algorithm for finding best matches in logarithmic time. ACM Trans. on Mathematical Software **3**(3) (1977)
23. Fung, B.C.M., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: A survey on recent developments. ACM Comput. Surv. **42** (2010)
24. Gkoulalas-Divanis, A., Loukides, G.: PCTA: Privacy-constrained Clustering-based Transaction Data Anonymization. In: EDBT PAIS, p. 5 (2011)
25. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: SIGMOD '84, pp. 47–57 (1984)
26. Hamming, R.W.: Coding and Information Theory. Prentice-Hall (1980)
27. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: SIGMOD, pp. 1–12 (2000)
28. He, Y., Naughton, J.F.: Anonymization of set-valued data via top-down, local generalization. PVLDB **2**(1), 934–945 (2009)
29. Homer, N., Szelinger, S., Redman, M., et al.: Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. PLoS Genetics **4**(8), e1000,167 (2008)
30. Iwuchukwu, T., Naughton, J.F.: K-anonymization as spatial indexing: Toward scalable and incremental anonymization. In: VLDB, pp. 746–757 (2007)
31. Iyengar, V.S.: Transforming data to satisfy privacy constraints. In: KDD, pp. 279–288 (2002)
32. Koudas, N., Zhang, Q., Srivastava, D., Yu, T.: Aggregate query answering on anonymized tables. In: ICDE '07, pp. 116–125 (2007)
33. LeFevre, K., DeWitt, D., Ramakrishnan, R.: Incognito: efficient full-domain k-anonymity. In: SIGMOD, pp. 49–60 (2005)
34. LeFevre, K., DeWitt, D., Ramakrishnan, R.: Mondrian multidimensional k-anonymity. In: ICDE, p. 25 (2006)

35. LeFevre, K., DeWitt, D., Ramakrishnan, R.: Workload-aware anonymization. In: KDD, pp. 277–286 (2006)
36. Li, J., Wong, R., Fu, A., Pei, J.: Achieving -anonymity by clustering in attribute hierarchical structures. In: DaWaK, pp. 405–416 (2006)
37. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: ICDE, pp. 106–115 (2007)
38. Li, T., Li, N.: Towards optimal k-anonymization. DKE **65**, 22–39 (2008)
39. Lin, Z., Altman, R.B., Owen, A.: Confidentiality in genome research. Science **313**(5786), 441–442 (2006)
40. Loukides, G., Denny, J., Malin, B.: The disclosure of diagnosis codes can breach research participants' privacy. Journal of the American Medical Informatics Association **17**, 322–327 (2010)
41. Loukides, G., Gkoulalas-Divanis, A., Malin, B.: Anonymization of electronic medical records for validating genome-wide association studies. Proceedings of the National Academy of Sciences **17**(107), 7898–7903 (2010)
42. Loukides, G., Gkoulalas-Divanis, A., Malin, B.: COAT: Constraint-based anonymization of transactions. KAIS **28**(2), 251–282 (2011)
43. Loukides, G., Gkoulalas-Divanis, A., Shao, J.: Anonymizing transaction data to eliminate sensitive inferences. In: DEXA, pp. 400–415 (2010)
44. Loukides, G., Shao, J.: Capturing data usefulness and privacy protection in k-anonymisation. In: SAC, pp. 370–374 (2007)
45. Loukides, G., Shao, J.: Preventing range disclosure in k-anonymised data. Expert Systems with Applications **38**(4), 4559–4574 (2011)
46. Loukides, G., Tziatzios, A., Shao, J.: Towards preference-constrained -anonymisation. In: DASFAA International Workshop on Privacy- Preserving Data Analysis (PPDA), pp. 231–245 (2009)
47. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. In: ICDE, p. 24 (2006)
48. Malin, B., Loukides, G., Benitez, K., Clayton, E.: Identifiability in biobanks: models, measures, and mitigation strategies. Human Genetics **130**(3), 383–392 (2011)
49. Meyerson, A., Williams, R.: On the complexity of optimal k-anonymity. In: PODS, pp. 223–228 (2004)
50. National Institutes of Health: Policy for sharing of data obtained in NIH supported or conducted genome-wide association studies. NOT-OD-07-088. 2007.
51. Nergiz, M.E., Clifton, C.: Thoughts on k-anonymization. DKE **63**(3), 622–645 (2007)
52. Ohno-Machado, L., Vinterbo, S., Dreiseitl, S.: Effects of data anonymization by cell suppression on descriptive statistics and predictive modeling performance. Journal of American Medical Informatics Association **9**(6), 115119 (2002)
53. Park, H., Shim, K.: Approximate algorithms for k-anonymity. In: SIGMOD, pp. 67–78 (2007)
54. European Parliament, C.: EU Directive on privacy and electronic communications. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:EN:NOT (2002)
55. Phillips, C., Salas, A., Sanchez, J., et al.: Inferring ancestral origin using a single multiplex assay of ancestry-informative marker snps. Forensic Science International: Genetics **1**, 273–280 (2007)
56. Rodgers, J.: Quality assurance and medical ontologies. Methods of Information in Medicine **45**(3), 267–274 (2006)
57. Rothstein, M., Epps, P.: Ethical and legal implications of pharmacogenomics. Nature Review Genetics **2**, 228–231 (2001)
58. Samarati, P.: Protecting respondents identities in microdata release. TKDE **13**(9), 1010–1027 (2001)
59. Sweeney, L.: k-anonymity: a model for protecting privacy. IJUFKS **10**, 557–570 (2002)
60. Terrovitis, M., Mamoulis, N., Kalnis, P.: Privacy-preserving anonymization of set-valued data. PVLDB **1**(1), 115–125 (2008)

61. Terrovitis, M., Mamoulis, N., Kalnis, P.: Local and global recoding methods for anonymizing set-valued data. VLDB J **20**(1), 83–106 (2011)
62. Texas Department of State Health Services: User manual of texas hospital inpatient discharge public use data file. http://www.dshs.state.tx.us/THCIC/ (2008)
63. Truta, T.M., Campan, A., Meyer, P.: Generating microdata with p -sensitive k -anonymity property. In: Secure Data Management, pp. 124–141 (2007)
64. U.S. Department of Health and Human Services Office for Civil Rights: HIPAA administrative simplification regulation text (2006)
65. Wang, R., Li, Y.F., Wang, X., Tang, H., Zhou, X.: Learning your identity and disease from research papers: information leaks in genome wide association study. In: CCS, pp. 534–544 (2009)
66. Wong, R.C., Li, J., Fu, A., K.Wang: alpha-k-anonymity: An enhanced k-anonymity model for privacy-preserving data publishing. In: KDD, pp. 754–759 (2006)
67. Xiao, X., Tao, Y.: Personalized privacy preservation. In: SIGMOD, pp. 229–240 (2006)
68. Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Fu, A.W.C.: Utility-based anonymization using local recoding. In: KDD, pp. 785–790 (2006)
69. Xu, Y., Wang, K., Fu, A.W.C., Yu, P.S.: Anonymizing transaction databases for publication. In: KDD, pp. 767–775 (2008)
70. Zerhouni, E.A., Nabel, E.: Protecting aggregate genomic data. Science **322**(5898) (2008)
71. Zhou, X., Peng, B., Li, Y.F., Chen, Y., Tang, H., Wang, X.: To release or not to release: evaluating information leaks in aggregate human-genome data. In: ESORICS, pp. 607–627 (2011)

http://www.springer.com/978-1-4614-5667-4

Anonymization of Electronic Medical Records to Support Clinical Analysis

Gkoulalas-Divanis, A.; Loukides, G.

2013, XV, 72 p. 23 illus., Softcover
ISBN: 978-1-4614-5667-4